System Monitoring with Real-Time Contrasts

HOUTAO DENG
Intuit, Mountain View, CA 94043, USA
GEORGE RUNGER
Arizona State University, Tempe, AZ 85287, USA
EUGENE TUV
Intel Corporation, Chandler, AZ 85226, USA

Abstract

Monitoring real-time data steams is an important learning task in numerous disciplines. Traditional process monitoring techniques are challenged by increasingly complex, high-dimensional data, mixed categorical and numerical variables, non-Gaussian distributions, non-linear relationships, etc. A new monitoring method based on real-time contrasts (RTC) between the reference and real-time data is presented. RTC assigns one class label to the reference data, and another class to a window of real-time data, and, thus, transforms the monitoring problem to a dynamic series of classification problems. This differs from previous work that trained a classifier one time at the start of monitoring. Furthermore, a number of monitoring statistics based on the generalized likelihood ratio principle are discussed. These include error rates and class probability estimates from classifiers. Typically, the window size for the real-time data is much smaller than the size of the reference data and this class imbalance is also considered in the approach. Variable contributor diagnostics can be simultaneously obtained from the approach and are briefly discussed. Both mean and variance shifts are illustrated. Experiments are used to compare monitoring statistics and to illustrate performance advantages of RTC relative to alternative methods.

Keywords: control charts, generalized likelihood ratio, multivariate control charts, nonparametric, statistical process control, supervised learning

Introduction

Traditional statistical process control is used to detect changes from normal operating conditions (also called reference conditions). Classical multivariate control charts have been used

Dr. Deng is a Data Scientist. His email address is htaodeng@gmail.com.

Dr. Runger is a Professor in the School of Computing, Informatics, and Decision Systems Engineering. His email address is george.runger@asu.edu.

Dr. Tuv is a Senior Staff Research Scientist in the Logic Technology Department. His email address is eugene.tuv@intel.com.

to detect process mean shifts. These include Hotelling's T^2 (Hotelling, 1947), multivariate CUSUM (MCUSUM) (Runger and Testik, 2004), multivariate exponentially weighted moving averages (MEWMA) (Lowry et al., 1992), and the U^2 multivariate control chart (Runger, 1996). However, nowadays the systems to be monitored have greater complexity with high-dimensional data (Apley and Shi, 2001; Wang and Jiang, 2009), nonlinear relationships in the reference data (Ding et al., 2006; Zou et al., 2008), non-Gaussian distributions (Chou et al., 1998; Borror et al., 1999; Stoumbos and Sullivan, 2002), categorical (Wang and Tsung, 2007) or mixed (categorical and numerical) variables, missing data (Imtiaz and Shah, 2008; Prabhu et al., 2009), numerical data with different scales of measurement, etc. Furthermore, one expects a monitoring technique to avoid stringent assumptions on the parametric forms of distributions (Qiu, 2008; Zou and Tsung, 2010).

To this end, Hwang et al. (2007) converted the monitoring problem to a supervised learning problem. In this approach, artificial data are generated to represent the off-target data from the process (typically uniformly distributed data are used). One labels the artificial data as one class and the reference data as another. After labels are assigned the problem reduces to supervised learning. Hence, one then trains a classifier (supervised learner) on the two-class data and assigns future data to either the reference or off-target class based on the output from the supervised learner.

Related to this method, work by Hu et al. (2007) considered non-uniformly distributed artificial data. If a fault is expected to shift the mean vector of the process in one (or more) specific direction(s), the artificial data can be generated to tune the control algorithm to be more sensitive to these anticipated shifts. Also, work by Hu and Runger (2010) applied time-weighting to control statistics used within the artificial contrast method to improve the performance. Work by Li et al. (2006) considered a change-point problem through the artificial contrast approach. Different from the process control scenario, in the change-point problem the entire history of the data is available, and one retrospectively looks to detect a process change. The artificial contrast method with an appropriate supervised learner can successfully relax assumptions of linear models and Gaussian data and open the monitoring problem to a full set of supervised learning tools.

Other research applied machine learning algorithms to generalize methods for process control. Neural network models were used by Cook and Chiu (1998). In another direction, the relationship between one-class classification models and process control were considered (Chinnam, 2002; Sun and Tsung, 2003; Camci et al., 2008; Sukchotrat et al., 2010). In a one-class model the reference data are used to build a classifier and future observations are either assigned to the reference class or considered to be off-target. The one-class models focused on support vector machine classifiers (e.g., Camci et al. (2008)). A distance measure based on nearest-neighbors was also used (Sukchotrat et al., 2010). However, because no model was generated, we consider this approach as more similar to a distance measure that generalizes the T^2 statistic.

A key point in the previous work is that a classifier is trained only one time, either based on artificial data, or through a one-class model. However, a classifier that is trained to distinguish the most recent data from the reference data should be more sensitive to specific anomalies in the current data. Consequently, here we propose to use ubiquitous computational capabilities and modern machine learners to re-generate models quickly. The most recent data are used in real time to contrast with the reference data. With each new observation, a new classifier is trained, and statistics (such as error rate estimates from the classifiers) are monitored. The dynamic series of classifiers generate the statistics for the monitoring procedure.

Work by Hwang et al. (2007) called their detection method artificial contrasts because the reference data were compared with the artificially generated data set. Here the real-time data are contrasted with the reference data so we call it a real-time contrast (RTC). Modern computational capabilities allow one to re-build classifiers and predict classes within seconds (or fractions of seconds), so that such an approach is feasible for many monitoring applications. We present the RTC method and discuss a number of important elements, such as alternative monitoring statistics, imbalanced classification, the type of supervised learner, signal diagnosis, and so forth. We compare RTC to competitors in a number of different experiments.

The outline of the paper is as follows. The Monitoring Models section uses the generalized likelihood ratio principle as a basis to generate control statistics to monitor. The RTC Method section describes the proposed method. In the Experiments section simulated data and real data from a machine learning repository are used to demonstrate the advantages of the method. The Conclusions section provides summary comments. An Appendix provides a summary table of the notation used here.

Monitoring Models

At each time t, a p-dimensional vector of measurements is obtained from the system and denoted as $\mathbf{x}_t$. We assume that reference data from normal operating conditions are available and denoted as S_0 with sample size N_0. The reference data are considered to be a random sample drawn from a distribution $f_0(\mathbf{x})$. Denote the real-time data stream as $S_1(t) = \{\mathbf{x}_i | i = 1, 2, ..., t\}$, where t stands for the current time point. An anomalous pattern is considered to be present in the data stream if the distribution of $\mathbf{x}_t$ changes from $f_0(\mathbf{x})$ to $f_1(\mathbf{x})$, where $f_1(\mathbf{x})$ can be any distribution different from $f_0(\mathbf{x})$. The objective is to detect the change of $\mathbf{x}_t$ quickly, with a low rate of false alarms.

Neither $f_0(\mathbf{x})$ nor $f_1(\mathbf{x})$ needs to be known for our method. However, the generalized likelihood ratio (GLR) principle can be used as a guide to develop a monitoring solution. The GLR principle can be formulated as follows. Consider the following hypotheses:

$$\begin{aligned} H_0 &: \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t \sim f_0(\mathbf{x}) \\ H_1 &: \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\tau-1} \sim f_0(\mathbf{x}), \mathbf{x}_\tau, \ldots, \mathbf{x}_t \sim f_1(\mathbf{x}) \end{aligned}$$

for an unknown change time τ. The GLR test statistic is

$$\begin{aligned} l_t &= \frac{\max_\tau \{\prod_{i=1}^{\tau-1} f_0(\mathbf{x}_i) \prod_{i=\tau}^{t} f_1(\mathbf{x}_i)\}}{\prod_{i=1}^{t} f_0(\mathbf{x}_i)} && (1) \\ &= \max_\tau \frac{\prod_{i=\tau}^{t} f_1(\mathbf{x}_i)}{\prod_{i=\tau}^{t} f_0(\mathbf{x}_i)} && (2) \end{aligned}$$

Upon taking the log of the test statistic, we have

$$L_t = \max_{1 \le \tau < t} \sum_{i=\tau}^{t} \ln \frac{f_1(\mathbf{x}_i)}{f_0(\mathbf{x}_i)} \tag{3}$$

The GLR test rejects H_0 when $L_t > UCL$ with the false alarm probability $\alpha = \Pr(L_t > UCL|H_0)$, where UCL denotes the upper control limit. The GLR solution assumes that the distributional forms of both $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ are known. The assumption is strong for many applications, particularly for multi-dimensional problems with both categorical and numerical attributes. However, the GLR approach provides a guide to construct a monitoring procedure for more general data streams, which we develop in the next section. Time-weighted information might also be considered, which is similar to time-weighted control charts such as CUSUM and EWMA charts. In fact, under the assumption of a univariate normal distribution with a step mean shift, the GLR is reduced to a CUSUM control chart (Runger and Testik, 2004).

Real-Time Contrasts (RTC) Method

The GLR method monitors the statistic in equation (3) in real time. The RTC method uses a similar strategy by building a sliding window on the real-time data. A sliding window was also recommended by Apley and Shi (1999) in the context of a GLR, but this was for a very differently structured problem. While GLR searches over all possible τ to maximize equation (3), the size of the sliding window is fixed in our work. This reduces the computational complexity.

The observations in the sliding window at time t are denoted as $S_w(t)$, and the length of the window is denoted as N_w. The sliding window includes the most recent observations and is updated whenever a new observation occurs, i.e., the newest point is added to the window, and the oldest point is deleted from the window. That is, $S_w(t) = \{\mathbf{x}_{t-N_w+1}, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$. For the RTC method, the transform to supervised learning defines a class variable y associated with the observations in S_0 and $S_w(t)$ to be 0 and 1, respectively. That is,

$$y(\mathbf{x}_i) = \begin{cases} 0 & \mathbf{x}_i \in S_0 \\ 1 & \mathbf{x}_i \in S_w(t) \end{cases} \tag{4}$$

Now a classifier (supervised learner) can be built on the data from the two classes. The classification error rates provide information on the control of the process. That is, when there is no shift, the data from both classes are essentially from the same distribution, and then the error rates are expected to be higher. Under a shift, the data are from different distributions, and then the error rates are expected to be lower. Some classifiers also provide class probability estimates and these can be more informative than only the error rates. Consequently, probability estimates can also be used for monitoring.

