# Offline Changepoint Detection Using Modifications on Binary Segmentation

Jasmine Burgess

## 1 Background

Changepoint detection is the problem of finding the points in an ordered dataset where the parameters of the underlying distribution shift abruptly. This problem has applications in genomics [Olshen et al., 2004], environmental monitoring [Beaulieu et al., 2012] and economics [Heßler et al., 2023]. It is possible to consider both online and offline changepoint detection; in offline scenarios, it is assumed all the data is available for us to analyse, whereas in online scenarios, the data is fed sequentially into the algorithm, with the aim of finding changepoints in real-time, and accuracy has to be balanced with how long detection takes after the changes occurs. Through this report, I will only consider offline changepoint detection.

If it is known that the data has at most one changepoint, the problem is relatively trivial in an offline setting. Since there are only $N-1$ possible locations of the changepoint, where $N$ is the length of the data, it is possible to evaluate a test statistic at each datapoint to determine if and where the changepoint occurs. However, if there is a maximum of $k$ possible changepoints, there will be more than $N$ choose $k$ possibilities, and it is probably not computationally feasible to calculate a statistic for all of them.

There are generally two approaches taken with changepoint detection algorithms. The first approach uses dynamic programming to directly solve a global minimisation problem over all possible number and locations of changepoints. Earlier methods taking this approach include Segment Neighbourhood Search and Optimal Partitioning; recently methods such as PELT [Killick et al., 2012] and FPOP [Maidstone et al., 2014] have managed to decrease computational cost by reducing (or "pruning") the number of possibilities that must be considered.

The second approach makes modifications to Binary Segmentation, the simplest method and, according to [Killick et al., 2011], "the most established search algorithm within the changepoint literature". Binary Segmentation relies on repeatedly looking for single changepoints within the data. In Section 2, we will establish the framework for modelling and solving changepoint problems and discuss Binary Segmentation. In Section 3, we will examine two recent algorithms which modify, before exploring Seeded Binary Segmentation in Section 4, which builds on both recent methods from Section 3.

## 2 Modelling Changepoints and The First Detection Algorithm

### 2.1 Modelling Changepoints Using Test Statistics

Let us consider our data $Y = \{y_1, y_2, ..., y_N\}$, and we wish to find the changepoints $\{\tau_1, ..., \tau_k\}$ where the parameters of the distribution generating the data change. The data between changepoints, e.g., $y_{\tau_1+1} : y_{\tau_2}$ are called **segments**, and the entire set of changepoints forms a **segmentation** of the data.

There are different options to quantify which segmentations fit the data better. For algorithms based on dynamic programming, since they minimise globally, it is necessary to have a function which determines the quality of any segmentation. Therefore, generally a model will be introduced and the following penalised cost functions will be used:

$$k\phi(N) - \sum_{j=1}^{k+1} f(y_{\tau_{j-1}+1} : y_{\tau_j} | \hat{\Theta}_j),$$

where $f$ is the log likelihood of the data given the parameters $\Theta$, where $\hat{\Theta}_j$ is the maximum likelihood estimators for that segment. The term $k\phi(N)$ prevents overfitting by penalising segmentations with more changepoints, since as the number of changepoints increases, it allows the likelihood to increase, as there is more flexibility to match the data.

However, for binary segmentation methods, we only look for a single changepoint each time, so different approaches can be taken. In general, the algorithms require a test statistic $T(l, u)(s)$ where $1 \leq l < u \leq N$ to determine the best point to split the data within the interval $[y_l, y_u]$, but there is more variation in functions used between methods, with algorithms such as Narrowest Over Threshold and Circular Binary Segmentation developing method-specific test statistics [Olshen et al., 2004] [Baranowski et al., 2019]. We can view the problem of detecting a changepoint as a hypothesis test [Killick et al., 2011]:

$$\begin{aligned} \mathcal{H}_0 &: k = 0, \quad \text{no changepoint in } [l, u] \\ \mathcal{H}_1 &: k = 1, \quad \text{one changepoint in } [l, u] \end{aligned}$$

If the test statistic is above a certain pre-determined threshold $A$, the null hypothesis will be rejected and the method will find a changepoint in $[l, u]$. The standard approach to constructing a test statistic is through a generalised log likelihood ratio approach. Thus:

$$T(l, u)(s) = 2[f(y_{1:s}|\hat{\Theta}_1) + f(y_{s+1:N}|\hat{\Theta}_2) - f(y_{1:N}|\hat{\Theta}_3)]$$

That is, $T(l, u)(s)$ is the maximum log likelihood of the data with a changepoint at $s$ with the maximum log likelihood of the data in the case of the null hypothesis subtracted. Then $s* = \arg\max_{s:l \leq s < u} T(l, u)(s)$ is considered the best candidate for a changepoint, and corresponds to finding the maximum likelihood estimator for the changepoint when fitting a piecewise constant function with one discontinuity at $s$ onto the data between $l$ and $u$. If $T(l, u)(s*) > A$, we would reject the null hypothesis.

