Nonparametric Monitoring of Multivariate Data via KNN Learning

Wendong Li[a], Chi Zhang[b], Fugee Tsung[b] and Yajun Mei[c]

[a] School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, China; [b] Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology, Kowloon, Hong Kong; [c] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, USA

ARTICLE HISTORY
Compiled August 9, 2020

ABSTRACT
Process monitoring of multivariate quality attributes is important in many industrial applications, in which rich historical data are often available thanks to modern sensing technologies. While multivariate statistical process control (SPC) has been receiving increasing attention, existing methods are often inadequate as they are sensitive to the parametric model assumptions of multivariate data. In this paper, we propose a novel, nonparametric k-nearest neighbors empirical cumulative sum (KNN-ECUSUM) control chart that is a machine-learning-based black-box control chart for monitoring multivariate data by utilizing extensive historical data under both in-control and out-of-control scenarios. Our proposed method utilizes the k-nearest neighbors (KNN) algorithm for dimension reduction to transform multivariate data into univariate data, and then applies the CUSUM procedure to monitor the change on the empirical distribution of the transformed univariate data. Extensive simulation studies and a real industrial example based on a disk monitoring system demonstrate the robustness and effectiveness of our proposed method.

KEYWORDS
Multivariate statistical process control; KNN algorithm; machine learning; categorical variable; CUSUM; empirical probability mass function

1. Introduction

Due to the rapid development of sensing technologies, modern industrial processes are generating large amounts of real-time measurements taken by many different kinds of sensors to reflect system status. For instance, in chemical factories, the boiler temperature, air pressure and chemical action time are all recorded in real time during the boiler heating process. For the purpose of quality control, it is important to take advantage of these real-time multivariate measurements and develop efficient statistical methods that can detect undesired events as quickly as possible before the catastrophic failure of the system.

The early detection of undesired events has been investigated in the subfield of statistical process control (SPC), and the corresponding statistical methods are often referred to as control charts. In general, a control chart computes a real-valued monitoring statistic at each time step, and raises an alarm whenever this monitoring

The first two authors contributed equally to this paper. CONTACT: Yajun Mei. Email: ymei@isye.gatech.edu

statistic exceeds a pre-specified control limit. In the SPC framework, the performance of a control chart is often measured by two kinds of average run lengths (ARLs): one is in-control (IC) ARL, and the other is out-of-control (OC) ARL. Here the IC and OC ARLs are defined as the average amount of time steps (or the average number of observations) from the start of monitoring to the first alarm, respectively, when the system is IC or OC. For a given IC ARL, a control chart with a smaller OC ARL will be able to monitor processes more efficiently.

In the context of monitoring multivariate data, many control charts have been developed in the literature under certain assumptions on the data distributions, see Lowry and Montgomery (1995), Lu et al. (1998) and Woodall and Montgomery (2014) for excellent literature reviews. To be more specific, there are two families of multivariate control charts. The first one is parametric, e.g., under the assumption that the data are multivariate normally distributed. These include the multivariate cumulative sum (MCUSUM, Woodall and Ncube 1985; Crosier 1988), the multivariate exponentially weighted moving average (MEWMA, Lowry et al. 1992), the regression-adjusted control chart (Hawkins 1991) and charts based on variable selection (Zou and Qiu 2009; Wang and Jiang 2009). The second family is nonparametric or model-free control chart that often makes a less restrictive assumption on the data distribution, e.g., the data distribution is unimodal or symmetric under the null, see Chakraborti, Van, and Bakir (2001) and Qiu (2018) for reviews. A selective list of references include Sun and Tsung (2003), Hwang, Runger, and Tuv (2007), Camci, Chinnam, and Ellis (2008), Qiu (2008), Sukchotrat, Kim, and Tsung (2010), Ning and Tsung (2012), Zou, Wang, and Tsung (2012), and Li et al. (2017). In particular, for simplicity and comparison purpose, the multivariate sign exponentially weighted moving average (MSEWMA) control chart proposed by Zou and Tsung (2011) will be chosen as the baseline method in our paper.

In this paper, we develop a nonparametric or distribution-free control chart for monitoring multivariate data. Our approach is different from the existing nonparametric SPC methods in the sense that we adopt a machine-learning-type black-box approach for monitoring. Instead of making any model assumptions, we assume that rich historical in-control (IC) and out-of-control (OC) data are available. Such assumption on data is reasonable in many modern processes thanks to the development of modern sensing technology. A concrete motivating example of our paper is the hard disk drive monitoring system (HDDMS), a computerized system that records various attributes related to disk failure and provides early warnings of disk degradation. In such an application, both IC and OC data are available in the historical dataset, and the challenge is how to take full advantage of these historical IC and OC data for effective online monitoring. In the traditional SPC, or more generally, statistical, literature, extensive data are often used to build multivariate models, which are then used for monitoring. Here we propose to bypass the model of the original multivariate data, and develop the control charts directly based on the historical data themselves through machine-learning techniques.

To be more specific, our proposed method applies the k-nearest neighbors (KNN) algorithm for dimension reduction to convert original multivariate data into one-dimensional categorical random variables, as monitoring univariate data is well studied in the SPC literature. For simplicity, here we use the CUSUM procedure to monitor the transformed one-dimensional categorical variables based on the estimated empirical probability mass function (p.m.f.) under both IC and OC states, although many other control charts can also be combined with our proposed KNN-based dimension reduction.

Our proposed monitoring scheme has the following advantages: (i) it is robust, data-driven and thus can be easily adapted to monitor any multivariate or high-dimensional data; (ii) it takes full advantage of historical IC and OC data and holds desirable performance; (iii) it is statistically appealing since it is based on the empirical p.m.f. of the derived one-dimensional categorical variable; (iv) it is easy to interpret if one treats our key idea of KNN learning as dimension reduction. This opens a new research direction when monitoring high-dimensional data in the SPC literature by adopting modern machine learning techniques as a dimension reduction tool combined with existing multivariate or univariate control charts.

The remainder of this paper is organized as follows. The problem of monitoring multivariate data is stated in Section 2, and our proposed KNN-based control chart is presented in Section 3. Extensive simulation studies are reported in Section 4, and the case study involving the HDDMS data is presented in Section 5. Several concluding remarks are made in Section 6, and some of the technical or mathematical details are provided in the appendix.

2. Problem Description and Literature Review

Following the standard SPC literature, we monitor a sequence of p-dimensional random vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i, ...\}$ over time from some process. Initially the process is in the IC state, and the cumulative distribution function (c.d.f.) of the $\boldsymbol{x}_i$ is $F(\boldsymbol{x}; \boldsymbol{\theta}_0)$ for some IC parameter $\boldsymbol{\theta}_0$. At some unknown time point τ, the process is OC in the sense that the $\boldsymbol{x}_i$s have another c.d.f. $F(\boldsymbol{x}; \boldsymbol{\theta}_1)$ for some OC parameter $\boldsymbol{\theta}_1$. In other words, the distributions of multivariate data $\boldsymbol{x}_i$ can be described by the following change-point model:

$$\boldsymbol{x}_i \sim \begin{cases} F(\boldsymbol{x}; \boldsymbol{\theta}_0) & \text{for } \ i = 1, \ldots, \tau \\ F(\boldsymbol{x}; \boldsymbol{\theta}_1) & \text{for } \ i = \tau + 1, \ldots \end{cases} \tag{1}$$

Here we assume that the data $\boldsymbol{x}_i$s are independent over time, but the p components within a given data can be cross-correlated. That is, in terms of spatial-temporal data analysis, we assume that data are correlated over the spatial domain, and independent over the time domain. In practice, the temporal independence assumption might not be as restrictive as one might think, since one can monitor independent residuals from some time series model that de-correlates the temporal correlation.

When monitoring the multivariate data $\boldsymbol{x}_i$, our goal is to raise an alarm as soon as possible after the process changes from the IC state to the OC state at time τ. This can be formulated as testing the simple null hypothesis $H_0 : \tau = \infty$ (i.e., no change) against the composite alternative hypothesis $H_1 : \tau = 1, 2, \cdots$ (i.e., a change occurs at some finite time), but with the twist that we must test these hypotheses at each time point until we feel we have enough evidence to reject the null hypothesis and declare that a change has occurred. To be more specific, a control chart raises an OC alarm at time point T based on the first T observations, and the expectation, $\mathbf{E}(T)$ is often referred as the IC or OC average run length (ARL), depending on whether the process is in the IC or OC state. One would like to develop a control chart that minimizes the OC ARL, $\mathbf{E}_{oc}(T)$, subject to the following false alarm constraint under the IC state:

$$\mathbf{E}_{ic}(T) \geq \gamma. \tag{2}$$

where $\gamma > 0$ is a pre-specified constant. In other words, the control chart is required to process at least γ observations on average before raising a false alarm when all observations are IC.

This problem is well studied in cases when the distributions of $\boldsymbol{x}_i$ under the IC and OC states are fully specified, and one classical method is the CUSUM procedure developed by Page (1954). To define the CUSUM procedure, denote by $g_{ic}(\cdot)$ and $g_{oc}(\cdot)$ the probability density functions (p.d.f.s) of the $\boldsymbol{x}_i$ under the IC and OC states, respectively. Next, define the CUSUM statistic recursively over time t as

$$W_t = \max\{W_{t-1} + \log \frac{g_{oc}(\boldsymbol{x}_t)}{g_{ic}(\boldsymbol{x}_t)}, 0\},\ t = 1, 2, \cdots, \tag{3}$$

where $W_0 = 0$. Then the CUSUM procedure raises an alarm at the first time point t whenever the CUSUM statistic $W_t \geq L$, where the pre-specified threshold $L > 0$ is chosen to satisfy the IC ARL constraint in (2). It is well known that the CUSUM statistic W_t is actually the logarithm of the generalized likelihood ratio statistic of all observations up to time point t, $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t\}$, and the CUSUM procedure enjoys certain exactly optimal properties (Moustakides 1986). Unfortunately the CUSUM procedure requires complete information of the IC and OC p.d.f.s, $g_{ic}(\cdot)$ and $g_{oc}(\cdot)$, and thus it may not be feasible in practice when it is nontrivial to model the distributions of multivariate data.