The RTC method is graphically illustrated in Figure 1. The data in the window at $t = 10$ is shown as a rectangle and labeled as $S_w(10)$. A classifier (denoted as Classifier(10)) is built to compare $S_w(10)$ to S_0. Results from this classification are summarized in a control chart at time $t = 10$. Then the time step is increased one unit to $t = 11$, and a new classifier is built to compare $S_w(11)$ to S_0 (denoted as Classifier(11)). Similarly, results from this classification are summarized in a control chart at time $t = 11$. This process continues through time.

We also note that the reference data can be updated over time. In traditional control charts, when the process is monitored and no shift is detected, one might re-estimate the control limits based on all the currently available data. Similarly, one might combine the current data with the reference data if no shift is detected. However, typically control charts fix the control limits and then evaluate performance on new data (in a Phase II analysis), and we use the same approach here. But if the reference data were not sufficient, one could consider an update to increase the size of the reference data over time.

Also, the window size N_w here has a similar role to the subgroup size in an $\bar{X}$ control chart. Larger windows are expected to be more sensitive to smaller shifts and vice versa. In an $\bar{X}$ chart one would select a subgroup that performs reasonably well over a range of off-target shifts, and the same objective holds for a window size. The effects on performance from different window sizes are compared in our experiments.

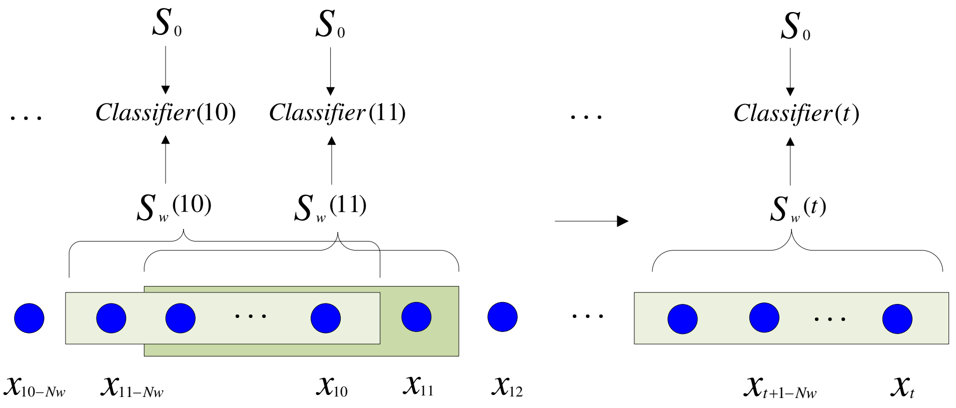


Figure 1: Framework for the real-time contrasts (RTC) method. Each filled circle represents an observation. The window length is N_w and the windows are shown at three different times $\{10, 11, t\}$

.

Supervised Learner: Random Forest

Any classifier can be used for real-time contrasts. However, a preferred learner should handle reference data with complex structures, and should provide class probability estimates for observations so that the strength of a class assignment can be used for monitoring decisions.

Tree-based classifiers can handle numerical or categorical data, missing data, different scales (units) between variables, interactions and nonlinearities, and they are robust to outliers. Furthermore, tree-based models produce effective probability estimates (Perlich et al., 2003). Therefore, here we consider a tree-based classifier known as a random forest (RF) learner (Breiman, 2001).

A RF builds a parallel ensemble of trees based on re-sampling the training data. Each tree in a RF is built on a random sample (with replacement) from the training data $\{S_0, S_w(t)\}$. At each node in each tree, m variables are selected at random from the total number of variables p. Here m is a fixed parameter that adjusts the complexity and performance of the model. However, work by (Breiman, 2001) showed that a RF is not sensitive to values of m over a broad range and recommended $m = \sqrt{p}$. We use this recommended value in our analysis. Each of the m variables is individually scored, with an impurity measure such as the Gini index (defined in the Fault Diagnosis via Variable Importance section), to best separate the classes based on a simple rule. For example, rows with $x_j \leq c$ and $x_j > c$ are assigned to the left and right child nodes, respectively, to increase the purity of the classes in each child.

Because each tree in a RF is constructed from a random sample with replacement, for each observation $\mathbf{x}_i$ in the training data, typically some trees in the RF are grown with $\mathbf{x}_i$ omitted. The set of observations omitted from a tree are known as the out-of-bag (OOB) samples of the tree. Let $nTree$ denote the number of trees in the RF. Let $\hat{y}(\mathbf{x}_i, T_j)$ denote the prediction for $\mathbf{x}_i$ from tree T_j for $j = 1, 2, \ldots, nTree$. Let OOB_i denote the set of trees where $\mathbf{x}_i$ is an OOB sample and $|OOB_i|$ be the number of trees in OOB_i. The OOB probability estimate for $\mathbf{x}_i$ belonging to class $k \in \{0, 1\}$ is

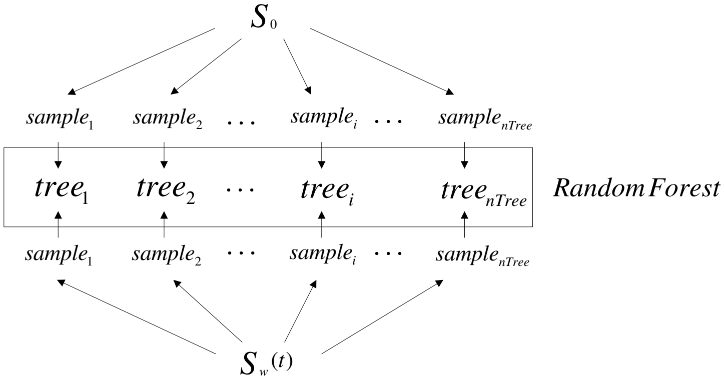
$$\hat{p}_k(\mathbf{x}_i) = \frac{\sum_{j \in OOB_i} I[\hat{y}(\mathbf{x}_i, T_j) = k]}{|OOB_i|} \tag{5}$$

where $I(\cdot)$ is an indicator function that equals one if its argument is true and zero otherwise. Also, the OOB predicted class of instance $\mathbf{x}_i$ is $\hat{y}(\mathbf{x}_i) = 1$ if $\hat{p}_1(\mathbf{x}_i) > 0.5$ and $\hat{y}(\mathbf{x}_i) = 0$ otherwise. The error rates computed from OOB predictions are easily obtained and have been shown to be good estimates of error rates on new, test data (Breiman, 2001). Consequently, OOB error rates and probability estimates are used in this work.

Preliminary experiments that we conducted showed that N_w (the size of $S_w(t)$) should be selected to be far less than N_0 (the size of S_0). Similar to a large subgroup size, too large a window size can include excessive historical data in the window and delay the response of the monitoring scheme to a process change. Consequently, the classifier in the RTC method can expect many more observations in class 0 than class 1 (and this is known as class imbalance (Byon et al., 2010)). For example, if $N_0 = 990$ and $N_w = 10$ a classifier that always predicts class 0 only generates an error in 10 rows out of 1000. Still, class 1 data are never predicted correctly and this classifier is not useful for process monitoring. The classifier in RTC should be designed for class imbalance.

Some classifiers allow one to assign prior probabilities to the classes to better equalize the error rates between the classes. A well-known example is the use of prior probabilities in linear discriminant analysis (Morrison, 1967). Tree models can also incorporate prior probabilities (Breiman et al., 1984), but not all classifiers easily integrate such priors. Furthermore, one would use prior probabilities in order to incorporate all data in S_0. But N_0 can be large enough to result in a computational burden. We prefer a simpler approach that can be applied to any classifier and that simultaneously reduces the computations.

Stratified sampling is another method to adjust for the class imbalance. The data in $S_w(t)$ can be up-sampled so that multiple replicates of the observations in $S_w(t)$ are used in the classifier. But if N_0 is of magnitude in the thousands, and N_w is of magnitude in the tens, extensive up-sampling is needed to approximately balance the classes. Conversely, the data in S_0 can be down-sampled. But a small sample from S_0 might not adequately represent the distribution, and this could result in false anomalies signaled. A convenient feature of an ensemble learner such as an RF is that multiple tree models are built on sampled data. Consequently, rather than training on a single sample from S_0, one can build each tree from an independent random sample (with replacement) from S_0. The collection of samples is expected to better represent the S_0 distribution, and therefore, to better contrast with the real-time data. Furthermore, the down-sampling reduces the computations for each tree and has the additional benefit that the trees in the RF are less correlated. The advantages of reduced correlations were discussed by Breiman (2001). Let k_0 and k_w denote the number of samples per tree selected from S_0 and $S_w(t)$, respectively. Consequently, down sampling with $k_0 = k_w = N_w$ is used here. Figure 2 illustrates the sampling procedure using a RF classifier.



