Change Point Detection

IE 683: Course Project Report

Submitted By:

Sayan Chatterjee

Roll Number: 193190005

Yogesh Nagar

Roll Number: 193190014

Rahul Khankar

Number: 193190008

Under Supervision of

Prof. Nandyala Hemchandra

Industrial Engineering and Operations Research

Indian Institute of Technology Bombay (IIT Bombay)

Powai, Mumbai

Maharashtra -400076

Contents

Abstract

Change Detection or Change Point Detection (CPD) is the method of detecting data points whenever there is alteration in the probability distribution of a time series or that of a stochastic process. It is having its inceptions very nearly 60 years prior when Page, Shiryayev and Lorden first worked on change-point in time series on sequential data. The process on which they worked on was ordinarily quality control especially on quality of a nonstop production process and the change-point demonstrated a rot in quality that should be identified and appropriate moves ought to be made. As of late, inspiring from a wide assortment of uses lead to various issues. Presently, individuals regularly study if a change has occurred. Additionally, discovering more than one changes and distinguishing the hours of any such changes is likewise a significant goal. Some specific applications, may be concerned with changes in some statistics like mean, variance, regression model of the process. Change Point Detection Algorithms able to detect changes which is missed by control charts in statistical quality control. Further, a portion of the calculations gives certainty levels and certainty spans to recognize changes. In this work, the main focus is to study and implement online and offline methods of change point detection. Under this umbrella, we study some optimal and heuristics of Change Point Detection. Also, general metrics of Change Point Detection and some practical applications of change point detection are also reviewed in this context.

Chapter 1

Introduction

Change point detection is the way of identifying the changes in the incoming signal or data points in the domain of signal processing/ time series/ stochastic process. Let $x_1, \ldots, x_n$ is a succession of time series data points. We say that $1 < i < n$ is change point if

$$P_{X_1} = \ldots = P_{X_{i-1}} \neq P_{X_i} = \ldots = P_{X_n}$$

where P_{X_i} denotes the PDF of the time series at that particular point x_i. Historically, the first work of CPD has taken place back in the 50s. The aim of the work was to discover a shift in the mean of *i.i.d.* Gaussian variables for quality assurance objective. From that onward, the issue of CPD is actively considered. This topic has generated important activity and attracted many researchers in statistics and signal processing. However, the application of CPD algorithms are not limited to only these three aspects, rather it has applications in speech processing, financial analysis, bio-informatics , climatology , network traffic data analysis etc. Current applications in account, bioinformatics, checking of complex frameworks have likewise animated ongoing advancement from the AI people group.

Most of the methods can be broadly divided into two sections: online methods, that focus to recognize changes when they show up in genuine time, and offline methods that find changes after all examples are gathered. The task of online detection is frequently alluded to as event or anomaly detection, while the offline detection at times is called signal segmentation.

Let us assume we are collecting data coming from a natural phenomenon during a time period where there is no trend or seasonality, then there is unlikely for the distribution parameters to change. However, the speculations frequently failed in real life, where the environment does undergo potential changes, and a more sensible hypothesis is to consider that the observed phenomenon is stationary only on smaller instance of time-units but not for a larger. The point of change-point detection is to discover these fragments as well as to find out change points as precisely as could be expected..

Figure 1.1 plots a sample time series that consists of several sub-signals, hence have various change points. The data is about mean annual temperature trends of certain place for time 1899-2010. Figure 1.1 point up the surveillance that the climate of the place went through six different regimes in this time. The segment of the time series is called as states when the parameters which govern the activity do not change. The change point detection recognize these boundaries that separate different sub-signals by locating multiple change points.

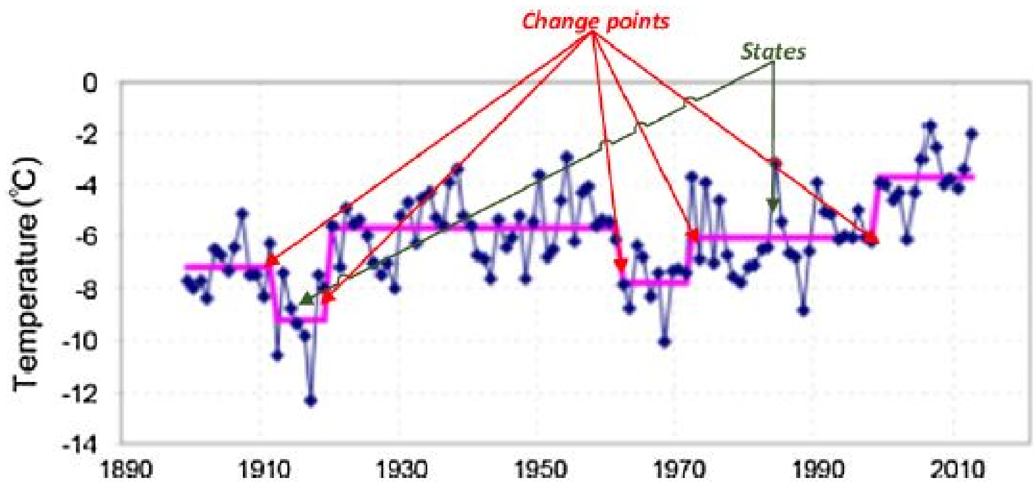

Figure 1.1: Presence of Different Sub-signals in Time Series Data

Chapter 2

Background

2.1 Different Components of Change Point Detection

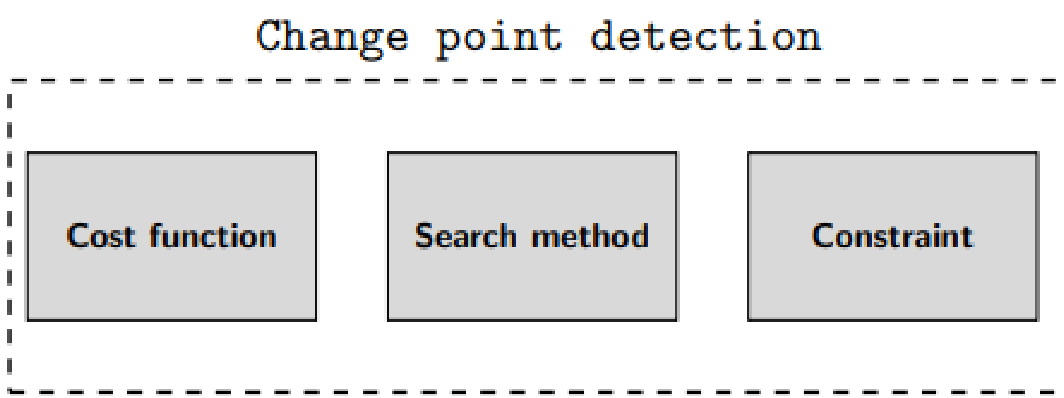

Figure 2.1: Different Components of Change Point Detection[2]

There are mainly 3 components of Change Point Detection, namely Cost Function, Search Method and Constraint.

1. **Cost Function**: It is basically the objective function of the optimization problem. This is principal component of change point detection.

2. **Search Method**:For Discrete optimization problems, search method is the solution procedure. There are several methods that can efficiently solve these optimization problems.

3. **Constraint**: Some penalty constraints are added when unknown number of changes is present. The output depends on the extent of penalty. If the penalty is very large, number of change points detected decrease.

