NTNU
Norwegian University of
Science and Technology

Changepoint model selection in Gaussian data by maximization of approximate Bayes Factors with the Pruned Exact Linear Time algorithm

Kristin Benedicte Bakka

Master of Science in Physics and Mathematics
Submission date: June 2018
Supervisor: Mette Langaas, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis constitutes the course *TMA4900 Industrial Mathematics Master's Thesis* which is a mandatory part of the master's degree in Industrial Mathematics at NTNU. The course accounts for 30 of 30 credits at the 10th semester of the master's degree. This thesis is in collaboration with the Sensor Project at the Big Insight SFI hosted by the Norwegian Computing Center.

The purpose of the collaboration is detection and prediction of anomalies on ships, and the main topic of this thesis is multiple changepoint detection. I am deeply grateful to Killick and Eckley for having developed the elegant algorithm Pruned Exact Linear Time that I have enjoyed so much becoming intimately acquainted with, and to Zhang and Siegmund whose mBIC has set my mind on fire. I hope it will be as enjoyable to read about the revelations I had when working with this project as it was to write them down.

I would like to thank Professor Ingrid Glad from the University of Oslo (UiO) for leading the collaboration. I would also like to thank Andreas Brandsæter from DNV GL as well as the other collaborators. Furthermore I would like to thank Professor Gunnar Taraldsen and Professor Øyvind Bakke. I am also deeply grateful to my supervisor Professor Mette Langaas for informative discussions, good advice, and for her investment of time and effort into my development as a statistician. Furthermore I would like to thank all the other brilliant professors and teachers who have taught me the skills and knowledge needed to embark on this thesis.

Last but not least I want to thank my beloved family, boyfriend and close friends for your understanding and assistance while I worked on this thesis. I love you guys.

Abstract

In this thesis we consider the changepoint detection in independently distributed Gaussian data. Detection of multiple changepoints in a data set is treated as a model selection problem where the model complexity is dependent on the number of changepoints. The Bayes Factor is a practical model selection tool of which the Bayesian Information Criterion (BIC) is a popular approximation. The BIC is twice the maximum log likelihood of the data under the model minus a penalty for number of changepoints, and is to be maximized. We develop the log likelihood for both univariate and multivariate Gaussian data.

Although the changepoint model is an irregular statistical model, BIC is asymptotically consistent when the data are univariate and independently Gaussian distributed with a known variance. For Gaussian data also two versions of the modified BIC (mBIC) are asymptotically consistent approximations of the Bayes Factor. As the penalty for model complexity is often treated as a tuning parameter in applications, we propose a range for it when the data are independently Gaussian distributed and the approximate value of the variance is known.

For data that are univariate Gaussian distributed with known variances the mBIC involves an additional penalty on the relative positions of the changepoints which is small when the changepoints are evenly distributed in the data and large when they are clustered together. Although in the mBIC criterion the penalty on the relative positions of the changepoints are set by maximization of the likelihood term, we instead let them be set by maximization of the sum of the likelihood and the penalty terms. Thus we get a criterion that we can maximize with the algorithm Pruned Exact Linear Time (PELT), which runs on $O(n)$ time under certain conditions. In the thesis we also suggest a modification to the algorithm Changepoint Detection for a Range of Penalties (CROPS) that lets us maximize the original mBIC using PELT.

In simulations we see that PELT performs better than the popular changepoint detection algorithm Binary Segmentation (BinSeg) when both are applied to maximize BIC. Although BIC is usually a strict criterion in the sense that it prefers a parsimonious model, on simulations where the variance is known it is outperformed by mBIC which has a higher penalty on model complexity than BIC for most data. For the case where the data are univariate Gaussian but the variance is not known, we do not find a simple criterion to maximize. Rather we propose an ad hoc criterion similar to both BIC as applied to these changepoint data, and to mBIC for Gaussian data with known variance. When p parameters are estimated in the likelihood, changepoints need to be separated by at least $p - 1$ points. We generalize PELT to account for this, and use Directed Asyclic Graphs to illustrate the inner workings of OP, PELT and our generalized PELT.

Sammendrag (Abstract in Norwegian)

Denne masteroppgaven omhandler deteksjon av endringspunkter (changepoints) i uavhengige normalfordelte data. Vi finner endringspunkter ved hjelp av modellseleksjon. Kompleksiteten til en modell avhenger av antall endringspunkter. Det bayesianske informasjonskriteriet (BIC) er en populær approksimasjon av bayes faktor, som er et praktisk verktøy for modellseleksjon. BIC består av to ledd og skal maksimeres. Det ene er to ganger logaritmen til den maksimale rimelighetsfunksjonen til datasettet. Det andre leddet er negativt og er en straff for antall endringspunkter. Vi utvikler logaritmen til rimelighetsfunksjonen for data både for univariate og multivariate normalfordelinger.

Selv om endringspunktmodellen er en irregulær statistisk modell er BIC asymptotisk konsistent når dataene er univariate og identisk normalfordelte med kjent varians. For normalfordelte data er to versjoner av modifisert BIC (mBIC) også asymptotisk konsistente approksimasjoner av bayes faktor. Siden straffen for hvor kompleks modellen er ofte blir behandlet som en justeringsparameter i anvendelser foreslår vi et intervall for straffen når dataene er uavhengig normalfordelte og verdien til variansen kan anslås.

For data som er univariat normalfordelte med kjent varians innebærer mBIC en ekstra straff for den relative posisjonen til endringspunktene. Straffen er stor når endringspunktene er nært hverandre, og liten når de er jevnt spredt utover datasettet. Selv om denne ekstra straffen i mBIC egentlig bestemmes ut fra posisjonene som maksimerer rimelighetsfunksjonen til modellen, velger vi å sette den slik at den maksimerer summen av straffen og rimelighetsfunksjonen. På den måten får vi et kriterium vi kan maksimere med algoritmen Pruned Exact Linear Time (PELT) som kjører i $O(n)$ tid under visse betingelser. I masteroppgaven foreslår vi også å modifisere algoritmen Changepoint Detection for a Range of Penalties (CROPS) slik at vi kan maksimere mBIC med PELT slik kriteriet opprinnelig er definert.

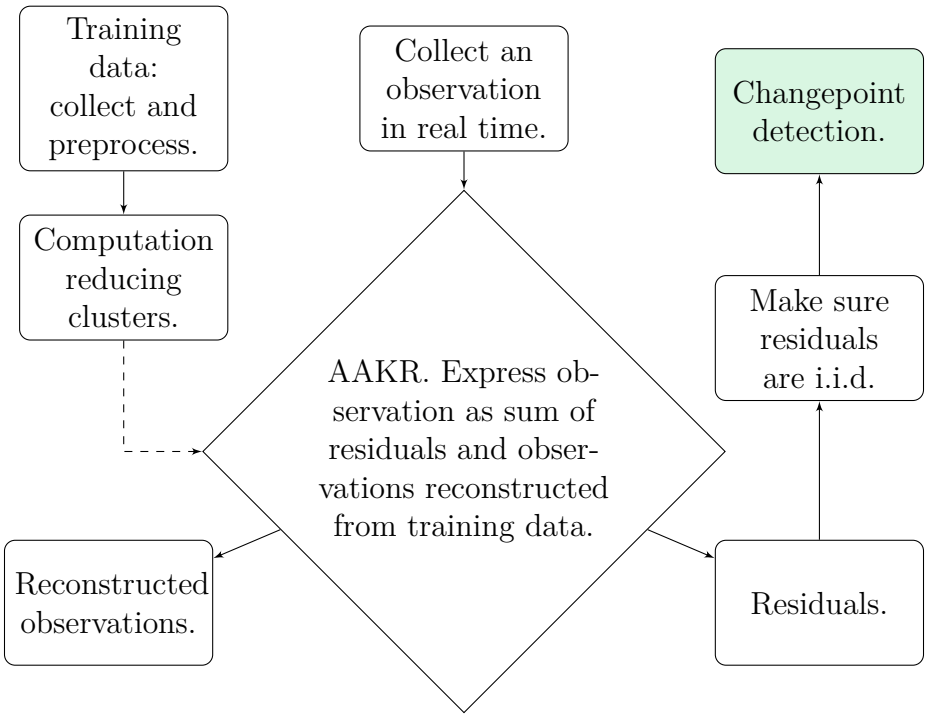
I simuleringer ser vi at PELT gir bedre resultater enn den populære algoritmen Binary Segmentation (BinSeg) når begge brukes til å maksimere BIC for deteksjon av endringspunkter. Selv om BIC vanligvis er et strengt kriterium som foretrekker en enkel modell ser vi i simuleringer hvor variansen er kjent at mBIC presterer bedre i kraft av å ha enda høyere straff for modellkompleksitet. Når dataene er univariate og normalfordelte men variansen ikke er kjent finner vi ikke et enkelt kriterium vi kan maksimere. I stedet foreslår vi et ad hoc kriterium som har felles egenskaper med både BIC og mBIC for normalfordelte data med kjent varians. Når p parametre blir estimert i rimelighetsfunksjonen må det være minst $p - 1$ datapunkter mellom hvert endringspunkt. Vi generaliserer PELT slik at algoritmen tar hensyn til det, og lager en grafisk fremstilling for å visualisere virkemåtene til OP, PELT og vår generaliserte PELT ved hjelp av rettede asykliske grafer.

Contents

1. Introduction

This thesis is associated with an ongoing research project on analysis of sensor data, which we will refer to as the Big Insight sensor project. An important objective is to solve a real world problem experienced by people on ships. A ship might have 500 sensors that collect data in real time. When in a specific operational mode, for instance transit at high speed (100mph), the problem is to detect it when anything unexpected happens. As the operational mode was supposed to stay the same, the unexpected event might need the attention of the crew. For instance if the speed suddenly drops, but everything else stays the same, the speedometer might be broken, or an inefficiency has happened somewhere in the system. This type of problem is called *anomaly detection* and it can be solved by *changepoint detection*. A main point is that we want to detect not some specific change, but any change in the system. To be able to detect unexpected events a so-called training set assumed to only contain normal events is collected. For each data point there is one observation for each sensor.

Figure 1.1: Overview of the full process to detect unexpected events in a sensor system on a ship. The box with colored background marks where changepoint detection algorithms such as PELT may be applied. In the box below the colored box i.i.d. is short for independent and identically distributed.



Various parameters of the sensor system change frequently, so that both the training data and the real time data are expected to contain multiple changepoints that are not indicators of an anomaly. In Brandsæter et al. (2016) the anomaly detection problem is divided into different parts. A schematic view of the detection process is illustrated in Figure 1.1. Initially some training data containing no known anomalies is gathered and preprocessed. To save computation time later, a technique where the training data are represented as clusters might be applied at this point. Then the on-line anomaly detection may begin. On every vector of real time observations *Auto Associative Kernel Regression* (AAKR) is performed. That is, the observation is reconstructed as a weighted average of the preprocessed training data. The reconstructed observations are the values displayed to the captain on the bridge. For instance the temperature outside might be measured at 10°C while the reconstructed value is 7°C. Then the captain will read off 7°C on the recalibrated thermometer. The difference between reconstructed values and observed values is the residual, and is what will be used in classifying the state of the system to normal or anomalous. The box with colored background marks an online changepoint problem. Another option is to perform changepoint detection on the training data in order to gain insight on the changepoint process as it is when it is in control. Then we may devise some other method to detect when the changepoint process is out of control, that is when an anomaly has occurred.

A common method of detecting multiple changepoints is to maximize some criterion that consists of a term that penalizes the number of changepoints and a term that penalizes changes in the data set if they occur anywhere but at the changepoints. Such criteria are often based on an assumption that the data are independently Gaussian distributed (Truong et al., 2018). In applications where the assumptions only hold approximately the term that penalizes model complexity is then slightly adjusted. In applications the term that penalizes the number of changepoints is often treated as a *tuning parameter*, a parameter that is set from the data so that the changepoints detected seem reasonable to the researcher. Another option is to set the penalty on number of changepoints according to a specialized model selection criterion that has good theoretical properties given some assumptions. To be able to maximize changepoint criteria for multiple changepoints specialized algorithms are needed. One challenge is that we want the algorithm to maximize the criterion and not simply find a set of changepoints that gives a large value for the criterion. Other challenges are that the algorithm should allow for criteria that are as complex as possible, and that the algorithm should also be fast.

One contribution from this thesis is that we generalize the changepoint detection algorithm named *Pruned Exact Linear Time* (PELT) so that it can be used to maximize criteria where more than one parameter is es-

timated. PELT finds multiple changepoints fast under certain conditions (Killick et al., 2012a). Schwarz (1978) and Zhang and Siegmund (2007) present specialized criteria that are shown to have good properties when one assumption is that the data are independently Gaussian distributed. One contribution from this thesis is that we thoroughly explain how these criteria may be used in changepoint detection. We also propose a method of selecting the range of the term that penalizes the number of changepoint in applications where it is treated as a tuning parameter.

Section 2 starts off with defining concepts relevant to detecting unexpected events, and to model selection in general. In Section 3 we present specialized criteria for changepoint detection in univariate Gaussian data when the variance is one, and detail how to use these to get criteria that can be maximized with the algorithms in Section 3.5. Only small alterations are needed when the data are Gaussian with a known variance of any other value, but we postpone handling this to Section 5 as it makes the presentation in Section 3 easier to follow. One of the algorithms we present is the currently popular fast changepoint detection algorithm *Binary Segmentation* (BinSeg). The performance of BinSeg and PELT on simulated data sets with different number of changepoints is presented in Section 4. In Section 4 also the performance of the changepoint detection criteria presented so far are evaluated using simulated data, and Section 4.4 contains a preliminary discussion of some of the subjects covered so far in the thesis. Then in Section 5 we present criteria for changepoint detection in univariate Gaussian data when both the mean and the variance need to be estimated. Section 5.6 details our generalization of PELT that allows more than one parameter to be estimated, and the implementation is available in Appendix B. In Appendix A we develop the likelihood into a form that may be used when maximizing the BIC. The discussion in Section 6 concludes the thesis.

2. Statistical background

The field of anomaly detection by changepoint detection developed in wartime out of the need to craft weapons of a certain quality, without the need of too many samples to detect when the weapons produced were no longer satisfactory. Since then it has been applied in various settings. This section presents the central concepts for anomaly detection and model selection.

2.1. Anomaly detection

We observe a data set $\boldsymbol{x}_t = (x_1, \ldots, x_t)$ sequentially. The data is the output from some system. If we assume the system is in control for $t = 1, \ldots, \kappa$, and out of control for $t = \kappa + 1, \kappa + 2, \ldots$, then a *fault* has occurred between observation x_κ and $x_{\kappa+1}$. We may consider $x_1, \ldots, x_t$ as being realizations from some probability distribution. When the system is in control we say that it is in *normal state* or *normal condition*. The system not being in normal state constitutes an *anomaly*.

Some anomalies result in changes in the underlying distribution of $x_1, \ldots, x_t$. Such a change may be *abrupt* and occur between x_κ and $x_{\kappa+1}$, or *gradual* and occur between x_κ and $x_{\kappa+k}$ for some $k \in \mathbb{N}$. If the state changes back to normal the change was *transient* and if not it is labeled a *persistent* change (Tveten, 2017). When the change in distribution is abrupt and persistent x_κ is a *changepoint*. We then have two batches of data, the in-control *batch* is the data set $x_1, \ldots, x_\kappa$, and the out of control batch is $x_{\kappa+1}, x_{\kappa+2}, \ldots$. A changepoint is characterized by marking a change in the underlying distribution, and in the general field of changepoint detection there may be multiple changepoints. Applied in an anomaly detection setting the changepoint is interpreted as the indicator to where the system transitions to an anomalous state.

Example 2.1. A ship has multiple sensors, and at time t the output of the system x_t is a vector with as many elements as there are sensors. In this example the system is in the normal state when the hull is intact, while the hull being damaged constitutes an anomaly. If the hull gets damaged between $t = \kappa$ and $t = \kappa + 1$ then a fault has happened and the state is out of control after $t = \kappa$. This may or may not affect the outputs $x_{\kappa+1}, \ldots, x_t$. Assuming that it affects the output such that $x_1, \ldots, x_\kappa$ are realizations from one distribution and $x_\kappa, \ldots, x_t$ are realizations from another distribution, then given $\boldsymbol{x}_t$ changepoint detection may be used to estimate the value of κ.

One approach to anomaly detection is to identify possible causes for faults and to analyze how they would affect the output of the system, in order to recognize such a fault when it occurs. With changepoint detection we may assume some underlying distribution for the output from the system

and detect some change in the properties of the distribution. We might seek to detect some specific property, for instance increase or in decrease of the mean, or to identify any possible change. Changepoint detection is also concerned with estimating the underlying distribution of the data in the two batches. If $x_1, \ldots, x_t$ each are univariate the problem is referred to as single stream changepoint detection problem. The focus in this work is anomaly detection by single stream changepoint detection.

Example 2.2. Assume a ship has multiple sensors. At time t the output of the system $\boldsymbol{x}_t$ is a vector with as many elements as there are sensors. Then any anomalous state should be detected from $\boldsymbol{x}_t$. This time the state is normal when everything works as it is supposed to. When anything is out of order, for instance the temperature of the engine is too high, the hull is deformed, or a sensor is broken, then the state is anomalous.

2.2. Detection setting

In statistics *sequential analysis* is statistical analysis where data are evaluated as they are collected until a pre-defined stopping condition is fulfilled. Accordingly *sequential change point detection* is sequential analysis with the goal of finding change point(s). Based on the original work of Wald Price (1948) states that sequential analysis is best suited to test hypotheses on data where it is expensive to obtain the samples, as it allows reaching a conclusion that is correct at a pre-assigned level of probability with fewer samples than with classical statistical methods. In some sequential methods when a new data point is collected, only some previously stored statistic and the new data point are involved in the evaluation of the hypothesis or stopping condition, so storing the previous data points is not necessary. Commonly this reduces the computational cost.

In *online changepoint detection* we also have a sequential data set where we evaluate the data as it is collected. Commonly the samples are not costly to obtain, but arrive at a set pace, for instance once every five seconds. In online changepoint detection we want to reach a reliable conclusion in as little time as possible, and thus with as few samples as possible. In addition the computation to be performed needs to be fast enough for the evaluation at the nth step to be finished before the $(n+1)$th data point is observed. Any algorithm that runs fast enough to evaluate before the next time step may be applied in an online setting. Conversely in *offline changepoint detection* all the data points are known in advance, and all the data are used to identify changepoints. However in online applications it is beneficial that in general the conclusion reached in the nth step is similar to the conclusion reached in the $(n+1)$th step.

An algorithm is commonly referred to as an *online algorithm* if it is readily applicable in an online setting, and otherwise it is referred to as an *offline algorithm*. However in the online setting all algorithms need to observe some samples from the new distribution in order to detect that a change has occurred. In order to compare methods in changepoint detection Aminikhanghahi and Cook (2017) defines an *ϵ–real time algorithm* as an online algorithm which needs at least ϵ data samples from the new batch of data to be able to identify the changepoint. All the algorithms considered in this thesis are 1-real time algorithms if applied in an online setting. However the two main algorithms under study are arguably not readily applicable in an online setting.

2.3. Evaluation of detection method

A *hypothesis* is a statement about a population parameter. Given a sample $\boldsymbol{X}$ from the population a *hypothesis test* is a rule to decide which of two hypotheses is true. The *null hypotheses* is usually denoted H_0 and is assumed to be true unless the sample indicates otherwise. The null hypothesis may be rejected in favor of the alternative hypotheses which is usually denoted H_1. The subset of the sample space for which H_0 is rejected and H_1 is accepted is the *rejection region* R of the hypothesis test. In practice the acceptance or rejection of H_0 is decided by the value of a *test statistic*, a real or vector valued function with a domain that includes the sample space.

Where the parameter space is $\Theta = \Theta_0 \cup \Theta_0^c$ and $\Theta_0 \cap \Theta_0^c = \emptyset$ the general format of the two hypotheses is $H_0 : \theta \in \Theta_0$ and $H_1 : \theta \in \Theta_0^c$. Then the *Type I error* $Pr(\boldsymbol{X} \in R|\theta \in \Theta_0)$ is the probability of falsely rejecting H_0 when H_0 is true. Conversely the probability of not rejecting H_0 when H_1 is true is $1 - Pr(\boldsymbol{X} \in R|\theta \in \Theta_0^c)$ and is called the *Type II error*. The function of θ defined by $\beta(\theta) = Pr(\boldsymbol{X} \in R)$ is the *power function* of a hypotheses test. When the two hypotheses are completely specified, they are *simple* and may be denoted $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$.

When we perform a hypothesis test we use the power and the type I and type II errors to evaluate the method. The presentation here is based on Chapter 8 of Casella and Berger (2002), which subscribes to the view that in a hypothesis testing problem either of two actions is going to be taken - either the assertion of H_0 or H_1. An alternative view is that the researcher does not believe H_0 is true, but is only willing to reject it if the sample is in the rejection region. Commonly the dimension of the parameter space under H_0 is no larger than under H_1.

In an online setting, where the data set $\boldsymbol{x}_t = (x_1, \ldots, x_\kappa, x_{\kappa+1}, \ldots, x_t)$ consists of two batches as in Section 2.1, we attempt to find κ. Then we denote by E^κ an expected value given κ and the distribution of the data in the two batches. When $\kappa \geq t$ and the data set contains no changepoints, we denote the expected value as E^∞. The expected number of samples T before a change is detected when there is no change is

$$\mathrm{E}^\infty(T), \tag{2.1}$$

and we call it the Average Run Length (ARL). For a requirement $\mathrm{E}^\infty(T) < c_1$ the expected time between occurrence and detection of changepoint is

$$\mathrm{E}^\kappa(T - \kappa | T > \kappa),$$

which we call the Expected Detection Delay (EDD).

If we want to do anomaly detection by changepoint detection in an online setting, we assume the state of the output from the system is in control at first. At any given time point we test whether a fault has occurred. So we

test H_0 : *no changepoint*, against H_1 : *a changepoint at* κ. Then EDD is a measure for the power of the test, and ARL is a measure for the type I error. A test with high power will have a low EDD, and a test with high ARL will have a small type I error.

2.4. Likelihood

We have a sample $\boldsymbol{x}_t = (x_1, \ldots, x_t)$ of length t that is a realization of $\boldsymbol{X}_t = (X_1, \ldots, X_t)$. Then we write $x_1 \sim f$ or $X_1 \sim f$ to denote that the random variable X_1 has probability distribution $f(x_1)$ (probability density function when X_1 is a continuous variable, or probability mass function when X_1 is a discrete variable). The probability distribution $f(x|\boldsymbol{\theta})$ is defined by the parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. Accordingly $\boldsymbol{x}_t \sim f(\boldsymbol{x}_t|\boldsymbol{\theta})$ denotes that $\boldsymbol{x}_t$ is a realization from $\boldsymbol{X}_t$ with probability distribution $f(\boldsymbol{x}_t|\boldsymbol{\theta})$, and we say that $f(\boldsymbol{x}_t|\boldsymbol{\theta})$ is the *underlying distribution* of $\boldsymbol{x}_t$. Given that $\boldsymbol{x}_t$ is observed, the function of $\boldsymbol{\theta}$ defined by $L(\boldsymbol{\theta}|\boldsymbol{x}_t) = f(\boldsymbol{x}_t|\boldsymbol{\theta})$ is the *likelihood* of $\boldsymbol{\theta}$. When $X_1, \ldots, X_t$ are independent and identically distributed (i.i.d.), the likelihood function for the observation $\boldsymbol{x}_t$ is

$$L(\boldsymbol{\theta}|\boldsymbol{x}_t) = \prod_{i=1}^{t} L(\boldsymbol{\theta}|x_i),$$

and we denote as the *log-likelihood*

$$\ell(\boldsymbol{\theta}|\boldsymbol{x}_t) = \log L(\boldsymbol{\theta}|\boldsymbol{x}_t) = \sum_{i=1}^{t} \ell(\boldsymbol{\theta}|x_i) = \sum_{i=1}^{t} \log(f(x_i|\boldsymbol{\theta})).$$

The likelihood denotes how likely the observation is under the distribution considered. We often want to choose parameters such that the likelihood is maximized. The notation $\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\boldsymbol{x}_t)$ denotes the largest likelihood for any parameter θ. We express the value of $\boldsymbol{\theta}$ such that the maximum is obtained with $\arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\boldsymbol{x}_t)$. The *maximum likelihood estimate* (MLE) of $\boldsymbol{\theta}$ is thus $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\boldsymbol{x}_t) = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\boldsymbol{x}_t)$, where we follow the convention to indicate a MLE with a hat above the parameter.

The probability density function (pdf) of a univariate normal random variable x is

$$f(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

which we will denote $\mathcal{N}(\mu, \sigma^2)$ in this thesis. Thus assuming $\boldsymbol{x}_t = (x_1, \ldots, x_t)$ are realizations from $X_1, \ldots, X_t$ with elements that are i.i.d. $\mathcal{N}(\mu, \sigma^2)$ we get

$$L(\mu, \sigma^2|\boldsymbol{x}_t) = \prod_{i=1}^{t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right),$$

and

$$l(\mu, \sigma^2|\boldsymbol{x}_t) = -\frac{t}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{t}(x_i-\mu)^2. \tag{2.2}$$

To find the MLE of the mean μ and the variance σ^2 we solve the system $\frac{\partial}{\partial\theta_i} l(\boldsymbol{\theta}|\boldsymbol{x}_t) = 0$. The resulting estimates are $\hat{\mu} = \frac{1}{t}\sum_{i=1}^{t} x_i$ and $\hat{\sigma}^2 =$

$\frac{1}{t}\sum_{i=1}^{t}(x_i - \hat{\mu})^2$, such that

$$\max_{\mu,\sigma^2} l(\mu, \sigma^2|\boldsymbol{x}_t) = -\frac{t}{2}\log(2\pi\hat{\sigma}^2) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{t}(x_i - \hat{\mu}).$$

In the expression for the maximum observed log-likelihood and the estimate for σ^2, the estimate $\hat{\mu}$ is replaced by the true value μ if it is known.

2.5. Likelihood ratio test

Now we move on to a popular test statistic presented in Casella and Berger (2002) as Definition 8.2.1. Then we will need that $\sup_\Theta L(\boldsymbol{\theta}|\boldsymbol{x}_t)$ denotes the smallest upper bound of the likelihood in the parameter space Θ.

Definition 2.1. The *likelihood ratio test* (LRT) statistic for testing $H_0 : \boldsymbol{\theta} \in \Theta_0$ versus $H_1 : \boldsymbol{\theta} \in \Theta_0^c$ is

$$\lambda(\boldsymbol{x}) = \frac{\sup_{\Theta_0} L(\boldsymbol{\theta}|\boldsymbol{x})}{\sup_\Theta L(\boldsymbol{\theta}|\boldsymbol{x})},$$

where $\Theta = \Theta_0 \cup \Theta_0^c$ and $\Theta_0 \cap \Theta_0^c = \emptyset$. An LRT is any test that has a rejection region on the form $\{x : \lambda(x) \leq c\}$, where c is any number satisfying $0 \leq c \leq 1$.

When the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ exists, $\sup_\Theta L(\boldsymbol{\theta}|x) = L(\hat{\boldsymbol{\theta}}|x)$. So $0 < \lambda(x) \leq 1$, where $\lambda(x) = 1$ when $\hat{\boldsymbol{\theta}} \in \Theta_0$. This means that with the simple null hypothesis (uniquely specified distribution) $H_0 : \boldsymbol{\theta} = \boldsymbol{\theta}_0$ and the alternative hypothesis $H_1 : \boldsymbol{\theta} \neq \boldsymbol{\theta}_0$ the i.i.d. normal observations $\boldsymbol{x}_t = (x_1, \ldots, x_t)$ give

$$\lambda(x) = \left(\frac{\hat{\sigma}}{\sigma_0}\right)^t \exp\left(\sum_{i=1}^t \left(\frac{(x_i - \hat{\mu})^2}{2\hat{\sigma}^2} - \frac{(x_i - \mu_0)^2}{2\sigma_0^2}\right)\right).$$

There are two interesting results about the distribution of the LRT statistic that we will present here. The first one is the Neymann-Pearson Lemma, and is found in Casella and Berger (2002) as Theorem 8.3.12.

Theorem 2.1. For a test of $H_0 : \boldsymbol{\theta} \in \Theta_0$ against $H_1 : \boldsymbol{\theta} \in \Theta_0^c$, suppose the elements of $\boldsymbol{X}_t = (X_1, \ldots, X_t)$ are i.i.d. $f(x|\boldsymbol{\theta})$, the type I error is $\alpha = Pr(\lambda(x) \geq c|H_0)$. Then the power of the test is smaller or equal to $Pr(\lambda(x) \geq c|H_1)$, which is the the power of the likelihood ratio test.

For the next theorem we need the χ^2 distribution which is defined by the probability density function

$$\chi_p^2 = \frac{1}{2^{\frac{p}{2}}\gamma(\frac{p}{2})} x^{\frac{p}{2}-1} e^{-\frac{x}{2}}, \quad x > 0, \tag{2.3}$$

where p is a natural number and is the degrees of freedom.

Theorem 2.2. For testing $H_0 : \boldsymbol{\theta} \in \Theta_0$ versus $H_1 : \boldsymbol{\theta} \in \Theta_0^c$ where $p = \dim(\Theta) - \dim(\Theta_0)$, suppose the elements of $\boldsymbol{X}_t = (X_1, \ldots, X_t)$ are i.i.d. $f(x|\boldsymbol{\theta})$, and that the regularity conditions discussed in Wilks (1938) hold. Then under H_0 as $t \to \infty$,

$$2W(\boldsymbol{X}_t) = -2\log(\lambda(\boldsymbol{X}_t)) \to \chi_p^2$$

in distribution, where χ_p^2 is the probability density function of the χ_p^2 distribution in Equation (2.3). This is known as the Wilks theorem.

Equivalently we may write

$$2W(\boldsymbol{X}) = 2(\log(\sup_{\Theta} L(\boldsymbol{\theta}|\boldsymbol{X})) - \log(\sup_{\Theta_0} L(\boldsymbol{\theta}|\boldsymbol{X}))).$$

With $0 < \alpha < 1$ and a k_α such that $P(k_\alpha > \chi^2_p) = \alpha$ we thus reject a null hypothesis in favor of the alternative hypothesis on confidence level $1 - \alpha$ when $2W(\boldsymbol{x}) > k_\alpha$. When $2W(\boldsymbol{x}) < k_\alpha$ we do not reject the null hypothesis.

2.6. Model selection

In this section x denotes an observation, and $\mathcal{M}$ denotes a statistical model with a parameter θ. In Bayesian statistics important quantities are the likelihood $Pr(x|\,\mathcal{M})$, the posterior probability $Pr(\mathcal{M}\,|x)$ and the prior probabilities $Pr(\mathcal{M})$ and $Pr(x)$. To compare the evidence for a model $\mathcal{M}_1$ against the evidence for another model $\mathcal{M}_0$ given an observation x, we may use the posterior odds

$$\frac{Pr(\mathcal{M}_1\,|x)}{Pr(\mathcal{M}_0\,|x)}, \tag{2.4}$$

regardless of whether the models are nested, that is whether the parameter space of one is a subset of the parameter space of the other. In Bayesian statistics we use Bayes formula for the posterior probability

$$Pr(\mathcal{M}\,|x) = \frac{Pr(\mathcal{M})Pr(x|\,\mathcal{M})}{Pr(x)}. \tag{2.5}$$

We can insert this expression into the posterior odds to get

$$\frac{Pr(\mathcal{M}_1\,|x)}{Pr(\mathcal{M}_0\,|x)} = \frac{Pr(\mathcal{M}_1)}{Pr(\mathcal{M}_0)}\frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_0)}.$$

The two terms on the right hand side of the equation are the prior odds, and the Bayes Factor $B_w(x)$ (Efron and Hastie, 2016, p. 244) is

$$B_w(x) = \frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_0)}. \tag{2.6}$$

A large Bayes Factor reflects that the evidence for $\mathcal{M}_1$ after data is collected is greater than in the prior. However if the prior odds is small then a large Bayes Factor is necessary to conclude on $\mathcal{M}_1$ in favor of $\mathcal{M}_0$.

When $\mathcal{M}_0$ and $\mathcal{M}_1$ are both simple hypotheses and the prior odds is known, we may compute the posterior odds and conclude that either model $\mathcal{M}_0$ or $\mathcal{M}_1$ is preferred. This is where the evaluation of the posterior odds in the Bayesian setting is different from a hypotheses test. In a hypothesis test we do or do not reject $\mathcal{M}_0$ in favor of $\mathcal{M}_1$, but we do not gather evidence for $\mathcal{M}_0$. When we evaluate the posterior odds we commonly use Jeffrey's scale, which is detailed in Table 2.1 (Efron and Hastie, 2016, p. 245). In many cases there is little information on the prior distributions. A convention for uninformative priors is to use the Laplace choice (Kass and Raftery, 1995) of $Pr(\mathcal{M}_1) = Pr(\mathcal{M}_2)$ such that the posterior odds equals the Bayes Factor.

When either $\mathcal{M}_0$ or $\mathcal{M}_1$ is not a simple model, as is usually the case, there are more steps to computing the Bayes Factor. We will assume that the model $\mathcal{M}$ has a parameter θ which takes on values from the parameter space Θ. If we for $Pr(x|\,\mathcal{M})$ use the maximal likelihood under $\mathcal{M}$ and the Laplace choice for prior odds, the posterior odds reduces to the likelihood

ratio. In the Frequentist perspective θ is an unknown parameter, and it makes sense to find the maximum likelihood estimate of that constant, and thus of $Pr(x|\,\mathcal{M})$. In the Bayesian perspective θ is a random variable with a distribution, and the likelihood $Pr(x|\,\mathcal{M})$ is a combination of the likelihoods under all the values θ can take on. So a formula consistent with the Bayesian approach is (Kass and Raftery, 1995)

$$Pr(x|\,\mathcal{M}) = \int_{\Theta} Pr(x|\theta, \mathcal{M}) Pr(\theta|\,\mathcal{M}) d\theta. \tag{2.7}$$

Then we need the prior distribution $Pr(\theta|\,\mathcal{M})$ for the parameters under the hypotheses.

The Bayes Factor can be used to compare the evidence for two models, that is it may be used for *model selection*. Sometimes we want to compare a number of different models, and then we use that

$$\frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_2)} = \frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_0)} \left(\frac{Pr(x|\,\mathcal{M}_2)}{Pr(x|\,\mathcal{M}_0)} \right)^{-1}.$$

As the Bayes Factor is derived from Equations (2.4) and (2.5), the two models do not need to be nested (Kass and Raftery, 1995).

An important class of probability distributions is the exponential distribution family, which has probability density on the form

$$f(x|\theta) = \exp\left(\theta \cdot y(x) - b(\theta)\right), \tag{2.8}$$

where $y(x)$ is the sufficient K-dimensional statistic, and θ is as before in the parameter space Θ. The following theorem is an approximation of the Bayes Factor that is widely used in model selection, even when its requirements are not satisfied.

Theorem 2.3. This Theorem expresses the procedure in Schwarz (1978) in a simplified manner, for the precise preconditions see Schwarz (1978). It requires a special class of prior (Schwarz, 1978) distributions $Pr(\theta|\,\mathcal{M})$. Let the Bayes Factor in question be

$$B_w = \frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_0)}$$

where $\mathcal{M}_0$ is the model where the parameter space is Θ, and the parameter space Θ_1 of $\mathcal{M}_1$ is a subspace of Θ and only has dimension p. Given the data set $\boldsymbol{x} = (x_1, \ldots, x_n)$ of independent realizations from identical distributions in the exponential family defined in Equation (2.8), the logarithm of an asymptotic approximation when n goes to infinity for the Bayes Factor is

$$\log B_w \approx B_{\mathrm{BIC}}(\boldsymbol{x}) = \log \lambda(\boldsymbol{x}) - \frac{p}{2} \log n, \tag{2.9}$$

where

$$\lambda(\boldsymbol{x}) = \frac{\sup_{\Theta_1} L(\theta|\boldsymbol{x})}{\sup_{\Theta} L(\theta|\boldsymbol{x})},$$

and Θ_1 denotes the parameter space under $\mathcal{M}_1$. We index this by BIC as it is known as the Bayesian Information Criterion (BIC) (Efron and Hastie, 2016, p. 246). It is also known as Schwarz Information Criterion (SIC).

As B_{BIC} is the logarithm of an approximation of the Bayes Factor we may use it to compare non-nested models. We will term the p in Equation (2.9) the *degrees of freedom* (df) of the model. For two models $\mathcal{M}_1$ and $\mathcal{M}_2$ with p_1 and p_2 degrees of freedom and parameter spaces Θ_1 and Θ_2 nested within Θ_0

$$\begin{aligned}
&\log \frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|M_2)} = \log \frac{Pr(x|\,\mathcal{M}_1)}{Pr(x|\,\mathcal{M}_0)} - \log \frac{Pr(x|\,\mathcal{M}_2)}{Pr(x|\,\mathcal{M}_0)} \\
&\approx B_{\text{BIC},1}(\boldsymbol{x}) - B_{\text{BIC},2}(\boldsymbol{x}) \\
&= \log \frac{\sup_{\Theta_1} L(\theta|\boldsymbol{x})}{\sup_{\Theta_0} L(\theta|\boldsymbol{x})} - \frac{p_1}{2} - \log \frac{\sup_{\Theta_2} L(\theta|\boldsymbol{x})}{\sup_{\Theta_0} L(\theta|\boldsymbol{x})} + \frac{p_2}{2},
\end{aligned}$$

so we get

$$B_{\text{BIC}}(\boldsymbol{x}) = B_{\text{BIC},1}(\boldsymbol{x}) - B_{\text{BIC},2}(\boldsymbol{x}) = \log \frac{\sup_{\Theta_1} L(\theta|\boldsymbol{x})}{\sup_{\Theta_2} L(\theta|\boldsymbol{x})} - \frac{p_1 - p_2}{2}.$$

In other words we may compare $\mathcal{M}_1$ and $\mathcal{M}_2$ by comparing their BICs with respect to $\mathcal{M}_0$. Since the BIC is an approximation of the Bayes Factor we should use Jeffrey's scale in Table 2.1. However if we are willing to disregard Jeffrey's scale and prefer $\mathcal{M}_1$ when $B_{\text{BIC}} > 1$ and $\mathcal{M}_2$ otherwise, interesting opportunities arise. Then we in effect prefer the model with the largest BIC.