S_0
$sample_1$ $sample_2$ … $sample_i$ … $sample_{nTree}$
$tree_1$ $tree_2$ ⋯ $tree_i$ $tree_{nTree}$
Random Forest
$sample_1$ $sample_2$ ⋯ $sample_i$ ⋯ $sample_{nTree}$
$S_w(t)$

Figure 2: Sampling procedure for constructing one classifier of Figure 1 using a random forest classifier

Monitoring Statistics

Under the RTC framework, a classifier such as a RF is trained to classify S_0 and $S_w(t)$ at time t, then the error rates and probability estimates are used to derive monitoring statistics. The error rate from either class is an obvious choice for monitoring and a simple starting point. The error rates from the S_0 and $S_w(t)$ data at time t are denoted as

$$err(S_0, t) = \frac{\sum_{\mathbf{x}_i \in S_0} \hat{y}(\mathbf{x}_i|t)}{N_0} \tag{6}$$

and

$$err(S_w, t) = \frac{\sum_{\mathbf{x}_i \in S_w(t)} (1 - \hat{y}(\mathbf{x}_i|t))}{N_w} \tag{7}$$

respectively, where $\hat{y}(\mathbf{x}_i|t)$ denotes the predicted class based on the OOB data. When the real-time data stream shifts, the error rates for both classes should decrease. We prefer the statistics to increase when there is a shift. Therefore, we use the accuracy from either S_0 or $S_w(t)$ data for monitoring. Still, the results for accuracy and its corresponding error rate are equivalent. The accuracies are denoted as

$$a(S_0, t) = 1 - err(S_0, t) \tag{8}$$

and

$$a(S_w, t) = 1 - err(S_w, t) \tag{9}$$

To illustrate how RTC works, two simple examples are simulated. In the example shown in Figure 3, both class 0 data (200 observations) and class 1 data (10 observations) are from bivariate normal distributions with zero mean vectors and identity covariance matrices. We refer to such distributions as standard normal distributions. The OOB classification results of the class 0 data using RF are shown in Figure 4. It is obvious that when the data are from the same distribution, the OOB error rate is high (and equal to 37.5% in this case). In the example shown in Figure 5, the class 0 data (200 observations) are still from a bivariate standard normal distribution, but the class 1 data (10 observations) have a two-unit mean shift in one variable. The OOB classification results of the class 0 data from RF are shown in Figure 6. It can be seen that the OOB error rate is lower when the two classes are different (13.5% in this case).

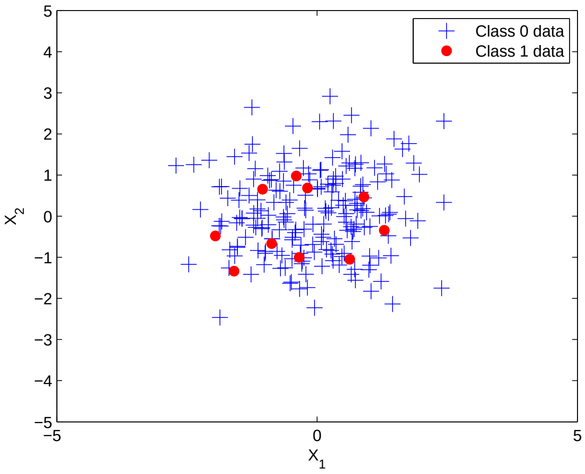


Figure 3: Class 0 and class 1 data are from the same distribution.

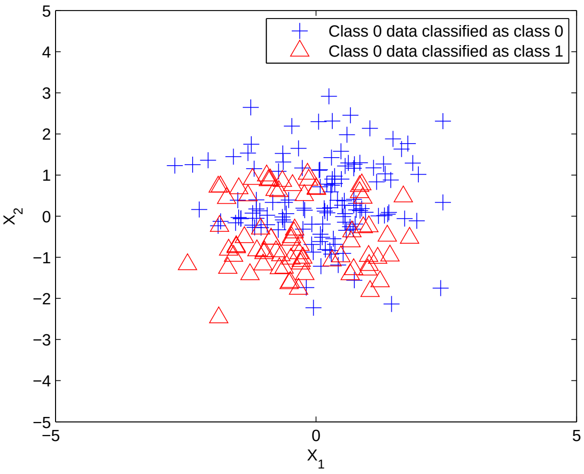


Figure 4: Class 0 and class 1 data are from the same distribution. OOB error rate for class 0 data points = 37.5%.

Besides error rates, two statistics based on class probability estimates related to S_0 and $S_w(t)$ can be considered. Let $\hat{p}_k(\mathbf{x}_i|t)$ denote the OOB class probability estimate $\hat{p}_k(\mathbf{x}_i)$ calculated from equation (5) at time t for $k = 0, 1$. Both S_0 and $S_w(t)$ contain a number of observations. Therefore, to summarize the results from the classifier, we compute the average of these estimates over S_0 and $S_w(t)$, and we consider either as a potential monitoring statistic. For $\mathbf{x}_i \in S_0$

$$p(S_0, t) = \frac{\sum_{\mathbf{x}_i \in S_0} \hat{p}_0(\mathbf{x}_i|t)}{N_0} \tag{10}$$

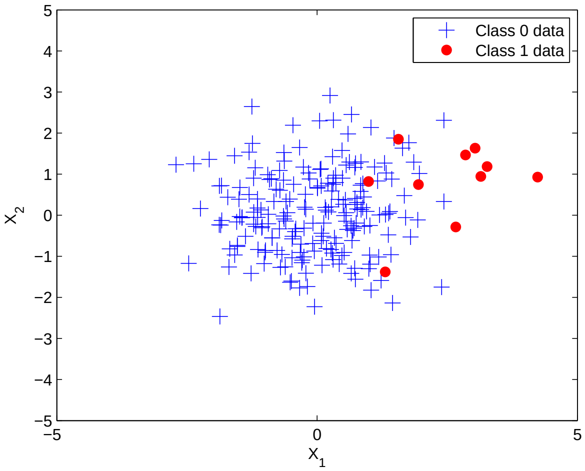


Figure 5: Data in the window has a two-unit shift in the mean.

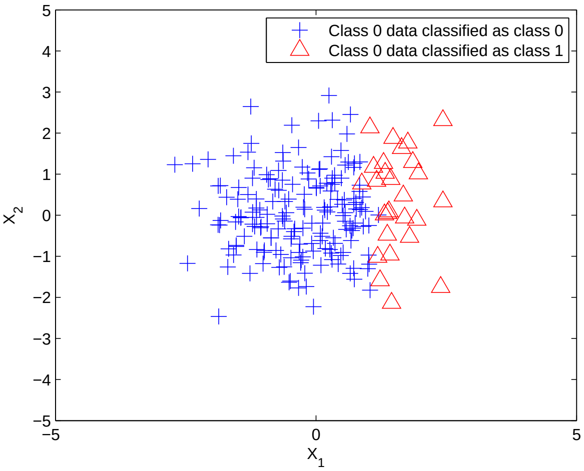


Figure 6: Data in the window has a two-unit shift in the mean. OOB error rate for class 0 data points = 13.5%.

and for $\mathbf{x}_i \in S_w$

$$p(S_w, t) = \frac{\sum_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)}{N_w} \tag{11}$$

Because $\hat{y}(\mathbf{x}_i|t)$ discretizes $\hat{p}_1(\mathbf{x}_i|t)$, the probability estimates are related to error rates, but contain more information about the class assignment.

In addition, a statistic directly based on GLR equation (3) can be derived

$$glr(t) = \sum_{\mathbf{x}_i \in S_w(t)} \ln \frac{\hat{p}_1(\mathbf{x}_i|t)}{\hat{p}_0(\mathbf{x}_i|t)} \tag{12}$$

The statistic $glr(t)$ is related to $p(S_w, t)$. On the one hand, $glr(t)$ is a strictly monotonic function of $\sum_{\mathbf{x}_i \in S_w(t)} \ln \hat{p}_1(\mathbf{x}_i|t)$, therefore it is also strictly monotonic with $\prod_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$. On the other hand, $p(S_w, t)$ is a strictly monotonic function of $\sum_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$. When the difference between the $S_w(t)$ and S_0 distributions is moderate, $\hat{p}_1(\mathbf{x}_i|t)$ (where $\mathbf{x}_i \sim S_w(t)$) should increase, and, thus, both $\prod_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$ and $\sum_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$ increase. Therefore, $glr(t)$ and $p(S_w, t)$ should have similar performance for moderate shifts. However, when the distributions of $S_w(t)$ and S_0 differ substantially, $\hat{p}_1(\mathbf{x}_i|t)$ (where $\mathbf{x}_i \sim S_w(t)$) increases substantially, and $\prod_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$ should have a larger change than $\sum_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$. This is consistent with the results from our later experiments, where $glr(t)$ and $p(S_w, t)$ have similar monitoring performance for small mean shifts, but $glr(t)$ is slightly better for large mean shifts.

A simpler statistic is based only on the most recent term in the GLR statistic

$$l(t) = \ln \frac{\hat{p}_1(x_t)}{\hat{p}_0(x_t)} \tag{13}$$

Although only the most recent probability estimate is used in $l(t)$, this estimate is based on a classifier that considers all the data in the most recent window. Furthermore, an EWMA control chart can be applied to $l(t)$, denoted as $l_E(t)$. A similar statistic was considered by Hu and Runger (2010), but a classifier was only trained initially against the artificial data. The estimates for $\hat{p}_1(x_t|t)$ and $\hat{p}_0(x_t|t)$ are much different here because classifiers are re-trained (and new probability estimates are obtained) every time there is a new observation.

As a practical issue, because probability estimates may be exactly zero or one, we set $\hat{p}_1(\mathbf{x}_i|t) = nTree/(nTree{+}1)$ when $\hat{p}_1(\mathbf{x}_i|t) = 1$ and $\hat{p}_1(\mathbf{x}_i|t) = 1/(nTree{+}1)$ when $\hat{p}_1(\mathbf{x}_i|t) = 0$. From the probability estimates defined in equation (5), the next most extreme values are $\hat{p}_1(\mathbf{x}_i|t) = (|OOB_1| - 1)/|OOB_i| < nTree/(nTree + 1)$ and $\hat{p}_1(\mathbf{x}_i|t) = 1/|OOB_i| > 1/(nTree + 1)$, so that one does not need to be concerned about additional truncation of probability estimates to zero or one.

Given a monitoring statistic M_t, a control limit UCL is selected, and a signal is generated if $M_t > UCL$. The choice for UCL is a tradeoff between false and true signals (as is usual for a monitoring problem). The distribution of any monitoring statistic proposed in the previous sections is complex, and it depends on the specific classifier as well as the original data distribution. Furthermore, windows corresponding to consecutive observations overlap many of the same data so that autocorrelation between monitored statistics is expected. Consequently, the ability of computers is again exploited to simulate the run length performance for a statistic M_t from the reference data in S_0 and determine control limits. The details are described in the experiments. Preliminary work considered models for the autocorrelation based on autoregressive approaches. However, the objective is to detect a change in M_t from its baseline value determined from the reference data, rather than a change to the autoregressive structure. Consequently, a simple constant control limit is applied here to monitor the system. Comments related to control limits for autocorrelated data were provided by Runger (2002).

