On Optimum Thresholding of Multivariate Change Detectors

William J. Faithfull and Ludmila I. Kuncheva

School of Computer Science, Bangor University, Bangor,
Gwynedd, Wales, United Kingdom
{w.faithfull,l.i.kuncheva}@bangor.ac.uk
pages.bangor.ac.uk/~eese11

Abstract. A change detection algorithm for multi-dimensional data reduces the input space to a single statistic and compares it with a threshold to signal change. This study investigates the performance of two methods for estimating such a threshold: bootstrapping and control charts. The methods are tested on a challenging dataset of emotional facial expressions, recorded in real-time using Kinect for Windows. Our results favoured the control chart threshold and suggested a possible benefit from using multiple detectors.

1 Introduction

Detecting a change point in a sequence of observations is a well researched statistical problem with applications in areas such as Economics [1], Data Stream Mining [2] and Quality Control [3]. The basic premise of change point detection is that, given a sequence of observations $x_1, x_2, ..., x_n$, there exists a change point t such that $x_1, x_2, ..., x_t$ was generated exclusively by some process P_0 and $x_t, x_{t+1}, ..., x_n$ was generated exclusively by some other process P_1.

Detecting change points in multivariate data is a challenging problem. A variety of multivariate change detectors have been proposed [4–6], some of which amount to a novel combination of univariate detectors, while others take a dimensionality reduction approach. In the latter case, the multidimensional data is reduced to a single statistic which should ideally correlate with the appearance of change. One of the main issues with such detectors is identifying a threshold on the single statistic for flagging a change. Here we examine the suitability of two approaches to setting a threshold: bootstrapping and control charts. Figure 1 illustrates the multivariate change detection process.

2 Related Work

Change detection has been an active area of research for more than 60 years, developing out of methods for statistical quality control. Being well researched and statistically grounded, Control Charts are the basis for many methods such as

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 364–373, 2014.

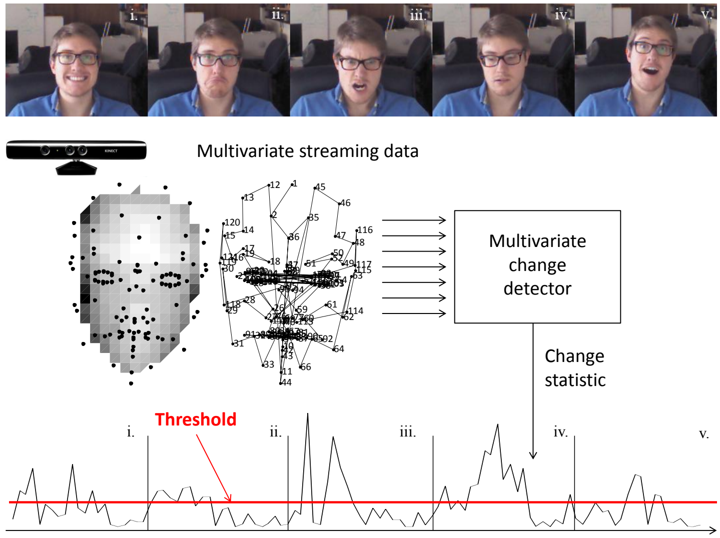

Fig. 1. Illustration of the process of change detection in streaming multidimensional data and the role of the threshold. The data was obtained from Kinect while a participant was acting a sequence of emotional states: *i*. Happiness, *ii*. Sadness, *iii*. Anger, *iv*. Indifference, *v*. Surprise.

CUSUM (Cumulative Sum) charts and EWMA (Exponentially Weighted Moving Average) charts. Some of the earliest work in the field is that of Shewhart [3, 7] and his development of the control chart for sequential process control, now widely adopted by industry. The field is now very broad, with a number of reference monographs including Wald [8], Basseville and Nikiforov [9] and Brodsky and Darkhovsky [10] although largely focussed on univariate data.