2.2 Classifications of Change Point Detection Methods

2.2.1 Online v/s Offline Settings

In online setting we obtain X_1,X_2, $\ldots$,X_n observations one at a time, while in offline setting we obtain all observation X_1,X_2, $\ldots$,X_n in one go. In offline case, there is no real time data processing, Note that on-line procedures are applicable to off-line setting as well, by going along the data-points once at a time as if they are observed. By and by, no CPD algorithm work in real time. Because to check whether a change has occurred between the new and old observation, it is necessary to check new data. ϵ-real time algorithm needs at least ϵ data points in the new lot of data to find change points. ∞-real time algorithm is an offline algorithm and the complete online algorithm is 1-real time because for every observation, it ought to anticipate if change point happens before the new data point. Smaller ϵ values means change point detection algorithm is strong.

2.2.2 Supervised and Unsupervised Models

Many ML algorithms have been drafted for CPD. Here, we have discussed basic algorithms those are which are commonly used for the CPD problem,but we mainly focus on unsupervised methods. The strategies incorporate both unsupervised and supervised methods, contingent upon the described output of the algorithm. Supervised learning algorithms are ML algorithms learning in which input data is mapped to target output.[3] When a supervised technique such as multiclass or binary classifiers used for CPD.
Unsupervised learning algorithms are to a great extent acclimated with discover designs in unlabeled information. Unsupervised learning is used to subdivide time series, to find change points in the for CPD, such algorithms are use statistical features of the observation. Unsupervised learning is alluring since it can handle many situations without requiring prior training for every situation.

2.2.3 Single v/s Multiple Change Point Detection

Detection of single or multiple change points in the off- line environment are contrasting. Multiple change-points in the off-line environment is hard since the number change points for the process increases from $n-1$ to 2^{n-1} if the quantity of change-points is obscure, Here n denotes number of observations. We can use multiple change points methods for detecting single change point. Vice versa is also feasible but there is no guarantee it will yield the desired outcome.

2.2.4 Parametric v/s Non-Parametric Models

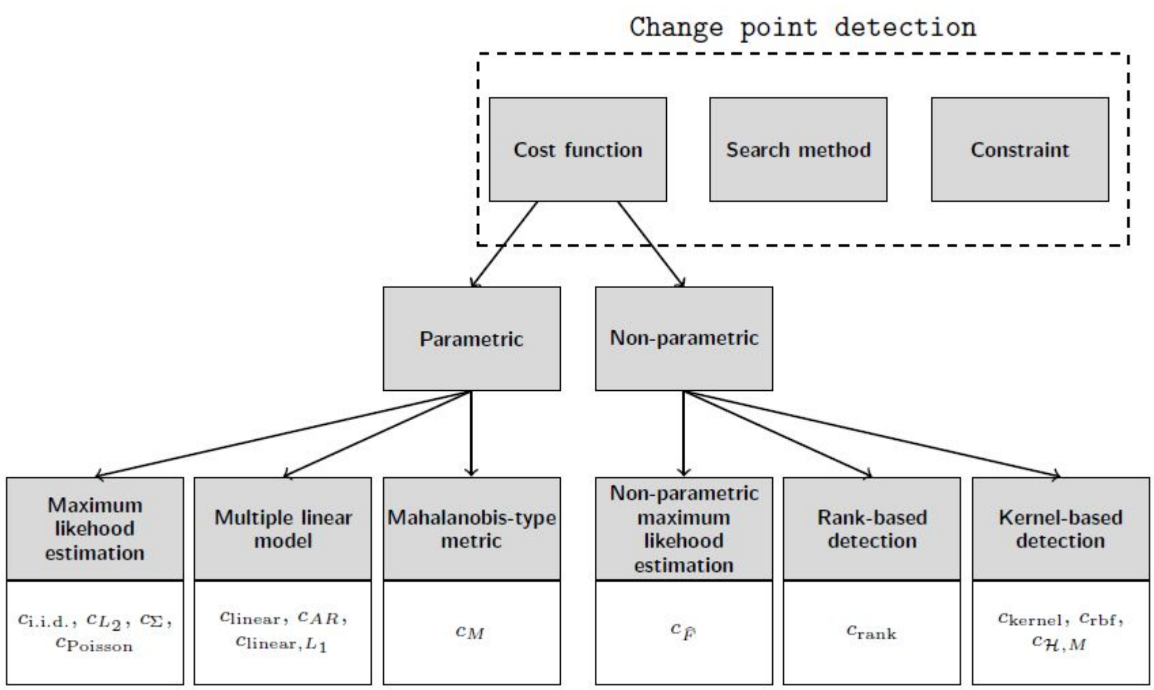

Figure 2.2: Typology of The Cost Functions[2]

Parametric model have assumption that the distribution of the data points which belongs to a variety of distributions can be delineate by a limited number of parameters. Thus the change occurs in minimum one parameter which describes the distribution. These assumptions are important design and the investigation of change-point procedures.

There are differences between parametric and non-parametric approaches. Non-parametric approaches achieved huge success for massively large data-

sets. Also, non-parametric models are highly cost efficient as compared to parametric model and undertake scale with the size of the data-set.
Characterization of the cost function is used for classification of parametric and nonparametric models. MLE, Multiple Linear Model and Mahalanobis Distances are some examples of Parametric models. Rank based detection and Kernel based detection are Non-parametric models.

2.3 Organization of the Report

The report of this project is organized as follows. Chapter 3 talks about various CPD Algorithms in online as well as in offline setup. The next chapter talks about different constraints that are useful when number of change points are unknown beforehand. Chapter-5 talks about performance measures of change point detection. We present some common applications of CPD in human life in chapter-6 and conclusion of the project is presented in chapter-7.

Chapter 3

Description of Algorithms

Extensively CPD techniques are partitioned into two areas: online and offline methods. Online methods means to identify changes focuses promptly as they happen in real time, and offline methods expects to find change point solely after all data points are received. The first task is called as event or anomaly detection, while the second is called as signal segmentation.

3.1 Offline Methods

Different offline algorithm used to identify various change focuses in multivariate time arrangement are talked about in this part.

3.1.1 Problem formulation

Let us take a multivariate non-stationary process $y = \{y_1, y_2, ...y_T\}$ which takes value in $R^d(d \geq 1)$ and which has total T data points. Here, the signal y is piece wise stationary, meaning is that some of the parameters of the observations change sharply at some unknown time $t_1^* \leq t_2^*.... \leq t_K^*$. CPD aims on evaluating the indexes t_k^*. Depending on the conditions, the number K^* of changes might be unknown, some times it has to be approximated too.
Officially, CPD is demonstrated as a model selection problem, which comprises in picking the most ideal partition T which minimizes cost function V(T, y). The cost function V(T) for a specific partition is a amount of costs of all the partition:

$$V(T, y) = \sum_{k=0}^{K} c(y_{t_k...t_{k+1}})$$

Here c() is a cost function which define goodness-of-fit of the sub-signal. The "best segmentation" T is the minimizer of the cost function $V(T)$. Contingent upon whether number of change points K^* known or unknown beforehand, CPD techniques broadly divided into 2 branches.

1. Number of change points K is known
   Solution of CPD problem with fixed known number of change points K is given by following discrete optimization problem.

$$\min_{|T|=K} V(T)$$

2. Number of change points K is unknown
   Solution of CPD problem with fixed unknown number of change points K is given by following discrete optimization problem.

$$\min_{|T|} V(T) + penalty(T)$$

   Here $penalty(T)$ is an measure of the cost of a segment T .

