Universiteit Leiden

Master Computer Science

Bayesian Online Change Point Detection: an application to sports and e-sports data

Name: Marco Haitink
Student ID: S2629194

Date: 30/06/2022

Specialisation: Data Science

1st supervisor: Dr. Mike Preuss
2nd supervisor: Dr. Rens Meerhoff

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

To my parents Patrizia and Oscar

Acknowledgements

I would first like to express my utmost gratitude to my supervisors, Doctor Meerhoff and Doctor Preuss, for their guidance and precious feedback throughout this thesis project. Then, I would like to thank my parents for their endless support and love. Lastly, I would like to thank Mati for sharing this experience, and many others, with me.

Contents

Chapter 1

Introduction

In recent years, both multimillionaire industries of traditional sports and e-sports have undergone profound revolutions, fuelled by data. An ever-increasing availability - and quality - of event data, coupled with more and more capable algorithms for their analysis, have inspired significant improvements in the way a game is understood and, hence, the creation of more and more effective strategies. The dominant strategy is also known in e-sports as the META (Most Effective Tactic Available). If a team can adapt to such new dominant strategy early, they may benefit from a competitive advantage over teams that are reluctant to radically changing their strategy. In addition to simply understanding the mechanics of a sport -or e-sport-, it may thus be beneficial to identify such changes in strategy.
For instance, NBA basketball has undergone the so-called "3 point revolution"[1]: in the last 10 seasons alone, the frequency of 3 point shots relative to all scoring attempts has doubled. Such a radical change originated from the realization that, historically, 3-point shot attempts have yielded, on average, more points (1.05, source: [2]) per attempt than regular 2-pointers (0.79). Such a discovery has altered the way the game is played profoundly, and teams who managed to adapt to this new paradigm quickly have benefited from it significantly: 7 out of the last 10 seasons, in fact, saw the NBA title being awarded to a team in the top 10 positions of the 3-point shot attempts League rank (source: *Basketball-reference.com*[3]).
Game developers of popular e-sports title 'Call of Duty: Warzone' have leveraged on the great amount of data harvested from their community to improve player experience. In particular, through the release of game balancing patches, they aimed at preventing the emergence of a *META* regarding the players' choice of weaponry in their first-person-shooter game, in an effort to offer more freedom of choice to their players.
In such ever-evolving contexts, both in traditional sports and in e-sports, being able to detect a change in tactics and strategies becomes therefore crucial. In order to discover such changes in an automated manner, one can consider applying Change-point Detection algorithms to event data. The goal of Change-point Detection (CPD) algorithms is, in fact, to identify change-points in a sequence, that is, points characterized by a significant variation of the underlying distribution of the sequence data. Such algorithms have been increasingly popular in recent years, with applications to a wide range of fields including, but not limited to, finance [4], biomedical research [5], quality monitoring [6], and text mining [7]. To the best of our knowledge, though, no efforts to adapt and test such algorithms on sports and e-sports data have been documented. Consequently, this is the goal of this thesis, and can be formalized with the following research question:

> "Can changes in dominant strategies in sports and e-sports be detected using CPD algorithms?"

It is important to note how each of the two different proposed applications, sports and e-sports, offer unique challenges and complement each other in creating a more complete testing environment for change point detection algorithm: changes in sports tend to be more gradual as they usually originate -excluding those deriving from regulation changes- organically, while changes in e-sports usually happen more abruptly in response to patches.
In answering the main research question, this study presents a number of use cases, where CPD algorithms are applied to the highest basketball league in the world (NBA) data and a very popular e-sports game ('Call of Duty: Warzone') data.

Chapter 2

Background

In this chapter, we firstly introduce some terminology specific to Change Point Detection algorithms. Secondly, we present an overview of relevant existing literature on the subject. Lastly, we examine the mathematical framework at the base of Bayesian Online Change Point Detection (BOCPD) in detail.

2.1 Change Point Detection algorithms

Before presenting an overview of the existing literature regarding Change Point Detection algorithms, let us introduce a series of definitions and specific terminology used throughout this thesis.

Definition 2.1.1. We define a data sequence S

$$S = \{x_1, x_2, ..., x_n\}$$

where x_t is the t-th data point in the sequence.

Definition 2.1.2. A change point is a data point x_t of S where a transition between different states exists in the generative process of the sequence S.

CPD Algorithms Classification. CPD algorithms can be classified according to three main characteristics: online vs. offline, model-based vs. non-parametric, and univariate vs. multivariate algorithms.
An *offline* CPD algorithm looks at the entire data sequence, in a retrospective fashion, to identify change points. An *online* one, on the other hand, considers new data points as they become available. For this reason, in some literature ([8],[9]), online CPD algorithms are also referred to as "real-time" algorithms.
A *model-based*, or *parametric*, CPD algorithm relies on an underlying set of parameters that are updated as the algorithm is trained on labelled data with the goal of learning the distribution of data in the sequence. A *non-parametric* one, conversely, operates without creating any assumption about the distribution of data.
Based on the data sequence nature, and in particular on the number of variables included in it, a final distinction can be made between *uni-variate*, if just one variable is considered, and *multi-variate* CPD algorithms. Having introduced these concepts, let us present the existing literature on Change Point Detection algorithms.

CPD Relevant Literature. CPD originates from industrial quality control processes, where change points were identified using a cost function in an online fashion. In their 1954 paper [10], Page introduced an algorithm based on the moving average of recent observations to identify abrupt changes of the distribution mean parameter Θ. An obvious shortcoming of such mechanism is the potential inability to detect small and gradual changes to Θ, especially for high values of it. Other relevant early studies include the work of Yao [11], who also opted for a frequentist approach, proposing an algorithm based on Schwarz' criterion to estimate the number of change points in a sequence. However, the mechanism proved to be inconsistent. In more recent years, a number of studies proposed to tackle the problem of CPD using probabilistic approaches. Pignatiello and Simpson [12] focused on Likelihood Ratio tests applied to CUSUM (cumulative sum) charts, while Fahmy and Elsayed [13] proposed to use Maximum Likelihood Estimation instead. Both studies managed to significantly outperform previous efforts. In this context, methods based on Bayesian statistics were also introduced. The first efforts in such direction are found in the work of Fearnhead and Liu [14] and Adams and MacKay [15], the latter being our reference paper. Methods based on Bayesian statistics, and Bayesian inference in particular, offer a great advantage over frequentist approaches: while the latter formulate a prediction regarding the next observed data point in the form of a single point, the former mandate the use of a distribution for the task. This allows methods based on Bayesian inference to better capture the variance of a sequence.

While a great number of studies exists in literature, expanding the framework of application of CPD algorithms and improving their performance, we were not able to find any study that applied such techniques to sports, or e-sports, data.

2.2 Bayesian Online Change Point Detection

In their 2007 paper [15], Adams and MacKay introduced a novel algorithm for Change Point Detection based on Bayesian statistics called Bayesian Online Change Point Detection (hereinafter abbreviated as BOCPD). Let us present the mathematical framework at the foundation of BOCPD, occasionally borrowing language from Gundersen et al. [16]

As per definition 2.1.1 above, let us assume we have a data sequence S where x_t is the t-th data point of S. Per the Partition Model introduced by Barry et al. [17], we can assume that S can be partitioned in such a way that all the data points in the same partition are drawn from the same distribution and are independent and identically distributed (i.i.d.). To formalize this assumption: for each partition p, we assume that observations x_t of p follow a distribution

$$P(x_t|\eta_p)$$

where η_p are the generative parameters of the distribution. For instance, in the case of a Gaussian distribution, η_p would represent the mean and variance parameters of the distribution.

According to this model, the partitions in which S is split are divided by change points. In order to identify such change points, Bayesian Online Change Point Detection aims at modelling variable r, representing the run length, i.e. the time passed - expressed in terms of number of data points - since the last observed change point. Variable r, at any t point in time, can only assume two values:

$$r_t = \begin{cases} 0, & \text{if a change point occurs at time } t \\ r_{t-1} + 1, & \text{otherwise} \end{cases}$$

The transition probabilities, hereinafter referred to as *changepoint prior*, for r are therefore defined as

$$P(r_t|r_{t-1})$$

In order to be able to detect a change point, BOCPD computes both the *run-length posterior distribution*, defined as:

$$P(r_t|x_{1:t})$$

and the *posterior predictive distribution*:

$$P(x_{t+1}|x_{1:t})$$

In a standard Bayesian inference model, the posterior predictive distribution, given a set of observations X and model parameters Θ, is calculated, for a new observation x, as:

$$P(x|X) = \int \overbrace{P(x|\Theta)}^{\text{Model}} \overbrace{P(\Theta|X)}^{\text{Posterior}} d\Theta$$

In BOCPD, this is rendered as:

$$P(x_{t+1}|x_{1:t}) = \sum_{r_t} \overbrace{P(x_{t+1}|r_t, x^{(l)})}^{\text{Model}} \overbrace{P(r_t|x_{1:t})}^{\text{Posterior}} \tag{2.1}$$

While quite similar in form to the standard Bayesian inference equation, this one substitutes the integration part in favor of a marginalization process over the run length, where marginalization simply implies repeating the calculation for each possible value of the variable.
In Equation 2.1, we introduced a new notation, $x^{(l)}$: this is used to indicate all those data points belonging, referring back to Barry et al. [17] again, to the same partition.
Let us first discuss how the Posterior portion of 2.1 is calculated. Following Bayesian statistics principles:

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{\sum_{r'_t} P(r_{t'}, x_{1:t})} \tag{2.2}$$

and recursively, by applying marginalization and the chain rule, we can calculate the joint distribution over run length r as

$$\begin{aligned} P(r_t, x_{1:t}) &= \sum_{r_{t-1}} P(r_t, r_{t-1}, x_t, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t, x_t|r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(x_t|r_t, x_t^{(l)}) P(r_t|r_{t-1}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{2.3}$$

Each of the three parts in the final form of 2.3 can be individually considered. For ease of reference, let's borrow the notation used in Gundersen et al. [16] and let's label each part:

$$P(r_t, x_{1:t}) = \sum_{r_{t-1}} \overbrace{P(x_t|r_{t-1}, x_t^{(l)})}^{\text{Model}} \underbrace{P(r_t|r_{t-1})}_{\text{CP prior}} \overbrace{P(r_{t-1}, x_{1:t-1})}^{\text{Message}} \tag{2.4}$$

We have then defined a recursive equation to compute the Posterior or, in our case, the Run-length Posterior, as a function of (1) The 'Model', referred to in Adams and MacKay [15] and hereinafter as the Underlying Probabilistic Model (UPM) Predictive, i.e. the likelihood of observation x_t given the run length at time $t-1$ and the observations seen since the last change point, (2) The

Change Point Prior ('CP prior') which expresses the probability of seeing a Change Point given a run length, and (3) The 'Message': the same joint distribution, calculated at the previous data point, thus the recursive nature of the equation.
Given that we can calculate the UPM Predictive and CP Prior portions 2.4, we will have everything needed to successfully implement BOCPD. Let us elaborate on these two components.

Underlying Probabilistic Model Predective

In order to simplify the calculations needed to compute the UPM Predictive, Adams and MacKay [15] leverage the properties of conjugacy and the exponential family. Normally, in fact, to compute

$$P(x_t|r_t, x_t^{(l)})$$

we would need to integrate over the model's parameters Θ:

$$P(x_t|r_t, x_t^{(l)}) = \int P(x_{t+1}|\Theta)P(\Theta|r_t, x_{1:t})\, d\Theta$$

Note that, here, $P(x_{t+1}|\Theta)$ is known, as we can arbitrarily decide that our data is best modelled by any member of the exponential family, that being a Gaussian distribution as it is the case in our experiment presented later, or any other. A great advantage offered by the exponential family is that we can also avoid computing the second part of the equation. Gundersen[18] offers a very clear explanation of how this is achieved.
First, let us review the prior predictive for any conjugate model. Let X be our observations, $\hat{x}$ a new observation, Θ the model parameters, and α the hyperparameters. The prior predective is the distribution of $\hat{x}$ given the marginalization [19] of α over Θ:

$$P(\hat{x}|\alpha) = \int \overbrace{P(\hat{x}|\Theta)}^{\text{Model}} \overbrace{P(\Theta|\alpha)}^{\text{Prior}}\, d\Theta \tag{2.5}$$

Knowing that, since we are considering a conjugate model, the prior has the same form of the posterior, allows us to write

$$P(\Theta|X, \alpha) = P(\Theta|\alpha')$$

for some hyperparameters α' different from α. And consequently

$$\begin{aligned} P(\hat{x}|X, \alpha) &= \int P(\hat{x}|\Theta)P(\Theta|X, \alpha)d\Theta \\ &= \int P(\hat{x}|\Theta)P(\Theta|\alpha') \\ &= P(\hat{x}|\alpha') \end{aligned}$$

If we can compute the hyperparameters α', then, we can calculate the posterior without needing to integrate. Exponential Family (EF) models offer a convenient way of calculating α'.
Equation 2.5 can be rewritten, in the specific case of EF models, as

$$P(x|\eta) = h(x)g(\eta)exp\{\eta\top u(x)\} \tag{2.6}$$

where η is the *natural parameter*, $h(x)$ is the *underlying measure*, $u(x)$ is the *sufficient statistic* of the data, $g(\eta)$ is a *normalizer*, and $\top$ is the transpose operator. Following Gutierréz-Peña et al. [20], we can write the conjugate prior, with hyperparameters υ and χ - replacing α' - as

$$P(x|\eta) = f(\chi, \upsilon)g(\eta)^{\upsilon}exp\{\eta\top\chi\} \tag{2.7}$$

with $f(\chi, \upsilon)$ depending on the particular member of the EF family chosen. The posterior is then obtainable by multiplying Equation 2.7 times the likelihood, defined as

$$(\prod_{i=1}^{N} h(x_n))g(\eta)^n exp\{\eta\top \sum_{n=1}^{N} u(x_n)\} \tag{2.8}$$

to obtain the final form of the conjugate posterior:

$$P(\eta|x, \chi, \upsilon) \propto g(\eta)^{N+\upsilon} exp\{\eta(n = 1 \sum_{n=1}^{N} u(x_n) + \chi)\}. \tag{2.9}$$

By using the properties of the exponential family, we successfully managed to define the posterior distribution in a simpler form. From an algorithmic perspective, in fact, we will just be interested in keeping the parameters υ' and χ' of 2.9 updated at every new observation in order to be able to calculate the 'UPM' portion of 2.4:

$$\begin{gathered} \upsilon' = \upsilon_{prior} + N \\ \chi' = \chi_{prior} + \textstyle\sum_{i=1}^{N} u(x_n) \end{gathered} \tag{2.10}$$

Once we have set the initial conditions, we can take advantage of the message passing mechanism to easily update parameters υ and χ. We have two possible cases: either we reset the run length having observed a changepoint, or we compute $\upsilon^{(l)}$ and $\chi^{(l)}$ for each possible run length l:

$$\begin{gathered} \upsilon_{t(0)} = \upsilon_{prior} \\ \chi_{t(0)} = \chi_{prior} \\ \upsilon_t^{(l)} = \upsilon_{t-1}^{(l-1)} + 1 \\ \chi_t^{(l)} = \chi_{t-1}^{(l-1)} + 1 \end{gathered} \tag{2.11}$$

Having taken care of the UPM Predictive portion in Equation 2.4, let us now present how the Change Point Prior is calculated.