Fault Diagnosis via Variable Importance

In addition to monitoring, fault diagnosis is also an important and challenging issue in multivariate process monitoring (Runger et al., 1996; Li et al., 2008; Kim et al., 2011). In the RTC framework, the fault diagnosis problem can be handled by scoring the importance of variables to the classifier at the time a signal is generated. The most important variables are considered the key contributors to the signal.

A RF provides a measure of variable importance embedded in the forest of trees. At every node of every tree in a RF an exhaustive search scores the m selected variables and splits to achieve the maximum reduction in impurity. Consequently, this approach implicitly considers the importance of a variable to the model with the impurity reduction as a measure of the relative importance (Breiman et al., 1984). For a single decision tree, the measure of importance for variable X_j is

$$VI(X_j, T) = \sum_{v \in T} \Delta I(X_j, v), \tag{14}$$

where $\Delta I(X_j, v)$ is the decrease in impurity due to a split on variable X_j at a node v of the tree T (Breiman et al., 1984). Here $I(v) = Gini(v)$ is the Gink index at node v, defined as

$$Gini(v) = \sum_{k=1}^{K} \hat{p}_k^v (1 - \hat{p}_k^v)$$

and $\hat{p}_k^v$ is the proportion of class k observations in v. The decrease $\Delta I(X_j, v)$ is the difference between the impurity at the node v and the weighted average of impurities at each child node of v. The weights are proportional to the number of observations that are assigned to each child from the split at node v so that $\Delta I(X_j, v) = I(v) - w_L I(v_L) - w_R I(v_R)$ where $I(v_L)$ and $I(v_R)$ are the impurity scores, and w_L and w_R are the weights for the left and right child nodes, respectively.

For an ensemble of $nTree$ trees this importance measure is easily generalized to the average over the trees

$$VI(X_j) = \frac{1}{nTree} \sum_{i=1}^{nTree} VI(X_j, T_i). \tag{15}$$

The averaging makes this measure more reliable. This relative importance measure automatically incorporates variable interaction effects and, thus, it is different from univariate measures of importance such as t-tests of individual variables.

We simulated a 10-dimensional example and a 100-dimensional example. For both examples, 2000 reference data are standard normally distributed. For the off-target data, the mean of one variable (X_1) shifts 2 standard deviations. The real-time data are composed of 200 in-control observations followed by 200 shifted observations. That is, the mean shift occurs at the 201th observation. Figures 7 and 8 plot the importance of X_1 and other variables for the 10- and 100-dimensional cases, respectively. It is obvious that the importance of X_1 increases after the 201th. Consequently, both plots indicate that X_1 may be a contributor to the signal.

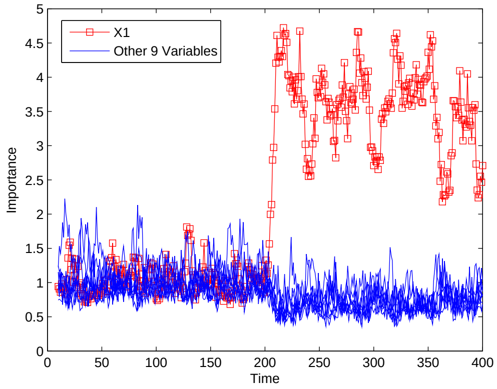


Figure 7: Plot of importance of X_1 and all other variables for the 10-dimensional case. The importance increase of X_1 is obvious after the 201th observation, where X_1 shifts 2 standard deviations.

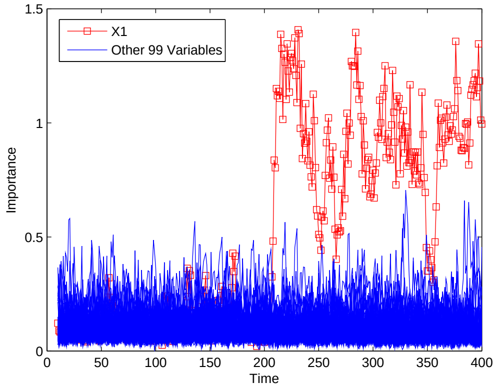


Figure 8: Plot of importance of X_1 and all other variables for the 100-dimensional case. The importance increase of X_1 is obvious after the 201th observation, where X_1 shifts 2 standard deviations.

Experiments

We first investigate the performance of RTC for detecting mean shifts for normal distributions in both 10 and 100 dimensions. Different monitoring statistics with varying window sizes are compared. The performance of RTC is also compared to the artificial contrasts method (AC) described by Hu and Runger (2010). This artificial contrast method had better performance than the earlier work by Hwang et al. (2007). We also compare to a MEWMA (that applies exponential weighting to the vector of observations at each time (Lowry et al., 1992)) and Hotelling's T^2 (Hotelling, 1947). We also illustrate some non-standard situations, such as a combination of mean and covariance shift, covariance shift, non-normal distributions and mixed categorical and numerical data.

In all experiments, we use sample sizes $k_0 = k_w = N_w$ and the number of trees $nTree = 500$ in the RF. The experiments are run in R (R Development Core Team, 2009) and Matlab software packages.

Average Run Length Evaluation

We use average run length (ARL) to evaluate the efficiency for detecting distribution changes. To obtain a control limit for each statistic, we first independently generate R replicates of reference data sets, each with N_0 observations. Denote these as $S_0^1, ..., S_0^k, ..., S_0^R$. Corresponding to each reference data set S_0^k, generate one replicate of data S_1^k which corresponds to data from the shifted condition. To set the control limit, the distribution of S_1^k is the same as S_0^k, and sufficient data are generated to obtain a signal from trial control limits. Because the RTC method needs N_w real time data in the sliding window, $N_w - 1$ data from the distribution of S_0 are simulated and added to the beginning of these data streams.

Then we apply the RTC method to S_1^k based on the reference data S_0^k. Denote the monitoring statistic at time t in the k^{th} replicate as M_t^k . Given a trial upper control limit UCL for the monitoring statistic, denote RL_k as the run length until $M_t^k > UCL$. Then ARL is estimated as

$$\frac{1}{R}\sum_{k=1}^{R} RL_k$$

The control limit UCL is selected so that ARL_0 is approximately 200. Standard errors of the run lengths can also be computed.

By a similar procedure, R replicates of shifted data are simulated and ARL_1 can be obtained according to the UCL under $ARL_0 \approx 200$. We use ARL_1 to evaluate the performance under distribution shifts. In our experiments, we use $R = 1000$ replicates for our primary ARL comparisons in Table 1, and we also provide standard error estimates. Other results use $R = 100$ replicates.

Our ARL evaluation approach is used to estimate control limits precisely so that the ARL performance of the monitoring schemes could be compared to alternatives. In practice, only one reference data set S_0 is expected to be available, and the control limit would be estimated from bootstrap samples from S_0. In this case bootstrap samples would also be selected to represent the real-time data and set the control limit. A benefit of the down-sampling approach to compensate for class imbalance (discussed previously) is that the bootstrap sample sizes are equal to the relatively small window size (approximately of size 10). Consequently, a single reference data set S_0 can be used.

In an example, a single reference data set S_0 ($N_0 = 1000$) was used to estimate the control limit. A random ordering was used to generate S_w data and bootstrap samples were used to calculate values for the statistic $p(S_0, t)$. The S_0 data were not changed, but the randomized procedure was repeated to determine a control limit based on 100 signals. The objective was to compare the control limit estimated from this simple approach (that applies to only a single reference data set S_0) with the estimate based on simulated data. Over five replicates (each one based on 100 signals) the mean and standard deviation of the control limit estimates for $p(S_0, t)$ were 0.66 and 0.001, respectively, and this compares well to the control limit estimate of 0.66 in Table 1 obtained from more extensive simulation. For another statistic, such as $l(t)$, the mean and standard deviation of the control limit estimates were 0.75 and 0.015, respectively. This can be compared to the control limit estimate of 0.84 in Table 1 from the more extensive simulations. Although the control limit estimate is not as good for this statistic, the ARL is not overly sensitive to such departures and potentially greater sample sizes could improve the results. Future work can explore control limit estimates more extensively.

Mean Shifts in 10- and 100-Dimensional Normal Data

First consider 10-dimensional and 100-dimensional standard normal data as the reference data. The size of the reference data is set to 2000 for both cases.

The magnitude of a shift is measured by

$$\delta = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)} \tag{16}$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are the mean before and after the shift, respectively, and $\boldsymbol{\Sigma}$ is the covariance matrix (assumed known). Because the covariance matrices are identity matrices, here Euclidean and Mahalanobis distances are equal.

Two out-of-control situations are considered for the 10 dimensional case: 5 variables each shifts one standard deviation ($\delta = \sqrt{5}$) and 10 variables each shifts one standard deviation ($\delta = \sqrt{10}$). For the 100-dimensional experiments, first consider a shift in one variable of 2 standard deviations ($\delta = 2$). This case tests the ability of RTC to detect a change in only a one-dimensional subspace among 100. Then consider the case where 10 variables each shifts one standard deviation in order to generate the same δ as in the second 10-dimensional case. Although the ARL of a chart based on a traditional T^2 statistic depends on the shift only through the parameter δ, the same is not the case for all learners used in RTC. Tree models generate axis-parallel partitions of the data that result in directional sensitivity of the method. Therefore, experiments that shift several variables an equal magnitude are expected to be unfavorable for the tree models. Still, the ensemble of trees tend to compensate for the axis-parallel concern, and the performance can be evaluated in the following results.