All CPD approaches considered during work will have exact or approximate solution to (1) or (2)

3.1.2 Search Methods

Search techniques are comprehensively order into two classifications, as demonstrated in figure beneath. Optimal methods gives the exact solution and the approximate methods gives an approximate solution. given algorithms can be merged with cost functions. In view of the chosen cost function, the computational complexity of the complete algorithm changes. Most common Search Methods are as follow:

- Optimal Method
  - Optimal: When number of Change points are known
  - Penalty: When number of Change points are unknown
- Approximate Method
  - Window-Sliding
  - Binary Segmentation
  - Bottom-Up Segmentation

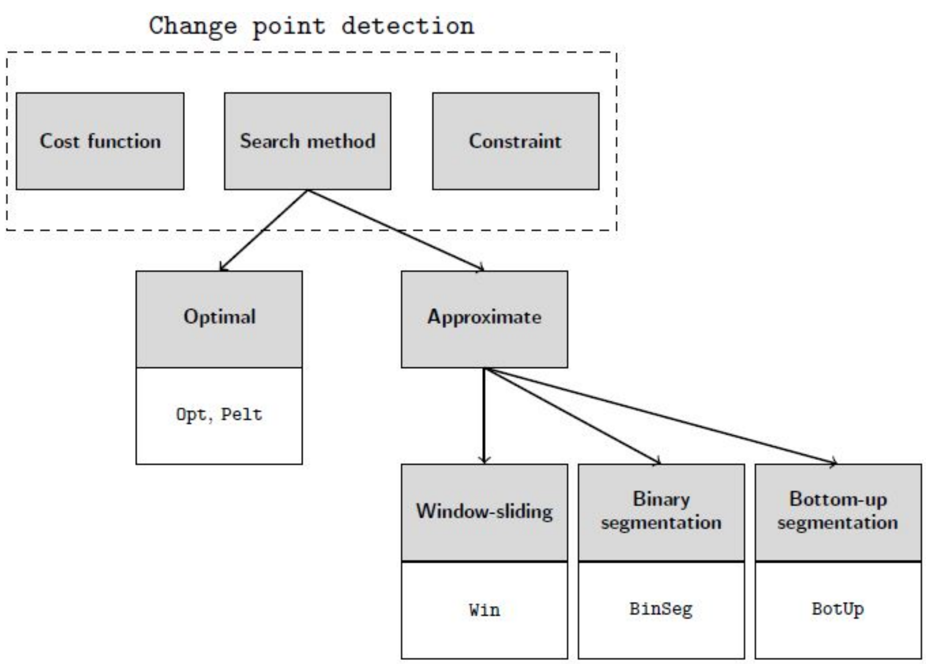

Figure 3.1: Typology of the cost functions[2]

Window Sliding

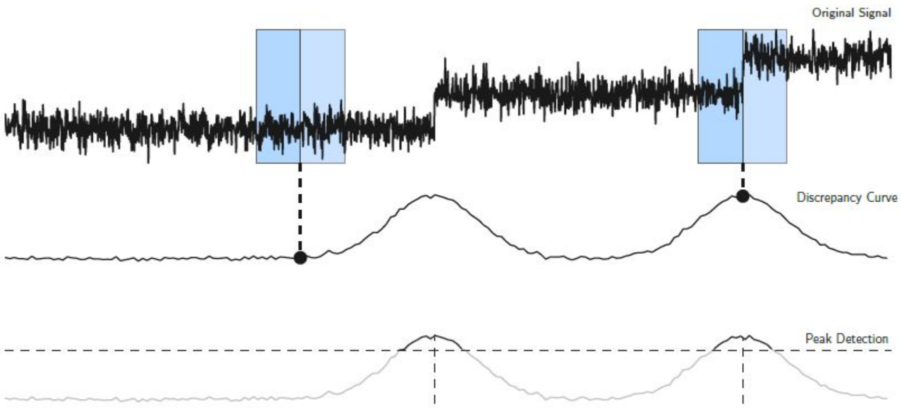

Figure 3.2: Window Sliding

The window-sliding algorithm is a fast approximate algorithm which is alternative to the optimal methods. Window sliding method compute the dissimilarity between 2 adjoining windows which slide along the signal. this dissimilarity between two sub-signals is given by,

$$d(y_{a..t}, y_{t..b}) = c(y_{a..b}) - c(y_{a..t}) - c(y_{t..b}) \quad (1 \leq a < t < b \leq T):$$

When 2 adjacent windows cover dissimilar segments and the dissimilarity reaches large values it flag that as change point, The main advantages of window sliding method is low complexity (linear) and ease of implementation.

Binary Segmentation

Binary segmentation, is likewise a choice to approximate the optimal methods, since it is mathematically straightforward and it can be implemented easily. Binary segmentation is a greedy and sequential algorithm, described as follows. The first change point estimation t^1 is given by

$$t^1 = \text{argmin}_{1 \leq t < T-1}\, c(y_{0..t}) + c(y_{t..T})$$

This method is greedy, because it look for the change point which lowers the summation of costs. The data points are divided into two at the location of t^1; the same operation is duplicated until termination.

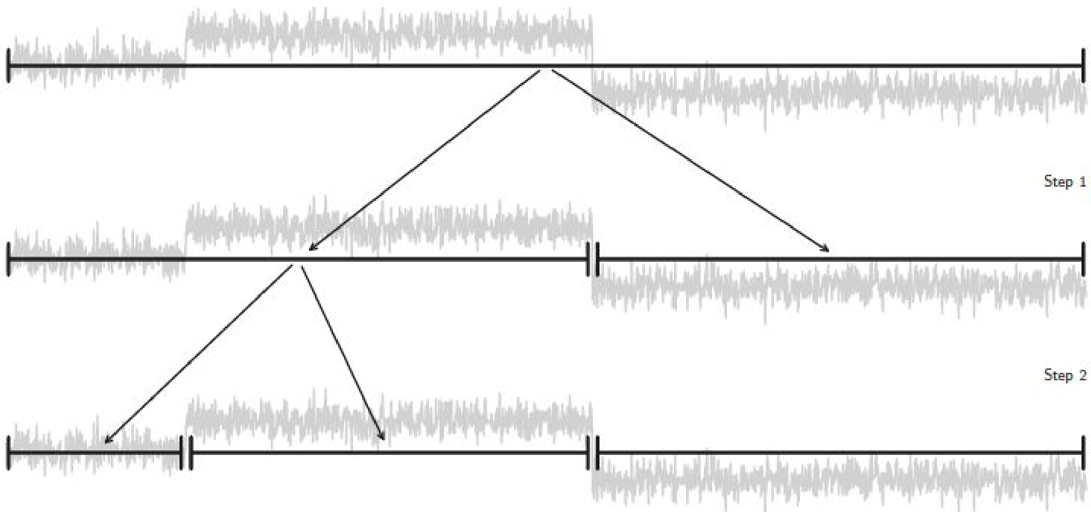

Figure 3.3: Schematic example of Binary Segmentation

```
algo = rpt.Binseg(model=model).fit(signal)
my_bkps = algo.predict(n_bkps=n_bkps)

# show results
rpt.show.display(signal, bkps, my_bkps, figsize=(10, 6))
plt.show()
```