Change Point prior

To discuss the calculation of the Change Point prior, we first need to introduce two concepts borrowed from survival analysis: survival, and hazard, functions [21].
In reference to our study, let us assume that $P(\tau)$ represents the probability that run length r is equal to τ, and let T be a random non-negative value for the run length. Then, the survival function $S(\tau)$ can be defined as the probability that T is greater than τ:

$$S(\tau) = P(X \geq \tau) = \sum_{\tau'=\tau}^{\infty} f(\tau')$$

The hazard function $H(\tau)$ is consequently defined as:

$$H(\tau) = \frac{f(\tau)}{S(\tau)}$$

We can now calculate the Change Point prior portion of 2.4. We can use the above introduced hazard function $H(\tau)$ to estimate the probability of a change point happening at any time, given that it hasn't been observed yet, as:

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}+1) & \text{if } r_t = r_{t-1}+1 \\ 0 & \text{otherwise} \end{cases}$$

The only step left before being able to implement BOCPD successfully is defining the initial condition of our recursive algorithm.

Initial Conditions

The first, and simplest, case is: we assume that a change point has been observed right *before* our first observed data point. In this case

$$P(r_0 = 0) = 1 \tag{2.12}$$

This condition could be met, for example, if we were to start observing event data from a single game of basketball, with no previous history.
Another scenario, perhaps more common in a real-life setting, would emerge if we were to start observing data about a process already happening, for which we could reasonably assume the existence of a change point in the future. An example borrowed from the original paper would be the increase in average temperatures as a result of the ozone layer depletion, for instance. In this case, to account for the existence of previous data, we would define the initial condition as

$$P(r_0 = \tau) = \frac{1}{Z}S(\tau) = \frac{1}{Z}\sum_{\tau'=\tau+1} f(\tau')$$

That is, the survival function S(τ) normalized by a constant Z. What this implements, practically, is a way to accumulate the probability of observing a change point the more we look into the future - i.e. the larger τ.
As a summary:

$$P(r_0) = \begin{cases} 1 & \text{if a change point occurs at time } t = 0 \\ P(r_0 = \tau) & \text{otherwise} \end{cases}$$

Algorithm Sequential Implementation

Having analyzed all the three components of 2.4, let us now schematically Report how Bayesian Online Changepoint Detection works in a sequential manner.

1. **Define priors and initial conditions**

$$P(r_0) = \begin{cases} 1 & \text{if change point at time } t = 0 \\ P(r_0 = \tau) & \text{otherwise} \end{cases}$$

$$\upsilon_1^{(0)} = \upsilon_{prior}$$

$$\chi_1^{(0)} = \chi_{prior}$$

$$t = 1$$

2. **Observe new data point** x_t

3. **Calculate UPM predictive probabilities** for each possible run length l. This is achieved by exploiting the forward message-passing mechanism presented in the dedicated section above.
$$\pi_{t-1}^{(l)} = p(x_t|\upsilon_{t-1}^{(l)}\chi_{t-1}^{(l)})$$

4. **Calculate growth probabilities**. The probability, for each run length l, to continue *growing*, i.e. no changepoint is detected.
$$p(r_t = l, x_{1:t}) = p(r_{t-1}, x_{1:t-1})\pi_{t-1}^{(l)}(1 - H(r_{t-1}))$$

5. **Compute Changepoint Probability**, i.e. the probability that the run length l resets to zero as a changepoint has been observed.
$$p(r_t = 0, x_{1:t}) = \sum_{r_{t-1}} p(r_{t-1}, x_{1:t-1})\pi_{t-1}^{(l)}(H(r_{t-1}))$$

6. **Compute the Evidence** in equation 2.2

7. **Compute the RL Posterior** in equation 2.2

8. **Update parameters υ and χ** in equation 2.11

9. **Calculate the prediction**, equation 2.1

10. **Observe next data point** wiht $t = t + 1$ and return to step 2

In the next chapter, we discuss the methodology used.

Chapter 3

Methodology

In this chapter, we present the methodology used throughout this thesis, introducing the proposed experiments and explaining the rationale behind each of them.
In order to test whether BOCPD can be successfully, and with profit, applied to sports and e-sports data in order to detect changes in the dominant strategy, we designed three experiments. In the following paragraphs, each of these experiments is presented.

3.1 Experiment on Artificial Data with Known Change Points

Firstly, we propose an experiment in which BOCPD is applied to artificial data with known change points. This initial experiment offers three significant advantages over the application to real-world data:

- The ability to determine the complexity of the change point detection task. Through the introduction of two parameters -magnitude of change and abruptness of change- for the data generation process, we can test the algorithm on a set of increasingly more difficult scenarios: intuitively, it is an easier task to detect a change that is significant in magnitude and happens abruptly, than a smaller and more gradual one.

- The ability to qualitatively evaluate the results obtained, comparing the result obtained against a ground truth. In particular, we will be considering F-Score and Recall metrics.

- The ability, when applying the algorithm to real-world data with (un)known change points, to make an a-priori hypothesis regarding the expected performance of the algorithm by comparing the characteristics of the data in terms of the two aforementioned parameters (magnitude of change and abruptness of change) to the data that will be used for this experiment.

Furthermore, results from this experiment will allow us to formulate hypotheses regarding the expected performance of BOCPD on real-world data by comparing their characteristics to the characteristics of the artificially generated data.

3.2 Experiment on Real-world Data with Known Change Points

Secondly, we present a series of experiments in which BOCPD is applied to real-world data with known change points, where a consensus exists among insiders regarding the presence of a clearly identifiable change.
For sports, we will present three different use cases, characterized by an increasing difficulty of the change point detection task. The first one concerns the Olympic discipline of javelin throw, where the world record progression is analyzed to assess whether the change observed in 1986 following the introduction of a new design for the javelins used is correctly detected by BOCPD. The second one uses data from Speed Skating, where the world record progression is inspected to test the ability of BOCPD to detect the less evident -in terms of magnitude of change in the differences registered, compared to the previous experiment- impact of the adaption of a new technology, clap skates, in 1997. The third and last one considers NBA basketball data, and in particular the number of 3 point shots attempted for each game day, of seasons 1992-93 to 1994-95, where the three point line distance was reduced. This experiment closely replicates the most likely application of BOCPD in a real-world scenario, as the algorithm is applied to granular data collected daily.
For e-sports, we will present two different use cases, where BOCPD is applied to data regarding the pick rate of various weapons in the first-person-shooter game 'Call of Duty: Warzone' across different game patches. In the second proposed use case, we also test the possibility of applying BOCPD in parallel to different data sequences sharing the same timeline, as a way to analyze possible effects that a change in popularity of one weapon introduces on the popularity of a second one.

3.3 Experiment on Real-world Data with Unknown Change Points

Lastly, we aim at replicating the experiment proposed by Adams and MacKay in the original paper [15]. In order to do so, we will present two different experiments. In the first one, we test BOCPD on NBA basketball 3 point shots attempts collected for each game day across multiple (2000-01 to 2019-20) seasons. In analyzing the results, we will be interested in assessing whether BOCPD has managed to detected any change point that might suggest the beginning of the '3 point revolution' introduced in Section 1. In the second experiment, we apply BOCPD to data reflecting the popularity of a particular weapons across all the different deployed game patches since the release of the game.

Chapter 4

Experiment on Artificial Data with Known Change Points

With the experiment proposed in this chapter, we aim to respond to the following research question:

> How does Bayesian Online Change Point Detection perform on artificial data with known change points?

In answering such question, we firstly present our contribution to the original algorithm present by Adams and MacKay [15], consisting of (1) a mechanism to extract detected change points in an automated way and (2) an evaluation pipeline to assess the perfomance of BOCPD on labeled data. Secondly, we present our first proposed experiment and its benchmark settings. Lastly, we discuss the results obtained, and we argue that BOCPD performs well in most scenarios, only showing sub-optimal results for small and gradual changes in the distribution mean.

4.1 Algorithm Adaptation

In order to be able to quantify the performance of BOCPD, we first need to adapt the original algorithm presented in Adams and MacKay[15]. In the original paper, in fact, the authors test the proposed algorithm on three datasets containing unlabeled data, one of which reports the daily returns of the Down Jones Industrial Average (Figure 4.1), and motivate manually-detected change points by looking at historically significant events that could have influenced such returns (i.e. introduce a change point). What we aim to do, instead, is evaluating BOCPD in an automated way and on the basis of a quantifiable metric. In the following sections, we present the changes introduced to enable this.

4.1.1 Change Point Detection Mechanism

By default, the output of BOCPD is the Run Length Posterior discussed in Chapter 2, rendered visually in the bottom portion of Figure 4.1. Let us examine the figure, as it presents various elements that are important to our discussion. On the top portion of the image, we see a sequence of data representing the daily returns of the Down Jones Industrial Average, overlayed with the predicted volatility (solid darker line). On the bottom one, the output of BOCPD is presented. On the y-axis, values for the Run Length, i.e. the length of a sequence until a change point occurs,

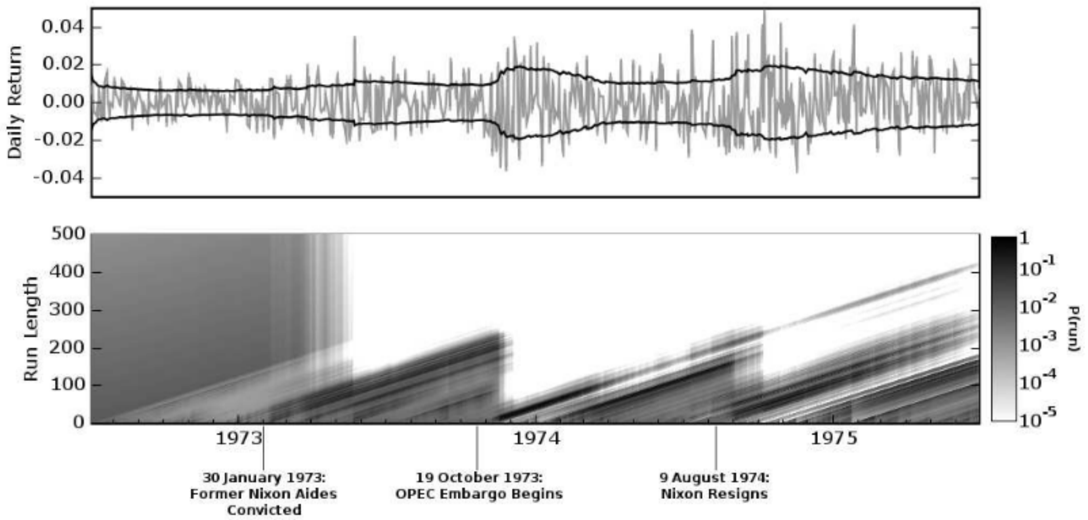


Figure 4.1: RL Posterior as output by BOCPD, image from [15]

variable r_t are reported. Each different sequence is identifiable by its starting point, corresponding graphically to those points in which the value of Run Length drops to zero. This gives the graph its saw-toothed shape, where a taller 'tooth' corresponds to a longer sequence. Each pixel is then colored using a logarithmic color scale, where darker indicates a higher probability in the Run Length Posterior probability. It is interesting here to note how, in the beginning of the sequence, where only few data points are available to the algorithm, the probability of the Run Length is almost equally distributed across value from 0 to 500. This represents an initial uncertainty which progressively diminish as more data points are analyzed, eventually bringing to the discovery of a first change point around year 1973.
To better illustrate the concept of Run Length Posterior, let us disregard Figure 4.1 for a moment and assume we are zooming in on the particular point of a sequence X of T=200 data points, say x_{100}. The Run Length Posterior, calculated following 2.4 and implemented on the practical side with a lower triangular matrix called R will be, for x_{100}, the 100-th column of matrix R. In particular, each element i in $R[x_{100}]$ will be filled with the probability that run length r at x_{100} is equal to i. Let us present two different opposite possible scenarios:

i	P(r=i)
100	**0.85**
99	0.12
98	0.10
...	
2	0.03
1	0.02
0	0.01

i	P(r=i)
100	0.00
99	0.00
98	0.00
...	
2	0.01
1	0.01
0	**0.96**

Table 4.1: Tabular visualizations of an hypothetical 100-th column of matrix R

In the case of the table on the left, the output of BOCPD would suggest that no change point has been observed so far, since the highest probability level is observed for a Run Length value of 100 (for brevity, we will refer to this as MAX(RL(x)) in subsequent pseudo-code snippets). Conversely, for the table on the right, the output would suggest that data point x_{100} is a change point.

