A Meta-Analysis of Metrics for Change Point Detection Algorithms

DRAFT v1.0

Matt Chapman
matthew.chapman@student.uva.nl

Spring 2017, 58 pages

Supervisor: Evangelos Kanoulas, Universiteit van Amsterdam
Host organisation: Buzzcapture B.V., http://www.buzzcapture.com

Universiteit van Amsterdam
Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Master Software Engineering
http://www.software-engineering-amsterdam.nl

Contents

Abstract 3

Acknowledgments 4

1 Problem Statement & Motivation 5
- 1.1 Problem Statement . . . 5
- 1.2 Motivation . . . 6

2 Related Works 7

3 Background & Context 8
- 3.1 Change Detection Algorithms . . . 8
  - 3.1.1 Pruned Exact Linear Time . . . 8
  - 3.1.2 Segment Neighbourhoods . . . 9
  - 3.1.3 Binary Segmentation . . . 10
- 3.2 Algorithm Configuration . . . 10
  - 3.2.1 Penalty Scores . . . 10
  - 3.2.2 Distribution Assumptions . . . 12
- 3.3 Evaluation Measures . . . 12
  - 3.3.1 F1 Score . . . 12
  - 3.3.2 Rand Index . . . 13
  - 3.3.3 Adjusted Rand Index . . . 14
  - 3.3.4 BCubed . . . 14

4 Research Method 16
- 4.1 Introduction . . . 16
- 4.2 Evaluation Pipeline Construction . . . 16
  - 4.2.1 Calculation of Changepoint Locations . . . 16
  - 4.2.2 Calculation of Evaluation Measures . . . 17
- 4.3 Measures Meta-Analysis . . . 17
  - 4.3.1 Experiment Listings . . . 18
- 4.4 Comparison of Measures Based Upon Functional Requirements . . . 20
- 4.5 Application of Algorithms to Real World Data . . . 20
  - 4.5.1 Data Preparation . . . 20
  - 4.5.2 Execution . . . 21

5 Research 22
- 5.1 Introduction . . . 22
- 5.2 Simulation Studies . . . 22
  - 5.2.1 Dependence on Sample Size . . . 22
  - 5.2.2 Impact of Data Preceding and Following a Known Change Point . . . 23
  - 5.2.3 Temporal Penalty . . . 24
  - 5.2.4 Penalisation for False Positives . . . 25
  - 5.2.5 Penalisation for False Negatives . . . 25
  - 5.2.6 Number of Change Points in Fixed Length Data Stream . . . 26

5.2.7 Number of Change Points in Variable Length Data Stream . . . 26
5.3 Functional Requirements . . . 27
5.4 Real-World Data Analysis . . . 27

6 Results **29**
6.1 Metric Behaviour . . . 29
6.1.1 F1 Score . . . 29
6.1.2 Rand Index . . . 30
6.1.3 Adjusted Rand Index . . . 30
6.1.4 BCubed F-Score . . . 30
6.2 Fulfilment of Functional Requirements . . . 31
6.3 Real-World Data Analysis . . . 31

7 Analysis & Conclusions **33**
7.1 Conclusions . . . 33
7.2 The Ideal Metric . . . 34
7.3 Threats to Validity . . . 35

8 Future Work **36**

APPENDICES **40**

A Pseudocode for Simulation Studies **41**

B Ground Truth Annotations **44**

C Real-World Change Point Detection Plots **45**

D Full Real World Data Scorings **51**

E Bibliography Annotations **54**

Abstract

Change point detection is a highly complex field that is hugely important for industries ranging from manufacturing and IT infrastructure fault detection, to online reputation management. There exists in this field a number of metrics or scores that are widely used for proving the veracity of new or novel approaches. However, there is little consensus as to which measure(s) are most effective in this field. This thesis carries out research into the field of change point detection, carrying out a comparative study of the behaviours of various scoring metrics using simulated data, the accuracy of change point detection methods using real world data, and makes a recommendation for the approach to be used for change point detection in the production systems of an online reputation management company. Finally, a short discussion on the ideal properties of a scoring metric from change point detection algorithms is carried out.

In this study, it is found that many of the established scoring metrics behave inconsistently when applied to change point detection problems, and exhibit properties that bring their usefulness and accuracy into question. It is also found that between certain metrics, there is no correlation or agreement in how algorithms are ranked according to score value.

Concluding, the study also shows that existing change point detection methods are perhaps not the most well suited methods for the use-case and requirements of the host organisation.

Acknowledgements

This thesis was only possible due to the contributions and assistance of a number of people. First thanks goes to Evangelos Kanoulas, the academic supervisor for this work. He always kept me on track, and was full of suggestions and insights into the data and experiments I was running. Without him, I have no doubt that at the time I am writing this, this thesis would still be in the planning stages.

Special thanks also go to Wouter Koot & Carina Dusseldorp at Buzzcapture. Wouter acted as my industry supervisor, and like Evangelos, was always on hand for support and suggestions. I very much look forward to continuing to work with and for him at Buzzcapture, following my graduation. Carina, as Head of Research at Buzzcapture, provided the data annotations that were invaluable for the smooth running of the experiments carried out on real world data. Without her insight into the data sets that I collated, the experiments on real world data would not have been possible.

I would also like to credit my friends Arend and Jetse, who have been excellent colleagues and friends throughout the duration of this course. Special thanks also goes to the erstwhile members of the 'Beerability' WhatsApp group, set up as the informal place for our course to whinge and moan about our work. It was the one place that I could go and not feel bad about my progress.

Finally, the most important thanks go to my girlfriend, Elisabeth. She is the one that put up with the sleepless nights, the whining about workload and the days I shut myself away to finish this work. Without her, I'm not sure I would have ever had the confidence to apply to do a Masters degree in the first place.

Chapter 1

Problem Statement & Motivation

1.1 Problem Statement

When asserting the veracity of an approach to a particular problem in software engineering, this is almost always achieved by utilising some sort of metric or measure. To give an example, when evaluating the quality of some new piece of software, this could be done through calculating measures such as McCabe Complexity [35] or unit test coverage.

The problem of detecting change points in a data stream is no different. There exists within this field a number of metrics that may be used to prove the accuracy and effectiveness of a given approach, and these often fall into one of three categories: binary classification (e.g. F1 score), clustering (e.g. BCubed, or the Rand Index), or information retrieval (utilising scoring systems such as those utilised in the TREC 2016 Real Time Summarisation Track: [47]).

Being that there are myriad ways to detect change points in a data stream, it is necessary to implement the application of some evaluation measure to back up an assertion that, for example, approach A is better than approach B for some data set y_s. The application of these measures can, in some cases, result in more questions than answers, especially in situations where two different measures may disagree with the results of the aforementioned approaches A and B.

However, for the number of change point detection methods that exist, there is an almost equal number of ways to evaluate the results of the methods. There is little agreement between researchers on the 'correct' method to use, and as this research will show, there are problems that exist when utilising measures that have generally been widely accepted for this purpose.

Outside of the 'normal' applications of change point detection (for example, spotting the onset of 'storm seasons' in oceanographic data [30] or the detection of past changes in the valuation of a currency, such as BitCoin [9], there is also an application for change point detection as an 'event detection' mechanism, when applied to data such as that sourced from online social media platforms.

There exists methods of event detection in social media data utilising approaches such as term frequency counting, lexicon-based recognition and context-free-grammar algorithms. However, these approaches rely on (sometimes computationally expensive) analysis of the content of messages being posted on social media platforms. Being that change point detection (and especially *online* change point detection) algorithms can be utilised for the detection of past events, it stands to reason that these algorithms can also be utilised for event detection when applied to pre-computed data such as conversation volume or reach of a particular conversation. There certainly exists a requirement in the field of online reputation management to be able to inform businesses (in a timely manner) that a spike in conversation volume is occurring, and thus they may need to carry out some action to mitigate reputational damage if the conversation sentiment is negative.

Therefore, this thesis intends to answer the following research question:

RQ How can changes be detected in conversation volume data from social media?

From this research question, the following sub-questions have been formulated:

SQ1 Are existing change point detection algorithms effective at detecting changes/events in social

media data in a timely fashion?

SQ1.1 Which algorithm, out of those being analysed, performs the 'best' when applied against real world data?

SQ2 Are measures not specifically defined for the purpose of change detection evaluation effective in this domain?

SQ2.1 In what way do established measures fail to behave correctly?

SQ2.2 Which evaluation measure provides the best evaluation, given the functional requirements set forth by the host organisation?

SQ3 If existing measures are deficient in some way, what would an 'ideal' measure look like?

1.2 Motivation

This particular research is motivated specifically by the online reputation management sector. The business hosting this research project (Buzzcapture b.v.[1]) is a Dutch consultancy that provides online reputation management services to other businesses throughout Europe. Chief among these services is the BrandMonitor application, which, among other features, provides a rudimentary notification system for clients that is triggered once there is an absolute increase in conversation volume (conversation volume being defined as the number of tweets relevant to the client over a given time period).

Buzzcapture made a project available to students of the Universiteit van Amsterdam, wherein they would provide a method for more effectively providing these notifications, based on more than just an arbitrary threshold on conversation volume or some other computed metric. Upon accepting this project, research was carried out into the field of change detection algorithms, during which it was found that there was not a *single* accepted approach for evaluating measures. Indeed, for every publication that described some novel change point detection algorithm, there was a slightly different approach for evaluating it, and proving the veracity of algorithm in a certain situation. Most publications made use of some sort of binary classification measure (for example, [38], [9], [37]), while others had small variations on that theme, providing additional takes on the binary classification approach using methods such as *Receiver Operating Characteristic* curves (for example [15] & [12]). Additionally, there were publications that made use of clustering measures, calculated using a segmentation of the time series based on computed change points, such as [34].

It is the intention of this thesis to not only answer the research questions set out in § 1.1, but also to provide a robust recommendation and working prototype of a change detection methodology which could then eventually be implemented into the Brand Monitor tool to supply timely and relevant notifications to clients when a conversation concerning their brand exhibits a change in behaviour or otherwise 'goes viral'.

[1] http://www.buzzcapture.com

Chapter 2

Related Works

TODO: coherent related works story

Chapter 3

Background & Context

3.1 Change Detection Algorithms

Change detection first came about as a quality control measure in manufacturing, and methods within this domain are generally referred to as *control charts*. Since the inception of approaches such as CUSUM [36] that provide the possibility for on-line evaluation of continuous data streams, change detection has grown as a field. With applications such as epidemic detection, online reputation management and infrastructure error detection, change detection is hugely useful both as an academic problem and in production systems of myriad application.

This thesis work utilises three different change point detection methods, discussed forthwith.

3.1.1 Pruned Exact Linear Time

Pruned Exact Linear Time (henceforth referred to as *PELT*) is a modern change point detection method proposed by Killick, Fearnhead, and Eckley in "Optimal detection of changepoints with a linear computational cost", in 2012 [28].

PELT is an *exact* method of computing change points, based on segmentation of data streams, in the same manner as Segment Neighbourhood and Binary Segmentation. PELT is a modification of the *Optimal Partitioning* method proposed by Brad Jackson et. al in 2003, in [23].

Jackson's method has a computational cost of $O(n^2)$, while Killick's PELT method improves on this with a computational complexity of $O(n)$ [28].

The PELT method works similarly to Segment Neighbourhood [4] and Binary Segmentation [23] [51] approaches, in that it attempts to fit a model of segmentation of a given data set, minimising the cost of the segmentation to achieve an optimal distribution of segments and therefore change points.

Algorithm 1 describes a pseudocode implementation of the PELT algorithm, as described in [14]

Algorithm 1: PELT Method for change point detection

Input: A set of data of the form, $(y_1, y_2, \ldots, y_n)$ where $y_i \in \mathbb{R}$.
A measure of fit $C(.)$ dependent on the data.
A penalty constant β which does not depend on the number or location of change points.
A constant K that satisfies $C(y_{(t+1):s}) + C(y_{(s+1):T}) + K \leq C(y_{t+1):T})$

Initialise: Let n = length of data and set $F(0) = -\beta$, $cp(0) = NULL$

for $\tau* = 1, \ldots, n$ **do**
  Calculate $F(\tau*) = min_{\tau \in R_{\tau*}}[F(\tau) + C(y_{\tau+1:\tau*}) + \beta]$
  Let $\tau^1 = arg\Big\{min_{\tau \in R_{\tau*}}[F(\tau) + C(y_{(\tau+1):\tau*}) + \beta]\Big\}$
  Set $cp(\tau*) = [cp(\tau^1), \tau^1]$
  Set $R_{\tau^*+1} = \{\tau \in R_{\tau^*} \cup \{\tau^*\} : F(\tau) + C(y_{\tau+1:\tau^*}) + K \leq F(\tau^*)\}$
end
Output: the change points recorded in $cp(n)$

3.1.2 Segment Neighbourhoods

Proposed by Auger and Lawrence in 1989 [4], Segment Neighbourhood (*SegNeigh*) is another example of an exact segmentation algorithm for the detection of change points in data. SegNeigh utilises dynamic programming to search the segmentation space (defined as the maximum number of change points in a given data stream) and then compute the cost function for every possible segmentation of the data. In this way, the location and number of change points can be computed exactly by taking the segmentation that returns the lowest cost function result.

Segment Neighbourhoods has a computational complexity of $O(n^2)$, significantly higher than that of PELT or Binary Segmentation. However, this additional cost in complexity is offset by improved performance, as shown by Braun, Braun, and Müller in "Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation" [8].

Algorithm 2 describes a pseudocode implementation of the SegNeigh algorithm, as described in [14]

Algorithm 2: Generic Segment Neighbourhoods method for change point detection

Input: A set of data of the form, $(y_1, y_2, \ldots, y_n)$
A measure of fit $R(.)$ dependent on the data which needs to be minimised.
An integer, $M - 1$ specifying the maximum number of change points to find.
Initialise: Let n = length of data.
Calculate $q^1_{i,j} = R(y_{i:j})$ for all $i, j \in [1, n]$ such that $i < j$.

for $m = 2, \ldots, M$ **do**
  for $j \in \{1, 2, \ldots, n\}$ **do**
    Calculate $q^m_{1,j} = min_{v \in [1,j]}(q^{m-1}_{1,v} + q^1_{v+1,j})$
  end
  Set $\tau_{m,1}$ to be the v that minimises $(q^{m-1}_{1,v} + q^1_{v+1,n})$
  for $i \in \{2, 3, \ldots, M\}$ **do**
    Let $\tau_{m,i}$ to be the v that minimises $(q^{m-i-1}_{1,v} + q^1_{v+1,cp_{m,i-1}})$
  end
end
Output: For $m = 1, \ldots, M$: the total measure of fit, $q^m_{1,n}$ for $m - 1$ change points and the location of the change points for that fit, $\tau_{m,1:m}$.

3.1.3 Binary Segmentation

Binary Segmentation [23] [51] is a popular method for change point detection, widely utilised in the field. Binary segmentation is a method that recursively applies a single change point detection method. On the first iteration, if a change point is detected, the data is split around that change point (resulting in two data sets) and the change point method is run again on the two resulting data sets. This process is repeated such that many data set segments are created, and runs until no further change points are detected.

Binary Segmentation is an *approximate* change point detection approach, and returns estimated change point locations - unlike PELT and SegNeigh - which return exact change point locations. It has the same computational cost as PELT, $O(n)$.

Algorithm 3 describes a pseudocode implementation of the BinSeg algorithm, as described in [14]

Algorithm 3: Generic Binary Segmentation method for change point detection

Input: A set of data of the form, $(y_1, y_2, \ldots, y_n)$
A test statistic $\Lambda(.)$ dependent on the data
An estimator of change point position $\hat{\tau}(.)$
A rejection threshold C.

Initialise: Let $C = \emptyset$, and $S = \{[1, n]\}$

```
while S ≠ ∅ do
    Choose an element of S, denote element as [s, t]
    if Λ(y_{s:t}) < C then
        remove [s, t] from S
    end
    if Λ(y_{s:t}) then
        remove [s, t] from S
        calculate r = τ̂(y_{s:t}) + s − 1, and add r to C
        if r ≠ s then
            add [s, r] to S
        end
        if r ≠ t − 1 then
            add [r + 1, t] to S
        end
    end
end
```

Output: the set of change points recorded in C

3.2 Algorithm Configuration

Change detection is an *unbounded* problem. Left without some system of constraint, the algorithm could theoretically run to infinity. Indeed, one of the algorithms utilised in this research, PELT, when left unbounded, will detect every data point in the time-series as a change point. This result is *technically* correct, but not useful for our purposes. For this reason, the algorithms implement a penalty system, allowing for an optimal number of change points to be detected.

3.2.1 Penalty Scores

Penalty scores operate as a mechanism for optimising an unbounded problem such as the one being addressed here. Haynes et al. define the problem as follows [20]: Given time series data points $y_1, \ldots, y_n$, the series will contain m change points such that their locations $\tau_{1:m} = (\tau_1, \ldots, \tau_m)$, where $\{\tau_i \in \mathbb{Z} \mid 1 \leqslant \tau_i \leqslant n-1\}$. τ_0 is assumed to be 0, and τ_{m+1} is assumed to be n. In this way we can state that a given change detection algorithm will split the time series into $m+1$ segments such that segment

i contains the points $y_{(\tau_{i-1}+1):\tau_i} = (y_{\tau_{i-1}+1}, \ldots, y_{\tau_i})$. A *cost function* for a segmentation of the data set y_s can be defined as $C = (y_{s+1:t})$. This cost function defines a cost for a given segmentation containing points $y_{s+1:t}$. To illustrate, the cost function C, in the context of binary classification, returns a value that increases for an 'incorrect' classification and decreases for a 'correct' classification.

At this point it can be stated that the *constrained minimisation problem* is as follows:

$$Q_m(y_{1:n}) = \min_{\tau_{1:m}} \left\{ \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i})] \right\} \tag{3.1}$$

Equation 3.1 is an equation showing the optimisation problem that change point detection presents. If the equation is taken intuitively, it shows that for a dataset with a *known* number of change points, the segmentation of the data around the change points can be effectively estimated by obtaining the segmentation that returns the minimum cost based on the cost function C.

However, the problem being solved by many change detection algorithms involves an unknown number of change points, at unknown locations. In this case we can estimate the following to obtain the number and location of the change points:

$$\min_m \left\{ Q_m(y_{1:n}) + f(m) \right\} \tag{3.2}$$

Equation 3.2 includes a *penalty function* $f(m)$ that increases proportionally with the number of change points m. Essentially, as the segmentation calculated as Q_m increases in size, so does the result of the penalty function $f(m)$. These two elements are at-odds with each other, presenting a problem that requires optimisation to correctly estimate the number and location of change points in the data set y_s. If $f(m)$ increases in a linear fashion with m, we can define a *penalised minimisation problem* in the following manner:

$$Q_m(y_{1:n}, \beta) = \min_{m, \tau_{1:m}} \left\{ \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] \right\} \tag{3.3}$$

In this equation, β is the penalty function defined previously as $f(m)$.

There are a number of established approaches to calculating penalty values for unbounded problems, chiefly among which are Schwarz Information Criterion (SIC)/Bayesian Information Criterion (BIC) [42], Akaike Information Criterion (AIC) [1] and Hannan-Quinn [19]. Of these approaches, it is necessary to experiment to find the scheme that produces the 'correct' number of change points for a given dataset. The penalty schemes are defined as follows:

$$\text{SIC} = \ln(n)k - 2\ln(\hat{L}) \tag{3.4}$$

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{3.5}$$

$$\text{HQC} = -2L_{max} + 2k\ln(\ln(n)) \tag{3.6}$$

Where n is the total number of observations, k is the number of *free parameters*[1] and $\hat{L}$ is the maximum value of a likelihood function.

The penalty function returns a value which is represented by β in Equation 3.3, which then limits the number of possible segmentations returned by Q. In this way, varying the penalty function β can alter the results given by a change detection algorithm by increasing or decreasing the number of change points detected.