There are differing approaches to the problem of detecting change in multivariate data. Lowry and Montgomery [11] reviewed multivariate control charts for quality control. Consider n p-dimensional vectors of observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. It is possible to simply create p individual charts, one for each feature, not reducing the dimensionality of the data. However, this approach does not account for correlation between the features. Even truly multivariate control chart approaches such as the Hotelling Control Chart [12] can be equated to dimensionality reduction and thresholding, as it reduces the p dimensions of the data to a single T^2 statistic. The list below demonstrates the inconsistency of approaches to setting such a threshold.

Work:	Decision method
Zamba & Hawkins [13]:	γ set according to a desired false alarm rate.
Song et al. [14]:	Original statistical test.
Dasu et al. [5]:	Monte Carlo Bootstrapping.
Kuncheva [15]:	Signficance of log-likelihood ratio.

The scope of this work is concerned with establishing a method for threshold setting that is applicable to multiple approaches to change detection.

3 Multivariate Change Detectors

Here we assume that the change detection criteria are calculated from prespecified windows of data W_1 and W_2. Change is sought between the distributions in the two windows.

3.1 Parametric Detectors: Hotelling

The two windows of data contain points $\mathbf{x} = [x_1, \ldots, x_p]^T \in \Re^p$. Hotelling [16] proposes a statistical test for equivalence of the means of the two distributions from which W_1 and W_2 are sampled. The null hypothesis is that W_1 and W_2 are drawn independently from two multivariate normal distributions with the same mean and covariance matrices. Denote the sample means by $\hat{\mu}_1$ and $\hat{\mu}_2$, the pooled sample covariance matrix by $\hat{\Sigma}$, and the cardinalities of the two windows by $M_1 = |W_1|$ and $M_2 = |W_2|$. The T^2 statistic is calculated as

$$T^2 = \frac{M_1 M_2 (M_1 + M_2 - p - 1)}{p(M_1 + M_2 - 2)(M_1 + M_2)} \times (\hat{\mu}_1 - \hat{\mu}_2)^T \hat{\Sigma}^{-1} (\hat{\mu}_1 - \hat{\mu}_2) \tag{1}$$

Under the null hypothesis, T^2 has F distribution with degrees of freedom p and $M_1 + M_2 - p + 1$. The T^2 statistic is the Mahalanobis distance between the two sample means multiplied by a constant. The p-value of the statistical test is instantly available and the desired significance level will determine the change threshold.

The obvious problem with the Hotelling test is that it is only meant to detect changes in the position of the means. Thus it will not be able to indicate change of variance or a linear transformation of the data that does not affect the mean.

3.2 Semi-parametric Detectors: SPLL

The semi-parametric log-likelihood criterion (SPLL) for change detection [6] comes as a special case of a log-likelihood framework, and is modified to ensure computational simplicity. Suppose that the data before the change comes from a Gaussian mixture $p_1(\mathbf{x})$ with c components each with the same covariance matrix. The parameters of the mixture are estimated from the first window of data W_1. The change detection criterion is derived using an upper bound of the log-likelihood of the data in the second window, W_2. The criterion is calculated as

$$SPLL = \max\{SPLL(W_1, W_2), SPLL(W_2, W_1)\}. \tag{2}$$

where

$$SPLL(W_1, W_2) = \frac{1}{M_2} \sum_{\mathbf{x} \in W_2} (\mathbf{x} - \mu_{i*})^T \Sigma^{-1} (\mathbf{x} - \mu_{i*}). \tag{3}$$

where M_2 is the number of objects in W_2, and

$$i* = \arg\min_{i=1}^{c} \left\{ (\mathbf{x} - \mu_i)^T \Sigma^{-1} (\mathbf{x} - \mu_i) \right\} \tag{4}$$

is the index of the component with the smallest squared Mahalanobis distance between $\mathbf{x}$ and its centre.

If the assumptions for p_1 are met, and if W_2 comes from p_1, the squared Mahalanobis distances have a chi-square distribution with p degrees of freedom. The expected value is p and the standard deviation is $\sqrt{2p}$. If W_2 does not come from the same distribution, then the mean of the distances will deviate from p. Subsequently, we swap the two windows and calculate the criterion again, this time $SPLL(W_2, W_1)$. By taking the maximum of the two, SPLL becomes a monotonic statistic.

