Chapter 1

Identifying corruption within acoustic sensing signals

1.1 Introduction

In Chapter **??** we introduced the Clustream (Aggarwal et al., 2003) algorithm and demonstrated how it could be used to create an online Spectral Clustering algorithm. In this chapter we apply Clustream to identify corruption within acoustic sensing signals. Distributed acoustic sensing (DAS) is a modern technique used to monitor oil flow at various depths throughout an oil well. DAS uses a fibre-optic cable to record vibrations at very high resolutions, up to 10000 observations a second. DAS is fairly cheap to implement and offers high frequency data, but unfortunately corruption can occur in the signal. Our challenge is to identify the locations in the signal where corruption occurs. Existing methods for detecting and removing interference in DAS signals involve using offline, univariate changepoint detection. However DAS signals are multivariate and require online processing. In this chapter we show that

Clustream provides an alternative approach to changepoints analysis to identify corruption within DAS signals.

1.2 Motivation

1.2.1 What is Distributed Acoustic Sensing?

Distributed Acoustic Sensing (DAS) is a technique which uses fibre-optic cables to measure vibrations travelling through the ground. DAS systems have recently become popular in the oil and gas industry and are used to monitor oil flow (Silkina, 2014; van der Horst et al., 2014) and to detect leaks in abandoned gas wells (Boone et al., 2014). When vibrations pass through the fibre-optic cable, they induce a change in the intensity of the reflection of the pulses of light being passed through the cable. This provides very high frequency data, often as high as 10kHz. It is also possible to collect this data at many different depths in the well simultaneously. Therefore DAS data is very high frequency and has high dimensionality. An example of DAS signal data is given in Figure 1.2.1.

In the figure, each plot is a series obtained at a different depth within the oil well. We can see that there are some disturbances in the signal. Engineers refer to these disturbances as corrupted data and the challenge of this application is to locate where the data is corrupted. We are told that if corruption is observed at one depth then the effect is also likely to be observed at multiple other depths simultaneously. This is visible in Figure 1.2.1, particularly at time point $t = 3750$, where there is a big drop in the signal which occurs in all ten series.