We may then choose between several models by simply preferring the one with the highest BIC as defined in Equation (2.9). In that process the likelihood of $\mathcal{M}_0$ becomes obsolete, and we may define

$$\text{BIC} = 2\ell(\hat{\theta}) - p\log(n). \tag{2.10}$$

This is an equation regularly referred to as the BIC of a model. We will use this[1] formula throughout this thesis. To maximize this expression is also the model selection rule that Schwarz (1978) arrives at. The p is a penalty on the degrees of freedoms in the model. When there are more degrees of freedom the maximum likelihood is larger, and so we need the penalty to be larger as well. And so often p is simply referred to as the degrees of freedom.

[1] In Kass and Raftery (1995) Equation (2.10) is referred to as the BIC, and then BIC/2 is referred to as SIC.

As we can see from Equations (2.9) and (2.10) the BIC is on the form

$$\ell(\hat{\theta}|\boldsymbol{x}) - pen(p, n),$$

which is a general form on which we can write several model selection criteria like Mallows CP and Akaikes AIC.

Table 2.1: Jeffreys' scale of evidence for the interpretation of Bayes Factors (see Equation (2.6)) as presented in Efron and Hastie (2016, p. 245).

Bayes Factor	Evidence for $\mathcal{M}_1$
<1	negative
1-3	barely worthwile
3-20	positive
20-150	strong
>150	very strong

3. Single parameter changepoint detection

In this section we consider the one parameter changepoint problem where there may be more than one changepoint and we make strict assumptions. In Section 3.1 we will establish the assumptions of this section and the language we will use to discuss the multiple changepoint problem. Then in Section 3.2 we develop the likelihood of data under the changepoint model. This likelihood is used further in Section 3.3 as it is a part of the model selection criteria. The model selection criteria are approximations of the Bayes Factor, and are to be maximized. In Section 3.4 we explain how to write the criteria on a form that may be maximized with the changepoint algorithms we detail in Section 4.

3.1. The changepoint model

We have a sequential data set $x_1, x_2, \ldots, x_n$ of realizations of independently distributed random variables $X_1, X_2, \ldots, X_n$. We have $X_1 \sim f_1$ and

$$X_{i+1} \sim f_j, \ j \in \{j, j+1\},$$

where f_1 and f_j are distributions from some given family, then *changepoint* number j is the last realization from f_j. In addition there is a zeroth *fictitious changepoint*, and so the changepoints are x_i such that $i \in \{\tau_0, \ldots, \tau_{m+1}\}$, with

$$0 = \tau_1 < \tau_2 < \cdots < \tau_m < \tau_{m+1} = n \ .$$

In this section the following distributional assumption is made.

Assumption 3.1. The data set $x_1, x_2 \ldots, x_n$ are realizations of $X_j \sim f_j, \ j \in \{j, j+1\}$ such that $f_j = \mathcal{N}(\mu_j, 1)$ and $\mu_j \neq \mu_{j+1}$ for $j \in (1, m+1)$. For all $i \neq j$ also X_i and X_j are independent.

The theory may be applied to other distributions as well, for instance normal distributions with both mean and variance available for estimation, which we consider in Section 5.

In this thesis data point x_i is referred to as the data point at *position* i, or simply as data point i. The τ_js are thus the positions of the changepoints, although they are often simply referred to as the changepoints. The changepoint vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+1})$ segments the data set into $m+1$ intervals where an *interval* is defined as the set of consecutive data points that are realizations from the same distribution. Equivalently the interval is the set of data points $i \in \{\tau_{j-1}+1, \ldots, \tau_j\}$ where $x_i \sim f_j$. The jth interval is of length $n_j = \tau_j - \tau_{j-1}$. For data points on the jth interval the *most recent changepoint* is data point τ_{j-1}, that is for data points $i \in \{\tau_{j-1}+1, \ldots, \tau_j\}$ the most recent changepoint is τ_{j-1}. The *predecessor* is the most recent changepoint

to a changepoint at that location[2], and we call it r, such that

$$r(\tau_j) = \tau_{j-1}.$$

A *changepoint model* is a distributional assumption on f_j, combined with the assumption that there are m changepoints. A common interpretation of what are the model parameters is Assumption 3.2. Another opinion is that a changepoint model is also defined by the changepoint positions $\boldsymbol{\tau}$, and that in the case of Assumption 3.1 the model parameters are simply the elements of the mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{m+1})$. Commonly BIC, as presented in Equation (2.10), is used as a model selection criterion. An argument for Assumption 3.2 is that BIC perform better when we make Assumption 3.2, so we will use this in the thesis.

Assumption 3.2. In the case of Assumption 3.1 the parameters of the changepoint model with m changepoints are

$$\theta = (\mu_1, \ldots, \mu_{m+1}, \tau_1, \ldots, \tau_m).$$

The intuitive meaning of a changepoint is that it indicates a change from one distribution to a new one. Accordingly we define a changepoint to be *internal* if it is non-fictitious and has a non-fictitious consecutive data point. A changepoint that is not internal is *external*, such that $0 = \tau_0$ and $n = \tau_{m+1}$ are external, and the rest are internal. These external changepoints are implicitly assumed to be available for any data set or set of changepoints. Figure 3.1 from Example 1 illustrates the concepts defined in the current section. The choices are traditional and reflect the language in Killick et al. (2012a) and Killick and Eckley (2014), except for the definition of changepoints and the categorization to internal, external, fictitious, and non-fictitious. Our motivation for these definitions is their simplicity, and disambiguation, as the word changepoint is ambiguous in literature.

Example 3.1. The purpose of this example is to illustrate the concepts described so far. We study the data set displayed in Figure 3.1, where $m = 2$ and the underlying distributions are $f_1 = \mathcal{N}(0.2, 1)$, $f_2 = \mathcal{N}(7.6, 1)$, $f_3 = \mathcal{N}(-4.2, 1)$. This data set is much smaller than a typical data set and has more frequent changepoints than what is usually expected. Data point numbers 3, 5, and 7 are changepoints since they are the last points from their distribution. Two equivalent ways to state this is that data points 3, 5, and 7 are changepoints, or that three changepoints have positions 3, 5, and 7. Furthermore the predecessor of 7 is 5, the predecessor of 5 is 3. The solution to the changepoint problem is the underlying distributions. The lower graph in Figure 3.2 represents

[2]In Killick et al. (2012a) $r(i)$ is denoted p_i, and is interchangeably referred to as the predecessor of data point i and as the last previous changepoint of data point i.

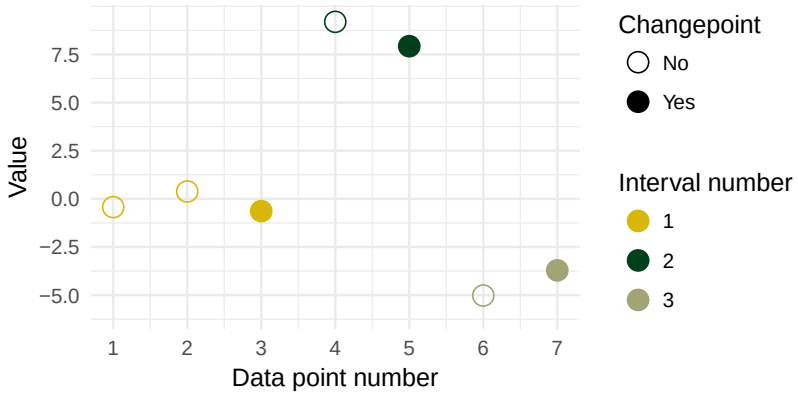


Figure 3.1: Example data consisting of 7 univariate observations from $f_1 = \mathcal{N}(0.2, 1)$,$f_2 = \mathcal{N}(7.6, 1)$,$f_3 = \mathcal{N}(-4.2, 1)$ with $m = 2$ internal changepoints.

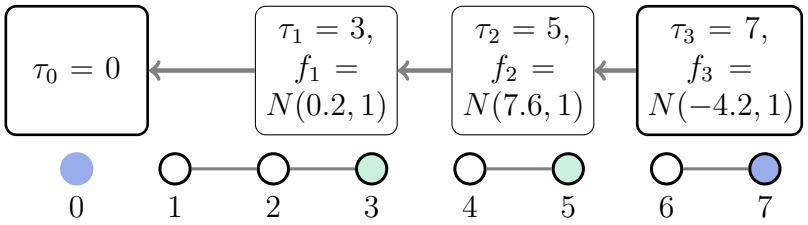


Figure 3.2: Solution represented as a changepoint DAG and data nodes on intervals. On the jth interval where τ_j is the last point, the observations are drawn from f_j. The external and internal changepoints are colored respectively blue and green. The zeroth data point is fictitious which is represented by a lack of outline. The predecessor of x_3 is x_1, while x_3 is the predecessor of x_5 and x_5 is the predecessor of x_7, or equivalently $r(7) = 5, r(5) = 3$, and $r(3) = 0$. The values for τ_j of the first and last nodes in the DAG are predetermined, which is marked by a more prominent outline.

the data points color coded for what type. There also the zeroth fictitious data point is represented.

Even though the model parameters of a changepoint model with mean shift are $\theta = (\tau_1, \ldots, \tau_m, \mu_1, \ldots, \mu_{m+1})$ we will as the literature on changepoint detection algorithms refer to $\boldsymbol{\tau}$ as the *solution* to the changepoint problem. It is then assumed that $(\mu_1, \ldots, \mu_{m+1})$ are the parameters that maximize the likelihood of the changepoint model. In the following section we find expressions for the maximum likelihood parameters.

3.2. Likelihood of the changepoint model

In this section we develop the likelihood under Assumption 3.1 of the changepoint model presented in the previous section. According to Assumption 3.1 the log likelihood of $\boldsymbol{x} = (x_1, \ldots, x_n)$ is for a given value of m

$$\ell(\mu_1, \ldots, \mu_{m+1}, \boldsymbol{\tau}|\boldsymbol{x}) = \sum_{j=1}^{m+1} \ell(\mu_j | x_{\tau_{j-1}+1}, \ldots, x_{\tau_j}, \sigma^2 = 1),$$

where μ_j is the mean of the jth distribution $f_j = \mathcal{N}(\mu_j, 1)$. From Equation (2.2) this becomes

$$\ell(\mu_1, \ldots, \mu_{m+1}, \boldsymbol{\tau}|\boldsymbol{x}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i - \mu_j)^2.$$

To count the number of possible changepoint vectors given values for m and n is equivalent to counting in how many ways n stars can be separated with m bars, given that each bar needs to be next to two stars. This is commonly called the stars and bars problem. It is equivalent to choosing m of the the $n-1$ spaces between the stars without replacement, and thus there are

$$(n-1)\mathrm{C}(m) = \frac{(n-1)!}{m!(n-1-m)!} \tag{3.1}$$

unique $\boldsymbol{\tau}$s.

In order to find the maximum likelihood estimate for the means for a fixed $\boldsymbol{\tau}$ we use that $(x_i - \mu_j)^2 = x_i^2 - 2x_i\mu_j + \mu_j^2$,

$$\begin{aligned}\frac{\partial}{\partial\mu_j}\ell(\boldsymbol{\tau}|\boldsymbol{x}) &= \frac{\partial}{\partial\mu_j}\left(-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i^2 - 2x_i\mu_j + \mu_j^2)\right)\\ &= -\frac{1}{2}\sum_{i=\tau_{j-1}+1}^{\tau_j}(-2x_i + 2\mu_j),\end{aligned}$$

and setting $\frac{\partial}{\partial\mu_j}\ell(\boldsymbol{\tau}|\boldsymbol{x}) = 0$ gives the maximum likelihood estimate $\hat{\mu}_j = \frac{1}{n_j}\sum_{i=\tau_{j-1}+1}^{\tau_j} x_i$ since $\frac{\partial^2}{\partial^2\mu_j}\ell(\boldsymbol{\tau}|\boldsymbol{x}) = -1$. Hence the maximum likelihood with respect to the means for a fixed $\boldsymbol{\tau}$ is

$$\ell(\boldsymbol{\tau}|\boldsymbol{x}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i - \hat{\mu}_j)^2. \tag{3.2}$$

From now on we will use the definition that $n_j = \tau_j - \tau_{j-1}$. Then the likelihood may alternatively be written

$$x_i^2 - 2x_i\mu_j + \mu_j^2\ell(\boldsymbol{\tau}|\boldsymbol{x}) = -\frac{n}{2}\log(2\pi) - (\frac{1}{2}\sum_{i=1}^{n}x_i^2 + \frac{1}{2}\sum_{j=1}^{m+1}n_j\hat{\mu}_j^2), \tag{3.3}$$

as $\sum_{i=\tau_{j-1}+1}^{\tau_j} x_i = n_j\hat{\mu}_j$ and the second term in Equation (3.2) is

$$\begin{aligned}
-\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2 &= -\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i^2-2x_i\hat{\mu}_j+\hat{\mu}_j^2)\\
&= -\sum_{i=1}^{n}x_i^2+2\sum_{j=1}^{m+1}\hat{\mu}_j\sum_{i=\tau_{j-1}+1}^{\tau_j}x_i-\sum_{j=1}^{m+1}n_j\hat{\mu}_j^2\\
&= -\sum_{i=1}^{n}x_i^2+2\sum_{j=1}^{m+1}\hat{\mu}_j(n_j\hat{\mu}_j)-\sum_{j=1}^{m+1}n_j\hat{\mu}_j^2\\
&= -\sum_{i=1}^{n}x_i^2+\sum_{j=1}^{m+1}n_j\hat{\mu}_j^2.
\end{aligned}$$

To find the maximum likelihood when m is fixed we may simply try each of the the finite number of possible $\boldsymbol{\tau}$s. When we compare the likelihoods to find the maximum we only need to look at the difference between the likelihoods, and so we only need to compute the terms $\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2$ to find out which $\boldsymbol{\tau}$ gives the maximal likelihood.

3.3. Model selection

The aim of this section is to find criteria to choose the m that defines the changepoint model described in Assumption 3.2. Then we can proceed to find the parameters μ_j and τ_j by maximum likelihood estimation from Equation (3.2). It is tempting to find m by maximum likelihood estimation too. To see why this does not work we return to Equation (3.2) and see that

$$\max\left(-\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2\right)=0,$$

which is the result when $\tau_j - \tau_{j-1} = 1$. And so

$$\max_m \ell(\tau_1,\ldots,\tau_m|\boldsymbol{x}) = -\frac{n}{2}\log(2\pi),$$

and the maximum likelihood estimate of m would be $\hat{m} = n - 1$. This means that all data points are always changepoints, which is not our desired solution. It is important to note that m is not a parameter in itself, but regulates how many parameters the model contains. We need methods for model selection to determine m. One popular method is to apply Schwarz' BIC directly. Another option is to use the modified Bayesian Information Criterion (mBIC), which is an approximation of the Bayes Factor specifically developed for the changepoint model with data from a normal distribution (Zhang and Siegmund, 2007). These two approaches are detailed in the following sections.

3.3.1. BIC

Under Assumption 3.2 there is a total of $2m+1$ parameters to determine. When the BIC formula in Equation (2.10) is used directly with the maximum likelihood from Equation (3.2) the BIC is

$$-\frac{n}{2}\log(2\pi)-\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2-(2m+1)\log n.$$

However we will use the BIC to select the model with the maximal BIC, and thus terms independent of the model parameters may be omitted from the equation. This gives the commonly used

$$\mathrm{BIC}_1 = -\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2-2m\log n, \tag{3.4}$$

where the indicator 1 denotes that it is the first BIC version for a changepoint model to be introduced in the thesis. However in the changepoint model the data points are in general not identically distributed, so the assumptions of Theorem 2.3 do not hold. The asymptotic consistency of BIC has been established when the changepoint data are independently normal distributed (Yao, 1988), and for a few other special changepoint situations.

3.3.2. mBIC

The modified Bayesian Information Criterion (mBIC) (Zhang and Siegmund, 2007) is under certain conditions optimal in the changepoint setting. It uses some notation we will now introduce. To write that a sequence X_n of random variables is $X_n = O_P(1)$ denotes that X_n is of order less than or equal to 1 in probability (Lehmann, 1999), that is for any $\epsilon > 0$, there exists a finite $M > 0$ and a finite $N > 0$ such that for all $n > N$,

$$Pr(|X_n| > M) < \epsilon.$$

Theorem 3.1. This is Theorem 1 from Zhang and Siegmund (2007) re-parametrized and slightly simplified. The theorem states that under certain priors on the model parameters that represent no information, and under Assumptions 3.1 and 3.2,

$$\begin{aligned} \log \frac{P(\boldsymbol{x}|\,\mathcal{M}_m)}{P(\boldsymbol{x}|\,\mathcal{M}_0)} = \frac{1}{2}\sum_{j=1}^{m+1}(\hat{\tau}_j - \hat{\tau}_{j-1})(\hat{\mu}_j - \frac{1}{n}\sum_{i=1}^{n} x_i)^2 \\ - \frac{1}{2}\left(\sum_{j=1}^{m+1}\log(\hat{\tau}_j - \hat{\tau}_{j-1}) + (1-2m)\log n\right) + O_p(1), \end{aligned} \tag{3.5}$$

where the $\hat{\tau}_j$s are the positions in $\hat{\boldsymbol{\tau}} = (\hat{\tau}_1, \ldots, \hat{\tau}_m)$ such that

$$\hat{\boldsymbol{\tau}} = \arg\max_{\boldsymbol{\tau}} \frac{1}{2}\sum_{j=1}^{m+1}(\tau_j - \tau_{j-1})(\hat{\mu}_j - \frac{1}{n}\sum_{i=1}^{n} x_i)^2 \tag{3.6}$$

for a given m, and $\mathcal{M}_0$ is the changepoint model with $m = 0$. This is the result of approximating the Bayes factor when n approaches infinity while τ_j/n approaches a constant. A solution sketch is found in the web appendix of Zhang and Siegmund (2007).

Loosely speaking the remainder term being $O_p(1)$ means that it is smaller than some value not depending on n. On the other hand the rest of the expression grows with n. And so when n approaches infinity the term that is $O_p(1)$ becomes negligible. Thus the mBIC is

$$\begin{aligned} D_1(m) = \frac{1}{2}\sum_{j=1}^{m+1}(\hat{\tau}_j - \hat{\tau}_{j-1})(\hat{\mu}_j - \frac{1}{n}\sum_{i=1}^{n} x_i)^2 \\ - \frac{1}{2}\left(\sum_{j=1}^{m+1}\log(\hat{\tau}_j - \hat{\tau}_{j-1}) + (1-2m)\log n\right), \end{aligned} \tag{3.7}$$

which we labeled D_1 as we will refer to it later. We want the model that maximizes the mBIC, similar to in Section 3.3.1 where we maximize the BIC.

There is no closed form expression for $\hat{\boldsymbol{\tau}}$ for a given m. It is instead convenient to use the expression

$$D_2(m, \boldsymbol{\tau}) = \frac{1}{2} \sum_{j=1}^{m+1} (\tau_j - \tau_{j-1})(\hat{\mu}_j - \frac{1}{n} \sum_{i=1}^{n} x_i)^2 - \frac{1}{2} \left(\sum_{j=1}^{m+1} \log(\tau_j - \tau_{j-1}) + (1 - 2m) \log n \right). \tag{3.8}$$

Here we have replaced $\hat{\tau}_j$ with τ_j. It is tempting to maximize this instead of the approximate Bayes Factor from Theorem (3.5). However for a given m, the $\boldsymbol{\tau}$ that gives $\max_{\boldsymbol{\tau}} D_2(m, \boldsymbol{\tau}) = D_2(m, \tilde{\boldsymbol{\tau}})$ is

$$\tilde{\boldsymbol{\tau}} = \arg\max_{\boldsymbol{\tau}} \left(\frac{1}{2} \sum_{j=1}^{m+1} (\tau_j - \tau_{j-1})(\hat{\mu}_j - \frac{1}{n} \sum_{i=1}^{n} x_i)^2 - \frac{1}{2} \sum_{j=1}^{m+1} \log(\tau_j - \tau_{j-1}) \right).$$

This is different from Equation (3.6). So $\max_{(m,\boldsymbol{\tau})} D_2(m, \boldsymbol{\tau})$ is in general not equal to $\max_m D_1(m)$.

The natural approach to finding $\arg\max_m D_1(m)$ is to find $\hat{\boldsymbol{\tau}}$ from Equation (3.6) for $m = 0, 1, \ldots, n-1$, and choose the model that maximizes Equation (3.5). The modified version (Zhang and Siegmund, 2007) of the Circular Binary Segmentation (CBS) algorithm (Olshen et al., 2004) finds for a given m a $\boldsymbol{\tau}$ such that the likelihood is large, but not necessarily maximal. So when the modified CBS is run for every $m = 1, \ldots, n-1$ and the model with the largest resulting $D_2(m, \boldsymbol{\tau})$ is chosen, it is not guaranteed that the resulting $\boldsymbol{\tau}$ is the maximum likelihood estimate, or that the resulting m maximizes Equation (3.5).

As we have just seen, neither by maximizing $D_2(m)$ or by the natural approach described above are we guaranteed to find the changepoint model that maximizes the mBIC $D_1(m)$ in Equation (3.7). Indeed Truong et al. (2018) states that to find the $(m, \boldsymbol{\tau})$ that maximizes the mBIC is not tractable. In this thesis when we want to find the parameters that maximize the Bayes Factor under the assumptions of Theorem 3.1, we will instead maximize $D_2(m, \boldsymbol{\tau})$. In the rest of this section we will interpret the first and second term of Equation (3.8) in that order. Then we will find a simple expression that is analogous to Equation (3.4); a simple expression that is minimal when $D_1(m, \boldsymbol{\tau})$ in Equation (3.8) is minimal.

3.3.3. mBIC likelihood term

The first term in Equation (3.8) represents the likelihood of the observations under $\mathcal{M}_m$ (Zhang and Siegmund, 2007). We will now ascertain this by comparing it to $\ell(\boldsymbol{\tau}|\boldsymbol{x})$ in Equation (3.2). To see this we write it in detail with $n_j = \tau_j - \tau_{j-1}$ and $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$, and we insert the maximum likelihood

estimates for the μ_js. So the first term of Equation (3.8) is

$$\begin{aligned}
&\frac{1}{2}\sum_{j=1}^{m+1} n_j(\hat{\mu}_j - \bar{x})^2 = \frac{1}{2}\sum_{j=1}^{m+1} n_j(\hat{\mu}_j^2 - 2\hat{\mu}_j\bar{x} + \bar{x}^2) \\
&= \frac{1}{2}\left(\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2 - 2\bar{x}\sum_{j=1}^{m+1} n_j\hat{\mu}_j + \bar{x}^2\sum_{j=1}^{m+1} n_j\right) \\
&= \frac{1}{2}\left(\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2 - 2\bar{x}^2 n + \bar{x}^2 n\right) = \frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2 - \frac{1}{2}\bar{x}^2 n,
\end{aligned}$$

and for a given data set this likelihood is maximized with respect to $\boldsymbol{\tau}$ when $\frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2$ is maximized. Likewise the likelihood from Equation (3.3) is maximized when $\frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2$ is maximized, and so the first term in Equation (3.8) may be said to represent the likelihood $\ell(\boldsymbol{\tau}|\boldsymbol{x})$ in Equation (3.2) maximized with respect to the means. Since

$$\begin{aligned}
&\frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2 - \frac{1}{2}\bar{x}^2 n - \ell(\boldsymbol{\tau}|\boldsymbol{x}) \\
&\frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2 - \frac{1}{2}\bar{x}^2 n - \left(-\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{n} x_i^2 + \frac{1}{2}\sum_{j=1}^{m+1} n_j\hat{\mu}_j^2\right) \\
&= -\frac{n}{2}\bar{x}^2 + \frac{n}{2}\log(2\pi) + \frac{1}{2}\sum_{i=1}^{n} x_i^2,
\end{aligned}$$

we may write (3.8) as

$$\begin{aligned}
\log\frac{P(\boldsymbol{x}|\,\mathcal{M}_m)}{P(\boldsymbol{x}|\,\mathcal{M}_0)} = \ell(\boldsymbol{\tau}|\boldsymbol{x}) - \frac{n}{2}\bar{x}^2 + \frac{n}{2}\log(2\pi) + \frac{1}{2}\sum_{i=1}^{n} x_i^2 \\
- \frac{1}{2}\left(\sum_{j=1}^{m+1}\log(\tau_j - \tau_{j-1}) + (2m-1)\log n\right) + O_p(1),
\end{aligned}$$

where $\ell(\boldsymbol{\tau}|\boldsymbol{x})$ is the likelihood from Equation (3.2).

3.3.4. mBIC penalty term

Zhang and Siegmund (2007) states that the second part of Equation (3.5) corresponds to a penalty. The second part of Equation (3.8) may be

rewritten as

$$
\begin{aligned}
&-\frac{1}{2}\left(\sum_{j=1}^{m+1}\log(\tau_j-\tau_{j-1})+(2m-1)\log n\right)\\
&=-\frac{1}{2}\left(\sum_{j=1}^{m+1}\log(\tau_j-\tau_{j-1})+(2m-1)\log n+(m+1)\log n-(m+1)\log n\right)\\
&=-\frac{1}{2}\left(\sum_{j=1}^{m+1}\log\left(\frac{\tau_j-\tau_{j-1}}{n}\right)+3m\log n\right),
\end{aligned}
$$

such that Equation (3.8) may be expressed as

$$
\begin{aligned}
D_2(m,\boldsymbol{\tau}) = \ell(\boldsymbol{\tau}|\boldsymbol{x}) - \frac{n}{2}\bar{x}^2 + \frac{n}{2}\log(2\pi) + \frac{1}{2}\sum_{i=1}^{n}x_i^2 \\
-\frac{1}{2}\left(\sum_{j=1}^{m+1}\log\frac{\tau_j-\tau_{j-1}}{n}+3m\log n\right).
\end{aligned}
\tag{3.9}
$$

The value of $\log\frac{\tau_j-\tau_{j-1}}{n}$ is always negative. When the changepoints are close together it is a large negative number, and when they are evenly spaced it is a small number. So the penalty is higher when the changepoints are close together. Now there are two extremes. One extreme is that all the changepoints are evenly spaced. Then the $m+1$ segments all have the same value of $\tau_j-\tau_{j-1}$, such that $n/(\tau_j-\tau_{j-1})=m+1$. Thus the maximum with respect to $\boldsymbol{\tau}$ when m and n is fixed is

$$
\max_{\boldsymbol{\tau}}\sum_{j=1}^{m+1}\log\frac{\tau_j-\tau_{j-1}}{n} = -\sum_{j=1}^{m+1}\log(m+1) = -(m+1)\log(m+1).
$$

The other extreme is that all the changepoints are next to each other, that is $\tau_j-\tau_{j-1}=1$. Then for j in 1 to m we get $\log\frac{\tau_j-\tau_{j-1}}{n}=-\log n$. For the last interval we get $\tau_m=m$, such that $\log\frac{\tau_{m+1}-\tau_m}{n}=\log\frac{n-m}{n}$ and

$$
\min_{\boldsymbol{\tau}}\sum_{j=1}^{m+1}\log\frac{\tau_j-\tau_{j-1}}{n} = -m\log n+\log\frac{n-m}{n}\approx -m\log n.
$$

In other words the penalty term when all the changepoints are next to each other becomes

$$
\begin{aligned}
&\min_{\boldsymbol{\tau}}\left(\sum_{j=1}^{m+1}\log\frac{\tau_j-\tau_{j-1}}{n}+3m\log n\right)\\
&=-m\log n+\log\frac{n-m}{n}+3m\log n \underset{n\gg m}{\approx} 2m\log n
\end{aligned}
\tag{3.10}
$$

for a fixed m and n. In comparison

$$\begin{aligned}&\max_{\boldsymbol{\tau}} \left(\sum_{j=1}^{m+1} \log \frac{\tau_j - \tau_{j-1}}{n} + 3m \log n \right) \\ &= -(m+1)\log(m+1) + 3m \log n \underset{n \gg m}{\approx} 3m \log n, \end{aligned} \tag{3.11}$$

when the changepoints are evenly spaced in the data set. This means that the penalty takes on a value between $2m \log n$ and $3m \log n$. Both mBIC and BIC are asymptotically consistent for the changepoint model under Assumption 3.1.

When we find the model such that the modified Bayesian Information Criterion (mBIC) in Equation (3.9) is maximized, the terms independent of the model parameters and m may be omitted so that we get the simplified expression

$$\text{BIC}_2 = -\sum_{j=1}^{m+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - \left(\sum_{j=1}^{m+1} \log \frac{\tau_j - \tau_{j-1}}{n} + 3m \log n \right). \tag{3.12}$$

The last part of the equation is similar to Equation (5) in Zhang and Siegmund (2007), however the expressions are different because they maximize D_1 in Equation (3.7) while we maximize D_2 in Equation (3.8).

3.3.5. mBIC interpretation

In Theorem 2.3 the penalty is set using the dimension p of the parameter space. This regulates how complex or parsimonious the model is. Occam's razor, also called the law of parsimony, dictates that we choose the most parsimonious model that fits the data. The term *degrees of freedom* (df) is used to describe how parsimonious a model is. The data set has a number of degrees of freedom available, and the model requires a given number of degrees of freedom. In this thesis we call the degrees of freedom the model requires the degrees of freedom of the model, such that p in Equation (2.10) represents the degrees of freedom of a model. Another choice is to call the degrees of freedom left after the model is fitted to a data set the degrees of freedom of the model. The principle of parsimony leads the penalty to be scaled by the number of degrees of freedom of the model.

One reason for the debate on what constitutes the model parameters in a changepoint setting is that we want to determine the degrees of freedom and set the penalty accordingly. When there is a debate on what are the model parameters in a changepoint setting, this is partly because we want to determine the degrees of freedom and thus the penalty. As mBIC is optimal on Gaussian changepoint data, ideally we would like BIC_1 to have the same penalty value. And so a penalty adjusted BIC_1 from Equation (3.4) is

$$\text{BIC}_{1,adj} = -\sum_{j=1}^{m+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - (d+1)m \log n, \tag{3.13}$$

where $p = (d+1)m$. We will call the p such that BIC_1 and BIC_2 have the same penalty the *effective degrees of freedom* of the model. We assume that the μ_js contribute 1 effective degree of freedom each to p in this setting, which gives a penalty of $m \log n$ in Equations (3.4) and (3.12). The entire penalty in Equation (3.12) is

$$2m \log n + \sum_{j=1}^{m+1} \log \frac{\tau_j - \tau_{j-1}}{n} + m \log n,$$

and so the penalty for $\boldsymbol{\tau}$ is

$$2m \log n + \sum_{j=1}^{m+1} \log \frac{\tau_j - \tau_{j-1}}{n}.$$

Then from Equation (2.10) we find the effective degrees of freedom for $\boldsymbol{\tau}$ by dividing by $\log n$, that is

$$2m + \frac{\sum_{j=1}^{m+1} \log(\frac{\tau_j - \tau_{j-1}}{n})}{\log n}. \tag{3.14}$$

And then we would like for d in Equation (3.13) to be equal to the effective degrees of freedom per τ_j, that is

$$edf = 2 + \frac{\sum_{j=1}^{m+1} \log(\frac{\tau_j - \tau_{j-1}}{n})}{m \log n}, \tag{3.15}$$

which we label *edf*. We will come back to *edf* in Section 4.2 to see how it depends on n and on the quantity and the positions of the changepoints. The difference between the d in Equation (3.13) and *edf* in the Equation above is that d is a constant determined prior to maximizing the criterion, while *edf* is a property that is computed for a fitted or underlying model.

As we see from Equation (3.9), it is not actually the changepoint locations that contribute effective degrees of freedom to p, but the scaled interval lengths $(\tau_j - \tau_{j-1})/n$. Longer intervals contribute more effective degrees of freedom. Intuitively when the interval $(\tau_{j+1} - \tau_{j-1})/n$ is divided by τ_j that division contributes more effective degrees of freedom when $(\tau_{j+1} - \tau_{j-1})/n$ is a long interval. For instance if $\tau_{j-1} = 1, \tau_j = 2$, and $\tau_{j+1} = 3$ the additional effective degrees of freedom are severely limited, but if $\tau_{j-1} = 10, \tau_j = 110$, and $\tau_{j+1} = 210$, the division affects the fit of 200 data points.

3.4. Optimization problem

In this section we will write the criteria on the form

$$-\left(\sum_{j=1}^{m+1} C(\tau_{j-1}+1,\tau_j)+\beta m\right). \tag{3.16}$$

Since the optimal parameters are those that maximize the chosen criteria, we want to minimize the expression inside the parentheses. The motivation for writing the criteria on this form is that the algorithms for changepoint detection is defined in this way. First we will present the terms and concepts used in the changepoint detection algorithms, and then we will specify how we use these when we maximize our criteria. In Section 3.4.3 a method to aid in the understanding of the algorithms of Section 3.5 is presented.

3.4.1. Optimal cost in changepoint detection algorithms

When a prospective interval starts at data point s and ends at data point t, we will associate with it a cost[3]

$$C(s,t)\ .$$

For instance the interval cost of the jth interval is $C(\tau_{j-1}+1,\tau_j)$. A high value for $C(\cdot,\cdot)$ indicates that the changepoint model is a bad fit to the data in this region. The total cost of with given parameters (m,θ) is

$$\sum_{j=1}^{m+1} C(\tau_{j-1}+1,\tau_j)+q(m)\ ,$$

the sum of the interval costs, and a penalty $q(m)$ for the number of changepoints. As with Equation (3.20) the (m,θ) that gives the minimal total cost for the data set is considered to be optimal. Even though all the parameters in Assumption 3.2 need to be specified to compute the total cost of a data set under Assumption 3.1, $\boldsymbol{\tau}$ is referred to as the *solution* or *prospective solution* of the changepoint problem in the algorithmic literature. An algorithm is considered good if it has low run time and the total cost of the acquired solution is as small as possible. Some exact algorithms that find the optimal solutions under certain requirements are Segmentation Neighborhood (SN), OP, and PELT. A popular algorithm that finds a good solution, in that it has a quite low cost, is Binary Segmentation (BinSeg).

A special case for solution cost is when the penalty term is linear in m, that is $q(m)=m\beta+B$ where β and B are freely chosen parameters. In Section 3.5.2 we see that a linear penalty is a requirement for the OP and PELT algorithms. The penalties of the two methods above are linear, and in the thesis we only consider linear penalties.

[3]The cost $C(s,t)$ is denoted as $C(x_{s:t})$ in Killick et al. (2012a)

Assumption 3.3. The penalty term is $q(m) = m\beta + B$, and is thus linear in m, for some B independent of m and the model parameters.

The prospective solution cost with linear penalty is

$$\begin{aligned} p(t) &= \sum_{i=1}^{m+1} \left(C(\tau_{i-1}+1, \tau_i)\right) + m\beta + B \ , \\ &= \sum_{i=1}^{m+1} \left(C(\tau_{i-1}+1, \tau_i) + \beta\right) - \beta + B \ . \end{aligned} \tag{3.17}$$

The last line of Equation (3.17) is why the linear penalty is particularly easy to work with; the penalty term $q(m)$ becomes a penalty β for each new interval. OP and PELT takes advantage of this to find the optimal solution in a systematic fashion, and therefore require the penalty to be linear.

For a fixed β the optimal solution for a given data set is the one which minimizes $p(t)$ with respect to m and the τ_is. Thus the term $-\beta + B$ outside the summation sign in Equation (3.17) does not affect which solution is chosen, and any B will give the same optimal changepoints. In Killick et al. (2012a) they chose $B = 0$, which gives the intuition that a data set with only one data point will have $p(1) = C(1, 1)$, the cost of only the first data point. Another natural choice is $B = \beta$ such that the total cost is the sum of the costs and a total penalty of $(m + 1)\beta$. We choose the first option in this thesis and we then define the prospective total cost of a solution for $x_1, \ldots, x_t$ as

$$p(t) = \sum_{i=1}^{m+1} \left(C(\tau_{i-1}+1, \tau_i) + \beta\right) - \beta \ , \tag{3.18}$$

where m and the τ_is are defined according to the prospective solution. Accordingly we define the total cost of the optimal ('final') solution on the same data as

$$F(t) = \min p(t) = \min \left(\sum_{i=1}^{m+1} \left(C(\tau_{i-1}+1, \tau_i) + \beta\right) \right) - \beta \ . \tag{3.19}$$

This means that $F(t) - p(t) \geq 0$ for any solution cost $p(t)$.

3.4.2. Optimal cost with the model selection criteria

To maximize the BIC from Equation (3.4) is equivalent to minimize its negative. With $\theta = (\tau_1, \ldots, \tau_m, \mu_1, \ldots, \mu_{m+1})$ we get that the optimal model m_1 and parameter vector $\hat{\theta}_1$ with respect to BIC_1 is

$$\begin{aligned} (m_1, \hat{\theta}_1) &= \arg\max_{m,\theta} \text{BIC}_1 = \arg\min_{m,\theta} (-\text{BIC}_1) \\ &= \arg\min_{m,\theta} \left(\sum_{j=1}^{m+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 + 2m \log n \right) , \end{aligned} \tag{3.20}$$

which we in turn may write as

$$(m_1, \hat{\theta}_1) = \arg\min_{m,\theta} \left(\sum_{j=1}^{m+1} C(\tau_{j-1}+1, \tau_j) + q(m), \right)$$

with

$$C(\tau_{j-1}+1, \tau_j) = \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2, \tag{3.21}$$
$$\text{and } q(m) = 2m \log n, \text{ such that } \beta = 2 \log n.$$

In that sense $C(\tau_{j-1}+1, \tau_j)$ is a cost that we want to minimize. We also want to minimize the penalty term $q(m) = 2m \log n$.