The `changepoint` package requires the provision of a penalty function for each algorithm, which has been chosen as 'SIC' for this research. This decision is based upon advice published in [14], in which it is stated that AIC (while popular in the field of change point detection as a penalty term) tends to asymptotically overestimate k.

While the selection of an optimal penalty term for a given data set and change point detection algorithm is an open research question in itself. There exists a method, *Changepoints for a Range of Penalties* (CROPS) [20], in which a change detection algorithm (PELT, in this case) is repeatedly

[1] variables that are unable to be predicted by a given model

executed in such a fashion that allows for the plotting of the number change points detected against the penalty value used. When conducting analysis against a single, static data source (offline analysis), this method can be useful for ensuring optimal penalty term selection.

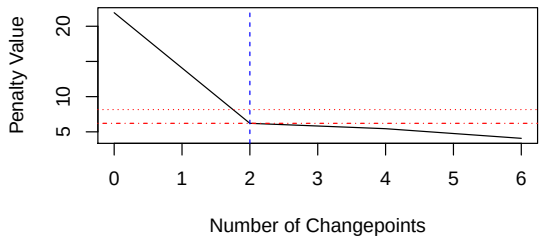

Figure 3.1: CROPS Algorithm Output

Figure 3.1 shows an example of the output of the CROPS algorithm, when executed against the 'Rabobank' data set used in the real world data analysis. To read this output, the optimal penalty value should be chosen such that the plot at that point has begun to level off, showing a fast increase in the number of change points for penalty values lower than this. This example shows an optimal penalty value of about 6 (backed up by the algorithm output of a penalty value of 6.21 for 2 change points - signified by the dash-dot red line on the plot). The penalty value for the same data set computed with SIC is 8.16 (signified by the dotted red line on the plot), which also results in a total of two change points being detected. Because the PELT algorithm has a computational complexity of $O(n)$, if it is necessary to carry out an analysis such as this for a static data set, it would be easy to do so.

However, the usefulness of such a method for application to streaming (online) data sets is questionable.

3.2.2 Distribution Assumptions

It is also important to note that many change point detection methods make an assumption about the underlying distribution of the data being analysed. This distribution can be, for example, a normal distribution, poisson distribution or a gamma distribution. The selection of this distribution can have an effect on the ability of a given algorithm to detect change points.

To limit the scope of this research, an assumption is made that all of the data being analysed follows a normal distribution, to allow for a 'like for like' comparison of results. It is possible for some algorithms to use a CUSUM test statistic that makes no assumptions about data distribution, but this is not supported by all of the methods implemented in `changepoint`. It is also possible to specify a poisson or gamma distribution when utilising the mean/variance test statistic, but this has not been done in this research, again, in order to ensure a 'like for like' comparison of approaches.

3.3 Evaluation Measures

As briefly discussed in the introduction to this thesis, there are a number of pre-existing approaches for the evaluation of change detection methods. Here, the measures being evaluated are briefly explained:

3.3.1 F1 Score

This measure is utilised for testing accuracy in problems of binary classification. It considers two different measures, *precision* and *recall*, and takes a harmonic mean of the two measures to compute the final score. This measure is used in [38], [9], and [37] (among others) for the purposes of evaluating change point detection methods.

To calculate Precision and Recall, a *confusion matrix* such as that in Figure 3.2 is first constructed to provide values for the total number of true positives, false negatives, false positives and true negatives:

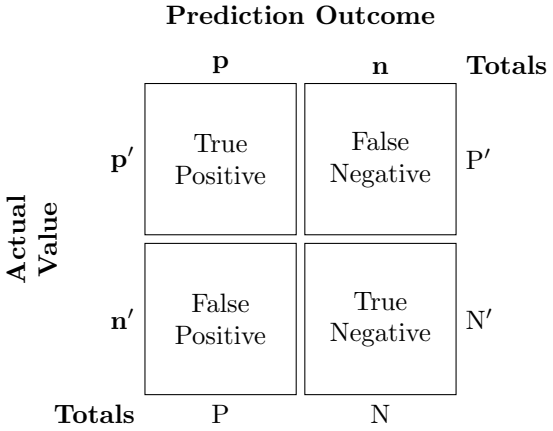

Figure 3.2: Confusion Matrix Example

Precision and Recall were proposed by Kent et al., in "Machine literature searching VIII. Operational criteria for designing information retrieval systems", for the purposes of scoring the effectiveness of information retrieval methods.

Recall is computed as the number of correct positive results, divided by the number of positive results that should have been detected [26]. *Precision* is computed as the number of correct positive results divided by the number of all possible positive results [26]:

$$Precision = \frac{TP}{TP + FP} \tag{3.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.8}$$

The F1 score calculation (the harmonic mean of both recall and precision) can be described in general terms as follows [41]:

$$F_1 = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.9}$$

The F1 score was first proposed by Rijsbergen in his seminal work, *Information retrieval*.

As the F1 score is a binary classification measure, it can only be used to test the precision of an algorithm in a single domain, that is, was the change detected or not. Precision, Recall and F1 Score all provide a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$.

3.3.2 Rand Index

This measure is for computing the similarity between two clusters of data points. It is used for calculating the overall accuracy of a given clustering approach, when compared against a set of ground truth clusters. It was first proposed by Rand in "Objective Criteria for the Evaluation of Clustering Methods". It was used in [34] for the purposes of evaluating change point detection approaches. The Rand Index is defined as [40]:

$$R = \frac{a + b}{a + b + c + d} \tag{3.10}$$

Given a set of data points S, partitioned through two different methods, which we shall refer to as X and Y. Based on this knowledge the following can be defined:

- a = total number of pairs that were partitioned into the *same* subset by both X and Y
- b = total number of pairs that were partitioned into *different* subsets by both X and Y
- c = total number of pairs that were partitioned into the *same* subset by X and into a different subset by Y
- d = total number of pairs that were partitioned into the *same* subset by Y and into a different subset by X

Intuitively, it can be stated that $a+b$ is the total number of agreements between methods X and Y, while $c+d$ is the total number of disagreements. This calculation will return 0 for completely different clusters and 1 for identical clusters. The Rand Index provides a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$

3.3.3 Adjusted Rand Index

Similar to the Rand Index, but adjusted to take into account the random chance of pairs being assigned to the same cluster by both approaches being compared. While the Rand Index is suited for comparing a segmentation method against a known-good *oracle* method, the adjusted index is more suited to comparing two differing approaches [34]. It was first proposed by Hubert and Arabie in "Comparing partitions". It was utilised as a similarity measure for computed clusters created by segmentation of data sets in [34]. It is defined as [22]:

$$R_{adjusted} = \frac{R - R_{expected}}{R_{max} - R_{expected}} \tag{3.11}$$

where $R_{expected}$ is defined as the expected Rand Index score for two completely random classifications of the given data [34] and R_{max} is defined as the maximum Rand Index value - generally 1.

The Adjusted Rand Index provides a score s such that $\{s \in \mathbb{R} \mid -1 \leq s \leq 1\}$. This is different to the score returned by the 'basic' Rand Index, in that it is possible for a negative value to be returned.

3.3.4 BCubed

BCubed is similar to the Adjusted Rand Index, in that it is adjusted to be appropriate for all constraints in a clustering problem. It was developed by Bagga and Baldwin in 1998 [5].

This measure was not found to be in use for the purposes of evaluating change point detection methods, but it was particularly interesting to investigate based on Matteson and James's use of clustering metrics in [34]. It was decided that this measure would be included as an additional well-known clustering measure, to investigate whether it would perform any better than the established Rand and Adjusted Rand indices.

BCubed operates in a similar way to the Recall and Precision classification metrics, in that it computes precision and recall values, before calculating a harmonic mean of the two metrics. BCubed also makes use of an additional *correctness* function defined as follows, where e and e' are equivalent elements in a ground truth and computed clustering respectively, L is a computed label and C is a ground truth category:

$$\text{Correctness}(e, e') = \begin{cases} 1 & \text{iff } L(e) = L(e') \leftrightarrow C(e) = C(e') \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

Equation 3.12 can be stated intuitively as returning 1 if and only if both the ground truth and computed clusterings place the element e into the same cluster, returning 0 in all other cases.

Precision and Recall can then be calculated as follows:

$$\text{Precision} = Avg_e[Avg_{e'.C(e)=C(e')}[Correctness(e, e')]] \tag{3.13}$$

$$\text{Recall} = Avg_e[Avg_{e'.L(e)=L(e')}[Correctness(e, e')]] \tag{3.14}$$

To obtain the F score, a harmonic mean of Precision and Recall is taken in the same manner as the F1 score for binary classification:

$$F_{bc} = 2 \cdot \frac{1}{\frac{1}{Recall_{bc}} + \frac{1}{Precision_{bc}}} \tag{3.15}$$

The BCubed F-Score provides a score s such that $\{s \in \mathbb{R} \mid 0 \leq s \leq 1\}$, much like the Rand Index and F1 score for binary classification.

Chapter 4

Research Method

4.1 Introduction

The purpose of this thesis is twofold: firstly, to conduct a meta analysis of evaluation measures, and how they perform in the domain of change point detection problems. Being that these measures are metrics designed for other problems that are not necessarily change point detection, it stands to reason that there are perhaps situations where families of metrics will disagree with each-other, or disagree with by-eye evaluation by domain experts. The second purpose is to evaluate the veracity of a number of existing change point detection algorithms when they are applied to data from social media.

This thesis consists of two distinct parts:

- Meta-analysis of evaluation measures, fulfilling RQ2 and associated sub-questions.
- Application of change detection approaches to real world data and evaluation based on requirements elicitation and evaluation measures.

4.2 Evaluation Pipeline Construction

This thesis will analyse three different change detection algorithms: *Pruned Exact Linear Time*[28], *Binary Segmentation* [23] and *Segment Neighbourhoods*[4]. These will be referred to as *PELT*, *BinSeg* and *SegNeigh* respectively. The algorithms will be briefly discussed in this thesis, along with the chosen critical values such as *minimum segment length*, *penalty scoring* and *assumed underlying distribution*.

The algorithms will then be applied to a collection of datasets falling into two categories: real-world conversation volume data taken from Twitter, and simulated data generated according to certain constraints which will also be discussed here.

The results provided by each technique will then be evaluated according to the following measures: Precision, Recall, F1 score, Rand Index, Adjusted Rand Index, Bcubed Precision, Bcubed Recall and Bcubed F-Score. A meta-analysis will take place to evaluate the effectiveness of these methods when compared with each other and by-eye analysis from social media domain experts.

The method of evaluating the approaches will be developed using a combination of Python and R. R is a combined language and environment created for the purpose of statistical computing [39].

Python is being utilised also due to the availability of relevant software packages for the purposes of evaluation measure calculation.

4.2.1 Calculation of Changepoint Locations

`changepoint` is a powerful R package that provides a number of different change detection algorithms, along with various approaches to penalty values. `changepoint` offers change detection in mean, variance and combinations of the two, using the AMOC (*At Most One Change*), PELT (*Pruned Exact Linear Time*), Binary Segmentation and Segment Neighbourhood algorithms.

`Changepoint` was developed by Rebecca Killick and Idris A. Eckley and is provided free of charge under the GNU general public license [29].

4.2.2 Calculation of Evaluation Measures

For the calculation of Precision, Recall and F1 Score, the R package `caret` is being utilised. `caret` is an acronym for 'Classification and Regression Training, and provides a number of tools for data manipulation, model creation and tuning, and performance measurements. Through the use of this package, it is possible to generate a *confusion matrix* based on algorithm output, and generate various performance measures based upon this. `caret` was written by Max Kuhn, and is provided free of charge under the GPL license [31].

For the calculation of the Rand Index and Adjusted Rand Index the R package `phyclust` is being used. This package was developed by Wei-Chen Chen, for the purposes of providing a phyloclustering implementation. While this approach is not something being examined in this thesis, the package does provide an implementation of the Rand Index and Adjusted Rand Index metrics, which are relevant to this research. This package is also provided free of charge under the GPL license [10].

Calculating the BCubed precision, recall and f-score metrics is being carried out in Python, using the `python-bcubed` project. This is a small utility library for the calculation of BCubed metrics, developed by Hugo Hromic and provided under the MIT license [21]. In order to interface with this library via R, the package `rPython` is being used. This package provides functionality for running Python code and retrieving results in R, and was developed by Carlos J. Gil Bellosta. It is provided under a GPL-2 license [17].

Additionally, the `ROCR` package was utilised during the research phase of this project to generate Receiver Operating Characteristic (ROC) curves - a method used by some publications to evaluate change point detection methods. While ROC curves are not being utilised to produce the results published in this thesis, it was nonetheless an important part of the research, and provided useful insights into binary classification as a field and as a method of evaluation. `ROCR` was developed by Tobias Sing et. al. and is provided free of charge under a GPL license [44]

4.3 Measures Meta-Analysis

The experiments hereunder were developed to fulfil the requirement to answer research question 2, and it's associated sub-questions:

- Are measures not specifically defined for the purpose of change detection evaluation effective in this domain?
  - In what way do established measures fail to behave correctly?
  - Which evaluation measure provides the best evaluation, given the functional requirements set forth by the host organisation?

The research questions above are to be answered using a series of experiments carried out using simulated data and algorithms. The experiments intend to test the following criteria:

1. Dependence on sample size (otherwise referred to here as time series length)
2. Impact of data preceding a known change point and its detection
3. Impact of data following a known change point and its detection
4. Ability to provide a temporal penalty for late, early or exact detections
5. Ability to penalise correctly for false positives
6. Ability to penalise correctly for false negatives
7. Impact of change point density in an analysed time series

What follows is a listing of the experiments carried out. Illustrative pseudocode algorithms for each experiment can be found in appendix A of this document.

4.3.1 Experiment Listings

For each experiment description, the sample size (otherwise known as time series length) is denoted by n, the true change point(s) in the time series is denoted as τ_n and any 'detected' change points are denoted by τ'_n. For each experiment, the changing variable (the iterator in the R source code) is denoted as i. For experiments where the computed change point(s) τ'_n are placed with some error, this error is denoted as $\Delta\tau$. $\Delta\tau$ can always be expected to adhere to $\{\Delta\tau \in \mathbb{R} \mid \Delta\tau \geq 0\}$, as values < 0 can be considered successful, early detections - and thus do not constitute an error.

Experiment 1

This experiment involves increasing the 'head' (sample size prior to a single known change point) of a time series, prepending to and thus lengthening the time series. The time series contains a single known change point, and a detected change point provided by a pseudo-algorithm is placed such that it is placed with a $\Delta\tau$ of 5.

For each prepended point, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 2.

The experiment runs such that $n = 55$ on commencement, and $n = 500$ upon completion, with each iteration adding a single data point such that $i = 445$ on completion. For each iteration, $\tau = i + 1$ and $\tau' = \tau + 5$.

For this experiment, the null hypothesis H_0 is that neither additional points before a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.

The alternative hypothesis H_1 is that the measures will be affected in some way by additional points before the change point. It is also hypothesised that data set length will have an affect on the metric value.

Experiment 2

This experiment involves increasing the 'tail' (sample size after a single known change point) of a time series, appending to and thus lengthening the time series. The time series contains a single known change point τ and a detected change point τ' provided by a pseudo-algorithm with a $\Delta\tau$ of 5.

For each appended point, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 3.

The experiment runs such that $n = 55$ on commencement, and $n = 550$ upon completion, with each iteration adding a single data point such that i runs from 2 to 500. For each iteration, $\tau = 51$ and $\tau' = 56$.

For this experiment, the null hypothesis H_0 is that neither additional points after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.

The alternative hypothesis H_1 is that the measures will be affected in some way by additional points after the change point. It is also hypothesised that data set length will have an affect on the metric value.

Experiment 3

This experiment involves moving a known change point and a computed change point through a fixed length time series. The time series contains a single known change point τ and a detected change point τ' with $\Delta\tau = 5$, provided by a pseudo-algorithm.

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The experiment evaluates both criteria 1 and 2.

The experiment runs such that n is constant at $n = 501$. For each iteration, $\tau = i$ and $\tau' = i + 5$, for values of i from 5 to 500.

For this experiment, the null hypothesis H_0 is that neither additional points before or after a change point, nor data set length, will have an affect on the scores provided by evaluation metrics.

The alternative hypothesis H_1 is that the measures will be affected in some way by additional points before or after the change point. It is also hypothesised that data set length will have an affect on the metric value.

Experiment 4

This experiment involves moving a detected change point τ' provided by a 'pseudo-algorithm' through a time series of fixed length, thus evaluating the ability of a measure to provide a temporal penalty for late, early, or exact detections.

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The experiment evaluates criterion 4.

The experiment runs such that n is constant at $n = 500$. For each iteration $\tau = 51$ and $\tau' = i$, where $i = 51$ at the commencement (such that $\tau = \tau'$) and $i = 500$ at completion. Thus, values of $\Delta\tau$ range from 50 to 500.

For this experiment, the null hypothesis H_0 is that all of the calculated metrics will correctly issue a penalty for late detections of a change point, in all instances.

The alternative hypothesis H_1 is that one or more measures will incorrectly issue scores for a detected change point occurring various distances from the true change point.

Experiment 5

This experiment involves adding 'false positive' change point detections of various values of $\Delta\tau$ to a time series of fixed length with a single known change point τ.

For each iteration of the experiment, all of the evaluation metrics are calculated and plotted. The Experiment evaluates criterion 4.

The experiment runs such that n is constant at $n = 500$. For each iteration $\tau = 51$ and $\tau' = i$, where $i = 51$ at the commencement of the experiment, $i = 55$ at the second iteration, and increases in increments of 5 until $i = 500$. This serves to move the detected change point through the time series, starting at the true change point and progressing to the end of the time series, ending at the final point of the time series.

For this experiment, the null hypothesis H_0 is that all of the metrics will correctly penalise for false positive detections in a time series.

The alternative hypothesis H_1 is that the measures will not correctly issue penalties for false positive detections.

Experiment 6

Removing 'false negative' results (by way of explanation, adding correct detections) to a fixed length time series, thus increasing the number of correctly detected change points on each iteration.

The experiment begins with a time series of $n = 900$ with τ being situated at intervals of 100. As the experiment progresses, τ' points are added such that $\tau_n = \tau'_n$ for each iteration. Thus, adding a computed change point detection at the same place as a ground truth detection (a $\Delta\tau$ of 0)

This experiment evaluates criterion 6, by recalculating the scoring metrics upon each iteration.

For this experiment, the null hypothesis H_0 is that removing false negative results will result in a linear increase in score from each metric.

The alternative hypothesis H_1 is that the measures will not appear to correctly increase in value as false negatives are removed from the data set - showing instead inconsistent behaviour.

Experiment 7

Adding both true change points and detected change points from a 'pseudo algorithm' that provides detections every 25 points to a time series, by appending to it, and thus increasing it's length. The experiment runs from $n = 50$ until $n = 1000$ in increments of 50. Each τ and τ' is placed upon a new iteration such that each $\tau'_n = \tau_n + 2$, giving a $\Delta\tau$ value of 2. This experiment tests both criteria 1 and 7.

For this experiment, the null hypothesis H_0 is that metrics will be unaffected by change point density in a time series, nor will they be affected for changes in time series length.

The alternative hypothesis H_1 is that the measures will be affected in some way by change point density and data set length, showing an increase or decrease in score value as density and length increases.

Experiment 8

Adding both true change points and detected change points from a 'pseudo algorithm', to a time series of fixed length. The experiment runs with a static data set size of $n = 1050$, and adds two τ and τ' on each iteration, effectively 'lifting' a spike out of a static data set where all values are 0 at the start of the experiment. Each τ' is placed such that $\tau'_n = \tau_n + 2$ (a $\Delta\tau$ of 2, simulating a slightly late detection. This ensures that the measures do not report perfect detections throughout the experiment, thus making it impossible to spot how change point density in a fixed length data set affects the calculation of metrics.