3.3 Non-parametric Detectors: Kullback-Leibler Distance

In this approach, the data distribution in window W_1 is represented as a collection of K bins (regions in $\Re^p$), with a probability mass value assigned to each bin. Call this empirical distribution $\hat{P}_1$. The data in W_2 is distributed in the bins according to the points' locations, giving empirical distribution $\hat{P}_2$. The criterion function is

$$KL(\hat{P}_2||\hat{P}_1) = \sum_{i=1}^{K} \hat{P}_2(i) \log \left\{ \frac{\hat{P}_2(i)}{\hat{P}_1(i)} \right\} \tag{5}$$

where i is the bin number, and $\hat{P}(i)$ is the estimated probability in bin i.

If the two distributions are identical, the value of $KL(P_2||P_1)$ is zero. The larger the value, the higher the likelihood that P_2 is different from P_1. Note that we have only approximations of P_1 and P_2. The usefulness of the KL criterion depends on the quality of the approximations and on finding a threshold λ such that change is declared if $KL > \lambda$.

In Dasu et al.'s change detector [5], W_1 is expanded until change is detected, giving a good basis for approximating P_1. On the other hand, P_2 has to be estimated from a short recent window, hence the estimate may be noisy. Dasu et al. approximate the P_1 probability mass function by building *kdq* trees which can be updated with the streaming data. Other approximations are also possible, including the clustering approach for SPLL.

The KL distance criterion is not related to a straightforward statistical test that will give us a fixed threshold λ, which was one of the motivations behind our study.

4 Threshold Setting Approaches

Hotelling T^2 detector has the advantage of a statistically interpretable threshold. However, it has a serious shortcoming in that it only detects change in the mean of the data. To equip SPLL and KL with a similar type of threshold, here we examine two threshold setting approaches for the change detection statistic.

4.1 Bootstrapping

Let $|W_1| = M_1$. To determine a threshold, a bootstrap sample of M_1 objects is drawn from W_1. A discrete probability distribution $\hat{P}_1$ is approximated from this sample. Subsequently, another sample of the same size is drawn from W_1 and its distribution $\hat{Q}_1$ is evaluated. For example, if $\hat{P}_1$ is a set of bins, $\hat{Q}_1$ is calculated as the proportion of the data from the second bootstrap sample in the respective bins. The match between $\hat{P}_1$ and $\hat{Q}_1$ is estimated using, for example, KL distance (5), which gives the change statistic. Running a large number of such Monte Carlo simulations, a distribution of the change statistic is estimated, corresponding to the null hypothesis that there is no change (all samples were drawn from the same window, W_1). We can take the Kth percentile of this distribution as the desired threshold. This approach was adopted by Dasu et al. [5] where the probability mass functions were approximated by a novel combination of kd-trees and quad trees, called kdq-trees. We direct the reader to [5] for an in-depth definition of kdq-trees. One drawback of this approach is the excessive computation load when a new threshold is needed.

4.2 Control Chart

A less computationally demanding alternative to bootstrapping is a Shewhart individuals control chart to monitor the change statistic. Inspired by this, our hypothesis is that the process underlying an appropriate change statistic will exhibit an out-of-control state when change occurs. Using a window of T observations, we calculate the centre line $\bar{x}$ as the mean of the values of the statistic returned from the change detector, and its standard deviation $\hat{\sigma}$. The upper and lower control limits are calculated as

$$\bar{x} \pm 1.96 \frac{\hat{\sigma}}{\sqrt{T}}. \tag{6}$$

If either of the control limits are exceeded, change is signalled. This (rather naive) threshold estimation assumes that the change statistic has normal distribution, and that we have a sufficiently large window so as to get reliable estimates. The above value is for significance level $\alpha = 0.05$. The bootstrap threshold does not rely on any such assumption but is more cumbersome.