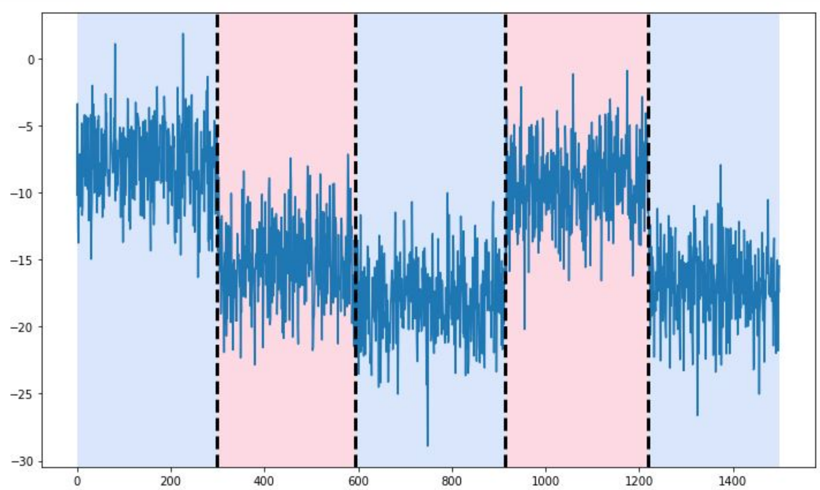

Figure 3.4: Output of Binary Segmentation When Change Points are known

Bottom Up Segmentation

Bottom-up segmentation, is inverse of Binary Segmentation. Opposite to Binary Segmentation, Bottom Up segmentation begins by dividing the original data points into numerous little parts and then step-wise add them till number

```
my_bkps = algo.predict(pen=50)
```

```
rpt.show.display(signal, bkps, my_bkps, figsize=(10, 6))
plt.show()
```

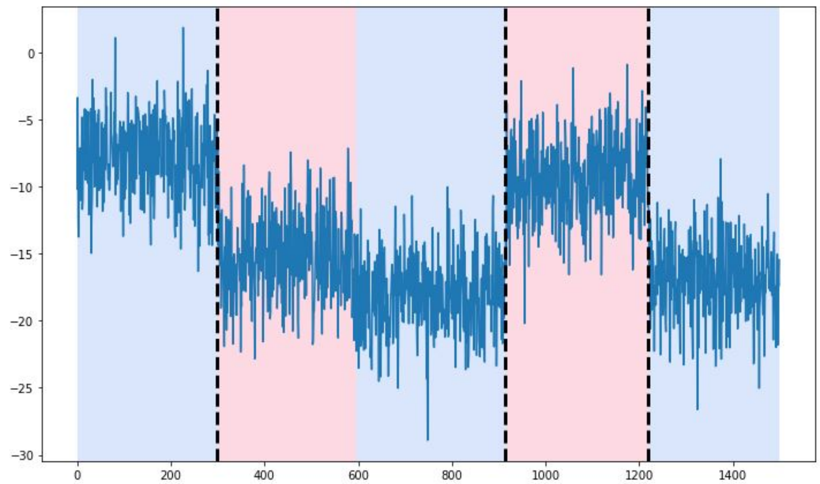

Figure 3.5: Output of Binary Segmentation When Change Points are Unknown and penalty is high

```
my_bkps = algo.predict(pen=2)
rpt.show.display(signal, bkps, my_bkps, figsize=(10, 6))
plt.show()
```

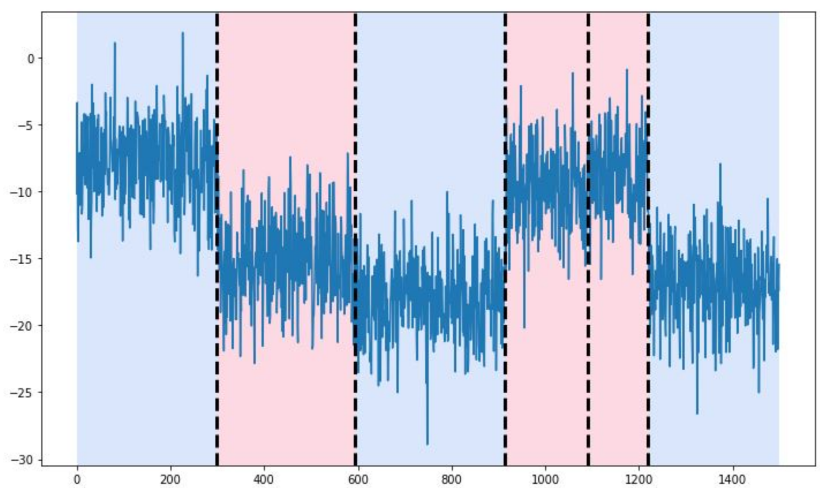

Figure 3.6: Output of Binary Segmentation When Change Points are Unknown and penalty is low

of change point remains only equal to K. At every step, number of possible change points are calculated using the dissimilarity function.

$$d(y_{a..t}, y_{t..b}) = c(y_{a..b}) - c(y_{a..t}) - c(y_{t..b}) \quad (1 \leq a < t < b \leq T):$$

Lowest dissimilarity change points are remocved means that the segments that change point separated are merged. Bottom Up segmentation algorithm is shown in the following figure.

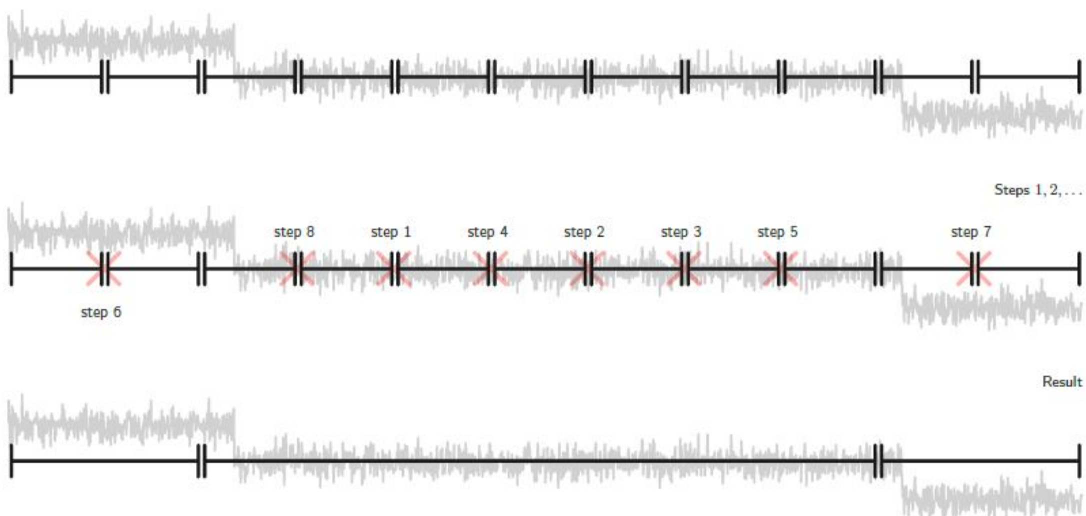

Figure 3.7: Schematic example of Bottom Up Segmentation

3.1.3 Dirichlet Change Point Detection Algorithm and its Variants

This was based on Dirichlet distribution. The key idea of this algorithm is to use Dirichlet parameterization for multivariate compositional data. Let $A = x_i : i = 1, \ldots, T$ with each $x_i = (x_{i1}, \ldots, x_{id})$ satisfying:

$$\sum_{j=1}^{j=d} x_{ij} = 1$$

and

$$x_{ij} \geq 0 \text{ for all } j = 1, \ldots, d$$

The problem of identifying a single change point in A can now be formulated as a hypothesis testing problem as follows:

$$H_0 : \text{The data in A comes from a single Dirichlet distribution}$$

H_τ: The data in A comes from two Dirichlet distribution separated at some τ where $1 < \tau < T$