This experiment fulfils criteria 7.

For this experiment, the null hypothesis H_0 is that metrics will be unaffected by change point density, maintaining a constant value.

The alternative hypothesis H_1 is that the measures will be affected in some way by changes in change point density. The hypothesis is that the value of the measures will increase or decrease as change point density increases.

4.4 Comparison of Measures Based Upon Functional Requirements

As part of the research being carried out, a set of functional requirements are elicited from the host organisation. Based on these functional requirements the measures are evaluated and compared in such a manner that we can choose the 'best' measure for the use case of the host organisation. The previous research section (§ 4.3) provides the results necessary to compare the behaviour of certain metrics in certain situations, against the priorities of the host organisation.

For example, if the host organisation expresses that they require an approach with an absolute minimum of false positives, it is important to judge the application of these algorithms against real world data, utilising measures that correctly penalise for false positive detections in a data stream.

A summary of the functional requirements gained as a part of this research can be found in § 6

4.5 Application of Algorithms to Real World Data

4.5.1 Data Preparation

The experiment is being carried out using conversation volume data taken from Buzzcapture's Brand Monitor application. Brand Monitor harvests tweets via the Twitter API and makes them searchable through a bespoke interface as well as providing a number of useful metrics for reputation management. In order to gain data for this experiment, Social Media Analysts from Buzzcapture were requested to provide query strings for the Brand Monitor application, showing a situation where they feel a client should have been informed of a distinct change in conversation volume. The analysts were then asked to manually annotate the data with the points at which they believe a change point detection algorithm should detect a change. In this way, bias is avoided when establishing the ground truth for the data - the author was not involved in this annotation process other than providing instructions. The result of this exercise is then a set of time series' showing the twitter conversation volume over time for a given brand. This is accompanied by a separate data set containing indices at which changes *should* be detected - thus serving as the ground truth for this part of the experiment.

Once the query strings provided were executed, the corpus of test data included the following data sets:

- Dirk
- Bol.com
- Connexion
- DAP

- Jumbo
- Kamer van Koophandel
- Rabobank
- Tele2
- UWV
- Ziggo

All data sets were trimmed such that the sample size equals 60 for every set. The sets were exported from the BrandMonitor application in CSV format, with daily conversation volume totals.

4.5.2 Execution

To perform the experiment, an R script is executed that reads the CSV file containing the data points being analysed, and reads them into a data frame object. At this point, change point analysis is conducted using the following algorithms and test statistics:

- Mean, using PELT
- Mean, using SegNeigh
- Mean, using BinSeg
- Variance, using PELT
- Variance, using SegNeigh
- Variance, using BinSeg
- Mean & Variance, using PELT
- Mean & Variance, using SegNeigh
- Mean & Variance using BinSeg

Once all of the analysis methods have been executed successfully, the change points are extracted and used to compute the various scoring metrics, when the change points are compared against the established ground truth detections.

There are some assumptions made when handling this data. Firstly, for the binary classification measure, as the data being used in this study consists of daily conversation volume statistics, anything other than an *exact* detection when compared with the ground truth is considered a *failure*. Detections of a change a day after the true change point are too late for useful notifications to be sent in a production system. Detections prior to the true change point, while possibly useful in some way for predicting a future change (certainly in a production system), are also considered a *failure*. The clustering measures should not be affected by this assumption, as by their nature they should provide a more granular score for detections slightly before or after the ground truth change point.

Secondly, as discussed before, the algorithms being evaluated require some assumption to be made as to the probabilistic distribution of the data being analysed. Diagnostic histogram plots created prior to experimentation showed that the various data sets varied considerably in terms of the probability distribution of data points. It is possible for algorithms configured to use a mean test to be configured to use a CUSUM test statistic that makes no assumption about the data distribution being analysed. Unfortunately, at the time of the experiments, the CUSUM test statistic was not supported for variance or mean/variance tests. The variance and mean/variance tests allow for the selection of other distributions such as poisson and gamma. As such, all algorithm runs were configured to assume a *normal* probability distribution. While this assumption may not hold for all of the data sets, it is necessary to take this step to ensure a like for like comparison as much as possible.

After all of the metrics are calculated and tabulated, it is possible to see which algorithm performed the 'best', and also carry out a comparison analysis to see which algorithms provided the most consistent results against all data sets.

Chapter 5

Research

5.1 Introduction

This section of the thesis describes the execution, results and conclusions of each individual study carried out. It is divided into three sections:

- Simulation studies - taking simulated data and simulated algorithm results to analyse the impact of various cases on calculated measures.
- Analysis of measures - taking the results of the simulation studies and testing their ability to match the functional requirements of the host organisation.
- Real-World Data - taking a set of real world twitter data and analysing the ability of change detection algorithms to detect changes in the data compared with established ground truth change points.

5.2 Simulation Studies

The simulation studies carried out are intended to compare each of the measures against the set of criteria described in § 4.3. The simulations are carried out according to the experiment descriptions previously explained, and the execution and results are presented here.

The scores given by 'intermediate' metrics used to calculate final metrics (recall and precision for both F1 Score and BCubed F-Score) are also plotted for completeness, though they will not be subject to the result discussion.

5.2.1 Dependence on Sample Size

Two of the experiments carried out in this thesis take particular note of sample size. Both involve increasing the sample size of the data, appending or prepending a number of points both prior to and following the ground truth change point. In each situation, a simulated algorithm detects a change n points after the ground truth, simulating detection that is relevant, though slightly late, to ensure that there that the baseline metric value at $t = 0$ is not equal to 1.

Figure 5.1 and Figure 5.2 show the results of this simulation study. Figure 1 demonstrates the value of metrics as the 'head' of the data is increased, while Figure 2 demonstrates the value of metrics as the 'tail' of the data is increased.

In this situation, we can see that both studies show a variation in metric score as the length of the data stream increases. For all of the clustering measures there is a clear increase in metric value as points are added. At $t = 0$, the clustering metrics appear to correctly penalise for a late detection, though this penalty decreases as t increases.

The Adjusted Rand Index shows a far more extreme temporal penalty for the change point detection, which increases sharply at the beginning of the experiment, before maintaining a rate of increase comparable to the other clustering measures.

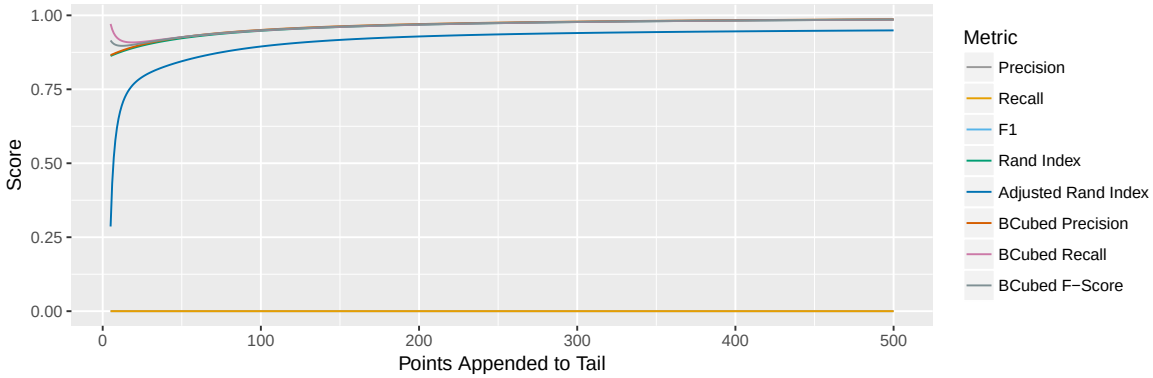

Figure 5.1: Tail-Append Results Plot

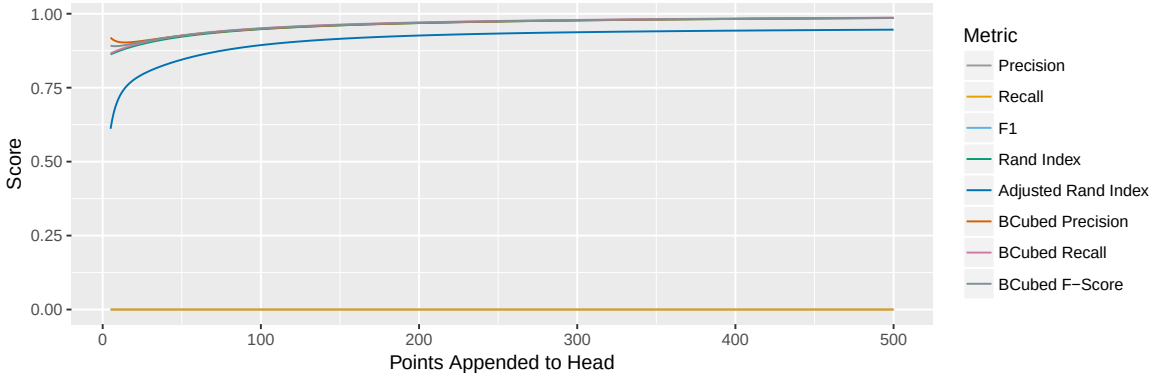

Figure 5.2: Head-Prepend Results Plot

The binary classification measures of Precision, Recall and F1 maintain a constant value of 0 throughout the experiment, classifying the change point detection as a 'missed' detection and penalising accordingly.

For these experiments (1 & 2) the null hypothesis H_0 is rejected, and the alternative hypothesis H_1 is accepted.

5.2.2 Impact of Data Preceding and Following a Known Change Point

The experiment run for this criteria is similar to those run in § 5.2.1, but differs in one important manner: the size of the time series is maintained at a constant value throughout. As points are added to the 'head' of the data set, points are removed from the 'tail'. In practice, this serves to move both the true and computed change points through the time series.

Figure 5.3 shows the results of the experiment. As with the previous experiment, the binary classification measures maintain a constant value of 0, with variation only being shown by the clustering measures. It is interesting to note that the change in all clustering measures aside from Adjusted Rand Index is minimal, with an almost constant value being held throughout the experiment, aside from when the change points are at the extreme ends of the time series.

The Adjusted Rand Index exhibits behaviour unique to itself, with a value approaching 0 as the change points approach either end of time series. From position 150 to 350, the Adjusted Rand Index holds an almost constant value.

From these results it can be inferred that it is only the Adjusted Rand Index that is effected in any meaningful way by the position of the change point in a fixed length data set.

For this experiment (3), the null hypothesis H_0 fails to be rejected for several measures. The only situation in which the alternative hypothesis H_1 holds is with the Adjusted Rand Index, which shows

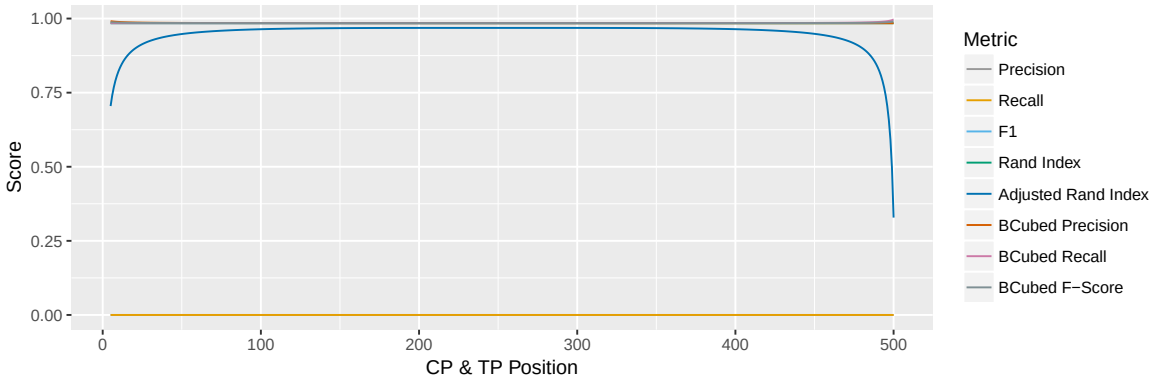

Figure 5.3: Fixed Length Append/Prepend Results Plot

a considerable range in values when the change point is at the extreme ends of the time series.

5.2.3 Temporal Penalty

This is experiment makes use of a fixed length data set with a single, static, known change point. A computed change point is moved through the time series, beginning at the known change point, and ending at the end of the time series. Figure 5.4 shows a plot of the results of this experiment:

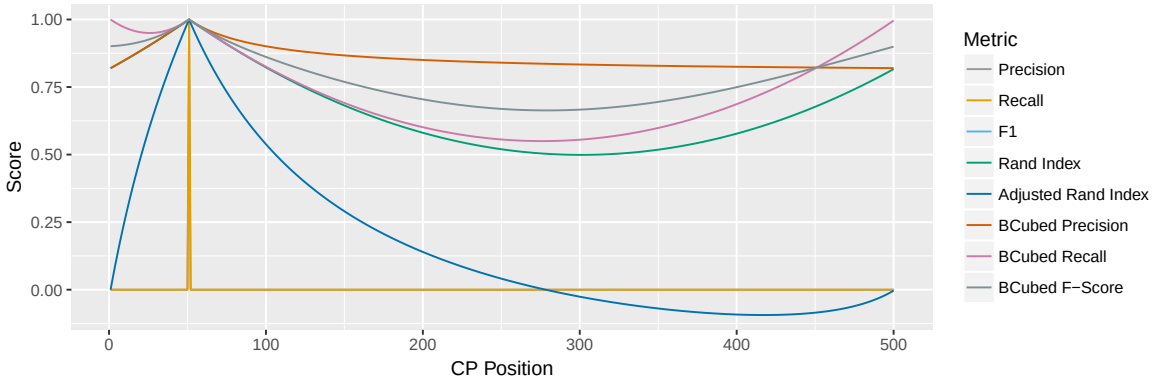

Figure 5.4: Temporal Penalty Results Plot

At the start of the experiment, when the ground truth and the computed change point are equal, all of the measures return a value of 1 as expected. Once the computed change point is moved through the time series, the binary classification measures immediately return a value of 0, showing that none of the change points in the series (a single change point, in this case) have been detected.

All of the clustering measure values decrease as the computed change point moves away from the ground truth change point, though exhibit some strange behaviour as the computed change point approaches the end of the time series. In all cases, the clustering measures show an increase in score as the computed point approaches the end of the time series, with the temporal penalty for a late detection reducing.

The Adjusted Rand Index once again exhibits behaviour unique to itself, dropping below 0 at approximately $t = 278$. This behaviour is expected from the Adjusted Rand Index, being that the adjusted nature of the metric (allowing for cluster classification occurring by chance) results in a metric capable of returning a value < 0.

The results of this experiment (4) cause the null hypothesis H_0 to be rejected. All of the measures show some issue with the application of a temporal penalty for late detections, and indeed also show issues with crediting algorithms for early detections. In this case, the alternative hypothesis H_1 holds.

5.2.4 Penalisation for False Positives

This experiment utilised a fixed length time series with a single ground truth change point, and a single computed change point that equals the ground truth change point. Successive false positives are added to the time series, moving from left to right across the time series.

Figure 5.5 shows the results of the experiment, with the score provided by each metric plotted against the number of false positives present in the data stream.

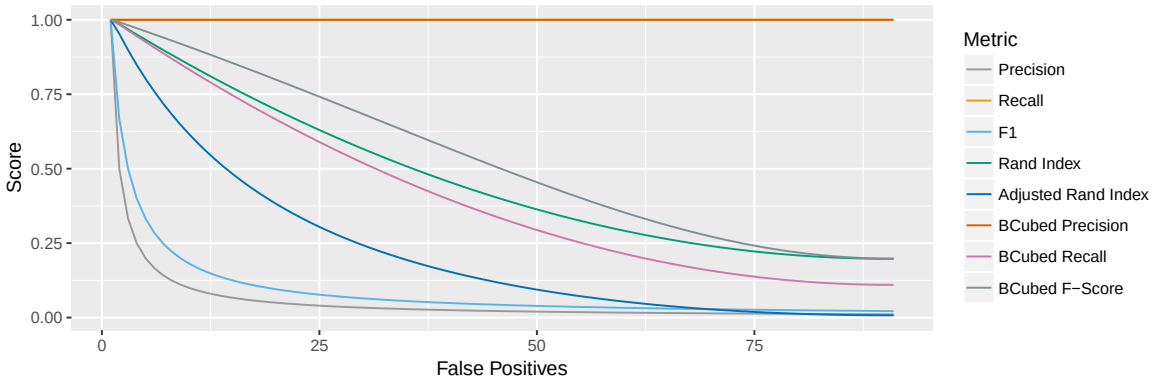

Figure 5.5: False Positives Results Plot

The results of this experiment show that almost all of the metrics behave in a similar fashion. The binary classification measure of Recall maintains a value of 1, while all of the other measures show a precipitous drop in score as the initial false positives are added. Precision and F1 scores for binary classification drop sharply, while the measures for clustering show a more gentle decrease in score. The Rand Index and BCubed F-Score end the experiment at approximately the same value, while the Adjusted Rand Index, F1 Score and Binary Classification Precision metrics also terminate at approximately the same value.

The results of this experiment cause the null hypothesis H_0 to hold, as all metrics behave correctly in this situation.

5.2.5 Penalisation for False Negatives

For this criteria evaluation, the experiment utilised a fixed length time series with multiple ground truth change points. As the experiment progressed, computed change points were added, equal to each of the ground truth change points. Figure 5.6 shows the results of the experiment, with the metric scores plotted against the number of false negatives present in the data stream.

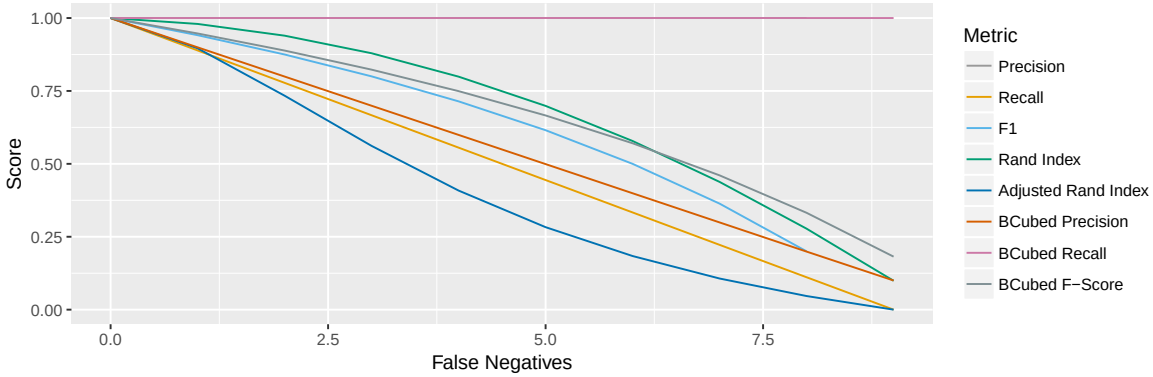

Figure 5.6: False Negatives Results Plot

The vast majority of measures see a consistent decrease in returned score as the number of false negatives increases. Indeed, once all of the ground truth change points have been correctly detected, all of the metrics return a score of 1. Only the Adjusted Rand Index returns a score of 0 after 10 false negatives are found in the data stream. As Expected, BCubed Recall remained at a value of 1 for the duration of the experiment.

5.2.6 Number of Change Points in Fixed Length Data Stream

This experiment adds change points to a data stream by 'lifting' slices of the data stream - adding a value to them to create two change points. This is carried out multiple times, each time adding two ground truth change points, and two computed change points 3 time units after the ground truth change points. Figure 5.7 shows the results of this experiment, plotting the score returned by the metric against the number of change points in the stream.

