Expert Systems With Applications 71 (2017) 358–369

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Real-time contrasts control chart using random forests with weighted voting

Seongwon Jang[1], Seung Hwan Park[1], Jun-Geol Baek[*]

School of Industrial Management Engineering, Korea University, Anam-dong, Seongbuk-gu 136-701, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 1 August 2016
Revised 18 November 2016
Accepted 2 December 2016
Available online 3 December 2016

Keywords:
Real-time contrasts (RTC)
Fault detection
Fault isolation
Random forests
Weighted voting
Class imbalance

ABSTRACT

Real-time fault detection and isolation are important tasks in process monitoring. A real-time contrasts (RTC) control chart converts the process monitoring problem into a real-time classification problem and outperforms existing methods. However, the monitoring statistics of the original RTC chart are discrete; this could make the fault detection ability less efficient. To make monitoring statistics continuous, distance-based RTC control charts using support vector machines (SVM) and kernel linear discriminant analysis (KLDA) were proposed. Although the distance-based RTC charts outperformed the original RTC chart, the distance-based RTC charts have a disadvantage in that it is difficult to analyze the causes of faults when using these charts. Therefore, we propose improved RTC control charts using random forests with weighted voting. These improved RTC control charts can detect changes more rapidly by making monitoring statistics continuous; additionally, they can also analyze the causes of faults in a similar manner to the original RTC chart. Further, the improved RTC control charts alleviate the class imbalance problem by using F-measure, G-mean, and Matthews correlation coefficient (MCC) as performance measures to assign proper weights to individual classifiers. Experiments show that the proposed methods outperform the original RTC chart and are more effective than the distance-based RTC charts using SVM and KLDA.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Because of technological advances in data collection, the use of multivariate statistical process control (MSPC) procedures is increasing (Woodall & Montgomery, 2014). Multivariate control charts, such as Hotelling's T^2 chart (Hotelling, 1947), the multivariate cumulative sum (MCUSUM) charts (Crosier, 1988), and the multivariate exponentially weighted moving average (MEWMA) charts (Lowry, Woodall, Champ, & Rigdon, 1992), are typically used to detect a process shift when predictor variables are correlated. The calculations for the monitoring statistics and control limit (CL) of these methods require the assumption that the process follows a multivariate normal distribution. In practice, however, the normality assumption can be easily violated in many real-world applications, and this degrades the performance of control charts.

To overcome this problem, several recent studies have applied machine learning algorithms to MSPC procedures. One-class classification (OCC) algorithms were utilized to build a classification boundary (Sukchotrat, Kim, & Tsung, 2010; Sun & Tsung, 2003). The OCC algorithms generate the classification boundary using only phase-I observations. In phase-II, each newly arriving observation is classified as on-target when it is included in the classification boundary; otherwise, it is regarded as off-target. Although the OCC control charts perform better than traditional MSPC methods, phase-II observations are ignored when the classification boundary of the OCC control charts is constructed. As phase-II observations usually have more recent information about current process conditions than phase-I observations, the performance of the OCC control charts could be less sensitive (Deng, Runger, & Tuv, 2012).

On the other hand, binary-class classification algorithms could also be considered for MSPC procedures. Several methods using artificial contrasts were proposed and have been improved (Hu, Runger, & Tuv, 2007; Hwang & Lee, 2015; Hwang, Runger, & Tuv, 2007). In these studies, random artificial data were generated to represent the off-target data against the in-control condition. By assigning an on-target class label to the phase-I data and an off-target class label to the artificial data, a classifier was trained to construct a decision boundary. In another approach, several meth-

[*] Corresponding author. Fax: +82-2-929-5888.
E-mail addresses: jsw900806@korea.ac.kr (S. Jang), udongpang@korea.ac.kr (S.H. Park), jungeol@korea.ac.kr (J.-G. Baek).

[1] Both authors contributed equally to this work.

http://dx.doi.org/10.1016/j.eswa.2016.12.002
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

ods making use of both past in-control and out-of-control information were proposed (Chongfuangprinya, Kim, Park, & Sukchotrat, 2011; Sukchotrat, Kim, Tsui, & Chen, 2011; Zhang, Tsung, & Zou, 2015). The monitoring statistics and CLs of these methods were calculated by the probability that an observation is classified as in-control or out-of-control. They used the k-nearest neighbor algorithm, linear discriminant analysis, and support vector machines as classifiers.

It is important to note that previously explained methods still exclude the real-time phase-II data for their CL. In addition, as the classifiers in the methods are trained only once ahead of the start of phase-II monitoring, their classification boundary is fixed after construction. If the off-target condition is not properly represented in the artificial contrasts or the past out-of-control data, the classifier may lose its fault detection ability because it creates a biased decision boundary. To resolve this problem, a real-time contrasts (RTC) control chart was developed (Deng et al., 2012). The real-time phase-II data within a window are regarded as contrasts, and random forests build a decision boundary using both phase-I reference data and real-time observations. This method constantly updates the classification boundary whenever new real-time measurements are available. In other words, the RTC control charts convert the process monitoring problem into a real-time classification problem. A simulation study showed that the RTC chart performs better than the traditional control chart, artificial contrasts method, and change-point detection method based on generalized likelihood ratio statistics. Furthermore, the RTC chart has advantages in that it not only can be applied to various data types, such as categorical data and missing data, but can also analyze the causes of faults.