Under H_0, the best parameter η_0 is obtained by maximizing the log likelihood of the entire data. Let LL_0 denote the log likelihood of A given parameters η_0. Under H_τ, the data is split into two parts for each τ: $L_\tau = x_i : 1 \leq i \leq \tau$ and $R_\tau = x_i : \tau + 1 \leq i \leq T$. . Parameters η_L and η_R are obtained for the left and right data by performing Dirichlet MLE estimations for both L_τ and R_τ. For each τ, the optimal log likelihood is denoted by LL_τ. The τ that maximizes the log likelihood of the data is chosen as the candidate change point.

$$\tau = \arg\max_\tau LL_\tau$$

where LL_τ is given by,

$$LL_\tau = ln[p(x_1, \ldots, x_\tau|\eta_L)] + ln[p(x_{\tau+1}, \ldots, x_T|\eta_R)]$$

τ is accepted as a change point if $z_\tau = LL_\tau - LL_0$ is greater than a threshold. The authors suggest using a random subset test, which is a faster version of the permutation test, to choose an appropriate value of the threshold.

Dirichlet Change Point Detection using Component wise Confidence Intervals (DCP-CCI)[4]

In the first approach, we compute confidence intervals for each of the data components by viewing each component as beta distributed random variable. Note that if the Dirichlet distribution parameters are $\alpha_1, \alpha_2, \ldots, \alpha_K$, then the marginal distribution of the i-th component is $Beta(\alpha_i, \sum_{j=1, j\neq i}^{K} \alpha_j)$. The Dirichlet distribution parameters estimated by maximizing the log-likelihood of a previously observed data segment. Let $\tilde{\alpha_1}, \tilde{\alpha_2}, \ldots \tilde{\alpha_K}$ denote these estimated Dirichlet parameters. The expectation of the i-th component of the estimated Dirichlet distribution is equal to $\tilde{\alpha_i} = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j}$. The confidence intervals for each component are computed by numerically solving the following equations for the edge lengths z_i:

$$\int_{\tilde{\alpha_i} - z_i/2}^{\tilde{\alpha_i} + z_i/2} f(x, \tilde{\alpha_i}, \sum_{1 \leq j \leq K, j \neq i} \tilde{\alpha_j})\, dx = 1 - p$$

$$\text{for } i = 1, 2, \ldots K$$

Here, $f(x, \tilde{\alpha}_i, \sum_{1 \leq j \leq K, j \neq i} \tilde{\alpha}_j)$ denotes the pdf of $Beta$ distribution $(\alpha_i, \sum_{j=1, j \neq i}^{K} \alpha_j)$ This algorithm uses a moving window of size w_2 which is moved one step with each data sample. The data inside the window is averaged. If n_c or more number of components of the averaged data vector lie outside their respective confidence intervals, then a change is declared.

1: $t \leftarrow 0$
2: Use data $x^{(t)}, x^{(t+1)}, \cdots, x^{(t+w_1-1)}$ to compute MLE estimates of the Dirichlet parameters: $\tilde{\boldsymbol{\alpha}}$
3: $t \leftarrow t + w_1$
4: For each component i, compute $\tilde{\alpha}_i^-, \tilde{\alpha}_i^+$ such that the interval $(\tilde{\alpha}_i^-, \tilde{\alpha}_i^+)$ contains $100(1-p)\%$ of the distribution Beta$\left(\tilde{\alpha}_i, \sum_{j \neq i} \tilde{\alpha}_j\right)$.
5: **while** True **do**
6: $\quad y^{(t)} = \frac{1}{w_2} \sum_{i=1}^{w_2} x^{(t-i+1)}$
7: $\quad m \leftarrow 0$
8: $\quad$ **for** $i = 1, 2, 3, 4$ **do**
9: $\quad\quad$ **if** $y_i^{(t)} \notin (\tilde{\alpha}_i^-, \tilde{\alpha}_i^+)$ **then**
10: $\quad\quad\quad m \leftarrow m + 1$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad$ **if** $m \geq n_c$ **then**
14: $\quad\quad$ Change is Declared.
15: $\quad\quad$ Goto 2.
16: $\quad$ **end if**
17: $\quad t \leftarrow t + 1$
18: **end while**

Figure 3.8: Pseudo Code for DCP-CCI

Dirichlet Change Point Detection using Confidence Sets (DCP-CS)[4]

The second approach involves finding a high probability cube centered at the mean of the Dirichlet distribution. This algorithm too maintains a moving window of size w_2 which moves one step after receiving a new data point. If all data points within the window lie outside the confidence cube, then change is declared. Let $(\tilde{\alpha_1}, \tilde{\alpha_2}, \ldots, \tilde{\alpha_K})$ denote the estimated Dirichlet distribution parameters and set $\tilde{\alpha_i} = \frac{\tilde{\alpha_i}}{\sum_{j=1}^{K} \tilde{\alpha_j}}$. Then, the confidence cube is computed over $f(x_1, x_2, \ldots, x_{K1}, \tilde{\alpha_1}, \tilde{\alpha_2}, \ldots, \tilde{\alpha_K})$ by finding a positive real number z, where $f(x_1, x_2, \ldots, x_{K1}, \tilde{\alpha_1}, \tilde{\alpha_2}, \ldots, \tilde{\alpha_K})$ is probability density function of Dirichlet distribution with parameters $(\tilde{\alpha_1}, \tilde{\alpha_2}, \ldots, \tilde{\alpha_K})$. To find such a z, one can use numerical root finding methods such as the bisection method or the secant method.

1: $t \leftarrow 0$
2: Use data $x^{(t)}, x^{(t+1)}, \cdots, x^{(t+w_1-1)}$ to compute MLE estimates of the Dirichlet parameters: $\tilde{\boldsymbol{\alpha}}$
3: $t \leftarrow t + w_1$
4: Compute the mean $\bar{\boldsymbol{\alpha}}$ of the distribution $\mathrm{Dir}(\tilde{\boldsymbol{\alpha}})$.
5: Find l such that the cube with side length l centered at $\bar{\boldsymbol{\alpha}}$ contains $100(1-p)\%$ of the distribution $\mathrm{Dir}(\tilde{\boldsymbol{\alpha}})$.
6: **while** True **do**
7:     **if** all $x^{(t-w_2+1)}, x^{(t-w_2+2)}, \cdots, x^{(t)}$ lie outside the l sided confidence cube centered at $\bar{\boldsymbol{\alpha}}$ **then**
8:         Change is declared.
9:         Goto 2.
10:     **end if**
11:     $t \leftarrow t + 1$
12: **end while**

Figure 3.9: Pseudo Code for DCP-CS

3.2 Online Methods

3.2.1 Cumulative Sum(CUSUM) Control Charts in Statistical Quality Control

The CUSUM-chart is utilized to observe mean of a process dependent on samples taken from the process at various occasions (it are often hours, shifts, days, weeks, months, etc.). Readings of the samples at a given instant presents a 'subgroup'. In place of examining mean of each subgroup, CUSUM chart examine the buildup of information of current and former samples. CUSUM-chart generally flags the process as out of control whenever upside or downside drift of cumulative total crosses the threshold. An alarm is flagged whenever out of control process is observed on the CUSUM chart.