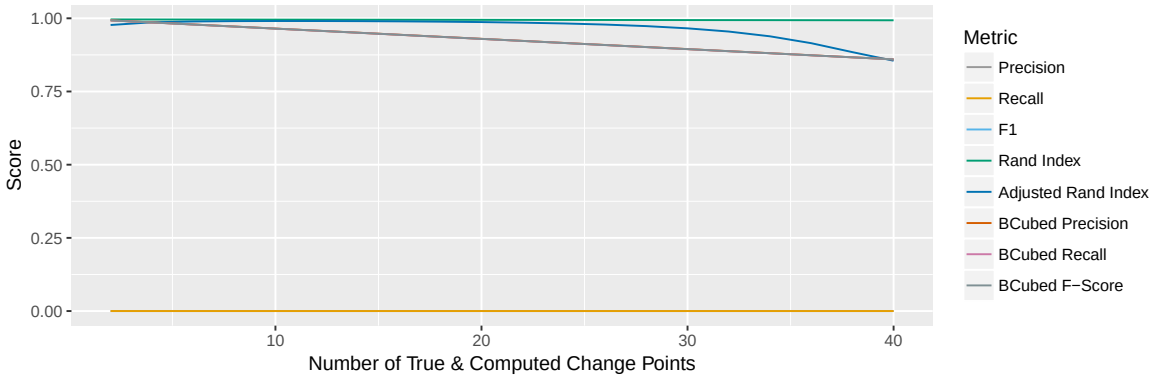

Figure 5.7: Change Point Density (Fixed Length) Results Plot

The results of this experiment show a drop in score provided by the Adjusted Rand Index and BCubed F-Score. The BCubed F-Score shows a linear rate of decline as change points are added to the stream, while the Adjusted Rand Index shows an increase in downward velocity towards the end of the experiment. All of the other metrics maintained a constant value of either 1 or 0, as expected.

5.2.7 Number of Change Points in Variable Length Data Stream

This final experiment is similar to the previous one, in that it increases the number of change points in a data stream. However, the difference here is that in this case, data is appended to the data stream, also increasing the length of the stream as each successive change point is added. The results of this experiment are shown in Figure 5.8:

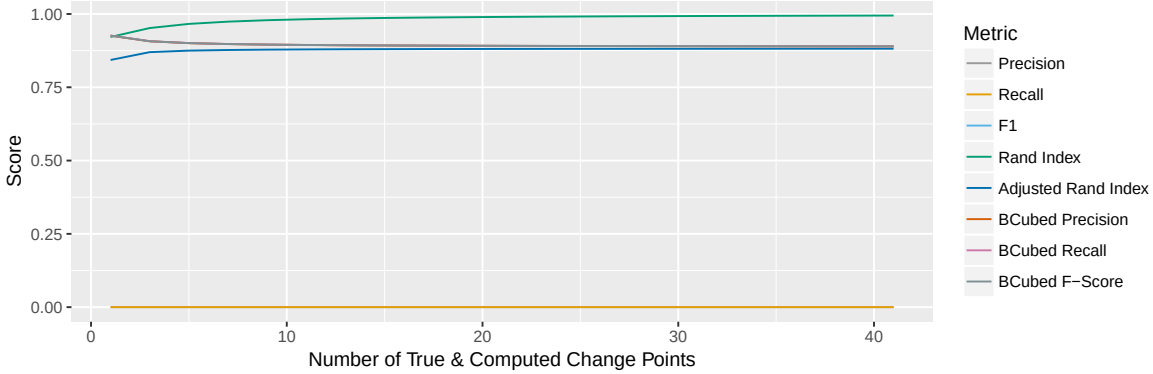

Figure 5.8: Change Point Density (Variable Length) Results Plot

The results of the experiment show some interesting behaviour in the Rand Index, Adjusted Rand Index and BCubed F-Score metrics. The Rand index and the Adjusted Rand Index increase in value as data points and change points are added to the time series, while the BCubed F-Score decreases in value. The Adjusted Rand Index begins at a slightly lower level than the Rand Index and BCubed F-Score metrics, while the Rand Index and BCubed F-Score metrics diverge almost immediately.

Aside from the large change in score value at the beginning of the experiment, the change in value for the Rand Index, Adjusted Rand Index and BCubed F-Score metrics is small.

5.3 Functional Requirements

5.4 Real-World Data Analysis

As described in § 4.5, this experiment executed change detection algorithms against sets of real world data taken from Twitter. The data was gathered and filtered to be relevant to certain brands (again, as listed in § 4.5) and then annotated with ground truth change points by social media analysts from the host organisation. The ground truth annotations can be found in Appendix B.

What follows is the results of that study. For each data set, the scores provided by the evaluation measures are tabulated and provided in Appendix D. The highest score(s) for each measure are highlighted for easy comparison and readability. Further discussion of the results and the extraction of conclusions will can be found in § 6 and § 7.

The final plots, including detected change points by each algorithm can also be found in Appendix C.

Analysis of the raw data used for Figure 5.9 shows that while there is a consensus on the best algorithm, some metrics provide higher scores for algorithms that perform objectively worse than others, when compared through by-eye analysis of the change point plots. Binary classification scores are universally low for all data sets.

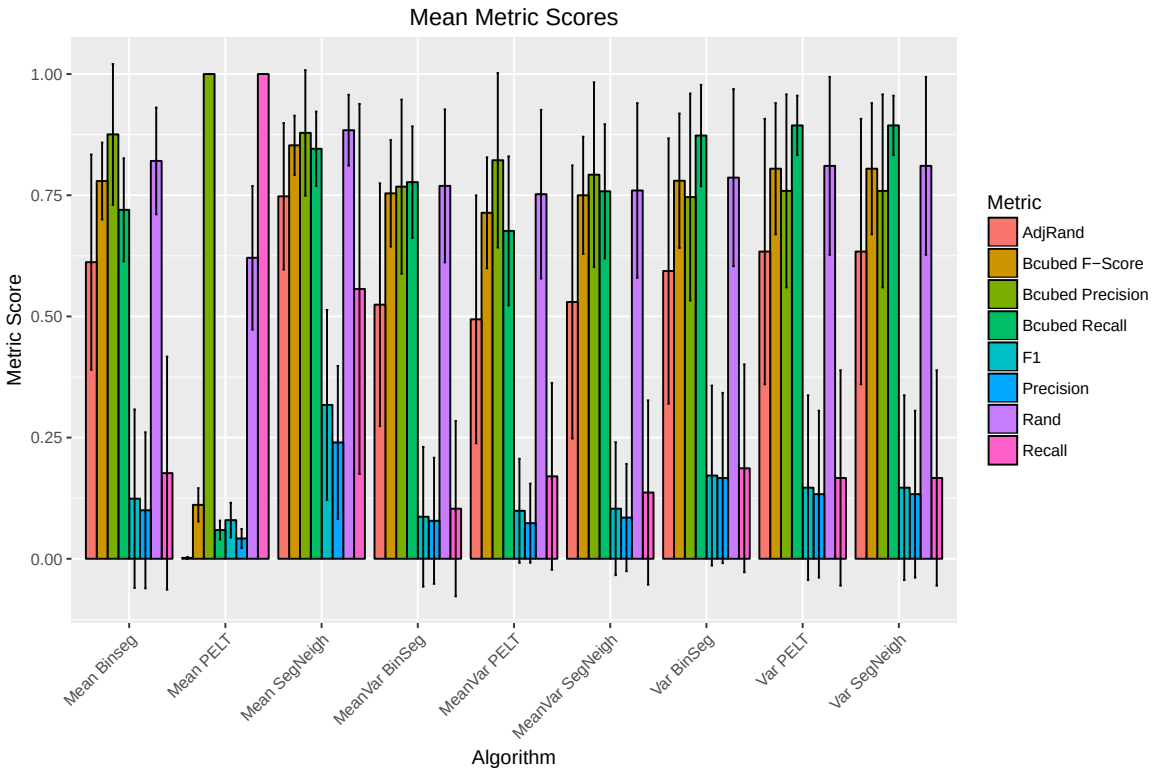

Figure 5.9: Average scores across all real-world data scenarios

Figure 5.9 shows a plot of average score values across all real world data sets, from each algorithm.

The error bars are drawn to show 1 standard deviation of error.

Taking the mean metric score for each algorithm across all of the experiments, we can rank the algorithms as follows, using the four main metrics that this research was focussing on:

Rank	F1 Score	Rand Index	Adjusted Rand Index	BCubed F-Score
1	Mean Segneigh	Mean SegNeigh	Mean SegNeigh	Mean SegNeigh
2	Var BinSeg	Mean BinSeg	Mean BinSeg	Var PELT
3	Var PELT	Var PELT	Var PELT	Var SegNeigh
4	Var SegNeigh	Var SegNeigh	Var SegNeigh	Var BinSeg
5	Mean BinSeg	Var BinSeg	Var BinSeg	Mean BinSeg
6	MeanVar SegNeigh	MeanVar BinSeg	MeanVar BinSeg	MeanVar BinSeg
7	Meanvar PELT	MeanVar SegNeigh	MeanVar SegNeigh	MeanVar SegNeigh
8	MeanVar BinSeg	MeanVar PELT	MeanVar PELT	MeanVar PELT
9	Mean PELT	Mean PELT	Mean PELT	Mean PELT

Table 5.1: Algorithm Rankings

Here it is clear that while there appears to be a consensus on the 'best' algorithm in this limited trial, the remaining rankings vary somewhat. In order to analyse this, the *Kendall's Tau* coefficient of ranking correlation is utilised [25]. This measure produces values of 1 for identical rankings, and -1 for fully different rankings. The results of this calculation follow:

	F1 Ranks	Rand Index Ranks	Adjusted Rand Index Ranks	BCubed F-Score Ranks
F1 Ranks		0.278	0.278	0.778
Rand Index Ranks	0.278		1	0.278
Adjusted Rand Index Ranks	0.278	1		0.278
BCubed F-Score Ranks	0.778	0.278	0.278	

Table 5.2: Calculated Kendall's Tau values between pairs of rankings

With these values in Table 5.2, for each pair we can test the null hypothesis H_0 that $\tau = 0$ and there is no correlation between the rankings by computing the $p-value$ for each τ. This gives results as follows:

	F1 Ranks	Rand Index Ranks	Adjusted Rand Index Ranks	BCubed F-Score Ranks
F1 Ranks		0.3585	0.3585	0.002425
Rand Index Ranks	0.3585		5.51×10^{-6}	0.3585
Adjusted Rand Index Ranks	0.3585	5.51×10^{-6}		0.3585
BCubed F-Score Ranks	0.002425	0.3585	0.3585	

Table 5.3: Calculated p values for Kendall's Tau tests

The values shown in Table 5.3 demonstrate that for the comparison of the rankings provided for F1 and BCubed F-Score, we can reject the null hypothesis H_0 that they are uncorrelated, with a 95% confidence interval, as $p < 0.05$. Similarly, we can reject the null hypothesis for the ranks provided by the Rand Index and Adjusted Rand Index, with a confidence interval of 95%, as $p < 0.05$. This is not unexpected, as the Adjusted Rand Index is derived from the classic Rand Index, and in the case of these experiments we would not expect to see any deviation in rankings provided. The other pairs of rankings show a p value of > 0.05 and as such, the null hypothesis H_0 that these ranks are not correlated holds, with a 95% confidence interval.

Chapter 6

Results

This chapter is intended to provide a factual explanation of the results of each experiment carried out as a part of this research. It does not provide any discussion or conclusions based upon these results. The discussion and conclusions gleaned from these experiments can be found in § 7.

6.1 Metric Behaviour

Behaviour	F1 Score	Rand Index	Adjusted Rand Index	BCubed F-Score
Credits correct detections	✓	✓	✓	✓
Penalises false negatives	✓	✓	✓	✓
Penalises false positives	✗	✓	✓	✓
Temporal penalty for late detections	✗	●	●	●
Unaffected by sample size	✓	✗	✗	✗
Unaffected by change point density	✓	✓	●	●
Unaffected by data before/after change point	✓	●	✗	●

Table 6.1: Metric Behaviour Summary

Table 6.1 shows a summary of how the metrics being tested in this study behaved when the experiments were run. Behaviours exhibited by a given metric are denoted by ✓, behaviours not exhibited as ✗, and behaviours that are exhibited with some caveats are denoted by ●.

6.1.1 F1 Score

The F1 score is a 'traditional' binary classification score, widely used for the evaluation of change point detection methods. Based upon the results of the experiments carried out, it is clear that this measure has some limitations in this domain.

Firstly, F1 score does not include any method for penalising a method based on late detections. Detections of a change point some number of points after the true change point may still be considered successful detections, depending on the functional requirements of the system that the method is implemented in. F1 Score returns a score of 0 for anything other than exact detections. However, the measure does properly penalise for false negatives. In a data set with two known change points, if one of these change points is successfully detected and the other not, the metric will return an F1 score of 0.5, which could be considered correct.

Secondly, F1 score does not appear to correctly penalise for false positives. In situations where a change point is detected some number of points after the true change point (so, as above, possibly a successful detection, albeit late), the metric continues to return a value of 0, even when false positives occur.

The F1 measure is, however, unaffected by changes in sample sizes in the experiments, and also exhibits no variation when the change point density in a given data set is increased or decreased.

6.1.2 Rand Index

The Rand Index is a clustering measure, calculated in these experiments by segmenting the data set around detected change points and treating all data points in each of these segmentations as a single cluster.

The Rand Index improves upon the F1 score, in that it successfully penalises for late detections, with some caveats. The Rand Index computed for a data set with a single late detection better reflects the true 'accuracy' of a method, but it behaves inconsistently when this detected change point is near to the beginning or end of a time series. In the experiment testing temporal penalty, the Rand Index successfully provides a decreasing value as the detected change point moves away from the true change point, but starts to increase as it approaches the end of the data set.

Taken intuitively, this means that if a method detects a change point late, the Rand Index only penalises up to a certain point. If the detection occurs towards the end of the data set, the Rand Index may return a score that is actually *higher* than that is returned if a method detects a change point late, but closer to the true change point. This is a clear limitation of the measure, and means it cannot be considered to completely fulfil this behaviour.

It is also clear, based on the results of these experiments, that the Rand Index is affected by the size of the data set, with a late detection being penalised higher in a small data set, than in a large. However, this difference is reasonably small - and it is safe to assume that in most situations, change detection methods will be compared in a like-for-like fashion, including ensuring that the test data sets are all of the same length.

6.1.3 Adjusted Rand Index

The Adjusted Rand Index is an additional clustering measure, calculated in a similar fashion to the 'standard' Rand Index, but takes into account the possibility that some items in a cluster may be accidentally, or randomly classified into that cluster.

The Adjusted Rand Index behaved in much the same fashion as the Rand Index in most situations, though exhibited different behaviour in more than one scenario. The Adjusted Rand Index is more affected by dataset size, which is especially apparent in the experiment where change point density is increased while also increasing the data set size.

This measure also reacted inconsistently when both the true change point and computed change point are moved through the time series - showing a large, steep drop in returned score at both the beginning and end of the data set.

Temporal Penalties were more effectively scored by the Adjusted Rand Index, however - as even though the score returned by the measure increased as the detected change point moved towards the end of the time series, the value returned by the Adjusted Rand Index did not become higher than 0 at any one point.

6.1.4 BCubed F-Score

As a clustering measure, it was expected that BCubed would behave in much the same way as the Rand and Adjusted Rand Indices. However, it does exhibit some minor differences.

The BCubed metric is less heavily affected by the parabolic curve behaviour exhibited by clustering metrics when plotting temporal penalty - though it is affected. The results returned by BCubed show less variation as the computed change point moves towards the end of the time series, but it shows the same curve with approximately the same velocity as the Rand Indices.

The BCubed score also reacts differently to the Rand Index when change point density is increased with time series length, showing an almost linear drop, converging with the Adjusted Rand index at the end of the experiment, unlike the Rand Index, which remains constant.

When compared with the behaviour in a fixed length data set, the BCubed measure behaves in the opposite fashion to the Adjusted Rand Index. The scores start diverged, but quickly converge to the

same value and remain almost constant towards the end of the experiment. BCubed also behaves in the opposite fashion to the 'standard' Rand Index, with the scores diverging from the beginning of the experiment.

6.2 Fulfilment of Functional Requirements

A summary of the functional requirements follows:

R1 The approach should not rely on arbitrary threshold values. It instead should adapt to the data that it is analysing.

R2 False positives should not be considered wholly harmful. The consequence of a false positive in production is only that a notification is sent. That said, false positives should be kept to a reasonable level.

R3 Late detections should be considered failure cases. Late detections are not useful to clients using the production system, as prompt detections are needed for effective damage control in situations where increased conversation volume is caused by a PR crisis.

R4 Similarly, false negatives should also be considered failure cases. False negatives are defined as when the algorithm fails to flag a change point that would have been flagged by manual, by-eye analysis. False negatives prevent clients from acting upon changes in conversation volume in a timely manner.

R5 The approach in it's first iteration should operate on conversation volume metrics provided by Brandmonitor, but should be extensible such that it can also be applied to other metrics.

It is clear from the behavioural analysis of the measures, that there is not a single measure that best fits all of the functional requirement criteria for the purposes of scoring the algorithms being evaluated here. Taking each requirement in turn, the following is found:

R1 None of the algorithms evaluated rely upon arbitrary threshold values. It is not possible in the context of this study, to choose a metric that evaluates on this criteria.

R2 False positives were penalised harshly by almost all measures. If the measures are taken in their default forms, false positives are punished heavily when there are few of them, with the effect diminishing as their numbers increase.

R3 This criteria is fulfilled, but with a caveat. All of the measures tested penalised for late detections, but also penalised for early detections - which intuitively should be considered a good thing in this use case.

R4 This requirement is fulfilled by all of the major measures being evaluated. Each measure shows an *almost* linear relationship between score and the number of false negatives in a time series.

R5 This requirement is incidental and was elicited as part of the requirements process, but does not have a bearing on how the methods were evaluated. As it stands, all of the methods can be applied to any arbitrary time-series data, where the values y_s are such that $\{y \in \mathbb{R}\}$.

6.3 Real-World Data Analysis

Intuitively, the results show that all of the measures agree that Segment Neighbourhoods utilising a mean test statistic, performed the 'best' in this limited study of real-world data sources.

However, it is important to note that the fact that there is no correlation between rankings (with the exception of Rand $\leftrightarrow$ Adjusted Rand and F1 Score $\leftrightarrow$ BCubed F-SCore) suggests that this cannot be taken on face value alone, and the lack of correlation may imply that even though the 1st ranked

algorithm is consistent across all rankings, it is simply not possible to state that all of the measures used agree that one measure was better than all of the others in these tests.

Further studies would need to be carried out, with a much greater sample size in terms of data sources, in order to gain a more accurate ranking of approaches.

Chapter 7

Analysis & Conclusions

This section constitutes the conclusions gleaned from the experiments that were carried out as a part of this thesis. Here, the research questions set out in § 1.1 are answered, based upon the results of the experiments. For ease of understanding, the research questions are repeated:

SQ1 Are existing change point detection algorithms effective at detecting changes/events in social media data in a timely fashion?

SQ1.1 Which algorithm, out of those being analysed, performs the 'best' when applied against real world data?

SQ2 Are measures not specifically defined for the purpose of change detection evaluation effective in this domain?

SQ2.1 In what way do established measures fail to behave correctly?

SQ2.2 Which evaluation measure provides the best evaluation, given the functional requirements set forth by the host organisation?

SQ3 If existing measures are deficient in some way, what would an 'ideal' measure look like?

7.1 Conclusions

The answer to *SQ1* has proven inconclusive. There are situations in which the algorithms performed well, detecting change points close to those that constituted the ground truth - but there were also many situations in which the algorithms did not perform in the manner expected. Late, early and missed detections were common, with algorithms occasionally showing many false positive detections throughout the data set. This poses two more questions: Were the algorithms incorrectly configured in some way? Or perhaps the selection of penalty function was not appropriate for this particular type of data? Taken at face value, the answer to the question is 'no' - though taking a closer look at the data provides a more nuanced answer. In order to answer this question fully, further work would certainly need to be carried out.