Figures 9-12 illustrate the ARL_1s of different monitoring statistics with varying window sizes for the 10- and 100-dimensional cases. The figures use the same scale for ARL_1. The monitoring statistic $a(S_w, t)$ in equation(9) is not shown in the figures. There are few values for the error rates ($\{0, \frac{1}{N_w}, \frac{2}{N_w}, ..., 1\}$), and, thus, UCL is not always available for $a(S_w, t)$

under $ARL_0 \approx 200$. Other experiments (not shown here) indicate that $p(S_w, t)$ is a better alternative than $a(S_w, t)$.

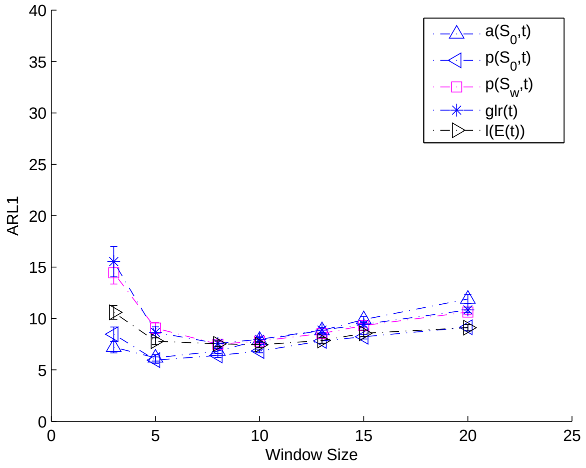


Figure 9: RTC with selected statistics for 10-dimensional normal data with a 1 standard deviation shift in 5 variables. ARL1 and standard errors are shown.

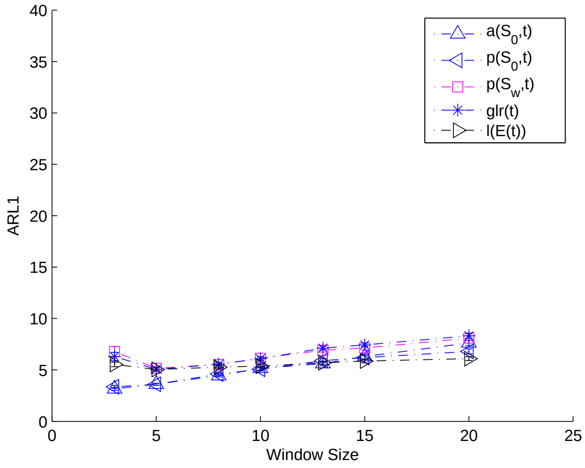


Figure 10: RTC with selected statistics for 10-dimensional normal data with a 1 standard deviation shift in all 10 variables. ARL1 and standard errors are shown.

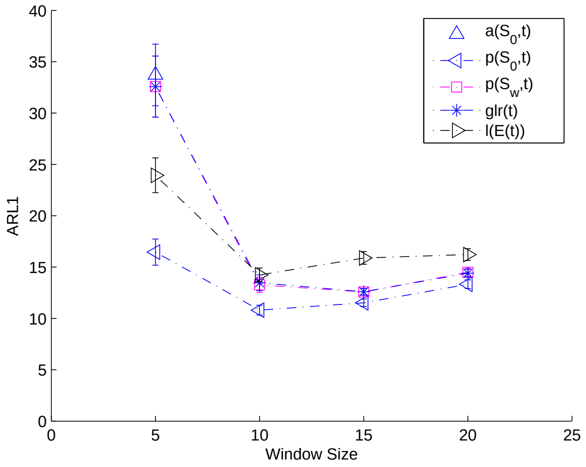


Figure 11: RTC with selected statistics for 100-dimensional normal data with a 2 standard deviation shift in one variable. ARL1 and standard errors are shown.

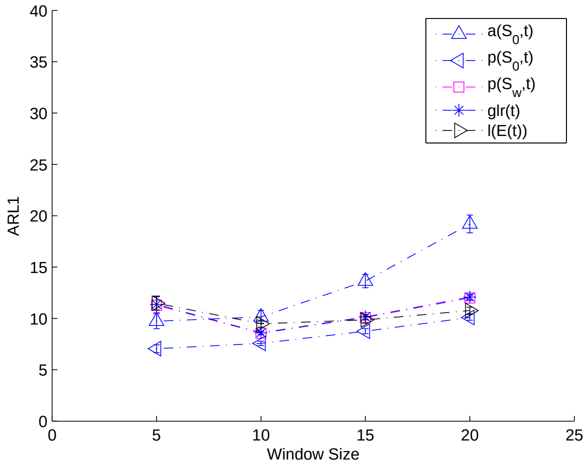


Figure 12: RTC with selected statistics for 100-dimensional normal data with a 1 standard deviation shift in each of 10 variables. ARL1 and standard errors are shown.

For larger shifts (Figures 10 and 12), the performance is not as sensitive to the window size as for cases with smaller shifts (Figures 9 and 11). Still, a smaller window size has slight advantages for the larger shifts. These results provide some guidance to select $N_w(t)$. The window size should be larger when the distribution change is smaller. This is similar to the usual result for univariate control charts. Larger subgroup sizes are more effective for smaller shifts. In the examples $N_w \in [10, 15]$ seems to be appropriate choices for all cases considered here. If the magnitude of the shift is known to be large, then a small window size such as $N_w = 5$ might be preferred.

In addition to ARL_1, the changing magnitude of monitoring statistics may be also important for visualization. We observe that larger window sizes lead to larger magnitude changes of the monitoring statistics. To demonstrate this, 200 in-control data followed by 200 out-of-control data are simulated in both 10 and 100 dimensions. The in-control data for both cases are standard normally distributed. Then 5 variables shift one standard deviation for the 10 dimensional case and 1 variable shifts two standard deviations for the 100-dimensional case. Figures 13-18 plot the monitoring statistic $p(S_0, t)$ in all cases. It can be seen that as the window size increases, the change in the monitoring statistic is greater in magnitude, but there is an expected longer delay for the statistic to increase.

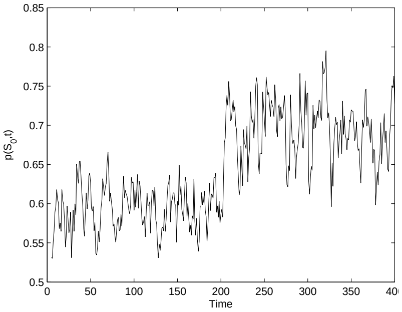


Figure 13: Plot of RTC with $p(S_0, t)$ under $N_w = 5$ versus time in 10 dimensions. The change of $p(S_0, t)$ is obvious after the 201^{th} point.

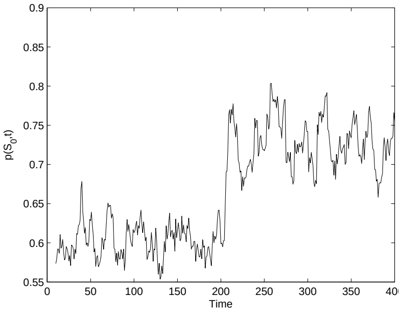


Figure 14: Plot of RTC with $p(S_0, t)$ under $N_w = 10$ versus time in 10 dimensions. The change of $p(S_0, t)$ is obvious after the 201^{th} point.

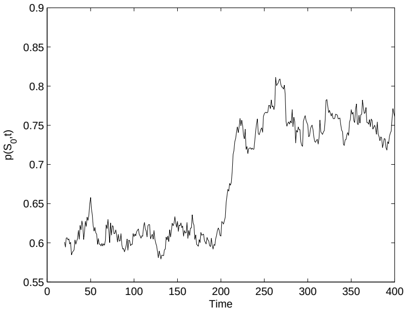


Figure 15: Plot of RTC with $p(S_0, t)$ under $N_w = 20$ versus time in 10 dimensions. The change of $p(S_0, t)$ is obvious after the 201^{th} point.

Consider the monitoring statistics next. Here $p(S_0, t)$ dominates all other statistics in 100 dimensions and is still one of the best statistics in 10 dimensions. The statistic $p(S_0, t)$ is based on N_0 observations and other statistics such as $p(S_w, t)$, $glr(t)$ are based on $N_w(t)$ samples. Generally, $N_w \ll N_0$. Therefore $p(S_0, t)$ might be expected to be a more stable estimate. Also, $a(S_0, t)$ performs well in 10 dimensions, but becomes much worse in the 100 dimensions. In cases with smaller shifts in 100 dimensions the ARL_1 for $a(S_0, t)$ are greater than 40 and thus are not shown in the Figure 11. From Table 1 it can be seen that the control limit for $a(S_0, t)$ increases from 0.88 in 10 dimensions to 0.99 in 100 dimensions, while the maximum value of $a(S_0, t)$ is 1.

Furthermore, $l_E(t)$, $p(S_w, t)$ and $glr(t)$ perform reasonably well in all cases. The statistics $p(S_w, t)$ and $glr(t)$ have similar ARL_1s. This is not surprising because of the relationship between $\prod_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$ and $\sum_{\mathbf{x}_i \in S_w(t)} \hat{p}_1(\mathbf{x}_i|t)$ as mentioned previously. We also mentioned that $glr(t)$ may be more sensitive than $p(S_w, t)$ for large shifts. To investigate whether $glr(t)$ has an advantage over $p(S_w, t)$ or even $p(S_0, t)$ for large shifts, a third 10-dimensional case with all variables shifting 5 standard deviations is considered. The ARL_1s for the different

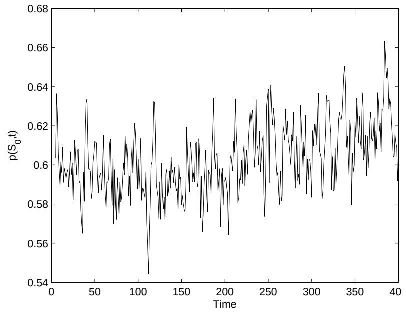


Figure 16: Plot of RTC with $p(S_0, t)$ under $N_w = 5$ versus time in 100 dimensions. The change of $p(S_0, t)$ is not obvious after the 201^{th} point.