Input signal is given in the first figure as shown below. Threshold to flag

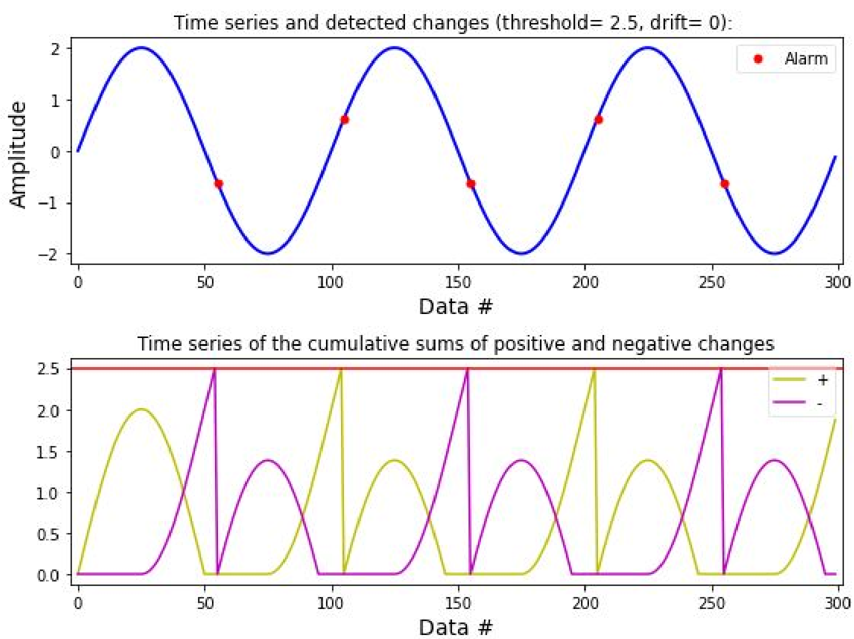

Figure 3.10: Application of CuSum Change Point detection

alarm is fixed at 2.5 units.

Second figure represents the threshold line and the time at which the accumulation is exceeds the threshold line.

Algorithm 1: CUSUM Algorithm

Initialization
Input data: x
$gp_0 = 0$
$gn_0 = 0$
drift = v
while *data* **do**
    $s_t = x_t - x_{t-1}$
    $gp_t = g_{t-1} + s_t - v$
    $gn_t = g_{t-1} - s_t - v$
    if $gp_t < 0$ **then**
        $gp_t = 0$;
        $t_{ap} =$ t;
    end
    if $gn_t < 0$ **then**
        $gn_t = 0$;
        $t_{an} = t$;
    end
    if $gn_t > h$ *or* $gp_t > h$ **then**
        $gn_t = 0$;
        $gp_t = 0$;
        Raise Alarm;
    end
end
Result: Alarm Points (t_a), and Estimated Change Times (k)

3.2.2 Bayesian Online Change-Point Detection

Bayesian change point detection algorithm is online CPD detection algorithm. It doesn't focus on segmentation instead it use concentrate on causative filtering; generating a distribution of the upcoming unseen point within sequence, and on the basis of sole information of already observed data.
Let us consider data-points are in given sequence $x_1, x_2, ..., x_T$ could be divided into non-overlapping parts. The delineations between parts is called as change points. let us further consider that for each part , the data-point within it are i.i.d. from some probability distribution $P(x_t|\eta_p)$. Here the parameters η_p, p = 1, 2, . . . are taken to be i.i.d. as well. let us represent the contiguous set of observations between time a and b inclusive as $x_{a:b}$. The discrete a priori probability distribution over the interval between change points will be denoted as $P_{gap(g)}$. Bayesian approach approximate the posterior distribution over the present "run length," or time since the last change point was detected, given the data point observed so far. let us represent length of present run at instant t by r_t. The notation x_t^r to represent set of data point observations associated with the run r_t. Value of r could be zero, x^r set could be blank. Example of connection of the run length r and some hypothetical uni-variate data in shown in the following figures
Figure shows some uni-variate data which is separated by change points on the mean into 3 segments of lengths $g_1 = 4$, $g_2 = 6$, and an undetermined length g_3. Second Figure (b) shows the length of run r_t which is function of time. r_t touches zero when a distribution changes.

Recursive Run Length Estimation

First on the given run length r_t we have to compute the predictive distribution conditional. Next integration over the posterior distribution on the present run length to find the marginal predictive distribution.
$P(x_{t+1}|x_{1:t}) = \sum_{r_t} P(x_{t+1}|r_t, x_t^r)P(r_t|x_{1:t})$
For posterior distribution calculation,
$P(r_t|x_{1:t}) = \dfrac{P(r_t, x_{1:t})}{P(x_{1:t})}$

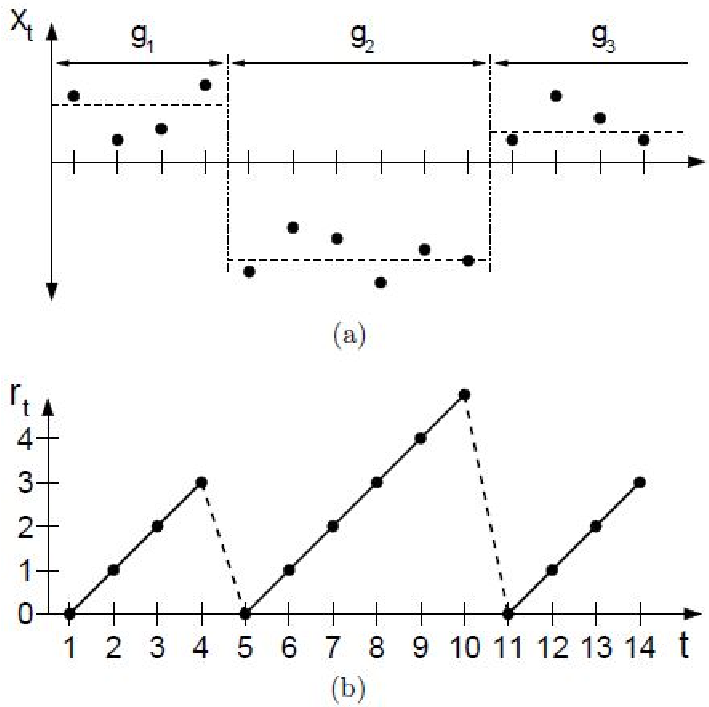

Figure 3.11: Description of a changepoint model expressed in terms of run lengths

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

It is observed that Predictive distribution $P(x_t|r_{t-1}, x_{1:t})$ mostly depends on most the most recent data x_t^r. And recursive message-passing algorithm will be generated a for joint distribution over the present run length and the data, supported on 2 calculations: 1) the prior over r_t given r_{t-1}, and 2) the predictive distribution over the newly-observed datum, given the data since the last change point.