As mentioned in the introduction, nonparametric control charts have been developed in the SPC literature. For instance, Qiu and Hawkins (2003) proposed a nonparametric CUSUM chart using the anti-ranks of observations. Boone and Chakraborti (2012) proposed two multivariate Shewhart charts based on componentwise signs and Wilcoxon signed-rank sums. Zou and Tsung (2011) and Zou, Wang, and Tsung (2012) separately adapted the multivariate spatial sign and rank (cf., Oja 2010) to construct multivariate nonparametric EWMA control chart. Holland and Hawkins (2014) proposed a change-point detection chart based on spatial rank. Li et al. (2017) further integrated the multivariate spatial rank with forward variable section for detecting sparse mean shifts. More detailed literature reviews can be found in Qiu (2018).

For the purpose of illustration and comparison, we chose the baseline method as the multivariate sign exponentially weighted moving average (MSEWMA) control chart proposed by Zou and Tsung (2011). The MSEWMA chart is based on the multivariate sign test. The observed p-dimensional multivariate vector $\boldsymbol{x}_i$ is first transformed:

$$\boldsymbol{v}_i = \frac{\boldsymbol{A}_0(\boldsymbol{x}_i - \boldsymbol{\theta}_0)}{||\boldsymbol{A}_0(\boldsymbol{x}_i - \boldsymbol{\theta}_0)||}, \tag{4}$$

where $\boldsymbol{\theta}_0$ is the affine equivariant multivariate median proposed by Hettmansperger and Randles (2002) and $\boldsymbol{A}_0$ is the associated transformation matrix. The two parameters $\{\boldsymbol{\theta}_0, \boldsymbol{A}_0\}$ in $\boldsymbol{v}_i$ are estimated from the IC dataset, and $||\cdot||$ denotes the Euclidean norm. Next, it monitors the transformed vectors $\boldsymbol{v}_i$'s by the exponentially weighted moving average (EWMA) control chart with the monitoring statistics

$$\boldsymbol{\omega}_i = (1-\lambda)\boldsymbol{\omega}_{i-1} + \lambda \boldsymbol{v}_i \quad \text{and} \quad Q_i = \frac{2-\lambda}{\lambda} p \boldsymbol{\omega}_i' \boldsymbol{\omega}_i. \tag{5}$$

Finally, an OC alarm is raised whenever Q_i exceeds some pre-specified control limit. Clearly, the MSEWMA chart, or in fact many other nonparametric control charts,

only utilize the historical IC dataset, and raise an OC alarm whenever the observed data do not follow the IC patterns. By doing so, the benefit is to be able to detect any general OC patterns. However, the disadvantage is also obvious: much statistical efficiency will be lost when one is interested in detecting specific OC patterns that are similar to the historical ones.

In this article, we do not make any parametric assumptions on the c.d.f.s or p.d.f.s of the multivariate data $\boldsymbol{x}_i$ under the IC or OC states. Instead we assume that two historical datasets are available: one is an IC dataset, denoted by $\boldsymbol{X}_{IC} = \{\boldsymbol{x}_{ic1}, \boldsymbol{x}_{ic2}, ...\}$, and the other is an OC dataset denoted by $\boldsymbol{X}_{OC} = \{\boldsymbol{x}_{oc1}, \boldsymbol{x}_{oc2}, ...\}$. When monitoring the online observation $\boldsymbol{x}_i$, we are interested in detecting those OC patterns that are similar to the historical OC patterns. Our task is to utilize these historical IC and OC datasets to build an efficient control chart that can effectively monitor the online observation $\boldsymbol{x}_i$ and detect OC patterns that are similar to those in the historical OC dataset.

3. The Proposed KNN-ECUSUM Control Chart

Let us first provide a high-level description of our proposed k-nearest neighbors empirical cumulative sum (KNN-ECUSUM) control chart which is designed for monitoring the p-dimensional observations $\boldsymbol{x}_i$ based on the historical IC and OC datasets $\boldsymbol{X}_{IC}$ and $\boldsymbol{X}_{OC}$. Our proposed KNN-ECUSUM control chart consists of the following three steps:

Step 1: Use the KNN method to transform p-dimensional data $\boldsymbol{x}$ to one-dimensional categorical data $z(\boldsymbol{x})$, which is the proportion of the k nearest neighbors of $\boldsymbol{x}$ that are IC. Here $\boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{knn}$ are the respective subsets of historical IC and OC data for training the KNN classifier.

Step 2: Estimate the empirical IC and OC p.m.f.s of the one-dimensional categorical data $z(\boldsymbol{x})$ by using the subsets of historical IC and OC data, $\boldsymbol{X}_{IC}^{emp}$ and $\boldsymbol{X}_{OC}^{emp}$, respectively, where $\boldsymbol{X}_{IC}^{emp} = \boldsymbol{X}_{IC} - \boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{emp} = \boldsymbol{X}_{OC} - \boldsymbol{X}_{OC}^{knn}$, respectively.

Step 3: Apply the classical CUSUM procedure to monitor the one-dimensional data $z(\boldsymbol{x}_i)$, where $z(\boldsymbol{x}_i)$ is computed for each new pieces of online observation $\boldsymbol{x}_i$ as in Step 1, and the IC and OC p.m.f.s of $z(\boldsymbol{x}_i)$ are estimated as in Step 2.

Below we will explain each step of our proposed KNN-ECUSUM control chart in details and then provide the guideline for practical use. For the purpose of easy understanding, the remainder of this section is divided into four subsections. Section 3.1 presents the construction of the transformation $z(\boldsymbol{x})$ based on the KNN algorithm, and Section 3.2 discusses the estimation of the empirical p.m.f. of the transformed variable. In Section 3.3, the classical CUSUM procedure is applied to the one-dimensional data $z(\boldsymbol{x}_i)$ for detecting changes. Section 3.4 gives a summary and practical guidance for our proposed KNN-ECUSUM control chart.

3.1. *Construction of one-dimensional data $z(\boldsymbol{x})$*

In this subsection, we discuss the transformation of the p-dimensional data $\boldsymbol{x}$ to the one-dimensional data $z(\boldsymbol{x})$ by applying the KNN algorithm. In order to train the KNN classifier, we randomly select some training data, $\boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{knn}$, from the provided IC and OC historical datasets $\boldsymbol{X}_{IC}$ and $\boldsymbol{X}_{OC}$. As a classification method, the KNN algorithm is then applied to the datasets $\boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{knn}$ to predict the label of any p-dimensional data $\boldsymbol{x}$ based on its nearest k neighbors. Here we extend the binary

classification output of the conventional KNN algorithm to k-category probability outputs.

When using the KNN algorithm, a crucial task is how to define the distance between two p-dimensional vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. In our simulations and case study, we adopt the Mahalanobis distance (cf., Varmuza and Filzmoser 2010) defined by

$$\mathrm{dis}(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j}) = [(\boldsymbol{x}_\mathrm{i} - \boldsymbol{x}_\mathrm{j})^\mathrm{T} \boldsymbol{C}^{-1} (\boldsymbol{x}_\mathrm{i} - \boldsymbol{x}_\mathrm{j})]^{1/2}, \tag{6}$$

where $\boldsymbol{C}$ is the $p \times p$ sample covariance matrix. The main benefit of using the Mahalanobis distance is that it takes into account the covariance structure and standardizes different components of p-dimensional random vectors. Moreover, the Mahalanobis distance is equivalent to the Euclidean distance when different components are scaled to become standardized and also independent of each other. We should also emphasize that there are many other ways to define the distance $\mathrm{dis}(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j})$ for the KNN algorithms, e.g., by kernels or Pearson's correlation coefficients, etc.

For any given p-dimensional vector $\boldsymbol{x}$, let $V(\boldsymbol{x}, k)$ denote the k nearest points of $\boldsymbol{x}$ in the training datasets $\boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{knn}$ under the distance criterion $\mathrm{dis}(\cdot, \cdot)$. Note that the k points in the set $V(\boldsymbol{x}, k)$ may contain both IC and OC training samples. Then the transformed variable $z(\boldsymbol{x})$ is defined as the probability output of the KNN algorithm:

$$z(\boldsymbol{x}) = \frac{\#\text{ of points in the intersection } V(\boldsymbol{x}, k) \cap \boldsymbol{X}_{IC}^{knn}}{k}, \tag{7}$$

i.e., $z(\boldsymbol{x})$ is the proportion of IC samples in the k neighbors of $\boldsymbol{x}$. Since the value of the transformed variable $z(\boldsymbol{x})$ can only be i/k for $i = 0, 1, 2, \cdots, k$, it is a one-dimensional categorical variable with $k + 1$ possible values. Therefore, the problem of monitoring the multivariate observation $\boldsymbol{x}$ now becomes a problem of monitoring the one-dimensional categorical variable $z(\boldsymbol{x})$.

We acknowledge that intuitively it will lose information when using the KNN algorithm to reduce p-dimensional data into one-dimensional data. However, we want to emphasize that in the context of monitoring or SPC, most, if not all, control charts raise alarms if some real-valued monitoring statistic is large, and such monitoring statistic is one-dimensional based on the probability distributions or the likelihood ratio test statistics of the multivariate data. In other words, we need to summarize the p-dimensional data over time as a single one-dimensional random variable for the evidence of changes. The traditional SPC approaches often first approximate the distributions of p-dimensional variables and then compute the one-dimensional monitoring statistics, see, empirical likelihood (Owen 2001) and kernel density estimation (Terrell and Scott 1992). Here we propose to approximate the one-dimensional monitoring statistics directly by the likelihood ratio statistics of the transformed one-dimensional random variable $z(\boldsymbol{x})$. Our simulation studies indicate that the loss information seems to be marginal in the context of online monitoring.

3.2. *Estimation of the empirical probability mass function*

After using the KNN algorithm to convert the p-dimensional variable $\boldsymbol{x}$ to a one-dimensional categorical variable $z(\boldsymbol{x})$, we now face the problem of monitoring one-dimensional categorical data. Existing methods for monitoring categorical data are available from Marcucci (1985), Woodall (1997) and Montgomery (2009). However,

these methods require counting the number of data points in each category, and are not suitable for our problem where the categorical variable $z(\boldsymbol{x})$ is calculated from a single observation $\boldsymbol{x}$. Here our proposed method applies the classical CUSUM procedure to $z(\boldsymbol{x})$, and for that purpose, we need to estimate the p.m.f. of $z(\boldsymbol{x})$ under the IC and OC states.