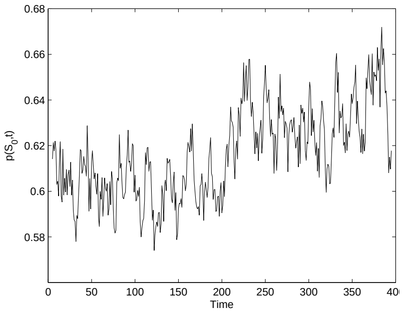


Figure 17: Plot of RTC with $p(S_0, t)$ under $N_w = 10$ versus time in 100 dimensions. The change of $p(S_0, t)$ is obvious after the 201^{th} point.

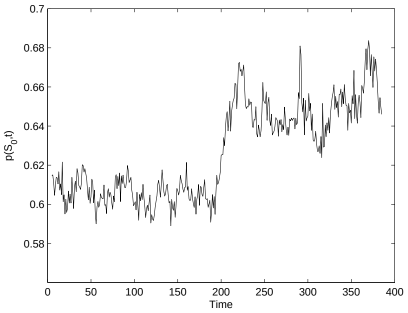


Figure 18: Plot of RTC with $p(S_0, t)$ under $N_w = 20$ versus time in 100 dimensions. The change of $p(S_0, t)$ is obvious after the 201^{th} point.

shifts in 10 dimensions using $p(S_0, t)$, $p(S_w, t)$, $glr(t)$, all with $N_w = 10$ are shown in Figure 19. It can seen that for the two smaller shifts, the performance of $p(S_w, t)$ and $glr(t)$ is close. When the shift is larger, $glr(t)$ is obviously better than $p(S_w, t)$.

Based on the previous discussion regarding these examples, $p(S_0, t)$ with $N_w \approx 10$ is a reasonable choice for monitoring. Therefore, we focus on $p(S_0, t)$ with $N_w = 10$ in later experiments.

We also study the size of the reference data. Reference data sizes (N_0) from 20 to 10,000 are considered for the same 10-dimensional cases as before. We use RTC with $p(S_0, t)$ and $N_w = 10$. The reference data are standard normally distributed. The shifts in 5 or 10 variables are 1 standard deviation each. These experiments are conducted in the same manner as the previous simulations with 1000 replicates, except only 100 replicates are used here. The ARL_1s and their standard errors are shown in Figure 20. Although there are some differences when the size of the reference data is as small as 20, in general, as the size of the reference data increases, the ARL_1s are stable.

To illustrate the effectiveness of the method, we also compare RTC to artificial contrasts (AC), a MEWMA control chart and Hotelling's T^2 control chart in the previous cases. For MEWMA, we set $\lambda = 0.2$. For AC, we use the time-weighted method (denoted as AC(EWMA)) and the default setting from Hu and Runger (2010). Because we know the parameters for the normal distribution, the T^2 control chart is also a χ^2 chart. The known parameters were also used in the MEWMA. We used the known parameters to eliminate the effects of parameter estimates on the performance when we compare these charts to RTC. This treats these charts favorably because, in practice, the parameters would need to be estimated.

Table 1 provides the results of the previous methods and RTC with different statistics under $N_w(t) = 10$. The table shows that some RTC statistics provide excellent performance, particularly for the higher-dimensional case. These are normally distributed data so that a method such as the MEWMA is expected to perform well here. The RTC method is a generalized approach, that is not expected to improve upon standard methods for multivari-

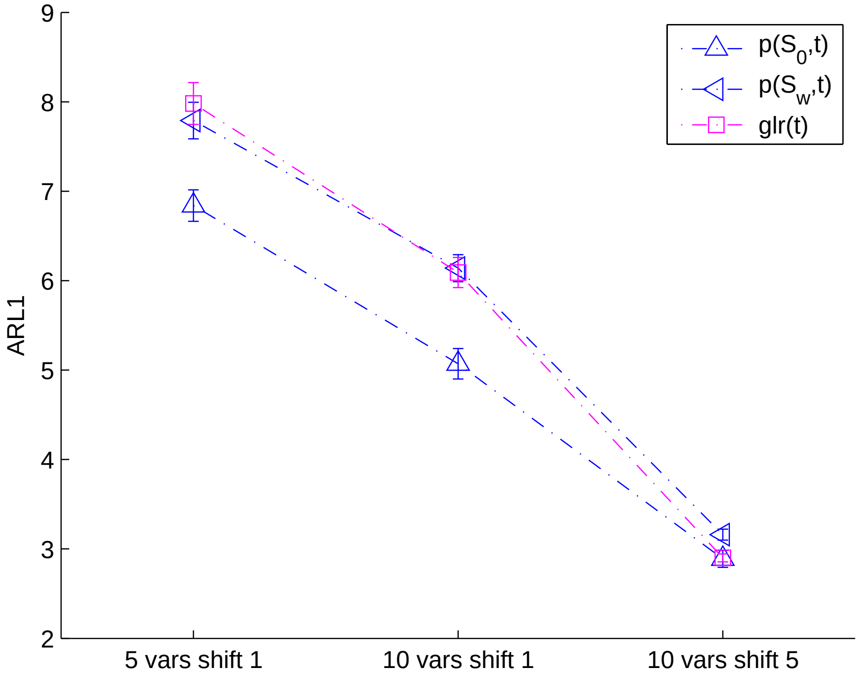


Figure 19: RTC statistics $p(S_w,t)$,$p(S_0,t)$, and $glr(t)$ with $N_w = 10$ compared for selected shift magnitudes. ARL_1s and standard errors are shown. The reference data are 10-dimensional, normally distributed, with zero mean vectors and identity covariance matrices.

ate normal data. Still, the RTC method is better than MEWMA for the smaller shift in 100 dimensions, and competitive to MEWMA in other cases. The advantage of RTC with $p(S_0,t)$ over $AC(EWMA)$ and T^2 becomes obvious in 100 dimensions.

Covariance Shift for Normal Data

Sometimes it is useful to monitor both the mean and variability of a process, and to alert if either of these characteristics indicates a special cause (Sullivan and Woodall, 2000; Hawkins and Deng, 2009; Zamba and Hawkins, 2009). The MCUSUM, MEWMA, and T^2 control charts are designed primarily to detect mean shifts. Although when the subgroup size is one, the T^2 chart can be used to detect variance changes. Still, they are usually supplemented with additional monitoring statistics to detect variance changes. However, RTC is able to detect a combination of mean and covariance shifts, and even a decrease in variance. We illustrate with a simple experiment. We generate 10-dimensional, standard normally distributed data. Here 1000 reference observations are used. To generate the signal, 5 variables are mean shifted to 1, and the covariance matrix changes from $5I$ to I, where I is the identity matrix. For RTC, the ARL_1 for the shift under $ARL_0 \approx 200$ is shown in Table 2. However, as expected, for the MEWMA its $ARL_1 > 200$ when $ARL_0 \approx 200$ in this case.

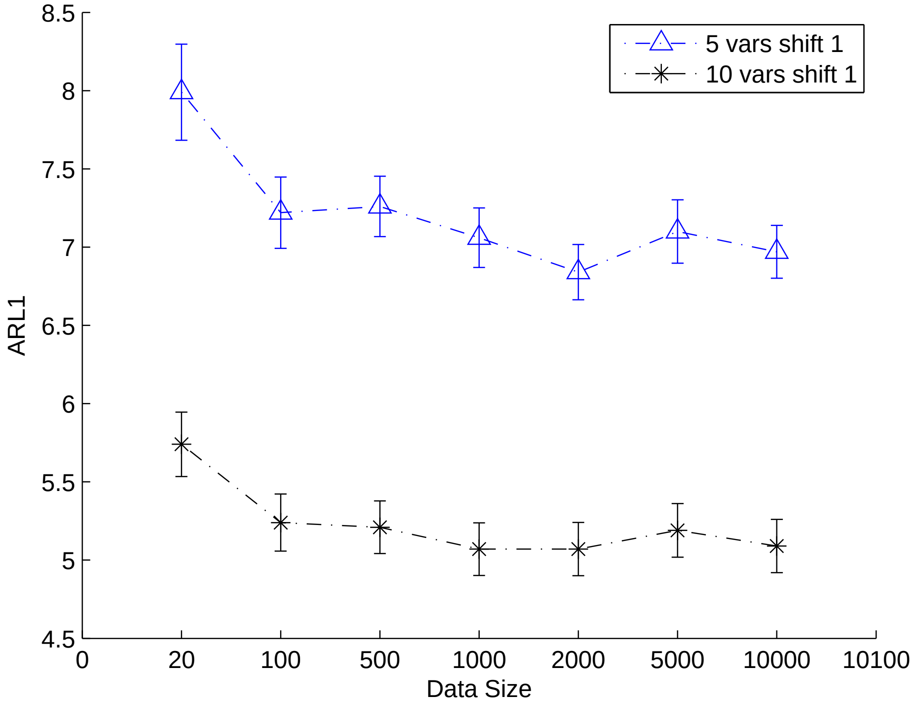


Figure 20: RTC using $p(S_0, t)$ with $N_w = 10$ sensitivity to the reference data set size N_0. ARL_1s and standard errors are shown.

Furthermore, RTC is sensitive to pure covariance changes. Two bivariate normal data sets with variance increases and variance decreases, respectively, are simulated to illustrate the effectiveness of RTC for dealing with covariance changes. The reference data for both cases have zero mean vectors. For the data set with increasing variance, the covariance matrix changes from an identity covariance matrix to 10 times the identity matrix. For the data set with decreasing variance, the covariance matrix changes from a covariance matrix with diagonal elements equal to 10 to an identity covariance matrix. For both cases, we simulated 500 reference observations followed by 500 shifted observations. We used RTC with $p(S_0, t)$ as the monitoring statistic and $N_w = 10$. The results are shown in Figures 21 and 22. In the figures, the y axis is $p(S_0, t)$, and the x axis is the observation time index. The figures illustrate large changes in the monitored statistic corresponding to the variance changes.