Likewise we may write the optimal model m_1 and parameter vector $\tilde{\theta}_2$ with respect to BIC_2 as

$$\begin{aligned}(m_2, \tilde{\theta}_2) &= \arg\min_{m,\theta} \left(\sum_{j=1}^{m+1} \left(\sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 + \log \frac{\tau_j - \tau_{j-1}}{n} \right) + 3m \log n \right) \\ &= \arg\min_{m,\theta} \left(\sum_{j=1}^{m+1} C(\tau_{j-1}+1, \tau_j) + q(m) \right),\end{aligned}$$

this time with

$$q(m) = 3m \log n, \text{ such that } \beta = 3 \log n, \text{ and} \tag{3.22}$$
$$C(\tau_{j-1}+1, \tau_j) = \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 + \log \frac{\tau_j - \tau_{j-1}}{n}.$$

The motivation for including $\log(n_j/n)$ in C instead of in q is to make sure that $q(m)$ is not a function of the x_is and the model parameters. This highlights a subtle difference between the term penalty as it is used in algorithmic settings and in the setting of statistical model selection criteria. In model selection penalty is whatever acts as a counter weight to the model complexity, that is, the penalty regulates how parsimonious the model is. But in the algorithms for changepoint detection the penalty is simply the terms that are independent of the data and not a part of the cost functions for the intervals.

Example 3.2. In Example (3.1) we knew which points were from which distribution. This time only the data set in Table 3.1 and Figure 3.3 is assumed known. We choose to maximize BIC_1, so we use Equation (3.21) for the cost and penalty. This means that $n = 7$, and

$$\beta = 2 \log(n) = 2 \log(7) = 3.89 \, .$$

Two natural guesses at the solution are $\boldsymbol{\tau}^{(1)} = (0, 3, 7)$ and $\boldsymbol{\tau}^{(2)} =$

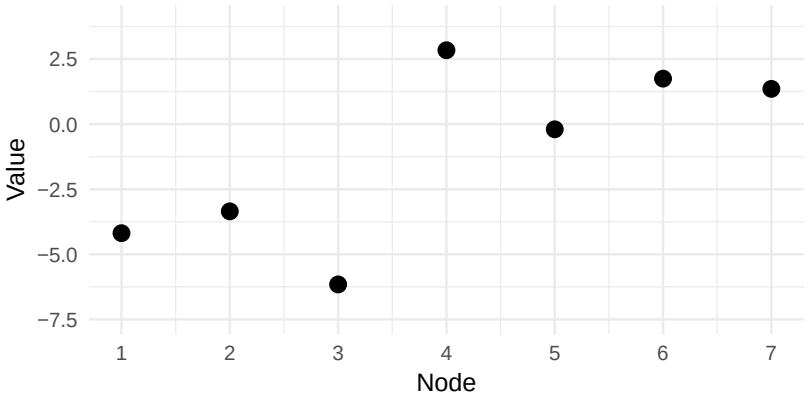


Figure 3.3: Example data consisting of 7 univariate observations from unknown standard normal distributions.

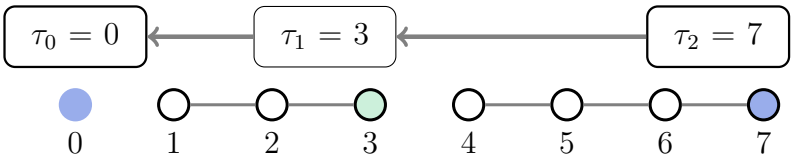


Figure 3.4: Prospective solution $\tau^{(1)} = (0, 3, 7)$.

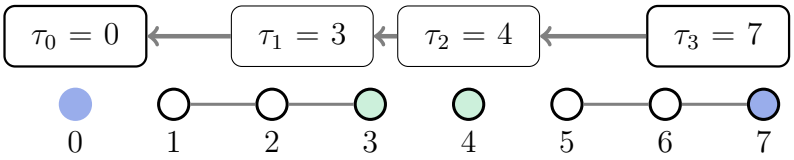


Figure 3.5: Prospective solution $\tau^{(2)} = (0, 3, 4, 7)$.

Table 3.1: Changepoint data set from standard normal distributions used in Example 3.2.

i	1	2	3	4	5	6	7
x_i	-4.19	-3.35	-6.17	2.84	-0.197	1.75	1.36

$(0, 3, 4, 7)$, which are displayed in Figures 3.4 and 3.5. For prospective solution $\boldsymbol{\tau}^{(1)}$ the length of the two intervals are $n_1^{(1)} = 3$ and $n_2^{(1)} = 4$, while for $\boldsymbol{\tau}^{(2)}$ the three interval lengths are $n_1^{(2)} = 3$, $n_2^{(2)} = 1$, $n_3^{(2)} = 3$. Using $\hat{\mu}_i = \frac{1}{n_i} \sum_{k=\tau_{i-1}+1}^{\tau_i} x_k$ then for both solutions $\hat{\mu}_1 = -4.56$. Accordingly for $\boldsymbol{\tau}^{(1)}$ we get $\hat{\mu}_2^{(1)} = 1.44$, and for $\boldsymbol{\tau}^{(2)}$ we get $\hat{\mu}_2^{(2)} = 2.84$ and $\hat{\mu}_3^{(2)} = 0.970$. In both cases the first interval cost is

$$C_e(1, 3) = (-4.19 + 4.56)^2 + (-3.35 + 4.56)^2 + (-6.17 + 4.56)^2 = 4.19 \, .$$

Computing the rest in this fashion we get

$$\begin{aligned} p_1(7) &= 2\beta_e + C_1(1,3) + C_1(4,7) \\ &= 2 \cdot 3.89 + 4.15 + 4.74 && = 16.7\ , \\ p_2(7) &= 2\beta_e + C_2(1,3) + C_2(4,4) + C_2(5,7) \\ &= 3 \cdot 3.89 + 4.15 + 1.84 + 2.12 && = 19.8\ , \end{aligned}$$

such that $\boldsymbol{\tau}^{(1)}$ gives a lower total cost than $\boldsymbol{\tau}^{(2)}$ and is thus the better solution . We see that introducing data point 4 as a changepoint reduce the sum of the interval costs since $3.96 = C_2(4,4) + C_2(5,7) < C_1(4,7) = 4.74$. However the reduction is smaller than the penalty. The sum of the interval costs usually decrease with increasing number of changepoints, which illustrates the need for a penalty. Indeed whenever $\beta = 0$ the optimal choice is to let every data point be a change point. The underlying distributions used to generate the examples were $f_1 = \mathcal{N}(-5.59, 1)$, $f_2 = \mathcal{N}(3.42, 1)$, $f_3 = \mathcal{N}(0.95, 1)$, with solution $\tau = (0, 3, 4, 7)$, which is equal to the prospective solution $\tau^{(2)}$. In general it is the case that the correct solution may not give the lowest total cost for an observed data set.

3.4.3. Changepoint DAG

A powerful data science tool is to represent components in problem solving as graphs which resemble flowcharts. A Directed Acyclic Graph (DAG) is a graph with finite number of nodes, where the paths between nodes are directed, and there are no cycles, such that any DAG may be topologically ordered. In this section we have applied the concept to visualize a prospective set of model parameters, which we will continue to refer to as a solution.

A DAG can be constructed such that it carries all the relevant information on the truth as we imagine it or on a suggested solution. In such a DAG each node corresponds to a changepoint. Each node points to the node of its predecessor (Figure 3.1). Since there is one changepoint per interval each node in the DAG also corresponds to one interval. Node i should carry information on changepoint τ_i. Additionally it may contain information on the distribution of the data points on its interval.

The first node in the DAG is the node such that no other node points to it. In this thesis we refer to the first node in the DAG as node number $m + 1$, which corresponds to the last changepoint which has data point number $\tau_{m+1} = n$. For ease of computation and thinking we defined a 0th fictional data point with no value $x_0 \in \emptyset$, which also is defined as a changepoint. This results in a node number $i = 0$ in the DAG which has no observations and no likelihood function. This makes the end node in the DAG well defined, and acts as an aid indicating where the DAG terminates.

When a solution is in the form of such a DAG the total cost is the sum of the costs of each node in the DAG and a penalty for the number of nodes.

Accordingly the cost of each node with BIC_1 is minus twice the log likelihood of the observations in the corresponding interval. As the true DAG is not known, the positions of its nodes, and the distribution indicated on each node must be estimated from the data. The problem of finding the optimal set of changepoints thus corresponds to finding the optimal changepoint DAG.

Example 3.3. The data from Example 3.1 is illustrated in Figure 3.1. The solution is illustrated as a DAG and intervals in Figure 3.2. Since $m = 2$ and $n = 7$ it is predetermined that $\tau_0 = 0$ and $\tau_{m+1} = \tau_3 = n = 7$. This makes the start and end nodes of the solution DAG easy to identify when implementing an algorithm.

3.5. Algorithms

Algorithms are evaluated on their run time, commonly using big O notation. In *Big-O notation* $g(n) = O(f(n))$ means that there exist an $M > 0$ and a $n_0 > 0$ such that $|g(n)| \leq Mf(n)$ for all $n \geq n_0$ (Knuth, 1976). A changepoint detection algorithm is a *pruning* of another algorithm if it finds the same solution in less computations and is an alteration of the other algorithm. Binary Segmentation (BinSeg) is an algorithm that finds a solution in $O(nlog(n))$ time when the data set is of length n. It does not find the optimal solution, but one that has a quite low total cost. It is well established in multiple changepoint detection due to the speedy runtime, and we will come back to it in Section 3.5.1.

An algorithm that finds the optimal solution is *Segmentation Neighborhood* (SN) (Auger and Lawrence, 1989). For each m from 0 to some upper limit M set by the user it computes all the total costs of the $(n-1)!/(m!(n-1-m)!)$ possible solutions. Then it returns the solution with the lowest total cost among the optimal solutions for each m. Accordingly it runs in $O(mn^2)$ time. The pruning pruned Dynamic Programming Algorithm (pDPA) (Rigaill, 2010) reduces the run time to $O(n \log n)$, but only allows for one parameter to be estimated. The Algorithm *Optimal Partitioning* (OP) (Jackson et al., 2005) also finds the optimal solution, and it is detailed in Section 3.5.2. It also computes all the possible solutions and runs in $O(n^2)$ time, but has the requirement that the cost $q(m)$ must be linear in m. PELT is a pruning of OP, but with an additional requirement on the cost function. PELT has a runtime of $O(n)$ under optimal conditions, although in the worst case it is identical to OP. We consider PELT in Section 3.5.3. Another pruning of OP is *Functional Pruning Optimal Partitioning* (FPOP) (R. Maidstone, 2014) which runs in $O(n)$ time, and where the method of pruning is similar to that used in pDPA. In FPOP only one parameter may be estimated, but with an adjustment that we make in Section 5 PELT may estimate multiple parameters. The algorithm *Changepoint Detection for a Range of Penalties* (CROPS) (Haynes et al., 2017a) finds the solutions given penalties in a continuous range by running some changepoint detection algorithm several times. The runtime of CROPS and the properties of the CROPS solution are thus determined by which algorithm is used.

3.5.1. Binary Segmentation

Binary Segmentation (BinSeg) is a popular algorithm in offline changepoint detection. For a data set $x_1, \ldots, x_n$ introducing a changepoint at x_t is a binary segmentation of the data set into $x_1, \ldots, x_t$ and $x_{t+1}, \ldots, x_n$, hence the name of the algorithm. With a linear penalty from Assumption 3.3, the total cost of a data set when splitting $x_1, \ldots, x_n$ at t is

$$C(1, s) + C(s + 1, n) + \beta \, .$$

BinSeg suggests a candidate changepoint position t as the position that gives minimum cost of the segments

$$s = \arg\min_t \left(C(1,t) + C(t+1,n) \right) \ .$$

Then it classifies x_s to be a changepoint if

$$\left(C(1,s) + C(s+1,n) \right) + \beta < C(1,n) \ .$$

If the data set is segmented, BinSeg is in turn applied to each of the segments. Applying a binary segmentation to the data set recursively, the BinSeg partitioning of the data set is found. Commonly the maximum number of internal changepoints the algorithm may find is restricted, to guarantee that BinSeg terminates before it has split the data set into n segments each consisting of only one point. This is a so called greedy approach as the algorithm makes the locally optimal choice of reducing the total cost as much as possible, and in general it does not find the globally minimal total cost.

Example 3.4. In this example we apply BinSeg to the data set in Figure 3.3 and Table 3.1 with penalty $\beta = 2\log(n) = 2\log(7)$. First we compute

$$C(1,t) + C(t+1,7)$$

for every binary segmentation of the data set. This is displayed in Table 3.2. Then $s = 3$ since $C(1,3) + C(4,7) = 8.90$ is the smallest cost. Because $C(1,7) = 70.6 > 12.8 = C(1,3) + C(4,7) + 2\log(7)$, the binary segmentation is accepted, and x_3 is labelled a changepoint.

Table 3.2: Costs computed in step 1 of applying the BinSeg algorithm to the data in Table 3.1.

t	1	2	3	4	5	6
$C(1,t) + C(t+1,7)$	59.8	51.2	8.90	47.3	50.4	63.4

As a second step we look for a binary segmentation of x_1, x_2, x_3. First we compute $C(1,1) + C(2,3) = 3.94$, $C(1,2) + C(3,3) = 0.351$ and $C(1,3) = 4.15$. Thus the best segmentation is for $s = 2$, but $C(1,3) = 4.15 < 4.24 = C(1,2) + C(2,3) + 2\log(7)$ so no changepoint is accepted. As we have not yet searched for a binary segmentation of x_4, x_5, x_6, x_7 we proceed with this. Among the segmentation costs in Table 3.3, $C(4,4) + C(5,7) = 2.12$ is smallest, so now $s = 4$. However since $C(4,7) = 4.7 < 6 = C(4,4) + C(5,7) + 2\log(7)$ this binary segmentation is not accepted either, and the BinSeg solution is simply $\tau = (0,3,7)$. The total cost is $2\log(7) + C(1,3) + C(4,7) = 12.8$, which is equal to the minimum total cost from Equation 3.19 only if $\tau = (0,3,7)$ is the optimal solution.

Table 3.3: Costs computed in step 3 of applying the BinSeg algorithm to the data in Table 3.1.

t	4	5	6
$C(4,t)+C(t+1,7)$	2.12	4.69	4.73

BinSeg may be used with either of Equations (3.4) or (3.12) by using their respective cost functions in Equations (3.21) and (3.22). However since BinSeg does not find the optimal solution, the solution BinSeg finds is in general not the one that maximizes the BIC or mBIC, but simply a solution for which they are large.

The BinSeg algorithm may be modified to account for a non-linear penalty term $q(m)$ by making it work with the entire data set at once. The first step would be the same. After a changepoint has been confirmed such that there are currently m internal changepoints it would select the additional new binary segmentation that would give the highest reduction in total cost. Then it would terminate with m internal changepoints if the reduction was larger than $q(m+1)-q(m)$, or else accept the proposed changepoint and continue.

3.5.2. Optimal Partitioning

OP is an exact algorithm which finds the optimal set of changepoints when the penalty is linear, and runs in $O(n^2)$ time in its basic form. To understand PELT it is important to understand OP, as PELT is simply a version of OP where superfluous computations are omitted. A schematic view of OP based on the presentation in Killick et al. (2012a) is displayed in Algorithm 1. Using a double for-loop the algorithm iterates through every possible partitioning of the data set. The value p and the list F contain respectively the prospective and optimal total costs from Equations (3.18) and (3.19), and $r(t)$ is the predecessor to x_t, which is defined in Section 3 as the most recent changepoint to a changepoint at t.

The outer for-loop at line number 2 in Algorithm 1 ensures that we first find the optimal total cost of $\{x_1\}$, then of $\{x_1, x_2\}$, and so on. In lines 3 through 10 the goal is to find the optimal previous changepoint at data point t, and save it in the vector r. This can be thought of as creating multiple suggestions to solution DAGs. For each increment of t the question *If data point t is a changepoint, which data points belong on the t-interval?* or equivalently *If data point t is a changepoint, where shall the equivalent solution DAG node point to?* is answered. By definition the last point in the data set is a changepoint. In lines 12 through 18 the changepoints are found by first adding data point n, then the data point which n points to, and so on. In the inner for-loop at lines 4 through 10 every single previous point is tested for being the optimal last changepoint. This is not always necessary, and is where PELT omits superfluous computation and improves the run time.

Algorithm 1: Optimal Partitioning Algorithm. Through a nested for-loop we iterate through all the possible partitions of the data set Y. The final estimate of the series of changepoints is kept in $\boldsymbol{\tau}$, and $F(t)$ is the final cost from data point 0 to data point t. The only functions here are *Sort* and the cost function C, the other entities represent scalars or vectors.

input : $Y = (y_1, \ldots, y_n)$, n=length(Y), β, $C(\cdot)$
output: $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+1})$

/* Initialize final total cost at zeroth node */
$F(0) = -\beta$
for $t \leftarrow 1$ **to** n **do**
  $F(t) = \infty$
  /* For each data point y_t find best previous changepoint y_s */
  for $s \leftarrow 0$ **to** $t-1$ **do**
    /* Calculate prospective total cost to t via s */
    $p = F(s) + C(s+1, t) + \beta$
    /* If reduction made by going via s */
    if $p < F(t)$ **then**
      /* Record new estimate for $F(t)$ */
      $F(t) = p$
      /* Record that best previous changepoint at t is s */
      $r(t) = s$
    end
  end
end
/* Build vector $\boldsymbol{\tau}$ from r */
changepoint = n
$i = 1$
while changepoint $\neq 0$ **do**
  $\boldsymbol{\tau}$(i) = changepoint
  changepoint = r(changepoint)
  $i = i + 1$
end
/* Now we have $\boldsymbol{\tau} = (\tau_{m+1}, \tau_m, \ldots, \tau_1)$, so we reverse it */
$\boldsymbol{\tau}$ = Sort($\boldsymbol{\tau}$)

The set of considered data points increasing incrementally can conceptually be thought of as time progressing and revealing one more observation for every increment of t. The main purpose is that being systematical in this

fashion allows for computation preserving memoisation; the optimal (minimal) total cost $F(t_1)$ is computed once, and used to determine the optimal total cost $F(t)$ at later steps where $t > t_1$. Another benefit is that the algorithm may very well be implemented such that t represents time, and that t is incremented only when another observation has been measured in the real world. Some challenges to this approach are discussed in Section 6.

To explain why it is an absolute requirement for OP that the total penalty $q(m) = \beta m$ in $F(\cdot)$ must be linear in the number of changepoints m we return to thinking of t as time. Then at time t any information that stems from $\{x_{t+1}, \ldots, x_n\}$ is off limits. From Equation (3.18) with a given $\boldsymbol{\tau}$ we have

$$\begin{aligned} p(t) &= \sum_{i=1}^{m+1} \left(C(\tau_{i-1}+1, \tau_i) + \beta\right) - \beta \ , \\ p(t) &= \sum_{i=1}^{m} \left(C(\tau_{i-1}+1, \tau_i) + \beta\right) - \beta + C(\tau_m + 1, \tau_{m+1}) + \beta \ , \\ p(t) &= p(t-1) + C(\tau_m + 1, \tau_{m+1}) + \beta \ . \end{aligned}$$

Inserting $\tau_m = r(t)$ and $\tau_{m+1} = t$ we get

$$p(t) = p(t-1) + C(r(t)+1, t) + \beta \ . \tag{3.23}$$

Using a penalty non-linear in the number of changepoints corresponds to using the inferred patterns of later observations to influence how earlier observations are interpreted. In the OP approach the total cost

$$C(\tau_{i-1}+1, \tau_i) + \beta$$

of node i in a prospective solution DAG is determined when the node is constructed. Since the penalty β used must be the same when constructing all the solution DAG nodes for the changepoints found to minimize (3.19). If for instance the penalty differed so that β_i grew as a function of i, the first changepoints would be relatively closer than the last changepoints. Since there are m nodes in the solution DAG, thus the total penalty $f(m)$ must be linear in the number of changepoints $f(m) = m\ \beta + B$.

Table 3.4: The values of $F(t)$ and $r(t)$ at line 12 before vector $\boldsymbol{\tau}$ is built.

t	0	1	2	3	4	5	6	7
$r(t)$		0	0	2	3	3	3	6
$F(t)$	-3.89	0	4.72	10.447	16.18	18.09	20.66	26.39

Example 3.5. In this first example we illustrate what happens in lines 12 through 19 of Algorithm 1 in order to show the purpose of generating $F(t)$ and $r(t)$ in lines 1 through 10. At line 12 we have two vectors $F(t)$ and $r(t)$. For a given data set these values are as displayed in Table 3.1.

Figure 3.6: Representation of $r(t)$ from 3.4 above. Below is resulting partitioning $\boldsymbol{\tau}$. The $n = 7$th data point is a changepoint. Since $r(7) = 6$, also

The vector $r(t)$ of predecessors can be visualized as in Figure 3.6. The nth data point is always a changepoint. Since $n = 7$ and $r(7) = 6$, $r(6) = 3$, and $r(3) = 2$, the changepoints are $\boldsymbol{\tau} = [0, 2, 3, 6, 7]$. In the graph this is found by going along the paths from data point 7 to data point 0. This is also what is done in Algorithm 1 lines 12 through 18. In the last line of Algorithm 1 the vector is reversed in order for it to contain the non fictitious changepoints in increasing order.

Example 3.6. In this example we reuse the data set from Example 3.2 and use OP to find the optimal solution. The data is found in Table 3.1 and Figure 3.3. The penalty and interval cost function used are the same here as in Example 3.2, that is

$$\beta = 2\log(7)$$

and

$$C(s+1, t) = \sum_{k=s+1}^{t} (x_k - \hat{\mu})^2 .$$

In line 1 of Algorithm 1 the vector F of final costs is initialized. When $t = 1$ the only choice for the predecessor of $t = 1$ is $r(t = 1) = 0$ since there is only one previous data point. When t is incremented to $t = 2$, either $r(2) = 0$ or $r(2) = 1$. So the final cost when the data set is only data point 1 is $F(1) = 0$ from Equation (3.23).

We denote by $p_c(t)$ the prospective cost of $\{x_1, \ldots, x_t\}$ when $r(t) = c$. Then prospective costs $p_0(2)$ and $p_1(2)$ of the data set $\{x_1, x_2\}$ where respectively $r(2) = 0$ or $r(2) = 1$ are also computed using (3.23). When computing these we benefit from having previously computed $F(0) = -3.89$ and $F(1) = 0$. All we are left with is to find $C(1, 2) = 0.351$ and $C(2, 2) = 0$ respectively to find that $p_0(2) = F(0)+C(1, 2)+\beta = 0.35$ and $p_1(2) = F(1)+C(2, 2)+\beta = 3.89$. Since $p_0(2) < p_1(2)$ then $r(2) = 0$. This computation is made for every increment of t. For instance at $t = 4$ and $t = 5$ the computation is as displayed in Table 3.5. This computation for all ts give the values displayed in Table 3.7. Performing the same operation as in Example 3.5 for building the vector $\boldsymbol{\tau}$ produces Figure 3.7. When representing $r(t)$ as a graph the optimal set of changepoints for a dataset of length t is find on the path from node t to node 0.

Since OP finds the optimal solution and both the BIC and mBIC in Equations (3.4) and (3.12) have linear penalties OP may be used to find the model and parameters that maximize the expressions. But computing prospective cost for every possible predecessor $r(t)$ is cumbersome and thus

Table 3.5: Values needed to find $r(4)$ with OP, most notably prospective costs $p_s(4)$ at $t = 4$ when the predecessor is at s. Lowest cost is $p_3(4)$, so $r(4) = 3$. Data from Example 3.6.

s	$F(s)$	$C(s+1,4)$	$p_s(4)$
0	-3.89	45.3	45.3
1	0	42.4	46.3
2	4.03	40.5	48.4
3	9.67	0	13.6

Table 3.6: Values needed to find $r(5)$ with OP, most notably prospective costs $p_s(5)$ at $t = 5$ when the predecessor is s. Lowest cost is $p_4(5)$, so $r(5) = 4$. The data is from Example 3.6.

s	$F(s)$	$C(s+1,5)$	$p_s(5)$
0	-3.89	50.3	50.3
1	0	45.5	49.3
2	4.03	41.9	49.8
3	9.67	4.61	18.2
4	15.4	0	15.4

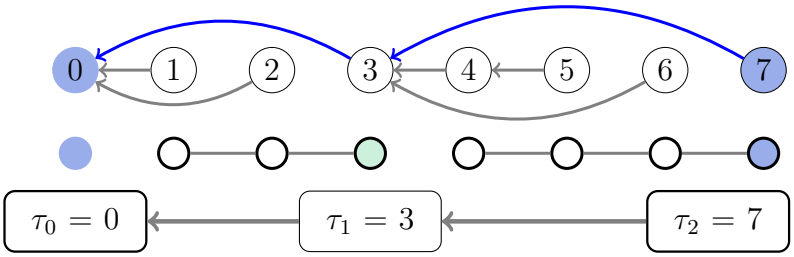


Figure 3.7: These three graphs represent the solution found by OP in Example 3.6 and by PELT in Example 3.7. The top graph is $r(t)$ from Table 3.7, where for instance nodes 4, 6 and 7 point to node 3 because $r(4) = r(6) = r(7) = 3$. The path marked in dark blue indicates the elected solution.

computationally heavy. In the next section we see that the PELT algorithm provides a rule for which data points can be omitted when finding $r(t)$.

Table 3.7: The values of $F(t)$ and $r(t)$ at lines 12 before vector $\boldsymbol{\tau}$ is built in Example 3.6.

t	0	1	2	3	4	5	6	7
$r(t)$		0	0	0	3	4	3	3
$F(t)$	-3.89	0	4.03	9.67	15.40	21.13	23.81	25.65

3.5.3. Pruned Exact Linear Time

The algorithm for PELT is given in Algorithm 2 and is very similar to the OP Algorithm 1. The difference is that the inner for loop in line 5 of Algorithm 2 only iterates through some of the previous data point numbers, and not all. In order to do this the considered data points must be implemented as some type of set, for instance a vector. The *set of consideration* is the *s.set* in Algorithm 2, and at line 16 of the algorithm when $t = t_2$ it is all t_1s such that $0 \leq t_1 < t_2$ that are possible optimal predecessors $r(t_3)$ of later data points t_3 with $t_2 < t_3$. Just before t is incremented (in lines 12 through 17 of Algorithm 2) the current data point number t is not simply appended to the set of consideration as in OP, but also some data point numbers are removed from the set. In this setting to *prune* t means to remove t from the set of consideration. Those pruned at outer loop number $t = t_2$ are data points numbers t_1 such that

$$F(t_1) + C(t_1 + 1, t_2) \geq F(t_2) \,, \tag{3.24}$$

where $t_1 < t_2 < t_3$.

For the pruning not to remove the optimal chain of changepoints from the considered set, the cost function requirement must hold for the cost function, namely

$$C(t_1 + 1, t_2) + C(t_2 + 1, t_3) \leq C(t_1 + 1, t_3). \tag{3.25}$$

We will see in Section 5.3.1 that this always holds when the interval cost is the negative log likelihood. Equations (3.25) and (3.24) are the equations central to the PELT algorithm, and I refer to them respectively as the *cost function requirement* and the *pruning condition* for PELT.

When applying PELT in practice we make sure to chose a cost function such that the assumption in Equation (3.25) holds everywhere, and prune whenever (3.24) also holds. A proof that when both hold data point t_1 can never be the optimal predecessor of data point t_3 is stated in Killick et al. (2012b). The proof is restated in this thesis under Theorem 3.2.

Theorem 3.2. Whenever (3.24) and (3.25) both hold for $t_1 < t_2 < t_3$ then data point number t_1 is not the optimal estimate for the predecessor of t_3.

Proof. First we add $C(t_2, t_3)$ on both sides of the pruning condition from

Equation (3.24), that is

$$F(t_1) + C(t_1 + 1, t_2) \geq F(t_2) \ ,$$
$$F(t_1) + C(t_1 + 1, t_2) + C(t_2 + 1, t_3) \geq F(t_2) + C(t_2 + 1, t_3) \ .$$

Then use the cost requirement from Equation (3.25) to get

$$\begin{aligned} F(t_1) + C(t_1 + 1, t_3) &\geq F(t_1) + C(t_1 + 1, t_2) + C(t_2 + 1, t_3) \\ &\geq F(t_2) + C(t_2 + 1, t_3) \ , \\ p_{t_1}(t_3) = F(t_1) + C(t_1 + 1, t_3) + \beta &\geq F(t_2) + C(t_2 + 1, t_3) + \beta = p_{t_2}(t_3) \ , \end{aligned}$$

where $p_a(t_3)$ is the prospective total cost at t_3 of a solution where $a = r(t_3)$. The optimal estimate for the predecessor $r(t_3)$ is the previous changepoint which has minimal prospective cost $p(t_3)$. Since the prospective total cost $p_{t_1}(t_3)$ at t_3 when last previous changepoint to t_3 is t_1 is greater than or equal to the prospective cost $p_{t_2}(t_3)$ when t_2 is last previous changepoint, t_1 can be never be the uniquely best predecessor of t_3 in the solution DAG. □

As with BinSeg and OP we would like to find the model and the parameters that maximize BIC_1 and BIC_2 in Equations (3.4) and (3.12). Since both result in linear penalties, the penalties are no problem. But we need to check that the resulting cost functions in Equations (3.21) and (3.22) fulfill the cost function requirement in Equation (3.25).

First we check the cost function for BIC_1. The cost functions are for $x_{t_1+1}, \ldots, x_{t_2}, x_{t_2+1} \ldots, x_{t_3}$

$$\begin{aligned} C(t_1 + 1, t_2) &= \sum_{i=t_1+1}^{t_2} (x_i - \frac{1}{t_2 - t_1} \sum_{j=t_1+1}^{t_2} x_j)^2, \\ C(t_2 + 1, t_3) &= \sum_{i=t_2+1}^{t_3} (x_i - \frac{1}{t_3 - t_2} \sum_{j=t_2+1}^{t_3} x_j)^2, \\ C(t_1 + 1, t_3) &= \sum_{i=t_1+1}^{t_3} (x_i - \frac{1}{t_3 - t_1} \sum_{j=t_1+1}^{t_3} x_j)^2 \\ &= \sum_{i=t_1+1}^{t_2} (x_i - \frac{1}{t_3 - t_1} \sum_{j=t_1+1}^{t_2} x_j)^2 + \sum_{i=t_2+1}^{t_3} (x_i - \frac{1}{t_3 - t_1} \sum_{j=t_2+1}^{t_3} x_j)^2, \end{aligned}$$

and since

$$\begin{aligned} C(t_1 + 1, t_2) &\leq \sum_{i=t_1+1}^{t_2} (x_i - \frac{1}{t_3 - t_1} \sum_{j=t_1+1}^{t_2} x_j)^2, \text{ and} \\ C(t_2 + 1, t_3) &\leq \sum_{i=t_2+1}^{t_3} (x_i - \frac{1}{t_3 - t_1} \sum_{j=t_2+1}^{t_3} x_j)^2 \end{aligned}$$

then also

$$C(t_1 + 1, t_2) + C(t_2 + 1, t_3) \leq C(t_1 + 1, t_3).$$

So it fulfills the cost function requirement in Equation (3.25), and PELT may be used to find the model and parameters that maximize Equation (3.4). This is utilized in the following example.

Example 3.7. Returning to the dataset in Table 3.1 and Figure 3.3 which was also used in Examples 3.2 and 3.6 we want to find $r(7)$ with PELT when we use the cost and penalty that maximize Equation (3.4). As PELT is simply a pruned version of OP the initial computations will be identical. When we are in the outer for-loop of PELT where $t = 4$ we may add one column to Table 3.5 to get the corresponding table for PELT which is Table 3.8. The lowest prospective cost is $p_3(4) = 13.6$, so $r(4) = 3$. Elements s such that $p_s(4) - \beta > 13.6$ are removed in lines 12 to 16 of Algorithm 2. This means that when $t = 5$ the set of possible predecessors is $\{3, 4\}$. In order to find $r(t = 5)$ we used Table 3.5 for OP, but for PELT we use Table 3.9. The latter table only uses two rows, while the former has five rows. The number of items to compute is reduced, and this reduces the run time of PELT compared to OP. However the solution they find is the same, so PELT will also result in the solution outlined in Figure 3.7

Algorithm 2: Pruned Exact Linear Time (PELT). The final estimate of the series of changepoints is kept in $\boldsymbol{\tau}$, and $F(t)$ is the final cost from data point 0 to data point t. The only functions here are cost function C, Remove(Set,a) which removes a from set Set, Append(Set,a) which appends a to the set Set, and Reverse(a) which reverses the vector a. The other entities represent scalars or vectors.

input : $Y = (y_1, \ldots, y_n)$, n=length(Y), β, $C(\cdot)$
output: $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+1})$

/* Initialize final total cost at zeroth node */
$F(0) = -\beta$
s.set $= \{0\}$
for $t \leftarrow 1$ **to** n **do**
  $F(t) = \infty$
  /* For each data point y_t find best previous changepoint y_s */
  for $s \in$ s.set **do**
    /* Calculate prospective total cost to t via s */
    $p = F(s) + C(s+1, t) + \beta$
    /* If reduction made by going via s */
    if $p < F(t)$ **then**
      /* Record new estimate for $F(t)$ */
      $F(t) = p$
      /* Record that best previous changepoint at t is s */
      $r(t) = s$
    end
  end
  for $s \in$ s.set **do**
    if $F(s) + C(s+1, t) \geq F(t)$ **then**
      Remove(s.set,s)
    end
  end
  Append (s.set,t)
end

/* Build vector $\boldsymbol{\tau}$ from r */
changepoint $=$n
$i = 1$
while changepoint $\neq 0$ **do**
  $\boldsymbol{\tau}$ (i)=changepoint
  changepoint $= r$ (changepoint)
  $i = i + 1$
end
/* Now we have $\boldsymbol{\tau} = (\tau_{m+1}, \tau_m, \ldots, \tau_1)$, so we reverse it */
$\boldsymbol{\tau}$ = Sort($\boldsymbol{\tau}$)

Table 3.8: Values needed to find $r(4)$ with PELT, most notably prospective costs $p_s(4)$ at $t = 4$ when the predecessor is s. Lowest cost is $p_3(4) = 13.6$, so $r(4) = 3$. Elements s such that $p_s(4) - \beta > 13.6$ are pruned. Corresponds to Table 3.5.

s	$F(s)$	$C(s+1,5)$	$p_s(4)$	$p_s(4) - \beta$
0	-3.89	45.3	45.3	41.4
1	0	42.4	46.3	42.4
2	4.03	40.5	48.4	44.5
3	9.67	0	13.6	9.67

Table 3.9: Values needed to find $r(5)$ with PELT, most notably prospective costs $p_s(5)$ at $t = 5$ when the predecessor is s. Lowest cost is $p_4(5)$, so $r(5) = 4$. Corresponds to Table 3.6.

s	$F(s)$	$C(s+1,5)$	$p_s(5)$	$p_s(5) - \beta$
3	9.67	4.61	18.2	14.3
4	15.4	0	15.4	15.4

The cost function in Equation (3.22) for BIC_2 has the term $\log((\tau_j - \tau_{j-1})/n)$ in addition to the term that is in BIC_1. That is denoting the cost functions of BIC_1 and BIC_2 as respectively C' and C, then

$$C(\tau_{j-1}+1, \tau_j) = C'(\tau_{j-1}+1, \tau_j) + \log \frac{\tau_{j-1}+1, \tau_j}{n}.$$

Given that both $C'(\tau_{j-1}+1, \tau_j)$ and $\log((\tau_j - \tau_{j-1})/n)$ satisfy the cost function requirement in Equation (3.25), then also $C(\tau_{j-1}+1, \tau_j)$ satisfies it. We have already proven that the requirement hods for $C'(\tau_{j-1}+1, \tau_j)$, and in the following theorem we show that it also holds for $\log((\tau_j - \tau_{j-1})/n)$. Thus we may also use PELT to find the model and parameters that fulfill the mBIC solution.

Theorem 3.3. For natural numbers t_1, t_2, t_3, and n such that $0 \leq t_1 < t_2 < t_3 \leq n$ we have

$$\log \frac{t_3 - t_1}{n} - \log \frac{t_3 - t_2}{n} - \log \frac{t_2 - t_1}{n} > 0.$$

Proof.

$$\begin{aligned}
&\log \frac{t_3 - t_1}{n} - \log \frac{t_3 - t_2}{n} - \log \frac{t_2 - t_1}{n} \\
&= \log(t_3 - t_1) - \log(t_3 - t_2) - \log(t_2 - t_1) + \log n \\
&= \log \frac{t_3 - t_1}{t_3 - t_2} + \log \frac{n}{t_2 - t_1} > 0.
\end{aligned}$$

□

However that the PELT requirement holds for the cost function is is dependent on the parametrization we chose in Equation (3.9). With $\log(\tau_j - \tau_{j-1})$ in the interval cost function instead of $\log((\tau_j - \tau_{j-1})/n)$, that is

$$C(s+1, \tau_j) = \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j) + \log(\tau_j - \tau_{j-1}), \tag{3.26}$$

and also $t_1 = 0$, $t_2 = 3$ and $t_3 = 5$, we get $t_3 - t_1 = 5$ and $t_3 - t_2 = 2$. Then $\log(5) = 1.61$ and $\log(3) + \log(2) = 1.79$ such that for small enough differences in the likelihoods $C(1,5) < C(1,3) + C(4,5)$. For instance with the data set (0.1, -1.8, 0.15, -1.1, 0.1) we would get $C(1,5) = 7.99$, $C(1,3) = 6.04$ and $C(4,5) = 2.13$ such that $C(1,3) + C(4,5) = 8.18 > 7.99$, such that the cost function does not satisfy the cost function requirement of PELT.