First, we briefly review how the empirical p.m.f. of a categorical variable Y is estimated. Assume that Y takes $k+1$ possible values, say, $t_0 \leq t_1 \leq \ldots \leq t_k$, and we observe G i.i.d. random samples $y_1, y_2, \ldots, y_G$ of Y. The empirical p.m.f. of Y is

$$\hat{f}(t_j) = \frac{1}{G}\sum_{i=1}^{n} I(y_i = t_j) = \frac{\#\text{ of } y_i \text{ in j-th category}}{G}, \quad j = 0, 1, \ldots, k \tag{8}$$

where $I(\cdot)$ denotes the indicator function. Next, let us illustrate how this empirical p.m.f. can be adapted to our problem by applying the boostrap or random sampling method to the IC and OC datasets. Taking the IC dataset as an example. Each time a subset of m IC observations are randomly (and possibly repeatedly) sampled from $\boldsymbol{X}_{IC}^{emp}$. After this subset of m pieces of IC data are input into the KNN classifier, we collect m transformed outputs $z(\boldsymbol{x})$s. This re-sampling and transformation procedure is repeated B times, and we obtain a total of Bm categorical observations $z(\boldsymbol{x})$s. As in Equation (8), the empirical p.m.f. of $z(\boldsymbol{x})$ under the IC scenario is simply the proportion of Bm observations that fall under each category, and can be calculated as

$$\hat{f}_{ic}(z) = \begin{cases} \frac{\#\text{ of } z(\boldsymbol{x})'s\ =z}{Bm} & \text{if } \ z = \frac{j}{k}, j = 0, 1, \ldots, k \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Note that this empirical IC p.m.f. satisfies $\sum_{i=0}^{k} \hat{f}_{ic}(\frac{i}{k}) = 1$, which is the fundamental requirement for a probability mass function. Similarly, we can derive the empirical OC p.m.f., $\hat{f}_{oc}(z)$, of the categorical variable $z(\boldsymbol{x})$ under the OC scenario by applying the above procedure to the OC dataset $\boldsymbol{X}_{OC}^{emp}$. Some issues on IC or OC samples for calculating the empirical densities will be discussed in more detail later in Subsection 3.4.

3.3. *Monitoring the transformed variables $z(\boldsymbol{x}_i)$*

After data transformation and p.m.f. estimation, our proposed control chart is to apply the classical CUSUM procedure to the transformed variables with the estimated IC and OC p.m.f.s. Specifically, when monitoring the p-dimensional data $\boldsymbol{x}_i$ in real time, we first use the KNN algorithm in Step 1 to obtain the transformed variable $z(\boldsymbol{x}_i)$. In Step 2, the corresponding p.m.f.s of the transformed variable $z(\boldsymbol{x}_i)$ can be estimated as $\hat{f}_{ic}(z)$ or $\hat{f}_{oc}(z)$. Thus the problem of monitoring the p-dimensional data $\boldsymbol{x}_i$ can be simplified into the problem of monitoring the transformed one-dimensional variable $z(\boldsymbol{x}_i)$ with a possible p.m.f. change from $\hat{f}_{ic}(z)$ to $\hat{f}_{oc}(z)$.

At the high-level, our proposed KNN-ECUSUM control chart applies the classical CUSUM control chart to the transformed one-dimensional variable $z(\boldsymbol{x}_i)$. To be more specific, our method defines the CUSUM statistic recursively as

$$W_n = \max(W_{n-1} + \frac{\hat{f}_{oc}(z(\boldsymbol{x}_i))}{\hat{f}_{ic}(z(\boldsymbol{x}_i))}, 0) \tag{10}$$

for $n \geq 1$ with the initial value $W_0 = 0$, and then raises an alarm whenever W_n exceeds a pre-specified control limit $L > 0$. Recall that the CUSUM procedure is exactly optimal when the p.d.f.s/p.m.f.s under the IC and OC scenarios are completely specified. In our context, we utilize the historical IC and OC datasets to estimate the p.m.f. of the transformed variable. When the OC patterns are similar to those in the historical data, the empirical p.m.f. will be similar to the true p.m.f., and thus the performance of our proposed KNN-ECUSUM control chart would be similar to that of the optimal CUSUM procedure with the true density functions, which is also verified in the simulation results in Section 4.

In summary, our proposed KNN-ECUSUM control chart can be summarized in Table 1. The novelty of our proposed control chart lies in combining four well established methods together: the KNN algorithm, the empirical probability mass function, bootstrapping, and CUSUM, and the fundamental idea is to use the KNN algorithm as a dimension reduction tool. Rather than directly estimating the distributions of the raw p-dimensional variables, we monitor the transformed one-dimensional categorical variables, whose distributions can be easily estimated from their empirical densities.

Finally, it is also useful to discuss the computational complexity of our proposed KNN-ECUSUM control chart, which includes three steps. The first two steps of training the KNN classifier and estimating the p.m.f.s only need to be done once in the off-line Phase I stage. The third step is the online Phase II stage and involves the computational complexity of $O((p^2 + k)n_{knn})$ at each time step. To see this, for each training data in KNN, we need to compute the Mahalanobis distance from the new observation to training set observations. Each Mahalanobis computation involves $O(p^2)$ runtime, and so it requires $O(p^2 n_{knn})$ runtime to compute all distances. Next, the KNN algorithms selects k smallest distance, and involves $O(n_{knn}k))$ runtime to loop though all n_{knn} distances. Moreover, it involve $O(k)$ time to update the CUSUM statistics W_t. Thus at each time step during the Phase II stage, the computational complexity of our proposed KNN-ECUSUM control chart is $O(p^2 n_{knn} + kn_{knn} + k)$, which is the same order as $O((p^2 + k)n_{knn})$.

3.4. *Design issues of the KNN-ECUSUM chart*

Our proposed KNN-ECUSUM control chart involves several important tuning parameters, which must be chosen carefully when applied in practice. Below we will discuss how to choose these tuning parameters appropriately.

3.4.1. Selecting k in the KNN classifier

When building the KNN model, the number of neighbors, k, is an important tuning parameter. Generally speaking, a small k may overfit the KNN training dataset and decrease the prediction accuracy. For the categorical variable $z(\boldsymbol{x})$, a small k will also lead to imprecise output. On the other hand, a large k increases the computational load and may not necessarily bring a significant increase in the classification rate for prediction. Often people use cross-validation to achieve an appropriate k. Following the empirical rule of thumb, k is often chosen to be of order $\sqrt{n_{knn}}$, where n_{knn} is the number of observations in the training IC and OC datasets reserved for training the KNN classifier. This is used as the starting point in some KNN software packages. Our empirical results from simulation and real-data studies suggest that $k = \alpha * \sqrt{n_{knn}}$ with $\alpha \in [0.5, 3]$ is a reasonable choice.

Table 1. Procedures for implementing the KNN-ECUSUM control chart.

Algorithm 1: Our proposed KNN-ECUSUM control chart	
Step 1	**Goal**: To Build the KNN classifier for data transformation.
	Input: Training Samples $\boldsymbol{X}_{IC}^{knn}$ and $\boldsymbol{X}_{OC}^{knn}$; a given testing sample $\boldsymbol{x}$.
	Output: The categorical variable $z(\boldsymbol{x})$.
Step 2	**Goal**: To calculate the empirical p.m.f. of the transformed categorical variable.
	Input: Samples $\boldsymbol{X}_{IC}^{emp}$ and $\boldsymbol{X}_{OC}^{emp}$.
	Model: The KNN model $z(\boldsymbol{x})$ built in Step 1.
	Iterate: For the IC scenario, $t = 1, 2, \ldots B$ times.
	1. Randomly select m IC samples from $\boldsymbol{X}_{IC}^{emp}$.
	2. The selected IC samples serve as input to $z(\boldsymbol{x})$; collect the model outputs.
	Output: For the IC scenario, the estimated IC p.m.f. of $z(\boldsymbol{x})$, $\hat{f}_{ic}(z)$.
	Repeat: Using the same model, repeat the same iterative procedure for the
	OC scenario and get the estimated OC p.m.f. of $z(\boldsymbol{x})$, $\hat{f}_{oc}(z)$.
Step 3	**Goal**: To monitor online multivariate observations.
	Input: Online samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\}$; control limit L.
	Charting Statistic: $W_n = \max(W_{n-1} + \frac{\hat{f}_{oc}(z(\boldsymbol{x}_i))}{\hat{f}_{ic}(z(\boldsymbol{x}_i))}, 0)$, $W_0 = 0$.
	Output: If $W_n > L$, the control chart raises an OC alarm;
	otherwise, the process is considered to be operating normally.

3.4.2. Selecting the control limit L

When constructing the control chart, it is crucial to find an accurate control limit L for the monitoring or charting statistics to satisfy the IC ARL constraint defined in (2). In our proposed KNN-ECUSUM chart, we propose a re-sampling method to find its value. For a given control limit L, we randomly select an observation from the IC historical dataset $\boldsymbol{X}_{IC}$, and the CUSUM statistic W_1 is calculated as in (10). If the CUSUM statistic $W_1 < L$, another observation will be randomly selected from $\boldsymbol{X}_{IC}$ and the CUSUM statistic W_2 will be calculated. This procedure is repeated until the CUSUM statistic $W_T \geq L$. This completes one Monte Carlo simulation, and the number T of IC observations is recorded and is often called one realization of the run length. Then, we repeat the above steps n_0 times and obtain n_0 realizations of the run length, denoted by $T_1, \cdots, T_{n_0}$. The ARL of our proposed KNN-ECUSUM control chart with the control limit L is then estimated as $\hat{ARL}(L) = (T_1 + \cdots + T_{n_0})/n_0$. The control limit L is then adjusted through bisection search so that the obtained $\hat{ARL}(L)$ is close to the preset ARL constraint γ. In this paper, n_0 is chosen as 10,000.

3.4.3. Sample size considerations

The sample size is another important factor that can affect the performance of our proposed KNN-ECUSUM control chart. There are two kinds of sample sizes in our proposed KNN-ECUSUM chart: one for training the KNN algorithm, and the other for estimating the p.m.f.s of the one-dimensional categorical variable. Both need enough data for good performance, and ideally data are disjoint for these two steps.

In Step 1 of the KNN-ECUSUM chart, it is crucial to have enough samples for training the KNN algorithm, since the quality of the transformed categorical variable $z(\boldsymbol{x})$ will affect the monitoring performance of the KNN-ECUSUM chart. The number of samples for KNN algorithm is very dependent on the specific problem, e.g., the dimension p of the data and the smoothness of underlying true model, see Chapter 2.5 of the classical statistical learning book by Hastie, Tibshirani and Friedman (2009). Meanwhile, due to the need to reserve sufficient historical data for calculating empirical densities, it is inappropriate for the KNN classifier to use too many training data. Under the setting of our simulation studies in Section 4, we empirically found that some good choices of the size of training data for the KNN algorithm are $n_{knn} \in [0.25, 0.75]N$, where N is the total size of the historical dataset. For the case study in Section 5, we choose $n_{knn} \approx 0.4N$. Of course, in other real-world applications, the choice of n_{knn} will likely depend on the applications and the availability of training data.