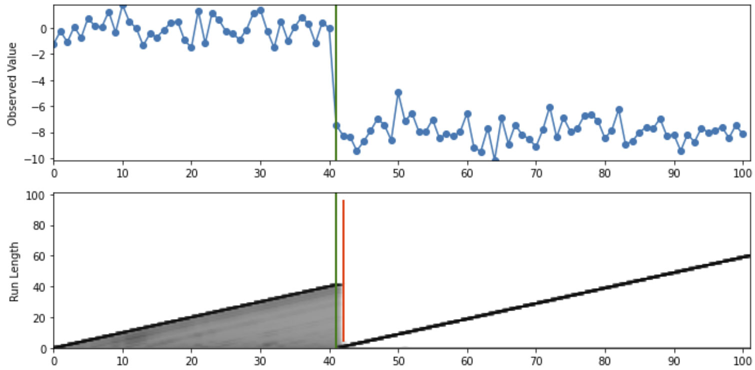
Returning now to Figure 4.1, we can observe how the authors inferred change points by looking at times in which the Run Length resets to 0 and labeled such points with possible reasons for the existence of a change point. While this visual representation of the output is informative, and we will refer back to it in following chapters, it is not suitable for evaluating the ability of BOCPD in detecting change points in an automated -i.e. without having to refer to the visual output and manually identifying change points- fashion, nor to quantify said performance with a metric. For this reason, we have implemented a mechanism to extract detected change points from the Run Length Posterior matrix similar to the one presented in [22]. Let us introduce it below.

In order to prepare BOCPD for our first proposed experiment, we want to obtain, from the Run Length Posterior R, a list of detected change points. By tracking for which run length the maximum probability is observed across consecutive data points, we can easily achieve this: if the Run Length value r for which we observe the highest probability for a certain data point x_t is smaller than the one from the previous point x_{t-1} plus one, which would be the expected behavior if no change point is detected, then x_t is identified as a change point. The code snippet used for change point detection is presented, in form of pseudo-code, in Algorithm 1.

Algorithm 1 Detected Change Points from RL Posterior

```
procedure DETECTED(data, R)
    detected = [ ]                                        ▷ Initialize empty list of detected CPs
    for each data point x_{2,...,}x_n do
        probShift = MAX(RL(R[x_t])) - MAX(RL(R[x_{t-1}]))
        if probShift <= 0 then
            detected ← x                                  ▷ Add x_t to detected CPs
        end if
    end for
    return detected
end procedure
```

In order to better illustrate how such detection mechanism works in practice, we report here an example where the graphical output is presented together with a tabular visualization of columns from matrix R (using the same representation as Table 4.2) right before and right after a detected change point at index 42. In the graph, the position of the known change point is denoted by the green line, while the red line corresponds to the point at which the change is detected.



i	P(r=i)
44	0.00
43	0.00
42	0.00
41	**0.76**
40	0.02
39	0.01

i	P(r=i)
5	0.00
4	0.00
3	0.01
2	0.01
1	**0.97**
0	0.01

Table 4.2: Tabular visualizations for (a portion of) column 41 and column 42 of matrix R

As we can see, the change point at index 42 is detected, as the difference between the index corresponding to the highest Run Length Posterior probability value of the two consecutive points 41 and 42, calculated at line 4 in Algorithm 1, yields a negative value (1-41<0). Furthermore, we can observe how the highest Run Length probability value observed at data point 42 corresponds to a length of 1, which correctly reflects a change point happening at data point 41.

With such mechanism, we have implemented a way to extract a list of detected change points that can be tested against a ground truth one. The next step in preparing our first experiment is generating some artificial data with ground truth change points.

4.1.2 Data Generation

To align to the original paper [15], we have decided to use for our first proposed experiment a sequence of T data points following a Gaussian distribution with fixed variance $var0$ and a changing mean. An arbitrary value of $T = 2000$ was chosen for the sequence length, which is also aligned to the size of datasets in the paper. In generating the sequence, a number of parameters are defined. These are presented in Table 4.3.

Once we have defined such parameters, let us introduce the algorithm used for data generation, presented in form of pseudo-code in Algorithm 2.

Having defined a way to extract the detected change points from the posterior distribution variable R, and a way to generate the data, the last step needed to prepare for our experiment is creating a function -in our case, a series of functions- to evaluate the performance of BOCPD. Let us introduce it below.

	Variable	Description
Data Sequence Length	T	The length of the sequence defined in terms of number of data points
Variance	$var0$	The fixed variance of the sequence
Initial Mean	$mean0$	The initial mean of the sequence
Mean after Change Point	$meanX$	The absolute difference between old and new mean after a change point is introduced
Hazard Function	h	The hazard function presented in Equation 2.2. (For ease of use, we define it in the algorithm as 1 on the expected distance between CPs (e.g. $\frac{1}{100}$))
Abruptness of Change	abr	The abruptness of the change in mean, defined as the number of data points before the new mean value is reached. Low values indicate a sudden change (few data points), high values a gradual one.

Table 4.3: The parameters used for generating the data

Algorithm 2 Data Generation

```
procedure DATA GENERATION(T,var0,mean0,meanX,h,abr))
    data = [ ]
    changepoints = [ ]
    meanX = mean0
    for i in range(0, d) do
        if random(0,1) < h then                       ▷ using h as defined in Table 4.3
            meanX = random(mean0, meanX)              ▷ calculating new mean value after CP
            for a in range(0, abruptness) do          ▷ diluting the change over abr data points
                data ← random(mean0, meanX)
            end for
            changepoints ← i
        end if
        data ← random_value(meanX, var0)
    end for
end procedure
```

4.1.3 Evaluation Functions

In order to assess the performance of BOCPD, we are interested in the F-Score metric, defined as:

$$FScore = \frac{TP}{0.5 * (FP + FN)}$$

where:

- TP = True Positives; correctly detected change points
- FP = False Positives; incorrectly detected change points
- FN = False Negatives; undetected change points

In order to calculate the F-Score for a particular experiment run, we have created a pipeline of three functions: (1) a function to calculate the F-Score of a given run, (2) a function that pairs each detected, or candidate, change point to the closest ground truth change point, and (3) an helper function that supports this pairing function. Let us present all three, and the reasoning behind the evaluation process.

The entry point of our evaluation function is the pairing function labeled with (2) above, which is reported in the form of pseudo-code in Algorithm 3. The function receives in input a list of candidate change points returned by Algorithm 1 and outputs a dictionary where each ground truth change point is marked as either detected or undetected. To explain how the mechanism to accomplish this works, we first need to make some important considerations.

As we can see in Figure 4.1, in fact, it can be the case that a change point is only detected -i.e. the RL posterior probability resets to zero - after some delay. For this reason, we need to introduce some tolerance in the system, and consider a ground truth change point as detected if the a candidate change point is within a certain distance window from it. Using the notation introduced in Table 4.3 for the hazard function, we define such distance window as $[0, h^{-1}]$. This limits the maximum delay to still consider a change point as detected to the expected maximum distance between change points. In order to take this possible delay in the discovery of a change point into account in our pairing function, we defined a helper function that returns, for each candidate change point, the closest preceding ground truth change point, and its distance to it. Given the straightforward nature of such function, we omit its formal presentation as pseudo-code.

While we discuss this aspect, let us make an important remark: while the delay in discovery is not taken into account for the purpose of evaluating the algorithm performance in this thesis, we recognise its importance, and thus decided to include in the section about possible future developments.

Having made these important considerations, we can now properly introduce our pairing function, reported in Algortihm 3. The logic is the one introduced in the paragraph above: each candidate change point gets paired with its closest preceding ground truth change point -returned by the helper function - and, if the conditions are met, it is labeled as correctly detected. Finally, the function feeds the F-Score calculator function, reported in Algorithm 4, a dictionary where ground truth change points have a value of 1 for ‘detected’ or 0 for ‘not detected’. As the final output of our evaluation pipeline, the F-Score, calculated following Equation 4.1.3, is returned.

Algorithm 3 Create Result dictionary

procedure RESULTS DICTIONARY(*ground truth change points*, *candidate change points*)
    results = {}
    if *detected* is empty **then**
        return *results*
    end if
    for *cp* in *cps* **do** ▷ for each change point in the ground truth list
        results[*cp*] ← [0, T+1] ▷ 0 for undetected, maximum delay possible
    end for
    for *d* in *detected* **do**
        nearest ← find_nearest(*cps*, *d*)
        if *nearest* and *d* - *nearest* $< h^{-1}$ **then**
            results[*nearest*] = $[1, d - nearest]$
        end if
    end for
    return *results*
end procedure

Algorithm 4 Calculate F-Score

procedure F-SCORE(*results*,*ground truth change points*, *candidate change points*)
  if no ground truth change points **then**
    if no candidate change point **then**
      return 1
    end if
    return 0
  end if
  $TP \leftarrow$ number of ground truth change points detected
  $FN \leftarrow$ number of ground truth change points undetected
  $FP \leftarrow$ number of candidate change point minus ground truth change points
  $FScore \leftarrow$ Equation 4.1.3
  return $FScore$
end procedure

Having presented how the original BOCPD algorithm was modified to accommodate our needs, let us now discuss the setup of our first proposed experiment and the results obtained.

4.2 Experiment

4.2.1 Experimental Settings

To present the setup for this first experiment, let us refer back to Table 4.3. As we discussed, we introduced a parameter *abr* defining the abruptness of the distribution shift after a change point occurs: low values of *abr* indicate a change happening across a small amount of data points, and therefore a sudden shift, while high values indicate a smoother one, spread across a higher number of data points. By combining different values of abruptness with the parameter *meanX*, representing the magnitude of the shift in the distribution mean, we can tweak the difficulty of the change point detection task: intuitively, for instance, detecting a sudden and significant change in the distribution values is easier than detecting a smooth and rather small one. In Table 4.4, we present the experimental setting in reference to the parameters for the data generation process described in Algorithm 2.

	Variable	Values
Data Sequence Length	T	2000
Variance	var0	1
Hazard Function	h	1/500
Initial Mean	mean0	1
Mean after Change Point (compared to Initial Mean)	meanX	[+1, +2, +5, +10]
Abruptness of Change (compared to Sequence Length)	abr	[1/2000, 1/1000, 1/500, 1/100]

Table 4.4: Experiment Parameters

This choice of parameters values reflects our best efforts to create a meaningful benchmark for our adaptation of BOCPD, also given the absence of similar experiments in related literature, where the algorithm is tested on increasingly more difficult scenarios. By combining values of mean shift and abruptness, we identified three difficulty levels: 'easy', where the mean value changes significantly (+10) and the abruptness of change is rather sudden (1/2000, 1/1000); 'medium', where a smaller mean value shift (+5) is couple with lower values of abruptness (1/500); 'hard', where a small (+2) change in the mean value is paired with a smooth change (1/100). Furthermore, we also created a 'realistic' setting, by extrapolating the values for mean shift (+1) and abruptness of change (1/100) from the use cases presented later in Chapter 5. Let us now discuss the results obtained.

4.2.2 Results

To present our results, we use the boxplot in Figure 4.2. The F-Score reported are, as indicated in the title of the plot, obtained from 500 iterations for each *(shift in mean, abruptness of change)* combination. On the y-axis, values of F-Score are reported. On the x-axis, the *(shift in mean, abruptness of change)* pairs are reported in their absolute form, i.e. not in relation to the sequence length, but with their nominal value.

