UNSUPERVISED ONLINE CHANGE-POINT DETECTION

{ VICTOR KHAMESI } IMPERIAL COLLEGE LONDON, DEPARTMENT OF MATHEMATICS

Imperial College London

1. INTRODUCTION

• **Aim.** Change-point detection is a well-known unsupervised problem in time series analysis and statistics whose aim is to identify times when the probability distribution of a stochastic process or time series changes.

• **Online.** In fact, it can be split into offline and online change-point detection, depending on the whether the data flow is observed in real time (online) or retrospectively (offline). Online change-point detection imposes additional constraints such as efficient memory usage, without which the algorithm would become increasingly slow, and the lowest possible time complexity, in order to detect changes as soon as possible.

• **Applications.** Change-point detection may focus on multivariate data as well as on univariate data and has a wide range of applications in weather and climate change monitoring, medical monitoring, cybersecurity or finance.

2. PROBLEM DEFINITION

• **Definition.** Consider a stream of observations $x_1, x_2, ...$ where $x_t \in \mathbb{R}^d$ is the observation at time t, sampled from i.i.d. random variables $X_1, X_2, ...$ with change-points $\tau_1, \tau_2, ...$ such that

$$X_1, X_2, ..., X_{\tau_1} \sim F_1,$$
$$X_{\tau_1+1}, X_{\tau_1+2}, ..., X_{\tau_2} \sim F_2,$$
$$X_{\tau_2+1}, X_{\tau_2+2}, ..., X_{\tau_3} \sim F_3, \text{ etc.}$$

where $F_1, F_2, ...$ denote distributions such that $F_k \neq F_{k+1}$ for all k. Change-point detection is the problem of estimating $\tau_1, \tau_2, ...$

• **Performance metrics.** Average Run Lengths are often used metrics in change-point detection problems: ARL0 is the average number of observations until a change-point is detected, when the algorithm is run over a sequence of observations with no change-point; ARL1 is the average number of observations between a change-point occurring and the change being detected.

3. METHODS

In general, detecting a change-point online consists in first **calculating a statistic of interest using observed data** so far, and then **build a decision rule to assert a change-point has occurred** by comparing this statistic to a threshold for example. We summarise below methods of interest in this project.

(1) CUSUM: If the stream is initially $\mathcal{N}(\mu, \sigma^2)$-distributed, one can introduce two statistics S_j and T_j defined as: $S_0 = T_0 = \mu$ and $S_j = \max\left(0, S_{j-1} + \frac{x_j - \mu}{\sigma - k}\right)$ and $T_j = \max\left(0, T_{j-1} - \frac{x_j - \mu}{\sigma - k}\right)$ for $j \in \{1, 2, ...\}$ and detect a change-point when $S_j > h$ or $T_j > h$. Here, k and h are model parameters and are tuned according to the expected size of jumps in the data and μ and σ are updated sequentially.

(2) EWMA: Let $Z_0 = \mu$ and let $Z_j = (1-r)Z_{j-1} + rx_j$ for $j \in \{1, 2, ...\}$. Then, the standard deviation of Z_j is $\sigma_{Z_j} = \sigma\sqrt{\frac{r}{2-r}\left(1-(1-r)^{2j}\right)}$ and a change is detected when $\left|\frac{Z_j - \mu}{\sigma_{Z_j}}\right| > L$. Again, r and L are model parameters that should be wisely chosen and μ and σ are updated sequentially.

(3) Online neural networks: The idea of this classification method is to **compare two mini-batches of observations** $\mathcal{X}(t-l)$ and $\mathcal{X}(t)$ obtained by formatting the initial data. Let $f(X, \theta)$ denote a neural network designed to process these mini-batches and trained using **cross-entropy loss function**, where $\mathcal{X}(t-l)$ are considered as the negative class and $\mathcal{X}(t)$ as the positive class. The neural network f is trained online processing each pair of mini-batch and can in the same time be used to compute a dissimilarity score based on the **Kullback-Leibler divergence** $D(\mathcal{X}(t-l), \mathcal{X}(t), \theta) = \frac{1}{n}\sum_{X \in \mathcal{X}(t-l)} \log \frac{1-f(X,\theta)}{f(X,\theta)} + \frac{1}{n}\sum_{X \in \mathcal{X}(t)} \log \frac{f(X,\theta)}{1-f(X,\theta)}$. This divergence measure is used as a dissimilarity measure to compare sequentially each mini-batch to the lagged mini-batch and can then be used to infer change-points.

4. RESULTS

• **Comparing the methods.** One could first evaluate some well-known start-of-the art change-point detection algorithms on artificially generated data to compare their performance. We run 7 different change-point algorithms algorithms on 10^5 artificially generated sequences sampled from a normal distribution, with a single change-point of size 3 standard deviations.

	ARL0	ARL1	Accuracy
CUSUM	189.74	3.06	0.62
EWMA	201.08	**2.27**	0.70
Mood	241.48	3.24	0.78
Lepage	**243.35**	2.98	**0.79**
Mann-Whitney	240.43	3.94	0.75
K-S	232.61	3.21	0.74
Cramer-von-Mises	240.77	3.90	0.77

Table 1: Performance summary

Results in **Table 1** show that the Lepage non-parametric test is in this case the most efficient method for change-point monitoring. However, one could focus on Run Lengths distribution instead of looking only at the average and try to infer more information, as shown on **Figure 1** where one can see that run lengths 0 distributions are right skewed and the median (dashed) is always less than the mean (solid).