In Step 2, more samples will lead to more precise estimates of the p.m.f.s of the transformed categorical variable. We suggest making use of possibly all of the historical data especially when the historical IC or OC dataset is not large enough. For large historical datasets containing millions of observations, it is acceptable to use a subset to estimate the p.m.f.s. It is important to note that the training data for building the KNN classifier in Step 1 should not overlap with those for estimating the empirical p.m.f.s.

In many fields such as machine learning, if one has a balanced dataset, then the classification problem is often easy to solve, including our proposed KNN-ECUSUM chart. However, we often face the unbalanced data in the field of SPC. For Phase I analysis in SPC, we often have many IC samples, but generally have few OC samples. In such a situation, to address the unbalanced issue, one possible solution is to apply

oversampling (or the so-called bootstrapping method) that replicates samples from the OC dataset in order to increase its cardinality.

3.4.4. When multiple OC patterns exist

So far, we have only considered the binary classification when constructing the KNN classifier in Step 1, as it is assumed that the historical OC dataset exhibits only one single OC pattern. However, sometimes multiple OC patterns may exist, and the historical OC dataset may contain more than one OC cluster. In such a situation, more efforts should be made to investigate different OC clusters.

It turns out that the proposed KNN-ECUSUM chart can be easily extended to handle the case of multiple OC clusters. To this end, we need to divide the historical OC dataset into several clusters. This clustering procedure might be available during Phase I analysis, or can be done by standard unsupervised learning methods such as the k-means clustering method, principal component analysis (PCA) and so on. Based on our experience, while the proposed method can be extended to any number C of OC clusters, a smaller number of clusters will generally lead to better performance in monitoring changes. Thus a small number $C \leq 4$ of OC clusters is suggested.

Now we briefly describe the extension of our proposed control chart. Denote the historical IC data and the C OC clusters by $\boldsymbol{X}_{ic}, \boldsymbol{X}_{oc}^1 \ldots, \boldsymbol{X}_{oc}^C$. As in Step 1 in Table 1, we also build a KNN classifier but now it is a multi-class KNN classifier based on the subsets of these $C+1$ training datasets. Next, we calculate the empirical p.m.f.s, $\hat{f}_{ic}(\cdot), \hat{f}_{oc}^1(\cdot), \ldots, \hat{f}_{oc}^C(\cdot)$, using boostrap in Step 2. Notice that there are $C+1$ probability densities to be calculated, rather than two densities in the case of one single OC cluster. In Step 3, we construct the CUSUM statistic for each OC cluster and obtain C CUSUM statistics with $W_{1,n} = \max(W_{1,n-1} + \frac{\hat{f}_{oc}^1(z(\boldsymbol{x}_i))}{\hat{f}_{ic}(z(\boldsymbol{x}_i))}, 0)), \ldots, W_{C,n} = \max(W_{C,n-1} + \frac{\hat{f}_{oc}^C(z(\boldsymbol{x}_i))}{\hat{f}_{ic}(z(\boldsymbol{x}_i))}, 0))$. Then we define the monitoring statistics based on the maximum of all the C CUSUM statistics, $W_n = \max(W_{1,n}, W_{2,n}, \ldots, W_{C,n})$, and raise an OC alarm when $W_n > L$.

4. Simulation Studies

In this section, we conduct extensive simulation studies to demonstrate the efficiency and robustness of the proposed KNN-ECUSUM chart. For the purpose of comparison, three alternative control charts are also considered: (i) the nonparametric MSEWMA chart in (5) proposed by Zou and Tsung (2011), (ii) the traditional parametric MEWMA method (Hotelling's T^2 type), and (iii) the classical parametric CUSUM chart which is optimal when the underlying IC and OC distributions are known (cf., Qiu 2014).

To better present our simulation results, we split this section into three subsections. In Section 4.1, the KNN-ECUSUM chart is compared with competitors when there is only one OC cluster in the historical OC dataset. It is extended to the case of multiple OC clusters in Section 4.2. Section 4.3 contains sample size analysis. Throughout the simulation, the IC ARL is fixed at 600. The MATLAB code for implementing the proposed chart is available from the authors upon request.

4.1. *When only one cluster exists in historical OC data*

In this subsection we will investigate the case when only one cluster exists in the historical OC data. Here we focus on detecting a change in the mean vector, and consider three different generative models of the p-dimensional multivariate data $\boldsymbol{x}_i$ in order to evaluate the robustness of the control charts:

- the multivariate normal distribution $N(\mathbf{0}_{p\times 1}, \Sigma_{p\times p})$.;
- the multivariate t distribution $t_\zeta(\mathbf{0}_{p\times 1}, \Sigma_{p\times p})$ with ζ degrees of freedom;
- the multivariate mixed distribution, $rN(\boldsymbol{\mu}_1, \Sigma_{p\times p}) + (1-r)N(\boldsymbol{\mu}_2, \Sigma_{p\times p})$, i.e., a mixture of two multivariate normal distributions. In our simulation, we set $r = 0.5$, and the $p \times 1$ mean vector $\boldsymbol{\mu}_1$ ($\boldsymbol{\mu}_2$) is $3(-3)$ in the first component but is 0 for all other components.

For all three generative models, the covariance matrix is chosen as $\Sigma_{p\times p} = (\sigma_{ij})_{p\times p} = 0.5^{|i-j|}$, and we consider the OC generative model with a mean shift δ in the first component of the observations unless stated otherwise, where the shift magnitude δ ranges from 0.2 to 2 with the step size of 0.2. In addition, we consider two choices of the dimension p: $p = 6$ and $p = 20$. The tuning parameter λ in the MEWMA and the MSEWMA charts is set to 0.2 for simplicity. We emphasize that our proposed control chart does not use any information pertaining to the generative IC or OC model, and will only use the historical data generated from the IC or OC model: the KNN algorithm in Step 1 is based on 1000 pieces of IC and OC training data with the number of nearest neighbors $k = 30$, and the p.m.f. estimation in Step 2 is based on $100,000$ pieces of IC and OC data with $B = 1000$ loops for bootstrap.

To gain deeper insight of our proposed control chart, Figure 1 plots the estimated p.m.f.s of the transformed one-dimensional categorical data $z(\boldsymbol{x})$ under the three generative models with the OC shift size δ equals 1. From the plots, it is clear that the estimated IC p.m.f.s ($\hat{f}_{ic}$) are concentrated on the right of the figure as the KNN output increases (except in some extreme situations), while the estimated OC p.m.f.s ($\hat{f}_{oc}$) are concentrated on the left of the figure. This is consistent with our intuition that the neighboring samples of an IC sample are likely from the IC distribution, and vice versa. The significant differences in the estimated IC and OC p.m.f.s in Figures 1 demonstrate that the transformed one-dimensional categorical variable $z(\boldsymbol{x})$ can be used to detect changes in the original multivariate data $\boldsymbol{x}$, and thus our proposed KNN-ECUSUM control chart should be effective.

Tables 2-4 summarize the simulated ARL performance of the four control charts under the three generative models. Table 2 presents the efficiency of our proposed KNN-ECUSUM chart in the case of detecting mean shifts for the multivariate normal distribution. First, subject to the same IC ARL value, the OC ARL values of our proposed KNN-ECUSUM chart are only slightly larger than those of the CUSUM chart, which is the optimal one under the normality assumption. Second, the KNN-ECUSUM chart has much smaller OC ARL values as compared to the MEWMA and MSEWMA charts for all mean shift magnitudes. The major reason is that the KNN-ECUSUM chart uses the precious OC information in the historical dataset, which is overlooked in the MEWMA and MSEWMA charts. Thus, it is not surprising that our method can detect mean shift more efficiently than these two alternatives. Third, it is interesting to see the effect of data dimension: as the data dimension increases from 6 to 20, both the MEWMA and MSEWMA charts have a much larger OC ARL when the mean shift δ is small, e.g., $\delta \in [0.4, 0.8]$, whereas the performance of our proposed KNN-ECUSUM chart does not change much. This demonstrates the effectiveness of

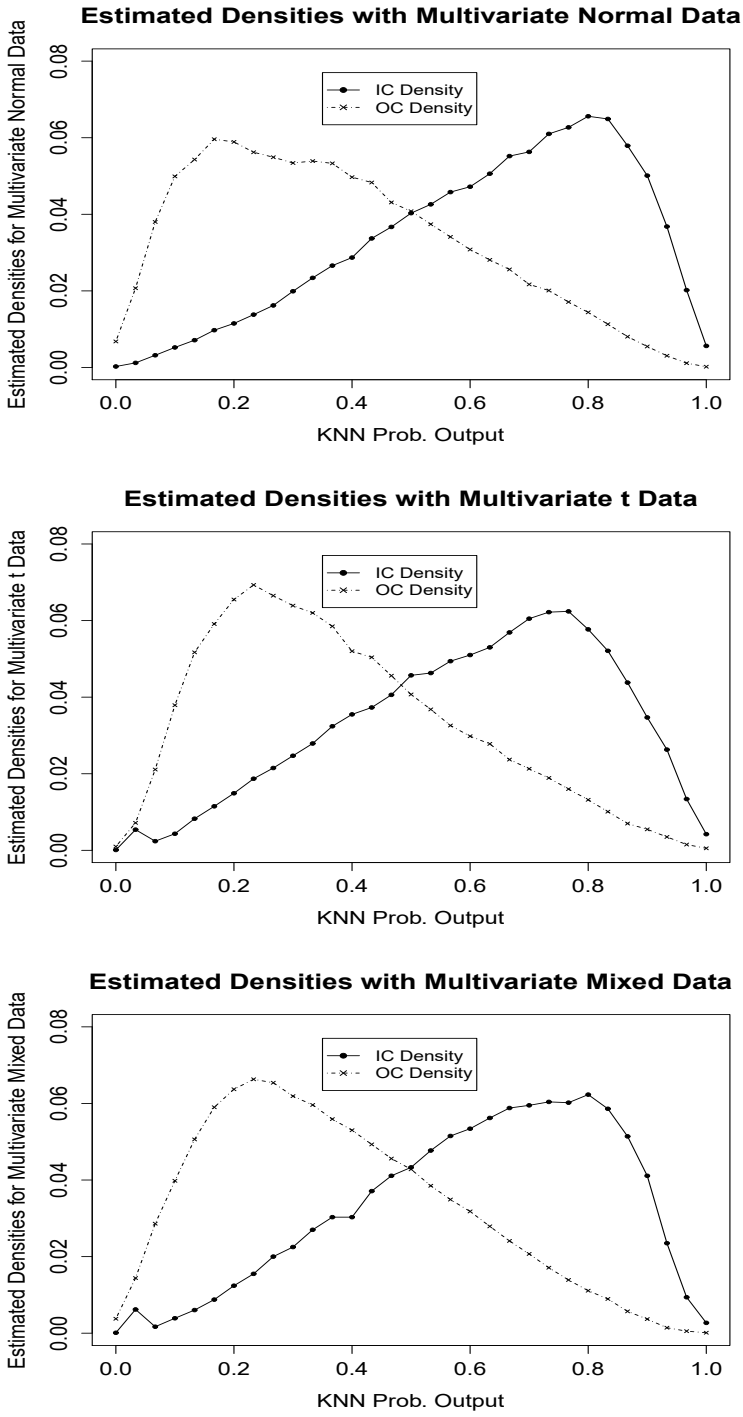

Figure 1. Estimated empirical densities of KNN output for multivariate normal, t and mixed data in IC and OC scenarios.