To answer *SQ1.1*, the rankings calculated after running the algorithms against a curated set of 10 different data sets, and the correlation coefficients calculated thereafter show that there is little agreement between the measures as to which method is the best for this particular application. The variation shown when plotting 1 standard deviation error bars also suggest that further studies would be necessary (with a much larger corpus of test data) to obtain useful data. As discussed in § 6, the lack of correlation between rankings of algorithms show that it is not possible to state (regardless of how the rankings appear) that the algorithm ranked highest by all of the measures is the 'best' for this particular application.

When the algorithms were applied to real world data sets, they behaved inconsistently, to the point where no measure agrees on which approach is 'best' for all of the data sets evaluated. When the change point plots are examined by-eye, it is clear that the algorithms underperformed when used in

this context. There were cases where the manual ground-truth annotations were correctly detected by the algorithms, but there were also many situations in which the algorithm output did not closely match the ground truth at all. Despite this, the metrics being used still scored the algorithms highly in many cases (assuming a score > 0.5 to be a 'successful' run).

In answer to *SQ2* and *SQ2.1*, There are multiple ways in which the measures evaluated in this thesis are shown to be inappropriate for use for evaluating change point detection approaches.

The clustering measures evaluated (Rand, Adjusted Rand, BCubed) show that they are often affected by the size of the data set, or by the density of change points in the data set, when scoring change points detected with some error.

Clustering measures especially exhibited inconsistent behaviour when penalising for late (or early) detections. In the experiments carried out, the fact that the score *increases* as the detected change point approaches the end of the time series shows that these measures may be unreliable for providing scores for approaches in which detections are extremely late.

Binary Classification metrics performed well in simulation studies, fulfilling all but two of the criteria tested. However, when applied to real world data and change points detected therein, they proved to be ineffective.

Answering the question of comparing the requirements against measure behaviour, it can be said that the answer to *SQ2.2* was also inconclusive. There was no single measure that fulfilled all of the functional requirement criteria for the selection of change point detection method for this domain. Indeed, it can also be stated that there is also not a single algorithm among those evaluated that fulfilled all of the functional requirements set forth by the host organisation. Concluding, the author asserts that the results of these experiments show that change point detection is perhaps not the optimal method for producing 'smart' notifications of changes in conversation volume, and a different approach not evaluated in this work may be more suited for this specific use-case.

These results do call into question some aspects of the validity of the experiments, which will be discussed in § 7.3.

7.2 The Ideal Metric

If the behaviours in § 6.1 are considered to be the ideal criteria for a measure to fulfil, then it is clear how we would expect an 'ideal' measure to be have. This section intends to answer *SQ3*, based on experiences and evidence collected during this project.

An ideal measure should provide proper credit to correct detections, as well as correctly penalising for false positives and false negatives accordingly. It would be sufficient for scoring if there was a mechanism for defining the 'relevance' of a given change, perhaps by incorporating computation of the area under the curve for the signal around the change when devising the ground truth. This would then provide higher relevance scores for large changes, and lower relevance scores for small changes. In this way, an algorithm could be penalised in a lesser fashion for missing a detection of a less relevant change, and penalised more heavily for missing a detection of a very relevant change.

An ideal measure would be unaffected by the number of data points either side of the change. This could be achieved by taking a 'slice' of the signal around the ground truth change point, and computing a metric based on the location of the true change point and the detected change point within that slice. This would have the additional benefit of causing a metric to be unaffected by the data points either side of a detected or ground truth change point.

This would mean, however, that a method would have to be devised to calculate a final score that was unaffected by change point density. In situations where the ground truth is known, a process such as computing a harmonic mean of detection scores for each change point would be effective, but data with a single 'badly' detected change point would still receive a higher score than data with multiple change points with a small distance between the detection and true change point.

Ideally, there would be a linear relationship between the score, and the relative distance that a computed change point was discovered from the true change point. This was a property that was definitely not exhibited by the measures being evaluated in this study, and it is the believe of the author that this is one of the most vital properties for an 'ideal' measure for change point detection problems.

This is an area that would certainly benefit from further research - the other results of the studies carried out show that there is a need for a change point problem specific measure.

7.3 Threats to Validity

It is not uncommon for an experiment or set of experiments to have some threats to validity. It is the intention of this section to detail the threats that have been identified at the time of writing.

Firstly, the corpus of data sets used for the real-world analysis was too small to be able to state with certainty that the results that were obtained are valid in all situations. It would be better in a future study if a curated data set containing several thousand signals could be analysed, but time constraints caused this kind of study (especially on real world data) to not be feasible in this situation.

This could be compensated for by pivoting away from real world data, and instead generating programatically, a number of data sets (perhaps $n = 10,000$) that have the same properties as the real world data sets, but with a random distribution of change points and change magnitudes. As part of this thesis, such a method was created and tested, but time constraints resulted in it not being included in this thesis as an experiment.

The author is also aware that there maybe problems with the way in which some of the metrics were calculated. As discussed in § 4.5, a number of assumptions were made when calculating these metrics. The first is that for the binary classification measure (F1 score), anything other than exact detection when compared with the ground truth was considered a failure. This should perhaps have been altered to allow for detections a certain number of points before the true change point to be considered a successful detection.

The clustering measures were calculated by segmenting the data around the detected change points and the true change points, and forming those segmentations into clusters. This method was chosen due to the segmentation approaches of the algorithms being evaluated, but it may well be that there is a more effective method of creating this clusterings for comparation - perhaps taking into account the magnitude as well as the time index of each point.

Chapter 8

Future Work

The results of this thesis open up numerous avenues for future research in the field. The validity issues discussed in § 7.3 suggest that changes to the experimental method could result in more interesting results.

It would also be interesting to formulate the 'ideal metric' discussed in § 7.2 mathematically, and prove it's veracity through simulation studies and analysis of real world data, much like that which was carried out in this thesis.

Further research should also be undertaken using much larger data sets, and more numerous instances of these data sets. One of the main threats to validity for this project is the small sample size of the real world data corpus, and this is something that, if resolved, would result in a considerable amount of results for analysis. Further, with a large enough test corpus, it would be possible to conduct statistical significance testing between the results, to evaluate exactly how much measures disagree with each other, if at all.

While this thesis deals specifically with change point detection approaches, it would be valuable to compare the performance of these approaches with machine learning and natural language processing approaches. Analysis techniques such as sentiment analysis have been shown in the past to be good methods of detecting 'events' occurring on social media, and it may well be that those approaches are far more effective in this domain. The Text Retrieval Conference of 2016 carried out an evaluation of real-time summarisation approaches [47], and supplied a number of scoring methods which were used as part of this evaluation. It would be useful to conduct a replication study utilising a corpus of data taken from the host company of this thesis and apply the approaches submitted to that track to it. In that way, the effectiveness of content summarisation studies could be evaluated as a method for detecting anomalous changes in subject, rather than just conversation volume.

Bibliography

[1] Hirotugu Akaike. "A New Look at the Statistical Model Identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. ISSN: 15582523. DOI: 10.1109/TAC.1974.1100705. arXiv: arXiv:1011.1669v3.

[2] Foteini Alvanaki et al. "EnBlogue: emergent topic detection in Web 2.0 streams". In: *Proc. ACM SIGMOD International Conference on Management of Data* (2011), pp. 1271–1274. ISSN: 07308078. DOI: 10.1145/1989323.1989473. URL: http://doi.acm.org/10.1145/1989323.1989473%7B%5C%%7D5Cnpapers2://publication/uuid/44E009D1-8574-49C1-A0AC-C2B247FE9043.

[3] Enrique Amigó et al. "A comparison of extrinsic clustering evaluation metrics based on formal constraints". In: *Information Retrieval* 12.4 (2009), pp. 461–486. ISSN: 13864564. DOI: 10.1007/s10791-008-9066-8.

[4] Ivan E. Auger and Charles E. Lawrence. "Algorithms for the optimal identification of segment neighborhoods". In: *Bulletin of Mathematical Biology* 51.1 (Jan. 1989), pp. 39–54. ISSN: 00928240. DOI: 10.1007/BF02458835. URL: http://link.springer.com/10.1007/BF02458835.

[5] Amit Bagga and Breck Baldwin. "Entity-based cross-document coreferencing using the vector space model". In: *Proceedings of the 17th international conference on Computational linguistics* -. Vol. 1. Morristown, NJ, USA: Association for Computational Linguistics, 1998, pp. 79–85. DOI: 10.3115/980451.980859. URL: http://portal.acm.org/citation.cfm?doid=980451.980859.

[6] M Basseville and Igor V Nikiforov. *Detection of Abrupt Changes: Theory and Application*. 1993. ISBN: 0-13-126780-9. DOI: 10.1016/0967-0661(94)90196-1. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.6896%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf.

[7] S. Bersimis, S. Psarakis, and J. Panaretos. "Multivariate statistical process control charts: An overview". In: *Quality and Reliability Engineering International* 23.5 (2007), pp. 517–543. ISSN: 07488017. DOI: 10.1002/qre.829.

[8] J V Braun, R K Braun, and H-G Müller. "Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation". In: *Biometrika* 87.2 (June 2000), pp. 301–314. ISSN: 00063444. DOI: 10.1093/biomet/87.2.301. URL: https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/87.2.301.

[9] Cody Buntain, Christopher Natoli, and Miroslav Zivkovic. "A Brief Comparison of Algorithms for Detecting Change Points in Data". In: *Supercomputing*. 2014. URL: https://github.com/cbuntain/ChangePointDetection.

[10] Chen and W.-C. *Overlapping Codon Model, Phylogenetic Clustering, and Alternative Partial Expectation Conditional Maximization Algorithm*. 2011. URL: http://gradworks.umi.com/34/73/3473002.html.

[11] Tamraparni Dasu et al. "Change (detection) you can believe in: Finding distributional shifts in data streams". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5772 LCNS (2009), pp. 21–34. ISSN: 03029743. DOI: 10.1007/978-3-642-03915-7_3.

[12] Frédéric Desobry, Manuel Davy, and Christian Doncarli. "An online Kernel change detection algorithm". In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2961–2974. ISSN: 1053587X. DOI: 10.1109/TSP.2005.851098.

[13] Allen B. Downey. "A novel changepoint detection algorithm". In: *Applied Microbiology and Biotechnology* (2008), pp. 1–11. arXiv: 0812.1237. URL: http://arxiv.org/abs/0812.1237.

[14] I.A. Eckley, P. Fearnhead, and R. Killick. "Analysis of Changepoint Models". In: *Bayesian Time Series Models* January (2011), pp. 205–224. DOI: 10.1017/CBO9780511984679.011.

[15] Tom Fawcett and Foster Provost. "Activity monitoring: Noticing interesting changes in behavior". In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* 1.212 (1999), pp. 53–62. ISSN: 09258574. DOI: 10.1016/j.ecoleng.2010.11.031. URL: http://portal.acm.org/citation.cfm?id=312195.

[16] Pedro Galeano and Daniel Peña. "Variance Changes Detection in Multivariate Time Series". 2004.

[17] Carlos J Gil, Bellosta Maintainer, and Carlos J Gil Bellosta. *'rPython' Package Allowing R to Call Python*. 2015. URL: https://cran.r-project.org/web/packages/rPython/rPython.pdf.

[18] Jeremy Ginsberg et al. "Detecting influenza epidemics using search engine query data." In: *Nature* 457.7232 (2009), pp. 1012–4. ISSN: 1476-4687. DOI: 10.1038/nature07634. URL: http://www.ncbi.nlm.nih.gov/pubmed/19020500.

[19] E J Hannan and B G Quinn. "The Determination of the Order of an Autoregression". In: *Journal of the Royal Statistical Society* 41.2 (1979), pp. 190–195. ISSN: 00359246. DOI: 10.2307/2985032. URL: http://www.jstor.org/stable/2985032.

[20] Kaylea Haynes, Idris A. Eckley, and Paul Fearnhead. "Efficient penalty search for multiple changepoint problems". In: (2014), pp. 1–23. arXiv: 1412.3617. URL: http://arxiv.org/abs/1412.3617.

[21] Hugo Hromic. *python-bcubed*. 2016. URL: https://github.com/hhromic/python-bcubed.

[22] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 01764268. DOI: 10.1007/BF01908075. URL: http://link.springer.com/10.1007/BF01908075.

[23] Brad Jackson et al. "An algorithm for optimal partitioning of data on an interval". In: *IEEE Signal Processing Letters* 12.2 (Sept. 2005), pp. 105–108. ISSN: 10709908. DOI: 10.1109/LSP.2001.838216. arXiv: 0309285 [math]. URL: http://arxiv.org/abs/math/0309285%20http://dx.doi.org/10.1109/LSP.2001.838216.

[24] Yoshinobu Kawahara and Masashi Sugiyama. "Change-point detection in time-series data by direct density-ratio estimation". In: *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), pp. 389–400.

[25] MG Kendall. "A new measure of rank correlation". In: *Biometrika* 30.1 (June 1938), pp. 81–93. ISSN: 0006-3444. DOI: 10.1093/biomet/30.1-2.81. URL: https://academic.oup.com/biomet/article-lookup/doi/10.1093/biomet/30.1-2.81%20http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/30.1-2.81%7B%5C%%7D5Cnhttp://www.jstor.org/stable/2332226.

[26] Allen Kent et al. "Machine literature searching VIII. Operational criteria for designing information retrieval systems". In: *American Documentation* 6.2 (Apr. 1955), pp. 93–101. ISSN: 0096946X. DOI: 10.1002/asi.5090060209. URL: http://doi.wiley.com/10.1002/asi.5090060209.

[27] Daniel Kifer, Shai Ben-david, and Johannes Gehrke. "Detecting Change in Data Streams". In: *Proceedings of the 30th VLDB Conference* (2004), pp. 180–191. ISSN: 00394564. DOI: 10.1016/0378-4371(94)90421-9. arXiv: 9310008 [cond-mat].

[28] R Killick, P Fearnhead, and I A Eckley. "Optimal detection of changepoints with a linear computational cost". In: *Journal of the American Statistical Association* 107 (2011), pp. 500–1590. ISSN: 0162-1459. DOI: 10.1080/01621459.2012.737745. arXiv: 1101.1438. URL: http://amstat.tandfonline.com/action/journalInformation?journalCode=uasa20%20http://dx.doi.org/10.1080/01621459.2012.737745%20http://arxiv.org/abs/1101.1438%7B%5C%%7D0Ahttp://dx.doi.org/10.1080/01621459.2012.737745.

[29] Rebecca Killick and Idris A. Eckley. "changepoint: An R Package for Changepoint Analysis". In: *Journal of Statistical Software* 58.3 (2014), pp. 1–19. DOI: 10.18637/jss.v058.i03. URL: http://www.jstatsoft.org/v58/i03/.

[30] Rebecca Killick, Idris Eckley, and Philip Jonathan. "Efficient Detection Of Multiple Changepoints Within An Oceanographic Time Series". In: (2011), pp. 4137–4142. URL: http://www.lancaster.ac.uk/%7B~%7Djonathan/2011%7B%5C_%7DISI%7B%5C_%7DChangePoints%7B%5C_%7DTalk.pdf%20http://eprints.lancs.ac.uk/56978/.

[31] Max Kuhn et al. *Caret: Classification and Regression Training*. 2012. URL: https://cran.r-project.org/web/packages/caret/caret.pdf.

[32] Martin Kulldorff et al. "A space-time permutation scan statistic for disease outbreak detection". In: *PLoS Medicine* 2.3 (2005), pp. 0216–0224. ISSN: 15491277. DOI: 10.1371/journal.pmed.0020059.

[33] Tze Leung Lai and Jerry Zhaolin Shan. "Efficient recursive algorithms for detection of abrupt changes in signals and control systems". In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 952–966. ISSN: 00189286. DOI: 10.1109/9.763211.

[34] David S. Matteson and Nicholas A. James. "A nonparametric approach for multiple change point analysis of multivariate data". In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI: 10.1080/01621459.2013.849605. arXiv: 1306.4933.

[35] Thomas J. McCabe. "A Complexity Measure". In: *IEEE Trans. Softw. Eng.* 2.4 (1976), pp. 308–320. URL: http://juacompe.mrchoke.com/natty/thesis/FrameworkComparison/A%20complexity%20measure.pdf.

[36] E. S. Page. "Continuous Inspection Schemes". In: *Biometrika* 41.1/2 (June 1954), p. 100. ISSN: 00063444. DOI: 10.2307/2333009. URL: http://www.jstor.org/stable/2333009?origin=crossref.

[37] Anita M Pelecanos, Peter a Ryan, and Michelle L Gatton. "Outbreak detection algorithms for seasonal disease data: a case study using Ross River virus disease." In: *BMC medical informatics and decision making* 10.1 (2010), p. 74. ISSN: 1472-6947. DOI: 10.1186/1472-6947-10-74. URL: http://www.biomedcentral.com/1472-6947/10/74.

[38] Abdulhakim A. Qahtan et al. "A PCA-Based Change Detection Framework for Multidimensional Data Streams". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pp. 935–944. DOI: 10.1145/2783258.2783359. URL: http://dl.acm.org/citation.cfm?id=2783258.2783359.

[39] R Core Development Team. *R: a language and environment for statistical computing, 3.2.1*. R Foundation for Statistical Computing. Vienna, Austria, 2015. ISBN: 3-900051-07-0. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3. URL: https://www.r-project.org/.

[40] William M. Rand. "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336 (Dec. 1971), pp. 846–850. ISSN: 0162-1459. DOI: 10.1080/01621459.1971.10482356. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356.

[41] C. J. van Rijsbergen. *Information retrieval*. Butterworths, 1979, p. 208. ISBN: 0408709294. URL: https://dl.acm.org/citation.cfm?id=539927.

[42] Gideon Schwarz. "Estimating the Dimension of a Model". In: *The Annals of Statistics* 6.2 (1978), pp. 461–464. ISSN: 0090-5364. DOI: 10.1214/aos/1176344136. URL: http://projecteuclid.org/euclid.aos/1176344136.

[43] D. Siegmund and E. S. Venkatraman. "Using the generalized likelihood ratio statistic for sequential detection of a change-point". In: *The Annals of Statistics* 23.1 (1995), pp. 255–271. ISSN: 0090-5364. DOI: 10.1214/aos/1176324466.

[44] T. Sing et al. "ROCR: visualizing classifier performance in R". In: *Bioinformatics* 21.20 (Oct. 2005), pp. 3940–3941. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti623. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti623.

[45] Alexander G Tartakovsky and Boris L Rozovskii. "A Nonparametric Multichart CUSUM Test for Rapid Intrusion Detection". In: *Proceedings of Joint Statistical Meetings* (2005), pp. 7–11.

[46] Alexander G. Tartakovsky et al. "Detection of intrusions in information systems by sequential change-point methods". In: *Statistical Methodology* 3.3 (2006), pp. 252–293. ISSN: 15723127. DOI: 10.1016/j.stamet.2005.05.003.

[47] Text Retrieval Conference 2016. *TREC 2016 Evaluation Guidelines*. URL: https://trecrts.github.io/TREC2016-RTS-guidelines.html (visited on 07/06/2017).

[48] Dang-Hoan Tran, Mohamed Medhat Gaber, and Kai-Uwe Sattler. "Change Detection in Streaming Data in the Era of Big Data: Models and Issues". In: *ACM SIGKDD Explorations Newsletter - Special issue on big data* 1 (2014), pp. 30–38. ISSN: 1931-0145. DOI: 10.1145/2674026.2674031.

[49] Alan S. Willsky and Harold L. Jones. "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems". In: *IEEE Transactions on Automatic Control* 21.1 (1976), pp. 108–112. ISSN: 15582523. DOI: 10.1109/TAC.1976.1101146.

[50] Yi Xu et al. "Pattern change discovery between high dimensional data sets". In: *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11* (2011), p. 1097. DOI: 10.1145/2063576.2063735. URL: http://dl.acm.org/citation.cfm?doid=2063576.2063735.