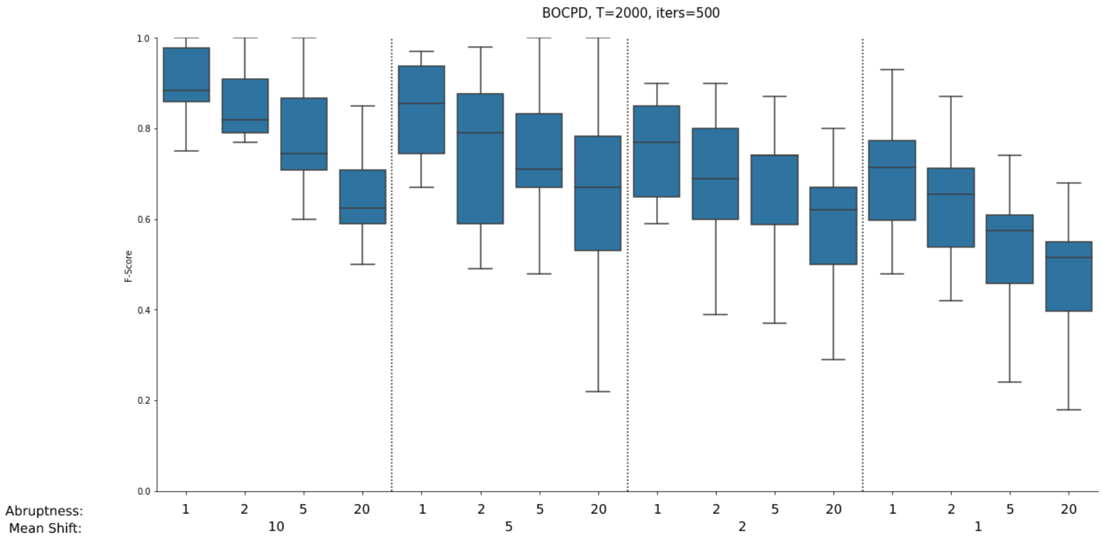


Figure 4.2: Boxplot reporting F-Score results on artificial data with known change point

The results obtained are aligned to what can be expected in light of the considerations made in the paragraph above, as the increasing difficulty introduced by a higher smoothness of change and by a lower shift in the mean is reflected in the scores obtained. Given the composite nature of the F-Score metric, it being the harmonic mean of precision and recall, it is quite difficult to define what can can be considered a 'good' F-Score. Nonetheless, the results obtained all see a median value, as can be inferred by the boxplot, above the mid-range value of 0.5, which is a desirable outcome. Additionally, we also report scores obtained with the Recall metric, where Recall is defined as the fraction of relevant observations -in our case, change points- that are correctly classified -in our case, detected-. In a formula:

$$Recall = \frac{TP}{TP + FN}$$

where:

- TP = True Positives; correctly detected change points
- FN = False Negatives; undetected change points

Before we continue the discussion and present the recall scores obtained by our adaption of BOCPD, it is important to also introduce the natural counterpart to the recall metric: the precision metric. Precision can be defined as the fraction of correct positive predictions -in our case, actual change point that are detected as change points- over the total number of positive predictions -in our case, all points that are detected as change points-. In a formula:

$$Precision = \frac{TP}{TP + FP}$$

where:

- TP = True Positives; correctly detected change points
- FP = False Positives; incorrectly detected change points

To understand why we believe recall could be a meaningful metric for our case, and possibly more than precision and F-Score alone, let us consider an hypothetical application of our implementation of BOCPD. For instance, let us consider the case of a data analyst working for a sports team, willing to use the algorithm as a support system that can send them alerts when it detects a possible change in a particular aspect of the game being tracked. In this case, the analyst would want to make sure that the algorithm has a good recall, as this would mean not missing out on a change point which, if detected, could guarantee their team a competitive advantage against teams that are unaware of it. Of course, the downside of the recall metric is not penalizing false positives. This would translate, in such hypothetical scenario, into them possibly receiving false alarms about non-existing change points. However, it can be argued that with enough expertise and experience, we can expect our hypothetical data analyst to be able to discriminate between true positive and false positives, at least in most cases.
Having explained the reasoning behind the choice of this additional evaluation metric, let us report in Table 4.5 the recall scores obtained by our algorithm with the same settings presented above in the discussion of the obtained F-Scores:
The results obtained for Recall are aligned to the F-Score ones. Solid performances are observed for almost all cases, with the exception of runs with the most difficult combinations of *(shift in mean,abruptness of change)* included in our benchmark.

4.2.3 Discussion

In this chapter, we first presented the changes we introduced to the original BOCPD algorithm presented by Adams and MacKay. Then, we introduced our experimental setup, designed to test the suitability and effectiveness of the adapted algorithm on different scenarios, characterized by an increasing difficulty of the change point detection task. In particular, we evaluated its effectiveness using both the F-Score metric and the Recall metric, whit comparable results. Such results showed high values for both metrics in those scenarios where the change happens more abruptly. In the context of the proposed experimental setup, in fact, where the artificial data used for testing are generated according to two main parameters, magnitude of change and abruptness of change, the latter seemed to play a more important role on the results obtained than the former. This finding suggests both a strength and a possible shortcoming of the proposed algorithm. While the results reflect a sensitivity to smaller but abrupt changes, they also indicate a certain difficulty in detecting gradual changes.

	Recall (Mean±Std)		Recall (Mean±Std)
Mean = 10 Abr = 1	0.89 ± 0.09	Mean = 2 Abr = 1	0.70 ± 0.19
Mean = 10 Abr = 2	0.82 ± 0.11	Mean = 2 Abr = 2	0.67 ± 0.20
Mean = 10 Abr = 5	0.77 ± 0.14	Mean = 2 Abr = 5	0.63 ± 0.28
Mean = 10 Abr = 20	0.72 ± 0.21	Mean = 2 Abr = 20	0.60 ± 0.33
Mean = 5 Abr = 1	0.83 ± 0.31	Mean = 1 Abr = 1	0.56 ± 0.36
Mean = 5 Abr = 2	0.75 ± 0.14	Mean = 1 Abr = 2	0.52 ± 0.38
Mean = 5 Abr = 5	0.72 ± 0.19	Mean = 1 Abr = 5	0.49 ± 0.38
Mean = 5 Abr = 20	0.69 ± 0.23	Mean = 1 Abr = 20	0.46 ± 0.41

Table 4.5: Recall score obtained on artificial data with known change points

Furthermore, the results obtained through this first proposed experiment allow us to formulate hypotheses regarding the expected performance of the algorithm, given the characteristics of the data it is applied to. In the next proposed experiment, where we will test BOCPD on real-world data with known change, we will present the characteristics of the data used in terms of abruptness and magnitude of change, and analyze the results obtained in reference to one of the categories -easy, medium, difficult- introduce in Section 4.2.1.

Effective Magnitude of Change and Abruptness of Change

In order to allow for an explicit comparison between the characteristics of the data sequence used for this experiment on artificial data and the ones using real-world data introduced in the following chapters, two metrics are calculated: Effective Magnitude of Change and Effective Abruptness of Change.
Effective Magnitude of Change is defined as the number of standard deviations the mean shifted from before to after the known or detected change point. Effective Abruptness of Change is calculated as the distance, in terms of data points, between the last data point before the known or detected change to fall within one standard deviation from the mean calculated before the change point and the first data point after the known or detected change to fall within one standard deviation from the mean calculated after the change point. In table 4.6 below, Effective Magnitude of Change and Effective Abruptness of Change are calculated for each of the possible combinations of *(magnitude of change, abruptness of change)* introduced in Table 4.4. The values presented reflect the average value of the two introduced metrics for 100 data sequences produced for each combination.

Mean Shift	Abruptness of Change	Effective Magnitude of Change	Effective Abruptness of Change
10	1	7.48	1.14
	2	8.07	1.92
	5	8.41	4.41
	20	8.28	17.36
5	1	4.44	1.30
	2	4.07	1.85
	5	4.77	4.23
	20	4.35	18.07
2	1	1.76	1.40
	2	1.75	1.93
	5	1.68	4.42
	20	1.64	16.46
1	1	0.74	1.37
	2	0.79	1.78
	5	0.92	4.07
	20	0.86	17.23

Table 4.6: Effective Magnitude of Change and Effective Abruptness of Change

Finally, we close this chapter by explicitly addressing the research question introduced in the beginning of it:

> How does Bayesian Online Change Point Detection perform on artificial data with known change points?

Based on the results obtained on the proposed benchmark, we argue that our adaptation of Bayesian Online Change Point Detection performs well on artificial data with known change points. In particular, optimal results can be observed for sudden and significant changes, representing the easiest scenarios in our benchmark. Satisfactory results are also observed for increasingly more gradual and smaller changes, covering the majority of the easy-to-difficult spectrum introduced by the proposed benchmark. Sub-optimal results are only registered for few cases representing the most difficult scenarios.

Chapter 5

Experiment on Known Change Points in Real World Data

In this chapter, we aim to respond to the following research question:

> Can Bayesian Online Change Point Detection detect known change points in real world sports and e-sports data?

In order to do so, we apply our algorithm to real world data from sports and e-sports where a general consensus was formed amongst insiders about the existence of a clearly identifiable change point, and evaluate the ability of BOCPD to detect such change. In the following sections, for each case study, we introduce the data used, giving context to the aforementioned agreed-upon change point, and present the result attained on such data. Finally, based on the results obtained, we deem the algorithm as suitable for application on both sports and e-sports data.

5.1 Sports Real World Data with Known Change Points

5.1.1 General Setting

Before presenting each individual case study, let us first introduce the general setting, common to all of them, that will be used for their discussion. We will introduce each case by summarizing the context in which the known change point occurred. In particular, the point in time in which such change occurred is used for our experiments as a cut-off point: we disregard any possible change point detected before it. After this introduction, we will present the results obtained. In case a change point is discovered, we will be interested in two aspects: the level of confidence (i.e. the posterior probability value) with which the change point is discovered, and the Run Length values associated to it, to assess any possible delay in the discovery.

5.1.2 Javelin Throw

The Olympic discipline of javelin throw underwent a pretty significant revolution in 1986, as the governing body (known at the time as the "International Amateur Athletic Federation", and called nowadays "World Athletics") decided to introduce a new design for the javelins used in competitions, as the previous one resulted in a lot of so-called "flat landings", i.e. the javelin would land and slide on the ground as opposed to piercing it, and because throws were consistently surpassing 100 meters of distance, posing a potential risk to the spectating crowd. This caused the world record, which was a mark of 104.80 meters registered by German track and field athlete Uwe Hohn in July 1984, to drop quite significantly, as the new post-regulations world record was set at 85.74 meters by German athlete Klaus Tafelmeier in 1986.

In order to assess whether our implementation of BOCPD is able to detected such change, we collected data of the Men's javelin throw world record progression, starting from year 1912. Let us present the results obtained using the visual output introduced and discussed in Figure 4.1.

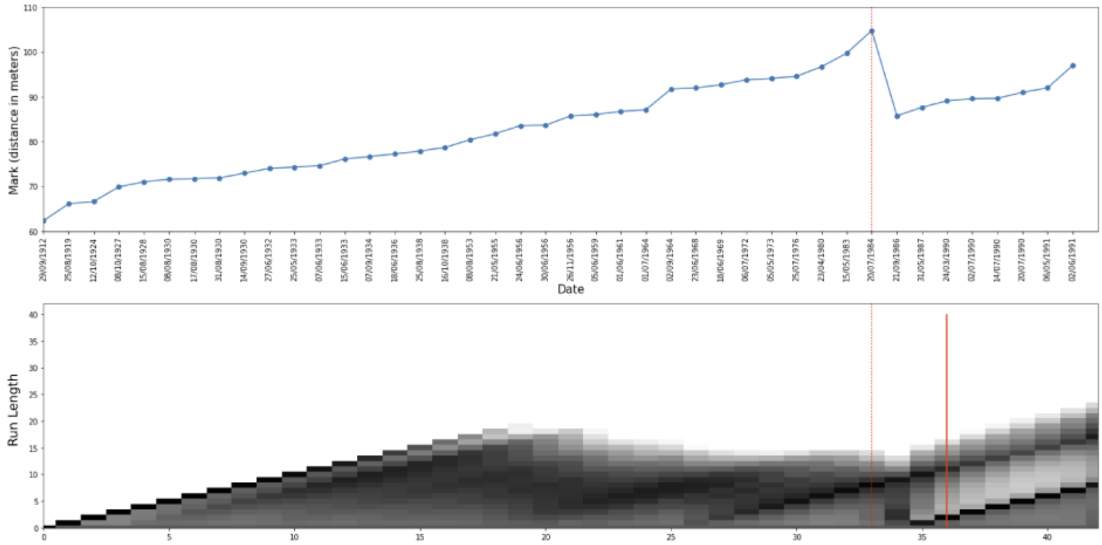


Figure 5.1: Results of the application of BOCPD on Javelin Throw Data

In the top portion of Figure 5.1, a plot reports the data and a dotted red line indicates the year 1986 cut-off point discussed above, coinciding with the implementation of the new design regulations. A caveat is to be made here about the fact that time on the x-axis is not continuous, as the different dates are snapshots of days in which the world record was broken. In the bottom portion, the run length posterior is plotted, and a solid red line indicates the first detected change point after the cut-off. In terms of the characteristics of the data used, which are important to report as they allows us to draw a comparison between the results obtained here and the ones presented in the previous chapter on artificial data, we can see how the magnitude of change is significant (from 104.8 meters to 85.74) and immediate. Therefore, to use the same classification introduced for the various settings identified for the experiment on artificial data, this setting would fall into the 'easy' category, given that both its Effective Magnitude of Change and Effective Abruptness of Change are 1.

Results obtained are aligned with what has been reported for such category as, after just 2 data points, BOCPD has already detected the change point. Furthermore, the intensity of the black color indicates a high degree of certainty about the presence of a change point, equal to 0.75, and the highest Run Length value associated to it, 2, correctly identifies the origin of the change.

Such optimal results are to be expected, given the nature of the setting and the fact that the new regulations were put in place specifically to have a significant impact on the discipline. Let us now discuss less obvious scenarios.

5.1.3 Speed Skating

Similar to javelin throw, the world of speed skating was also revolutionized by a change in the tooling used. Contrary to javelin throw, however, speed skating benefited from the change. This happened with the introduction of Dutch-patented clap skates by Gerrit Jan van Ingen Schenau [23]. This new type of skates was characterized by the fact that the blade was not attached to the full length of the boot anymore, but held in place by a hinge at the front, allowing the blade to remain in contact with the ice longer and athletes to fully extend their ankle at the end of a stroke. Despite not being immediately adopted by elite level athletes, as it took 10 years after the original patent to see clap skates becoming the predominant athletes' choice, this invention eventually had a profound impact on the sport. In the 1998 Nagano (Japan) Winter Olympics, in fact, across 10 main events -500m, 1'000m, 1'500m, 5'000m, and 10'000m for both men and women- nine previous world records set with traditional skates were broken. Furthermore, according to data from the International Skating Union [24], before the introduction of clap skates, it took 16 years -from Eric Heiden's 1:13.60 in 1980 to Manabu Horii's 1:11.67 in 1996- to bring the world record down by two seconds in the men's 1'000m race. The same result was accomplished, after the adoption of clap skates, in just two years.
In order to test our implementation of BOCPD, we collected data of the 1'000m Men's Speed Skating world record progression, starting from year 1900. Let us report the results of the experiment.