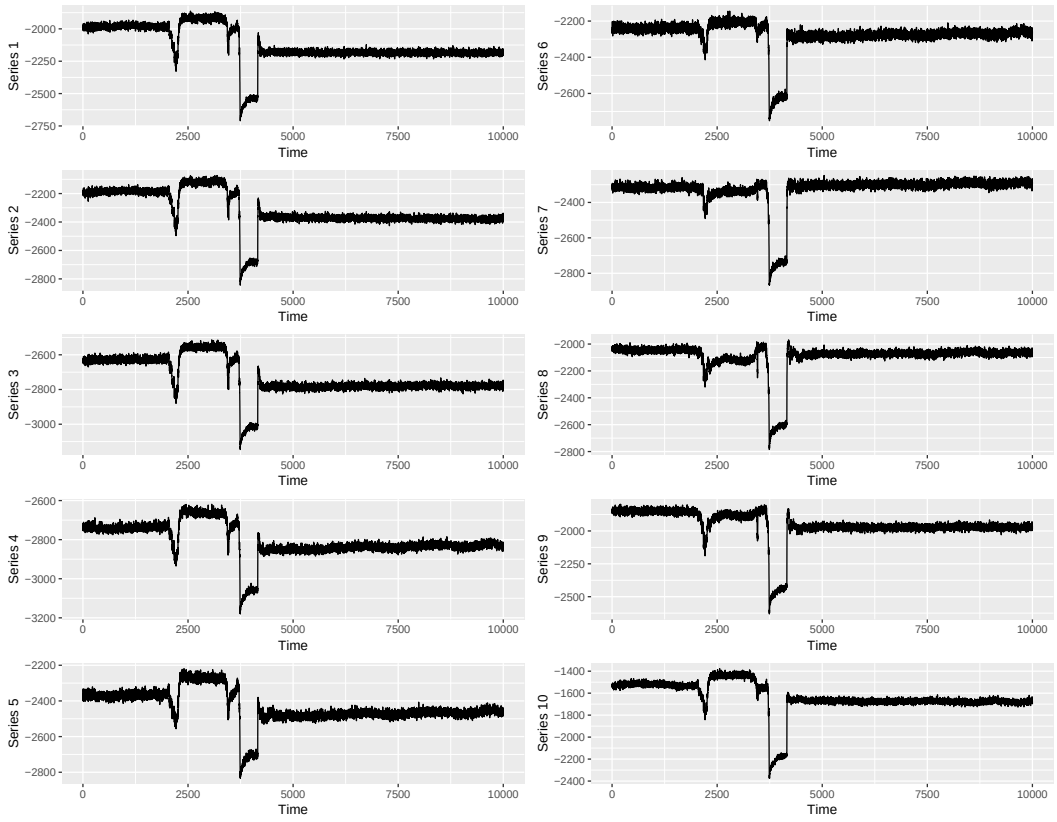


Figure 1.2.1: An example of acoustic sensing data observed at various depths in an oil well.

1.2.2 Relevant literature

Detecting corruption within a time series is usually framed as a changepoint detection problem. A changepoint is defined as a time-point at which a change occurs in one or more of the statistical properties of a time series. The first published article concerning changepoints was in Page (1954) which considered testing for a potential single changepoint and was motivated by a quality control setting in manufacturing. Over the decades, changepoint analysis has developed rapidly with multiple changepoints, different types of data and other assumptions being considered. Many methods for detecting changepoints exist ranging from approximate (heuristic) fast methods, to exact methods which take longer to run. A review of recent changepoint methods can be found in Chen and Gupta (2012); Eckley et al. (2011).

Much of the work in changepoint detection has focused on the scenario where the observations are univariate, although some extensions have been developed for the multivariate setting. The available changepoint algorithms which are multivariate cannot currently deal with the online scenario due to computational restraints. However due to the high frequency and dimensionality of DAS data, an online method is required.

1.2.3 Using Clustream to identify boundary locations

We consider the problem of identifying corruption within a DAS data stream as a two-stage clustering problem. The first stage is purely online, and consists of updating micro-clusters as a way of storing information about the data stream without storing all of the data points. The second stage is applied on a small, recent section of the data stream, and allows the user to request a segmentation of that section of the stream to look for where the signal is corrupted.

1.2.4 Stage one: Micro-clustering

Stage one is essentially the micro-clustering step of Clustream introduced in Section **??**. Clustream is a method of clustering data streams, based on the concept of micro-clusters. Micro-clusters are data structures which summarise a set of instances from the stream, and are composed of a set of statistics which are easily updated and allow fast analysis. The number of micro-clusters used is a user chosen parameter. Using a large number of micro-clusters will represent the data stream better than a smaller number, at the cost of increased computation. We found using 250 micro-clusters to be sufficient for this application.

1.2.5 Stage two: Identifying corruption

Stage two is an offline procedure which is performed in batch on a recent section of the signal. This step uses the micro-cluster summaries to identify a set B of *boundary locations*, points in the signal where there is a change in the signal. First the k-means algorithm is applied on the micro-cluster centres. The clusters generated by the k-means step are referred to as macro-clusters. Then we consider the N most recently observed data points in the signal, $\{x_1, ..., x_N\}$. Each of these N points is assigned to a k-means macro-cluster using the nearest neighbour algorithm. We can now plot the signal coloured by the macro-cluster assignments. An example of this is given in Figure 1.2.2.