[51] Yi-Ching Yao. "Estimation of a Noisy Discrete-Time Step Function: Bayes and Empirical Bayes Approaches". In: *The Annals of Statistics* 12.4 (Dec. 1984), pp. 1434–1447. ISSN: 0090-5364. DOI: 10.1214/aos/1176346802. URL: http://projecteuclid.org/euclid.aos/1176346802.

Appendix A

Pseudocode for Simulation Studies

Algorithm 4: Experiment to plot the effect of increasing 'head' length of a dataset on metrics

```
begin
    foreach iteration do
        Add 1 point to head of data, increasing length
        Calculate metrics for current τ, τ′ against time series
        Store results
    end
end
```

Algorithm 5: Experiment to plot the effect of increasing 'tail' length of a dataset on metrics

```
begin
    foreach iteration do
        Add 1 point to tail of data, increasing length
        Calculate metrics for current τ, τ′ against time series
        Store results
    end
end
```

Algorithm 6: Experiment to plot the effect of increasing 'tail' length of a dataset on metrics

```
begin
    foreach iteration do
        Move τ and τ′ 1 point further in time series
        Calculate metrics for current τ, τ′ against time series
        Store results
    end
end
```

Algorithm 7: Experiment to plot the effect of temporal distance between τ and τ' on metrics

begin
  foreach *iteration* **do**
    Leave τ unchanged
    $\tau' \leftarrow \tau' + 1$
    Calculate metrics for current τ, τ' against time series
    Store results
  end
end

Algorithm 8: Experiment to plot the effect of increasing the number of false positive detections on metrics

begin
  foreach *iteration* **do**
    Add 1 additional false positive, such that $\tau'_{n+1} = \tau'_n + 5$
    Calculate metrics for current τ, τ' against time series
    Store results
  end
end

Algorithm 9: Experiment to plot the effect of decreasing the number of false negative detections on metrics

begin
  foreach *iteration* **do**
    Add one additional τ' such that $\tau'_n = \tau_n$
    Calculate metrics for current τ, τ' against time series
    Store results
  end
end

Algorithm 10: Experiment to plot the effect of the number of change points on metrics, by appending

begin
  new_point $\leftarrow$ time series segment with 1 changepoint
  foreach *iteration* **do**
    Append *new_point* to data, increasing length
    Add additional τ at tau_{new_point}
    Add additional τ' at $tau'new\_point$
    Calculate metrics for current τ, τ' against time series
    Store results
  end
end

Algorithm 11: Experiment to plot the effect of the number of change points on metrics, by adding 1 to slices of the time series & maintaining constant time series length

```
1 begin
2     foreach iteration do
3         Add 1 to next slice of time series
4         Add τ and τ′ such that τ′ = τ + 2
5         Calculate metrics for current τ, τ′ against time series
6         Store results
7     end
8 end
```