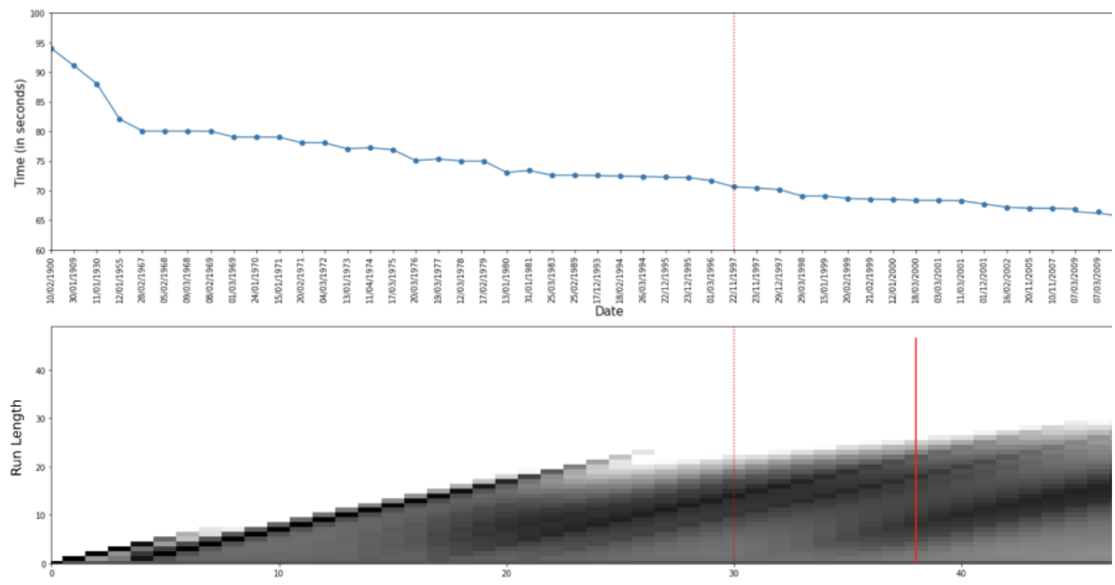


Figure 5.2: Results of the application of BOCPD
on Speed Skating Data

On the top portion of the graph, we have our data plotted and a dotted red line representing the cut-off point discussed above, coinciding with the large adoption of the new clap skate design starting in year 1997 season. The same caveat made for javelin throw data has to be made here, as time on the x-axis is not continuous. On the bottom, the Run Length Posterior is represented in the usual manner and a solid red line represents the first detected change point after the cut-off. As we can see from the scatter plot on top, by looking at the world record progression alone without any help from our algorithm, it would be quite difficult to hypothesise the existence of

any change point. However, our implementation of BOCPD is able to detect a change point. As we can see in the bottom portion of the image, in fact, a change point is detected at x_{38} with the maximum probability value being observed for run length value 8, which indicates the presence a change point at x_{30}. As we can expect, given the relative complexity of detecting the impact of this new technological introduction by just looking at the world record progression, where the difference between values are relatively small, the probability value for run length 8 at x_{38} is not high (0.23). However, as we can see from the increasing intensity of black in the run length posterior probability associated to the detected change point, the algorithm is able to build more and more confidence as more data after x_{38} become available. Considering the nature of the data used for this experiment, where the magnitude of change following the change point is small (few tenths of a second for each new registered value after the change point, yielding a calculated Effective Magnitude of Change of 1.77), this setting could be compared to the ones identified by the 'hard' category in the experiments presented in the previous chapter. Nonetheless, results obtained are promising, and seem to suggest that the algorithm is sensitive enough to detected small changes.
While the results attained for this use case are promising, it is important to note that we are still far from recreating a setting in which our implementation of BOCPD could provide valuable insights. By looking at the world record progression, in fact, we would have only detected the change point after 3 years -x_{38} corresponds to March 2000- since the first world record set using clap skates. By that time, every insider in the world of speed skating had realized the impact and benefit of clap skates. For this reason, in the following section, we will test our algorithm on data collected with a daily frequency.

5.1.4 Basketball

As we mentioned in the introduction of this thesis, NBA basketball has underwent, in the last 20 years, a profound revolution, known to insiders as the 'Three Point Revolution'. This shift in the way the game is played is, however, not the first one happening in the highest basketball league. Back in the 1994, in fact, a decision to move the three point line closer by one foot (from 23 down to 22 feet) was taken by the NBA governing body in order to counter a negative trend in points scored per game. This change lead coaches to instruct their players to take more 3 point shots, which in turn reflected in a significant increase in the statistics for 3 point shots attempted per game: in the season before the introduction of such change, 1993-94, the league-wide average was in fact 9.9, while the three seasons following the change - which would eventually be reverted in season 1997-1998- registered values of 15.3, 16.0, and 16.8. To assess whether our implementation of BOCPD could pick up this change, and how rapidly it would accomplish so, we gathered data about 3 point shots attempted per game for every game day, aggregating all league games played on that day, for seasons 1992-93 to 1994-1995. To ensure that each season had the same number of games played (82 per team), we only considered the regular season and excluded play-offs.
On the top portion of the graph, we have our data plotted and a dotted red line representing the cut-off point discussed above, coinciding with the start of 1994-95 season, the first one seeing the introduction of the 22-feet three point line. On the bottom portion, the run length posterior and a solid red line representing the first detected change point after the cut-off. It is important to note here how the figure, due to an increased number of data points, and consequently more pixels representing Run Length values, looks smoother than the saw-toothed shapes observed for experiments with smaller data sets. The change point is detected at x_{401}, with the highest probability value being observed for a run length value corresponding to 55, which indicates a change point at x_{346}, just 3 game days after the beginning of the season. As for the degree of certainty with which the change point is detected, the probability value for a run length of 55

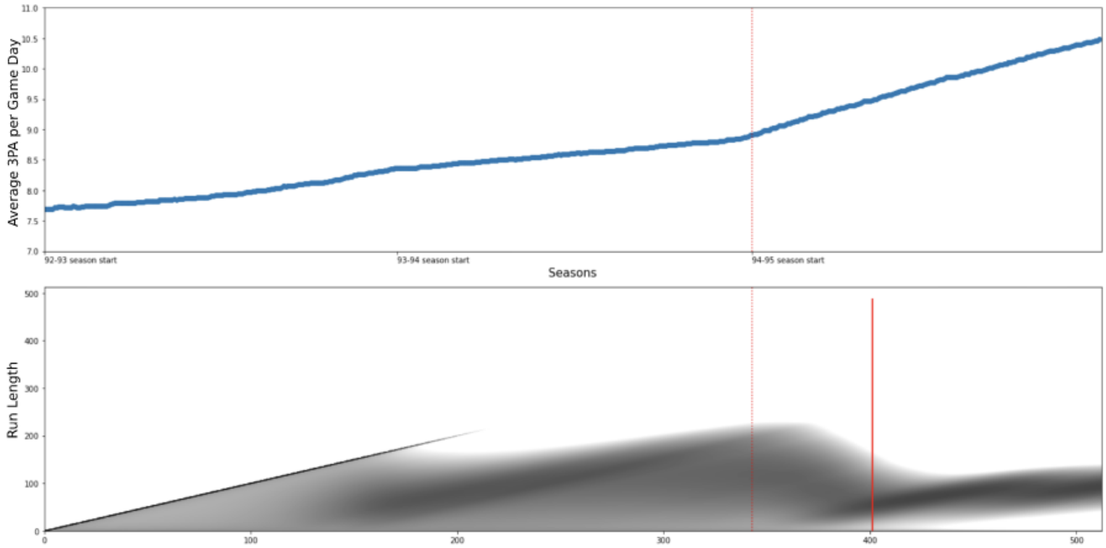


Figure 5.3: Results of the application of BOCPD on NBA Basketball Data

at x_{401} is 0.30. Despite this not being reflective of a high degree of certainty, we can see in the posterior probability plot in the bottom portion of the graph how such certainty level increases as more data points become available. If we were, as we did for the previous two use cases, to classify this setting with the same metrics used for the experiments on artificial data in the previous chapter, this would fall into the 'realistic' category (Effective Magnitude of Change 2.24 and a significant Effective Abruptness of Change of 73), as change in mean and abruptness parameters for such a setting were derived from the ones in this distribution. For this reason, results obtained here are quite remarkable, considering that the change is detected within the first three game days. If we return to the hypothetical scenario drawn in Section 5.1.1, this would mean that by using BOCPD to monitor 3 point shots attempts per game, an analyst working for a team competing in the league could have detected the ongoing trend very early on in the season and, for instance, advised their coach to work on possible defensive solutions to counter this new strategy.

5.1.5 Discussion

In this chapter, we tested our adaptation of BOCPD on real-world data with known change points, i.e. points in time where a general consensus was formed amongst insiders about the existence of a clearly identifiable change. Regarding sports, we used data from three different disciplines. We first analyzed the world record progression for javelin throw, where the change point was introduced by a new design for the javelins imposed by the governing body of the discipline. The results obtained reflect an optimal performance, which is aligned with our hypothesis formulated by referencing back to the difficulty of the change point detection task introduced in Section 4.2.1 and deeming this as an 'easy' task. Secondly, we analyzed the world record progression for speed skating, where a change point is identified by insiders with the introduction of the new clap skate design. This second use case also showed good results, although less optimal in terms of the timing of the discovery compared to the previous. This is in line with our hypothesized result based on the difficulty of the change point detection task, which falls under the 'hard' category.

Lastly, we introduced a use case where the algorithm was tested on data regarding 3-point shots attempt in NBA basketball following the adjustment to the distance of the 3-point line, which was brought closer to the basket in seasons 1994-95 to 1996-97. This experiment represent the hardest and most realistic scenario of the three, given that it consider daily collected data, with more gradual changes between consecutive data points. Nevertheless, the results obtained for it are also good. This suggests the suitability of our adaptation of BOCPD for the application to sports data. A possible shortcoming observed through the three use cases is the degree of certainty with which change points are detected. In the case of NBA basketball data, for instance, the change is in fact only detected with a posterior probability value of 0.30. In a real-life context in which the algorithm is actually deployed to notify its users about possible change points, this might be problematic as it could cause a lack of confidence towards the output presented. It is also important to note, however, how the degree of certainty with which a change of point is detected increases as more data points are analyzed.
In concluding this first section about real world sports data with known change points, let us answer the research question introduce in the beginning of this chapter:

> Can Bayesian Online Change Point Detection detect known change points in real world sports data?

Across the three use cases presented for the application of BOCPD to sports data, characterized by an increasing level of complexity for the change point detection task, very good results have been observed, suggesting the possibility of successfully implementing BOCPD in the context of sports in order to detect significant changes. It is important to note here how these use cases, while covering three different disciplines and various levels of difficulty of the change point detection task, only represent a small subset of the much larger possibilities and scenario of application. Further testing will be needed before it is possible to draw conclusions about the effectiveness of BOCPD on sports data.
Having tested and analyzed the performance of our adaptation of BOCPD on sports data, we are now interested in testing its efficacy on e-sports data. In particular, we will use for our use cases data from the videogame 'Call of Duty: Warzone'.

5.2 E-Sports Real World Data with Known Change Points

In testing our algorithm on e-sports data, we propose a different angle compared to the previous section. As we will discuss in following sections, in fact, we want to consider the point of view of game developers, rather than game users, which would have been the equivalent of coaches and players from the sports world. If for sports we focused on cases where change point detection would help in reacting to changes in rules and/or tactics, for e-sports we propose an hypothetical scenario in which change point detection would be used by game developers in two different scenarios: firstly, to monitor the state of the game and its *meta*, and secondly to make sure that the game patches released have the desired impact on the game. Let us introduce our use cases, based on data from videogame 'Call of Duty: Warzone' (hereinafter, Warzone) (Activison Publishing, March 2020).

5.2.1 Call of Duty: Warzone

Warzone is a free-to-play Battle Royale title released by software house Activision in March 2020, widely considered one of the most successful in its genre, counting more than 100 million active players worldwide as of April 2021 [25]. Warzone offers different game modes, characterized by

different objectives and team sizes, the main one -which we will be focusing on for the purpose of our proposed experiments- being called 'Warzone Battle Royale'. This mode sees players, organized in teams spanning from just one person (Battle Royale Solo) to four people (Battle Royale Quads), fighting for survival in a large scale map, which is reduced in size as the game progresses by an ever-shrinking 'safe zone'. This mechanism, quite popular in titles belonging to the Battle Royale genre (for instance, Apex Legends or Fortnite), pushes players closer and closer together, until just one (team) is left and is crowned as the match winner.
One of the novel features introduced to the Battle Royale genre by Warzone is the ability for players to use customized weapons (hereinafter, loadouts), where customized indicates equipped with a series of attachments (for instance, optics or bigger maganizes) that are unlocked by the player as they accumulate experience with that particular weapon and progress it through different weapon levels. The choice of weaponry in previous Battle Royale games was based solely on a *looting* mechanism, where players would have to explore areas of the map to collect resources and find weapons. Warzone decided instead to allow players to be able to use their own loadouts, acquired during a game with mechanisms beyond our interest. From the developers' point of view, this means not being able to explicitly control the variety and distribution of weapons across the battle field. As we mentioned in the Introduction 1, this fostered the creation of a *meta*: a limited number of weapons which, thanks to favorable individual characteristics, are picked by the majority of players. Monitoring and controlling the state of the *meta* is of crucial importance for game developers: if they allow it to become too closed - i.e. just one or two weapon out of the more than 150 included in the game being picked by the majority of players -, the game can quickly become boring to play for the lack of diversity. If it becomes to unfair - i.e. a weapon becomes, for instance as the results of a mistake in a game patch balancing changes, too powerful -, the game can quickly became frustrating to play. For this reason, monitoring the pick rate of each weapon available in the game can give valuable insights regarding the state of the meta. In order to monitor this in an automated fashion, game developers could use BOCPD. Let us then introduce our first use case and experiment.