The generalised log-likelihood ratio can also be modified so the test statistic is recursive; that is, for $T(l, u)(s)$ can be calculated by using $T(l, u)(s - 1)$, decreasing computational cost; this was proposed by [Page, 1954] and is called the CUSUM statistic. If the non-changing parameters are not known, the CUSUM statistic is sub-optimal compared to the likelihood ratio approach, however it can be used optimally in certain settings [Granjon, 2014].

For example, let us consider the following simple setting of univariate Gaussian means, where the CUSUM statistic will commonly be used. Our data is modelled as:

$$Y_i = f_i + \epsilon_i, \qquad i = 1, ..., N,$$

where $f$ is a one-dimensional, piecewise constant function with discontinuties at $\{\tau_1, ..., \tau_k\}$ and the random noise $\epsilon$ is i.i.d standard Normal. It is generally possible to accurately estimate the variance of noise using methods such as median absolute deviation, so this is a reasonable assumption.

In this case, the CUSUM statistic is as follows:

$$T(l, u)(s) = \sqrt{\frac{u - s}{(u - l)(s - l + 1)}} \sum_{i=l+1}^{s} Y_t - \sqrt{\frac{s - l + 1}{(u - l)(u - s)}} \sum_{i=s+1}^{u} Y_t.$$

### 2.2 Binary Segmentation

Binary Segmentation is an intuitive and commonly used algorithm for changepoint detection, proposed by Scott A.J. and Knott M. (1974) [Scott and Knott, 1974], which works as follows;

1. Find $s* = \arg\max_{s:1\leq s<N} |T(1,N)(s)|$ by evaluating the test statistics for all data points. If $|T(1,N)(s*)| > A$, where $A$ is a threshold, then add $s*$ to the list of estimated changes-points. Otherwise, the algorithm is finished.

2. Split the data into $\{y_1, ..., y_{s*}\}$ and $\{y_{s*+1}, ..., y_N\}$ and repeat the previous stage for the two sets.

3. Continue looking for changepoints and splitting the data until every segment has the maximum test statistic for a changepoint under the threshold.

Using the CUSUM test statistic or generalised likelihood-ratio statistic, the first stage of Binary Segmentation involves finding a maximum likelihood estimator for a single changepoint in the data, so, if there is more than one changepoint, this is essentially model misspecification and can lead to extreme mistakes. This is particularly likely in settings where the signal has discontinuities that cancel out, by first increasing then decreasing, for example. Additionally, Binary Segmentations is only consistent if changepoints are more than $N^{3/4}$ apart, even when the size of the jump is bounded away from zero [Fryzlewicz, 2014].

The biggest advantage to Binary Segmentation is its computation efficiency, which is of order $\mathcal{O}(N\log(N))$. Dynamic programming based methods typically have computation efficiency of maximum order $\mathcal{O}(N^2)$, although, while this is still the worst case scenerio for PELT and FPOP, they generally much quicker than this, and FPOP can be competitive with Binary Segmentation [Maidstone et al., 2014].

## 3 Algorithms Using Random Intervals

Wild Binary Segmentation [Fryzlewicz, 2014] and Narrowest Over Threshold [Baranowski et al., 2019] are two similar methods designed to improve the statistical properties of binary segmentation, although they do compromise on computational cost.

If there is only one changepoint in the data, which is sufficiently far from the endpoints, so there are data before and after the function changes, the test statistic is likely to correctly identify the changepoint. The intuition behind both Wild Binary Segmentation and Narrowest Over Threshold is that if we look for changepoints over different subintervals of the data, it is probable that some intervals will only contain one changepoint which is not close to the endpoints of the intervals, making it easy to detect and leading to a high test statistic at those points.

Both methods begin as follows, requiring a pre-determined interval number $M$, test statistic $T(l,u)(s)$ and a threshold $A$:

1. Form $M$ random intervals from the data $[l_m, u_m]$ where $m = 1, ..., m$ by drawing the lower and upper limits uniformly, independently and with replacement from $\{1, ..., N\}$. One should choose $M$ to be as large as computationally feasible; increasing $M$ reduces the effect of randomness on the resulting estimated changepoints and results in greater expected accuracy at the expense of computational cost.