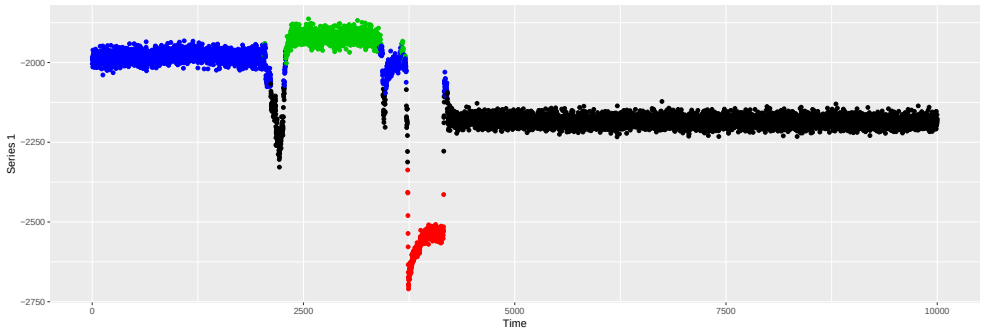
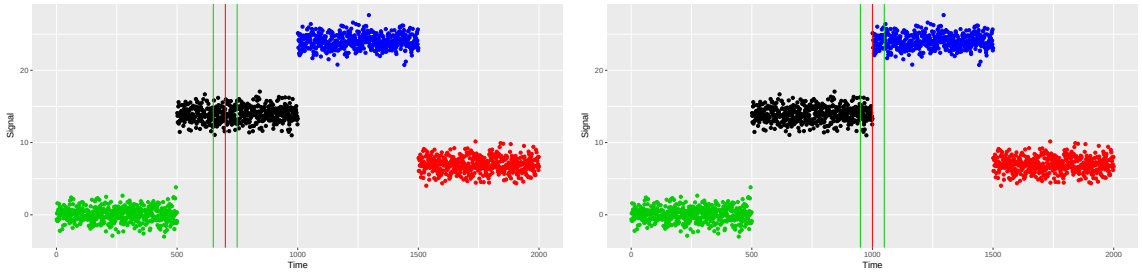


Figure 1.2.2: Series one of the DAS data coloured by macro-cluster assignments.

Visually we can get intuition of where a change in structure occurs as indicated by the change in cluster assignment. In Figure 1.2.2 this is shown as a change in the colour of the signal. However, we would like to output a set of locations in time where change occurs. Note that during the clustering steps, we do not use the timestamps as input to the clustering. This means that we do not specifically tell the clustering to consider points closer in time as more similar. As a result, there is not necessarily a clear change in the cluster assignments. The method that we use to convert the clustering of assignments into a set of boundary

locations is as follows. Consider a data point in $\{x_1, ..., x_N\}$, lets call it τ. In order to decide whether it is likely to be a boundary point we consider the k-means assignments of the data points directly before τ and directly after τ. If the assignments of those points are different, τ is likely to be a boundary location, if the assignments either side of τ are similar then τ is not likely to be a boundary location. The number of of data points we look either side of τ is given by the search parameter γ. A simple example is shown in Figure 1.2.3.



(a) $\tau = 700$, $\gamma = 50$ (b) $\tau = 1000$, $\gamma = 50$

Figure 1.2.3: Example of searching for boundary locations on a simple change in mean example. The value of τ is given by the vertical red line, and the green lines show $\tau - \gamma$ and $\tau + \gamma - 1$.

In Figure 1.2.3a, $\tau = 700$ and we can see that the 50 points before τ are the same colour as the 50 points after τ. This implies that τ is not a likely to be a boundary location. In Figure 1.2.3b, $\tau = 1000$ and we can see that the 50 points before τ are mostly different in colour to the 50 points after τ. This implies that τ is likely to be a good boundary location. In order to quantify this, we use a categorical similarity measure. Define set L to be the set of points to the left of τ given by $L = \{x_{\tau-\gamma}, \ldots, x_{\tau-1}\}$. Similarly, define set R to be the set of points to the right of τ given by $R = \{x_\tau, \ldots, x_{\tau+\gamma-1}\}$. In order to calculate how similar the cluster assignments of sets L and R are we calculate the following similarity metric. Let

$n_{L,j}$ be the number of data points in set L assigned to cluster j, where $j \in 1, \ldots, k$ and similarly for $n_{R,j}$. The categorical similarity metric is defined in equation (1.2.1).