Example 3.8. Using PELT with BIC_2 the optimal segmentation of the data set in Table 3.1 and Figure 3.3 is $\boldsymbol{\tau} = (0, 2, 3, 7)$. With the cost function from Equation (3.22) the solution cost is

$$\begin{aligned}
&C(1,2) + C(3,3) + C(4,7) + 2\beta \\
&= 0.90 - 1.95 + 4.18 + 6\log(7) = 13.01,
\end{aligned}$$

On the other hand the output from the function `cpt.mean` in the R package `changepoint` (Killick and Eckley, 2014) is the segmentation $\boldsymbol{\tau}' = (0, 3, 7)$, which has solution cost

$$\begin{aligned} &C(1, 3) + C(4, 7) + \beta \\ &= 3.35 + 4.18 + 3\log(7) = 13.37. \end{aligned}$$

The solutions found with these algorithms are different and have different solution costs.

4. Simulations and discussion

We will now study the concepts and algorithms we have introduced so far by applying them to some simulated data. In this section we first compare the results from maximizing BIC_1 with BinSeg and PELT. Then in Section 4.2 we study the additional penalty on the relative positions of the changepoints that was introduced in BIC_2. The insights derived in these sections allow Section 4.3 where we compare the performance of BIC_1 and BIC_2 on simulated data to be concise. The simulations are followed up with a preliminary discussion in Section 4.4 on some of the concepts covered so far in the thesis.

All analyses are performed in R (R Core Team, 2017). The central pieces of code are displayed in Appendix B, and are also available in an R package at `https://github.com/kristinbakka/generalizedPELT`. Most of the simulations are analyzed with our generalization of PELT that we present in Section 5.6.3, although in Section 4.1 also the implementation of PELT in the package changepoint (Killick and Eckley, 2014) is used. In the package changepoint several changepoint detection methods are implemented, but we only use the implementations of BinSeg and PELT. The syntax to employ these algorithms are displayed in Appendix B.1. In this section we will refer to BIC_1 and BIC_2 as respectively BIC and mBIC, since they are adaptations of these criteria. In the figures we will indicate each simulation or set of simulations with a dot or a cross, and connect the dots with lines for readability. The exception is that we will omit the dots when they are very close to each other.

4.1. Compare PELT and BinSeg using BIC

BinSeg is one of the most popular algorithms for multiple changepoint detection. In this section we look at how PELT performs in comparison to BinSeg when they are used to maximize BIC_1. All analyses are performed in R (R Core Team, 2017). We use the implementation of PELT in the package changepoint Killick and Eckley (2014) when it gives the same result as our own implementation. We also use it to maximize BIC_1 with BinSeg. The syntax of the function we have used is described in Appendix B.1. There are respectively zero, one and multiple internal changepoints in the simulated data in Sections 4.1.1, 4.1.2 and 4.1.3.

4.1.1. No internal changepoints

First we want to compare the performance of the methods when there are no internal changepoints. In order to do this we simulate time series of length $h_{max} = 5000$ where each point is from a standard normal distribution. Then we evaluate the first h data of each series with BinSeg and PELT, where $1 \leq h_1 < h_2 < \cdots \leq h_{max}$. Three simulations with $h_1 = 5$, $h_2 = 50$ and $h_3 = 85$ are illustrated in Figure 4.1. In the simulations the maximum

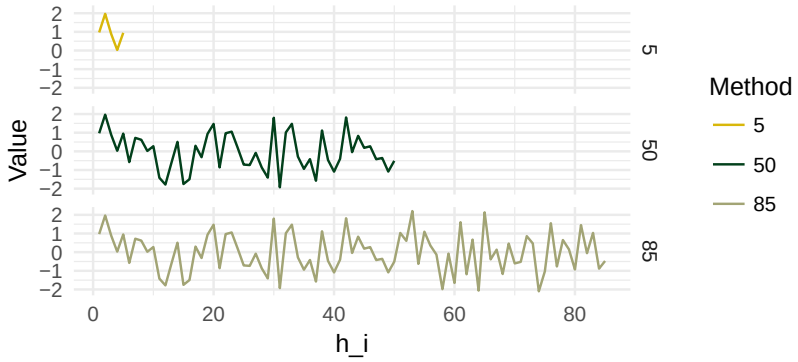


Figure 4.1: Illustration of a simulation evaluated at $h_1 = 5$, $h_2 = 50$ and $h_3 = 85$ of a process where the data points are realizations from $\mathcal{N}(0, 1)$, and there is no changepoint.

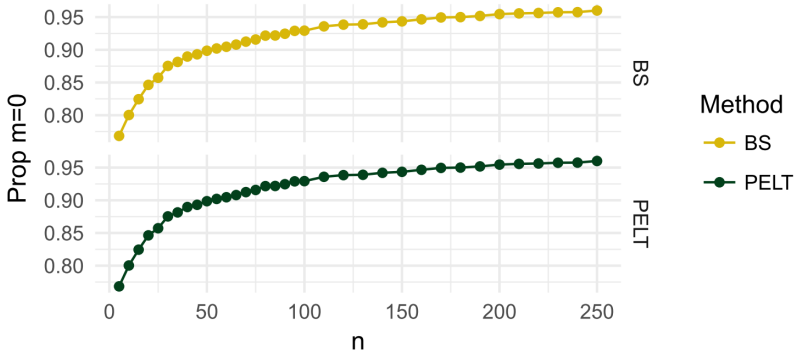


Figure 4.2: Proportion of 10000 simulations where no internal changepoints was detected as a function of the length of the data set h. The simulations contained no changepoints and the data points were realizations from $\mathcal{N}(0, 1)$. BS is another abbreviation for Binary Segmentation, and signifies that BinSeg was used to analyze the data.

number of changepoints with BinSeg is set to $Q = 5$, and when evaluating at the ith length h_i we use the BIC_1 penalty $\beta_1 = 2\log(h_i)$ from Equation (3.21). The penalty thus increases as the data set increases in size. Evaluating the first h data of each a series with BinSeg and PELT is comparable to a setting where we receive a data set in real time and evaluate the data set with BinSeg and PELT in order to find whether a changepoint has occurred yet.

With a simulation where there are 50 different values of h and with varying increase in h the proportion of simulations where BinSeg or PELT correctly find no changepoints is displayed in Figure 4.2. In all the simulations the algorithms either found no changepoints or one changepoint, never multiple changepoints. The figure illustrates that for a short data set the probability of not finding a changepoint that is not there is a little smaller than for a long data set. This makes sense since the algorithms use the information from all the data points to determine whether there is a changepoint

present. If a high proportion of the data points in a data set is improbable under a null hypotheses of no changepoints, then the null hypotheses should be rejected and a changepoint detected. When a data set of realizations from some distribution is long, it is less likely that a high proportion of the data set is realizations with low probability density, and so the likelihood of correctly finding no changepoint increases with the length of the data set.

In order to avoid detecting changepoints that are not there, the penalty could be adjusted to be larger for small data sets. For instance it could be changed to $\beta = ((h_i)^{-1} + 2)\log(h_i)$. Any such adjustment would however affect analyses of different data sets in multiple ways. It could make PELT or BinSeg dislocate or not find changepoints when they are present. It could also render undetectable changepoints belonging to short intervals.

Another result from the simulation is that both methods found zero or one changepoint in all the cases considered. Here more than one internal changepoint is not a viable option. As stated in Section 3 PELT finds the likelihood based optimal segmentation of the data set, while as stated in Section 3.5.1 BinSeg finds the likelihood based optimal binary segmentation of the data set recursively. When more than one internal changepoint is not a viable option BinSeg is not applied recursively, and both BinSeg and PELT find the optimal binary segmentation if it is optimal to split the data into two segments. So in this case PELT and BinSeg estimate the exact same number (and positions) of changepoints, which is why the results from the two methods are exactly the same in Figure 4.2.

4.1.2. One internal changepoint

In this section there will be exactly one internal changepoint in the middle of the data set, and we will seek to identify it correctly. Figure 4.3 illustrates the setting. The first points are drawn independently from $\mathcal{N}(0, 1)$, and the rest of the points from $\mathcal{N}(\Delta, 1)$. In the short data set in the figure it is easy to see that a changepoint occurs at $t = 5$ when Δ is 4 or 7, but it is not easily discernible when Δ is 1.5 or lower.

We will continue to evaluate the performance of maximizing the BIC_1 from Equation (3.4) with BinSeg and PELT. There are two main questions to answer when we evaluate a changepoint detection method, and those are *Does it find the correct number of changepoints?* and *Does it find the correct position?*. First we look at how many changepoints the model finds, displayed in Figure 4.4. For a data set of this length the methods find either $0, 1, 2$ or 3 changepoints, and so for each Δ in Figure 4.4 the proportions sum to 1. For instance when $\Delta = 0.5$ the proportion of the simulations where PELT finds $0, 1, 2$ and 3 changepoints is respectively 0.01, 0.06, 0.21 and 0.72, and $0.01 + 0.06 + 0.21 + 0.72 = 1$.

The black lines in Figure 4.4 represent the proportion of simulations that find the correct number of changepoints in data sets of length $n = 10$. PELT performs better than BinSeg when $\Delta < 1$, while it is the other way around

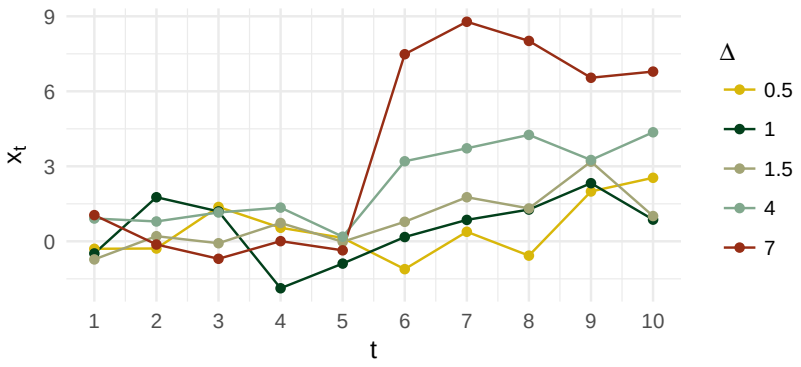


Figure 4.3: An illustration of simulations with the parameters $\boldsymbol{\tau} = (0, 5, 10)$ and $\boldsymbol{\mu} = (0, \Delta)$ for different Δ.

when $\Delta > 1$. The rest of the lines demonstrate that for every Δ PELT finds 0 changepoints a lower portion of the simulations than BinSeg, and finds more than 1 changepoints a higher portion of the simulations. So the reason BinSeg performs better than PELT when $\Delta > 1$ is that PELT has more of a tendency to find too many changepoints. This is an advantage when there is one changepoint but $\Delta < 1$, and so then PELT performs better.

The same tendencies are apparent in Figure 4.5 as well as in Figure 4.4, but the performance of PELT and BinSeg is much more equal and a $\Delta \geq 0.75$ is sufficient for the algorithms to detect the changepoint in 0.75 portions of the simulations. This demonstrates that a short data set represents a higher difficulty. Since a short data set means less data points to determine the number of changepoints it also means less information. So the longer the data set, the better and more similar the performance of BinSeg and PELT with respect to the number of changepoints detected. When we move on to data sets with more changepoints the ratio m/n is often larger than in the long data sets considered in this section. Sometimes the ratio m/n is quite large, and therefore it is interesting to look at data where $m = 1$ and the ratio m/n is quite large as well. If PELT or BinSeg are to be applied in an online setting the EDD must be taken into account, and the behaviour on short data sets ought to be taken into consideration. The next point to investigate is whether the algorithms find the correct position of the changepoint. In Figures 4.6 and 4.7 the proportion of simulations where the methods find the correct $\boldsymbol{\tau}$ is displayed for data sets of lengths 10 and 100. In the long data set the performance of BinSeg and PELT is virtually the same, while BinSeg performs better on the short data set. For a high enough Δ both methods find the correct $\boldsymbol{\tau}$ whenever it finds the correct m, but a Δ of 5.5 or more is necessary. This is a very high value, as we can see from Figure 4.3, and it does not seem to depend on n. However when n is larger the proportion of simulations where each of the methods find the correct $\boldsymbol{\tau}$ is

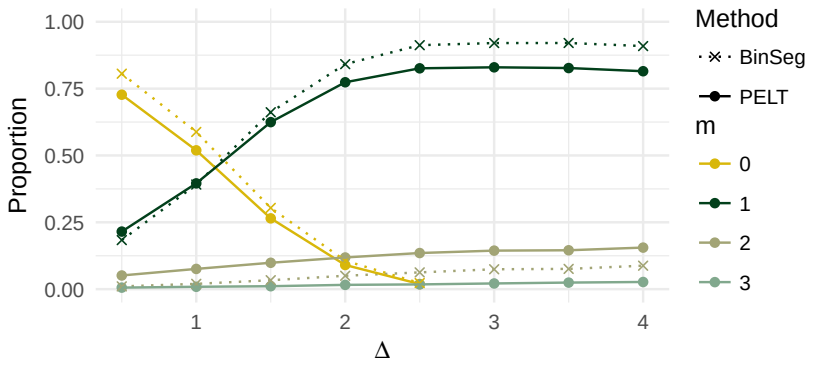


Figure 4.4: Proportion of simulated data sets where the method detects between 0 and 3 changepoints. There are 20000 simulated data sets for each Δ. Each data set is simulated with the parameters $m = 1$, $n = 10$, $\boldsymbol{\tau} = (0, 5, 10)$ and $\boldsymbol{\mu} = (0, \Delta)$, and evaluated with BIC_1 from Equation (3.4).

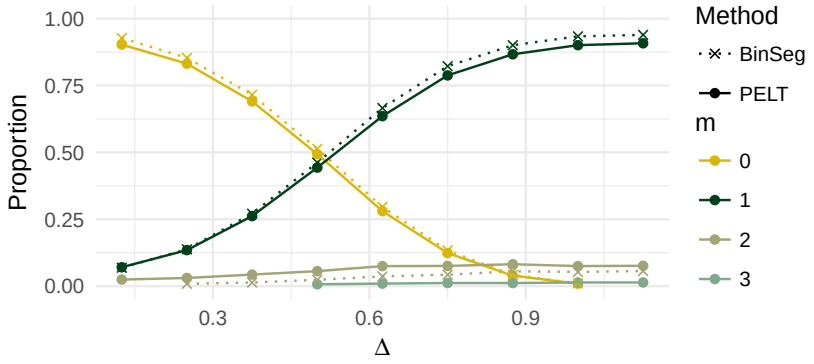


Figure 4.5: Proportions of simulated data sets where the method detects between 0 and 3 changepoints, given that the proportion is greater than 0.005. There are 10000 simulated data sets for each Δ. Each data set is simulated with the parameters $m = 1$, $n = 100$, $\boldsymbol{\tau} = (0, 50, 100)$ and $\boldsymbol{\mu} = (0, \Delta)$, and evaluated with BIC_1 from Equation (3.4).

greater for every Δ. For instance for $\Delta = 2$ and BinSeg the proportions are respectively 0.59 and 0.53 when $n = 100$ and $n = 10$.

From Figure 4.7 it is evident that there are three different situations each with their specific challenge that can occur depending on the size of Δ when we have one true changepoint. If Δ is very small it is difficult to detect that there has been a change at all. If the Δ is quite small it is easier to detect it, but it is difficult to detect its position. When Δ is quite large then the position is usually correct when the correct number of changepoints is found, but the risk of finding too many changepoints is more prominent. Which Δ is small, intermediate or large clearly depends on either n/m or n, although as $m = 1$ in this section we do not yet have enough information to ascertain which of these it is. For instance Figures 4.6 and 4.5 show that intermediate values for Δ are approximately $1 < \Delta < 4.5$ when $n = 100$ and

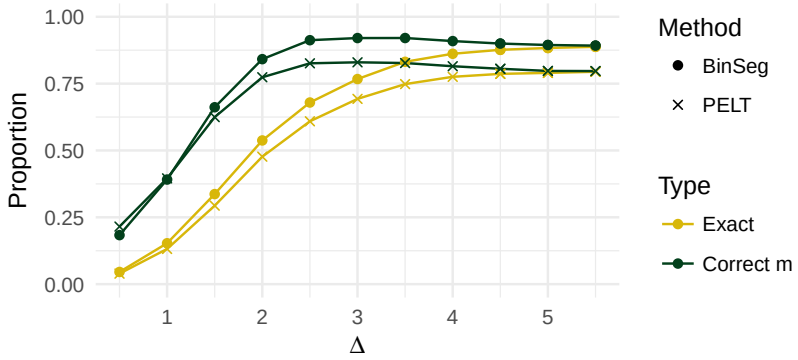


Figure 4.6: Proportion of simulations where the method finds the exact position of m or merely the correct m when $m = 1$, $n = 10$, $\boldsymbol{\tau} = (0, 5, 10)$.

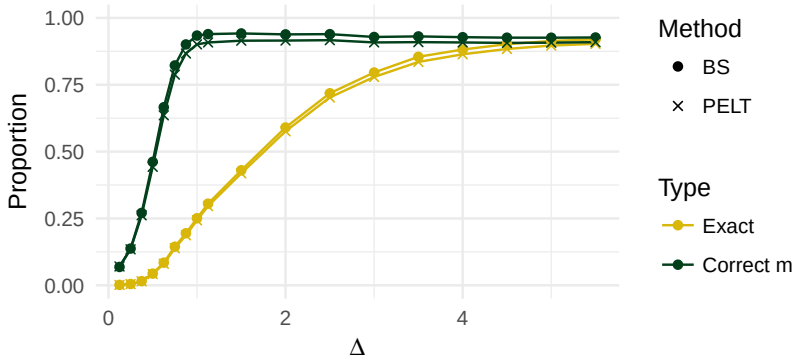


Figure 4.7: Proportion of simulations where the method finds the exact position of m or merely the correct m when $m = 1$, $n = 100$, $\boldsymbol{\tau} = (0, 50, 100)$.

$2.5 < \Delta < 4.5$ when $n = 10$. Thus we will use the term *intermediate region* to refer to the region of the parameter space where the algorithms usually find the correct number of changepoints, but not the correct positions.

The natural next question that arises is *What changepoint position do the methods find when they find the correct number of changepoints?*. In order to answer this we have Figures 4.8 and 4.9 which display the proportion of simulations where the methods detected one changepoint positioned at each of these values for t. In the first plot we see that already at the small value $\Delta = 0.5$ the most likely changepoint position to be identified is the correct one. The figure also shows that the farther away a position is from the correct one, the less likely it is that the method detects it as the changepoint position.

The histogram for BinSeg and PELT in Figure 4.8 are not identical although we know that when both algorithms find one changepoint, then they find the same position. This is because there are instances where one method and not the other finds the correct number of changepoints. With a large

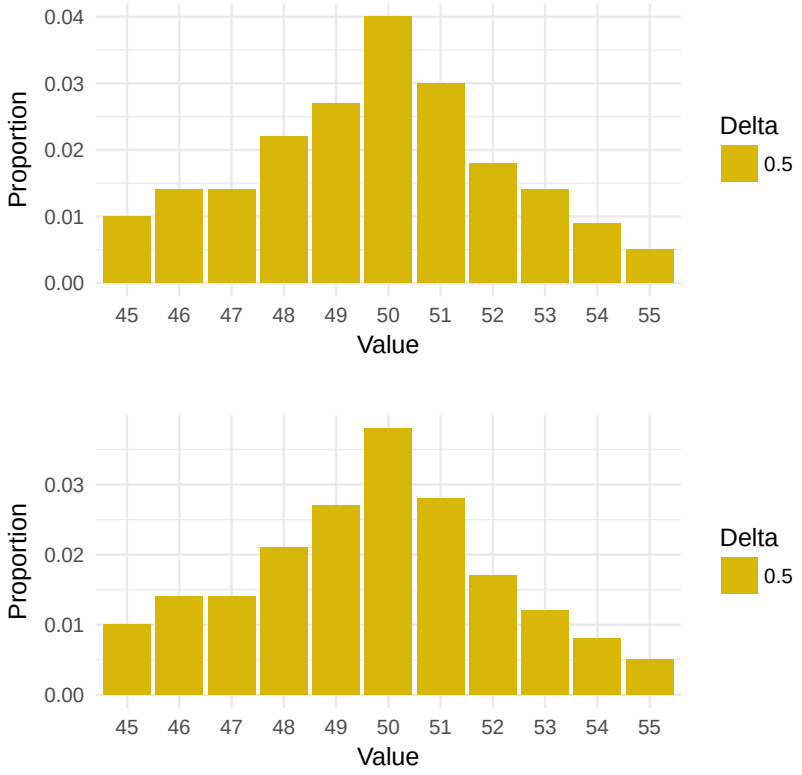


Figure 4.8: Proportion of simulations that correctly classify to one changepoint and identifies the changepoint position as the value indicated when $m = 1$, $n = 100$, and $\boldsymbol{\tau} = (0, 50, 100)$. The upper histogram is for BinSeg, the lower is for PELT.

enough number of simulations the plots would be symmetric about $t = 50$, since the ordering of time does not make any difference to BinSeg and PELT, and the changepoint is centered in the data set. The difference between the histograms for BinSeg and the histograms for PELT in Figure 4.9 is very small so this is an issue for both the algorithms. As expected BinSeg identifies the correct changepoint position at a slightly higher proportion of the simulations than PELT for all Δs. This is because BinSeg identifies the correct number of changepoints in a higher proportion of the simulations.

From Figure 4.7 we know that the intermediate region when $n = 100$ is for Δ approximately such that $1 \leq \Delta \leq 4.5$. And so in Figure 4.9 the histograms at 50 are higher for $\Delta \in \{4, 7, 9\}$ than for $\Delta \in \{1, 2\}$. Also when $\Delta = 2$ the proportion of simulations in Figure 4.9 where the methods find the changepoint position to be at $|t - 50| = 5$ is negligible. This means that the proportions for each position t when $|t - 50| > 5$ are even smaller. This means that from the middle of the intermediate region the changepoint position is detected close to the correct location when it is not detected at the exactly correct position.

From Figure 4.9 we also see that although it is about as likely for the

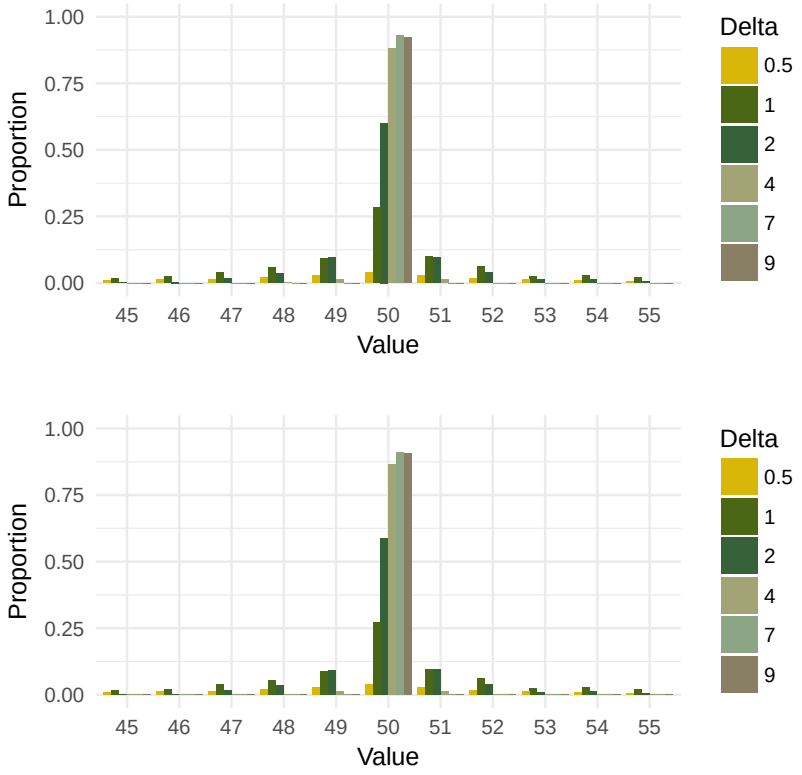


Figure 4.9: Proportion of simulations that correctly classify to one changepoint and identifies the changepoint position as the value indicated when $m = 1$, $n = 100$, and $\boldsymbol{\tau} = (0, 50, 100)$. The upper histogram is for BinSeg, the lower is for PELT.

methods to find a changepoint at $|t-50| = 1$ when $\Delta = 1$ and when $\Delta = 2$, it is far more likely to find one at $|t-50| = 2$ when $\Delta = 2$. Also the proportion of simulations where the position is located at t decreases less as t increases. This means that in the start of the intermediate region the position detected is not reliable. Also it means that in the intermediate region the position becomes gradually more focused. In applications we might have different tolerances for how close to the correct position the changepoint needs to be located. For instance if used directly in an anomaly detection setting on a boat it is conceivably of no significance when in a period of one minute the anomaly happens. With for instance 50 sample point per second that would allow for a 'wiggle room' of 3000 points. Thus when we now move on to the situation with more changepoints we will only register what is the intermediate region. We will not try to divide it into when the solution is acceptable, since that depends entirely on the application.

4.1.3. Multiple internal changepoints

In this section we investigate the properties of the solutions when $m = 5$. This is when we expect PELT to perform better than BinSeg, as PELT

is constructed to detect multiple changepoints in complex data. The parameters of the simulations we will test the methods on are such that $\boldsymbol{\tau} = (0, \tau_1, 2\tau_1, \ldots, 6\tau_1)$ and $\boldsymbol{\mu} = (0, \Delta, 2\Delta, \ldots, 5\Delta)$, as illustrated in Figure 4.10 for $\tau_1 = 10$ and for different Δ values. We will again seek to answer the two questions *Does the method find the correct number of changepoints?* and *Does the method find the correct position?*.

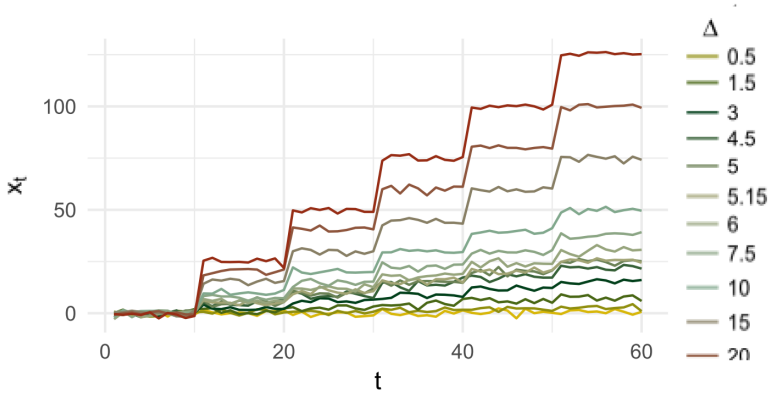


Figure 4.10: One simulation for each of the 12 Δ values where $m = 5$ $\boldsymbol{\tau} = (0, 10, \ldots, 60)$ for, $\boldsymbol{\mu} = (0, \Delta, 2\Delta, \ldots, 5\Delta)$ and $n = 60$.

In Figures 4.11 and 4.12 the proportion of simulations where the methods find the correct $\boldsymbol{\tau}$ or the correct number of changepoints is displayed. The graphs for correct number of changepoints start with a steep ascent. Already at $\Delta = 1.5$ PELT finds the correct number of changepoints in more than 87.5% of the simulations. The other graphs also all eventually plateau at a proportion of approximately 87.5%. This means that for large Δ values it is only important whether Δ is bigger than some limit, and not how large the value is. The correct number of changepoints found increases quicker for PELT than for BinSeg initially, and quicker when $n = 240$ than when $n = 60$. Figure 4.12 shows that PELT finds the correct number of changepoints in more than 87% of the simulations already at $\Delta = 1.5$, while BinSeg only reaches that level after $\Delta = 7$.

In Figure 4.13 the different m values found by the algorithms are displayed. In this case BinSeg finds more than the correct number of 5 changepoints a larger proportion of the simulations than PELT. When approximately $\Delta < 1.20$ the methods still find the wrong number of changepoints 25% of the simulations. Then both of the methods fail because they find too few changepoints. On the other hand when approximately $\Delta > 1.40$ the concern is whether the methods find too many changepoints. And in the intermediate region BinSeg finds more changepoint than are actually there. This makes sense because with multiple changepoints in the data set then the likelihood that an interval that is the result from a binary segmentation will contain one or more changepoints increases. And so when the algorithm

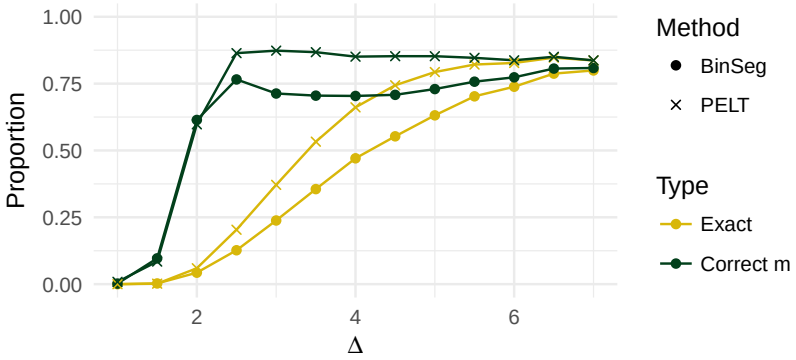


Figure 4.11: Proportion of 10000 simulations where the methods find the exact position or merely the correct m when $m = 5$, $n = 60$, $\boldsymbol{\tau} = (0, 10, 20, \ldots, 60)$ and $\boldsymbol{\mu} = (0, \Delta, 2\Delta, \ldots, 5\Delta)$.

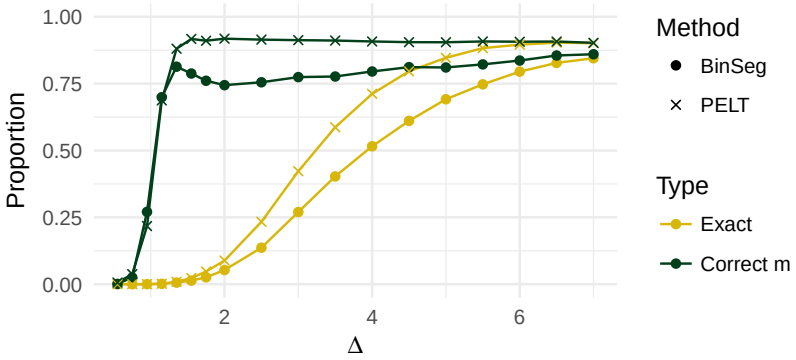


Figure 4.12: Proportion of 60000 simulations where the methods find the exact position of m or merely the correct m when $m = 5$, $n = 240$, $\boldsymbol{\tau} = (0, 40, 80, \ldots, 240)$ and $\boldsymbol{\mu} = (0, \Delta, 2\Delta, \ldots, 5\Delta)$.

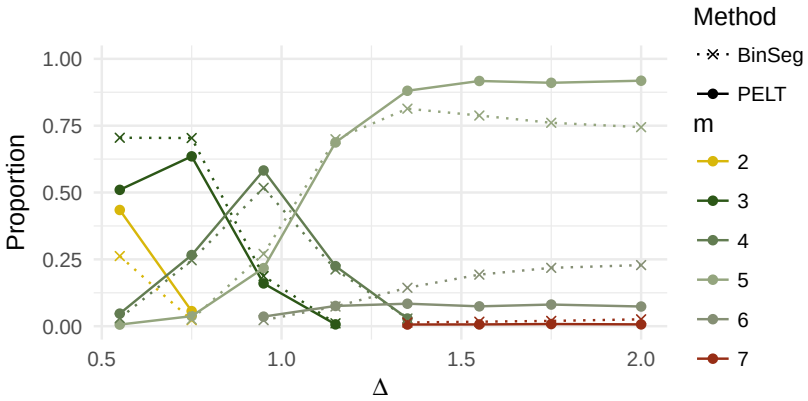


Figure 4.13: Proportion of 60000 simulations where the methods find different m when $m = 5$, $n = 240$, $\boldsymbol{\tau} = (0, 40, 80, \ldots, 240)$ and $\boldsymbol{\mu} = (0, \Delta, 2\Delta, \ldots, 5\Delta)$.

puts a split somewhere other than on the changepoint the resulting intervals

will sometimes contain a true changepoint that the algorithms is able to detect. Then there might be detected more changepoints than are actually there. It looks like this effect is compounded by the size of m relative to n as the difference between PELT and BinSeg is smaller at $\Delta = 7$ in Figure 4.11 than in 4.12.

4.2. The mBIC penalty

In this section we want to study the penalty in BIC_2 in detail. This is primarily so that we can predict and interpret the workings of model selection with BIC_2. We will discuss it in terms of the value for d in Equation (3.13) given by the *edf* in Equation (3.15), so that we get a number that is comparable across data sets of different length and with different number of data sets.

Depending on the assumptions we make and on the length of n the effective degrees of freedom per interval in Equation (3.15), ought to be different. Some common and reasonable assumptions are that

1. Every vector $\boldsymbol{\tau}$ is equally probable for a given m.
2. The number of changepoints increases linearly with n.

The latter point is true for instance when the time between changepoints is exponentially distributed, or when the changepoints are evenly spaced throughout the data set. For evenly spaced changepoints *edf* may be computed easily. We will now move on to compute both of these in order to study the *edf* on these types of data.

It is possible to compute the average value of *edf* for Alternative 1 combinatorially. Instead we draw the $\boldsymbol{\tau}$ uniformly, that is we draw the m changepoint positions without replacement from $\{1, \ldots, n-1\}$ with identical probabilities on each value. Counts of the different resulting *edf* values are displayed in Figures 4.14, 4.15 and 4.16. The first axis is marked by *Penalty* because *edf* is computed from the total penalty. The three dashed lines mark the 10%, 50% and 90% quantiles, while the solid line marks the mean.

The upper limit of the support is from Equation (3.11)

$$\frac{-(m+1)\log(m+1) + 2m\log n}{m\log n} = 2 - \frac{m+1}{m}\frac{\log(m+1)}{\log n},$$

which is for $m = 5$ and $n = 120$ 1.55, while with $m = 3$ it is respectively 1.70 and 1.83 for $n = 500$ and $n = 50000$. From Equation (3.10) the lower limit is

$$\frac{1}{m\log n}(m\log n + \log\frac{n-m}{n}) = 1 + \frac{1}{m\log n}\log\frac{n-m}{n},$$

and is 1.0000, 0.9997, and 0.9982 when (n, m) is respectively $(50000, 3)$,$(500, 3)$ and $(125, 5)$. And so the there are less possible values for the *edf* in Figures 4.14, 4.15 and 4.16 for smaller n.

The sample distributions we may derive from the histograms in Figures 4.14, 4.15 and 4.16 are markedly skewed to the right, very close to the limit of their support. When n is larger the distribution is also more skewed, which we for instance can see from the distance between the median and the mean. The count at the mode is also higher when n is higher, and the

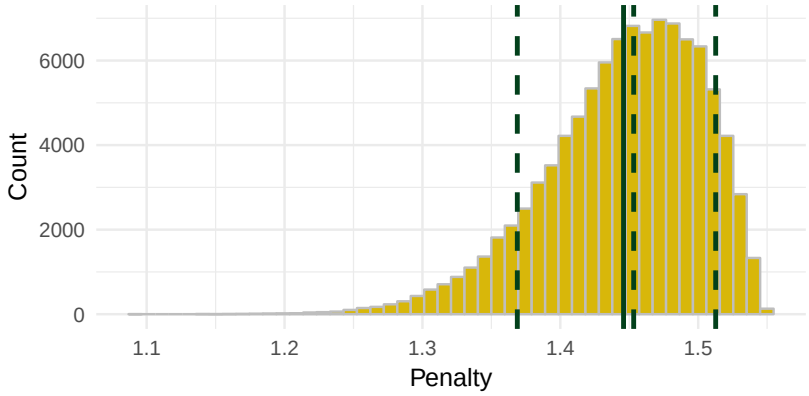


Figure 4.14: For 100000 $\boldsymbol{\tau}$s uniformly drawn according to Alternative 1 defined on page 60 with $m = 5$ and $n = 120$, this is the count of the number of $\boldsymbol{\tau}$s that give rise to these values of Equation (3.15).

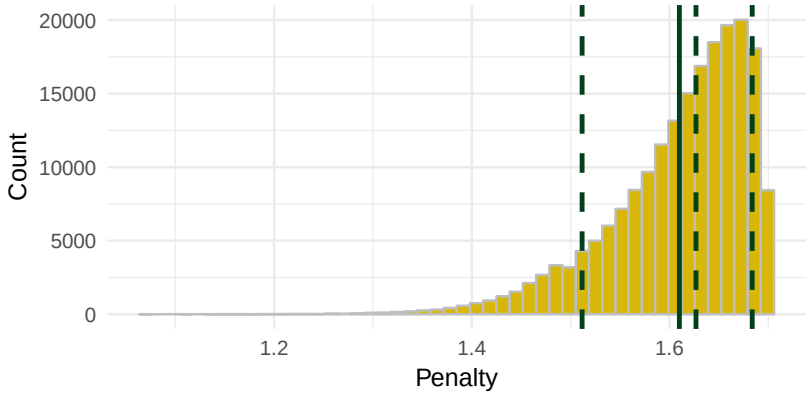


Figure 4.15: For 100000 $\boldsymbol{\tau}$s uniformly drawn according to Alternative 1 defined on page 60 with $m = 3$ and $n = 500$, this is the count of the number of $\boldsymbol{\tau}$s that give rise to these values of Equation (3.15).