5.2.2 Pick Rate Monitor

To test the suitability of BOCPD for the purpose of detecting significant changes in the popularity of a particular weapon, we present the use case of a weapon becoming too popular and dominant as a result of a judgment error in a balancing patch.
As part of a patch released on the 16th of June 2021 game developers decided to improve the statistics of a weapon known in game with the name 'Milano' with the intention of giving it more popularity, as its pick was substantially non-existing before the patch release. The introduced changes, however, had a much more significant effect than the one envisioned: the 'Milano' became the de-facto *meta*, with a pick rate of around 20% (very significant if the total number of weapons -more than 150- available in game is considered), benefiting from new well-above-average statistics. Following these changes, players had to wait for almost a month before seeing this issue -the weapon was largely considered unfair by the community- addressed by the game developers with a new patch, which would eventually be released on the 14th of July. Let us now assume, for the purpose of this experiment and perhaps wrongly, that game developers always act in good faith, aiming to offer the best and most balanced experience to their users. If this assumption holds, than the only explanation for such a situation would be that it took game developers around a month to realize the lack of balance introduced by the patch released in June. By analyzing the pick rate data for the 'Milano' with our implementation of BOCPD, we are therefore interested in finding out if our algorithm would have made the developers aware of such imbalance in a shorter time. In order to prepare for the experiments proposed in this section, we collected daily pick

rate statistics for each weapon included in the game, since release (or since its addition, in case of weapons only introduced at a later stage). Let us present the results obtained for this experiment in Figure 5.4.

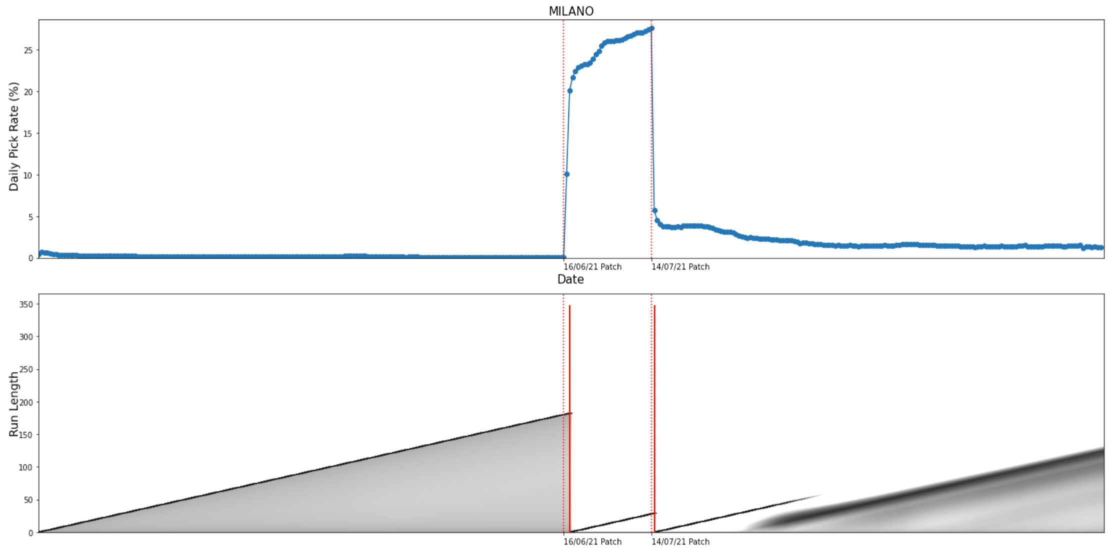


Figure 5.4: Results of the application of BOCPD
on Call of Duty: Warzone data

As we can see in the graph, where the usual notation discussed for sport in the previous section is used, the change is detected immediately by our algorithm. Naturally, it can be argued that such a shift is so significant and sudden that no assisting technology would be needed to detect it (which is confirmed by the two calculated metrics of Effective Magnitude of Change equal to 23.47 and Effective Abruptness of Change of 1), but if we imagine a scenario in which our implementation of BOCPD is deployed by the game developers in order to monitor the pick rate of all 150 weapons included in the game, and we also consider the delay with which game developers addressed this issue, then the ability of BOCPD to promptly identify changes could be leveraged to implement a system that sends automated notifications informing developers of possible detected anomalies.
As we discussed in the introduction of this chapter, BOCPD could be also used for another purpose: making sure that game patches periodically released by game developers have the desidered impact on the game.

5.2.3 Game Patches Effect Monitor

To better understand how developers might want to steer the *meta* in a particular direction through the release of game patches, it is important to introduce two consideration: (1) Warzone inside the Call of Duty ecosystem, as Warzone is coexisting with other annually-released regular Call of Duty games, and (2) the weapon leveling mechanism.
(1) Warzone inside the Call of Duty ecosystem. Before free-to-play Warzone first came out, Activision used to release one paid-for new game per year. Each new game would be characterized by their own universe, and hence their own set of weapons, and be completely independent from the release prior to it. While this is still true, and Activision still release one paid-for new game

each year, Warzone became an overarching title to the ecosystem, enriched every year by the addition of weapons from the regular release title of that year. When it first came out, in fact, Warzone only included weapons from the latest released title 'Call of Duty: Modern Warfare' (released in October 2019). In December 2020, however, following the release of the new 'Call of Duty: Black Ops - Cold War' (released in October 2020), developers decide to integrate weapons from their latest release into Warzone. This happened again the following year, as December 2021 saw the addition to Warzone of weapons from 'Call of Duty: Vanguard' (released in October 2021).

(2) Weapon leveling mechanism. As we discussed, Call of Duty players are able to customize and improve the performance of their weapons by unlocking a series of attachments as they move through different, and incrementally longer in terms of the Weapon Experience requested to level up, Weapon Levels. Weapon Experience is awarded as players complete different game objectives and get eliminations of other players with it. For game-specific dynamics that are beyond the scope of this thesis, this is more easily achieved in regular paid-for releases of Call of Duty, rather than in Warzone. This aspect in particular will be pivotal to the discussion that follows.

Having discussed these two aspects, we can now understand the developers' point of view better: since weapon leveling is accomplished more easily in regular paid-for Call of Duty releases than inside of free-to-play Warzone, having a *meta* that includes weapons from the latest release is economically advantegeous, as it will encourage players to level up such weapons, and hence incentivise them to buy such latest release. Consequently, it is in the interest of game developers to introduce balancing patches to the game in order to shift the *meta* towards weapons from the most recent release. In this context, BOCPD can be used to automatically monitor the state of the *meta* by tracking weapons' pick rate, and ensure that the balancing patches deployed are yielding the desired -i.e. shifting the *meta* toward weapons from the latest release - effect.

Meta-shifting Patches

For this experiment, we will focus on the game patch released on June 16th, 2021, as this patch represented a turning point for the game: up until that point, in fact, despite the integration of weapons from the 'Black Ops - Cold War' happened in December of the previous year, weapons from this newer release struggled to enter the *meta*, for reasons beyond our interest. After the release of this patch, instead, the *meta* shifted significantly, and became dominated by 'Black Ops - Cold War' weapons. As an example, we will be present results obtained by running BOCPD on the daily pick rates of two weapons:

- an assault rifle from 'Call of Duty: Modern Warfare' known in game with the name 'CR-56 Amax', which was dominant in the *meta* up until the patch release
- an assault rifle from 'Call of Duty: Black Ops - Cold War' known in game with the name 'FARA', which became dominant in the *meta* after the patch release

Let us present the results obtained in Figure 5.5.

As we can see, not only the change points are identified correctly, quickly -the change points are picked up withing 2 days in both cases-, and with a high degree of certainty, but we can also see how the patch released on June 16th succesfully shifted the meta towards a weapon belonging to the newer release. This confirms the possibility of adopting our adaptation of BOCPD in order to monitor the effectiveness of patches released.

While running these experiments, we almost coincidentally gathered another -perhaps even more interesting- insight: if we focus on the results obtained for the 'FARA', on the bottom portion of Figure 5.5, we can notice a sudden drop in its popularity following the increase result of June 16th patch. This reflects the effects of another patch, released on July 14th 2021. In this patch,

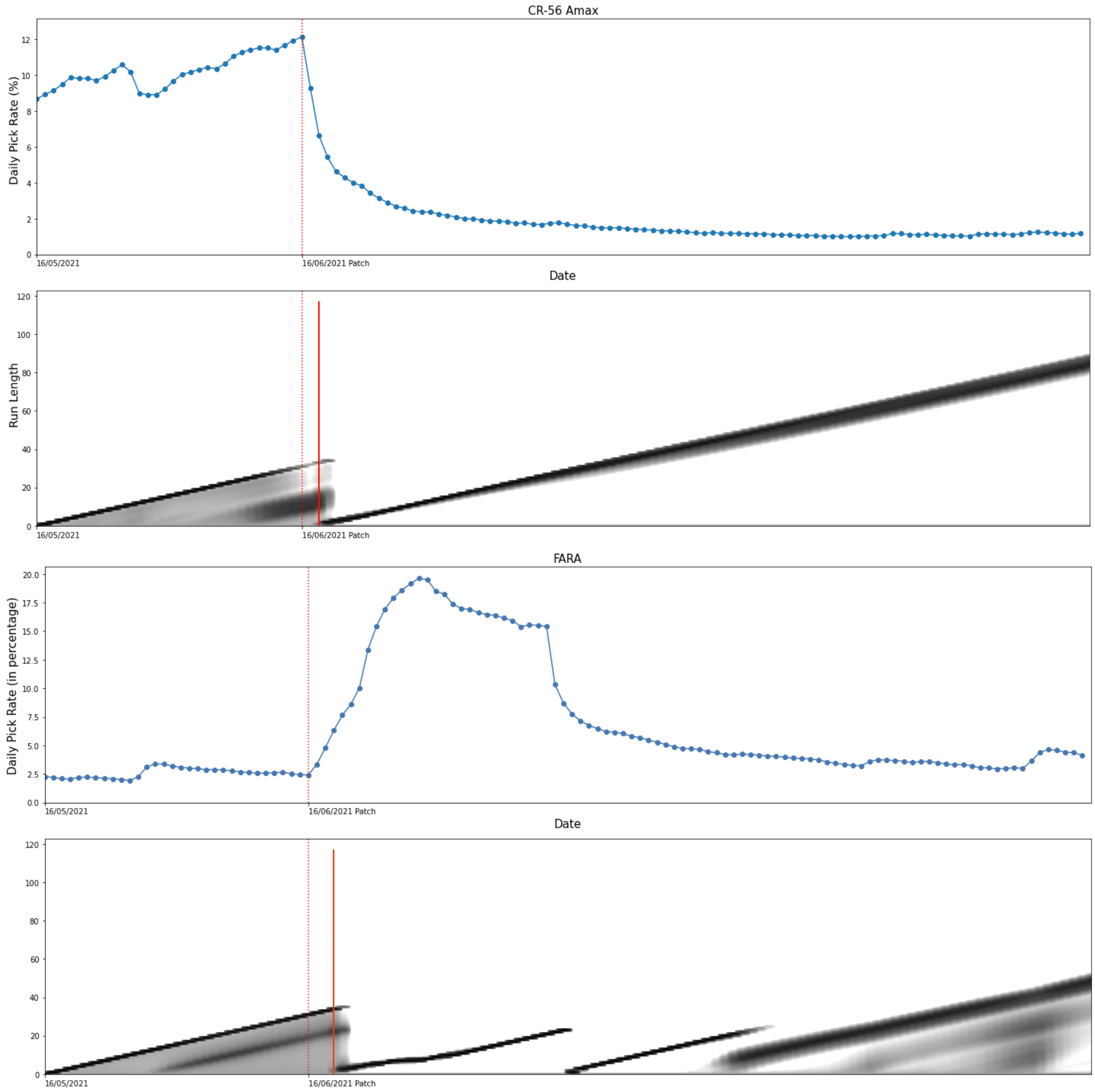


Figure 5.5: Results of the application of BOCPD
on Call of Duty: Warzone data

developers decided to *nerf* the ‘FARA’ -i.e. negatively adjust the weapon statistics- in order to prevent it from becoming a forced choice for players. This could happen, in fact, in the event that a single weapon grants competitive advantage so significant to funnel all players into using it. This is known in videogames as a *closed meta*.

After observing this, we decided to run another experiment, considering the ‘FARA’ again but this time in comparison to another assault rifle from ‘Call of Duty: Black Ops - Cold War’, the ‘Krig 6’, to see if the *nerf* to the former resulted in an increased popularity for the latter, and if our algorithm could also detect this change.