$$\text{sim}(\tau, \gamma) = \frac{\sum_{j=1}^{k} \min(n_{L,j}, n_{R,j})}{\gamma}. \tag{1.2.1}$$

Note that this categorical similarity measure will be bound between 0 and 1, where 0 indicates perfect dissimilarity and 1 indicates that the sets are identical. We can think of this similarity measure as an indicator of how likely τ is to be a boundary location. If $\text{sim}(\tau, \gamma) = 0$ then τ is very likely to be a boundary location. We search over all values of $\tau \in \{\gamma + 1 : N - \gamma + 1\}$ and define the set of boundary locations B to be the values of τ which satisfy $\text{sim}(\tau, \gamma) = 0$. The whole procedure for Stage 2 is summarised in Algorithm 1.

The number of boundary locations identified given by $|B|$ will depend on the choice of k in the k-means clustering and the value of γ selected. Generally, the smaller the value of γ, the more boundary locations will be identified. By searching over a range of values of γ and k this will give engineers a number of possible options of boundary locations for their consideration. The effect of these parameters on performance is explored in the next section.

Algorithm 1 Stage Two: Identifying Boundary Locations

Input: Data points = $\boldsymbol{x_1} \ldots \boldsymbol{x_N}$, micro-cluster centres, number of macro-clusters k, search parameter γ.

Output: A set of boundary locations B

Set $B = \emptyset$.

Apply k-means on the micro-cluster centres.

Assign each data point in $\{\boldsymbol{x_1} \ldots \boldsymbol{x_N}\}$ to a k-means macro-cluster.

for $\tau \in \{\gamma + 1 : N - \gamma + 1\}$ **do**

    Calculate $n_{L,j}$ and $n_{R,j}$ for all $j \in \{1, \ldots, k\}$.

    Calculate $\text{sim}(\tau, \gamma) = \frac{\sum_{j=1}^{k} \min(n_{L,j}, n_{R,j})}{\gamma}$.

    if $\text{sim}(\tau, \gamma) = 0$ **then**

        $B = B \cup \tau$

    end if

end for

1.3 Results on DAS data

The aim of this section is to use Clustream with Algorithm 1 to identify the location of corrupted data within acoustic sensing data. The data stream we consider consists of 10000 data points (one second of data) and ten different series relating to different depths within the oil well. In order to compare performance of our algorithm we compare against a ground truth. The ground truth is shown in Figure 1.3.1 and consists of six manually chosen boundary locations.