2. In each of the $M$ intervals, find the best candidate for a changepoint $s*$ and the corresponding value of the test statistic $T(l_m, u_m)(s*)$. If the value of the test statistic is less than the threshold $A$, discard this interval.

3. Select one of the candidate changepoints with test statistic over the threshold to add to the list of estimated changepoints. The two methods differ in their selection criterion.

4. Discard any interval which contains the estimated changepoint. This is necessary so the same changepoint is not found in slightly different locations.

5. Continue selecting changepoints and discarding intervals which contain them until there are no intervals left. The list of estimated changepoints is then returned.

If the selection criterion is greedy and the candidate changepoint with the highest test statistic is chosen, then this algorithm is called **Wild Binary Segmentation**. Generally, the CUSUM statistics is used for this method, as it works best for the univariate Gaussian setting.

Alternatively, the **Narrowest Over Threshold** method selects the candidate changepoint that was found over the narrowest interval - that is, where $u_m - l_m$ is smallest. This method is applicable to settings other than piecewise constant functions; for example, it can detect changes in the gradient of an underlying function or variance of the random noise better than Binary Segmentation or Wild Binary Segmentation. This is because in these settings, large intervals containing multiple changepoints may result in the test statistic incorrectly identifying points as changepoints with a high corresponding test statistic and so Wild Binary Segmentation and Binary Segmentation selects them as true changepoints.

Both methods perform better in settings with fewer changepoints [Fryzlewicz, 2020]. As the number of changepoints increases while $M$ remains fixed, there is a decreasing probability that there is a suitable interval to detect every changepoint, but since it is hard to calculate the true number of changepoints, it may not be clear whether it is necessary to increase $M$ in order for the methods to be accurate. Additionally, increasing $M$ can drastically increase computational cost, meaning that it may not be feasible to choose a very large $M$ "just in case".

# 4 Seeded Binary Segmentation

Seeded Binary Segmentation [Kovács et al., 2022] builds on the Wild Binary Segmentation and Narrowest Over Threshold methods by changing Step 1 of the previous methods.The disadvantage to these methods is that due to the randomness of interval selection it is expected there will be many intervals of length $\mathcal{O}(N)$. Not only do longer intervals require more computational effort to search for the candidate changepoint, they are less likely to contain only one changepoint and so are less useful than shorter intervals. The idea of Seeded Binary Segmentation is to deterministically create intervals with fewer large intervals, significant overlaps between intervals of a fixed size and the shortest intervals around the length of the minimum segment size. Thus, for each changepoint, there will be an interval where it is not close to the endpoint and is the only changepoint in that interval, while constructing fewer intervals than in the previous methods. This also makes Seeded Binary Segmentation robust to the number of changepoints; regardless of the number of changepoints, there will be a suitable interval for each changepoint's detection, and increasing the number of changepoints won't increase computational cost.

Concretely, for a given decay parameter $a \in [\frac{1}{2}, 1)$, and $1 \leq k \leq \lceil -\log_a(N) \rceil$, we define the $k^{th}$ layer as the collection of $n_k$ intervals with approximate length $l_k$, with approximate $s_k$ shift between intervals as

$$\mathcal{I}_k = \bigcup_{i=1}^{n_k} \{[\lfloor (i-1)s_k \rfloor, \lceil (i-1)s_k + l_k \rceil]\}$$

where

$$n_k = 2\lceil a^{1-k} \rceil - 1, \quad l_k = Na^{k-1}, \quad s_k = \begin{cases} 0 & \text{if } k = 1, \\ (N - l_k)/(n_k - 1) & \text{otherwise} \end{cases}$$

and the total collection of intervals is the union of all the layers. If it is known that the minimum length of segments is some $l$, the layers where $l_k < l$ do not need to be calculated or used. However, if all layers are used, the final layer will contain $n_{\lceil -\log_a(N) \rceil} \approx 2N - 1$ intervals of length $\mathcal{O}(1)$. Both the greedy selection method employed by Wild Binary Segmentation and the narrowest over threshold selection detailed above can be used, with the narrowest over threshold selection allowing the Seeded Binary Segmentation algorithm to be applied in more complicated settings.