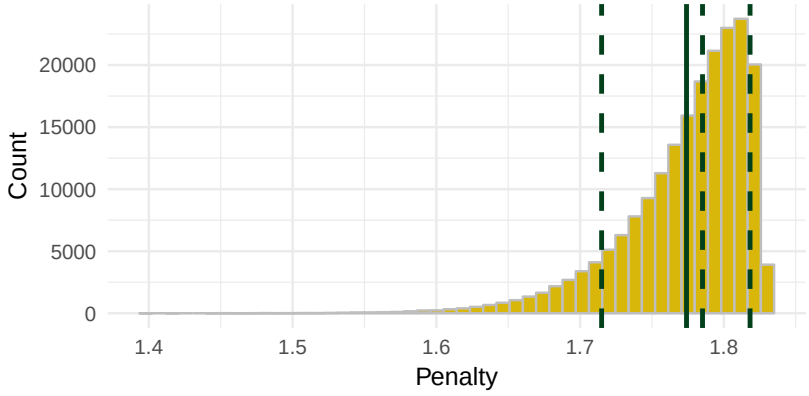


Figure 4.16: For 100000 $\boldsymbol{\tau}$s uniformly drawn according to Alternative 1 defined on page 60 with $m = 3$ and $n = 50000$, this is the count of the number of $\boldsymbol{\tau}$s that give rise to these values of Equation (3.15).

distance between the 10% and 90% quantiles is shorter. This is because there are more ways to draw $m = 3$ or $m = 5$ points spread out throughout the data set than there are ways to draw them close together. And also because in the longest data sets the most frequent set of distances are more frequent. In Figures 4.17 and 4.18 these patterns are easily discernible, and an important exception is illustrated. Namely that when n is small enough then $(n-1)\mathrm{C}(m)$ from Equation (3.1) is a low number such that there are few possible values of *edf*. Then the counts of the most frequent values of *edf* is larger than for slightly larger n. Another consequence is that the distribution becomes ragged instead of smooth, that is bins next to each other may be of quite different heights. This signals that *edf* values within occur with different frequency, and is to be expected since the sample distribution is discrete with a support of a maximum of $(n-1)\mathrm{C}(m)$ values. Since $(n-1)\mathrm{C}(m)$ is also small when $|\frac{1}{2}n - m|$ is large, that is when m is close to n or to 0, these conclusions may also be drawn for such m.

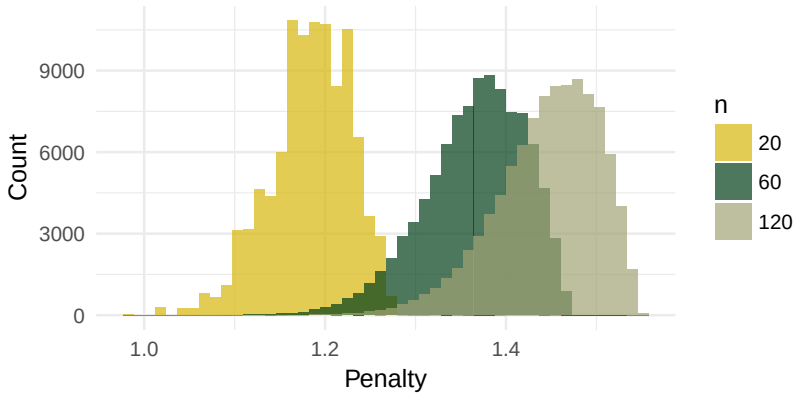


Figure 4.17: For Alternative 1 defined on page 60 this is the distribution of Equation (3.14) divided by $m = 5$ with three different n. The edf was computed for 100000 drawn $\boldsymbol{\tau}$.

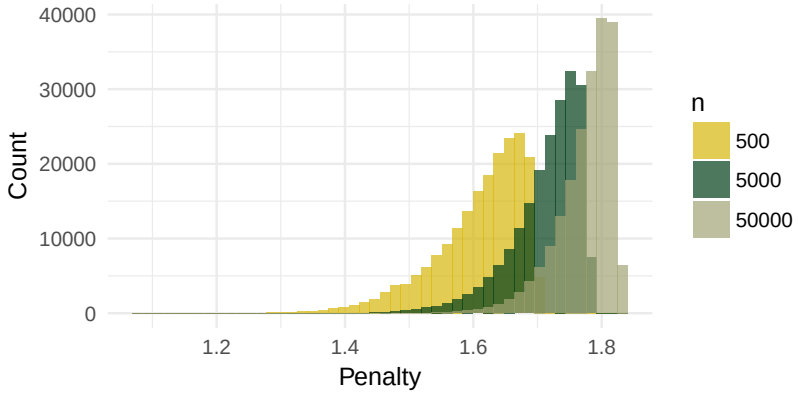


Figure 4.18: For Alternative 1 defined on page 60 this is the distribution of Equation (3.14) divided by $m = 3$ with three different n. The edf was computed for 100000 drawn $\boldsymbol{\tau}$.

Now that we have analyzed how the *edf* values differ when the change-points are uniformly distributed we may move on to investigate what happens to the *edf* for different m when n increases. This is displayed in Figure 4.19 for two different intervals of n. The *edf* increases with n, but increases less as n gets higher. The simple form of Alternative 2 defined on page 60 corresponds to the maximal *edf* and is thus close to the mode of the distribution for Alternative 1.

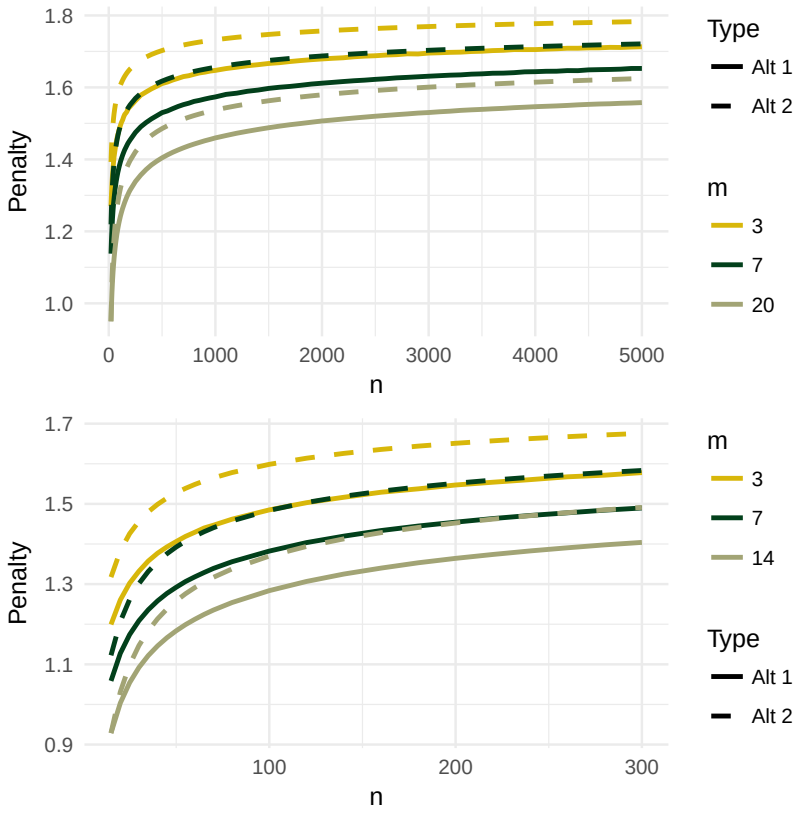


Figure 4.19: The plots display Equation (3.15). The solid lines are marked with Alt 1, which means that they are the means of 10000 uniformly drawn $\boldsymbol{\tau}$s for each n. The dashed lines are marked with Alt 2 because they are the edf values when $\boldsymbol{\tau} = (0, \frac{n}{m+1}, \ldots, \frac{n(m+1)}{m+1})$.

As $\text{BIC}_{1,adj}$ is for reasonable choices of d a hybrid between the well used BIC_1 and BIC_2, it would be interesting to use it as a third criteria to compare the other two criteria with in the next section. Then we could for instance see the difference between estimating the τ_js only based on the likelihood and a constant penalty, or also based on the interval lengths. It is tempting to point out that in Figure 4.19 there is a difference of approximately 0.5 between the lowest and highest *edf* value in the graphs for a given n, and to claim that we can use this knowledge about the *edf*, and that n is usually known to select a good d for $\text{BIC}_{1,adj}$ from Equation (3.13). But when m is

high compared to n the maximal *edf* is

$$2 - \frac{m+1}{m}\frac{\log(m+1)}{\log n} \tag{4.1}$$

which in the extreme case when $m = n - 1$ is

$$2 - \frac{n}{n-1}\frac{\log n}{\log n} < 1.$$

And so the pattern that each line in Figure 4.19 starts at very low *edf* values would also be there when m is larger. Thus all d values in the interval may be the optimal choice according to the *edf*, and we need prior knowledge on the approximate number of changepoints in the interval to set a good d value for $\text{BIC}_{1,adj}$. It is however reasonable for d to be set somewhere between the maximal and minimal possible *edf* values for the n of the data set in question instead of to 1. We would need an extensive set of simulations to evaluate the resulting $\text{BIC}_{1,adj}$ as it would be quite different on different data, and it would not be fair to simply pick one. And so when we in the next section compare BIC_1 and BIC_2 we will leave out $\text{BIC}_{1,adj}$. One result from this study is however that we know more about how to choose the region that d ought to be in. This comes in handy when computer scientists want to use the CROPS method (Haynes et al., 2017a) for changepoint detection, where you only supply the minimal and maximal values of $d+1$ and get the different $\boldsymbol{\tau}$s that $\text{BIC}_{1,adj}$ produces for penalties in that region.

4.3. Compare BIC and mBIC using PELT

In this section we compare the performance of the methods when BIC_1 and BIC_2 are maximized. The plots corresponding to maximization of BIC_2 will be marked by mBIC as it is an approximation to the mBIC criterion. There are only 2000 simulations for each Δ because use our own slow R-implementation instead of the fast C implementation in Killick and Eckley (2014).

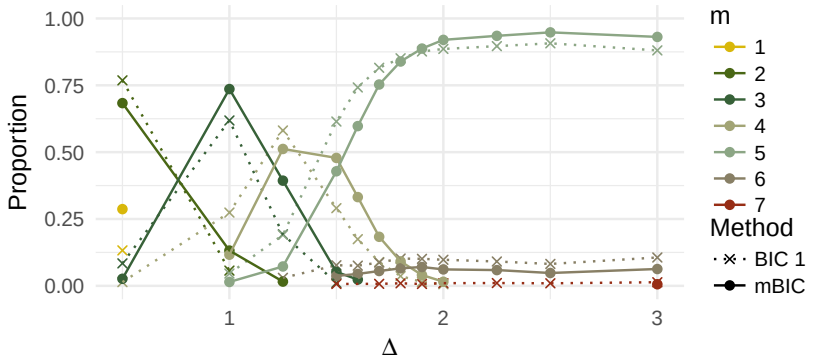


Figure 4.20: Proportion of 2000 simulations where the methods find the correct m when $m = 5$, $\boldsymbol{\tau} = (0, 20, 40, \ldots, 120)$ and $\boldsymbol{\mu} = (0, \Delta, \ldots, 5\Delta)$..

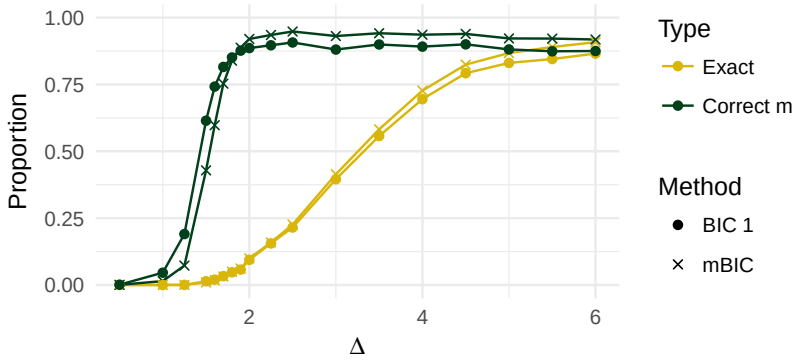


Figure 4.21: Proportion of 2000 simulations where the methods find the exactly correct changepoint vector or merely the correct m when $m = 5$, $\boldsymbol{\tau} = (0, 20, 40, \ldots, 120)$ and $\boldsymbol{\mu} = (0, \Delta, \ldots, 5\Delta)$.

In Figures 4.20 and 4.21 the changepoints are evenly distributed across the data. For all values of Δ then BIC_1 finds more changepoints than mBIC, and so it also finds the correct number $m = 5$ in a higher proportion of the simulations when $\Delta < 1.75$. However mBIC quickly surpasses BIC 1, and detects the correct m from $\Delta = 2$ and throughout the intermediate region. Also mBIC finds the correct changepoint vector on more simulations than BIC_1 for every value of Δ. And so mBIC performs better than BIC_1 on the data set these figures are based on. A potential problem with the

BIC_1 criterion is that the penalty is too small, and these simulations show that that is indeed a problem. However the parameters of the simulations on which BIC_1 and BIC_2 are tested in Figures 4.20 and 4.21 are artificially favourable for mBIC. This is because the changepoints are evenly spread in the data set, and thus the BIC penalty for the data sets are close to the maximal penalty for $m = 5$ and $n = 120$.

One option is to study simulations where the changepoint vector is $\boldsymbol{\tau} = (0, 1, \ldots, m, n)$. Then mBIC will have a lower resulting penalty and may find too many changepoints, or BIC may find too few. However such a changepoint vector is a theoretical setting that is seldom of interest in an application. So in the remainder of this section we will instead for each simulation draw the changepoint vector uniformly in the way detailed on page 60. The goal will then be to find how the various attributes in Table 4.1 affect the proportion of the simulations at which BIC and mBIC find the correct number of changepoints. For this we use Figures 4.22, 4.23 and 4.24.

Table 4.1: The model parameters we use in the simulations where $\boldsymbol{\tau}$ are drawn uniformly and the resulting key numbers. The value of max *edf* is from Equation (4.1). Combinations are the number of possible $\boldsymbol{\tau}$ from Equation (3.1).

n	m	$n/(m+1)$	m/n	max *edf*	$\log n$	Combinations
24	5	4	0.21	1.32	3.18	$3.4 \cdot 10^4$
105	20	5	0.19	1.31	4.65	$1.3 \cdot 10^{21}$
100	4	20	0.04	1.56	4.61	$3.8 \cdot 10^6$

We start by looking at Figure 4.22 which is where mBIC performs the worst compared to BIC_1. Key information about the simulations in this plot is in Table 4.1 under $n = 24$. The performance of BIC_1 is marginally worse than that of mBIC when $\Delta > 3.5$ that is when the proportion of simulations where the correct m is found is larger than 0.45. In Figure 4.23 we have increased n and m such that $\log n = 4.65$ as detailed in Table 4.1. The difference between the performance of BIC_1 and mBIC increases from Figure 4.22 to Figure 4.23. The reason for this is that $\log n$ increases, so when the effective degrees of freedom of $\boldsymbol{\tau}$ is different from 1 the difference between the penalties is larger, and the criteria are less similar. This also illustrates that the meager differences in these small data sets is exacerbated when n is larger, so they may amount to large differences for the long data sets on which PELT is usually applied.

Next we look at Figure 4.24 and compare it to Figure 4.23. The difference here is that m is reduced such that the maximal number of degrees of freedom increases (see Table 4.1). This leads to the difference between the two criteria increasing as well, and mBIC performs better than BIC_1 when $\Delta > 3$ which is when the proportions of simulations with correctly detected m surpass 65%.

The differences in the shapes of the curves for the different correct ms

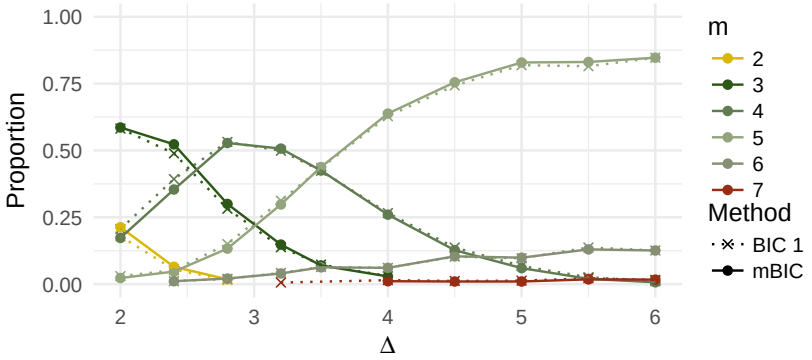


Figure 4.22: Proportion of 1000 simulations where the methods find the correct m when $m = 5$, $n = (m+1)4 = 24$, $\boldsymbol{\tau}$ is uniformly drawn for each simulation, and $\boldsymbol{\mu} = (0, \Delta, \ldots, 5\Delta)$.

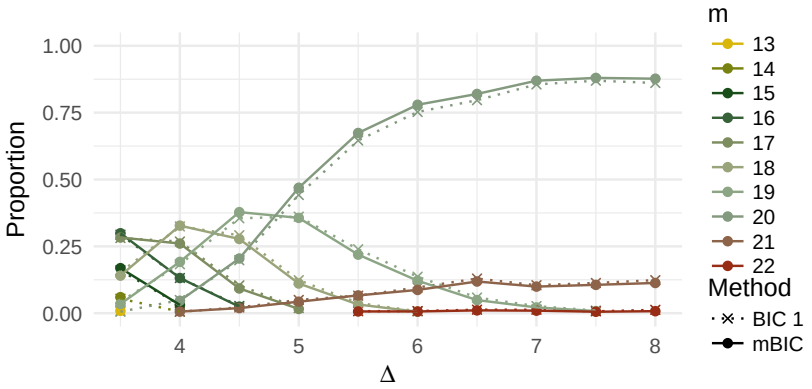


Figure 4.23: Proportion of 2000 simulations where the methods find the correct m when $m = 20$, $n = (m+1)5 = 105$, $\boldsymbol{\tau}$ is uniformly drawn for each simulation, and $\boldsymbol{\mu} = (0, \Delta, \ldots, 20\Delta)$.

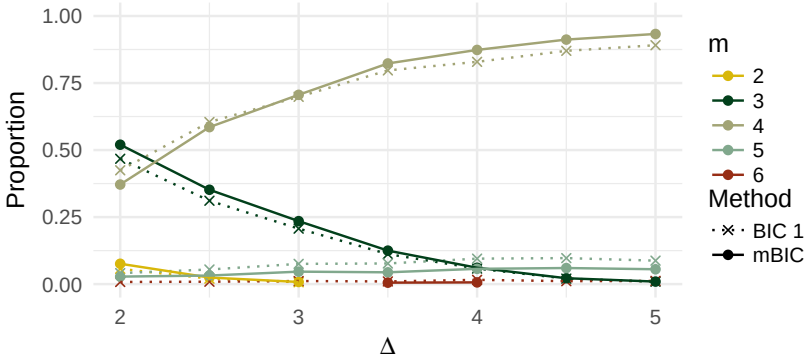


Figure 4.24: Proportion of 2000 simulations where the methods find the correct m when $m = 4$, $n = (m+1)20 = 10$, $\boldsymbol{\tau}$ is uniformly drawn for each simulation, and $\boldsymbol{\mu} = (0, \Delta, \ldots, 4\Delta)$.

are due to the window sizes. Furthermore the first Δ at which the proportion of simulations where the criteria gives the correct number of changepoints has surpassed 0.75 is respectively $\Delta = 4.5$, $\Delta = 6$ and $\Delta = 3.5$ in Figures 4.22, 4.23 and 4.24. The lowest value is because $n/(m+1)$ is significantly higher than for the two options, such that there is more information per changepoint interval. And intuitively a method is less at risk to finding too many changepoints when it identifies the changepoints close to where they truly are. For the top two figures $n/(m+1)$ is so close that this effect does not come into play. Instead the top gets a good success rate faster because there are less viable ms to mistakenly detect. For instance when $\Delta = 4$ the method detects either 4, 5 or 6 internal changepoints, while when $n = 105$ and $m = 20$ the methods detect 18, 19, 20 or 21 internal changepoints.

4.4. Preliminary discussion

4.4.1. PELT vs BinSeg

So far in this thesis the main focus has been to study how the PELT algorithm can be used to detect anomalies in independent time series data. We saw in Sections 3 and 4 that this is possible. This section contains remarks and questions that may be answered by further work. An important remark from Section 4 is that a value for Δ may be small, intermediate or large relatively to the other parameters. For intermediate Δ values the algorithms were able to identify the number of changepoints to some extent, but not the correct placement of the changepoints. We opted to call this subset of the parameter space the intermediate region. Further investigations on the behaviour of the intermediate region is of interest.

The main difference between PELT and BinSeg is that PELT finds the optimal set of multiple changepoints, while BinSeg finds one changepoint at a time in an optimal manner. As we saw above when there was no changepoint or one changepoint, then BinSeg performs better or as good as PELT. The simulations in Section 4.1 showed that PELT finds more changepoints than BinSeg when the penalty is the same. This might be seen as BinSeg inducing a lower model complexity. This makes sense as BinSeg does not in general find the optimal model parameter, and thus does not maximize over the entire parameter space in the criterion. If BinSeg for instance has effective degrees of freedom per changepoint interval $edf_{BinSeg} = 0.95edf$, then the penalty $\beta = edf\log(n) = 2\log(n) = 1.05edf_{BinSeg}\log(n)$. So for BinSeg the penalty is relatively higher compared to the number of degrees of freedom. It would be interesting to conduct more simulations and find out what the apparent reduction in model complexity amounts to for BinSeg. The way we propose to do this is to calibrate on 'easy' data sets, where the value for Δ is large. In other words to make the algorithms plateau at the same value in Figure 4.10 for a multitude of data sets. This might also aid people who are faced with a multiple changepoint problem, and who want to switch from BinSeg to PELT when both use the BIC criterion. Both may of course

be implemented with the mBIC criterion. Then the criterion may be strict enough on model complexity that PELT performs better than BinSeg. It would be interesting to see if this is the case, although it is likely that BinSeg would still perform better on data sets with one changepoint and worse on data sets with more changepoints.

Assuming anomalies are relatively infrequent then a data set might be expected to contain at most one anomaly. PELT is tailored to finding multiple changepoints, and it runs in $O(n)$ time when the number of changepoints increases linearly with the length of the data set. When there are few or no changepoints the pruning condition in Equation (3.24) will hold for a small subset of the data points. So under these conditions the runtime of PELT is much longer than the runtime of BinSeg. Additionally we demonstrated in Section 4.1.2 that BinSeg is better at finding the correct changepoint when there is only one changepoint. So assuming anomalies are relatively infrequent PELT is relatively poorly suited to analyze the residuals as indicated in Figure 1.1.

In the raw sensor data from a ship there are many changepoints. An alternative use for PELT is then to apply it with a time series cost function per interval to the raw sensor data. The nature of the changepoints may be categorized by another application, and when a new type of change is found it may be labeled an anomaly.

4.4.2. Online application

The most straight forward way to use PELT and BinSeg in an online setting is to run them for every new data point, or for every few new data points. This is analogous to the simulation in Section 4. The simulation can be seen as investigating the same property as is quantified in the ARL defined in Equation (2.1). ARL is the expected time until a changepoint is detected when there is no changepoint. Receiving a data set in real time online algorithms typically terminate once they find a changepoint. In our setting we may see this as the algorithm having detected an anomaly, and thus does not look for more.

For BinSeg the probability of correctly identifying that there are no changepoints increase with the length of the data set. Life time analysis may be utilized to find the approximate ARL if we analyze much longer simulations. The ARL depends heavily on the upper tail of the distribution of the time until a changepoint is detected. Since the probability to correctly find the changepoint is high and increases when the data set is of length 1 to 50000, we assume the upper tail is heavy when PELT or BinSeg is applied in this fashion. Figure 4.9 shows proportion of simulations where BinSeg and PELT identified the changepoint position at the different locations. This is not comparable to EDD as EDD says something about when sequentially the changepoint was located.

A challenge when applying an algorithm online is that when it at time t

detects that a certain number of changepoints has occurred, we do not want it to reach a radically different conclusion at time $t + 1$. In other words we want it to be *stable*. For instance if it reaches the conclusion at time $t = 1000$ that there are internal changepoints at $(5, 100, 877)$, we do not want it to identify $(5, 11, 500)$ as internal changepoints at $t = 1001$.

The graph in Figure 4.2 illustrates that the naive online application of PELT and BinSeg is unstable even when there are no changepoints, in the sense that a changepoint detected in such a fashion might be decided not to be present after observing more data points. One conclusion from this is that an online application would preferably apply PELT or BinSeg in a more ingenious fashion. An interesting topic is whether it is possible to redesign PELT as an online sequential algorithm.

PELT is a sequential algorithm, where each step is computed in $O(n)$ time. It is possible to set up PELT to run sequentially with time, but that leaves the penalty value out in the open. For a long data set a small penalty will make PELT identify too many changepoints, but if the penalty is large when the data set is still short actual changepoints may not be identified. An alternative is to try to introduce a moving window concept, where only the recent data points are considered.

As illustrated in Figure 3.7 PELT keeps multiple competing solutions. The concept of the algorithm is to link each data point to one solution, and to keep only the solutions that may turn out to minimize the total cost for the full data set. And so it is not stable in the sense above. However when all solutions kept at a time involve a specific changepoint, that changepoint is guaranteed to be a part of the optimal solution. It would be interesting to explore whether this could be used to make PELT readily applicable in an online setting.

4.4.3. BIC vs mBIC

The criterion BIC_2 is an approximation to the mBIC from Zhang and Siegmund (2007), and thus we will refer to it as mBIC in this section. However first we would like to note that it would be interesting to compute the optimal parameters of mBIC and of BIC_2 and compare the solutions to see when they differ, and by how much. Both the ture mBIC criterion and BIC_2 makes sure that the changepoints are more likely to be clustered together as this gives a lower penalty, but the tendency might be far stronger with BIC_2 as it is a direct part of the cost function.

Both BIC and mBIC are large sample approximations to the Bayes Factor. As seen from Equations (3.4) and (3.12) the likelihood terms are equivalent, but in the limit as n goes to infinity the BIC penalty is $2m \log n$ while the mBIC penalty may be up to $3m \log n$, as seen in Equation (3.11). This illustrates that there is not one intuitive way for the sample size of a changepoint dataset to increase. If we assume that the positions $\tau_1, \ldots, \tau_m$ of the changepoints are constant while the length of the last changepoint interval

increases to infinity, then mBIC and BIC have the same limiting penalty value. However mBIC is developed under the assumption that $\frac{1}{n}\tau_1, \ldots, \frac{1}{n}\tau_m$ stay constant while the sample size n grows. A third assumption is that the number of changepoints increases linearly with n as n increases, and is the assumption under which PELT runs in $O(n)$ time (Killick et al., 2012a). The difference in performance on the simulated data sets between the BIC and mBIC criteria in Section 4.3 is not as large as the difference between BinSeg and PELT with BIC in Section 4.1. Furthermore when we use BIC we find the correct number of changepoints with slightly lower Δ. However with mBIC we find the correct changepoint vector for lower Δ, and with mBIC we stabilize at finding the correct m for a higher proportion of the simulations. The differences between the performance of the two criteria in Section 4.3 is underwhelming, but they illustrate for what types of data mBIC performs better than BIC. It also illustrates that when mBIC performs better, it is because the penalty of BIC is not high enough. That is using BIC we find more changepoints than with mBIC. It would be interesting to perform the same study with longer data sets and see at what lengths of the interval data set the difference between the performance of the two criteria becomes considerable for different types of applications. On a side note there are more terms of mBIC that may be included for it to perform better on short data sets, although that would introduce a bias (Zhang and Siegmund, 2007).

Although the performance with BIC and mBIC differ marginally on these data sets we would use mBIC on real data, because the performance is better sometimes. We assume that the performance will be similar on data sets with only one changepoint as well. Since the minimal penalty of mBIC is smaller than the penalty of BIC, mBIC may perform as good or better. It would also be interesting to investigate this further. Furthermore we want to know hoe BIC and mBIC perform on real data. One distinction is that in many applications the data is from a distribution with an unknown variance, or with a known variance different from 1. Thus in the next section we will look at criteria that accounts for this, and a PELT algorithm that may compute some of the some criteria.

5. Multi-parameter changepoint detection with PELT

So far in the thesis we have detailed changepoint model selection when there is one parameter only in the Gaussian data; the mean. Now we generalize the theory and the algorithms to account for the variance of the univariate Gaussian distribution. The resulting changepoint model is presented in Section 5.1, and the likelihood of each changepoint interval is found in Section 5.2. In Section 3 we saw that the aim of developing a likelihood is to find cost functions that may be used with PELT when we maximize a model selection criterion. Hence we study the likelihood based cost functions in detail and develop the likelihood based cost functions in Section 5.3. Similar cost functions for multivariate Gaussian data are presented in Appendix A. In Sections 5.4 and 5.5 we also present model selection criteria that are based on approximations of the Bayes Factor for univariate Gaussian data. Then in Section 5.6 we detail the generalizations of OP and PELT that make it possible to estimate more than one parameter on each changepoint interval.

5.1. The changepoint model

In the rest of the thesis we will consider the changepoint detection problem when there are multiple parameters and Gaussian data. In this section we thus substitute Assumption 3.1 for either of Assumptions 5.1 and 5.2 below.

Assumption 5.1. The data set $x_1, x_2 \ldots, x_n$ are realizations of $X_j \sim f_j,\ j \in \{j, j+1\}$ such that $f_j = \mathcal{N}(\mu_j, \sigma^2)$, with σ^2 is a known number in $\mathbb{R}$ and $\mu_j \neq \mu_{j+1}$ for $j \in (1, m+1)$. For all $i \neq j$ also X_i and X_j are independent.

Assumption 5.2. The data set $x_1, x_2 \ldots, x_n$ are realizations of $X_j \sim f_j,\ j \in \{j, j+1\}$ such that $f_j = \mathcal{N}(\mu_j, \sigma_j^2)$, $\mu_j \neq \mu_{j+1}$ and $\sigma_j^2 \neq \sigma_{j+1}^2$ for $j \in (1, m+1)$. For all $i \neq j$ also X_i and X_j are independent.

Note that the only difference between the three assumptions is the distribution for f_j. Assumptions 3.1 and 5.1 only require us to estimate one parameter per interval, while we must estimate two parameters per interval under 5.2.

The changepoint model in this section is the same as in Section 3.1 except that we introduce one more term that will be important when we estimate more than one parameter. This was not introduced before as it complicates the algorithms as we will see in Section 5.6, and since it is not necessary when there is only one parameter per interval.

Definition 5.1. The *minimum segment length* g is

$$g = \min_{j \in \{1,\ldots,m+1\}} (\tau_j - \tau_{j-1}),$$

where the τ_js are the true changepoint positions defined in Section 3.1.

We also define that the *observed minimum segment length* is

$$\min_{j\in\{1,\ldots,m+1\}} (\tau_j - \tau_{j-1})$$

when the τ_js are the changepoint positions of some prospective solution. We will make sure that the observed minimal segment length is never shorter than what we assume to be the minimal segment length. That is when we evaluate the total cost of a solution we will operate with a new cost function labeled C_g with domain restricted to $s, t \in \mathbb{N}$ such that $t - s + 1 \geq g$. This cost function may be defined as

$$C_g(s,t) = C(s,t), \quad \text{for } t - s + 1 \geq g, \tag{5.1}$$

where C is some cost function that applies when $g = 1$. The new cost function requirement is then that

$$C_g(t_1 + 1, t_3) \geq C_g(t_1 + 1, t_2) + C_g(t_2 + 1, t_3), \tag{5.2}$$

when $t_3 - t_2 \geq g$ and $t_2 - t_1 \geq g$. When it is clear from the context what the value of g is, it will be omitted. Also as the true minimum segment length is in general not known we will refer to the assumed g in Equation (5.1) as the minimum segment length as well.

5.2. Likelihood of a changepoint interval

All observations are independent and thus the observations on each interval in the data set are independent of the observations on the other intervals for a given set of intervals. Hence we may find the maximum likelihood estimates of the parameters by only considering one interval at a time. We will label the observations on the interval $\boldsymbol{x} = (x_s, \ldots, x_t)$, and they are realizations of the random variables $\boldsymbol{X} = (X_s, \ldots, X_t)$.

Assume then that we have one random variable with probability density function

$$f(x|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x-\mu)^2}{2\sigma^2},$$

where $\exp a$ signifies Eulers constant e to the power of a. In this section the parameter vector is $\theta = (\mu, \sigma^2)$. For $t - s + 1$ independent observations the probability density function is

$$f(\boldsymbol{x}|\theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{t-s+1} \exp -\frac{1}{2\sigma^2}\sum_{i=s}^{t}(x_i-\mu)^2,$$

the log likelihood is

$$\ell(\mu, \sigma^2) = -\frac{t-s+1}{2}\left(\log 2\pi + \log \sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=s}^{t}(x_i-\mu)^2, \tag{5.3}$$

and $(x_i - \mu)^2 = (x_i^2 - 2x_i\mu + \mu^2)$. We may solve the equations $\frac{\partial}{\partial\mu}\ell(\mu, \sigma^2) = 0$ and $\frac{\partial}{\partial\sigma^2}\ell(\mu, \sigma^2) = 0$ separately to get the maximum likelihood estimates if the parameters are orthogonal, that is if the off diagonal elements of the Fisher information matrix are zero. The Fisher information matrix in this setting is (Casella and Berger, 2002)

$$\mathcal{I}(\theta) = -\operatorname{E}_\theta\left(\frac{\partial}{\partial\theta^2}\log f(\boldsymbol{X}|\theta)\right),$$

which written out is

$$\mathcal{I}(\theta) = -\operatorname{E}\begin{bmatrix} \frac{\partial^2}{\partial^2\mu}\log f(\boldsymbol{X}|\theta) & \frac{\partial^2}{\partial\sigma^2\partial\mu}\log f(\boldsymbol{X}|\theta) \\ \frac{\partial^2}{\partial\mu\partial\sigma^2}\log f(\boldsymbol{X}|\theta) & \frac{\partial^2}{\partial^2\sigma^2}\log f(\boldsymbol{X}|\theta) \end{bmatrix}. \tag{5.4}$$

Given that the off-diagonal entries of the Fisher information matrix are zero, then the diagonal entries need to be positive for there to be a maximum likelihood estimate for each parameter. We will proceed to compute each component of $\frac{\partial}{\partial\theta^2}\log f(\boldsymbol{X}|\theta)$.