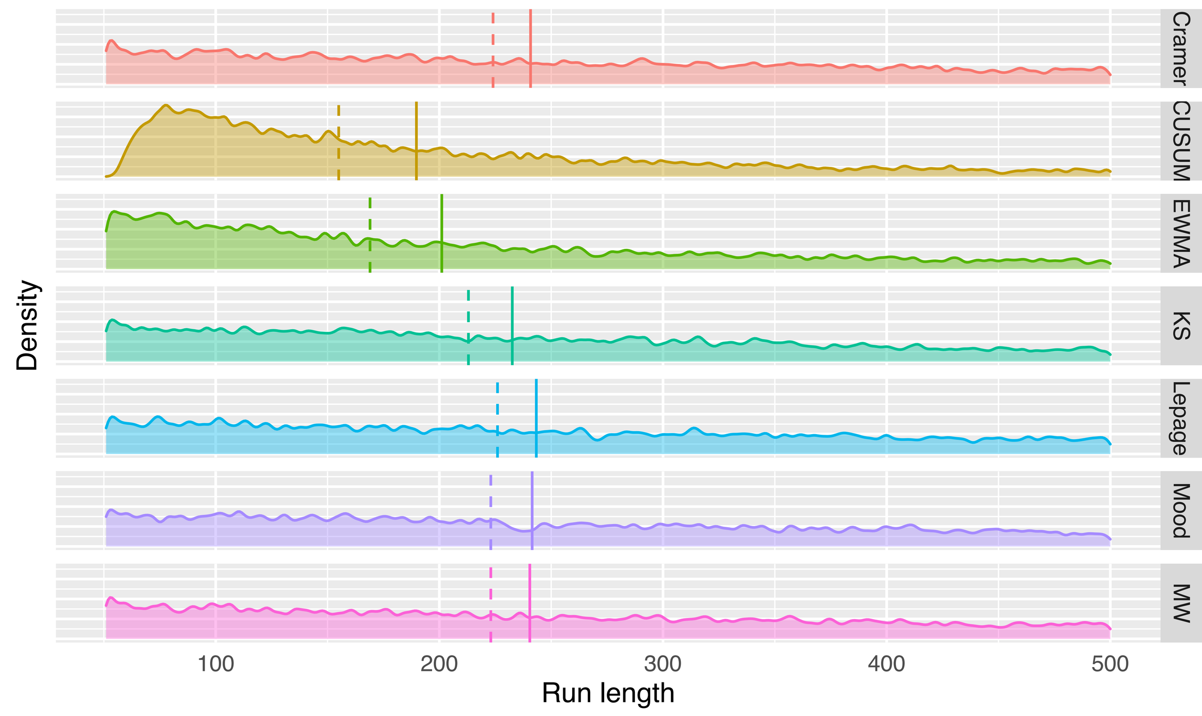

Figure 1: Run Length 0 distributions

• **Applications.** Online change-point detection algorithms can be used to detect changes in volatility, as shown on **Figure 2** where it was used to detect changes in Tesla's stock daily returns.

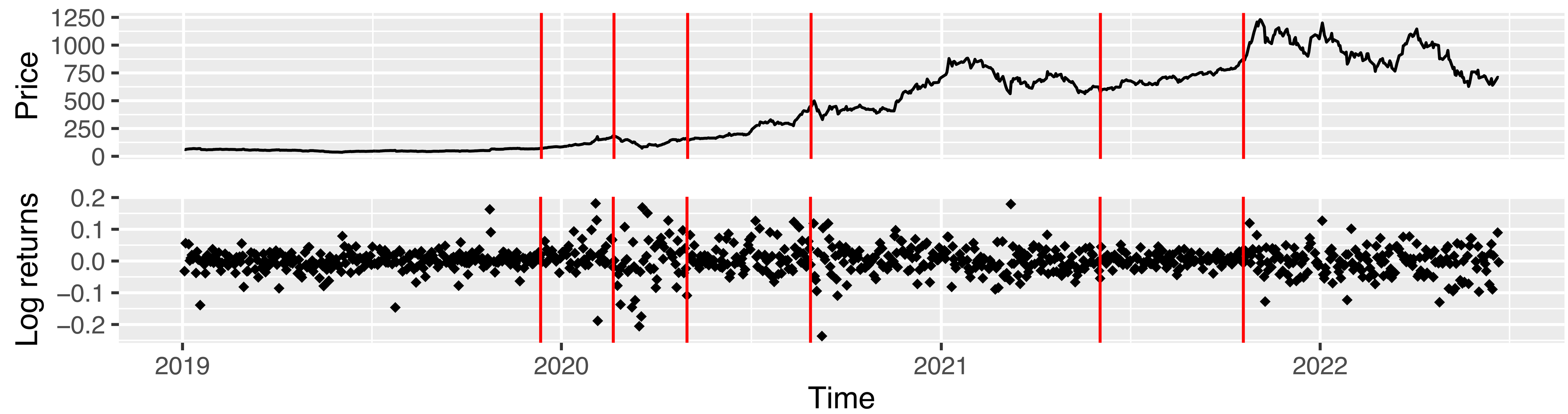

Figure 2: Online change-point detection algorithm (Lepage) on Tesla's stock daily data between 2019 and today

5. FUTURE RESEARCH

We have so far used change-point detection algorithms on univariate data: we may then focus on **neural networks methods** as the one described in (3) applied to **multivariate data** and how its performances are affected by the **choice of its parameters** (RNN, LSTM, Convolutional, MLP, windowing, lag). Other Deep Learning techniques involve **variational auto-encoders**.

6. REFERENCES

[1] Dean A. Bodenham and Niall M. Adams. Continuous monitoring for changepoints in data streams using adaptive estimation. 2017.

[2] Mikhail Hushchyn, Kenenbek Arzymatov, and Denis Derkach. Online neural networks for change-point detection. 2020.

[3] Gordon Ross, Dimitris Tasoulis, and Niall Adams. Nonparametric monitoring of data streams for changes in location and scale. 2012.