As we can see from the results presented in Figure 5.6, changes to the pick rate introduced by the patch released on July 14th were correctly identified as well, confirming the optimal results

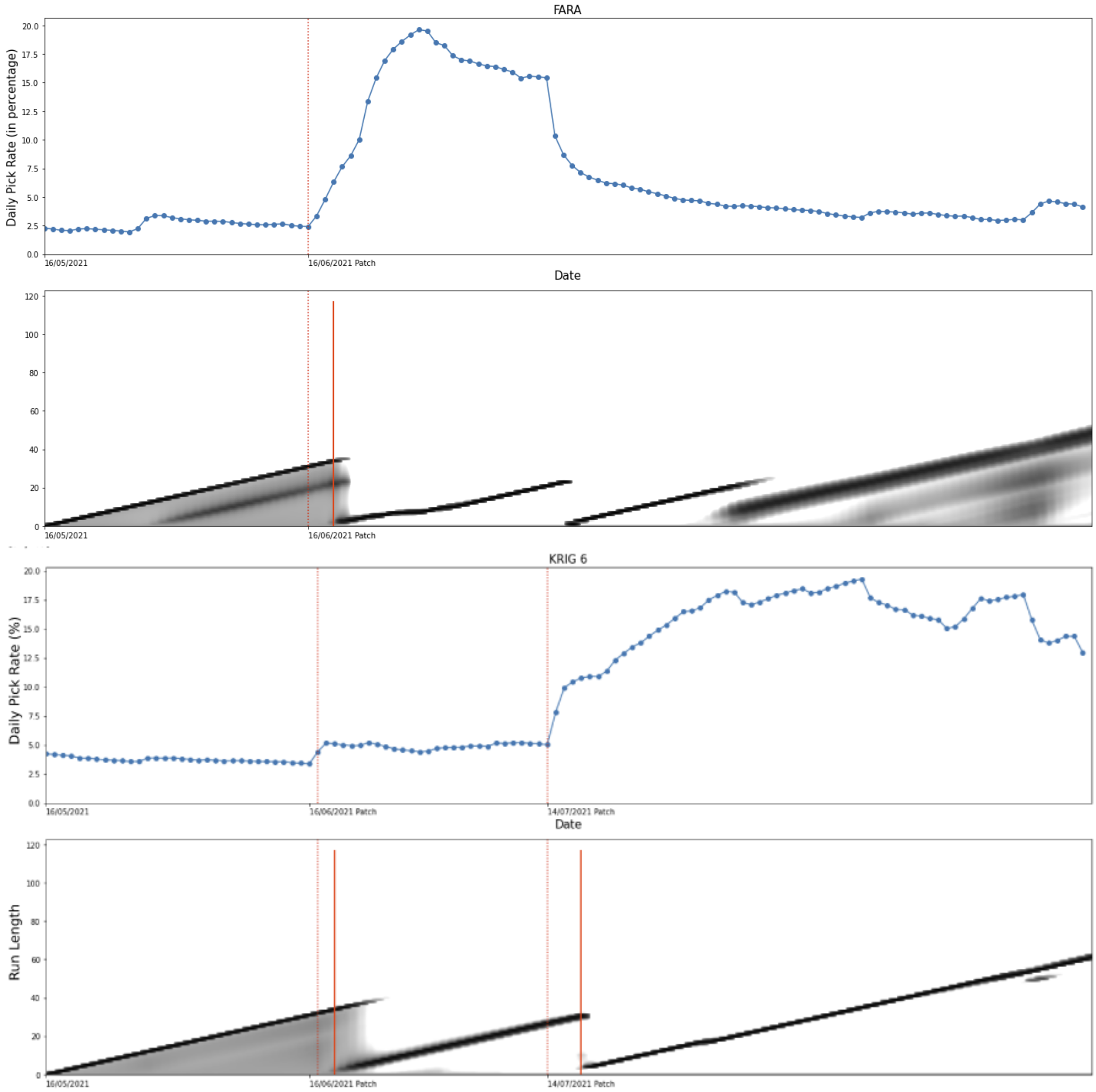


Figure 5.6: Results of the application of BOCPD
on Call of Duty: Warzone data

already observed for the first experiment on Warzone data regarding the pick rate of the weapon called 'MILANO'.

5.3 Discussion

In this section about the application of BOCPD to e-sports data, we presented two different use cases, where the algorithm is tested on data reflecting the pick rate of particular weapon inside first-person-shooter videogame 'Call of Duty: Warzone'. First, we considered a case where a weapon, as a result of a mistake from the developers, benefited from significantly above-the-average statistics and suddenly became the dominant (META, as introduced in Section 5.2.1) choice of the majority of players. Results obtained for this use case are optimal, as the change was detected quickly and with a high degree of certainty. This is aligned to the hypothesized performance, given that the characteristics of the data are aligned to the ones presented for the 'easy' category introduced in Section 4.2.1. Then, we presented a second use case, in which the algorithm is applied in parallel to two different weapons, with the intention to assess whether a decrease in popularity for one weapon would correspond to the increase in popularity for another one. Optimal results are observed for this use case as well. This suggests that the parallel application of the algorithm to different time series could be used to investigate possible cause-effect mechanisms. We conclude this chapter by addressing the research question introduced in the beginning of this chapter:

> Can Bayesian Online Change Point Detection detect known change points in real world e-sports data?

In light of the results obtained, we confirm the suitability of BOCPD for applications on e-sports data. In particular, the optimal results suggest the possibility for game developers of implementing BOCPD in their game balance monitoring systems.
In the following and final chapter, we aim to replicate the work conducted by Adams and MacKay in the original paper [15], of which an example has already been discussed with Figure 4.1: we present two use cases, one for sports and one for e-sports data, where BOCPD is applied to unlabeled data, where the existence, and eventual location, of change points is not known. Based on the output observed, and given the existence of any identified change point, we then consider possible reasons that might have caused it.

Chapter 6

Experiment on Unknown Change Points in Real World Data

In this chapter, we aim to replicate the work proposed by Adams and MacKay in the original paper [15], testing BOCPD on unlabeled data from sports and e-sports, where the existence, and the eventual location, of any change point is not guaranteed. This will allow us to finally respond to the research question opening this thesis work:

> "Can changes in dominant strategies in sports and e-sports be detected using CPD algorithms?"

Let us present the two separate experiments in the following dedicated sections.

6.1 Unknown Change Points in Real World Sports Data

In order to test BOCPD on unknown change points in real world sports data, we will be using data about 3-point shot attempts for the past 20 NBA season. As already introduced in Section 5.1.4, this data report the average number of 3-point shots attempts for each game day of the season in which at least one League game one played, for the past 20 season (starting from season 2000/01 up to season 2019/2020). In analyzing the output of BOCPD on such data, we are interested in seeing whether we can find any clear indication of the start of the so-called '3-point shot revolution' discussed in the introduction of this thesis. Furthermore, if the output of BOCPD indicates the presence of other possible change points, we are interested in assessing whether we can find a possible event that introduced such change points. Let us analyze the results obtained, reported in Figure 6.1.

While the structure of the graph, with the data plotted in the top portion and the posterior run length probability R plotted in the bottom portion, is the one used throughout this thesis, there is one important distinction to make: the solid red lines in the bottom portion of the graph don't represent change points detected in an automated manner by BOCPD. They indicate instead change points that we were able to hypothesize the existence of based on the visual representation of R.

At first sight, the results obtained might look discouraging: the only change point which we were able to clearly hypothesize the existence of is in fact the one denoted with the solid red line shortly after the start of 2007/2008. One additional change point might also be inferred from R somewhere in the second half of season 2002/2003, while the rest of the visual representation of the posterior run length variable suggest the absence of any other clear change point. However, there are some important considerations to be made here. First of all, while it is easy to recognize such a significant shift in the way NBA basketball is played by looking at the full extent of data available, we must keep in mind that BOCPD is an on-line change point detection method, which means that at any point, while analyzing the data, it only has access to data points seen so far. This renders detecting a change happening across 20 seasons (more than 3200 game days) a much harder task. Secondly, while it is difficult to estimate at which point this on-going 'revolution' was first identified by insiders to NBA basketball, a quick online research suggest that the first articles about the strategical importance of 3 point shots attempts are from year 2009. Lastly, the calculated values for Effective Abruptness of Change report significant values for both detected change points, 56 for the first one and 230 for the second one. This confirms the complexity of the task.

Having made these important considerations, let us go back to the change points that we were able to infer from BOCPD output presented in Figure 6.1, and in particular to the one at the beginning of season 2007/08. As we can see, the season was characterized by a sudden increase in the average number of 3-point shots attempts per game day. This might not be casual: in the previous season (2006/07), the NBA elected German athlete Dirk Nowitzki of the Dallas Mavericks as the Most Valuable Player (M.V.P.) of the year. Nowitzki, largely considered one of the best 3-point shooters in the history of NBA basketball - sitting at position 13 in the all-time standing for scored 3-point shots - made of 3-pointers one his best and most recognizable offensive weapons. This could explain, at least to some extent, the increase in 3-point shots attempts in the following season: players might have tried to mimic Nowitzki's way of playing, hoping to see similar success. Furthermore, in the 2006/07 season, Nowitzki also won the 3 NBA all-star 3-point contest, which might have further fueled this phenomenon.

As for seasons following the 2007/08, the output of BOCPD does not seem to suggest the presence of another change point. However, there might be something interesting to be observed here: if we analyze the maximum run length values observed, we can see a decreasing trend, which seems to be now converging to values close to 200. This could be indicative, given the fact that the average number of game days in a season is around 200, of the inability of BOCPD to surpass the maximum run length obtainable by considering one season, and therefore be reflective of a possible upcoming plateau for this on-going trend.

Results

To conclude this section about unknown change points in real world sports data, let us address the research question opening the chapter:

> "Is it possible to apply CPD algorithms to sports data in order to detect significant changes in their mechanics?"

While the results obtained for unknown change points in real world sports data are not optimal, the output of BOCPD still allowed us to identify a change point in the beginning of season 2007/08. Given the complexity of the task of detecting a change point in such a gradual and extended in time shift, this can still be regarded as a good result.

6.2 Unknown Change Points in Real World E-Sports Data

In order to test BOCPD on unknown change points in real world e-sports data, we are using data from Warzone again. In particular, we are considering the daily pick rate of the weapon known in game as 'CR-56 AMAX', which we have already analyzed for the experiment reported in Figure

5.5. This time, however, we are interested in looking at the entirety of the data available, and not only at days preceding and following a patch release. In doing so, we are interested in assessing whether it possible to find a motivation for the existence and location of the eventual change points detected by BOCPD, as this would represent the testing scenario closest to a real-world application of the algorithm, where no a-priori information about significant events (for instance, a patch release) is available. Let us present the results obtained for this final experiment in Figure 6.2.

While the structure of the graph, with the data plotted in the top portion and the posterior run length probability R plotted in the bottom portion, is the one used throughout this thesis, there is one important distinction to make: the solid red lines in the bottom portion of the graph don't represent change points detected in an automated manner by BOCPD. They indicate instead change points that we were able to hypothesize the existence of based on the visual representation of R. For such possible change points, we reported on the top portion of the graph the corresponding date. In the bottom portion, instead, events that could have influenced the popularity of the analyzed weapon and happened in the identified dates are reported. Let us discuss those.

In order to better understand the hypothesized change points, it is important to briefly introduce the 'Season' mechanism present in Warzone. In an effort to offer new content to their players every so often, and keep them interested in playing the game, game developers have in fact opted to introduce a new Season every few months. Each Season is characterized by a new portion of the lore behind Warzone being presented, some new in-game content and, most importantly, a weapon balancing patch. For this reason, the beginning of each Season usually corresponds to a new set of weapon rising in popularity.

Having understood this is crucial to interpret many of the change points that we inferred from the visual representation of R in Figure 6.2: 3 out of 5 (marked as "Season 1", "Season 1 R" for "Season 1 Reloaded", and "Season 3" in the bottom portion of the graph), in fact, correspond to the start of a new Season, and hence justify the variation in popularity observed. The date of the first hypothesized change point (marked as "?" in the graph), instead, does not correspond to any major event in Warzone, and is therefore difficult to justify. The rise in popularity observed could just represent the result of an organic process in which players in the community influenced each others' choice of weapon. The date of the last hypothesized change point, instead, correspond to the Patch release on June 16th, 2021, which we have already discussed in the experiment reported in Figure 5.5. For completeness, we report here the values reported for Effective Magnitude of Change and Effective Abruptness of Change for the 5 aforementioned change points: 2.67 and 2 for the first one, 0.56 and 2 for the second one, 6.23 and 2 for the third one, 3.97 and 1 for the fourth one, and 7.1 and 28 for the last one.

Results

To conclude this section about unknown change points in real world e-sports data, let us address the research question opening the chapter:

> "Is it possible to apply CPD algorithms to e-sports data in order to detect significant changes in their mechanics?"

The results obtained for this final experiment seem to confirm the optimal results obtained by BOCPD across all the proposed experiments on e-sports data. This further suggests the possibility of successfully applying of BOCPD in such contexts.