When we differentiate $\log f(\boldsymbol{X}|\theta)$ with respect to μ we get

$$\begin{aligned}\frac{\partial}{\partial\mu}\log f(\boldsymbol{X}|\theta) &= \frac{\partial}{\partial\mu}\left(-\frac{t-s+1}{2}\left(\log 2\pi + \log\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=s}^{t}(X_i-\mu)^2\right) \\ &= -\frac{1}{2\sigma^2}\sum_{i=s}^{t}(-2X_i + 2\mu),\end{aligned}$$

and we get

$$\frac{\partial}{\partial\sigma^2}\log f(\boldsymbol{X}|\theta) = \frac{\partial}{\partial\sigma^2}\left(-\frac{t-s+1}{2}\left(\log 2\pi + \log\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=s}^{t}(X_i-\mu)^2\right)$$

$$= -\frac{t-s+1}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=s}^{t}(X_i-\mu)^2.$$

When we differentiate with respect to σ^2. Then in turn

$$\frac{\partial^2}{\partial^2\mu}\log f(\boldsymbol{X}|\theta) = \frac{\partial}{\partial\mu}\left(-\frac{1}{2\sigma^2}\sum_{i=s}^{t}(-2X_i+2\mu)\right) = -\frac{1}{\sigma^2}, \tag{5.5}$$

and

$$\frac{\partial^2}{\partial^2\sigma^2}\log f(\boldsymbol{X}|\theta) = \frac{\partial}{\partial\sigma^2}\left(-\frac{t-s+1}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=s}^{t}(X_i-\mu)^2\right) \tag{5.6}$$

$$= \frac{t-s+1}{2\sigma^4} - \frac{1}{\sigma^6}\sum_{i=s}^{t}(X_i-\mu)^2.$$

Also

$$\frac{\partial^2}{\partial\sigma^2\partial\mu}\log f(\boldsymbol{X}|\theta) = \frac{\partial}{\partial\sigma^2}\left(-\frac{1}{\sigma^2}\sum_{i=s}^{t}(-X_i+\mu)\right) = \frac{1}{\sigma^4}\sum_{i=s}^{t}(-X_i+\mu) \tag{5.7}$$

and

$$\frac{\partial^2}{\partial\mu\partial\sigma^2}\log f(\boldsymbol{X}|\theta) = \frac{\partial}{\partial\mu}\left(-\frac{t-s+1}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=s}^{t}(X_i^2-2X_i\mu+\mu^2)\right)$$

$$= \frac{1}{\sigma^4}\sum_{i=s}^{t}(-X_i+\mu).$$

Since the Fisher information matrix is defined by the expected values we will need to find the mean $\mathrm{E}\,X_i$ for Equation (5.7).

$$\mathrm{E}\,X_i = \int_{x_i=-\infty}^{\infty}\frac{x_i}{\sqrt{2\pi\sigma^2}}\exp-\frac{(x_i-\mu)^2}{2\sigma^2}\mathrm{d}x_i.$$

With the change of variables $x = x_i - \mu$ we get $x_i = x + \mu$ and $\mathrm{d}x_i = \mathrm{d}x$. Hence

$$\mathrm{E}\,X_i = \int_{x=-\infty}^{\infty}\frac{x+\mu}{\sqrt{2\pi\sigma^2}}\exp-\frac{(x)^2}{2\sigma^2}\mathrm{d}x,$$

where $x\exp-x^2$ is odd such that

$$\mathrm{E}\,X_i = \mu\int_{x=-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}\exp-\frac{(x)^2}{2\sigma^2}\mathrm{d}x \tag{5.8}$$

$$= \mu\int_{x=-\infty}^{\infty}\log f(x|\mu=1,\sigma^2)\mathrm{d}x = \mu.$$

For Equation (5.6) we also need to compute the variance $\mathrm{E}(X_i - \mu)^2$, that is

$$\begin{aligned}\mathrm{E}(X_i - \mu)^2 &= \int_{x_i=-\infty}^{\infty} \frac{(x_i-\mu)^2}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x_i-\mu)^2}{2\sigma^2} \mathrm{d}x_i. \\ &= \sigma\sqrt{\frac{2}{\pi}} \int_{x_i=-\infty}^{\infty} \left(\frac{x_i-\mu}{\sqrt{2}\sigma}\right)^2 \exp - \left(\frac{x_i-\mu}{\sqrt{2}\sigma}\right)^2 \mathrm{d}x_i.\end{aligned}$$

We will use the change of variables $x = \left(\frac{x_i-\mu}{\sqrt{2}\sigma}\right)^2$ which gives

$$\frac{\mathrm{d}x}{\mathrm{d}x_i} = \frac{2}{\sqrt{2}\sigma}\left(\frac{x_i-\mu}{\sqrt{2}\sigma}\right) = \frac{\sqrt{2}}{\sigma}\sqrt{x},$$

such that

$$\begin{aligned}\mathrm{E}(X_i - \mu)^2 = &\ \sigma\sqrt{\frac{2}{\pi}}\frac{\sigma}{\sqrt{2}} \int_{x_i=-\infty}^{\infty} \frac{x}{\sqrt{x}} \exp -x \ \mathrm{d}x \\ &= 2\sigma\sqrt{\frac{2}{\pi}}\frac{\sigma}{\sqrt{2}} \int_{x=0}^{\infty} \frac{x}{\sqrt{x}} \exp -x \ \mathrm{d}x \\ &= \frac{2\sigma^2}{\sqrt{\pi}} \int_{x=0}^{\infty} x^{\frac{3}{2}-1} \exp -x \ \mathrm{d}x = \frac{2\sigma^2}{\sqrt{\pi}}\Gamma\left(\frac{3}{2}\right),\end{aligned}$$

where the Gamma function is $\Gamma(\alpha) = \int_{x=0}^{\infty} x^{\alpha-1} \exp -x \mathrm{d}x$. Since $\Gamma(\frac{3}{2}) = \frac{\sqrt{\pi}}{2}$ thus

$$\mathrm{E}(X_i - \mu)^2 = \sigma^2. \tag{5.9}$$

Then we compute each of the components of the Fisher information matrix from Equations (5.5),(5.6) and (5.7) using Equations (5.8) and (5.9) that is

$$\begin{aligned}&\mathrm{E}\,\frac{\partial^2}{\partial^2\mu}\log f(\boldsymbol{X}|\theta) = -\frac{1}{\sigma^2} \\ &\mathrm{E}\,\frac{\partial^2}{\partial\sigma^2\partial\mu}\log f(\boldsymbol{X}|\theta) = \mathrm{E}\,\frac{\partial^2}{\partial\mu\partial\sigma^2}\log f(\boldsymbol{X}|\theta) = \frac{1}{\sigma^4}\sum_{i=s}^{t}\mathrm{E}(-X_i+\mu) = 0 \\ &\mathrm{E}\,\frac{\partial^2}{\partial^2\sigma^2}\log f(\boldsymbol{X}|\theta) = \frac{t-s+1}{2\sigma^4} - \frac{1}{\sigma^6}\sum_{i=s}^{t}\mathrm{E}(X_i-\mu)^2 \\ &= \frac{t-s+1}{2\sigma^4} - \frac{t-s+1}{\sigma^4} = -\frac{t-s+1}{2\sigma^4}.\end{aligned}$$

When we insert this into Equation (5.4) we get

$$\mathcal{I}(\theta) = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{t-s+1}{2\sigma^4} \end{bmatrix},$$

such that we may find the maximum likelihood estimates by differentiating with respect to one parameter at a time as the diagonal elements are both positive, and the off-diagonal elements are both zero. Thus we get the maximum likelihood estimates

$$\hat{\mu} = \frac{1}{t-s+1}\sum_{i=s}^{t} x_i \tag{5.10}$$

from

$$\frac{\partial}{\partial\mu}\ell(\mu,\sigma^2) = \left(-\frac{1}{2\sigma^2}\sum_{i=s}^{t}(-2x_i+2\mu)\right) = 0,$$

and

$$\hat{\sigma}^2 = \frac{1}{t-s+1}\sum_{i=s}^{t}(x_i-\hat{\mu})^2 \tag{5.11}$$

from

$$\frac{\partial}{\partial\sigma^2}\ell(\mu,\sigma^2) = -\frac{t-s+1}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=s}^{t}(x_i-\hat{\mu})^2 = 0,$$

where the partial derivatives of the log likelihood are seen from Equations (5.6) and (5.5).

5.3. Likelihood maximization with PELT

In the current section we show how to use the likelihood that was treated in the last section to create cost functions for PELT similar to the cost functions in Section 3.4. We also prove that a certain type of likelihood based cost function can always be used with PELT, and study these likelihood based cost functions in detail.

5.3.1. Likelihood based cost functions

The cost function requirement from Equation (5.2) needs to be fulfilled in order for PELT to be guaranteed to find the same optimal solution that OP would give. Now we will show that minus twice the log likelihood of the observations give rise to a cost function that satisfies the requirement. In order to do that we need the following result on the log likelihood of observations.

Definition 5.2. Denote by $\ell(i, \theta)$ the logarithm of the likelihood of observation x_i with parameter vector θ, and denote by $\ell(s, t, \theta) = \sum_{i=s}^{t} \ell(i, \theta)$ the logarithm of the likelihood of independent and identically distributed observations $x_s, \ldots, x_t$ when the parameter vector is θ. Furthermore let $\ell(s, t, \hat{\theta})$ denote $\ell(s, t, \theta)$ with parameter vector $\hat{\theta}$ being the maximum likelihood estimator based on observations $x_s, \ldots, x_t$. When it is clear from the context s and t are omitted from $\ell(s, t, \theta)$.

According to Definition 5.2 the likelihood of a data set under the change-point model is

$$\ell(\theta) = \sum_{j=1}^{m+1} \ell(\tau_{j-1} + 1, \tau_j, \theta_j), \tag{5.12}$$

where θ_j are the parameters of that interval, for instance σ_j^2 and μ_j. In the following theorem we present a property of the likelihood at the maximum likelihood estimates that we will make use of promptly.

Theorem 5.1. As illustrated in Figure 5.1 let $\hat{\theta}_0$, $\hat{\theta}_{1,1}$ and $\hat{\theta}_{1,2}$ be the maximum likelihood estimators of θ based on respectively $x_{t_1}, \ldots, x_{t_3}$, $x_{t_1}, \ldots, x_{t_2}$, and $x_{t_2+1}, \ldots, x_{t_3}$ such that

$$\ell(t_1, t_2, \hat{\theta}_0) \leq \ell(t_1, t_2, \hat{\theta}_{1,1}) \text{ and } \ell(t_2 + 1, t_3, \hat{\theta}_0) \leq \ell(t_2 + 1, t_3, \hat{\theta}_{1,2}), \tag{5.13}$$

with notation from Definition. Then 5.2

$$\ell(t_1, t_3, \hat{\theta}) \leq \ell(t_1, t_2, \hat{\theta}) + \ell(t_2 + 1, t_3, \hat{\theta}).$$

Proof. The proof is short and simple, and hinges on Equation (5.13).

$$\begin{aligned}
\ell(t_1, t_3, \hat{\theta}) &= \ell(t_1, t_3, \hat{\theta}_0) = \ell(t_1, t_2, \hat{\theta}_0) + \ell(t_2 + 1, t_3, \hat{\theta}_0) \\
&\leq \ell(t_1, t_3, \hat{\theta}_{1,1}) + \ell(t_2 + 1, t_3, \hat{\theta}_{1,2}) \\
&= \ell(t_1, t_3, \hat{\theta}) + \ell(t_2 + 1, t_3, \hat{\theta}).
\end{aligned}$$

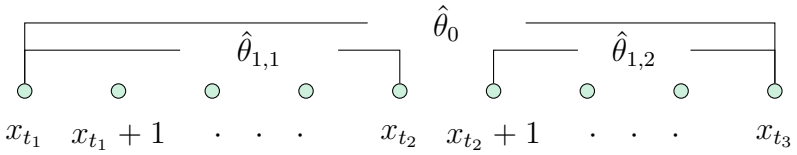


Figure 5.1: Illustration for Theorem 5.1. The maximum likelihood estimators of θ based on respectively $x_{t_1}, \ldots, x_{t_3}$, $x_{t_1}, \ldots, x_{t_2}$, and $x_{t_2+1}, \ldots, x_{t_3}$ are labeled $\hat{\theta}_0$, $\hat{\theta}_{1,1}$ and $\hat{\theta}_{1,2}$.

□

Now we are ready for the theorem presenting the likelihood based cost functions that may be used with PELT

Theorem 5.2. When the log likelihood satisfies the requirements of Theorem 5.1, then

$$C_g(s,t) = -2\ell(s,t,\hat{\theta}) - \sum_{i=s}^{t} a(x_i) \tag{5.14}$$

satisfies the cost function requirement in Equation (5.2) for any set $a(x_s), \ldots, a(x_t) \in \mathbb{R}$ where the values only depend on the value of x in each data point.

Proof. From Theorem 5.1 we get

$$\begin{aligned}
\ell(t_1,t_3,\hat{\theta}) &\leq \ell(t_1,t_2,\hat{\theta}) + \ell(t_2+1,t_3,\hat{\theta}),\\
-2\ell(t_1,t_3,\hat{\theta}) &\geq -2\ell(t_1,t_2,\hat{\theta}) - 2\ell(t_2+1,t_3,\hat{\theta}),\\
-2\ell(t_1,t_3,\hat{\theta}) - \sum_{i=t_1}^{t_3} a(x_i) &\geq -2\ell(t_1,t_2,\hat{\theta}) - 2\ell(t_2+1,t_3,\hat{\theta}) - \sum_{i=t_1}^{t_3} a(x_i).
\end{aligned}$$

Which in turn gives

$$C_g(t_1,t_3,\hat{\theta}) \geq C_g(t_1,t_2,\hat{\theta}) + C_g(t_2+1,t_3,\hat{\theta}),$$

such that the cost function satisfies Equation (5.2). □

Now we know a way to construct cost functions based on the likelihood that fulfill the PELT requirement in Equation (5.13). However as in Section 3.4.1 we want a simple expression for $C_g(s,t)$ such that maximizing a criterion on the form of Equation (2.10) is equivalent to minimizing

$$\sum_{j=1}^{m+1} C_g(\tau_{j-1},\tau_j) + m\beta, \tag{5.15}$$

for some value of β not depending on m and $\boldsymbol{\tau}$. It is evident that when Equation (5.14) fulfills the cost function requirement of PELT in Equation (5.2), then also

$$C_g(s,t) = -2\ell(s,t,\hat{\theta}) \tag{5.16}$$

and $C_g(s,t) = -\ell(s,t,\hat{\theta})$ fulfill the requirement. Also we know from Equation (5.12) that maximizing Equation (2.10) is equivalent to minimizing Equation (5.15) when we use the cost function in Equation (5.16). However the manipulated cost function in Equation (5.14) allows us the freedom of choosing a simpler cost function. In the following section we will demonstrate that the using $-2\ell(s,t,\theta)$ or $-2\ell(s,t,\theta) - \sum_{i=s}^{t} a(x_i)$ as the cost function gives the same minimization of Equation (5.15).

5.3.2. Detailed study of cost functions

In order to maximize some criterion we look for the optimal solution; that is the solution with cost $F(t)$ from Equation (3.19). The object of this section is to present which manipulations of the cost function may be done without it affecting what criterion we are maximizing. The first manipulation is motivated in the last section. Second we look at adding a constant to each segment, and then we multiply $C_g(\tau_{i-1}+1,\tau_i)$ with a constant. We will see that the second manipulation changes what criterion we are maximizing. We will also see that the penalty may be changed such that the third manipulation does not change what criterion we are maximizing.

In Theorem 5.2 we add the sum of a function evaluated on each of the individual data points. The following theorem proves that such a manipulation of the cost function does not affect what is the optimal solution that PELT and OP find. This is important because if the first cost function may be used to optimize some criterion, that also the altered cost function does so.

Theorem 5.3. The optimal solution on a data set $x_1,\ldots,x_n$ with penalty β and cost function C_g is identical to the solution with penalty β and cost function C_g' when

$$C_g(\tau_{i-1}+1,\tau_i) = C_g'(\tau_{i-1}+1,\tau_i) + \sum_{t=\tau_{i-1}+1}^{\tau_i} a(x_t),$$

for some predetermined function $a(x_t)$.

Proof. The optimal solution is the set of τ_is that for a data set $x_1,\ldots,x_n$

with $n = \tau_{m+1}$ minimizes

$$\begin{aligned} p(t) &= \sum_{i=1}^{m+1} C_g(\tau_{i-1}+1, \tau_i) + m\beta + B \\ &= \sum_{i=1}^{m+1} \left(C'_g(\tau_{i-1}+1, \tau_i) + \sum_{t=\tau_{i-1}+1}^{\tau_i} a(x_t) \right) + m\beta + B \\ &= \sum_{i=1}^{m+1} C'_g(\tau_{i-1}+1, \tau_i) + m\beta + \sum_{t=1}^{\tau_{m+1}} a(x_t) + B, \end{aligned}$$

where B is some constant independent of m and $\tau_1, \ldots, \tau_m$. Any set of τ_is that minimizes $p(t)$ also minimizes $p'(t) = p(t) - \sum_{t=1}^{\tau_{m+1}} a(x_t)$ since $\sum_{t=1}^{\tau_{m+1}} a(x_t)$ is a constant. Thus the PELT solution on a data set with penalty β and cost function C_g is identical to the solution with penalty β and cost function C'_g. □

The most notable function $a(x_t)$ is $a(x_t) = 1$, with $\sum_{t=1}^{\tau_{m+1}} 1 = \tau_i - \tau_{i-1}$. This means that two cost functions C_g and C'_g such that

$$C'_g(\tau_{i-1}+1, \tau_i) = C_g(\tau_{i-1}+1, \tau_i) + (\tau_i - \tau_{i-1})$$

give the exact same optimal solution. Such a manipulation of the cost function also guarantees that if C_g fulfills the PELT cost function requirement, then C'_g also does, since $C_g(t_1+1, t_3) \geq C_g(t_1+1, t_2) + C_g(t_2+1, t_3)$ implies that $C'_g(t_1+1, t_3) \geq C'_g(t_1+1, t_2) + C'_g(t_2+1, t_3)$. Another result from this that might seem strange at first is that the solution cost $p(t)$ might well be negative, and this has no consequence. If we set $a(x_t)$ to be a large enough positive constant then any solution may be given a positive total cost. When the cost of two solutions are $p^1(t)$ and $p^2(t)$, and the solution with cost $p^1(t)$ is the optimal one then $p^2(t) - p^1(t) \geq 0$.

We now look at what happens to the optimal solution if we add a constant to each segment cost, such that $C'_g(\tau_{i-1}+1, \tau_i) = C_g(\tau_{i-1}+1, \tau_i) + b$. Then the cost of the former, $p'(t)$, is

$$\begin{aligned} p'(t) &= \sum_{i=1}^{m+1} (C_g(\tau_{i-1}+1, \tau_i) + b) + m\beta + B \\ p'(t) &= \sum_{i=1}^{m+1} (C_g(\tau_{i-1}+1, \tau_i)) + m(\beta + b) + (B + b). \end{aligned}$$

Since $B + b$ is a constant the alteration of the cost function is equivalent to a change in the constant penalty term β. This implies that when $b < 0$ it makes sense to make sure that $\beta + b \geq 0$. When $b < 0$ we get that C'_g satisfies the PELT cost function requirement whenever C_g does. However if the old cost function C_g satisfied the PELT cost function requirement, then

the new cost function C'_g does not when $b > 0$. A simple counter example is when $C_g(\tau_{i-1} + 1, \tau_i) = 0$. Then

$$0 = C_g(t_1 + 1, t_3) \geq C_g(t_1 + 1, t_2) + C_g(t_2 + 1, t_3) = 0,$$

but

$$b = C'_g(t_1 + 1, t_3) < C'_g(t_1 + 1, t_2) + C'_g(t_2 + 1, t_3) = 2b,$$

for $b > 0$.

Finally we study the effect of multiplying the segment cost with a number not depending on the model parameters, such that $C'_g(\tau_{i-1} + 1, \tau_i) = bC_g(\tau_{i-1} + 1, \tau_i)$. It turns out that with an adjustment to β the resulting optimal solutions are identical, such that the two sets of cost function and penalty result in the maximization of the same criterion.

Theorem 5.4. When $C'_g(\tau_{i-1} + 1, \tau_i) = bC_g(\tau_{i-1} + 1, \tau_i)$ and $\beta' = b\beta$, such that the total costs are are

$$p(t) = \sum_{i=1}^{m+1} C_g(\tau_{i-1}+1, \tau_i)+m\beta+B \text{ and } p(t) = \sum_{i=1}^{m+1} C'_g(\tau_{i-1}+1, \tau_i)+m\beta'+B,$$

then the optimal solutions to the two problems on a data set $x_1, \ldots, x_n$ are identical.

Proof. Inserting $C'_g(\tau_{i-1} + 1, \tau_i) = bC_g(\tau_{i-1} + 1, \tau_i)$ and $\beta' = b\beta$ into the expression for $p(t)$ yields

$$\begin{aligned} p(t) &= \frac{1}{b} \sum_{i=1}^{m+1} C'_g(\tau_{i-1} + 1, \tau_i) + m\frac{1}{b}\beta' + B \\ &= \frac{1}{b}p'(t) + (1 - \frac{1}{b})B. \end{aligned}$$

Since $(1 - \frac{1}{b})B$ and b are constants then $\min p(t)$ and $\min(p'(t))$ are attained for the same solution and the optimal solutions to the two problems are identical. Thus the cost functions and penalties result in the maximization of the same criterion. □

5.3.3. Estimate the mean only

First we assume that σ^2 is known. Now we will use twice the negative log likelihood to create a cost function that results in the maximization of the likelihood when it is applied with PELT under Assumption 5.1. From Equation (5.3) we get that

$$\begin{aligned} -2l(\hat{\mu}_j, \sigma^2) &= (t - s + 1)(\log 2\pi + \log \sigma^2) + \frac{1}{2\sigma^2} \sum_{i=s}^{t} (x_i^2 - 2\hat{\mu}_j x_i + \hat{\mu}_j^2), \\ &= \sum_{i=s}^{t} a(x_i) + \frac{1}{\sigma^2} \sum_{i=1}^{n} (-\hat{\mu}_j x_i + \hat{\mu}_j^2), \end{aligned}$$

with $a(x_i) = \log 2\pi + \log \sigma^2 + x_i^2$ and $\hat{\mu}_j$ from Equation (5.10). Then according to Theorems 5.2 and 5.3 a cost function for observations $x_s, \ldots, x_t$ that satisfies the cost function requirement in Equation (5.2) and corresponds to minimizing minus twice the log likelihood is

$$C(s,t) = \frac{1}{\sigma^2} \sum_{i=s}^{t} (-2\hat{\mu}_j x_i + \hat{\mu}_j^2),$$

when the value for σ is known. Another such cost function is

$$C(s,t) = \frac{1}{\sigma^2} \sum_{i=s}^{t} (x_i - \hat{\mu}_j)^2. \tag{5.17}$$

While the latter is strictly positive, the former may take on negative values. Even though it might seem counter intuitive for a cost to be negative, there is no such restriction on the cost. Due to Theorem 5.3 the former and the latter cost functions give the same optimal solution. It is the difference between the costs that determines which solution is optimal, so the cost function may take on any finite value in $\mathbb{R}$.

5.3.4. Estimate the mean and variance

This time we follow Assumption 5.2 when we find the cost function. Since

$$\frac{1}{\hat{\sigma}_j^2} \sum_{i=s}^{t} (x_i - \hat{\mu}_j)^2 = \frac{\hat{\sigma}_j^2 (t - s + 1)}{\hat{\sigma}_j^2},$$

then from Equation (5.3)

$$\begin{aligned} -2l(\hat{\mu}_j, \hat{\sigma}_j^2) &= (t - s + 1)(\log 2\pi + \log \hat{\sigma}_j^2 + 1), \\ &= \sum_{i=s}^{t} a(x_i) + (t - s + 1) \log \hat{\sigma}_j^2, \end{aligned} \tag{5.18}$$

this time with $a(x_i) = \log 2\pi + 1$. A cost function that satisfies the requirement in Equation (5.2) and corresponds to minimizing minus twice the log likelihood is

$$C_1(s,t) = (t - s + 1) \log \hat{\sigma}_j^2, \tag{5.19}$$

due to Theorems 5.2 and 5.3, where the number 1 signifies that $t - s + 1 \geq 1$ following the notation defined in Equation (5.1). However there is a problem when $s = t$. Then $\hat{\mu} = x_s$ such that from Equation (5.11) $\hat{\sigma}^2 = 0$ and $C(s,s) = -\infty$. When the cost of one segment is minus infinity, then the selection of the other segments is arbitrary since they would all give the same total cost. Therefore Equation (5.19) is not a cost function that may be used in either OP or PELT. A reasonable cost function is instead

$$C_2(s,t) = (t - s + 1) \log \hat{\sigma}_j^2, \tag{5.20}$$

where we now restrict $t - s + 1 \geq 2$, and this also satisfies the requirement in Equation (5.2). Written out using the formula for $\hat{\sigma}^2$ in Equation (5.11)

$$C_2(s,t) = (t-s+1)\left(\log \sum_{i=s}^{t}(x_i - \hat{\mu}_j)^2 - \log(t-s+1)\right).$$

5.4. Model selection when the variance is known

In this section we consider model selection based on approximations of Equation (2.10) under Assumption 5.1. All the criteria are written on the form from Equation (3.16), which we want to maximize with respect to the choice of m and $\boldsymbol{\tau}$.

5.4.1. BIC

As the parameters we estimate are the same as in Assumption 3.3 we may interpret there to be $2m+1$ degrees of freedom here as well. The appropriate cost function under this assumption is displayed in Equation (5.17). In other words with these assumptions the criterion is to minimize

$$\text{BIC}_3 = -\sum_{j=1}^{m+1} \frac{1}{\sigma^2} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - 2m \log n. \tag{5.21}$$

This is quite similar to BIC_1 in Equation (3.4), the only difference being the factor $\frac{1}{\sigma^2}$ in the first term. According to Theorem 5.4 minimizing BIC_3 is equivalent to minimizing BIC_1, except that instead of the β in Equation (3.21) we use $\beta = 2\sigma^2 \log n$. That is the same $\boldsymbol{\tau}$ minimizes BIC_3 and

$$-\sum_{j=1}^{m+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - 2\sigma^2 m \log n,$$

because we have multiplied both the penalty and each interval cost function with σ^2. This means that for a given data set the quantity and placements of the changepoints we identify depend on the value of σ^2. This makes sense intuitively as for instance the data set $(0.1, 0.5, 2.5, -3.2)$ might be assumed to have $m = 3$ if $\sigma^2 = 0.01$, but $m = 2$ is $\sigma^2 = 1$ and $m = 0$ if $\sigma^2 = 10$.

Similarly the version of BIC_3, adjusted to take into account that according to mBIC the degrees of freedom contributed by each element of $\boldsymbol{\tau}$ is arguably some number d generally different from 1, is

$$\text{BIC}_{3,adj} = -\sum_{j=1}^{m+1} \frac{1}{\sigma^2} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - (d+1)m \log n.$$

This criterion is in turn similar to Equation (3.13). Maximizing this criterion is thus equivalent to maximizing

$$-\sum_{j=1}^{m+1} \sum_{i=\tau_{j-1}+1}^{\tau_j} (x_i - \hat{\mu}_j)^2 - (d+1)\sigma^2 m \log n. \tag{5.22}$$

5.4.2. mBIC

According to Zhang and Siegmund (2007) we may assume that $\sigma^2 = 1$ without loss of generality when σ^2 is known in what is presented in this thesis as Theorem 3.1. Thus only the expressions for the likelihoods in Equations

(3.5) and (3.6) change when Assumption 5.1 is substituted for Assumption 3.1 in Theorem 3.1. This is the same element that is changed between BIC_1 in Equation (3.4) and BIC_3 in Equation (5.21). If we know the means and the variance of a changepoint data set we may scale it to have another variance. This will not change the value of the likelihood at the maximum likelihood estimate as long as we compute the likelihood with the correct variance. And so the approximation to the mBIC criterion in Equation (3.12) becomes

$$\text{BIC}_4 = -\sum_{j=1}^{m+1}\sum_{i=\tau_{j-1}+1}^{\tau_j}\frac{1}{\sigma^2}(x_i-\hat{\mu}_j)^2 - \left(\sum_{j=1}^{m+1}\log\frac{\tau_j-\tau_{j-1}}{n} + 3m\log n\right). \tag{5.23}$$

Thus with known σ^2 the cost function is

$$C(\tau_{j-1}+1,\tau_j) = \frac{1}{\sigma^2}\sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2 + \log\frac{\tau_j-\tau_{j-1}}{n}, \text{ and } \beta = 3\log n.$$

According to Theorem 5.4 the criterion is also maximized when we find the optimal solution with the cost and penalty

$$C(\tau_{j-1}+1,\tau_j) = \sum_{i=\tau_{j-1}+1}^{\tau_j}(x_i-\hat{\mu}_j)^2 + \sigma^2\log\frac{\tau_j-\tau_{j-1}}{n}, \text{ and } \beta = 3\sigma^2\log n,$$

and it is not possible to remove the value of σ^2 completely from this cost function as it was for BIC_3 in the previous section.

5.4.3. Range of penalties (CROPS)

In Section 3.5 we mentioned several changepoint detection algorithms, among them CROPS (Haynes et al., 2017a). In this section we will present some applications of CROPS. In some applied settings where Assumption 5.1 holds but we only know the approximate value of σ^2 we may take advantage of Equation (5.22). On the other hand since σ^2 may not be removed from the cost function of BIC_4 we may not easily maximize the BIC_4 criterion for an unknown constant σ^2. However when we maximize Equation (5.22) the β in Equation (3.16) may be set to be in the range that corresponds to the possible values for σ^2 and with a value for d that is expected to be similar to the *edf* of $\boldsymbol{\tau}/m$ in Equation (3.15). That is β may be set to be in in the range such that

$$\begin{aligned}\min\beta &= (\min \mathit{edf}+1)(\min\sigma^2)\log n, \text{ and}\\ \max\beta &= (\max \mathit{edf}+1)(\max\sigma^2)\log n.\end{aligned}$$

From Equations (3.15) and (3.10)

$$\begin{aligned}\min \mathit{edf} &= \frac{m\log n + \log\frac{n-m}{n}}{m\log n}\\ &= 1 + \frac{\log(1-\frac{m}{n})}{m\log n}\end{aligned}$$

and from Equation (3.11)

$$\begin{aligned}\max edf &= \frac{2m\log n-(m+1)\log(m+1)}{m\log n}\\ &= 2-(1+\frac{1}{m})\frac{\log(m+1)}{\log n}.\end{aligned}$$

Knowing the approximate value for m will reduce the range it is necessary to investigate. As β needs to be large in order for us to detect few internal changepoint, and small in order for us to detect many internal changepoints

$$\begin{aligned}\min\beta &= (2+\frac{\log(1-\frac{\max m}{n})}{(\max m)\log n})(\min\sigma^2)\log n, \text{ and} \qquad (5.24)\\ \max\beta &= \left(3\log n-(1+\frac{1}{\min_{m\geq 1} m})\log(1+\min_{m\geq 1} m)\right)(\max\sigma^2),\end{aligned}$$

where $m\geq 1$ since when $m=0$ any number that is sufficiently large is an appropriate value for β, and so there is no upper limit. If it is possible that $m=0$ then the value of the upper limit should be increased beyond the value for $\max\beta$ in the previous equation.

Because we have $\max m = n-1$ and the minimal possible m we can insert into the expression is $m=1$ we get

$$\begin{aligned}\min_m\beta &= (2+\frac{\log(1-\frac{n-1}{n})}{(n-1)\log n})(\min\sigma^2)\log n\\ &= (2+\frac{\log(n-(n-1))-\log n}{(n-1)\log n})(\min\sigma^2)\log n\\ &= (2-\frac{1}{(n-1)})(\min\sigma^2)\log n, \text{ and}\\ \max_m\beta &= \left(3\log n-(1+\frac{1}{1})\log(1+1)\right)(\max\sigma^2).\\ &= (3\log n-2\log 2)(\max\sigma^2).\end{aligned}$$

In practice the solutions for a range of penalties may be used with CROPS. In Haynes et al. (2017a) the cost function is on the form

$$C(\tau_{j-1}+1,\tau_j)=\sum_{i=\tau_{j-1}+1}^{\tau_j}x_i^2-\sum_{i=\tau_{j-1}+1}^{\tau_j}\frac{x_i^2}{n_i},$$

which gives

$$\begin{aligned}-\sum_{j=1}^{m+1}C(\tau_{j-1}+1,\tau_j) &= -\sum_{j=1}^{m+1}\left(\sum_{i=\tau_{j-1}+1}^{\tau_j}x_i^2-\sum_{i=\tau_{j-1}+1}^{\tau_j}\frac{x_i^2}{n_j}\right)\\ &= -\sum_{i=1}^{n}x_i^2+\sum_{j=1}^{m+1}\hat{\mu}_j\sum_{i=\tau_{j-1}+1}^{\tau_j}x_i=-\sum_{i=1}^{n}x_i^2+\sum_{j=1}^{m+1}n_j\hat{\mu}_j^2,\end{aligned}$$

such that from Equation (3.3)

$$\sum_{j=1}^{m+1} C(\tau_{j-1}+1, \tau_j) + 2\ell(\boldsymbol{\tau}|\boldsymbol{x}) = n\log(2\pi).$$

Hence the range of β may be set as in Equation (5.24). With another choice of cost function the range for β would need to be adjusted such that the cost functions are equivalent by Theorem 5.3 and 5.4.

The method of finding all resulting solutions for a continuous range of penalties may also be used for model selection with mBIC. To do this first we find the solutions for β values such that Equation (5.24) holds using BIC_3. That will yield a number of different solutions that are optimal for different penalties with regard to BIC_3. The solution among these that maximize BIC_4 is then another approximation to the Bayes Factor that may be used for model selection in the case where σ^2 is known. The advantage of this method over maximizing BIC_2 or BIC_4 directly is that $\boldsymbol{\tau}$ is not set as the maximizer of an expression that contains $\sum_{j=1}^{m+1} \log n_j$. It is however unclear whether the solution that maximizes the BIC is guaranteed to be among the solutions yielded from CROPS.

Another setting where we may take advantage of a range of penalties is when Assumption 5.1 only holds approximately. Since Hocking et al. (2013) demonstrated that mBIC did not perform satisfactory on a real data set where the assumptions only hold approximately, the range in Equation (5.24) may only be used as a guide to what values must at least be included in the range in that case.

5.5. Model selection when the variance is unknown

Model selection is more complicated when the variance of the univariate Gaussian distribution also needs to be estimated in each interval. In this section we present various approximations for the Bayes Factor under Assumption 5.2. In Section 5.5.1 we present mBIC and in Section 5.5.2 we present BIC for this assumption. As the expression for mBIC is too complicated to yield itself to maximization with PELT, we also present an ad hoc cost function inspired by mBIC in Section 5.5.2.

5.5.1. mBIC

As well as a modified BIC for independent changepoint data from $\mathcal{N}(\mu_j, 1)$ distributions, Zhang and Siegmund (2007) present a modified criterion when the changepoint data are from $\mathcal{N}(\mu_j, \sigma_j^2)$ distributions. This is also derived as an approximation of the Bayes factor. This time however there are more parameters to integrate over. Let $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_{m+1}^2)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{m+1})$ and $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_m)$ so Equation (2.7) becomes

$$
\begin{aligned}
&Pr(x|M_m) = \int_\Theta Pr(x|\theta, M_m) Pr(\theta|M_m) d\theta, \\
&Pr(x|M_m) = \\
&\int\int\int Pr(x|\theta, M_m) Pr(\boldsymbol{\sigma}^2|M_m) Pr(\boldsymbol{\mu}|M_m, \boldsymbol{\sigma}^2) Pr(\boldsymbol{\tau}|\boldsymbol{\sigma}^2, \boldsymbol{\mu}, M_m) d\boldsymbol{\sigma}^2 d\boldsymbol{\mu} d\boldsymbol{\tau}.
\end{aligned}
$$

Hence the mBIC criterion displayed in the following theorem is more complex than the mBIC criterion in Theorem 3.1 when the σ^2 has a constant known value.

Theorem 5.5. Theorem 2 from Zhang and Siegmund (2007) states that under Assumption 5.2, where M_m is the model with m internal changepoints and M_0 is the model with 0 internal changepoints under priors on the parameters that represent no information

$$
\begin{aligned}
\log \frac{P(\boldsymbol{x}|M_m)}{P(\boldsymbol{x}|M_0)} =& \frac{n-m+1}{2} \log\left(1 + \frac{SS_{bg}(\hat{\boldsymbol{\tau}})}{SS_{wg}(\hat{\boldsymbol{\tau}})}\right) \\
&+ \log \frac{\Gamma\left(\frac{n-m+1}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)} + \frac{m}{2} \log(SS_{all}) \\
&- \frac{1}{2}\left(\sum_{j=1}^{m+1} \log(n_j) + (1-2m)\log n\right) + O_p(1),
\end{aligned}
\tag{5.25}
$$

where SS_{bg} is the term that represented the likelihood in Equation (3.5), namely with $\bar{x} = \sum_{i=1}^n x_i/n$

$$
SS_{bg}(\hat{\boldsymbol{\tau}}) = \frac{1}{2}\sum_{j=1}^{m+1} n_j(\hat{\mu}_j - \bar{x})^2, \qquad \hat{\boldsymbol{\tau}} = \arg\max \frac{SS_{bg}(\boldsymbol{\tau})}{SS_{wg}(\boldsymbol{\tau})},
$$

and

$$SS_{all} = \sum_{j=1}^{m+1}(x_j - \bar{x}), \qquad SS_{wg}(\hat{\boldsymbol{\tau}}) = SS_{all} - SS_{bg}(\hat{\boldsymbol{\tau}}).$$

According to Zhang and Siegmund (2007) the first line on the right hand side of Equation (5.25) represents the likelihood, the middle line is the result of integrating out the nuisance parameter, and the last line represents the penalty. In that sense the penalty is the same as when there was only one parameter, but the terms from the nuisance parameter also regulates how parsimonious the optimal model is. Furthermore the nuisance parameter makes the distribution of μ_j Student-t instead of Gaussian, which is why the likelihood is so different from in Equation (3.7) and $\hat{\sigma}_j^2$ is not a part of the expression.

The likelihood in Equation (5.25) is maximized when $SS_{bg}(\hat{\boldsymbol{\tau}})/SS_{wg}(\hat{\boldsymbol{\tau}})$ is maximized. When we let the data points on an interval constitute a group, then SS_{bg} in Theorem is the between group variability, and SS_{wg} is the variability within each group. An interpretation of the likelihood is thus that $SS_{bg}(\hat{\boldsymbol{\tau}})/SS_{wg}(\hat{\boldsymbol{\tau}})$ represents an odds, and when it is large more of the variability is explained by the difference between the groups, than by the variability within each group. So for a certain m the criterion chooses the $\hat{\mu}_j$ and $\hat{\tau}_j$s such that the variability between groups is maximal, while it does not have to choose a value for $\hat{\sigma}_j^2$.

The first term of Equation (5.25) may be written

$$\begin{aligned}
&\frac{n-m+1}{2}\log\left(1+\frac{SS_{bg}(\hat{\boldsymbol{\tau}})}{SS_{wg}(\hat{\boldsymbol{\tau}})}\right)\\
&=\frac{n-m+1}{2}\log\left(1+\frac{SS_{bg}(\hat{\boldsymbol{\tau}})}{SS_{all}-SS_{bg}(\hat{\boldsymbol{\tau}})}\right)\\
&=\frac{n-m+1}{2}\log\left(\frac{SS_{all}}{SS_{all}-SS_{bg}(\hat{\boldsymbol{\tau}})}\right)\\
&=\frac{n-m+1}{2}\left(\log SS_{all}-\log(SS_{all}-SS_{bg}(\hat{\boldsymbol{\tau}}))\right)\\
&=\frac{n-m+1}{2}\left(\log SS_{all}-\log\left(SS_{all}-\frac{1}{2}\sum_{j=1}^{m+1}n_j(\hat{\mu}_j-\bar{x})^2\right)\right),
\end{aligned}$$

which does not directly give rise to a separate cost function for each interval. But Equation (5.25) is maximized when $(P(\boldsymbol{x}|M_m)/P(\boldsymbol{x}|M_0))^{\frac{2}{n-m+1}}$ is maximized for a given m, and so we may continue with

$$1+\frac{SS_{bg}(\hat{\boldsymbol{\tau}})}{SS_{wg}(\hat{\boldsymbol{\tau}})} = SS_{all}\left(SS_{all}-\frac{1}{2}\sum_{j=1}^{m+1}n_j(\hat{\mu}_j-\bar{x})^2\right)^{-1}.$$

However when we compute with PELT we do not consider m to be known. Proceeding along these lines it may be possible to formulate a criterion that is maximal when mBIC is maximal as well as being on the form of a sum of interval costs and a penalty term, but such a criterion is not guaranteed to work with PELT.