5 Experimental Investigation

All thresholds considered here, including the threshold of the Hotelling method, are meant to control the type I error ("convict the innocent", or accepting that there is a change when there is none). If we set all these thresholds to 0.05, we should expect to have false positive rate less than that. Nothing is guaranteed about the type II error ("free the guilty", or missing a change when there is one). Thus we are interested to find out how the three chosen change detectors behave for the two type of thresholds, in terms of both error types.

5.1 Facial Expression Data

We chose a challenging real-life problem to test the change detectors. Sustained facial expressions of five emotions were taken to be the stable states, and the transition from one emotion to another was the change.

While a number of facial expression databases exist, they require camera equipment and intermediate computer vision techniques to record data. In our approach, we utilise the Face Tracking toolkit distributed with the Kinect SDK to extract data directly from the device. This approach lends itself to analysis of real-time streaming data The advantage of having a minimal setup is that data capture does not have to be intrusive. This presents the opportunity of capturing real-time data about a participant's posture and facial expression whilst they interact with the computer.

The Kinect Face Tracking SDK utilises the Active Appearance Model (AAM) [17], taking into account the data from the depth sensor to allow head and face tracking in 3D. The features we take from the Kinect are as follows:

- Features extracted by the Kinect software
  - ○ Face Points : 123 3D points on the face
  - ○ Skeleton Points : 10 3D points on the joints of the upper body
  - ○ Animation Units: 6 Animation Units $[-1, 1]$

- Six animation units and their equivalents in the Candide3 model

Animation Unit	Candide3 [18]	Description
AU0	AU10	Upper Lip Raiser
AU1	AU26/27	Jaw Lowerer
AU2	AU20	Lip Stretcher
AU3	AU4	Brow Lowerer
AU4	AU13/15	Lip Corner Depressor
AU5	AU2	Outer Brow Raiser

5.2 Data Capture

Each participant sat with their eyes trained on a computer screen, with a Kinect observing them. Emotional transitions are triggered by visual instructions. The participants were asked to hold their facial expression until instructed to change it. The duration of a facial expression is 3 seconds. The timestamps of these instructions are logged to provide the true positive values for the experiment. Thus each experimental run produces about 5 expressions × 3 seconds × 30 FPS = 540 frames. Figure 2 shows an example of one of the animation units throughout one run. The periods of sustained facial expressions are labelled. The initial warm-up period, as well as the transition periods of 7 frames are also indicated.

The process is facilitated by a bespoke application written in the C# language, which utilises the Kinect SDK to retrieve frames from the sensor and extract the features. The application acts as a TCP client which connects to a server

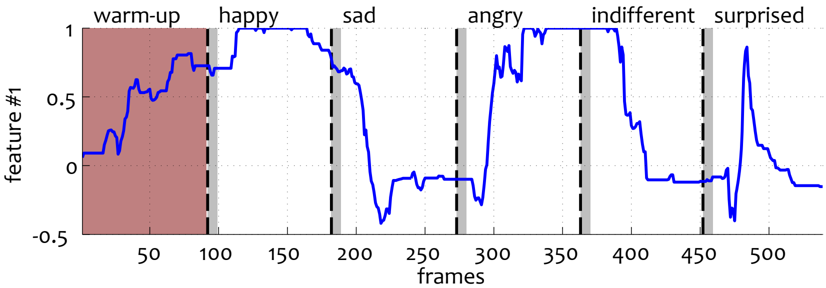

Fig. 2. An example of an animation unit along one experimental run for collecting data. The dashed vertical lines are the time points where the participant is prompted to change their facial expression. The shaded regions are transition stages.

running in MATLAB, where the extracted features and timestamps are streamed in real-time, ready for analysis.