Table 2. (OC) ARL performance comparison for multivariate normal data

Dimension p	Shift Size δ	KNN-ECUSUM	MEWMA	MSEWMA	CUSUM
6	0	602(595)	600(601)	602(594)	599(597)
	0.2	128(126)	318(314)	322(323)	122(120)
	0.4	38.2(34.3)	101(95.3)	104(97.3)	35.5(31.7)
	0.6	17.3(13.5)	36.7(30.8)	39.5(33.0)	15.1(11.6)
	0.8	9.67(6.47)	18.0(12.5)	20.2(13.4)	8.67(5.72)
	1	6.78(3.76)	11.0(6.48)	13.3(7.24)	5.87(3.30)
	1.2	5.08(2.44)	7.84(3.83)	9.80(4.36)	4.43(2.15)
	1.4	4.16(1.74)	6.08(2.63)	7.97(2.87)	3.58(1.60)
	1.6	3.55(1.32)	4.97(2.00)	6.93(2.18)	3.00(1.21)
	1.8	3.19(1.09)	4.25(1.56)	6.24(1.78)	2.62(0.98)
	2	2.86(0.89)	3.69(2.66)	5.76(1.45)	2.31(0.80)
20	0	600(586)	599(606)	599(589)	601(589)
	0.2	130(127)	410(408)	396(400)	118(114)
	0.4	40.1(36.0)	171(167)	163(158)	33.1(29.6)
	0.6	17.2(13.6)	62.9(57.5)	62.1(54.1)	14.2(11.0)
	0.8	9.97(6.37)	28.6(22.2)	29.1(21,7)	7.91(5.18)
	1	6.91(3.91)	15.9(10.3)	16.8(10.2)	5.41(2.98)
	1.2	5.19(2.50)	10.6(5.58)	11.7(5.71)	4.10(1.96)
	1.4	4.32(1.86)	7.93(3.54)	8.94(3.68)	3.31(1.44)
	1.6	3.65(1.74)	6.26(2.47)	7.41(2.59)	2.77(1.08)
	1.8	3.21(1.16)	5.25(1.94)	6.40(1.97)	2.41(0.88)
	2	2.86(0.93)	4.57(1.57)	5.71(1.58)	2.17(0.76)

NOTE: Standard deviations are in parentheses

Table 3. OC performance comparison for multivariate t data

Dimension p	Shift Size δ	KNN-ECUSUM	MEWMA	MSEWMA	CUSUM
6	0	599(603)	600(606)	600(587)	600(603)
	0.2	146(146)	578(568)	342(339)	135(132)
	0.4	45.2(41.4)	479(480)	115(108)	41.3(36.7)
	0.6	19.1(14.7)	348(350)	44.7(37.4)	17.7(13.7)
	0.8	11.0(7.33)	228(228)	23.1(16.6)	10.0(6.63)
	1	7.54(4.10)	133(128)	15.1(8.74)	6.89(3.75)
	1.2	5.89(2.86)	73.3(65.4)	11.1(5.32)	5.31(2.54)
	1.4	4.86(2.04)	40.0(32.3)	9.05(3.86)	4.43(1.90)
	1.6	4.25(1.64)	23.7(16.0)	7.77(2.88)	3.85(1.48)
	1.8	3.81(1.27)	15.5(8.80)	6.94(2.36)	3.42(1.22)
	2	3.53(1.10)	11.6(5.55)	6.38(1.95)	3.19(1.04)
20	0	601(589)	599(603)	600(589)	599(597)
	0.2	137(135)	595(602)	420(414)	129(126)
	0.4	43.1(39.6)	564(578)	180(171)	35.5(32.7)
	0.8	10.3(14.4)	492(479)	33.1(25.8)	8.48(5.60)
	1	7.16(4.04)	430(348)	19.1(12.5)	5.91(3.29)
	1.2	5.51(2.67)	346(348)	13.1(7.19)	4.54(2.25)
	1.4	4.48(1.94)	282(281)	10.0(4.63)	3.70(1.69)
	1.6	3.90(1.49)	211(205)	8.25(3.35)	3.21(1.38)
	1.8	3.45(1.25)	149(144)	7.10(2.59)	2.89(1.17)
	2	3.15(1.10)	99.9(91.0)	6.34(2.09)	2.63(1.01)

NOTE: Standard deviations are in parentheses

Table 4. OC performance comparison for multivariate mixed data

Dimension p	Shift Size δ	KNN-ECUSUM	MEWMA	MSEWMA	CUSUM
6	0	599(600)	600(597)	601(583)	600(594)
	0.2	131(128)	598(598)	584(583)	124(120)
	0.4	39.4(36.3)	609(615)	565(562)	35.3(32.2)
	0.6	17.2(13.5)	591(573)	528(530)	15.0(11.6)
	0.8	9.75(6.39)	600(593)	516(514)	8.52(5.50)
	1	6.66(3.76)	597(605)	471(467)	5.85(3.28)
	1.2	5.19(2.44)	595(591)	455(446)	4.42(2.14)
	1.4	4.14(1.73)	596(604)	427(423)	3.58(1.57)
	1.6	3.58(1.38)	599(591)	414(407)	3.04(1.20)
	1.8	3.17(1.11)	603(593)	382(381)	2.65(0.95)
	2	2.87(0.92)	605(605)	377(373)	2.37(0.76)
20	0	602(594)	599(602)	600(581)	600(590)
	0.2	130(128)	609(605)	604(615)	122(122)
	0.4	41.1(37.1)	591(596)	595(589)	33.5(29.5)
	0.6	17.4(13.3)	612(617)	585(580)	14.0(11.2)
	0.8	10.0(6.68)	588(582)	584(583)	7.97(5.09)
	1	6.94(3.88)	593(586)	568(564)	5.41(3.00)
	1.2	5.31(2.57)	608(607)	556(546)	4.06(1.92)
	1.4	4.29(1.84)	593(596)	549(559)	3.32(1.41)
	1.6	3.67(1.41)	608(600)	539(546)	2.80(1.10)
	1.8	3.20(1.13)	594(601)	528(529)	2.46(0.86)
	2	2.89(0.96)	598(605)	524(525)	2.23(0.70)

NOTE: Standard deviations are in parentheses

our proposed KNN-ECUSUM chart under the multivariate normal distribution.

Table 3-4 report the robustness of the KNN-ECUSUM chart under different distribution assumptions: Table 3 is for the multivariate t distribution, whereas Table 4 is for multivariate mixed distribution. The superiority of our method is still maintained in the sense that the KNN-ECUSUM chart still works much more effectively than the MEWMA and MSEWMA charts. Note that both the MEWMA and MSEWMA charts fail to detect the shift under the mixed distribution, partly because the mixed data is not elliptically distributed, which is a fundamental requirement for the MEWMA and MSEWMA charts. It is also interesting to compare the results in Tables 2 and 3: when the true underlying distribution changes from multivariate normal to multivariate t, the performance of the baseline MEWMA chart deteriorates significantly. Meanwhile, the performance of our proposed KNN-ECUSUM charts does not change much, and so does the MSEWMA. All these results illustrate that our proposed KNN-ECUSUM control chart, as a nonparametric method, is indeed robust to the model distribution assumption.

4.2. *When multiple clusters exist in historical OC data*

In this subsection, we conduct additional simulation studies to illustrate the effectiveness of the proposed KNN-ECUSUM chart in cases when multiple OC clusters exist in the historical OC dataset. Here we only report the results of multivariate normal and multivariate t distributions for simplicity. The dimension ranges from $p = 6$ $p = 20$ to $p = 40$, and the number of clusters is chosen as $C = 2$ or 3. The mean and covariance matrix of the IC data are still $\mathbf{0}_{p\times 1}$ and $\Sigma_{p\times p}$), and for each OC cluster, the shift magnitude is 1 but the shift occurs in various dimensions in different clusters. For example, when the number of OC clusters $C = 2$ and data dimension $p = 2$, the mean vectors of the two OC clusters are $[1, 0]$ and $[0, 1]$, respectively. The MEWMA and MSEWMA charts are still considered for comparison. The other settings are similar to those under one OC cluster in the previous section.

The simulation results are reported in Table 5. From this table, we can see that the performance of our proposed KNN-ECUSUM chart is still desirable regardless of the number C of OC clusters. In particular, while all charts have similar OC performance under the multivariate normal distribution, our proposed KNN-ECUSUM chart generally has smaller OC ARL than the baseline MEWA and MSEWMA charts under the multivariate t distribution. Moreover, while the performances of both the MEWMA and MSEWMA charts are still affected by data dimension, our method is more robust to the dimension. Combining the results in Sections 4.1-4.2, we conclude that our proposed KNN-ECUSUM chart is a powerful tool for monitoring multivariate data, and is suitable for practical use when it is non-trivial to model the data distributions.