5.5.2. BIC inspired cost functions

When we want to approximate the Bayes Factor in Equation (2.6) one option is to use Equation (2.10) in the same way as in Section 3.3.1. Following Assumption 5.2 the parameters are $\boldsymbol{\tau} = (\mu_1, \ldots, \mu_{m+1}, \sigma_1^2, \ldots, \sigma_{m+1}^2, \tau_1, \ldots, \tau_m)$, which constitutes a total of $3m + 2$ parameters. These may be interpreted as $3m + 2$ degrees of freedom. Then from Equation (5.18) the BIC is

$$-\sum_{j=1}^{m+1} n_j(\log 2\pi + \log \hat{\sigma}_j^2 + 1) - (3m + 2)\log n,$$

with $n_j \geq 2$ for all j. Maximizing twice the likelihood is according to Equation (5.20) equivalent to maximizing

$$-\sum_{j=1}^{m+1} n_j \log \hat{\sigma}_j^2.$$

Furthermore $-2\log n$ is a constant for a given data set and does not affect the maximization, to see this we may note that it is equivalent to $a(x_i) = -(2\log n)/n$ for all x_i in Theorem 5.3. Thus a criterion based on Equation (2.10) when we estimate both μ and σ^2 is

$$\text{BIC}_5 = -\sum_{j=1}^{m+1} n_j \log \hat{\sigma}_j^2 - 3m \log n, \quad n_j \geq 2 \; \forall j \tag{5.26}$$

such that

$$\beta = 3\log n \text{ and } C_2(\tau_{j-1} + 1, \tau_j) = n_j \log \hat{\sigma}_j^2.$$

However the situation in the previous section shows that this approach is overly simplistic. Thus the criterion may be a very bad approximation of the Bayes Factor.

An ad hoc option for model selection is to let the cost be either

$$C_2(\tau_{j-1} + 1, \tau_j) = n_j \log \hat{\sigma}_j^2 \tag{5.27}$$

or

$$C_2(\tau_{j-1} + 1, \tau_j) = n_j \log \hat{\sigma}_j^2 + \log \frac{n_j}{n},$$

and let the penalty be somewhere in a continuous range. Then we may again select the solution among only a few ones where the mBIC is maximal, or we may simply treat the penalty as a tuning parameter.

5.6. Algorithms

In Section 3.5 we mentioned several algorithms for changepoint detection, and detailed BinSeg, OP, and PELT. In this section we generalize OP and PELT so that we may estimate more than one parameter. We can easily adjust BinSeg so that it can estimate more than one parameter per changepoint interval, but based on the algorithms performance on the simulations in Section 4.1 we opt not to do this. Our generalization of OP is presented thoroughly in Section 5.6.1 in order to prepare the reader for the presentation in Section 5.6.3 of our generalized PELT. In Section 5.6.2 a straight forward way to attempt to generalize PELT is also presented because it makes it easier to understand the generalization of PELT.

5.6.1. gOP

The Optimal Partitioning algorithm only needs to be slightly adjusted to accommodate for a restriction on the minimum segment length. OP iteratively decides what is the best previous changepoint given that there is a changepoint at the data point in question. Without a restriction it considers every single previous data point as a possible predecessor. In order to generalize OP we only need the following theorem.

Theorem 5.6. When the minimum segment length is restricted to g, then the best predecessor of a changepoint at t is contained in the set $\{0, g, g+1, \ldots, t-g\}$.

Proof. Since $\kappa_1 - \kappa_0 \geq g$ and $\kappa_0 = 0$ then $\kappa_1 \geq 0$ and $\{1, \ldots, g-1\}$ may not contain any changepoints. When there is a changepoint at t then for some value q we have that $\kappa_q = t$. As $\kappa_q - \kappa_{q-1} \geq g$ then $\kappa_{q-1} \leq t - g$. Thus the best predecessor of a changepoint at t is not contained in the set $\{1, \ldots, g-1, t-g+1, \ldots, t-1\}$, and must be in the set $\{0, g, g+1, \ldots, t-g\}$. □

One way to implement gOP is displayed in Algorithm 3, it may also be found in R (R Core Team, 2017) code in Appendix B.3. At lines 1 through 4 we compute the final total cost for no changepoints in the simple case where there may be no internal changepoints. When the data set is of length 0 we define the total cost to be $-\beta$. However the cost of a data set with for instance $g-1$ data points is not defined, as there is no way to compute the cost $C_g(1, g-1)$ of $g-1$ data points. When the data set is of length g the predecessor must be 0, since it is the only data point for which the total cost is defined, that is $F(g) = F(0) + C(1, g) + \beta = C(1, g)$. Also when the data set is of length t in $\{g+1, \ldots, 2g-1\}$ the only possible predecessor is 0 since the other points are either too close to 0 or t. So then $F(t) = F(0) + C(1, t) + \beta = C(1, t)$.

When t is in $\{2g, \ldots, n\}$ the available predecessors are all the ones that were available at $t-1$, as well as the newly available point $t-g$. This is why the `s.set` is expanded in line 6 in Algorithm 3. Then in lines 8 through 14

the optimal predecessor is selected among the possible predecessors exactly like in OP and PELT, that is Algorithms 1 and 2. In lines 16 through 23 we also construct the vector τ from r in the exact same fashion as OP and PELT. So the differences between OP and gOP is in lines 2 to 4 and in line 6. The following example illustrates gOP applied to a data set.

Example 5.1. Example of gOP employed to a data set in Table 5.3 with $g = 2$, where we fit both mean and variance, as in Equation (5.17), and let the penalty be $\beta = 0$. Tables 5.1 and 5.2 support the text in this example, and the solution that is found is displayed in Figure 5.2. First we execute lines 1 to 4 of Algorithm 3. For $t = 2$ and $t = 3$ the only possible previous changepoint is 0, and the computation of $F(2)$ and $F(3)$ is straightforward, see Table 5.3. Then we move on to line 5 of Algorithm 3. For $t = 4$ either 0 or 2 may be the predecessor. Since $C_2(1, 4) = -1.23$, and $C_2(3, 4) = 0.662$, we get that

$$-1.23 = F(0) + C_2(1, 4) > F(2) + C_2(3, 4) = -5.39, \tag{5.28}$$

such that $r(4) = 2$ and $F(4) = -5.39$. Then in the next step $t = 5$ and the predecessor may be either $0, 2$ or 3. The computation is displayed in Table 5.1. In the next step $t = 6$ and the considered data points are $0, 2, 3$ or 4, so that the computation is as in Table 5.2.

Continuing forward in this manner yields the table of costs and predecessors presented in Table 5.3. When the data set is of length 8 thus the changepoint vector is $\boldsymbol{\tau} = (0, 3, 6, 8)$.

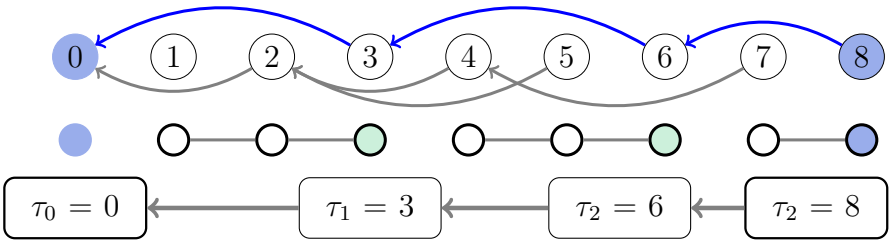


Figure 5.2: These graph represent the solution found by the generalized OP in Example 5.1 and by the generalized PELT in Example 5.3. The top graph is $r(t)$ from Table 5.3, where for instance nodes 4 and 5 point to node 2 because $r(4) = r(5) = 2$. The path marked in dark blue indicates the elected solution.

Algorithm 3: Generalized Optimal Partitioning (gOP).

input : $Y = (x_1, \ldots, x_n)$, n=length(Y), β, $C(\cdot)$, g
output: $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+1})$

/* Set s.set, r, and final total cost up to $2g$ */
s.set $=\{0\}$; $F(0) = -\beta$
for $t \leftarrow g$ **to** $2g-1$ **do**
  r (t)=0; F $(t) = C(1,t)$
end
for $t \leftarrow 2g$ **to** n **do**
  /* Add newly available predecessor to s.set */
  s.set $= \{$s.set$, t-g\}$
  /* For a changepoint at t find best most recent changepoint s */
  $F(t)=\infty$
  for $s \in$ s.set **do**
    $p = F(s)+C(s+1,t)+\beta$
    if $p < F(t)$ **then**
      $F(t)=p$
      r $(t)=s$
    end
  end
end
/* Build vector $\boldsymbol{\tau}$ from r */
changepoint =n
i =1
while changepoint $\neq 0$ **do**
  $\boldsymbol{\tau}$ (i)=changepoint
  changepoint $= r$ (changepoint)
  $i = i+1$
end
$\boldsymbol{\tau}$ = Sort($\boldsymbol{\tau}$)

Table 5.1: OP algorithm $t = 5$ and $\beta = 0$. Since the lowest prospective total cost is -6.23 for $s = 2$ then $r(5) = 2$.

s	F(s)	C(s+1,5)	p(5)
0	0	-2.63	-2.63
2	-6.06	-6.23	-6.23
3	-4.43	-1.29	-5.71

Table 5.2: OP algorithm $t = 6$ and $\beta = 0$. Since the lowest prospective total cost is -6.56 for $s = 3$ then $r(6) = 3$.

s	F(s)	C(s+1,6)	p(6)
0	0	-1.61	-1.61
2	-6.06	0.176	-5.88
3	-4.43	-2.13	-6.56
4	-5.39	-1.10	-6.49

Table 5.3: The values of $F(t)$ and $r(t)$ at line 12 before vector $\boldsymbol{\tau}$ is built in Example 5.1, as well as the data set $x_1, \ldots, x_t$ which is used in Examples 5.2 and 5.3.

t	0	1	2	3	4	5	6	7	8
x_t		0.99	0.55	-0.17	2.19	0.74	2.26	0.02	1.20
$r(t)$			0	0	2	2	3	4	6
$F(t)$	0		-6.06	-4.43	-5.39	-6.23	-6.56	-5.81	-8.67

5.6.2. Straight forward PELT

We run into problems if we try to employ PELT in the straightforward manner when there is a restriction on the minimum segment length. The next example demonstrates this.

Example 5.2. Now we employ an erroneous version of the PELT algorithm where only the same adjustments that the OP algorithm needs are made in order to accommodate for the restriction $g = 2$. The data set is the same as in the last example, and is displayed in Table 5.3. The solution it finds is displayed in Figure 5.3. Furthermore $\beta = 0$, and we fit both mean and variance with the cost function in Equation (5.17). The computation for data points $0, 1, 2, 3$ and 4 is identical to the computation with OP in Example 5.1 because the data points are close to the boundary.

Due to Equation (5.28), and $\beta = 0$ then $F(4) = -5.39$ and the old pruning condition says that only nodes s such that $F(s) + C(s+1, 4) \leq F(4)$ need to be evaluated again at the next iteration of t. So when $t = 5$ we would only get the rows such that $s \in \{2, 3\}$ in Table 5.1. This works out fine since $r(5) = 2$ is one of the considered previous changepoints.

The only $s \in \{2, 3\}$ such that $F(s)+C(s+1, 5) \leq F(5)$ is of course 2, and so in the next iteration we only compute the rows in Table 5.1 such that $s \in \{2, 4\}$. As can be seen from the table in reality $r(6) = 3$, which is not in the set. This erroneous version of PELT would pick 4 as the predecessor of 6. Then at $t = 7$ it would pick 4, and at $t = 8$ it would pick 6 such that the prospective solution it finds is $\boldsymbol{\tau} = (0, 2, 4, 6, 8)$, which has a total cost of -8.60. This total cost is higher than the total cost of the OP solution, which we from Table 5.3 know to be $F(8) = -8.67$.

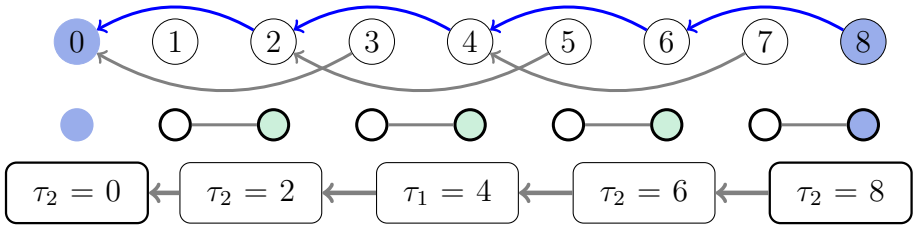


Figure 5.3: These graphs represent the solution found by the erroneous PELT in Example 5.2. The top graph is $r(t)$, where for instance nodes 6 and 7 point to node 4 because $r(7) = r(6) = 4$. The path marked in dark blue indicates the elected solution. These graphs are quite dissimilar to the corresponding graphs for gOP and gPELT applied to the same data set in Figure 5.2

Example 5.2 provides a counterexample that proves that a straightforward PELT application with a restriction of $g > 1$ does not in general provide the optimal OP solution for any $\beta \geq 0$ even when the cost function satisfies Equation (5.2). Further numeric investigation shows that there are

numerous counterexamples for any β, and that factors that contribute to the erroneous approach yielding the wrong changepoint vector is the data set being long, the minimum segment length g being high, and β being low. These simulations are not presented here as we do not consider it relevant to the contents of this master thesis.

We have implemented the erroneous straight forward PELT application outlined in Example 5.2 under the name `pelt2.mycpt` in our R (R Core Team, 2017) package at `https://github.com/kristinbakka/generalizedPELT`. When we employ the PELT algorithm implemented in the changepoint package (Killick and Eckley, 2014) with a restriction that $g > 1$ to a large number of simulated data sets, we get the same output as from the straight forward PELT application outlined in Example 5.2. One of the contributions from this master thesis is thus the development of PELT generalized to account for a restriction of $g > 1$, which is presented in the subsequent section. In theory we may simply use the gOP algorithm to analyze data sets when we need to restrict $g > 1$, but in practice the quadratic run time of OP makes it only possible to analyze relatively short data sets. Especially when we want to generalize to multiple streams we are in need of a correct generalization of the PELT algorithm to account for a restriction on g.

5.6.3. gPELT

In this section the PELT algorithm generalized to accommodate for $g > 1$ is developed. We refer to it as gPELT in this thesis. The cost function requirement is displayed in Equation (5.2). The pruning condition is unchanged. Hence the pruning condition on t_1 and t_2, where $t_1 < t_2$ is

$$F(t_1) + C_g(t_1 + 1, t_2) \geq F(t_2) \ , \tag{5.29}$$

but it must be employed in a slightly different fashion. The following theorem is the mathematical basis for the pruning in generalized PELT.

Theorem 5.7. Whenever (5.2) and (5.29) both hold for $t_1 < t_2 < t_3$, where $t_3 - t_2 \geq g$ and $t_2 - t_1 \geq g$, then data point number t_1 is not the optimal estimate for the predecessor of t_3.

Proof. First we add $C_g(t_2 + 1, t_3)$ on both sides of the pruning condition from Equation (5.29), that is

$$\begin{aligned} F(t_1) + C_g(t_1 + 1, t_2) &\geq F(t_2) \ , \\ F(t_1) + C_g(t_1 + 1, t_2) + C_g(t_2 + 1, t_3) &\geq F(t_2) + C_g(t_2 + 1, t_3) \ . \end{aligned}$$

Then according to Equation (5.2)

$$F(t_1) + C_g(t_1 + 1, t_3) \geq F(t_1) + C_g(t_1 + 1, t_2) + C_g(t_2 + 1, t_3),$$

so that we get

$$F(t_1) + C_g(t_1 + 1, t_3) + \beta \geq F(t_2) + C_g(t_2 + 1, t_3) + \beta.$$

The optimal estimate for the predecessor of t_3 is the one which gives the minimal total cost at t_3. As the total cost with $r(t_3) = t_1$ is not smaller than the total cost with $r(t_3) = t_2$, then t_1 is not the optimal estimate for the predecessor of t_3. □

From Theorem 5.6 we get that a minimum segment length of g means that in the gOP algorithm at loop t, only $0, g, g+1, g+2, \ldots, t-g$ are considered as possible positions of the predecessor. The consequence of the requirement in Theorem 5.7 that $t_3 - t_2 \geq g$ is that at $t = t_2$ the data point numbers t_1 that fulfill Equation (5.29) may not be the optimal predecessor of a changepoint at t_3. That is in the PELT algorithm at $t = t_{3,1}$ the optimal predecessor is either in $\{t_2, t_2 - 1, \ldots, t_2 - g + 1\}$, or it is a data point t_1 such that $F(t_1) + C_g(t_1 + 1, t_2) < F(t_2)$, with $t_2 = t_{3,1} - g$. An easy way to think of this is that at $t = t_{3,1}$ we get an `s.set` that is a combination of the *earned* data points $\{t_2, t_2 - 1, \ldots, t_2 - g + 1\}$ and the data points inherited from the computation at t_2.

There are several ways to implement gPELT, and one straight forward way is displayed in Algorithm 4. Note how closely it relates to gOP in Algorithm 3. The first change is that a vector named `Inherit` is introduced. At line 6 of Algorithm 4 we call `Inherit(`$t-g$`)`. Subsequently we will label $t_2 = t - g$, since this is in accordance with the labeling in Theorem 5.7. `Inherit(`t_2`)` are the t_1s such that $t_1 \leq t_2 - g$ and $F(t_1) + C_g(t_1, t_2) \leq F(t_2)$. That is `Inherit(`$t_2$`)` is a set of data points that may contain the predecessor of $t_3 = t_2 + 1$. Lines 1 to 5 are the same in gOP and gPELT apart from the introduction of `Inherit`.

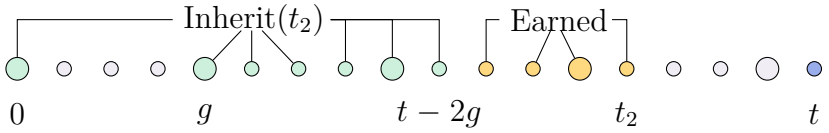


Figure 5.4: Represents a portion of a data set where $g_\kappa = 4$. Big circles indicate minimum distance between changepoints.

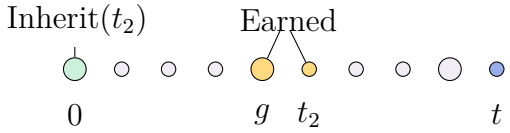


Figure 5.5: Represents a portion of a data set where $g_\kappa = 4$ and $t = 9$. Big circles indicate minimum distance between changepoints.

The next difference between the algorithms is in line 6 of Algorithms 3 and 4. While in gOP the `s.set` at t is the `s.set` at $t - 1$ combined with

the newly available point $t-g$, in gPELT the `s.set` at t is the data points from `Inherit`$(t-g)$ as well as the admissible predecessors that may never be in `Inherit`$(t-g)$. In Figures 5.4 and 5.5 this is illustrated with $g=4$, $t=17$, $t_2=t-g=13$. The data points marked in gray are not admissible as predecessors to t, and the points that may be in `Inherit`(t_2) are marked in green. The rest of the data points are indicated with *Earned*, to signify that they are the admissible predecessors that may never be in `Inherit`(t_2). In Figure 5.4 the earned data points are $t-2g+1,\ldots,t_2$, while in Figure 5.5 they are only g and t_2. In lines 15 through 21 of Algorithm 4 we remove some elements of `s.set`, and save the remaining elements as `Inherit`(t). In the following example we apply gPELT to data.

Example 5.3. In this example we apply gPELT in Algorithm 4 to the data in Table 5.3, which is the same data that we used in Examples 5.1 and 5.2. We arrive at the same solution as is illustrated in Figure 5.2, as we use the same penalty and cost function, that is we use $\beta=0$ and the cost function in Equation (5.17) which estimates both the mean and the variance. In the following figure `Inherit`$(t-g)$ is marked in dark green, and the earned data points are marked in orange. The data points that to our knowledge at $t-g$ may not be the predecessor of t are marked in light grey if they are not admissible, or else in light green. The current t is marked in blue. Those points that will be a part of `Inherit`(t) are marked with a thick border.

First we perform the steps in lines 1 through 5 in Algorithm 4. For t in $2,3$ we get `Inherit`$(t)=\{0\}$ since $g=2$ and $2g-1=3$. When we enter the for-loop at line 6 there are two data points to choose from, and we get as in Example 5.1 that

$$-1.23=F(0)+C_2(1,4)>F(2)+C_2(3,4)=-5.39.$$

The 0th point is thus chosen as predecessor to 2. As $\beta=0$ then $F(4)<F(2)+C_2(3,4)$ and we get that `Inherit`$(4)=\{0\}$. In Figure 5.6 this is signified with a broad outline at $t=4$, and with dark green fill at $t=6$.

In the next iteration we inherit the 0th data point, and earn two new data points, so `s.set` $=\{0,2,3\}$. Thus we get Table 5.1 from Example 5.1, and conclude that $r(5)=2$. Then we find the elements s of `s.set` such that $F(s)+C(s+1,5)\leq F(5)$. Since $\beta=0$ this is only true for $s=5$, but for another β there might have been more such data points. In Figure 5.6 we mark this by a dark green node for $t=7$, and with a broad outline for $t=5$.

In the next step when $t=6$, only the data points indicated by orange and dark green may be the optimal predecessor, so `s.set` $=\{0,3,4\}$. Data point 2 is a predecessor such that it is possible to calculate the total cost, but due to the pruning at $t-g=6-2=4$ we know that it may not be the optimal predecessor. As seen from Table 5.2 the optimal

predecessor is $r(6) = 3$. In Figure 5.6 the computation is done for $t = 7$ and $t = 8$ as well. At $t = 7$ we get that `s.set` $= \{2, 4, 5\}$, and at $t = 8$ we get that `s.set` $= \{3, 5, 6\}$. Theorem 5.7 guarantees that the optimal predecessor is within `s.set`, and this is also what the figure shows.

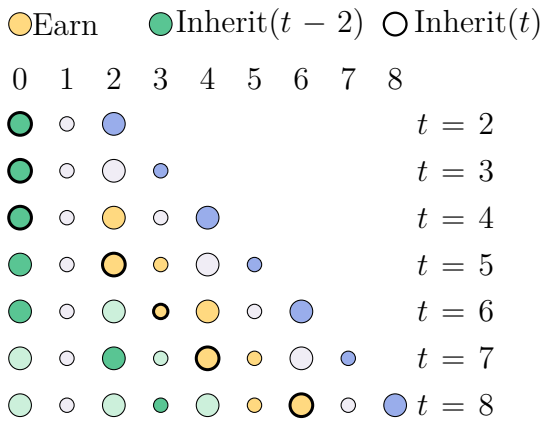


Figure 5.6: Represents a portion of a data set where $g_\kappa = 2$. Big circles indicate minimum distance between changepoints.

Algorithm 4: Generalized Pruned Exact Linear Time (gPELT).

input : $Y = (y_1, \ldots, y_n)$, n=length(Y), β, $C(\cdot)$, g
output: $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+1})$

/* Set s.set, r, final total cost, and Inherit up to $2g$ */
s.set $=\{0\}$; $F(0) = -\beta$
for $t \leftarrow g$ **to** $2g-1$ **do**
  r (t)=0; F $(t) = C(1,t)$; Inherit(t)=0
end
for $t \leftarrow 2g$ **to** n **do**
  /* Combine inherited and earned data points to get s.set */
  s.set $= \{$Inherit$(t\text{-}g), \max(g, t-2g+1) : (t-g))\}$
  /* For a changepoint at t find best most recent changepoint s */
  $F(t)=\infty$
  for $s \in$ s.set **do**
    $p = F(s)+C(s+1,t)+\beta$
    if $p < F(t)$ **then**
      $F(t)=p$
      r $(t)=s$
    end
  end
  /* Remove non-optimal predecessors */
  for $s \in$ s.set **do**
    if $F(s)+C(s+1,t) \geq F(t)$ **then**
      Remove(s.set,s)
    end
  end
  /* Remember which data points to inherit */
  Inherit(t) $\leftarrow$ s.set
end
/* Build vector $\boldsymbol{\tau}$ from r */
changepoint =n
i =1
while changepoint $\neq 0$ **do**
  $\boldsymbol{\tau}$ (i)=changepoint
  changepoint $= r$ (changepoint)
  $i = i+1$
end
$\boldsymbol{\tau}$ = Sort($\boldsymbol{\tau}$)

6. Discussion and conclusion

In Section 5 we have introduced various model selection criteria under Assumptions 5.1 and 5.2, and a generalization of PELT such that it may be used in the setting where we need to maximize more than one parameter. A natural next step is to study the properties of these criteria, and the properties of gPELT when g is large. However this is outside the scope of this thesis. Furthermore under Assumption 5.2 the penalty of BIC_4 in Equation (5.23) is likely to be too low, and extensive simulations are needed to find alternate penalties for which the criterion performs better. CROPS used with gPELT is well suited to aid in this endeavor. Finding the optimal penalty is an open research question. In applications the penalty is often treated as a tuning parameter, found by trial and error until the wanted sensitivity to potential changes appears to be achieved.

It would be interesting to endeavor to write the expression of mBIC in Theorem 5.5 on the form of sum of interval costs and a penalty. It would also be interesting to derive an mBIC for when all the data have the same variance, but that variance is unknown. It is possible that such a criterion would be simple enough to yield itself to maximization with PELT. One of the downsides with PELT and BS is that they only allow for a total penalty linear in m. Killick et al. (2012a) suggests an algorithm based on PELT that allows for a total penalty that is not linear in m. The suggestion is to run PELT on the data set with some β, and then run PELT again with a new β determined from the last solution until the total penalty corresponds to the desired function of m. They further argue that this may be done since PELT is fast under certain conditions. This is similar to the way we suggest to use CROPS to compute the mBIC in Equations (3.12) or (5.26). When we want to use PELT to maximize the approximated Bayes Factor with the help of either BIC or mBIC under Assumptions 3.1 or 5.1, then the penalty is linear and so the restriction that the penalty must be linear is of no consequence.

The runtime of gPELT is likely to be close to the runtime of PELT as the algorithms are so similar, but a simulation study to estimate the empirical run time of PELT is of interest as well. A property of PELT that we have ignored in this thesis is that Killick et al. (2012a) allows the PELT requirement in Equation (3.25) to instead be

$$C(t_1 + 1, t_3) - C(t_1 + 1, t_2) - C(t_2 + 1, t_3) \geq K,$$

which slightly adjusts what cost functions are applicable. As we did not need this in the thesis, we have not mentioned it before. The code of gPELT may of course be adjusted to account for the K, but this is left for further work. One advantage of including a K is that the cost function in Equation (3.26) could then to be made to fulfill the PELT requirement in Equation (3.25),

since when

$$\sum_{i=t_1+1}^{t_3}(x_i-\hat{\mu}_j)-\sum_{i=t_1+1}^{t_2}(x_i-\hat{\mu}_j)-\sum_{i=t_2+1}^{t_3}(x_i-\hat{\mu}_j)$$
$$+\log\frac{t_1+1,t_3}{n}-\log\frac{t_1+1,t_2}{n}-\log\frac{t_1+1,t_2}{n}\geq 0$$

then

$$\sum_{i=t_1+1}^{t_3}(x_i-\hat{\mu}_j)-\sum_{i=t_1+1}^{t_2}(x_i-\hat{\mu}_j)-\sum_{i=t_2+1}^{t_3}(x_i-\hat{\mu}_j)$$
$$+\log(t_1+1,t_3)-\log(t_1+1,t_2)-\log(t_1+1,t_2)\geq-\log n.$$

6.1. Alternate model selection criteria

An other model selection criterion that may be used with PELT in this setting is *Minimum Description Length* (MDL) (Wu and Hsieh, 2006), or *Information Complexity* (ICOMP) (Bozdogan and Haughton, 1998). It would be interesting to compare the performances of MDL, ICOMP and of our BIC_3 criterion with PELT. MDL and ICOMP may also be used in other settings, and so may PELT. If we have other assumptions on the data, for instance that they are independent draws from Gamma distributions we would substitute the likelihood of the normal distributions in the cost function of PELT with the likelihood of the distribution in question. The penalty will likely also need to be altered, but as long as it is linear and the cost function fulfills the requirement in Equation (5.2) we may analyze these data with CROPS and gPELT. Assuming another model is thus equivalent to changing the model selection criterion.

The algorithms always evaluate the cost of all data points in an interval simultaneously. So it is not necessary to require that the data points in a changepoint interval are independent. We might fit any distribution on each interval, for instance an AR(3) time series process. In the Big Insight sensor project our initial idea was to see whether PELT could be used to identify anomalies that materialize as changepoints in the residuals, see Figure 1.1. One challenge is that Tveten (2017) found that the residuals were not i.i.d. normal, but were dependent on the previous residuals. Tveten (2017) fitted an AR(3) model to the residuals, and the residuals from the fit were approximately normal. It is also an option to fit a non-parametric distribution, as was done in Haynes et al. (2017b). In order to take the time between observations into account, it may be included as a value of each data point.

When each observation is a number on $\mathbb{R}^2$ or $\mathbb{R}^3$ or some higher dimension we may for instance assume that the data points are multivariate realizations from i.i.d. multivariate normal distributions. The appropriate cost functions in this setting are developed in Appendix A. However as the model selection criterion based on BIC is so different from the optimal model

selection criterion in Theorem 5.5, there is no reason to believe that a naive BIC will perform well when the data are from multivariate normal distributions. Thus we are reduced to treating the penalty as a tuning parameter. It would be interesting to investigate which penalty values would be optimal for simulated data, which we again may perform with PELT and CROPS. In our package which is available at `https://github.com/kristinbakka/generalizedPELT` the algorithms are implemented in such a way that they may be used with multivariate data. It would be natural to compare this performance to the performance of Multi Stream Continuous Hidden Markov Models as applied in Missaoui et al. (2013).

6.2. Conclusion

In this thesis we have considered the changepoint detection problem when the data on the jth changepoint interval are i.i.d. $\mathcal{N}(\mu_j, 1)$, and then we have considered the changepoint problem when jth changepoint interval are i.i.d. $\mathcal{N}(\mu_j, \sigma^2)$ for some known σ^2. For these types of data we have made a slight adjustment of the mBIC criterion that allows it to be maximized or approximately maximized with methods like BinSeg and PELT. We have also compared the BIC as it is commonly applied to these types of data to this slightly altered mBIC criterion. In real data we might not know the exact value of σ^2, and so we have proposed upper and lower limits to the range of penalties to investigate with CROPS that are based on our analysis of the mBIC criterion. Additionally we have proposed a method to compute the exact mBIC criterion for these types of data with a combination of CROPS and PELT. We have also found that PELT is not readily applicable in an online setting, which is of interest in the Big Insight sensor project.

In this thesis we have also considered the changepoint problem when jth changepoint interval are i.i.d. $\mathcal{N}(\mu_j, \sigma_j^2)$. Our most significant contribution in this case is that we have developed a generalized version of PELT that may be applied when there is a restriction on the minimum length of an interval, for instance when more than one parameters are estimated per changepoint interval. This allows methods like CROPS which apply other methods for changepoint detection repeatedly more options on what problems they may tackle. Furthermore we have made systems for visualizing the changepoint solution and how PELT and our generalized PELT work, illustrated in Figures 3.7 and 5.6. Additionally we have shown that a certain type of likelihood based cost is always applicable to PELT and the generalized PELT, and proven some ways in which such cost functions may and may not be simplified. As an extension we have developed cost functions to be used when the data are multivariate normal.

Bibliography

Aminikhanghahi, S., Cook, D. J., 2017. A survey of methods for time series change point detection. Knowledge Information Systems 51 (2), 339–367.

Anderson, T., Olkin, I., 1985. Maximum-likelihood estimation of the parameters of a multivariate normal distribution. Elsevier 70, 147–171.

Auger, I., Lawrence, C., 1989. Algorithms for the optimal identification of segment neighborhoods. Bulletin of Mathematical Biology 51 (1), 39–54.

Bozdogan, H., Haughton, D. M., 1998. Informational complexity criteria for regression models. Computational Statistics Data Analysis 28 (1), 51 – 76.
URL http://www.sciencedirect.com/science/article/pii/S0167947398000255

Brandsæter, A., Manno, G., Vanem, E., Glad, I. K., June 2016. An application of sensor-based anomaly detection in the maritime industry. In: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM). pp. 1–8.

Casella, G., Berger, R., 2002. Statistical Inference, 2e. Duxbury/Thomson Learning, Pacific Grove, California.

Efron, B., Hastie, T., 2016. Computer Age Statistical Inference Algorithms, Evidence, and Data Science. Cambridge university press, New York, NY.

Haynes, K., Eckley, I. A., Fearnhead, P., 2017a. Computationally efficient changepoint detection for a range of penalties. Journal of Computational and Graphical Statistics 26 (1), 134–143.

Haynes, K., Fearnhead, P., Eckley, I. A., Sep 2017b. A computationally efficient nonparametric approach for changepoint detection. Statistics and Computing 27 (5).

Hocking, T., Rigaill, G., Vert, J.-P., Bach, F., 17–19 Jun 2013. Learning sparse penalties for change-point detection using max margin interval regression. In: Dasgupta, S., McAllester, D. (Eds.), Proceedings of the 30th International Conference on Machine Learning. Vol. 28 of Proceedings of Machine Learning Research. PMLR, Atlanta, Georgia, USA, pp. 172–180.
URL http://proceedings.mlr.press/v28/hocking13.html

ISO/IEC, 2011. ISO International standard ISO/IEC 9899:2011(E) - Programming Language. International Organization for Standardization (ISO), Geneva, Switzerland.
URL https://www.iso.org/standard/57853.html

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Snagtrakulcharoen, P., Tan, L., Tsai, T. T., 2005. An algorithm for optimal partitioning of data on an interval. IEEE Signal Processing Letters 12, 105–108.

Kass, R. E., Raftery, A. E., 1995. Bayes factors. Journal of the American Statistical Association 90 (430), 773 – 795.
URL http://www.jstor.org/stable/2291091

Killick, R., Eckley, I. A., 2014. changepoint: An R package for changepoint analysis. Journal of Statistical Software 58 (3).

Killick, R., Fearnhead, P., Eckley, I. A., 2012a. Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association 107 (500), 1590–1598.

Killick, R., Fearnhead, P., Eckley, I. A., 2012b. Supporting material: Optimal detection of changepoints with a linear computational cost.

Knuth, D. E., 1976. Big omicron and big omega and big theta. ACM SIGACT News 8 (2), 18–23.
URL http://www.phil.uu.nl/datastructuren/10-11/knuth_big_omicron.pdf

Lehmann, E. L., 1999. Elements of Large-Sample Theory. Springer New York, New York, NY.
URL https://doi.org/10.1007/0-387-22729-6_2

Missaoui, O., Frigui, H., Gader, P., 2013. Multi-stream continuous hidden markov models with application to landmine detection. EURASIP Journal on Advances in Signal Processing 2013 (1), 40, blahblah.

Olshen, A. B., Venkatraman, E. S., Lucito, R., M.Wigler, 2004. Circular binary segmentation for the analysis of array-based dna copy number data. Biostatistics 5, 557–572.

Price, D. O., 1948. Sequential analysis. by abraham wald. new york: John wiley and sons, inc., 1947. 212 pp. $4.00. Social Forces 27 (2), 170–171.

R Core Team, 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL https://www.R-project.org

R. Maidstone, T. Hocking, G. R. P. F., 2014. On optimal multiple changepoint algorithms for large data. Statistics and Computing.

Rigaill, G., 2010. Pruned dynamic programming for optimal multiple changepoint detection. arXiv preprint arXiv:1004.0887.

Schwarz, G., 1978. Estimating the dimension of a model. The Annals of Statistics 6 (2), 461–464.
URL https://doi.org/10.1214/aos/1176344136

Truong, C., Oudre, L., Vayatis, N., 2018. A review of change point detection methods. CoRR abs/1801.00718.
URL http://arxiv.org/abs/1801.00718

Tveten, M., 2017. Multi-stream sequential change detection using sparsity and dimension reduction. Master's thesis, University of Oslo.

Wilks, S. S., 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. Ann. Math. Statist. 9 (1), 60–62.

Wu, C.-H., Hsieh, C.-H., 2006. Multiple change-point audio segmentation and classification using an mdl-based gaussian model. IEEE 14 (2), 647 – 657.

Yao, Y.-C., 1988. Estimating the number of change-points via schwarz' criterion. Statistics Probability Letters 6 (3), 181 – 189.
URL http://www.sciencedirect.com/science/article/pii/0167715288901186

Zhang, N., Siegmund, D., 2007. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. Biometrics 63 (1), 22–32.

A. Likelihood and cost functions for multivariate Gaussian data

The probability density function of an observation x from a multivariate normal distribution is

$$f(x) = (2\pi)^{-\frac{u}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp -\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu),$$

where x and μ are column vectors of length u, and Σ is a positive definite matrix of rank u. In this section the changepoint model is as in Sections 3.1 and 5.1, except that f_j is a multinormal distribution, which we will denote as $f_j = \mathcal{N}(\mu_j, \Sigma_j)$. In the first section $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_{m+1} = \Sigma$, and the value of Σ is considered known. In the next section we will assume that only the diagonal elements of Σ_j are nonzero, and after that we will look at what happens when all the elements of Σ_j are considered unknown. As the observations in disjoint intervals are independent we will proceed to develop the cost function for an interval containing observations $x_s, \ldots, x_t$, like we did in Section 5.3. In most of this appendix we will express the equations by the centered observations $z_i = x_i - \hat{\mu}$, where $\hat{\mu}$ is found from the interval in question. Furthermore we will denote the kth element of an observation as $x_{i,k}$, so that a single observation is $x_i = (x_{i,1}, \ldots, x_{i,u})$. We will also index z_i and μ accordingly.