5.3 Experimental Methodology

The experiment was conducted using the Animation Units from six participants, each of whom recorded ten runs using the apparatus. Human reaction time to visual stimuli is 180-200 ms. In a recording at approximately 30 frames per second, a true positive detection should appear no earlier than $180/30 = 6$ frames after the labelled change (prompt to change the facial expression). For each run, we test Hotelling, KL Distance with Bootstrapping, KL Distance with Control Charts, SPLL with Bootstrapping and SPLL with Control Charts. The protocol below was followed for each run and for each participant:

1. Split the data into segments by label.
2. Sample a window W_1 of T contiguous frames from a random segment S, with cardinality $|S| = M$ and random starting frame F, $7 \leq F \leq (M - T)$.
3. Sample W_2 from a random segment. If drawn from the same label as W_1, test for false positives, else test for true positives.
4. Calculate the threshold from W_1 using the chosen method.
5. Calculate change statistic from W_1 and W_2 and compare with the threshold. Store 'change' or 'no change', as well as the time taken to execute the iteration steps.
6. Repeat 1–5 K times sampling W_1 and W_2 from the same label, K times sampling W_1 and W_2 from different random labels. Calculate and return the true positive and false positive rates for the chosen detector and threshold.

Five hundred runs were carried out for determining the bootstrapping threshold.

To simulate a window of running change statistic only from data window W_1, we adopted the following procedure. A sliding split point m was generated, which was varied from 3 to $T - 3$. This point was used to create windows W_1', with data from 1 to m, and W_1'', with data from $m+1$ to T. The statistic of interest was calculated from these sub windows, which were assumed to come from the same distribution.

We used $T = 50$, in order that the window size be above 50% of an expression duration. While there is a great deal of literature on the subject of adaptive windowing [19–21], this is beyond the scope of this paper. Such a technique could be used to set T. We set $K = 30$. The experiment was performed on a Core i7-3770K 4.6GHz Windows machine with 16GB RAM.

5.4 Results

We can examine the relative merit of the detectors and thresholds by plotting them on a Receiving Operating Characteristic (ROC) curve. The x-axis is '1− Specificity' of the test, which is the false positive rate, and the y-axis is the 'Sensitivity' of the test, which is the true positive rate. Each run for each participant can be plotted as a point in this space. An ideal detector will reside in the top left corner (point (0,1)), for which true positive rate is 1 and false positive rate is 0. The closer a point is to this corner, the better the detector is.

Figure 3 shows 30 points (6 participants × 5 detector-threshold combinations).

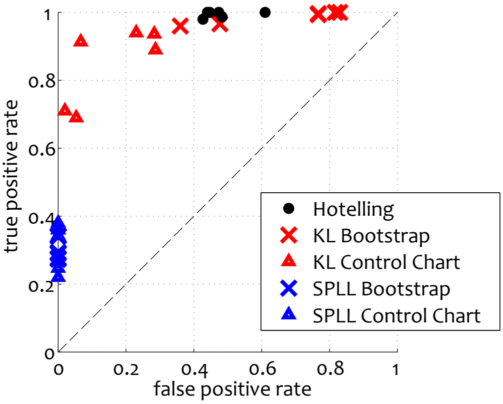

Fig. 3. Results for the 5 detector-threshold combinations. Each point is the average (FP,TP) for one participant, across the $K = 30$ iterations and 10 runs.

Each point corresponds to a participant. The marker and the colour indicate the detector-threshold combination. The figure shows that, although the detectors are not perfect individually, the points collectively form a high-quality ROC curve.

All thresholds were calculated for level of significance 0.05. Applying this threshold is supposed to restrict the false positives to that value. This happened only for the SPLL detector. The price for the zero FP-rate is a low sensitivity, making SPLL the most conservative of three detectors. The Hotelling detector does not live up to the expectation of FP < 0.05. It is not guaranteed to have that FP rate if the assumptions of the test are not met - clearly the situation here. Between this test and KL with bootstrap threshold, Hotelling is both faster and more accurate (lower FP for the same TP). The best combination for our type of data appeared to be the KL detector with the control chart threshold. It exhibits an excellent compromise between FP and TP, and is faster to calculate.