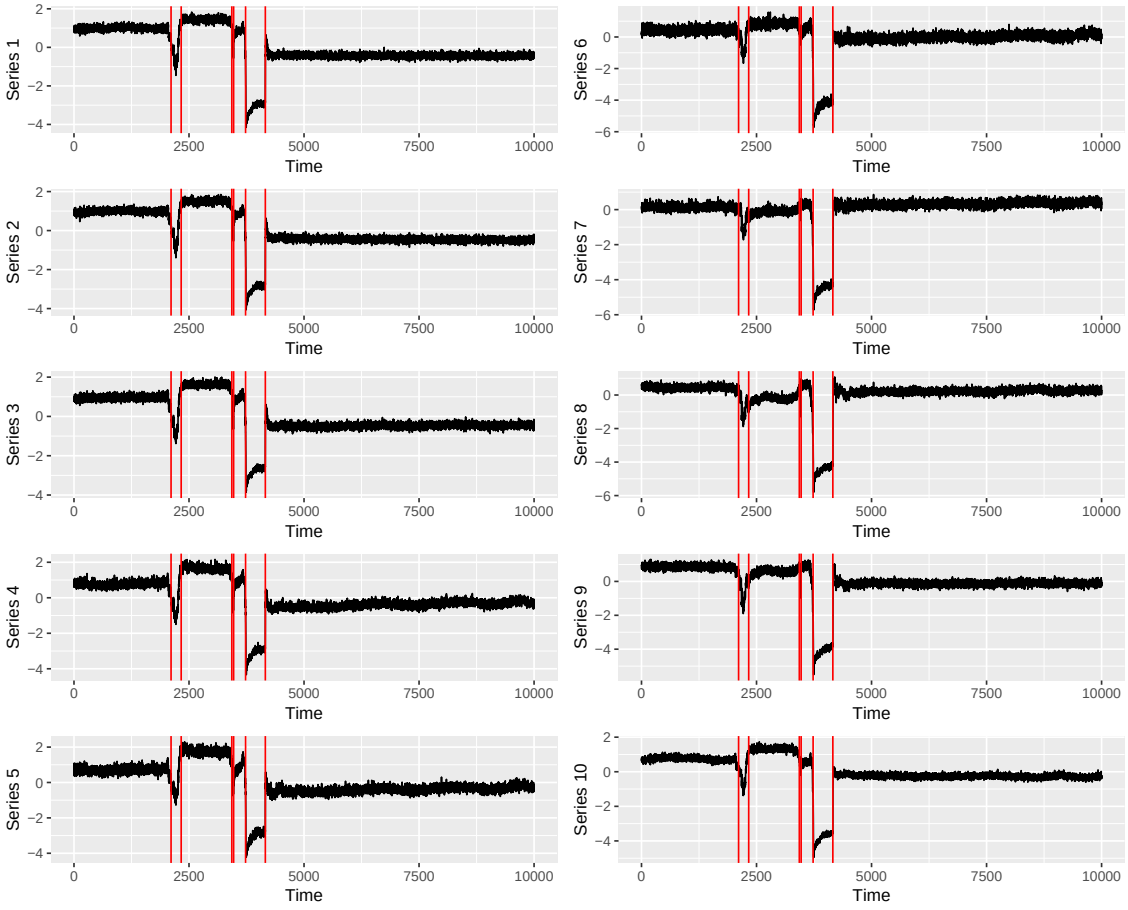


Figure 1.3.1: Ground truth for the DAS data shown for all ten series. The six true boundary locations are shown with red vertical lines.

We compare the boundary locations found by Algorithm 1 against the ground truth by using the V-measure. The V-measure (Rosenberg and Hirschberg, 2007) is a performance measure which rates the quality of a given segmentation compared to the ground truth segmentation. V-measure rates the segmentation by balancing both homogeneity and completeness. A larger V-Measure value indicates higher accuracy, with a value of 1 indicating a perfect segmentation. For full details, see in Section **??**.

Clustream was applied on the DAS data stream using 250 micro-clusters. At $t = 10000$ we applied Algorithm 1 on the signal for a range of values of k and γ to identify boundary locations. Due to the randomness induced by the k-means step, the results for each experimental setting are averaged out over 10 runs. The variation between runs was minimal. The average V-measure for each setting is presented in Table 1.3.1.

k	γ									
	5	10	15	20	25	30	35	40	45	50
2	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**	**0.742**
3	**0.967**	0.828	0.828	0.828	0.828	0.828	0.828	0.828	0.828	0.828
4	**0.956**	0.88	0.88	0.88	0.88	0.829	0.829	0.829	0.829	0.828
5	0.937	0.946	**0.961**	**0.961**	**0.961**	0.931	0.931	0.931	0.931	0.828
6	0.798	0.935	**0.961**	**0.961**	**0.961**	0.93	0.93	0.93	0.931	0.828
7	0.802	0.935	0.952	0.952	**0.961**	0.942	0.942	0.93	0.93	0.868
8	0.805	0.936	0.944	0.944	**0.961**	0.955	0.955	0.93	0.93	0.93