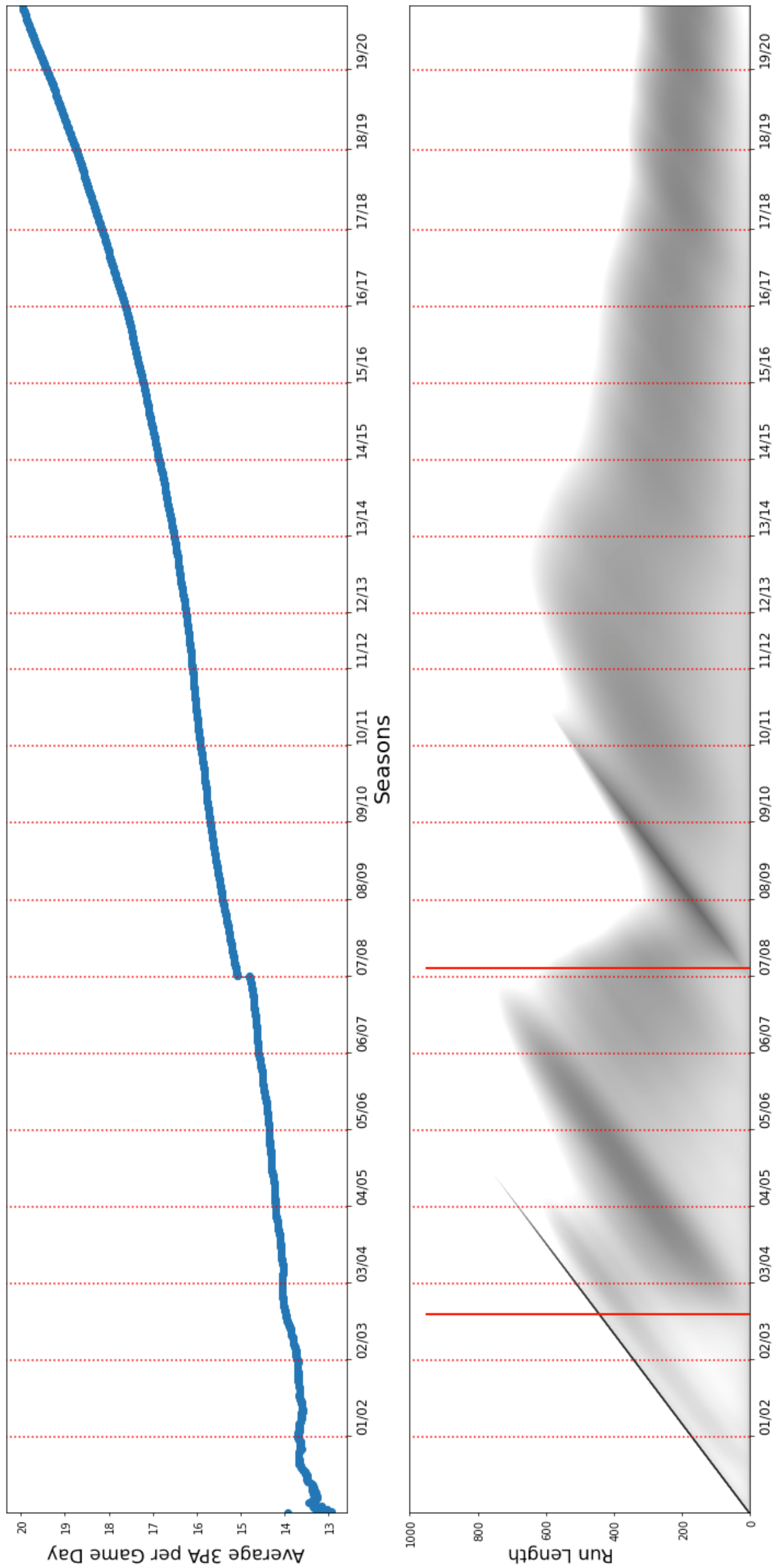


Figure 6.1: Results of the application of BOCPD on NBA 3PA data

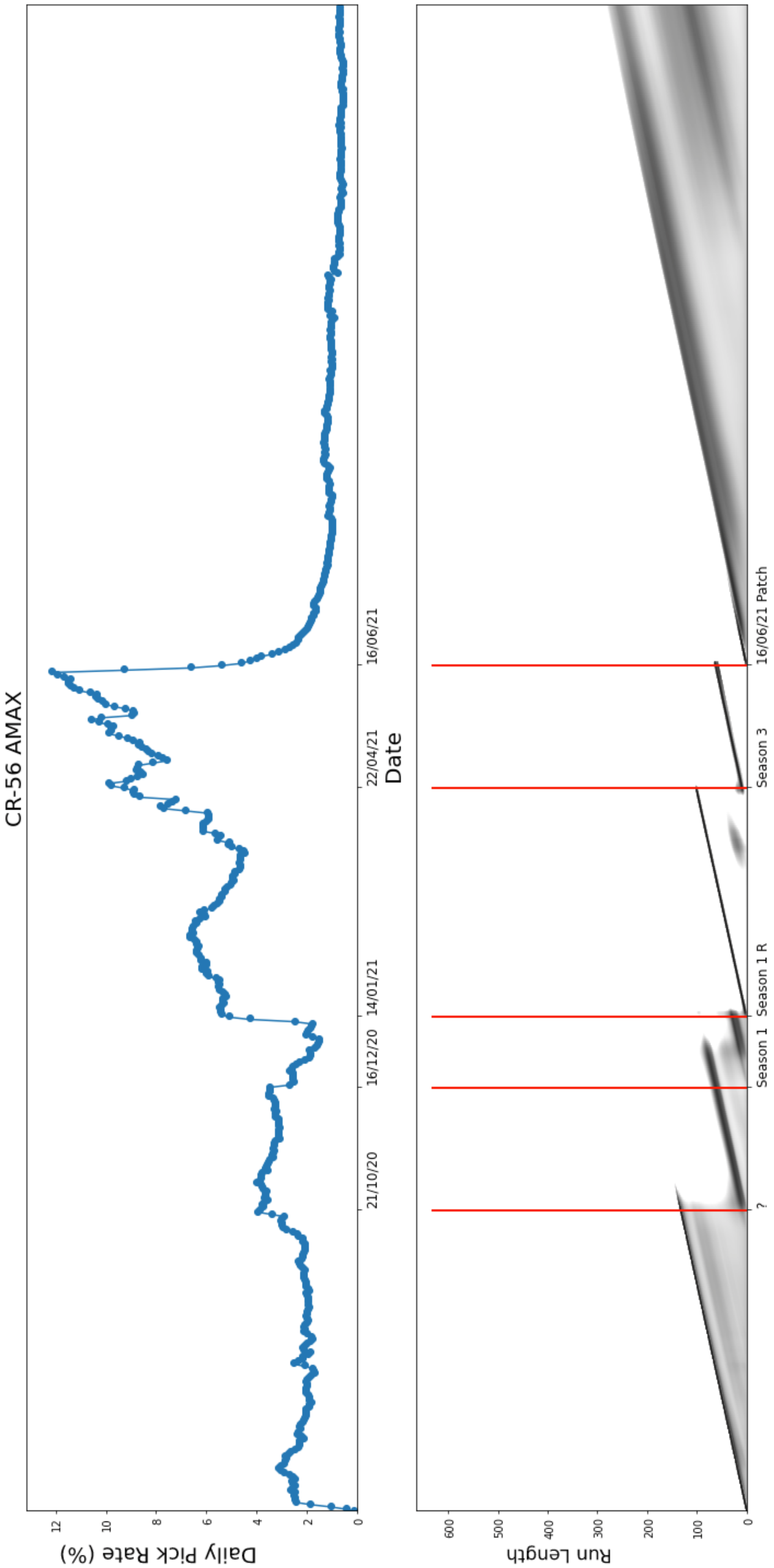


Figure 6.2: Results of the application of BOCPD on Call of Duty: Warzone data

Chapter 7

Conclusions

The goal of this thesis was to assess whether Change Point Detection algorithm Bayesian Online Change Point Detection (BOCPD) (Adams and MacKay [15]) was suitable for application to sports and e-sports data with goal of discovering significant changes in the dominant strategy. In order to do so, we firstly analyzed the mathematical framework behind BOCPD. Secondly, we presented how we adapted the algorithm and our first proposed experiment on artificial data with known change points, where F-Score and Recall metrics were used to evaluate the performance of the algorithm. Findings from this experiment suggests that the algorithm is sensitive enough to detected smaller changes, given that they happen over a not too long period of time, as very gradual changes seemed to be harder to detect. We argue, however, that such changes are less difficult to adapt to, as they happen organically in the development of a discipline, it belonging to sports or e-sports. Thirdly, we presented our second proposed experiment, using real world sports and e-sports data with known change points. The results obtained for both types of data suggest that BOCPD can be successfully applied to both sports and e-sports real world data. We observed that in the case of rather small and gradual change, BOCPD detects a change point with a lower degree of certainty. This could create problems in a practical application of the algorithm to a real-life scenario, as the low degree of certainty could introduce some uncertainty about the output presented. Lastly, we replicated the experiment proposed in the original paper by applying BOCPD to real world sports and e-sports data with unknown change points. Results obtained in this last experiment were not aligned for the two different types of data. While results obtained on sports data were not optimal, while still insightful, results from the experiment on e-sports data were instead very good. This likely originates from the fact that data used for e-sports presented far more significant changes in terms of both magnitude and abruptness of change. More testing, including sequences with less obvious changes would be necessary in this context.

As for possible future developments, we identified two main aspects that could be investigated more extensively. First, as we mentioned in Section 4.1.3, the delay with which a change point is detected would be an interesting aspect to take into account, especially if the algorithm were to be deployed in a real-life scenario. Secondly, implementing a more extensive testing framework, in particular for sports, would allow for . For instance, it would be interesting to see if comparable results can be obtained for javelin throw when considering results from single events rather than the progression of the world record.

Bibliography

[1] NBA.com. *NBA's 3-point revolution: How 1 shot is changing the game*. 2021. URL: https://www.nba.com/news/3-point-era-nba-75 (visited on 07/29/2022).

[2] Shea S. *The 3-point revolution*. 2020. URL: https://shottracker.com/articles/the-3-point-revolution (visited on 07/29/2022).

[3] *Basketaball Reference*. 2022. URL: https://www.basketball-reference.com/ (visited on 07/29/2022).

[4] Bodenham D. and Adams N.M. "Continuous monitoring for changepoints in data streams using adaptive estimation." In: *Statistics and Computing* 27.5 (2017), pp. 1257–1270.

[5] Malladi R., Kalamangalam G.P., and Aazhang B. "Online Bayesian Change Point Detection Algorithms for Segmentation of Epileptic Activity". In: *Proceedings of 2013 Asilomar conference on signals, systems and computers*. Asilomar Hotel and Conference Grounds Pacific Grove, California, 2013, pp. 1833–1837.

[6] Wan X., Wang Y., and Zhao D. "Quality monitoring based on dynamic resistance and principal component analysis in small scale resistance spot welding process". In: *The International Journal of Advanced Manufacturing Technology volume* 86 (2016), pp. 3443–3451.

[7] Zhang Y. et al. "Detecting and predicting the topic change of Knowledge-based Systems: A topic-based bibliometric analysis from 1991 to 2016". In: *Knowledge-Based Systems* 133 (2017), pp. 255–268.

[8] Aminikhanghahi S., Wang T., and Cook D. "Real-Time Change Point Detection with Application to Smart Home Time Series Data". In: *IEEE Transactions on Knowledge and Data Engineering* 31.5 (2019), pp. 1010–1023. DOI: 10.1109/TKDE.2018.2850347.

[9] Kawahara Y., Yairi T., and Machida K. "Change-Point Detection in Time-Series Data Based on Subspace Identification". In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. 2007, pp. 559–564. DOI: 10.1109/ICDM.2007.78.

[10] Page E.S. "Continuous Inspection Schemes". In: *Biometrika* 41 (1954), pp. 110–115.

[11] Yao Y. "Estimating the number of change-points via Schwarz' criterion". In: *Statistics Probability Letters* 6 (1988), pp. 181–189.

[12] Pignatiello J. and Simpson J. "Magnitude-robust control chart for monitoring and estimating step changes for normal process means". In: *Quality and Reliability Engineering* 18 (2002), pp. 429–441.

[13] Fahmy H. and Elsayed E. "Drift time detection and adjustment procedures for processes subject to linear trend". In: *International Journal of Production Research* 44 (2006).

[14] Fearnhead P. and Liu Z. "On-Line Inference for Multiple Changepoint Problems". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 69 (2007), pp. 589–605.

[15] Adams R.P. and MacKay D.J.C. "Bayesian Online Changepoint Detection". In: *ArXiv* (2007).

[16] Gundersen G. W et al. "Active multi-fidelity Bayesian online changepoint detection". In: *arXiv preprint arXiv:2103.14224* (2021).

[17] Barry D. and Hartigan J.A. "Product Partition Models for Change Point Problems". In: *The Annals of Statistics* 20 (1992), pp. 260–279.

[18] Gregory Gundersen. *Bayesian Online Changepoint Detection*. 2019. URL: `http://gregorygundersen.com/blog/2019/08/13/bocd/` (visited on 05/03/2022).

[19] Tunnicliffe Wilson G. "On the Use of Marginal Likelihood in Time Series Model Estimation". In: *Journal of the Royal Statistical Society, Series B* 51 (1989), pp. 15–27.

[20] Gutiérrez-Peña E. et al. "Exponential and bayesian conjugate families: Review and extensions". In: *TEST - Official Journal of the Spanish Society of Statistics and Operations Research* 6 (1997), pp. 1–90.

[21] Machin D., Cheung Y.B., and Mahesh Parmar. *Survival Analysis: A Practical Approach, 2nd Edition"*. Wiley, 2006.

[22] Philip Bretz. *Notes on Bayesian Changepoint Detection*. 2020. URL: `https://math.ou.edu/~pbretz/Changepoint_Detection.pdf`.

[23] van Ingen Schenau G. J. "KLAPSCHAATS: SCHAATS WAARBIJ DE STAND VAN DE SCHOEN (EN VOET) TEN OPZICHTE VAN DE BUIS (INKL. STAAL) VAN STAND KAN VERANDEREN". 8500483.

[24] International Skating Union - ISU. *1'000m Men World Record Progression*. 2022. URL: `http://www.isu.html.infostradasports.com/cache/TheASP.asp@PageID%3D103150&SportID%3D103&RecordtypeID%3D10&EventID%3D2&GenderID%3D1&TaalCode%3D2&StyleID%3D0&Cache%3D2.html?695919`.

[25] Inc. Activision Publishing. *Two incredible milestones: Call of Duty®: Warzone™ Reaches 100 Million Players, Premium Call of Duty® Game Sales Eclipses 400 Million*. 2021. URL: `https://www.callofduty.com/blog/2021/04/Incredible-Warzone-and-Game-Sales-Milestones`.