4.3. *Sample size analysis*

When implementing the KNN-ECUSUM chart, a practical issue is the sample size. As mentioned before, the sample size can affect the choice of the number k of neighbors and the accuracy of the estimated density functions. Although the number of required samples depends on multiple factors such as data dimension and the true data distribution, it is still necessary to investigate how large a sample size is generally appropriate and how the estimated densities change as the sample size increases. In this subsection, We perform simulation studies to demonstrate the effect of the

Table 5. OC performance comparison for the case of multiple clusters

Dimenson	# of Clusters	Historical OC Mean	Multivariate Normal			Multivariate t		
p	C		KNN-ECUSUM	MEWMA	MSEWMA	KNN-ECUSUM	MEWMA	MSEWMA
		[1,0,0,0,0,0]	12.7(7.68)	11.9(6.94)	13.2(7.09)	13.6(7.93)	136(130)	15.0(8.90)
6	2	[0,1,0,0,0,0]	9.89(5.57)	12.0(6.97)	13.4(7.19)	12.2(7.25)	132(124)	15.3(9.10)
		[1,0,0,0,0,0]	12.6(7.31)	11.9(6.94)	13.2(7.09)	13.7(7.65)	136(130)	15.0(8.90)
6	2	[0,1.5,0,0,0,0]	4.43(2.23)	5.74(2.37)	7.47(2.55)	5.61(2.93)	30.5(22.8)	8.45(3.38)
		[1,0,0,0,0,0]	19.9(13.4)	11.9(6.94)	13.2(7.09)	17.6(10.9)	136(130)	15.0(8.90)
6	3	[0,1,0,0,0,0]	14.5(8.76)	12.0(6.97)	13.4(7.19)	17.7(10.6)	132(124)	15.3(9.10)
		[0,0,1,0,0,0]	16.4(10.5)	11.9(6.99)	13.5(7.32)	17.9(10.7)	133(127)	15.4(9.08)
		1 in the 1st com	12.8(7.84)	15.9(10.3)	16.8(10.2)	10.7(6.30)	420(424)	19.1(12.3)
20	2	1 in the 2nd com	11.6(6.83)	16.0(10.3)	17.4(10.7)	12.7(7.70)	419(413)	19.7(13.1)
		1 in the 1st com	11.3(6.66)	15.9(10.3)	16.8(10.2)	11.3(6.67)	420(424)	19.1(12.3)
20	2	1.5 in the 2nd com	4.88(2.66)	6.96(3.00)	8.17(3.11)	4.88(2.66)	248(243)	9.20(4.03)
		1.5 in the 1st com	10.1(6.11)	8.79(4.07)	9.82(4.18)	10.1(6.11)	405(408)	11.1(5.42)
40	3	1.5 in the 2nd com	8.57(4.84)	8.73(4.00)	10.2(4.42)	8.57(4.84)	408(409)	11.1(5.47)
		1.5 in the 3rd com	7.68(4.19)	8.77(4.02)	9.97(4.26)	7.68(4.19)	407(411)	11.2(5.44)

NOTE: Standard deviations are in parentheses

sample size.

As the KNN classifier delivers a probability estimator, let us first consider how to achieve the "theoretical true" probability. According to the conditional probability formulation, the true probability that sample $\boldsymbol{x}$ is of IC status can be written as

$$\mathrm{P}(\boldsymbol{x}) = \mathrm{P}(\boldsymbol{x} \in \mathrm{IC} | \boldsymbol{x} \in \mathrm{IC \text{ or } OC}) = \frac{g_{ic}(\boldsymbol{x})}{g_{ic}(\boldsymbol{x}) + g_{oc}(\boldsymbol{x})} \ . \tag{11}$$

Then the corresponding density function can be derived as

$$g(t) = \frac{d\mathrm{P}(\mathrm{P}(\boldsymbol{x}) \leq t)}{dt} \tag{12}$$

and we call $g(\cdot)$ the true probability density.

In the simulation, both the IC and OC samples are assumed to be normally distributed with $p = 6$, $\boldsymbol{\mu}_0 = \mathbf{0}$, $\boldsymbol{\mu}_1 = [1, 1, 1, 0, 0, 0]$, and the covariance matrix equals the identity matrix. Following Equations (11) and (12), the true density can be obtained after simple calculations (provided in the appendix) as

$$g(t) = \frac{1}{\sqrt{\pi}} \exp(-\frac{1}{6}(\frac{3}{2} - \log \frac{t}{1-t})^2)(\frac{1}{1-t} + \frac{1}{t}). \tag{13}$$

As we are interested in how the estimated densities approximate the true densities, various sample sizes are considered and the results are shown in Figures 2 and 3. In the plots, "num-train" and "num-test" denote the sample sizes for training the KNN classifier and for estimating the densities, respectively. In Figure 2, we fix "num-train" and discuss the effect of "num-test" and the choice of k, while in Figure 3, the effect of these three factors are all discussed. From Figure 2, we can observe that fewer test samples may lead to unstable density estimation, and the density curve becomes smoother as the number of test samples increases. From Figure 3, we can observe that the estimated density approximates the true one better with more training samples.

Combining the results from all the plots, we suggest that at least 1,000 training samples and 10,000 test samples are required for practical use.

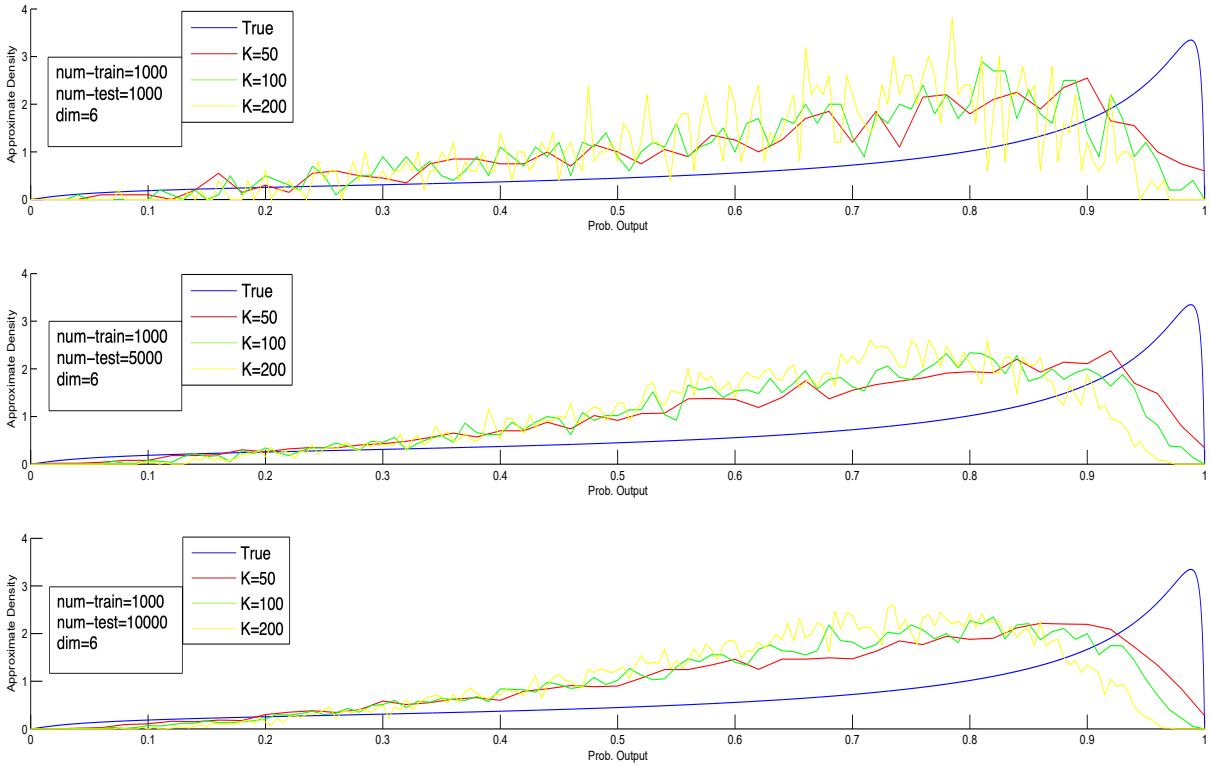

Figure 2. Empirical density approximates the true density: Effect of test sample size and the choice of k.

Another interesting finding from Figures 2 and 3 is the choice of k. Recall that we suggested $k = \alpha * \sqrt{n_{knn}}$ before. This general suggestion is validated by the results in these two figures. We can observe that when the training sample size equals 1000, the density curve with $k = 50$ is the closest one to the true curve. Also, as the sample size increases to 4000, the estimated curves with $k = 50$ and $k = 100$ are quite similar. Furthermore, when the training sample size is not large enough, it is inappropriate to choose a large k, such as $k = 500$. Thus, the selection of k relies heavily on the training sample size.

5. A Real Data Application

In this section, we use a real-world example from the HDDMS to demonstrate the effectiveness of our proposed KNN-ECUSUM chart. The HDDMS records various attributes and provides early warnings of disks failure. The provided datasets (the IC and OC datasets) consist of the 14 attributes observed from 23010 IC disks and 270 OC disks that were constantly monitored once every hour for one week. Since the values of some attributes are constant or barely change in some disks, we select their average values to denote the working status (we also tried the full dataset in our exper-

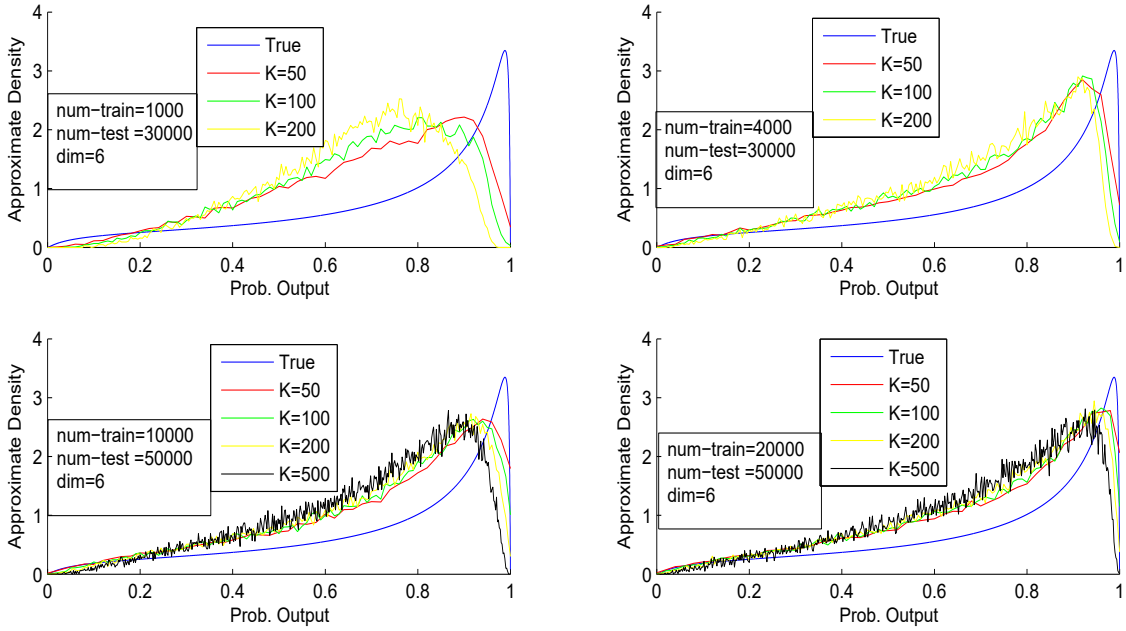

Figure 3. Empirical density approximates the true density: Effect of training sample size, test sample size and the choice of k.

iment, and the results are similar). After the necessary data preprocessing, we delete four useless attributes, which leads to a final dataset with $p = 10$ attributes. The 10 attributes are summarized in Table 6. To reveal the nature of unknown distribution of the collected data, we pick three attributes and 200 random samples from the IC dataset for demonstration. The 200 samples are plotted in Figure 4 (a)-(c), while their normal Q-Q plots are shown in Figure 4 (d)-(f). From the plots, we can conclude that the attributes do not follow the normal distribution, or any other commonly used distributions. Thus, our method is suitable in this scenario.