Appendix B

Ground Truth Annotations

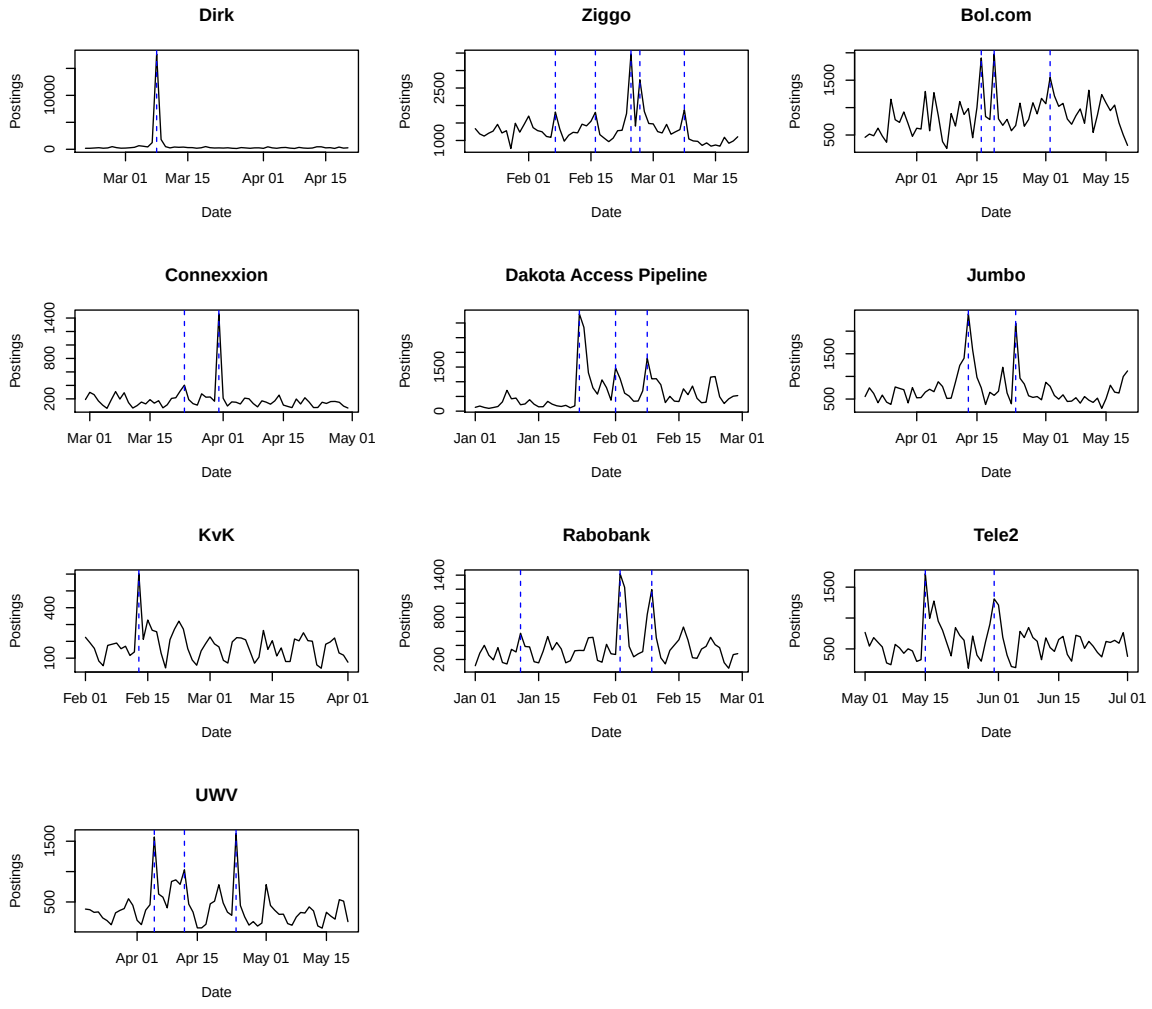

Figure B.1: Unadjusted Ground Truth Plots

Appendix C

Real-World Change Point Detection Plots

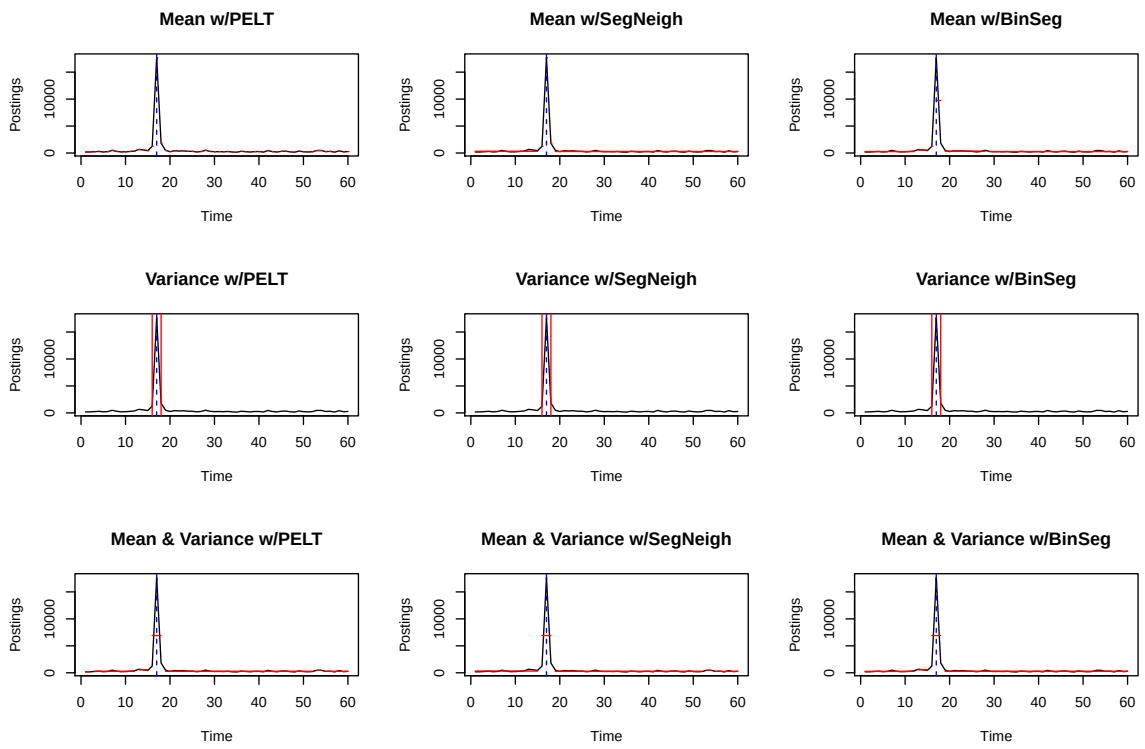

Figure C.1: 'Dirk' Change Point Detections

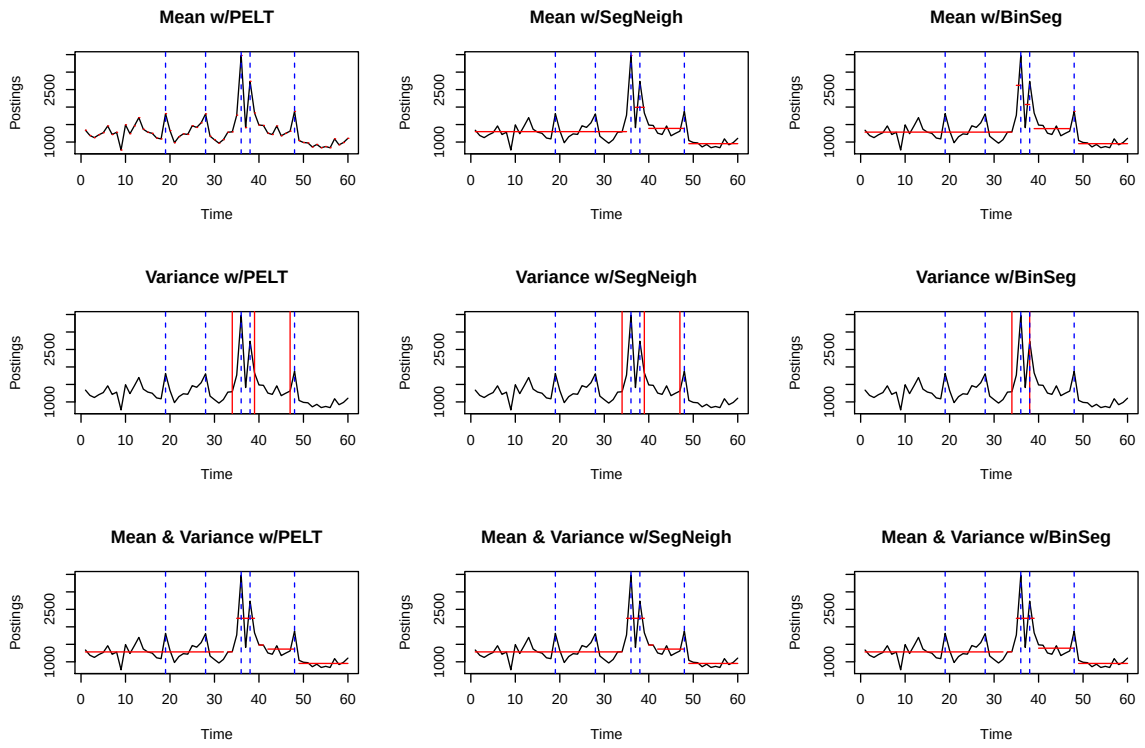

Figure C.2: 'Dirk' Change Point Detections

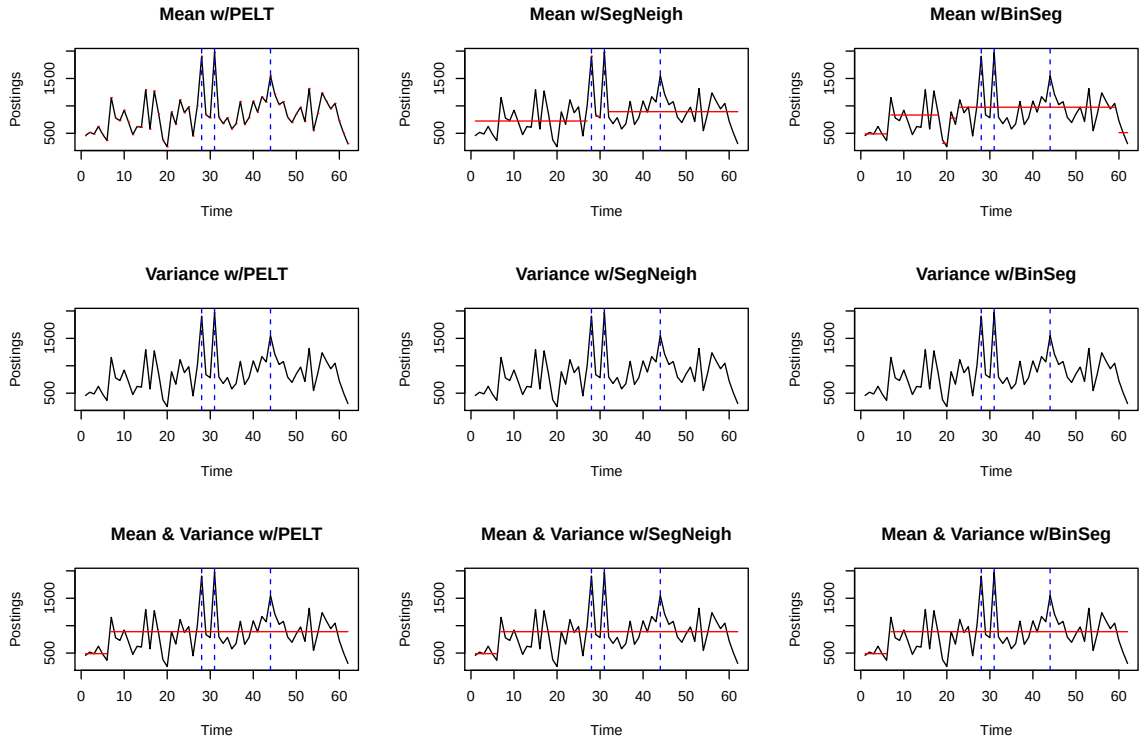

Figure C.3: 'Dirk' Change Point Detections

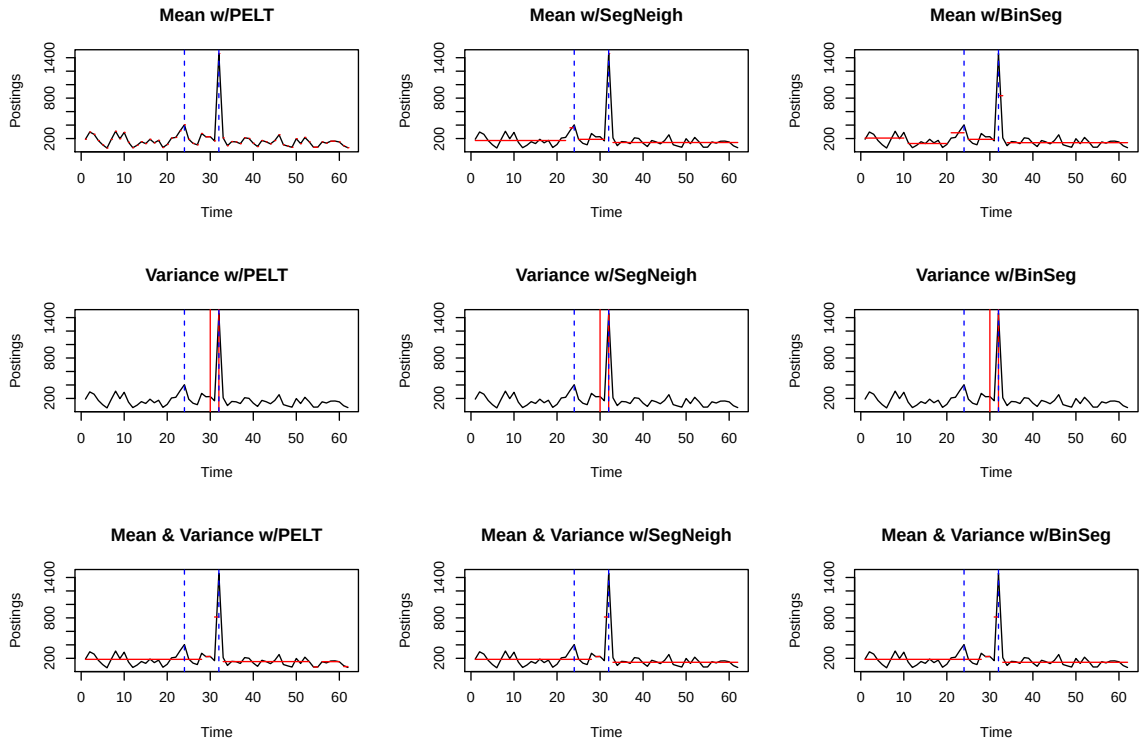

Figure C.4: 'Dirk' Change Point Detections

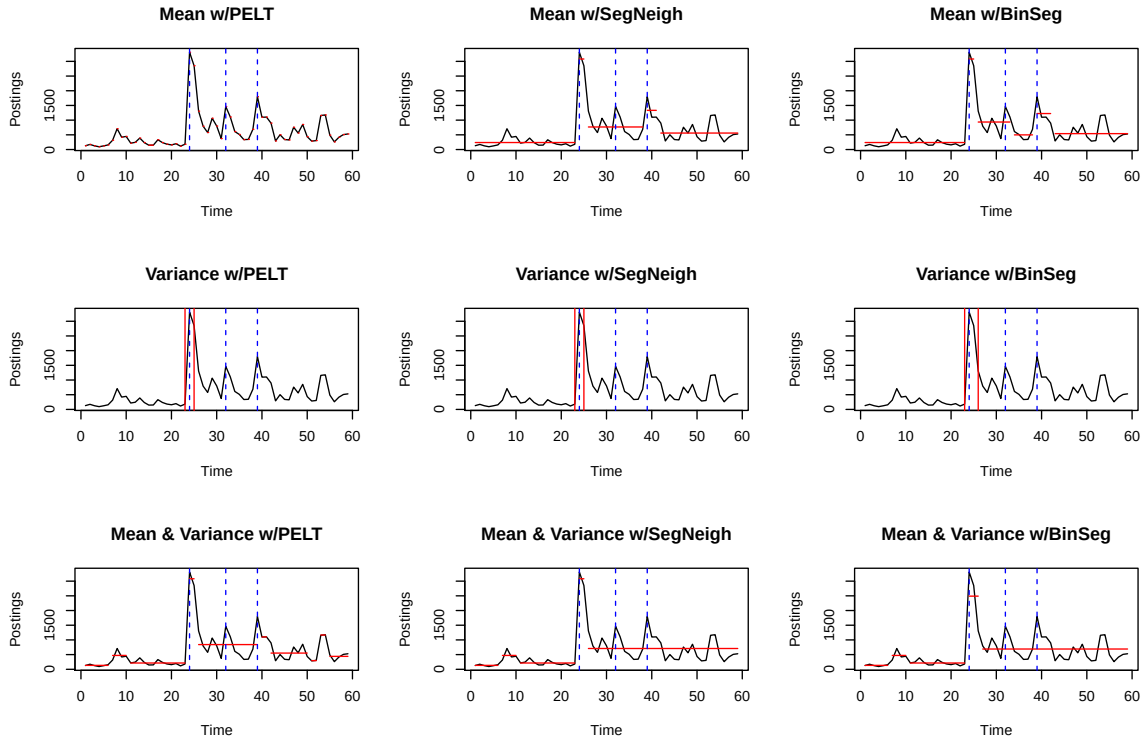

Figure C.5: 'Dirk' Change Point Detections

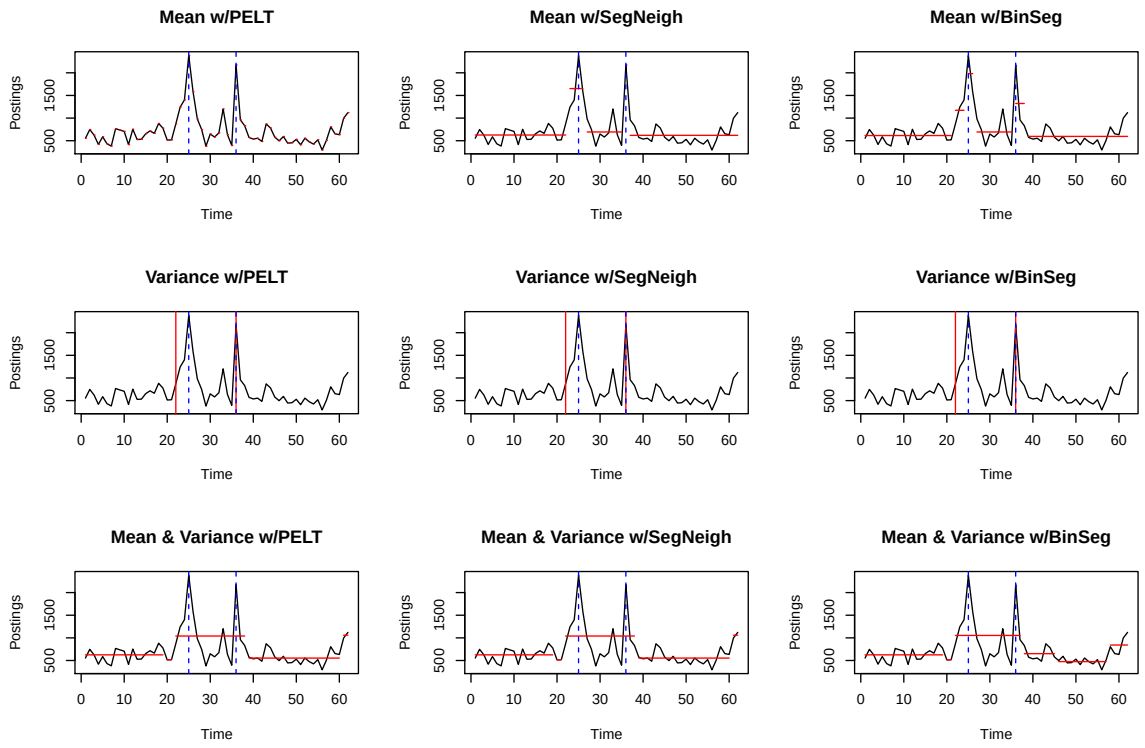

Figure C.6: 'Dirk' Change Point Detections

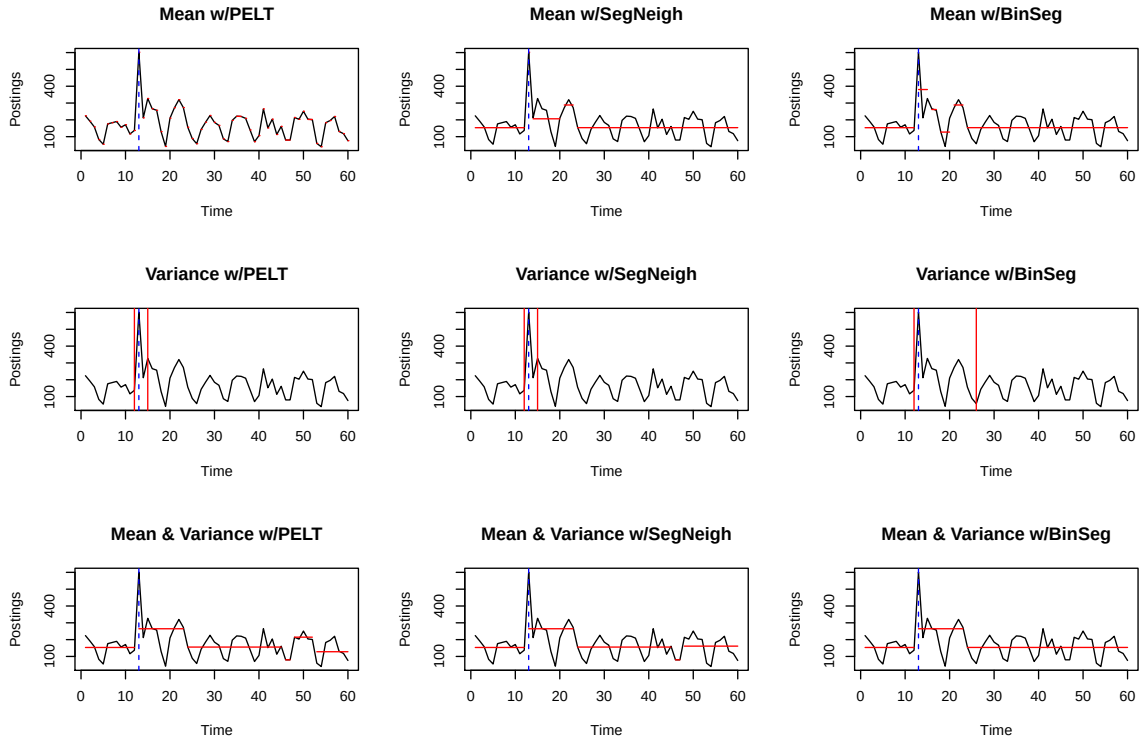

Figure C.7: 'Dirk' Change Point Detections

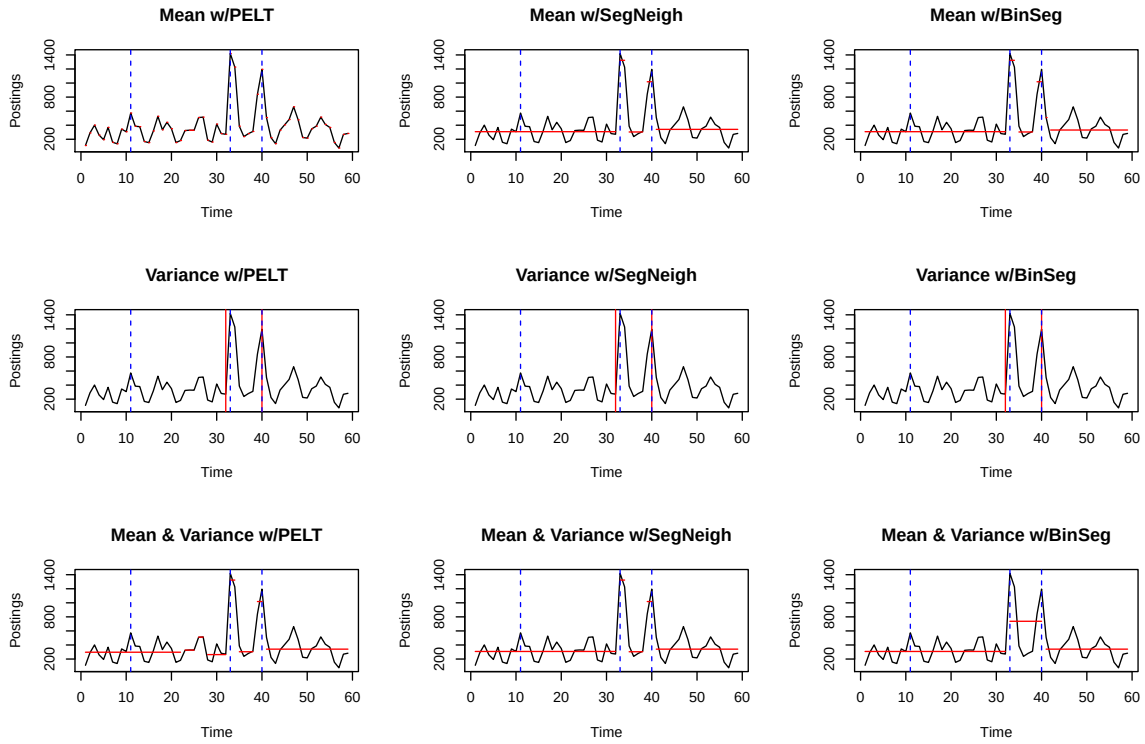

Figure C.8: 'Dirk' Change Point Detections

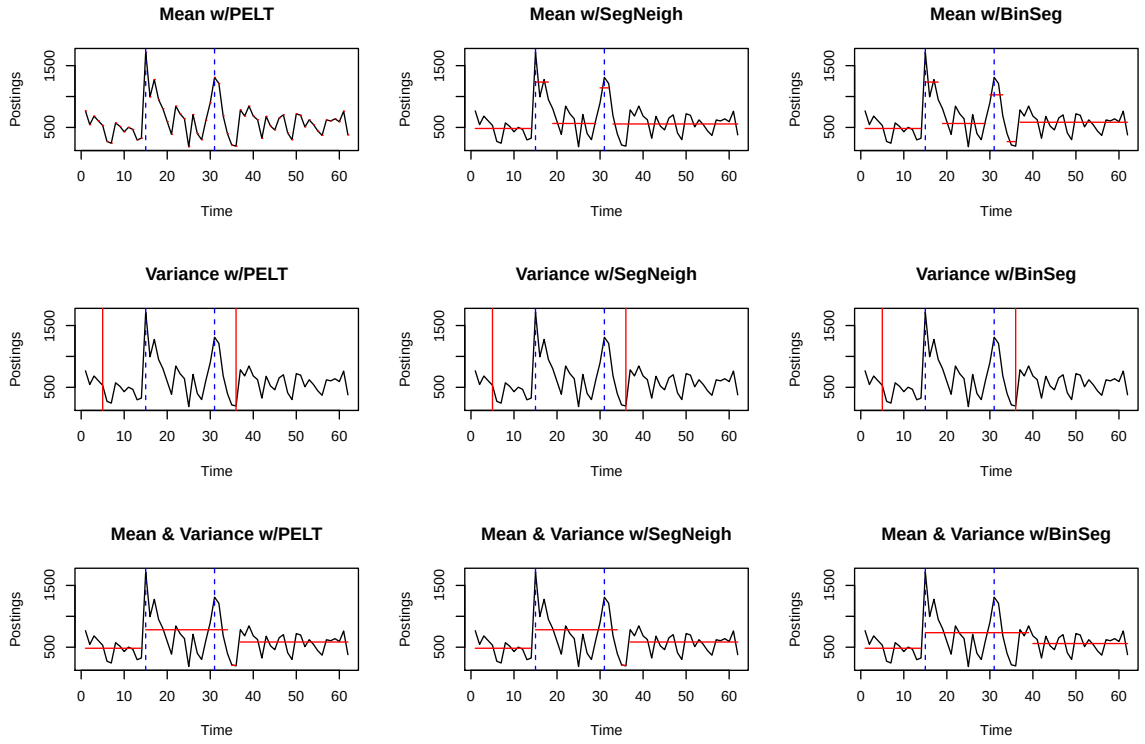

Figure C.9: 'Dirk' Change Point Detections

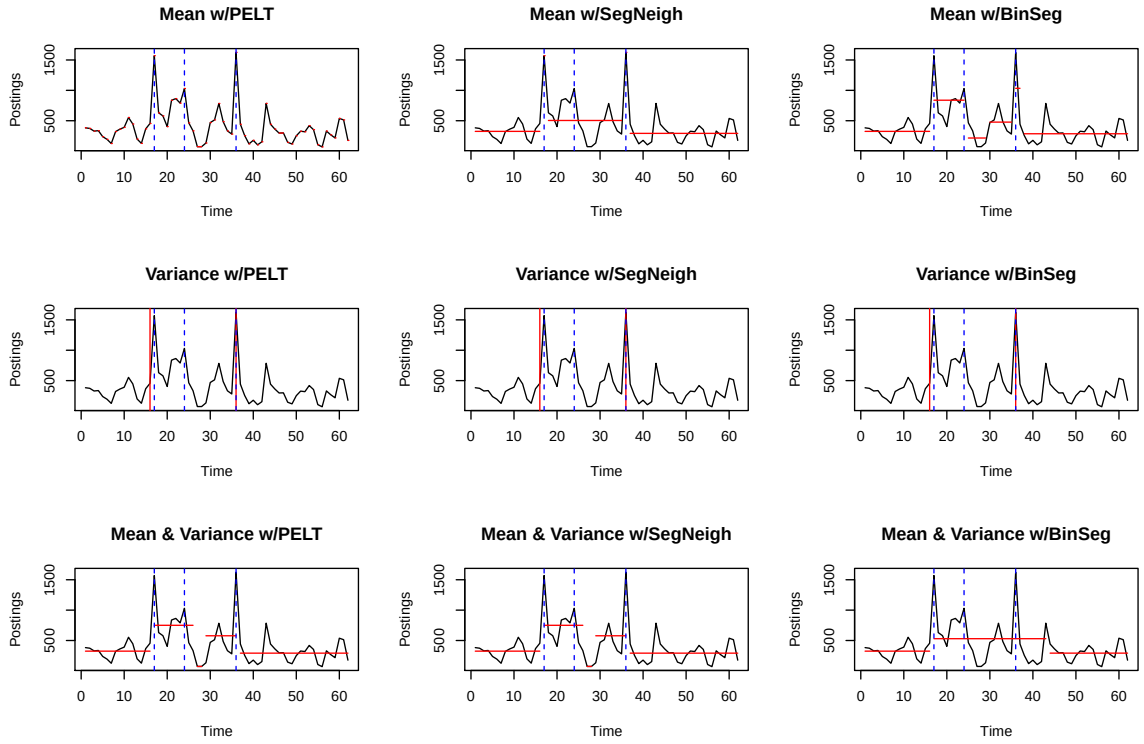

Figure C.10: 'Dirk' Change Point Detections

Appendix D

Full Real World Data Scorings

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.916384181	0.831326108	0.977777778	0.808522727	0.885131649
MeanVar SegNeigh	0	0	NA	0.774576271	0.57039828	0.977777778	0.662121212	0.789569859
MeanVar PELT	0	0	NA	0.750282486	0.528564301	0.977777778	0.572537879	0.722194622
Var BinSeg	0	0	NA	**0.942937853**	**0.883671294**	**0.983333333**	**0.904356061**	**0.942192569**
Var SegNeigh	0	0	NA	**0.942937853**	**0.883671294**	**0.983333333**	**0.904356061**	**0.942192569**
Var PELT	0	0	NA	**0.942937853**	**0.883671294**	**0.983333333**	**0.904356061**	**0.942192569**
Mean BinSeg	0	0	NA	0.73559322	0.503863154	**0.983333333**	0.564015152	0.716858425
Mean SegNeigh	**0.2**	**1**	**0.333333333**	0.93559322	0.8691606	1	0.875189394	0.933441067
Mean PELT	0.017241379	**1**	0.033898305	0.398305085	0.000746372	1	0.034090909	0.065934066

Table D.1: Scores for 'Dirk' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0.2	0.2	0.2	**0.819774011**	0.54185782	0.656344086	0.855897436	0.742954366
MeanVar SegNeigh	0.2	0.2	0.2	0.785875706	0.477814329	0.619393939	0.842564103	0.713945385
MeanVar PELT	0.166666667	0.2	0.181818182	0.811864407	**0.515766487**	0.656344086	0.809230769	0.724812967
Var BinSeg	0.333333333	0.2	0.25	0.733333333	0.418172701	0.457575758	**0.919230769**	0.611004825
Var SegNeigh	0	0	NA	0.782485876	0.477512609	0.588624709	0.862564103	0.699738779
Var PELT	0	0	NA	0.782485876	0.477512609	0.588624709	0.862564103	0.699738779
Mean BinSeg	**0.5**	**0.6**	**0.545454545**	0.798305085	0.510612723	**0.672727273**	0.889230769	**0.765974212**
Mean SegNeigh	0.4	0.4	0.4	0.784745763	0.492259321	0.633660131	0.910064103	0.747117038
Mean PELT	0.083333333	1	0.153846154	0.807344633	0	1	0.1	0.181818182

Table D.2: Scores for 'Ziggo' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
MeanVar SegNeigh	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
MeanVar PELT	0	0	NA	0.363828662	-0.033083215	0.380760369	0.838017984	0.523612904
Var BinSeg	0	0	NA	0.332099418	0.002293403	0.341618191	**0.969439728**	0.505207652
Var SegNeigh	0	0	NA	0.332099418	0.002293403	0.341618191	**0.969439728**	0.505207652
Var PELT	0	0	NA	0.332099418	0.002293403	0.341618191	**0.969439728**	0.505207652
Mean BinSeg	0	0	NA	0.584346906	0.100269497	0.594594595	0.584795322	0.589654248
Mean SegNeigh	**0.4**	0.666666667	**0.5**	**0.851930196**	**0.685228297**	**0.756503642**	0.916871031	**0.829002955**
Mean PELT	0.048387097	**1**	0.092307692	0.6811211	0	1	0.064516129	0.121212121

Table D.3: Scores for 'Bol.com' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0.25	0.5	0.333333333	0.924907456	0.844343916	0.890083632	0.888137357	0.889109429
MeanVar SegNeigh	0.25	0.5	0.333333333	0.924907456	0.844343916	0.890083632	0.888137357	0.889109429
MeanVar PELT	0.142857143	0.5	0.222222222	0.812268641	0.590695386	0.890083632	0.666493236	0.762229907
Var BinSeg	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	**0.920395421**	0.881900364
Var SegNeigh	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	**0.920395421**	0.881900364
Var PELT	0.333333333	0.5	0.4	0.904812269	0.805123425	0.846496107	**0.920395421**	0.881900364
Mean BinSeg	0.166666667	0.5	0.25	0.87678477	0.727378477	0.983870968	0.677560738	0.802479376
Mean SegNeigh	**0.4**	**1**	**0.571428571**	**0.958223162**	**0.910909287**	**1**	0.881650681	**0.937103459**
Mean PELT	0.033898305	**1**	0.06557377	0.607086198	0.004865842	**1**	0.05450052	0.10336746

Table D.4: Scores for 'Connexxion' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.7106955	0.351202345	0.667180277	0.682527985	0.674766871
MeanVar SegNeigh	0	0	NA	0.698421975	0.334534305	0.655034895	0.703714426	0.678502647
MeanVar PELT	**0.1**	0.333333333	**0.153846154**	0.779661017	0.378254169	0.86440678	0.476111521	0.614022242
Var BinSeg	0	0	NA	0.790765634	0.562324673	0.667180277	0.884443977	0.760601126
Var SegNeigh	0	0	NA	0.77849211	0.543596211	0.655034895	**0.905630417**	0.760213636
Var PELT	0	0	NA	0.77849211	0.543596211	0.655034895	**0.905630417**	0.760213636
Mean BinSeg	0	0	NA	**0.928696669**	**0.819906546**	0.927966102	0.770037197	**0.841657276**
Mean SegNeigh	0	0	NA	**0.920514319**	0.805466429	0.850934376	0.818463347	0.834383069
Mean PELT	0.051724138	**1**	0.098360656	0.701344243	0.002735198	**1**	0.06927045	0.129565817