Non-Normal Data

First we consider a nonlinear relationship between variables in the reference data. Two-dimensional reference data are generated with the relationship $X_2 = X_1^2 + \epsilon$, where ϵ is standard normally distributed. Here X_1 is uniformly distributed between $[-2, 2]$ and 1000 reference observations are used. The shift changes the joint distribution to be normally distributed as shown in Figure 23. For this shift, the MEWMA has $ARL_1 > 200$ when

Table 1: RTC method with selected statistics and $N_w(t) = 10$ compared to competitors for 10- and 100-dimensional normal data. Control limits (CL) and ARLs are shown.

Dimension	10 Dimensions				100 Dimensions			
	CL	ARL0	$\delta = \sqrt{5}$	$\delta = \sqrt{10}$	CL	ARL0	$\delta = \sqrt{5}$	$\delta = \sqrt{10}$
$l(t)$								
Average	0.84	200.35	9.79	6.49	0.20	200.15	34.80	13.92
Std error		6.09	0.14	0.09		6.50	0.96	0.25
$l_E(t)$								
Average	-0.09	202.46	7.70	5.34	-0.27	201.06	15.99	9.58
Std error		6.35	0.09	0.05		6.25	0.25	0.11
$a(S_0, t)$								
Average	0.88	200.49	7.22	5.24	0.99	248.35	51.07	10.01
Std error		6.50	0.08	0.05		7.72	1.51	0.13
$p(S_w, t)$								
Average	0.48	200.87	7.84	6.18	0.44	203.03	13.17	8.69
Std error		6.10	0.06	0.05		6.32	0.22	0.07
$glr(t)$								
Average	-0.83	200.24	7.97	6.37	-2.55	203.54	13.32	8.74
Std error		6.04	0.06	0.05		6.3077	0.2188	0.0691
$p(S_0, t)$								
Average	0.66	202.16	6.74	5.37	0.63	208.84	10.72	7.44
Std error		6.03	0.06	0.05		6.59	0.14	0.06
MEWMA								
Average	24.19	200.52	4.89	3.27	138.42	209.88	12.84	6.39
std error		6.28	0.05	0.03		6.32	0.20	0.05
AC(EWMA)								
Average	-2.66	200.09	9.67	4.64	-2.34	202.08	94.35	35.50
Std error		6.13	0.21	0.07		6.73	3.66	1.38
T^2	25.19	200.00	14.46	4.33	140.17	200.00	85.87	30.41

Table 2: RTC using $p(S_0, t)$ with $N_w = 10$ for a combination of mean and variance shift. ARL_0, ARL_1, and standard errors are shown.

	Control Limit	ARL0	ARL1
Average	0.658	201.11	9.05
Std error		20.94	0.25

$ARL_0 \approx 200$. However, RTC detects the shift efficiently (Table 3).

To illustrate the method can be effective for mixed numerical and categorical data, we applied the method to credit data (Hettich and Bay., 1999). This data set contains 20 variables, with 7 numerical and 13 categorical. The class labels of the data are ”good” or “bad” credit risk. Here 400 randomly selected “good” cases are considered as the reference data, and 300 “good” cases followed by 300 “bad” cases are used as the real-time data stream. Although not actually time-ordered, the data are ordered for the purpose of an

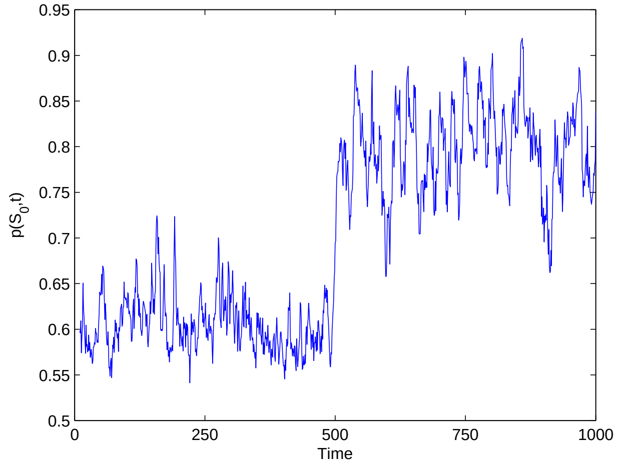


Figure 21: Plot of RTC using $p(S_0, t)$ with $N_w = 10$ versus time for a variance increase at the 501th point. The change after index 500 is clear.

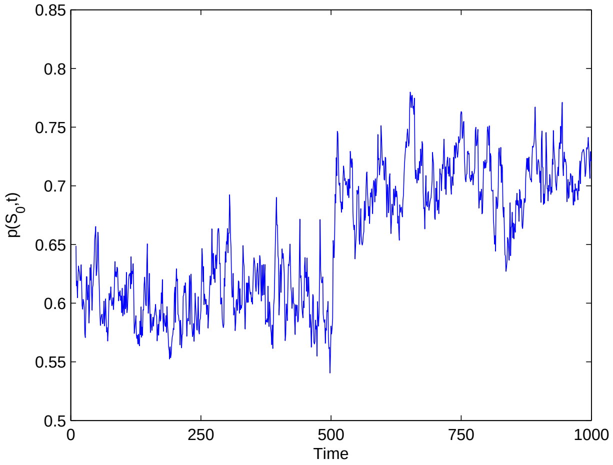


Figure 22: Plot of RTC using $p(S_0, t)$ with $N_w = 10$ versus time for a variance decrease at the 501th point. The change after index 500 is clear.

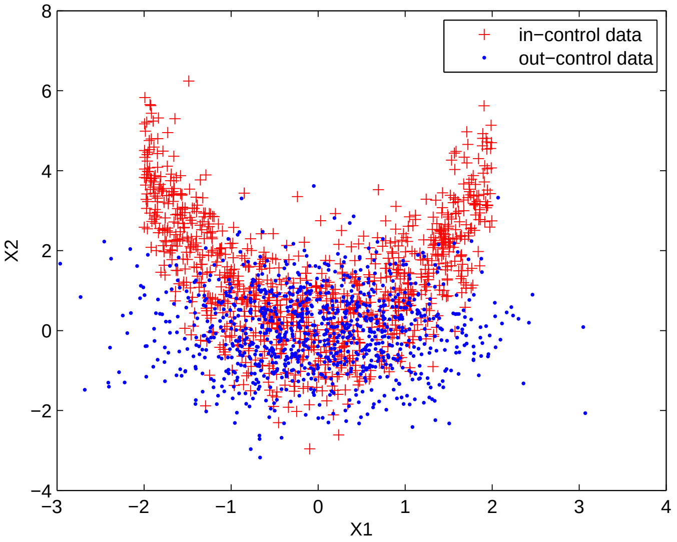


Figure 23: Data with the nonlinear relationship $X_2 = X_1^2 + \epsilon$ shifted to normally distributed data.

Table 3: RTC using $p(S_0, t)$ with $N_w = 10$ for the change from the nonlinear relationship. ARL_0, ARL_1 and standard errors are shown.

	Control Limit	ARL0	ARL1
Average	0.786	201.82	15.56
Std error		17.38	0.86

experiment. We use $p(S_0, t)$ as the monitoring statistic and three window sizes are used: $N_w = \{5, 10, 15\}$. Figures 24-26 plot the monitoring statistic with different window sizes, respectively. In all cases, the change in the monitoring statistic is noticeable and $N_w = 10$ still provides suitable results.

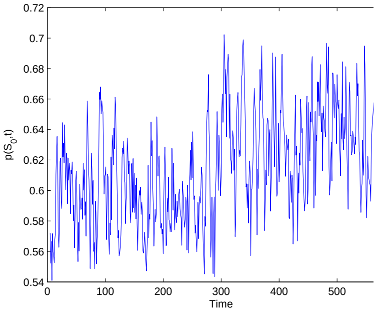


Figure 24: Plot of RTC with $p(S_0, t)$ and $N_w = 5$ versus time for the credit data. The shift occurs at the 301^{th} point.

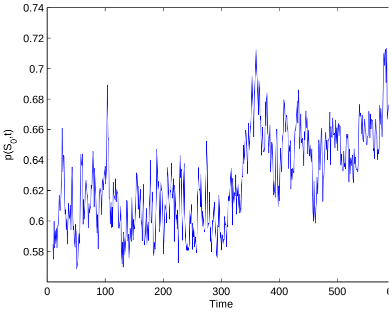


Figure 25: Plot of RTC with $p(S_0, t)$ and $N_w = 10$ versus time for the credit data. The shift occurs at the 301^{th} point.

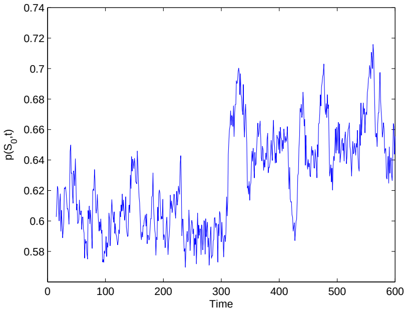


Figure 26: Plot of RTC with $p(S_0, t)$ and $N_w = 15$ versus time for the credit data. The shift occurs at the 301^{th} point.

Conclusions

A monitoring method based on real-time contrasts (RTC) is proposed. A classifier is retrained at each new observation to contrast the real-time data with the reference data. The objective is to detect a difference between the two data sets. This is notably different from previous methods (such as artificial contrasts) that train a classifier one time. With the capabilities of modern computers, such a computationally intensive approach is now feasible. The method can handle combinations of numerical and categorical data, in high dimensions, and with complex reference distributions. Our method does not require any fault conditions to be pre-specified, can detect unusual changes (such as a variance decrease), and can diagnose variables that contribute to a signal. Guided by generalized likelihood ratio principles, several monitoring statistics are proposed and discussed. Performance comparisons are provided. In our experiments, RTC performs well when compared to artificial contrasts or traditional methods.