In phase I analysis, although the IC and OC datasets are provided separately, we still need to cluster the historical OC dataset to find out possible shift patterns. To this end, the k-means clustering algorithm is applied in this study, in which the number of clusters k is chosen to be 2 by using the well-known "elbow" method. As a result, the 270 disks are divided into two groups that contains 185 and 85 disks respectively. Since more than one OC pattern exists in the historical data, we apply the multi-cluster strategy and build the KNN-ECUSUM chart in phase II monitoring.

After clustering the historical OC data, the KNN classifier can be constructed easily following Step 1 in Table 1. The number of training samples in each cluster is selected to be 50, and the number of nearest neighbors k equals 15. In Step 2, the empirical densities can be computed immediately when the KNN classifier is provided. The estimated p.m.f.s of the categorical variable are shown in Figure 5. As in the simulation studies, Figure 5 also shows that the empirical density of the IC cluster is concentrated on the right of the figure, whereas that of the OC clusters is concentrated on the left of the figure. In particular, for the second OC cluster, its empirical density values are all 0 when the KNN output is larger than $\frac{2}{15}$, indicating that the shift size of this cluster is quite large compared to that of the first OC cluster.

In Step 3, the KNN-ECUSUM chart can be implemented for online monitoring. We

Table 6. The attributes used for the real-data analysis.

ID	Attribute Name	Description
1	Read Error Rate	The rate of hardware read errors that occurred when reading data
2	Reallocated Sector Count	The count of the bad sectors that have been found and remapped
3	Current Pending Sector Count	The count of unstable sectors
4	Airflow Temperature	The temperature of airflow
5	Spin-Up Time	Average time from zero RPM to fully operational
6	Spin Retry Count	The count of the spin start attempts to reach the fully operational speed
7	Seek Time Performance	Average time of seek operations of the magnetic heads
8	Throughput Performance	Overall read/write throughput performance
9	Available Reserved Space	Number of physical erase cycles completed as a percentage of the maximum physical erase cycles designed
10	Soft Read Error Rate	The number of uncorrectable software read errors

Table 7. OC performance comparison with the real data

	Control Limit	IC ARL	OC Cluster 1	OC Cluster 2
KNN-ECUSUM	6.14	1000(1000)	4.21(1.67)	2.28(0.45)
MEWMA	285.7	1000(997)	22.0(21.5)	1.99(0.08)
MSEWMA	27.94	1000(983)	28.9(21.3)	4.26(0.66)

fix the IC ARL to 1000, and the control limit of our chart is found to be 6.14. We compare our chart with the traditional MEWMA method and nonparametric MSEWMA chart through a steady-state simulation, where 30 IC samples are selected before the change point. The ARL values are obtained from 5,000 replicated simulations. The results are shown in Table 7. From the table, we can see that in detecting the first OC cluster, our method outperforms the competitors significantly. For the second OC cluster, the performance levels of our method and the MEWMA chart are close, and they both work slightly better than the MSEWMA chart. This demonstrates the advantage of our proposed KNN-ECUSUM chart in this real dataset.

6. Conclusions and Future Work

In this paper, we propose a novel nonparametric charting scheme for monitoring multivariate data with unknown distribution by using the KNN learning algorithm. The proposed KNN-ECUSUM control chart is derived by employing the empirical density function of the KNN output. It is implemented by making full use of historical OC data, and transforming the multivariate data into a one-dimensional categorical variable. Our method achieves satisfactory detection performance, and can also be adapted to various multivariate distributions, which is statistically appealing and easy to implement. Its usefulness has been demonstrated through extensive numerical simulations and a real case study.

There are still some theoretical and practical issues needed to be studied further. First, since our method is completely constructed from clustered data, Phase I analysis is particularly important in our method. Although we have suggested the traditional

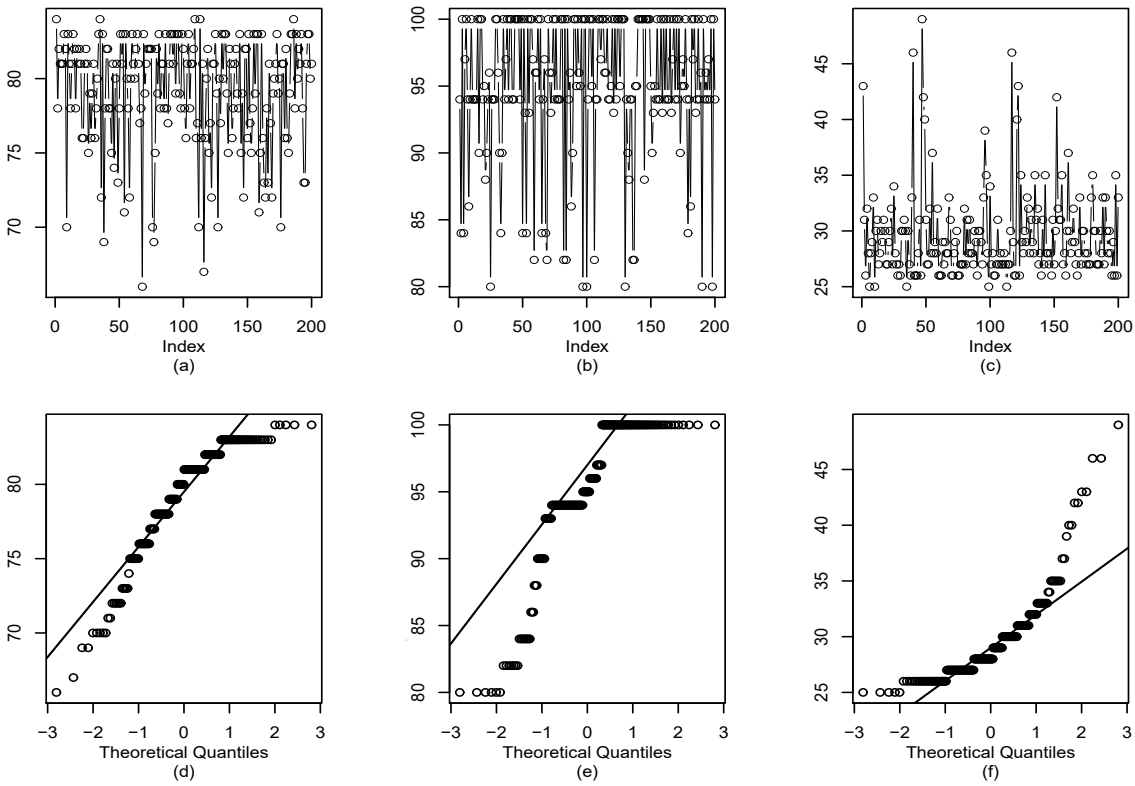

Figure 4. (a)-(c): Plots of the three attributes of 200 pieces of randomly picked data; (d)-(f): the normal Q-Q plots for the three attributes.

K-means clustering method, it is still essential to select a useful dataset that is able to capture the major historical OC shifts before online monitoring. Second, it is useful to provide more specific suggestions for determining a proper sample size, which is crucial to the effectiveness of our proposed KNN-ECUSUM control chart. While we have discussed the potential effects of sample size through simulation analysis, this may not be adequate, and it is important to determine the dependence of the sample size on the data distribution, the actual shifts and the number of OC clusters. Finally, in many real-world applications, it will be important to investigate how to deal with both spatial and temporal correlations in the context of online monitoring spatial-temporal data, which will be one of the main research directions in the future. Note that our proposed method focuses on the spatial correlation by utilizing the KNN algorithm for dimension reduction to transform multi-attribute data into univariate data, and then conduct online monitoring under the assumption of temporal independence. It will be interesting to combine our proposed KNN-based method with the existing univariate control charts designed for monitoring serially correlated data (cf., e.g., Qiu, Li, and Li 2020; Li and Qiu 2020), so that we can effectively address both spatial and temporal correlation. This is beyond the scope of this paper, and will be investigated systematically in the future.

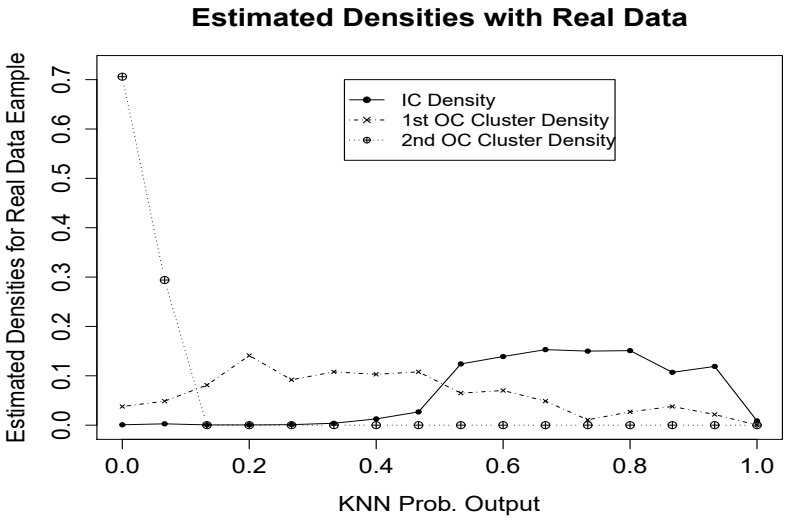

Figure 5. Empirical densities of the KNN output estimated from the real data.