However, the original RTC chart faces two notable problems. First, the monitoring statistics of the original RTC chart have discrete values because they are calculated using classification accuracy or the probability of correct classification by random forests, which are an ensemble of decision trees. It is possible that observations with different degrees of abnormality may have equivalent monitoring statistic values, and this could make the fault detection ability less sensitive. To make monitoring statistics continuous, distance-based RTC control charts using support vector machines (SVMs) (He, Jiang, & Deng, 2016) and kernel linear discriminant analysis (KLDA) were proposed (Wei, Huang, Jiang, & Zhao, 2016). Contrary to the original RTC chart, they used the distance from the decision boundary as the monitoring statistic, and this enabled the distance-based RTC charts to outperform the original RTC chart. However, the distance-based RTC charts do not have the merits of the original RTC chart. They cannot be applied to various data types and analyze the causes of faults. To make monitoring statistics continuous, weighted voting can be used. Several studies were carried out in the area of weighted voting based on random forests (Guenter & Bunke, 2004; Robnik-Sikonja, 2004; Tsymbal, Pechenizkiy, & Cunningham, 2006) and compared with each other (Tripoliti, Fotiadis, & Manis, 2013). However, they did not consider the class imbalance problem. Although weighted voting methods considering class imbalance were proposed (Bhowan, Johnston, Zhang, & Yao, 2013; Chawla & Sylvester, 2007), they used a genetic algorithm, which is not suitable for RTC charts owing to its computation time.

The original RTC method still suffers from the class imbalance problem. Deng et al. (2012) utilized a stratified sampling method to address the class imbalance problem, but there is room for improvement. Most prominently, balanced random forests (BRF) and weighted random forests (WRF), both based on random forests, were proposed to deal with class imbalance (Chen, Liaw, & Breiman, 2004). BRF is identical to the stratified sampling method except that it changes vote cutoffs for the final prediction; WRF assigns further weights to the minority class, thereby more heavily punishing misclassification of the minority class. However, these methods need additional parameters, which are the cutoff and weight. These hyperparameters have to be tuned experimentally to improve the performance; this results in a computational burden for the RTC methods.

The following are the main contributions of this paper. We propose improved RTC control charts using random forests with weighted voting. These improved RTC control charts not only detect a shift more rapidly by making monitoring statistics continuous but can also analyze the origins of faults. Moreover, the improved RTC control charts relieve the class imbalance problem. We compare the proposed method to the existing RTC methods in different experiments.

The remainder of the paper is organized as follows. In Section 2, we give an overview of the conventional RTC control chart. Section 3 introduces the proposed methods, and the procedure for fault isolation is discussed. In Section 4, experimental results for the proposed methods are presented, including a performance comparison with the existing RTC charts using simulated data. Finally, Section 5 contains our concluding remarks and suggestions on further research opportunities.

2. Real-time contrasts (RTC) control chart

In this section, we describe the original RTC control chart in which random forests are used to calculate the monitoring statistics. We introduce the RTC method first, followed by random forests.

2.1. Real-time contrasts (RTC) method

RTC control charts convert the process monitoring problem into a real-time classification problem. In the RTC method (Deng et al., 2012), the phase-I data S_0, which are from the in-control condition, are referred to as reference data. Multivariate measurements are acquired from the process at each time t and denoted by x_t. The measurements in a moving window with window size N_w are denoted by $S_w(t)$. $S_w(t)$ contains the most recent N_w measurements and is updated whenever a new measurement arrives, i.e., $S_w(t) = \{x_{t-N_w+1}, \ \cdots, \ x_{t-1}, \ x_t\}$. In addition, $S_w(t)$ is used as the RTC against the reference data. For supervised learning, the reference data S_0 and RTC data $S_w(t)$ are labeled with two classes, Class 0 and Class 1, respectively.

The RTC method builds a classification boundary between S_0 and constantly arriving $S_w(t)$. The classification accuracies and probabilities of correct classification contain information about the process condition. When there is no shift in the real-time process, it is hard to distinguish one class from the other; the accuracy and probability of correct classification will be low. On the contrary, when the real-time process is out of control, the accuracy and probability of correct classification will be high. Although both the accuracy and classification probability can be used as the monitoring statistics for the MSPC procedures, the probability of correct classification detects the shift more rapidly than accuracy does in most situations (Deng et al., 2012; Wei et al., 2016).

Let $\hat{p}_k(x_i|t)$ denote the predicted probability that a measurement x_i is classified as k at time t ($k = 0, \ 1$). As both S_0 and $S_w(t)$ include multiple observations, we need to calculate the average of S_0 and $S_w(t)$ to summarize the classification results and utilize them as the monitoring statistics. For $x_i \in S_0$,

$$p(S_0, t) = \frac{\sum_{x_i \in S_0} \hat{p}_0(x_i|t)}{N_0} \tag{1}$$

and for $x_i \in S_w(t)$,

$$p(S_w, t) = \frac{\sum_{x_i \in S_w(t)} \hat{p}_1(x_i|t)}{N_w} \tag{2}$$

Download English Version:

https://daneshyari.com/en/article/4943547

Download Persian Version:

https://daneshyari.com/article/4943547

Daneshyari.com