The Change Point Prior

Conditional prior of the change point gives $P(r_t|r_{t-1})$, this algorithm is computational efficient because it has non zero mass at only 2 outcomes: the run length either continues to grow and $r_t = r_{t-1} + 1$ or a change point occurs and $r_t = 0$.

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

Hazard function represented by H(τ):
$H(\tau) = \dfrac{P_{gap}(g=\tau)}{\sum_{t=\tau}^{\infty} P_{gap}(g=\tau))}$
Here $P_{gap}(g)$ is a discrete exponential (geometric) distribution with timescale λ, this process is memory-less and hazard function is constant at $H(\tau) = 1/\lambda$

Computational Experiment

Randomly generated dataset having normal distribution are shown in first Figure shows . The CPD algorithm run on these data points using a univariate Gaussian model.
Log probability over the current run length at each time step are shown in second figure. We use grey-scale color-map, black is 0, white 1 to plot the distributions. Furthermore, we additionally plot the likelihood at each time venture for a sequence length of 0, i.e. the likelihood of the current time step to be a change point.
Likelihood of current information highlight be the change point is appeared in the third figure, the spike long shows that it is change point. The rate of the discrete exponential prior, λ_{gap}, was 250.

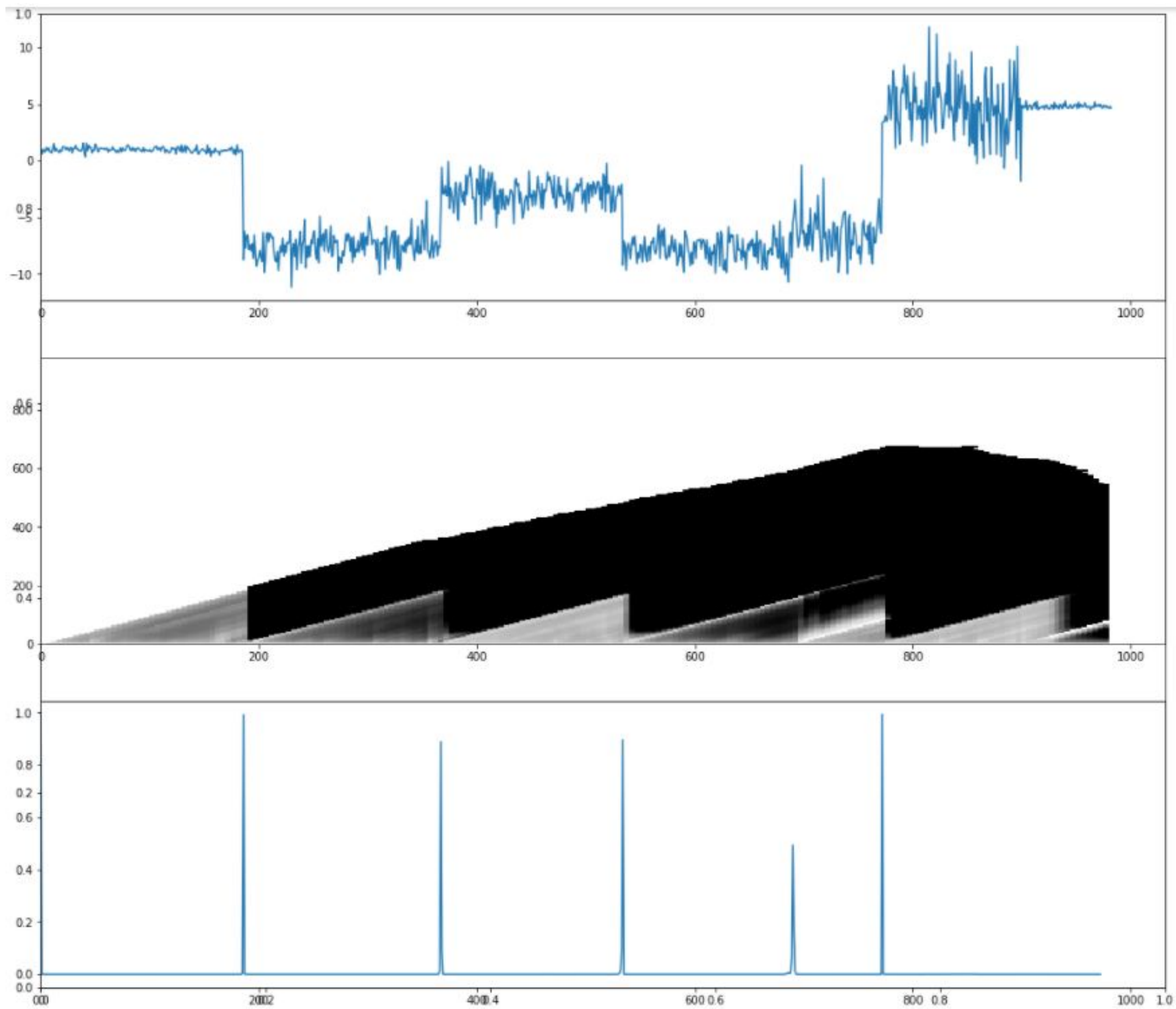

Figure 3.12: Output of Bayesian Algorithm

Chapter 4

Different Constraints in Change Point Detection

Constraints in change point detection is component of change detection methods. These are used when the number of change points is unknown.

4.1 Linear penalty

Linear penalty (which is $l0$ penalty) is most common and popular choice of penalty. It is a general form of several well-known penalty criteria such as the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC). The linear penalty, denoted pen_{l_0}.

$$pen_{l_0}(T) = \beta \mid T \mid \tag{4.1}$$

where $\beta > 0$ is known as smoothing parameter. It controls trade-off between complexity and goodness-of-fit. Low values of β favour partition with many pattern and high values of β discard change points.

How to choose the value of β: From practical point of view, when the cost function is chosen, the solitary parameter to evaluate is the smoothing parameter. This can be checked on the validation data and the best performing β can be chosen.

4.1.1 Related penalties

Some exceptional cases of the linear penalty pen_{l_0} are given below.

pen_{BIC} **penalty function**

$$pen_{BIC}(T) = \frac{p}{2} \log T \mid T \mid \tag{4.2}$$

where $p >= 1$ is dimension of parameter space.

pen_{BIC,L_2} penalty function

Model of an univariate Gaussian signal, with fixed variance σ^2 and piecewise constant mean, the penalty pen_{BIC} becomes pen_{L2}.

$$pen_{BIC,L_2}(T) = \sigma^2 logT \mid T \mid \tag{4.3}$$

where σ^2 is the standard deviation and T is the number of samples.

pen_{AIC,L_2} penalty function

In the same setting, AIC, which is a generalization of Mallows' C_p, also yields a linear penalty, namely pen_{AIC,L_2}.

$$pen_{AIC,L_2}(T) = \sigma^2 \mid T \mid \tag{4.4}$$

4.2 Fused Lasso

For the special case where the cost function is c_{L_2} , a faster alternative to pen_{l_0} can be used. To that end, the l_0 penalty is relaxed to a l_1 penalty.

4.2.1 pen_{l_2} penalty function

$$pen_{l_1}(T) = \beta \sum_{k=1}^{|T|} || \, y_{t_{k-1}....t_k} - y_{t_k....t_{k+1}} \, ||_1 \tag{4.5}$$

where $\beta > 0$ is the smoothing parameter, the t_k are the elements of T and $y_{t_{k-1}....t_k}$ is the empirical mean of sub-signal $y_{t_k....t_{k+1}}$.