Acknowledgements

The authors thank two anonymous reviewers and the Associate Editor for their thoughtful and constructive comments that greatly improved the quality and presentation of this article.

Funding

W. Li's research was supported in part by China Postdoctoral Science Foundation (2020M671064), National Science Foundation of Shanghai (19ZR1414400), and National Science Foundation of China (71931004). F. Tsung's research was supported in part by the grants, RGC GRF 16201718 and 16216119, and NSFC 71931006. Y. Mei's research was supported in part by NSF grants 1830344 and 2015405.

References

Boone, J. M., and S. Chakraborti. 2012. "Two Simple Shewhart-Type Multivariate Nonparametric Control Charts." *Applied Stochastic Models in Business and Industry* 28: 130–140.

Camci, F., Chinnam, R. B., and R. D. Ellis. 2008. "Robust Kernel Distance Multivariate Control Chart Using Support Vector Principles." *International Journal of Production Research* 46: 5075–5095.

Chakraborti, S., Van der Laan, P., and S. T. Bakir. 2001. "Nonparametric Control Charts: An Overview and Some Results." *Journal of Quality Technology* 33: 304–315.

Crosier, R. B. 1988. "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes." *Technometrics* 30: 243–251.

Hastie, T., Tibshirani, R., and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., New York: Springer.

Hawkins, D. M. 1991. "Multivariate Quality Control Based on Regression-Adjusted Variables." *Technometrics* 33: 61–75.

Hettmansperger, T. P., and R. H. Randles. 2002. "A Practical Affine Equivariant Multivariate Median." *Biometrika* 89: 851–860.

Holland, M. D., and D. M. Hawkins. 2014. "A Control Chart Based on A Nonparametric Multivariate Change-point Model." *Journal of Quality Technology* 46: 63–77.

Hwang, W., Runger, G., and E. Tuv. 2007. "Multivariate Statistical Process Control with Artificial Contrasts." *IIE Transactions* 39: 659–669.

Li, W., Pu, X., Tsung, F., and D. Xiang. 2017. "A Robust Self-Starting Spatial Rank Multivariate EWMA Chart Based on Forward Variable Selection." *Computers & Industrial Engineering* 103: 116–130.

Li, W., and P. Qiu. 2020. "A General Charting Scheme for Monitoring Serially Correlated Data with Short-Memory Dependence and Nonparametric Distributions." *IISE Transactions* 52(1): 61–74.

Lowry, C. A., Woodall, W. H., Champ, C. W., and S. E. Rigdon. 1992. "Multivariate Exponentially Weighted Moving Average Control Chart." *Technometrics* 34: 46–53.

Lowry, C. A., and D. C. Montgomery. 1995. "A Review of Multivariate Control Charts." *IIE Transactions* 27: 800–810.

Lu, X. S., Xie, M., Goh, T. N., and C. D. Lai. 1998. "Control Charts For Multivariate Attribute Processes." *International Journal of Production Research* 36: 3477–3489.

Marcucci, M. 1985. "Monitoring Multinomial Process." *Journal of Quality Technology* 17: 86–91.

Montgomery, D. C. 2009. *Statistical Quality Control: A Modern Introduction.* 6th ed., New York: Wiley.

Moustakides, G. V. 1986. "Optimal Stopping Times for Detecting Changes in Distributions." *The Annals of Statistics* 14: 1379–1387.

Ning, X., and F. Tsung. 2012. "A Density-Based Statistical Process Control Scheme for High-Dimensional and Mixed-Type Observations." *IIE Transactions* 44: 301–311.

Oja, H. 2010. *Multivariate Nonparametric Methods with R.* Springer-Verlag: New York.

Owen, A. B. 2001. *Empirical Likelihood.* Chapman & Hall/CRC.

Page, E. S. 1954. "Continuous Inspection Schemes." *Biometrika* 41: 100–114.

Qiu, P. 2008. "Distribution-Free Multivariate Process Control Based on Log-Linear Modeling." *IIE Transactions* 40: 664–677.

Qiu, P. 2014. *Introduction to Statistical Process Control.* Boca Raton, FL: Chapman & Hall/CRC.

Qiu, P. 2018. "Some Perspectives on Nonparametric Statistical Process Control." *Journal of Quality Technology* 50: 49–65.

Qiu, P., and D. M. Hawkins. 2003. "A Nonparametric Multivariate CUSUM Procedure for Detecting Shifts in All Directions." *JRSS-D (The Statistician)* 52: 151–164.

Qiu, P., Li, W., and J. Li. 2020. "A New Process Control Chart for Monitoring Short-Range Serially Correlated Data." *Technometrics* 62(1): 71–83.

Sukchotrat, T., Kim, S. B., and F. Tsung. 2010. " One-Class Classification-Based Control Charts for Multivariate Process Monitoring." *IIE Transactions* 42: 107–120.

Sun, R., and F. Tsung. 2003. "A Kernel-Distance-Based Multivariate Control Chart Using Support Vector Methods." *International Journal of Production Research* 41: 2975–2989.

Terrell, G. R., and D. W. Scott. 1992. "Variable Kernel Density Estimation." *Annals of Statistics* 20: 1236–1265.

Varmuza, K., and P. Filzmoser. 2010. *Introduction to Multivariate Statistical Analysis in Chemometrics.* CRC Press.

Wang, K., and W. Jiang. 2009. "High-Dimensional Process Monitoring and Fault Isolation via Variable Selection." *Journal of Quality Technology* 41: 247–258.

Woodall, W. H., and M. M. Ncube. 1985. "Multivariate CUSUM Quality Control Procedures." *Technometrics* 27: 285–292.

Woodall, W. H. 1997. "Control Charts Based on Attribute Data: Bibliography and Review." *Journal of Quality Technology* 29: 172–183.

Woodall, W. H., and D. C. Montgomery. 2014. "Some Current Directions in the Theory and

Application of Statistical Process Monitoring." *Journal of Quality Technology* 46: 78–94.

Zhang, C., Tsung, F., and C. Zou. 2015. "A General Framework for Monitoring Complex Processes with both In-Control and Out-of-Control Information." *Computers & Industrial Engineering* 85: 157–168.

Zou, C., and P. Qiu. 2009. "Multivariate Statistical Process Control Using LASSO." *Journal of American Statistical Association* 40: 1586–1596.

Zou, C., and F. Tsung. 2011. "A Multivariate Sign EWMA Control Chart." *Technometrics* 53: 84–97.

Zou, C., Wang, Z., and F. Tsung. 2012. "A Spatial Rank-based Multivariate EWMA Control Chart." *Naval Research Logistic* 59: 91–110.

Appendix A. Derivation of Equation (13)

By Equation (11), the true probability that a test sample $\boldsymbol{x}$ is in the IC status is given by

$$\mathrm{P}(\boldsymbol{x}) = \mathrm{P}(\boldsymbol{x} \in \mathrm{IC} | \boldsymbol{x} \in \mathrm{IC \text{ or } OC}) = \frac{g_{ic}(\boldsymbol{x})}{g_{ic}(\boldsymbol{x}) + g_{oc}(\boldsymbol{x})}.$$

As $\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6]$ is a IC sample, it follows a normal distribution (with $\boldsymbol{\mu} = [0, 0, 0, 0, 0, 0]$ and the covariance matrix equals to the identity matrix). Thus, the c.d.f of $\mathrm{P}(\boldsymbol{x})$ is

$$\begin{aligned}
\mathrm{P}(\mathrm{P}(\boldsymbol{x}) \leq t) =& \mathrm{P}(\frac{g_{ic}(\boldsymbol{x})}{g_{ic}(\boldsymbol{x}) + g_{oc}(\boldsymbol{x})} \leq t) \\
=& \mathrm{P}(\frac{\exp(-0.5 * ||\boldsymbol{x}||^2)}{\exp(-0.5 * ||\boldsymbol{x}||^2) + \exp(-0.5 * ||\boldsymbol{x} - \boldsymbol{\mu}_1||^2)} \leq t) \\
=& \mathrm{P}((||\boldsymbol{x}||^2 - ||\boldsymbol{x} - \boldsymbol{\mu}_1||^2 \geq 2 * \log \frac{1-t}{t}) \\
=& \mathrm{P}(\sum_{i=1}^{6} x_i^2 - \sum_{i=1}^{3} (x_i - 1)^2 - \sum_{j=4}^{6} x_j^2 \geq 2 * \log \frac{1-t}{t}) \\
=& \mathrm{P}(2 * (x_1 + x_2 + x_3) - 3 \geq 2 * \log \frac{1-t}{t}) \\
=& \mathrm{P}(x_1 + x_2 + x_3 \geq \frac{3}{2} + \log \frac{1-t}{t}).
\end{aligned}$$

Since x_1, x_2 and x_3 are normal i.i.d random variables, $x_1 + x_2 + x_3$ is also a normal variable with mean 0 and variance 3. Then the above equation becomes:

$$\begin{aligned}
\mathrm{P}(\mathrm{P}(\boldsymbol{x} \in IC) \leq t) =& \mathrm{P}(\frac{g_{ic}(\boldsymbol{x})}{g_{ic}(\boldsymbol{x}) + g_{oc}(\boldsymbol{x})} \leq t) \\
=& \mathrm{P}(x_1 + x_2 + x_3 \geq \frac{3}{2} + \log \frac{1-t}{t}) \\
=& \mathrm{P}(Z \geq \frac{\sqrt{3}}{2} + \frac{1}{\sqrt{3}} \log \frac{1-t}{t}) \\
=& 1 - \Phi(\frac{\sqrt{3}}{2} + \frac{1}{\sqrt{3}} \log \frac{1-t}{t}),
\end{aligned}$$

where $\Phi(z) = P(Z \leq z)$ is the cumulative distribution function of a standard normal variable $Z \sim N(0, 1)$. Therefore, the true density for $\mathrm{P}(\boldsymbol{x})$ is:

$$\begin{aligned}
g(t) =& \frac{d\mathrm{P}(\mathrm{P}(\boldsymbol{x} \in IC) \leq t)}{dt} \\
=& \frac{1}{\sqrt{6\pi}} \exp(-\frac{1}{6}(\frac{3}{2} - \log \frac{t}{1-t})^2)(\frac{1}{1-t} + \frac{1}{t}).
\end{aligned}$$