Appendix: Notation Table

Table 4: Notation Table

Notation	Meaning
p	dimension of the observation vectors
$\mathbf{x}_i$	p-dimensional observation vector at time i
$f_0(\mathbf{x})$	reference data distribution
$f_1(\mathbf{x})$	shifted distribution
S_0	reference data
$S_1(t)$	real time data stream
S_0^k	the kth replicate of reference data
$S_w(t)$	window data at time t
$y(\mathbf{x}_i)$	class of observation $\mathbf{x}_i$
N_0	size of reference data
N_w	size of window data
k_w	resampling sample size from window data
k_0	resampling sample size from reference data
l_t	a generalized likelihood ratio (GLR) statistic at time t
L_t	log of l_t
$\hat{y}(\mathbf{x}_i)$	predicted class of $\mathbf{x}_i$ from a classifier
$\hat{y}(\mathbf{x}_i, T_j)$	predicted class of $\mathbf{x}_i$ based only on tree T_j
$\hat{y}(\mathbf{x}_i\|t)$	predicted class of $\mathbf{x}_i$ at time t
$\hat{p}_k(\mathbf{x}_i)$	class k probability estimate for $\mathbf{x}_i$ from a classifier
$\hat{p}_k(\mathbf{x}_i\|t)$	class k probability estimate for $\mathbf{x}_i$ at time t
$err(S_0, t)$	error rate from reference data at time t
$err(S_w, t)$	error rate from window data at time t
$a(S_0, t)$	accuracy from reference data at time t
$a(S_w, t)$	accuracy from window data at time t
$p(S_0, t)$	monitoring statistic based on class probability estimates from the reference data at time t
$p(S_w, t)$	monitoring statistic based on class probability estimates from the window data at time t
$l(t)$	monitoring statistic based on the most recent observation
$l_E(t)$	EWMA applied to statistic $l(t)$
$glr(t)$	monitoring statistic based on GLR
UCL	upper control limit
M_t	a generic monitoring statistic
M_t^k	a generic monitoring statistic at replicate k
RL_k	run length until $M_t^k > UCL$
RF	random forest classifier
$nTree$	number of trees in a random forest
v	tree node
v_L	left child node of tree node v
v_R	right child node of tree node v
w_L	proportion of observations assigned to left child node of tree node v
w_R	proportion of observations assigned to right child node of tree node v
$I(v)$	impurity score at node v of a tree
$\Delta I(X_j, v)$	decrease in impurity from a split on X_j at tree node v
$VI(X_j, T)$	variable importance score of variable X_j from tree T
m	number of candidate variables considered for a split at a node in a random forest
OOB_i	set of trees in a random forest where observation x_i is out of bag
$\|OOB_i\|$	cardinality of the set OOB_i
$VI(X_j)$	variable importance score of variable X_j from all trees in a random forest

References

Apley, D. and Shi, J. (1999). “The GLRT for Statistical Process Control of Autocorrelated Processes”. *IIE Transactions*, 31(12), pp. 1123–1134.

Apley, D. and Shi, J. (2001). “A Factor-Analysis Method for Diagnosing Variability in Mulitvariate Manufacturing Processes”. *Technometrics*, 43(1), pp. 84–95.

Borror, C.; Montgomery, D.; and Runger, G. (1999). “Robustness of The EWMA Control Chart to Non-Normality”. *Journal of Quality Technology*, 31(3), pp. 309–316.

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, MA.

Breiman, L. (2001). “Random Forests”. *Machine Learning*, 45(1), pp. 5–32.

Byon, E.; Shrivastava, A.; and Ding, Y. (2010). “A Classification Procedure for Highly Imbalanced Class Sizes”. *IIE Transactions*, 42(4), pp. 288–303.

Camci, F.; Chinnam, R.; and Ellis, R. (2008). “Robust Kernel Distance Multivariate Control Chart Using Support Vector Principles”. *International Journal of Production Research*, 46(18), pp. 5075–5095.

Chinnam, R. (2002). “Support Vector Machines for Recognizing Shifts in Correlated and Other Manufacturing Processes”. *International Journal of Production Research*, 40(17), pp. 4449–4466.

Chou, Y.; Polansky, A.; and Mason, R. (1998). “Transforming Non-Normal Data to Normality in Statistical Process Control”. *Journal of Quality Technology*, 30(2), pp. 133–141.

Cook, D. and Chiu, C. (1998). “Using Radial Basis Function Neural Networks to Recognize Shifts in Correlated Manufacturing Process Parameters”. *IIE transactions*, 30(3), pp. 227–234.

Ding, Y.; Zeng, L.; and Zhou, S. (2006). “Phase I Analysis for Monitoring Nonlinear Profiles in Manufacturing Processes”. *Journal of Quality Technology*, 38(3), pp. 199–216.

Hawkins, D. and Deng, Q. (2009). “Combined Charts for Mean and Variance Information”. *Journal of Quality Technology*, 41(4), pp. 415–425.

Hettich, S. and Bay., S. (1999). “The UCI KDD Archive [http://kdd.ics.uci.edu]”.

Hotelling, H. (1947). “Multivariate Quality Control-Illustrated by The Air Testing of Sample Bombsights”. In Eisenhart, C.; Hastay, M.; and Wallis, W., editors, *Techniques of Statistical Analysis*, pages 111–184. McGraw-Hill, New York.

Hu, J. and Runger, G. C. (2010). “Time-Based Detection of Changes to Multivariate Patterns”. *Annals of Operations Research*, 174(1), pp. 67–81.

Hu, J.; Runger, G. C.; and Tuv, E. (2007). “Tuned Artificial Contrasts to Detect Signals”. *International Journal of Production Research*, 45(23), pp. 5527–5534.

Hwang, W.; Runger, G. C.; and Tuv, E. (2007). “Multivariate Statistical Process Control with Artificial Contrasts”. *IIE Transactions*, 39, pp. 659–669.

Imtiaz, S. and Shah, S. (2008). “Treatment of Missing Values in Process Data Analysis”. *The Canadian Journal of Chemical Engineering*, 86(5), pp. 838–858.

Kim, S.; Sukchotrat, T.; and Park, S. (2011). “A Nonparametric Fault Isolation Approach through One-Class Classification Algorithms”. *IIE Transactions*, 43(7), pp. 505–517.

Li, F.; Runger, G. C.; and Tuv, E. (2006). “Supervised Learning for Change-Point Detection”. *International Journal of Production Research*, 44(14), pp. 2853–2868.

Li, J.; Jin, J.; and Shi, J. (2008). “Causation-based T^2 Decomposition for Multivariate Process Monitoring and Diagnosis”. *Journal of Quality Technology*, 40(1), pp. 46–58.

Lowry, C. A.; Woodall, W. H.; Champ, C. W.; and Rigdon, S. E. (1992). “A Multivariate Exponentially Weighted Moving Average Chart”. *Technometrics*, 34, pp. 46–53.

Morrison, D. (1967). *Multivariate Statistical Methods.* McGraw-Hill, New York.

Perlich, C.; Provost, F.; and Simonoff, J. S. (2003). “Tree Induction vs. Logistic Regression: A Learning-Curve Analysis”. *Journal of Machine Learning Research*, 4, pp. 211–255.

Prabhu, A.; Edgar, T.; and Good, R. (2009). “Missing Data Estimation for Run-To-Run Ewma-Controlled Processes”. *Computers & Chemical Engineering*, 33(11), pp. 1861–1869.

Qiu, P. (2008). “Distribution-Free Multivariate Process Control Based on Log-Linear Modeling”. *IIE Transactions*, 40(7), pp. 664–677.

R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2009. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Runger, G.; Alt, F.; and Montgomery, D. (1996). “Contributors to A Multivariate Statistical Process Control Chart Signal”. *Communications in Statistics–Theory and Methods*, 25(10), pp. 2203–2213.

Runger, G. C. (1996). “Projections and The U^2 Multivariate Control Chart”. *Journal of Quality Technology*, 28, pp. 313–319.

Runger, G. C. (2002). “Assignable Causes and Autocorrelation: Control Charts on Observations or Residuals?”. *Journal of Quality Technology*, 34(2), pp. 165–170.

Runger, G. C. and Testik, M. C. (2004). “Multivariate Extensions to Cumulative Sum Control Charts”. *Quality and Reliability Engineering International*, 20(4), pp. 387–396.

Stoumbos, Z. and Sullivan, J. (2002). "Robustness to Non-Normality of the Multivariate EWMA Control Chart". *Journal of Quality Technology*, 34(3), pp. 260–276.

Sukchotrat, T.; Kim, S.; and Tsung, F. (2010). "One-Class Classification-Based Control Charts for Multivariate Process Monitoring". *IIE Transactions*, 42(2), pp. 107–120.

Sullivan, J. and Woodall, W. (2000). "Change-Point Detection of Mean Vector or Covariance Matrix Shifts Using Multivariate Individual Observations". *IIE Transactions*, 32(6), pp. 537–549.

Sun, R. and Tsung, F. (2003). "A Kernel-Distance-Based Multivariate Control Chart Using Support Vector Methods". *International Journal of Production Research*, 41(13), pp. 2975–2989.

Wang, K. and Tsung, F. (2007). "Run-to-Run Process Adjustment Using Categorical Observations". *Journal of Quality Technology*, 39(4), pp. 312–325.

Wang, K. and Jiang, W. (2009). "High-Dimension Process Monitoring and Fault Isolation via Variable Selection". *Journal of Quality Technology*, 41(3), pp. 247–258.

Zamba, K. D. and Hawkins, D. M. (2009). "A Multivariate Change-Point Model for Change in Mean Vector and/or Covariance Structure". *Journal of Quality Technology*, 41(3), pp. 285–303.

Zou, C. and Tsung, F. (2010). "Likelihood Ratio-Based Distribution-Free EWMA Control Charts". *Journal of quality technology*, 42(2), pp. 174–196.

Zou, C.; Tsung, F.; and Wang, Z. (2008). "Monitoring Profiles Based on Nonparametric Regression Methods". *Technometrics*, 50(4), pp. 512–526.