The log likelihood of the observation x_i is

$$\ell(i, \mu, \Sigma) = -\frac{1}{2}(u \log 2\pi + \log\det(\Sigma) + (x_i - \mu)^T \Sigma^{-1}(x_i - \mu)),$$

and for independent and identically distributed observations $x_s, \ldots, x_t$ we get

$$2\ell(s, t, \mu, \Sigma) = -\sum_{i=s}^{t}(u \log(2\pi) + \log\det(\Sigma)) - \sum_{i=s}^{t}(x_i - \mu)^T \Sigma^{-1}(x_i - \mu).$$

We will now use the notation that the trace of a matrix A with diagonal elements $a_{11}, a_{22}, \ldots, a_{NN}$ is

$$\operatorname{tr}(A) = \sum_{i=1}^{N} a_{ii},$$

and that for a constant c and conformable matrices A, B and C we have that

$$\operatorname{tr}((A+B)c) = (\operatorname{tr}(A) + \operatorname{tr}(B))c$$

and that $\operatorname{tr}(ABC) = \operatorname{tr}(BCA)$. Since $(x_i - \mu)^T \Sigma^{-1}(x_i - \mu)$ is a scalar

$$\begin{aligned}
&\sum_{i=s}^{t}(x_i - \mu)^T \Sigma^{-1}(x_i - \mu) = \sum_{i=s}^{t} \operatorname{tr}\left((x_i - \mu)^T \Sigma^{-1}(x_i - \mu)\right) \\
&= \sum_{i=s}^{t} \operatorname{tr}\left(\Sigma^{-1}(x_i - \mu)(x_i - \mu)^T\right) = \operatorname{tr}\left(\Sigma^{-1} \sum_{i=s}^{t}(x_i - \mu)(x_i - \mu)^T\right).
\end{aligned}$$

With $\bar{x} = \frac{1}{t-s+1}\sum_{i=s}^{t} x_i$ we have that

$$\sum_{i=s}^{t}(x_i - \bar{x})(\bar{x} - \mu)^T = (\sum_{i=s}^{t} x_i - \sum_{i=s}^{t} \bar{x})(\bar{x} - \mu)^T = 0$$

since $\sum_{i=s}^{t} x_i = \sum_{i=s}^{t} \bar{x} = (t-s+1)\bar{x}$. As $(x_i - \mu) = (x_i - \bar{x} + \bar{x} - \mu)$ we get that

$$\sum_{i=s}^{t}(x_i - \mu)(x_i - \mu)^T = \sum_{i=s}^{t}(x_i - \bar{x})(x_i - \bar{x})^T + \sum_{i=s}^{t}(\bar{x} - \mu)(\bar{x} - \mu)^T.$$

To find the maximum likelihood estimator of μ, we use that $\ell(s,t,\mu,\Sigma)$ is maximal when $\operatorname{tr}\left(\Sigma^{-1}\sum_{i=s}^{t}(x_i - \mu)(x_i - \mu)^T\right)$ is minimal, which is when $\hat{\mu} = \bar{x}$, that is

$$\hat{\mu} = \frac{1}{t-s+1}\sum_{i=s}^{t} x_i.$$

The likelihood at the maximum likelihood estimate for μ may thus be written

$$-2\ell(s,t,\hat{\mu},\Sigma) = \sum_{i=s}^{t}(u\log(2\pi) + \log\det(\Sigma)) + \operatorname{tr}\left(\Sigma^{-1}\sum_{i=s}^{t}(x_i - \hat{\mu})(x_i - \hat{\mu})^T\right). \tag{A.1}$$

This holds true regardless of the constraints on Σ. In the following we will consider the likelihood of observations in the cases where Σ is known, Σ is diagonal, and when Σ is any positive definite matrix. After that we will consider what happens when Σ is a positive semi-definite matrix with rank k. Our objective for developing these likelihoods is to construct reasonable cost functions. To simplify the notation we will in place of x_i use the centered observations $z_i = x_i - \hat{\mu}$. In addition we will use the notation that z_{ij}, x_{ij}, $\bar{x}_j$ and μ_j denote the jth element of respectively z_i, x_i, $\bar{x}$ and μ.

A.1. Known covariance matrix

When Σ is known then minus twice the log likelihood of the centered observations $z_s, \ldots, z_t$ is

$$-2\ell(\hat{\mu}) = \sum_{i=s}^{t} a(x_i) + \operatorname{tr}\left(\Sigma^{-1}\sum_{i=s}^{t} z_i z_i^T\right),$$

with $a(x_i) = (u\log(2\pi) + \log\det(\Sigma))$ such that by Theorem 5.2 a cost function that satisfies the cost function requirement in Equation (5.2) for a known Σ is

$$C(s,t) = \operatorname{tr}\left(\Sigma^{-1}\sum_{i=s}^{t} z_i z_i^T\right),$$

or

$$C(s,t) = \sum_{i=s}^{t} z_i^T \Sigma^{-1} z_i.$$

In the special case when $\Sigma = I_p$ then this cost function reduces to

$$C(s,t) = \sum_{i=s}^{t} z_i^T z_i.$$

A.2. Diagonal covariance matrix

Assume that the centered observations $z_s, \ldots, z_t$ are realizations from a multivariate normal distribution with mean zero and a diagonal covariance matrix Σ with diagonal elements $\sigma_1^2, \ldots, \sigma_u^2$. Then $z_{i,1}, \ldots, z_{i,u}$ are realizations from independent normal distributions with variances $\sigma_1^2, \ldots, \sigma_u^2$. Section 5.3.4 contains the special case when $u = 1$. Since the determinant of a matrix is the product of its eigenvalues

$$\det(\Sigma) = \sigma_1^2 \cdots \sigma_p^2, \qquad \log\left(\det(\Sigma)\right) = \sum_{i=1}^{u} \log \sigma_i^2.$$

As z_i is a column vector, $z_i z_i^T$ is a j by j matrix where element number (j,j) is $z_{i,j}^2$. Thus the diagonal elements of $\Sigma^{-1} \sum_{i=s}^{t} z_i z_i^T$ are $\sum_{i=s}^{t} z_{i,j}^2/\sigma_j^2$, and the trace is

$$\operatorname{tr}\left(\Sigma^{-1} \sum_{i=s}^{t} z_i z_i^T\right) = \sum_{j=1}^{u} \frac{1}{\sigma_j^2} \sum_{i=s}^{t} z_{i,j}^2.$$

When we combine this with Equation (A.1) we get that the likelihood at $\hat{\mu}$ may be written as

$$2\ell(\hat{\mu}, \Sigma) = -(t-s+1)\left(u \log(2\pi) + \sum_{j=1}^{u} \log \sigma_j^2\right) - \sum_{j=1}^{u} \frac{1}{\sigma_j^2} \sum_{i=s}^{t} z_{i,j}^2,$$

where then $\mathrm{E} z_i = 0$. The maximum likelihood estimate for σ_j^2 is thus found from

$$\frac{\partial}{\partial \sigma_j^2} 2\ell(\hat{\mu}, \Sigma) = \frac{-(t-s+1)}{\sigma_j^2} + \frac{1}{(\sigma_j^2)^2} \sum_{i=s}^{t} z_{i,j}^2 = 0,$$

which is identical to the maximum likelihood estimate for the univariate variance of the jth element of the z_is, namely

$$\hat{\sigma}_j^2 = \frac{1}{(t-s+1)} \sum_{i=s}^{t} z_{i,j}^2.$$

We have that

$$\frac{1}{\hat{\sigma}_j^2} \sum_{i=s}^{t} z_{i,j}^2 = (t-s+1)$$

such that

$$-2\ell(\hat{\mu}, \Sigma) = \sum_{i=s}^{t} a(x_i) + (t-s+1) \sum_{j=1}^{u} \log \hat{\sigma}_j^2,$$

with $a(x_i) = u\log(2\pi) + 1$. Thus with the notation from Equation (5.1) a cost function that satisfies the cost function requirement is

$$C_2(s,t) = (t-s+1)\sum_{j=1}^{u}\log\hat{\sigma}_j^2, \tag{A.2}$$

in the multivariate normal case where the nonzero elements of Σ are on the diagonal. This is equivalent to use 5.20 to compute the costs of $z_{sj},\ldots,z_{tj}$ for each j, and then sum those costs. Thus restricting the covariance matrix to be diagonal is equivalent to finding changepoints in u data sets with the one dimensional approach under the restriction that the changes must happen simultaneously. Note that the sizes of the different σ_j^2s make the total costs of each of these one dimensional solutions different. But it is only the difference between the possible total costs that determine which solution is optimal, and so a stream with consistently high values for σ_j^2 and a stream with consistently low values for σ_j^2 will influence what is the optimal solutions equally. Rather it is a stream with a large difference between the values in different segments that will influence what is the optimal solution the most.

A.3. Unknown covariance matrix

The likelihood for observations $x_s,\ldots,x_t$ for some Σ at $\hat{\mu}$ is

$$2\ell(\Sigma) = -\sum_{i=s}^{t}\left(u\log(2\pi) + \log\det(\Sigma)\right) - \operatorname{tr}\left(\Sigma^{-1}\sum_{i=s}^{t} z_i z_i^T\right), \tag{A.3}$$

when Σ is positive definite. Labeling the observed quantity

$$V = \frac{1}{t-s+1}\sum_{i=s}^{t} z_i z_i^T, \tag{A.4}$$

our goal is now to prove that $\hat{\Sigma} = V$. This formula is the same as in Equation 5.11, except that this time $z_i z_i^T$ is a u by u matrix. First we need to derive a simpler expression which is maximal at the maximum likelihood estimate of Σ. This is done in the following theorem.

Theorem A.1. The likelihood in Equation A.3 is maximal when

$$h(\Psi, V) = \log\det(\Psi) - \operatorname{tr}(\Psi V)$$

is maximal, where $\Psi = \Sigma^{-1}$ and we label the function h for further use in this section.

Proof. The likelihood is maximal when $-\log\det(\Sigma) - \operatorname{tr}(\Sigma^{-1}V)$ is maximal. Since Σ is positive definite, then $\Psi = \Sigma^{-1}$ is a one-to-one transformation of Σ. As such the likelihood is also maximal when

$$h(\Psi, V) = \log\det(\Psi) - \operatorname{tr}(\Psi V)$$

is maximal. □

The next step is to derive a canonical form of h.

Theorem A.2. Given a square nonsingular matrix C such that $V = CC^T$, $\tilde{\Psi} = C^T \Psi C$ and $\Sigma = C\tilde{\Psi}^{-1}C^T$, then the maximum likelihood estimate is

$$\hat{\Sigma} = C(\arg\max_{\tilde{\Psi}} h(\tilde{\Psi}, I_p))^{-1}C^T,$$

where the likelihood is as in Equation (A.3) and V is from Equation (A.4). Furthermore

$$\hat{\Sigma} = V \iff \arg\max_{\tilde{\Psi}} h(\tilde{\Psi}, I_p) = I_p.$$

Proof. We get that

$$\operatorname{tr}(\Psi V) = \operatorname{tr}(\Psi CC^T) = \operatorname{tr}(C^T \Psi C) = \operatorname{tr}(\tilde{\Psi}),$$

and that

$$-\log\det(V) + \log\det(\tilde{\Psi}),$$

because

$$\log\det((C)^{-1}) + \log\det((C^T)^{-1}) = -\log\det(CC^T) = -\log\det(V)$$

in

$$\begin{aligned}\log\det(\Psi) &= \log\det((C)^{-1}) + \log\det(C^T\Psi C) + \log\det((C^T)^{-1}) \\ &= -\log\det(V) + \log\det(\tilde{\Psi}).\end{aligned}$$

This allows us to write h in the canonical form, namely

$$\begin{aligned}h(\Psi, V) &= \log\det(\Psi) - \operatorname{tr}(\Psi V) \\ &= -\log\det(V) + \log\det(\tilde{\Psi}) - \operatorname{tr}\left(\tilde{\Psi}\right) \\ &= -\log\det(V) + h(\tilde{\Psi}, I_p)\end{aligned}$$

So according to Theorem A.1 the likelihood is maximal when $h(\tilde{\Psi}, I_p) = \log\det(\tilde{\Psi}) - \operatorname{tr}\left(\tilde{\Psi}\right)$ is maximal. If the maximum is attained for $\tilde{\Psi} = I_p$, then $\hat{\Sigma} = CI_pC^T = V$, so that the proposition holds. □

Now we are ready to prove that $\hat{\Sigma} = V$ by induction.

Theorem A.3. The maximum likelihood estimate for Σ when the likelihood is given by Equation (A.3) is

$$\hat{\Sigma} = \frac{1}{t-s+1}\sum_{i=s}^{t} z_i z_i^T,$$

that is $\hat{\Sigma} = V$ from Equation (A.4).

Proof. When $u = 1$ we get

$$\frac{d}{d\tilde{\Psi}} h(\tilde{\Psi}, I) = 0 = \tilde{\Psi}^{-1} - 1$$

so that $\tilde{\Psi} = 1$ and $\hat{\Sigma} = V$ as proposed. For $u > 1$ we partition

$$\underset{u\times u}{\tilde{\Psi}} = \begin{bmatrix} \underset{(u-1)\times(u-1)}{\tilde{\Psi}_{11}} & \underset{(u-1)\times 1}{\tilde{\Psi}_{12}} \\ \underset{1\times(u-1)}{\tilde{\Psi}_{21}} & \underset{1\times 1}{\tilde{\Psi}_{22}} \end{bmatrix},$$

and assume that $h(\tilde{\Psi}_{11}, I_{u-1})$ is maximized at $\tilde{\Psi}_{11} = I_{u-1}$. Since $\tilde{\Psi}$ is positive definite, also $\tilde{\Psi}_{11}$ and the Schur complement $\tilde{\Psi}_{22} - \tilde{\Psi}_{21}\tilde{\Psi}_{11}\tilde{\Psi}_{12}$ are positive definite. Since

$$\det \tilde{\Psi} = (\det \tilde{\Psi}_{11})(\tilde{\Psi}_{22} - \tilde{\Psi}_{21}\tilde{\Psi}_{11}\tilde{\Psi}_{12})$$

we have that

$$\log\det \tilde{\Psi} = \log\det \tilde{\Psi}_{11} + \log(\tilde{\Psi}_{22} - \tilde{\Psi}_{12}\tilde{\Psi}_{22}\tilde{\Psi}_{21}).$$

When we also take into account that $\operatorname{tr} \tilde{\Psi} = \operatorname{tr} \tilde{\Psi}_{11} + \tilde{\Psi}_{22}$ we get that

$$h(\tilde{\Psi}, I) = (\log\det \tilde{\Psi}_{11} - \operatorname{tr} \tilde{\Psi}_{11}) + \log(\tilde{\Psi}_{22} - \tilde{\Psi}_{12}\tilde{\Psi}_{22}\tilde{\Psi}_{21}) - \tilde{\Psi}_{22}.$$

This in turn we want to maximize with the restriction that $\tilde{\Psi}$ is positive definite. First we notice that for fixed values of $\tilde{\Psi}_{11}$ and $\tilde{\Psi}_{22}$ the maximum is at $\tilde{\Psi}_{12} = \tilde{\Psi}_{21} = 0$ and that $(\log\det \tilde{\Psi}_{11} - \operatorname{tr} \tilde{\Psi}_{11}) = h(\tilde{\Psi}_{11}, I_{u-1})$. Then

$$\max_{\tilde{\Psi}} h(\tilde{\Psi}, I) = \max_{\tilde{\Psi}_{11}} h(\tilde{\Psi}_{11}, I_{u-1}) + \max_{\tilde{\Psi}_{22}} (\log \tilde{\Psi}_{22} - \tilde{\Psi}_{22}),$$

which is attained at $\tilde{\Psi} = I_p$ because $\max_{\tilde{\Psi}_{11}} h(\tilde{\Psi}_{11}, I_{u-1}) = h(I_{u-1}, I_{u-1})$ and $\max_{\tilde{\Psi}_{22}} (\log \tilde{\Psi}_{22} - \tilde{\Psi}_{22}) = \log(1) - 1$.

Since $\arg\max_{\tilde{\Psi}} h(\tilde{\Psi}, I_p) = I_p$ is true whenever it is true for $u = 1$, and it is true for $u = 1$, then for any $u \geq 1$ we have that $\arg\max_{\tilde{\Psi}} h(\tilde{\Psi}, I_p) = I_p$. According to Theorem A.2 thus the maximum likelihood estimate of Σ is $\hat{\Sigma} = V$Since $\hat{\Sigma} = V$ for $u = 1$, and whenever $\hat{\Sigma} = V$ for $u - 1$ then also $\hat{\Sigma} = V$ for u, the proposition is proven by induction. For multiple ways to derive this theorem see Anderson and Olkin (1985). □

Now that we know the maximum likelihood estimates for both μ and Σ we are ready to plug them into the likelihood to get the maximum likelihood and thus a cost function. But we still need to know what restriction on the minimum segmentation length is necessary for the cost function, that is how many observations are needed for both the mean and the covariance matrix

to be of full rank. The rank of $\hat{\Sigma}$ is the dimension of its column space. We will call the column spaces of $\hat{\Sigma}$ and S_i respectively $K_{\hat{\Sigma}}$ and K_i, such that

$$\dim(K_{\hat{\Sigma}}) = \sum_{i=s}^{t} \dim(K_i) - \dim(K_s \cap \cdots \cap K_t).$$

Since $0 \leq \dim K_i \leq 1$ and $1 \leq \dim(K_s \cap \cdots \cap K_t)$ the maximal value of this expression is

$$\max \dim(K_{\hat{\Sigma}}) = \sum_{i=s}^{t} 1 - 1.$$

we need at least $u + 1$ observations x_i for $\hat{\Sigma}$ to be of rank u.

When $\hat{\Sigma}$ is of full rank

$$\hat{\Sigma}^{-1} \sum_{i=s}^{t} z_i z_i^T = (t - s + 1)\hat{\Sigma}^{-1}\hat{\Sigma} = (t - s + 1)I_u.$$

As $\operatorname{tr}((t-s+1)I_p) = (t-s+1)u = \sum_{i=s}^{t} u$ we may use $a(x_i) = u(\log(2\pi)+1)$ to write

$$-2\ell(\hat{\mu}, \hat{\Sigma}) = \sum_{i=s}^{t} a(x_i) + (t - s + 1) \log\det \hat{\Sigma},$$

such that a cost function that satisfies the cost function requirement is

$$C_{u+1}(s, t) = (t - s + 1) \log\det(\hat{\Sigma}). \tag{A.5}$$

If $1 < \dim(K_s \cap \cdots \cap K_t)$ then the estimate $\hat{\Sigma}$ will not be of full rank. When the estimate $\hat{\Sigma}$ is not of full rank the determinant is zero, and the cost would become negative infinity. This satisfies the cost function requirement, but does not yield a useful result. In some real data sets we may make sure that this never happens by choosing a g that is suitable to that data set and has $g \geq u + 1$. A way to get a finite cost function that is defined when the rank of $\hat{\Sigma}$ is k such that $k < u + 1$ is to substitute the determinant with the generalized determinant, and the inverse with the generalized inverse in the probability density function in Equation (A.3) and find an appropriate cost from this. This is equivalent to the likelihood of a singular multivariate normal distribution. Note that such a cost function does not fulfill the cost function requirement if it is used in conjunction with some other cost function, only if all the data is from the singular normal distribution, such that all the intervals are evaluated with the same cost function.

Another extension is to look at when only u_0 of u streams change. Then for each changepoint interval we may select which streams change as those that give the smallest interval cost, and only treat the streams that change as observations when we compute the interval costs.

B. R-code

This appendix contains the R (R Core Team, 2017) code that will aid the reader the most in understanding the concepts described in this thesis. In this appendix the first section presents the parts of the package `changepoint` that we make use of in Section 4. The other sections present the most important parts of our implementations of gOP and gPELT. This R code is also available at https://github.com/kristinbakka/generalizedPELT as part of an R package, along with the files `test-univariate.R`, `test-multivariate.R` and `plot-univariate.R`. These three files illustrate how to employ the different algorithms, and the difference in their perfrmance. The objective of the implementations of gOP and gPELT is to make the algorithms as easy to understand as possible.

B.1. Make use of the package changepoint

Although we have implemented PELT ourselves to properly understand it, we use the implementation of BinSeg and PELT in the changepoint package (Killick and Eckley, 2014) in R in the simulations whenever these give the same solutions as our own implementation. That is because the computationally heavy parts of the algorithms are implemented efficiently in C ISO/IEC (2011) in Killick and Eckley (2014), such that it runs in less time than our implementation. The function we use is `cpt.mean()`, which finds changes in the mean of a normal distribution. Although it does not say so explicitly in the documentation for the PELT package, we have found from testing `cpt.mean()` against our own implementation of PELT that it assumes the data points are i.i.d. $\mathcal{N}(\mu, 1)$, that is with variance equal to 1. Below is code that simulates one data set, and analyzes it with BinSeg and PELT.

```
library(changepoint)
## Create a data set
Delta <- 0.5
data.set <-  matrix(sapply(1:5,function(i)
t(rnorm(20,Delta*i,1))),nrow=1)

## BinSeg
# BIC penalty (2*log(n))
BSbic <- cpt.mean(data=data.set,penalty="BIC",
   Q=20, method="BinSeg", class=TRUE)
# Manual penalty (2*log(n))
BSmanual <- cpt.mean(data=data.set,
   penalty="Manual",pen.value="2*log(n)",method="BinSeg",Q=20)

## PELT
```

```
# BIC penalty (2*log(n))
PELT<- cpt.mean(data=data.set,
   penalty="BIC", Q=20, method="BinSeg")

## List of objects of S4 class 'cpt'
print(BSmanual[[1]])
PELT[[1]]@pen.value

## Display data
plot(BSbic[[1]])
plot(PELT[[1]])
```

When the `penalty="BIC"`, the β for BIC_1 from Equation (3.21) is used. The syntax to set the penalty manually is displayed where the variable `BSmanual` is initialized. The function `cpt.mean()` returns a list of evaluated series if the data is a matrix of time series. If `class` is set to `TRUE`, each element in the list returned becomes an object of type `cpt` that may be plotted. When the method is `BinSeg` a parameter `Q` must be set, which restricts the maximal number of changepoints the algorithm may identify. A warning to reapply BS with a higher value for `Q` is returned if the algorithm identifies `Q` changepoints. Two other functions in the package are `cpt.var()` and `cpt.meanvar()`. They respectively find changes in the variance given a constant known mean, and changes in the mean and the variance. All the functions allow for several methods of changepoint detection other than PELT.

B.2. Cost functions

The cost functions determine which criterion is maximized, and in this section all the cost functions we have implemented are presented. The code below shows an overview of the possible cost function.

```
cost.mycpt <- function(intv.dat,type="1d.mean",n=1){
  return(
  switch(type,
         "1d.mean"=cost.1d.mean.mycpt(intv.dat=intv.dat),
         "1d.meanvar"=cost.1d.meanvar.mycpt(intv.dat=intv.dat),
         "pd.mean"=cost.pd.mean.mycpt(intv.dat=intv.dat),
         "pd.meanvar.diag"=
         cost.pd.meanvar.diag.mycpt(intv.dat=intv.dat),
         "pd.meanvar.full"=
         cost.pd.meanvar.full.mycpt(intv.dat=intv.dat),

         "mbic.1d.mean"=
```

```
          cost.mbic.1d.mean.mycpt(intv.dat=intv.dat,n=n),
          "mbic.1d.meanvar"=
          cost.mbic.1d.meanvar.mycpt(intv.dat=intv.dat,n=n),
          "mbic.pd.mean"=
          cost.mbic.pd.mean.mycpt(intv.dat=intv.dat,n=n),
          "mbic.pd.meanvar.diag"=
          cost.mbic.pd.meanvar.diag.mycpt(intv.dat=intv.dat,n=n),
          "mbic.pd.meanvar.full"=
          cost.mbic.pd.meanvar.full.mycpt(intv.dat=intv.dat,n=n)
  )
  )
}
```

The following two sections contain the cost functions of the BIC or mBIC for univariate and multivariate Gaussian data with different restrictions.

B.2.1. Univariate

The first two functions are designed to maximize criteria BIC_1, BIC_2 and BIC_3, and their expressions are given by Equations (3.21) and (3.22). The third and fourth cost functions are from Equations (5.20) and (5.27). The fourth cost function can be used to maximize BIC_4.

```
cost.1d.mean.mycpt <- function(intv.dat,t=0){
  return(sum((intv.dat-mean(intv.dat))^2))
}

cost.mbic.1d.mean.mycpt <- function(intv.dat,t=0,n){
  return(sum((intv.dat-mean(intv.dat))^2)+
  log(length(intv.dat)/n))
}

cost.1d.meanvar.mycpt <- function(intv.dat,t=0){
  t.n=length(intv.dat)
  #sigma.sq.hat=(t.n-1)*var(intv.dat)/t.n
  sigma.sq.hat=sum((intv.dat-mean(intv.dat))^2)/t.n
  if(sigma.sq.hat<0.0000000001){
    sigma.sq.hat=0.0000000001
  }
  return(t.n*log(sigma.sq.hat))
}

cost.mbic.1d.meanvar.mycpt <- function(intv.dat,t=0,n){
  t.n=length(intv.dat)
```

```
  #sigma.sq.hat=(t.n-1)*var(intv.dat)/t.n
  sigma.sq.hat=sum((intv.dat-mean(intv.dat))^2)/t.n
  if(sigma.sq.hat<0.0000000001){
    sigma.sq.hat=0.0000000001
  }
  return(t.n*log(sigma.sq.hat)+
  log(length(intv.dat)/n))
}
```

B.2.2. Multivariate

These are implementations of the cost functions when the data are multivariate Gaussian. The first block of code contain cost functions based on Equation (A.2). When $p = 1$ these simplify to the cost functions implemented in B.2.1. Note that p in this code is the same as u in Appendix A, namely the dimension of x_i.

```
cost.pd.mean.mycpt <- function(intv.dat,t=0){
  # When Sigma is known to be I_p
  mu.hat=colMeans(intv.dat)
  return(sum((intv.dat-mu.hat)^2))
}

cost.pd.meanvar.diag.mycpt <- function(intv.dat,t=0){
  log.sigma.sq.hat = log(colSums((intv.dat-colMeans(intv.dat))^2)/
  dim(intv.dat)[1])
  return(dim(intv.dat)[1]*sum(log.sigma.sq.hat))
}

cost.mbic.pd.mean.mycpt <- function(intv.dat,t=0,n){
  # When Sigma is known to be I_p
  mu.hat=colMeans(intv.dat)
  return(sum((intv.dat-mu.hat)^2)+
  log(length(intv.dat)/n))
}

cost.mbic.pd.meanvar.diag.mycpt <- function(intv.dat,t=0,n){
  log.sigma.sq.hat =
  log(colSums((intv.dat-colMeans(intv.dat))^2)/
  dim(intv.dat)[1])
  return(dim(intv.dat)[1]*sum(log.sigma.sq.hat)+
  log(length(intv.dat)/n))
}
```

The following cost functions are based on Equation (A.5), and are for multivariate Gaussian data when all the elements of Σ are estimated.

```
cost.pd.meanvar.full.mycpt <- function(intv.dat,t=0){
  ## intv.dat had one time stamp in the same row.
  #  Each column is a stream (i.e. temperature in the same column)
  ## Fits a p-dim normal to data and returns cost and mean,var
  # Number of observations
  len = dim(intv.dat)[1]
  p = dim(intv.dat)[2]

  # Mean ML-estimate
  mu.hat=colMeans(intv.dat)
  # Subtract mean from data
  z=as.matrix(sweep(intv.dat,2,mu.hat))

  ## Compute sigma.hat
  # For every row compute t(x_i-mu)(x_i-mu) and #sum over i
  sigma.hat=apply(z, 1, function(x) t(z)%*%(z))
  # Sum each t(x-mu)(x-mu), put into matrix,
  #divide by normalizing
  sigma.hat=matrix(sigma.hat[,1],ncol=p,nrow=p )/len

  ## SVD (singular value decomposition for faster computing)
  # Compute eigenvalues
  eigen = svd(x=sigma.hat,nu=0,nv=0)
  # kutte ut numerisk null
  eigen$d = eigen$d[eigen$d>10^-10]
  # Compute cost based on rank=eigen$d and det(S)=prod(eigen$d)
  cost=len*(length(eigen$d)*(log(2*pi)+1)+
  log(prod(eigen$d))) #cost.K is negative
  return(cost)
}


cost.mbic.pd.meanvar.full.mycpt <- function(intv.dat,t=0,n){
  ## intv.dat had one time stamp in the same row.
  #  Each column is a stream  (i.e. temperature in the same column)
  ## Fits a p-dim normal to data and returns cost and mean,var
  # Number of observations
  len = dim(intv.dat)[1]
  p = dim(intv.dat)[2]

  # Mean ML-estimate
```

```
  mu.hat=colMeans(intv.dat)
  # Subtract mean from data
  z=as.matrix(sweep(intv.dat,2,mu.hat))

  ## Compute sigma.hat
  # For every row compute t(x_i-mu)(x_i-mu) and
  #sum over i
  sigma.hat=apply(z, 1, function(x) t(z)%*%(z))
  # Sum each t(x-mu)(x-mu), put into matrix,
  #divide by normalizing
  sigma.hat=matrix(sigma.hat[,1],ncol=p,nrow=p )/len

  ## SVD
  # NB: requirement is not fulfilled if singular
  # Compute eigenvalues
  eigen = svd(x=sigma.hat,nu=0,nv=0)
  # drop numaric zero
  eigen$d = eigen$d[eigen$d>10^-10]
  # Compute cost based on rank=eigen$d and det(S)=prod(eigen$d)
  cost=len*(length(eigen$d)*(log(2*pi)+1)+
  log(prod(eigen$d))) #cost.K is negative
  return(cost+log(length(intv.dat)/n))
}
```

B.3. Implementation of generalized OP

This is our implementation of the of gOP which may also be found at https://github.com/kristinbakka/generalizedPELT, or in a schematic form in Algorithm 3. The objective of the implementation is to make the algorithms as easy to understand as possible. It is easier to understand gPELT after one understands gOP. The reader is invited to set `my.debug=TRUE` and run all lines except for `for(t in (attrb$minseglen):attrb$n)` in `op.mycpt()`.

```
dat=c(-4.19 , -3.35 , -6.17 , 2.84 , -0.197 , 1.75 , 1.36)
attrb=list(p=1,n=length(dat),minseglen=1,
pen=2*log(length(dat)))

op.mycpt <- function(attrb,dat,type="1d.mean"){
  # exact same as OP, nothing is ever pruned
  # Not manually debugging
  my.debug=FALSE
  # Is type among the selection of cost functions
```

```
if(attrb$p==1){
  if(!is.element(type,c("1d.mean","1d.meanvar",
  "pd.meanvar.diag","mbic.1d.mean,
  "mbic.1d.meanvar","mbic.pd.meanvar.diag"))){
    return("Type is not valid.")
  }
  dat=matrix(dat,ncol=1)
}else{
  if(!is.element(type,c("pd.mean",
  "pd.meanvar.diag","pd.meanvar.full",
  "mbic.pd.mean","mbic.pd.meanvar.diag",
  "mbic.pd.meanvar.full"))){
    return("Type is not valid.")
  }
}

## Initialize first step such that
# s = 0, F(0) = -\pen, s.set={0}, r(0)=0
# Outer data frame of t,F,r
permanent <- data.frame(t=seq(0,attrb$n),
F.val=rep(NA,attrb$n+1),r=rep(NA,attrb$n+1))
permanent[1,2:3]=c(-attrb$pen,0)
s.set=c(0)
if(my.debug){t=attrb$minseglen-1}

## Compute for all data sets lengths shorter
#than attrb$n+1
# Work in delay by starting at minseglen
for(t in (attrb$minseglen):attrb$n){
  if(my.debug&&(t%%25==0)){cat("t=",t,".\n")}
  if(my.debug){t=t+1}

  ## Use cost function to compute int.cost C(s+1,t)
  #  for all s in s.set
  # This is the only place the cost function
  #is evaluated
  temp<-data.frame(
    s=s.set,
    int.cost = sapply(s.set, function(x)
    cost.mycpt(intv.dat=dat[(x+1):t,],
    type=type,n=attrb$n))
  )
```

```
  # This is an overly complex way to do it,
  # but gives a table "permantent" that is
  # easier to interpret to understand the
  # algorithm
  ## Compute full cost and pruning cost
  temp$full.cost <- permanent[s.set+1,2] +
  temp$int.cost + attrb$pen
  temp$prune.cost<- permanent[s.set+1,2] +
  temp$int.cost

  ## Determine smallest (optimal) full cost
  # Save smallest (optimal) full cost
  permanent$F.val[t+1]=min(temp$full.cost)

  # Save previous changepoint, the s with
  #smallest full cost
  # That is the last s for which F.val is
  # minimal
  permanent$r[t+1]=tail(temp$s[
  temp$full.cost==permanent$F.val[t+1]],1)

  ## Prune - prepare next s.set
  # s with smaller pre-beta cost
  A=temp$prune.cost<=permanent$F.val[t+1]
  # or superceding t
  B=temp$s>permanent$r[t+1] #####
  #    B=rep(FALSE,length(A))
  #    if(B&!A){
  #      warning(paste("B&!A for t=",t,".\n"))
  #    }

  ## Only add element to set if it has a valid
  #predecessor
  if(t>=(2*attrb$minseglen-1)){
  s.set = c(s.set,t+1-(attrb$minseglen))
  }

  # Debug
  if(my.debug){cat("t=",t,".\n")}
  if(my.debug){temp}
  if(my.debug){permanent}
  if(my.debug){View(permanent)}
}
```

```
  return(permanent)
}
```

B.4. Implementation of generalized PELT

This is the implementation of the of gPELT. As the objective of the implementation is to make the algorithms as easy to understand as possible, it is readable but not fast. The reader is invited to set `my.debug=TRUE` and run all lines in `gpelt.mycpt()` except for `for(t in (2*attrb$minseglen):(attrb$n))`. The code is also available at https://github.com/kristinbakka/generalizedPELT. The algorithm is also presented as Algorithm 4.

```
dat=c(-4.19 , -3.35 , -6.17 , 2.84 , -0.197 , 1.75 , 1.36)
attrb=list(p=1,n=length(dat),minseglen=1,
pen=2*log(length(dat)))

gpelt.mycpt <- function(attrb,dat,type="1d.mean"){
  # Not manually debugging
  my.debug=FALSE

  # This is an overly complex way to do it, but gives a table
  # "permantent"
  # that is easier to interpret to understand the algorithm


  # Is type among the selection of cost functions
  if(attrb$p==1){
    if(!is.element(type,c("1d.mean","1d.meanvar",
    "pd.meanvar.diag","mbic.1d.mean","mbic.1d.meanvar",
    "mbic.pd.meanvar.diag"))){
      return("Type is not valid.")
    }
    dat=matrix(dat,ncol=1)
  }else{
    if(!is.element(type,c("pd.mean","pd.meanvar.diag",
    "pd.meanvar.full","mbic.pd.mean",
    "mbic.pd.meanvar.diag","mbic.pd.meanvar.full"))){
      return("Type is not valid.")
    }
  }


  ### Initialize first step such that
```

```
# inherit = 0, F(0) = -\pen, s.set={0}, r(0)=0
# Outer data frame of t,F,r
permanent <- data.frame(t=seq(0,attrb$n),
F.val=rep(NA,attrb$n+1),r=rep(NA,attrb$n+1))
permanent[1,2:3]=c(-attrb$pen,0)


### Initialize first step such that
for(t in attrb$minseglen:min(2*attrb$minseglen-1,attrb$n)){
  # predecessor is 0th data point
  permanent[permanent$t==t,2:3]=
    c(cost.mycpt(intv.dat=dat[(1):t,],type=type,n=attrb$n),0)
}
# Return if finished
if(attrb$n<2*attrb$minseglen){
  return(permanent)
}
# Else construct Inherit such that
# When we inherit from time t, we get the s.set
# at Inherit[[t+1]]
# inherit$q is the data point we inherit from,
# inherit$s is the pruned s.set at the time we inherit from
Inherit=as.list(c(rep(0,2*attrb$minseglen),
rep(NA,attrb$n-3*attrb$minseglen+1)))

if(my.debug){t=2*attrb$minseglen-1}


####
## Compute for the rest of the data points
for(t in (2*attrb$minseglen):(attrb$n)){
  if(my.debug&&(t%%25==0)){cat("t=",t,".\n")}
  if(my.debug){t=t+1}
  ###  Combine inherited and earned data points to get s.set
  s.set=c(Inherit[[t-attrb$minseglen+1]], #inherited
          max(attrb$minseglen,
          t-2*attrb$minseglen+1):(t-attrb$minseglen)) #earned


  ###  For a changepoint at t find
  #    best most recent changepoint s
  # Use cost function to compute int.cost C(s+1,t)
  # for all s in s.set
```

```
    temp<-data.frame(
      s=s.set,
      int.cost = sapply(s.set, function(x)
      cost.mycpt(intv.dat=dat[(x+1):t,],type=type,n=attrb$n))
    )
    ## Compute full cost and pruning cost
    temp$full.cost <- permanent[s.set+1,2] + temp$int.cost +
    attrb$pen
    temp$prune.cost<- permanent[s.set+1,2] + temp$int.cost

    ## Determine smallest (optimal) full cost
    # Save smallest (optimal) full cost
    permanent$F.val[t+1]=min(temp$full.cost)

    # Save previous changepoint, the s with smallest full cost
    # That is the last s for which F.val is minimal
    permanent$r[t+1]=
    tail(temp$s[temp$full.cost==permanent$F.val[t+1]],1)


    ### Remove non-optimal predecessors
    ### Remember which data points to inherit
    # s with smaller pre-beta cost, the ones to keep
    A=temp$prune.cost<=permanent$F.val[t+1]

    ## Only add element to next s.set if it has a
    #  valid predecessor
    if(length(A==TRUE)==0){
      Inherit[[t+1]]=NULL
    }else{
      Inherit[[t+1]]=temp$s[A]
    }

    # Debug
    if(my.debug){t}
    if(my.debug){s.set} #current s.set to go through,
    # out to be 0 until 2*minseglen

    if(my.debug){t}
    if(my.debug){temp}
    if(my.debug){Inherit[[t+1]]}
    if(my.debug){t-(attrb$minseglen)} # Inherited from
    if(my.debug){inherit$s[inherit$q==t]} # legacy
```

```
    # (inheritance passed on from this node (ought to be 0
    # until 2*attrb$minseglen)


    if(my.debug){temp}
    if(my.debug){permanent}
    if(my.debug){View(permanent)}
  }
  if(FALSE){cat('\n1 run of gPELT performed.\n')}
  return(permanent)
}
```