4.3 Complex penalties

Some other penalties are more complex. Computing the optimal partition with K change points, with K = 1,2,3....K_{max} for a sufficiently large K_{max}, and use that value of K that minimizes the penalized sum of costs.

4.3.1 pen_{mBIC} penalty function

$$pen_{mBIC}(T) = 3 \mid T \mid \log T + \sum_{k=1}^{|T|+1} \log \frac{t_{k+1} - t_k}{T} \tag{4.6}$$

where the t_k are the elements of T .

4.3.2 pen_{Leb} penalty function

$$pen_{Leb}(T) = \frac{\mid T \mid +1}{T} \sigma^2 (a_1 \log \frac{\mid T \mid +1}{T} + a_2) \tag{4.7}$$

where a_1, a_2 are positive parameters and σ^2 is the variance of noise.

Chapter 5

Performance Measures Of CPD Methods

5.1 Scalability

Scalability is associate degree attribute describing the flexibility of a method or the capability to be modified in size. Real world data from sources for example, human exercises and remote sensing satellites are getting larger in both number of data points and range of dimensions.Change detection methods has got to be computationally effcient in order that it are often scaled to large data sizes which too with larger dimensions.
It is vital to check the process price of the algorithms, whether the algorithm is parametric or non-parametric. As per the literature, non-parametric approaches have incontestable bigger success for giant data-sets. for the most part, the computational expense of parametric strategies is above non-parametric methodologies.

5.2 Performance Evaluation

In order to match different CPD algorithms, performance measures are required. The output of CPD algorithms will contain the following:

- Change point *yes*/*no* decisions
- Change point detection with different levels of precision (i.e., the change point happened within x time units. This sort of algorithm utilizes a multi-class classifier or unsupervised learning approaches).

- The time of the next change point (or the times of all change points in the series)

In case of the first two types of output described above, performance in standard classification task may be used. It can be evaluated using confusion matrix which summarizes the result related to actual and predicted classes.

	Classified as change point	Classified as non-change point
True change point	TP	FN
True non-change point	FP	TN

Here FP = False Positive , FN = False Negative , TP = True Positive , TN = True Negative

Similar metrics are often utilized in in the context of multi-class classification as well.

- **Accuracy**
  It is calculated as the proportion of correctly identified change points to total data points. This measure algorithm's performance.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Sensitivity**
  It is also called as the true positive rate (TPR). This refers to the proportion of a Change Points that are recognized correctly.

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

- **G Mean**
  G-mean utilizes both Sensitivity and Specificity to measure the performance of the algorithm.

$$Gmean = \sqrt{\left(\frac{TP}{TP + FN}\right) * \left(\frac{TN}{TN + FP}\right)}$$

- **Precision**
  This is computed as the proportion of true positives to total data points that are labeled as change points.

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

For the third case, where time of change point is predicted, then classification metrics are not appropriate. The following metrics are useful in such contexts.

- **Mean absolute error (MAE)**
  This directly quantify the proximity of the predicted CP and actual CP. The absolute value of the difference between the predicted and actual CP time is added and standardize.

$$MAE = \frac{\sum_{i=1}^{|CP|} |Prediction(CP) - Actual(CP)|}{|CP|}$$

- **Mean-squared error (MSE)**
  Here, the resulting measure is very sensitive to outliers.

$$MSE = \frac{\sum_{i=1}^{|CP|} (Prediction(CP) - Actual(CP))^2}{|CP|}$$

- **Mean Signed Difference (MSD)**
  This measure examine the direction of the error (predicting before or after the actual CP time).

$$MSE = \frac{\sum_{i=1}^{|CP|} (Prediction(CP) - Actual(CP))}{|CP|}$$

- **Root-mean-squared error (RMSE)**
  This is simply the squared root of MSE.

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^{|CP|} (Prediction(CP) - Actual(CP))^2}{|CP|}\right)}$$

Chapter 6

Different Applications of CPD

Here, we put forward some real life instance where CPD can be utilize. It will help us to realize how crucial it is now a days. Historically, change-point detection was first used in quality control [Shewhart, 1931], to monitor some processes that may be either in control or out of control.

6.1 Analysis of Images

Researchers collect image data over a certain time period or they may use video data, for video-based monitoring as well as surveillance. CPD can help in detection of security breaches and different other abrupt activities.

6.2 Climate Change Detection

Over the previous couple of decades, climate monitoring, analysis, and prediction methods that utilize change point detection have got crucial regarding recognizing temperature change and identifying the ascent of ozone harming substances in air.

6.3 Analysis of Human Activities

Detecting Change Points supported on the characteristics observed using sensors from smart homes, smart cities, IoT and mobile devices are often formulated as a change point detection problem. It helps in recognizing changes in conduct of resident that give bits of knowledge of wellbeing standing.

6.4 Speech Recognition

Speech recognition addresses the technique for changing spoken discourse expressions to words or text. Change point detection techniques are applied here for audio segmentation and recognize limits between quietness, sentences, words, and commotion.

6.5 Medical Condition Monitoring

Observing of a patient's wellbeing includes analysis of physiological factors like electroencephalogram (EEG), pulse and electrocardiogram (ECG). It is necessary in order to perform machine driven, actual time monitoring of human health. CPD can be used to discover sleep problems, epilepsy, magnetic resonance imaging (MRI) and for better comprehension of cerebrum exercises.

Chapter 7

Conclusion

In this Project, we briefly presented different models in change point detection, compared and analyzed their Performance and summarized different areas of application for change point detection. It is important to know that some of the CPD models are matches for online detection of a change point, while some are more suited in offline setup. In some certifiable applications, change points got to be detected as soon as possible and few systems require timely actions, hence discovering the change focuses quick is essential. Anytime algorithms can conceivably be used to utilized for algorithm delays and adjust the computational time in balance with the standard of the detected change points. In practise, some heuristics involving smaller window sizes may be employed to calculate change point scores, e.g. Bayesian methods. For nearly all the methods, change point detection depends on the window size. The smaller window size is better for speed but it might detect more local changes compared to large windows and it cannot look beyond the data and that leads to increase of cost. Incorporating variable window sizes could also be helpful in combining the trade-off between this two.

One major concern regarding CPD algorithms are its robustness. Generally in practise, non-parametric methods are more robust than parametric ones. Notwithstanding, there is no hypothetical evidence of this in writing. Also, the algorithms need to be scalable, accurate and real time efficient.

Future Work: In many real-world problems, the problem of change point detection by itself may be sub-problem instead of the actual problem of interest. A worldwide environmental change scientist could also be interested in finding degree of change in temperature and humidity than simply distinguishing that a change happened. The overall writing for the most part discusses discovering the change point and not the level of progress. We accept this is a pleasant region for future work.

Bibliography

[1] Adams, R. P. and MacKay, D. J. (2007). Bayesian online changepoint detection. arXivpreprint arXiv:0710.3742 .

[2] Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. Signal Processing, 167

[3] Damien Garreau.(2017). Change-point detection and kernel methods. PhD thesis. PSL Research Univer-sity, 2017.

[4] Shobhit Bhatnagar, Single and Multi-Agent Reinforcement Learning in Changing Environments. Master's thesis, IIT Bombay, 2020