Table D.5: Scores for 'Dakota Access Pipeline' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.776837652	0.464020988	0.881048387	0.548984468	0.676462612
MeanVar SegNeigh	0	0	NA	0.837123215	0.627783501	0.859582543	0.703106332	0.773510247
MeanVar PELT	0	0	NA	0.837123215	0.627783501	0.859582543	0.703106332	0.773510247
Var BinSeg	0.333333333	0.5	**0.4**	0.935483871	0.85755302	0.923963134	**0.884259259**	**0.903675299**
Var SegNeigh	0.333333333	0.5	**0.4**	0.935483871	0.85755302	0.923963134	**0.884259259**	**0.903675299**
Var PELT	0.333333333	0.5	**0.4**	0.935483871	0.85755302	0.923963134	**0.884259259**	**0.903675299**
Mean BinSeg	0	0	NA	0.905341089	0.782728087	0.962365591	0.744311394	0.839408606
Mean SegNeigh	0.2	0.5	0.285714286	**0.941300899**	**0.868296668**	**0.975806452**	0.828540784	0.896163916
Mean PELT	0.032258065	**1**	0.0625	0.639344262	0	1	0.048387097	0.092307692

Table D.6: Scores for 'Jumbo' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.752542373	0.525490081	0.96969697	0.679861111	0.799316213
MeanVar SegNeigh	0	0	NA	0.551412429	0.241900508	0.96969697	0.432638889	0.598328306
MeanVar PELT	0	0	NA	0.528813559	0.214187614	0.96969697	0.404861111	0.571227361
Var BinSeg	0	0	NA	0.710169492	0.458374641	0.969047619	0.629861111	0.763477488
Var SegNeigh	0	0	NA	**0.915254237**	**0.818859861**	0.977777778	**0.874305556**	**0.923151273**
Var PELT	0	0	NA	**0.915254237**	**0.818859861**	0.977777778	**0.874305556**	**0.923151273**
Mean BinSeg	0	0	NA	0.736158192	0.502540289	0.977777778	0.654166667	0.783886525
Mean SegNeigh	**0.2**	**1**	0.333333333	0.746327684	0.519210875	**1**	0.665277778	0.798999166
Mean PELT	0.01754386	**1**	**0.034482759**	0.327118644	0.001636731	**1**	0.035416667	0.068410463

Table D.7: Scores for 'Kamer van Koophandel' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	**0.333333333**	0.333333333	**0.333333333**	**0.849795441**	**0.667534638**	0.740705303	**0.935439137**	**0.826760168**
MeanVar SegNeigh	0.2	0.333333333	0.25	0.845119813	0.653321966	0.753417168	0.867642527	0.806505494
MeanVar PELT	0.125	0.333333333	0.181818182	0.821157218	0.536777118	**0.805488297**	0.650385208	0.719674713
Var BinSeg	**0.333333333**	0.333333333	**0.333333333**	**0.849795441**	**0.667534638**	0.740705303	**0.935439137**	**0.826760168**
Var SegNeigh	**0.333333333**	0.333333333	**0.333333333**	**0.849795441**	**0.667534638**	0.740705303	**0.935439137**	**0.826760168**
Var PELT	**0.333333333**	0.333333333	**0.333333333**	**0.849795441**	**0.667534638**	0.740705303	**0.935439137**	**0.826760168**
Mean BinSeg	0.166666667	0.333333333	0.222222222	0.834599649	0.625980308	0.753417168	0.837134052	0.793072438
Mean SegNeigh	0.2	0.333333333	0.25	0.845119813	0.653321966	0.753417168	0.867642527	0.806505494
Mean PELT	0.051724138	**1**	0.098360656	0.715955582	0.002935273	1	0.069337442	0.129682997

Table D.8: Scores for 'Rabobank' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.799048123	0.561286116	0.803870968	0.753312212	0.777770817
MeanVar SegNeigh	0	0	NA	0.86938128	0.711729799	0.891935484	**0.801699309**	0.844413523
MeanVar PELT	0	0	NA	0.86938128	0.711729799	0.891935484	**0.801699309**	0.844413523
Var BinSeg	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Var SegNeigh	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Var PELT	0	0	NA	0.740349022	0.459125414	0.698231009	0.745535714	0.721108398
Mean BinSeg	0	0	NA	0.868852459	0.699871621	0.943548387	0.678715438	0.789515055
Mean SegNeigh	0	0	NA	**0.925965098**	**0.835786141**	**0.954301075**	0.785570276	**0.861754013**
Mean PELT	**0.032258065**	**1**	**0.0625**	0.626123744	0	1	0.048387097	0.092307692

Table D.9: Scores for 'Tele2' data set

Algorithm	Precision	Recall	F1	Rand	Adjusted Rand	BCubed Precision	BCubed Recall	BCubed FScore
MeanVar BinSeg	0	0	NA	0.779481756	0.488322974	0.709677419	0.779793907	0.743085305
MeanVar SegNeigh	0.2	0.333333333	0.25	**0.947117927**	**0.869205369**	0.925806452	0.84192055	0.881873135
MeanVar PELT	0.2	0.333333333	0.25	**0.947117927**	**0.869205369**	0.925806452	0.84192055	0.881873135
Var BinSeg	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Var SegNeigh	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Var PELT	0.333333333	0.333333333	0.333333333	0.923849815	0.822681578	0.833870968	0.938694743	0.883183387
Mean BinSeg	0.166666667	0.333333333	0.222222222	0.939185616	0.845868213	0.955645161	0.798611111	**0.870099604**
Mean SegNeigh	**0.4**	0.666666667	**0.5**	0.931782126	0.838129718	0.862007168	**0.909124851**	0.88493927
Mean PELT	0.05	**1**	0.095238095	0.705975674	0.005045764	**1**	0.068399044	0.128040257

Table D.10: Scores for 'UWV' data set

Appendix E

Bibliography Annotations

What follows is an annotated bibliography of works that were read as part of the preparation for this research. It is not an exhaustive list, merely a set of papers, books and other publications which are relevant to this research or provided some insight into how best to carry it out. Full bibliographical citations are provided for each text.

[9] **Cody Buntain, Christopher Natoli, and Miroslav Zivkovic. "A Brief Comparison of Algorithms for Detecting Change Points in Data". In: *Supercomputing*. 2014. URL:** https://github.com/cbuntain/ChangePointDetection

One of the first papers on the subject read by the thesis author, this paper discusses and compares three different types of change detection algorithm: the *Likelihood Ratio Test* (LRT), the *Cumulative Sum* (CUSUM) test, and the *Kernel-based Change Detection* (KCD) algorithm. The authors of this paper also kindly provided source code implementations of these algorithms, once contact was established. The algorithms were applied to several data sets, such as historical Bitcoin valuations and data from structural stress sensors.

[24] **Yoshinobu Kawahara and Masashi Sugiyama. "Change-point detection in time-series data by direct density-ratio estimation". In: *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), pp. 389–400**

This paper proposes what the authors call a "novel non-parametric change detection algorithm"[24]. It is applied alongside four other approaches, against three different data sets generated by models borrowed from other research papers on time-series data change detection. The approaches are evaluated according to accuracy rate and degree, though not for performance or other relevant metrics.

[27] **Daniel Kifer, Shai Ben-david, and Johannes Gehrke. "Detecting Change in Data Streams". In: *Proceedings of the 30th VLDB Conference* (2004), pp. 180–191. ISSN: 00394564. DOI:** 10.1016/0378-4371(94)90421-9**. arXiv:** 9310008 [cond-mat]

This paper, much like the one written by Kawahara and Sugiyama, presents a novel change detection algorithm for evaluation. However, this particular paper differs in that the presented algorithm is also useful for estimation of the detected change. They also discuss the use of a 'two window' approach to applying change detection algorithms to data streams, so as to limit memory usage in practice. This part in particular is relevant to the business case of the organisation hosting this thesis.

[16] **Pedro Galeano and Daniel Peña. "Variance Changes Detection in Multivariate Time Series". 2004**

This paper served as the basis of two of the algorithms tested in the paper written by Buntain, Natoli, and Zivkovic This paper also provides its own evaluations of the approaches contained therein.

[12] **Frédéric Desobry, Manuel Davy, and Christian Doncarli. "An online Kernel change detection algorithm". In: *IEEE Transactions on Signal Processing* 53.8 (2005), pp. 2961–2974. ISSN: 1053587X. DOI:** 10.1109/TSP.2005.851098

Desobry, Davy, and Doncarli write concerning an implementation of a Kernel Change Detection (KCD) algorithm that is considered *online* - that is, applicable to moving and updating data sets such as those that are processed by the host organisation of this thesis. This is particularly important, as any final implementation of a change detection method in the production systems of the thesis host company will be handling online data.

This paper makes use of Receiver Operating Characteristic curves to show the accuracy of change point detection algorithms, which is a variation on standard binary classification measures concerning false/true positives/negatives.

[46] **Alexander G. Tartakovsky et al. "Detection of intrusions in information systems by sequential change-point methods". In: *Statistical Methodology* 3.3 (2006), pp. 252–293. ISSN: 15723127. DOI:** 10.1016/j.stamet.2005.05.003

This paper discusses applying a CUSUM (Cumulative Sum) approach for detection of network intrusions. The approach taken by Tartakovsky et al. was intended to research the possibility of change detection while maintaining a low rate of false alarms. The idea was to apply an algorithm for this purpose that would not need to take into account pre-change and post-change models in order to be effective.

[45] **Alexander G Tartakovsky and Boris L Rozovskii. "A Nonparametric Multichart CUSUM Test for Rapid Intrusion Detection". In: *Proceedings of Joint Statistical Meetings* (2005), pp. 7–11**

A collection of papers and discussions by the same authors as the above papers, expanding somewhat on the problem they tackled and the solutions found.

[34] **David S. Matteson and Nicholas A. James. "A nonparametric approach for multiple change point analysis of multivariate data". In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI:** 10.1080/01621459.2013.849605**. arXiv:** 1306.4933

Discussion of an 'offline' (that is, applying an algorithm to a fixed data-set as opposed to processing moving 'live' data) approach to change detection in multi-variate data. The authors carry out a simulation study to compare various approaches to this problem and present their results.

[43] **D. Siegmund and E. S. Venkatraman. "Using the generalized likelihood ratio statistic for sequential detection of a change-point". In: *The Annals of Statistics* 23.1 (1995), pp. 255–271. ISSN: 0090-5364. DOI:** 10.1214/aos/1176324466

A paper on using a *Generalized Likelihood Ratio* test for the detection of change points in a data set. An old piece of text that pre-dates social media, yet is still useful in explaining how the GLR test can be used for the detection of change points. The GLR approach is compared against standard CUSUM tests.

[49] **Alan S. Willsky and Harold L. Jones. "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems". In: *IEEE Transactions on Automatic Control* 21.1 (1976), pp. 108–112. ISSN: 15582523. DOI:** 10.1109/TAC.1976.1101146

Much like the paper written by Siegmund and Venkatraman, this paper pre-dates social media as a data source to which change detection algorithms could be applied. It is however, a useful look into how change detection algorithms have progressed and improved over the years. This paper is primarily focussed on uses of change detection in automated controls (being that it was published in the IEEE Transactions on Automatic Control), but is still a useful resource to understand how change detection algorithms can be evaluated.

[33] **Tze Leung Lai and Jerry Zhaolin Shan. "Efficient recursive algorithms for detection of abrupt changes in signals and control systems". In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 952–966. ISSN: 00189286. DOI:** 10.1109/9.763211

Another paper concerning change point detection in control systems, Lai and Shan write primarily regarding Generalised Likelihood Ratio tests, and how these can be configured, specifically with regards to window size.

[7] **S. Bersimis, S. Psarakis, and J. Panaretos. "Multivariate statistical process control charts: An overview". In: *Quality and Reliability Engineering International* 23.5 (2007), pp. 517–543. ISSN: 07488017. DOI:** 10.1002/qre.829

Control charts are a mechanism used in various industries to decide whether a given process is 'in control' or 'out of control'. In this way, control charts are used to discover outlying or anomalous results in a given data set, or inform operators of a sudden change in the data. This makes control charts particularly relevant to my research.

This particular paper discusses various approaches to control charts, and the methods behind their operation.

[13] **Allen B. Downey. "A novel changepoint detection algorithm". In: *Applied Microbiology and Biotechnology* (2008), pp. 1–11. arXiv:** 0812.1237. **URL:** http://arxiv.org/abs/0812.1237

Downey discusses his creation of a 'novel change detection algorithm' that can also be used for "...predicting the distribution of the next point in the series."[**downey2008novel**]

He discusses in some detail the difference between online and offline change detection algorithms, and compares his implementation of a new algorithm with implementations of existing and established algorithms.

[30] **Rebecca Killick, Idris Eckley, and Philip Jonathan. "Efficient Detection Of Multiple Changepoints Within An Oceanographic Time Series". In: (2011), pp. 4137–4142. URL:** http://www.lancaster.ac.uk/%7B~%7Djonathan/2011%7B%5C_%7DISI%7B%5C_%7DChangePoints%7B%5C_%7DTalk.pdf%20http://eprints.lancs.ac.uk/56978/

This paper carries out research similar in some ways to that being conducted in this thesis. Change detection methods are briefly explained and then applied to time series data from an oceanographic study. The ability for change detection approaches to correctly identify the onset and end of 'storm seasons' - which form the ground truth in this study.

[34] **David S. Matteson and Nicholas A. James. "A nonparametric approach for multiple change point analysis of multivariate data". In: *Submitted* 14853 (2012), pp. 1–29. ISSN: 0162-1459. DOI:** 10.1080/01621459.2013.849605. **arXiv:** 1306.4933

A comparison study of several change point detection methods. This study utilises the Rand Index as a measure for comparing algorithm output against a ground truth, and the Adjusted Rand Index for computing the similarity between two computed outputs. The approaches are then applied to various real world data sets from fields such as genetics and finance. This paper formed the basis for the decision to concentrate mainly on the performance of clustering metrics in the field of change point detection evaluation.

[28] **R Killick, P Fearnhead, and I A Eckley. "Optimal detection of changepoints with a linear computational cost". In: *Journal of the American Statistical Association* 107 (2011), pp. 500–1590. ISSN: 0162-1459. DOI:** 10.1080/01621459.2012.737745. **arXiv:** 1101.1438. **URL:** http://amstat.tandfonline.com/action/journalInformation?journalCode=uasa20%20http://dx.doi.org/10.1080/01621459.2012.737745%20http://arxiv.org/abs/1101.1438%7B%5C%%7D0Ahttp://dx.doi.org/10.1080/01621459.2012.737745

The paper that initially proposed the PELT algorithm for change point detection. The paper mainly concentrates on the computational cost of the PELT algorithm, comparing to other, existing methods of change point computation. Some analysis on the basis of algorithm accuracy is carried out, though limited to the use of binary classification measures.

[37] **Anita M Pelecanos, Peter a Ryan, and Michelle L Gatton. "Outbreak detection algorithms for seasonal disease data: a case study using Ross River virus disease." In: *BMC medical informatics and decision making* 10.1 (2010), p. 74. ISSN: 1472-6947. DOI:** 10.1186/1472-6947-10-74. **URL:** http://www.biomedcentral.com/1472-6947/10/74

A paper written concerning possible approaches for the detection of disease outbreaks. This publication was not interesting in terms of the approaches used, but rather in how they were evaluated. This paper restricts itself to the use of binary classification measures, tallying true/false positives and true/false negatives.

[38] **Abdulhakim A. Qahtan et al. "A PCA-Based Change Detection Framework for Multidimensional Data Streams". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15* (2015), pp. 935–944. DOI:** 10.1145/2783258.2783359. **URL:** http://dl.acm.org/citation.cfm?id=2783258.2783359

An evaluation of change point detection methods, using traditional binary classification measures. The measure results are presented purely in the form of true positives, late detections, false positives and false negatives, and thus are easily digested and if necessary translated into the usual terms of recall, precision and F1.

[3] **Enrique Amigó et al. "A comparison of extrinsic clustering evaluation metrics based on formal constraints". In: *Information Retrieval* 12.4 (2009), pp. 461–486. ISSN: 13864564. DOI:** 10.1007/s10791-008-9066-8

A detailed evaluation of BCubed as a measure for document clustering. It was based off of this paper that BCubed was chosen as an additional metric to test. BCubed has a good reputation for accuracy, and this paper backs up the ability for BCubed to be effective with detailed comparisons of it's constraints and ability to perform correctly in various situations.

[2] **Foteini Alvanaki et al. "EnBlogue: emergent topic detection in Web 2.0 streams". In: *Proc. ACM SIGMOD International Conference on Management of Data* (2011), pp. 1271–1274. ISSN: 07308078. DOI:** 10.1145/1989323.1989473. **URL:** http://doi.acm.org/10.1145/1989323.1989473%7B%5C%%7D5Cnpapers2://publication/uuid/44E009D1-8574-49C1-A0AC-C2B247FE9043

An alternative method of achieving the business goals of the host organisation of this thesis, "EnBlogue: emergent topic detection in Web 2.0 streams" provides an explanation and example implementation of a method for detecting changes in *conversation topic* as opposed to conversation volume.

[6] **M Basseville and Igor V Nikiforov. *Detection of Abrupt Changes: Theory and Application*. 1993. ISBN: 0-13-126780-9. DOI:** 10.1016/0967-0661(94)90196-1. **URL:** http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.6896%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf

The seminal text on change point detection methods and evaluation. This text is considerably old at the time of writing, but is still an invaluable source of insights and explanations as to how change point detection works, and the criteria by which they should be evaluated.

[11] **Tamraparni Dasu et al. "Change (detection) you can believe in: Finding distributional shifts in data streams". In:** ***Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*** **5772 LCNS (2009), pp. 21–34. ISSN: 03029743. DOI:** 10.1007/978-3-642-03915-7_3

An interesting text discussing the application of distributional shift detection as a change detection measure. The approach is not necessary applicable to the work being carried out in this thesis, but it is interesting to read nonetheless - considering the differences between this approach and those tested in this research.

[18] **Jeremy Ginsberg et al. "Detecting influenza epidemics using search engine query data." In:** ***Nature*** **457.7232 (2009), pp. 1012–4. ISSN: 1476-4687. DOI:** 10.1038/nature07634**. URL:** http://www.ncbi.nlm.nih.gov/pubmed/19020500

Another study in the same vein as [37], this time carried out by Google. This study uses data streams to attempt to predict influenza outbreaks with surprising effectiveness. Here, a log-likelihood approach is used to determine the probability that a random visit by a doctor is related to an illness like influenza.

[32] **Martin Kulldorff et al. "A space-time permutation scan statistic for disease outbreak detection". In:** ***PLoS Medicine*** **2.3 (2005), pp. 0216–0224. ISSN: 15491277. DOI:** 10.1371/journal.pmed.0020059

An additional study in the same category as [18] and [37], it is interesting to this project due to it's use of observed/expected cases to prove the veracity of their prediction model, in the same manner as binary classification scores are utilised in other publications.

[48] **Dang-Hoan Tran, Mohamed Medhat Gaber, and Kai-Uwe Sattler. "Change Detection in Streaming Data in the Era of Big Data: Models and Issues". In:** ***ACM SIGKDD Explorations Newsletter - Special issue on big data*** **1 (2014), pp. 30–38. ISSN: 1931-0145. DOI:** 10.1145/2674026.2674031

A paper discussing the importance of change point detection techniques when applied to *big data* scenarios. No comparison study is carried out, but the paper does serve to espouse the importance of change point detection in scenarios where the sheer *volume* of data may serve to make many approaches impractical.

[50] **Yi Xu et al. "Pattern change discovery between high dimensional data sets". In:** ***Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*** **(2011), p. 1097. DOI:** 10.1145/2063576.2063735**. URL:** http://dl.acm.org/citation.cfm?doid=2063576.2063735

This study compares various approaches for the application of event detection in data streams. It provides experiments evaluating approaches for topic change detection in documents and news streams, and also event detection in surveillance video streams. In a similar fashion to [2], it focuses mainly on using analysis techniques to spot changes in topic in documents as opposed to simply measuring volume.