The computational cost of Seeded Binary Segmentation primarily depends on the total length of all of the intervals, as the method requires calculating the test statistic for each point in each interval. Due to rounding to integer values, an upper bound for the interval length of the $k^{th}$ layer is $l_k + 2$, and we can bound the total length as follows:

$$\begin{aligned}
\sum_{k=1}^{\lceil -\log_a(N) \rceil} n_k(l_k + 2) &= \sum_{k=1}^{\lceil -\log_a(N) \rceil} (Na^{k-1} + 2)(2\lceil a^{1-k} \rceil - 1) \\
&\leq \sum_{k=1}^{\lceil -\log_a(N) \rceil} (Na^{k-1} + 2)(2a^{1-k} + 1) \\
&= \sum_{k=1}^{\lceil -\log_a(N) \rceil} 2N + 2 + 4a^{1-k} + Na^{k-1} \\
&\leq \lceil -\log_a(N) \rceil (2N + 2 + 4a^{1-\lceil -\log_a(N) \rceil} + N) \quad \text{since } a^{k-1} \text{ is decreasing in } k \\
&\leq \lceil -\log_a(N) \rceil (7N + 2) = \mathcal{O}(N \log_{1/a}(N)) = \mathcal{O}(N \log(N)).
\end{aligned}$$

This means that the computational cost of Seeded Binary Segmentation can be competitive with Binary Segmentation, while improving its accuracy.

The threshold $A$ determines whether a potential changepoint is considered significant enough. While it is possible to derive an approximate value of $A$ for a given significance level using asymptotic theory [Gupta and Chen, 1996], Kovacs S. et al suggest calculating the set of changepoints with different values of $A$ and then optimising over the possible sets using an Information Criterion to further improve accuracy. In this is done, the computational cost of Seeded Binary Segmentation will be $\mathcal{O}(N \log(N))$ if using greedy selection or $\mathcal{O}(N^2 \log(N))$ if using the narrowest over threshold selection.

## 5 Conclusions

We have examined four methods for detecting changepoints: binary segmentation and three methods which build upon it. Binary segmentation is computationally fast but inaccurate, and struggles in settings other than changes in mean with Gaussian noise. Both Wild Binary Segmentation and Narrowest Over Threshold improve the statistical properties of Binary Segmentation while compromising on computational efficiency; in addition, the Narrowest Over Threshold algorithm is applicable to a wider range of settings. Finally, Seeded Binary Segmentation reduces their computational cost, while maintaining the improved accuracy, and can be used with greedy or narrowest over threshold selection methods, allowing it to be used in a wide range of settings.

There are several extensions that could be done to explore this area further. Firstly, we could explore Wild Binary Segmentation 2, which was proposed in [Fryzlewicz, 2020] and is also designed to

improve on Wild Binary Segmentation. Wild Binary Segmentation 2 remains fairly specific to the univariate Gaussian means setting, and there is no investigation of how it works in other settings. Another algorithm based on Binary Segmentation is Circular Binary Segmentation ([Olshen et al., 2004]), which is specifically designed to detect changes in DNA sequence copy number. Secondly, it would be desirable to perform a simulation study using different datasets to demonstrate the strength and weaknesses of the algorithms discussed.

# References


[Baranowski et al., 2019] Baranowski, R., Chen, Y., and Fryzlewicz, P. (2019). Narrowest-over-threshold detection of multiple change points and change-point-like features. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(3):649–672.

[Beaulieu et al., 2012] Beaulieu, C., Chen, J., and Sarmiento, J. (2012). Change-point analysis as a tool to detect abrupt climate variations. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 370:1228–49.

[Fryzlewicz, 2014] Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6).

[Fryzlewicz, 2020] Fryzlewicz, P. (2020). Detecting possibly frequent change-points: Wild binary segmentation 2 and steepest-drop model selection. *Korean Stat. Soc.*, 49:1027–1070.

[Granjon, 2014] Granjon, P. (2014). The cusum algorithm - a small review. *HAL*.

[Gupta and Chen, 1996] Gupta, A. and Chen, J. (1996). Detecting changes of mean in multidimensional normal sequences with applications to literature and geology. *Computational Statistics*, 11(3):211–221.

[Heßler et al., 2023] Heßler, M., Wand, T., and Kamps, O. (2023). Efficient multi-change point analysis to decode economic crisis information from the s&p500 mean market correlation.

[Killick et al., 2011] Killick, R., Eckley, I. A., and Fearnhead, P. (2011). *Analysis of changepoint models.*

[Killick et al., 2012] Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *J. Amer. Statist. Assoc.*, 107(500):1590–1598.

[Kovács et al., 2022] Kovács, S., Bühlmann, P., Li, H., and Munk, A. (2022). Seeded binary segmentation: a general methodology for fast and optimal changepoint detection. *Biometrika*, 110(1):249–256.

[Maidstone et al., 2014] Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. (2014). On optimal multiple changepoint algorithms for large data.

[Olshen et al., 2004] Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4):557–572.

[Page, 1954] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.

[Scott and Knott, 1974] Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30(3):507–512.