Interestingly, the threshold-setting approach did not affect SPLL but did affect the KL-detector. The control chart approach improved on the original bootstrap approach by reducing dramatically the false positive rate without degrading substantially the true positive rate.

We note that the way we sampled W_1 and $W2$ may have induced some optimistic bias because the samples from the same label could be overlapping. This makes it easier for the detectors to achieve low FP rates than it would be in true streaming data. Nevertheless, this set-up did not favour any of the detectors or threshold-calculating methods, so the comparison is fair.

The execution time analyses favoured unequivocally the control-chart approach to finding a threshold. Also SPLL is the slowest of the detectors, followed by KL and Hotelling. Therefore we recommend the KL-detector with a control-chart threshold.

6 Conclusion

This paper examines the use of control charts as an alternative to the more traditional bootstrap approach for determining a threshold for change detectors. Our experimental study with a real-life dataset of facial expressions taken in real time favoured the KL-detector with a control chart threshold.

We also observed that the statistical significance of the thresholds (type I error) is not matched in the experiments, except for the SPLL detector. The non-parametric bootstrap approach, was expected to give a more robust threshold, not affected by a false assumption about the distribution of the change statistic. The opposite was observed in our experiments for the KL-detector. The reason for this could be that the window was too small to account for the variability of the data sampled from the same label. The results of the experiment led us to recommend the KL-detector with a control chart threshold for difficult streaming data such as facial expressions and behavioural analysis. SPLL with control chart threshold would be preferable where a conservative detector is needed. The same detection accuracy would be achieved with a bootstrap threshold but the extra computational expense is not justified.

Observing the excellent ROC curve shape offered by the collection of detectors, a combination of change detectors with different threshold-setting strategies looks a promising future research avenue. Investigation of adapting methods for classifier fusion to this problem is required, to assess the feasibility of creating a decision ensemble of change detectors.

References

1. Andrews, D.W.: Tests for parameter instability and structural change with unknown change point. Econometrica: Journal of the Econometric Society, 821–856 (1993)
2. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases. VLDB Endowment, vol. 30, pp. 180–191 (2004)
3. Shewhart, W.A.: Economic control of quality of manufactured product (1931)
4. Lowry, C.A., Woodall, W.H., Champ, C.W., Rigdon, S.E.: A multivariate exponentially weighted moving average control chart. Technometrics 34(1), 46–53 (1992)
5. Dasu, T., Krishnan, S., Venkatasubramanian, S., Yi, K.: An information-theoretic approach to detecting changes in multi-dimensional data streams. In: Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, Citeseer (2006)
6. Kuncheva, L.: Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge Data Engineering 25(5), 1175–1180 (2013)
7. Shewhart, W.A.: Quality control charts. Bell System Technical Journal 5, 593–603 (1926)
8. Wald, A.: Sequential analysis. Courier Dover Publications (1947)
9. Basseville, M., Nikiforov, I.V., et al.: Detection of abrupt changes: theory and application, vol. 104. Prentice Hall, Englewood Cliffs (1993)
10. Brodsky, B.E., Darkhovsky, B.S.: Nonparametric methods in change point problems, vol. 243. Kluwer Academic Pub. (1993)
11. Lowry, C.A., Montgomery, D.C.: A review of multivariate control charts. IIE Transactions 27(6), 800–810 (1995)
12. Hotelling, H.: Multivariate quality control. Techniques of statistical analysis (1947)
13. Zamba, K., Hawkins, D.M.: A multivariate change-point model for statistical process control. Technometrics 48(4), 539–549 (2006)
14. Song, X., Wu, M., Jermaine, C., Ranka, S.: Statistical change detection for multidimensional data. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 667–676. ACM (2007)
15. Kuncheva, L.I.: Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge and Data Engineering 25 (2013)
16. Hotelling, H.: The generalization of Student's ratio. Annals of Mathematical Statistics 2(3), 360–378 (1931)
17. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
18. Ahlberg, J.: Candide-3-an updated parameterised face (2001)
19. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: SDM, vol. 7 (2007)
20. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
21. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)