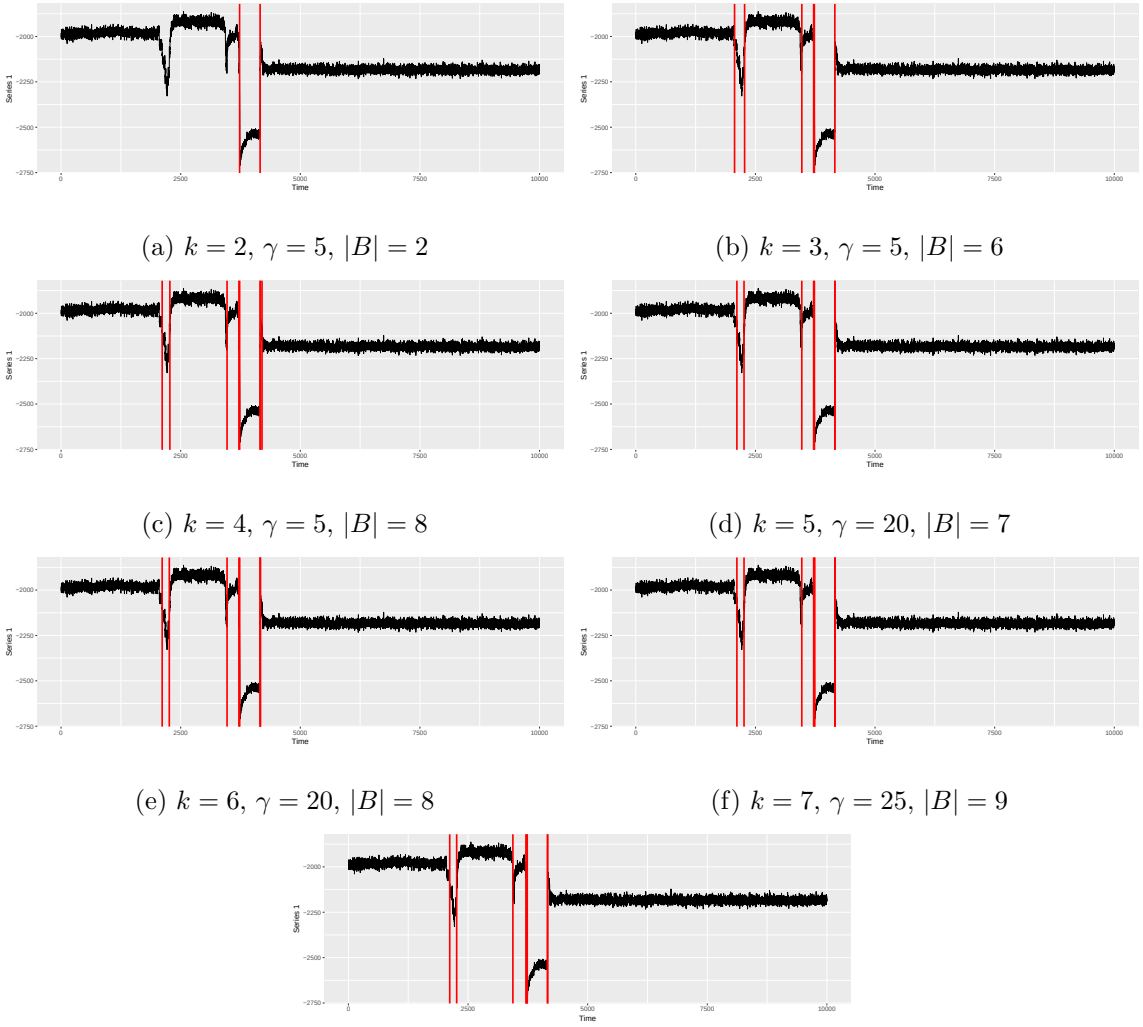
Table 1.3.1: V-measure results on the DAS data for a range of k and γ. The best performance for each value of k is highlighted in bold.

The best performance in terms of V-measure is given by $k = 3$, $\gamma = 5$ however, good performance is found across the settings. The choice of γ had no effect for the experiments where $k = 2$. Generally as k increases, the ideal choice of γ for that value of k also increases. This makes sense as when k increases, $\text{sim}(\tau, \gamma)$ is more likely to take values of 0, resulting in more boundary locations being identified. By choosing a larger γ for larger values of k, this prevents the number of boundary locations chosen by the algorithm from growing too large. The average number of boundary location identified by the algorithm under the different scenarios is given in Table 1.3.2.

k	γ									
	5	10	15	20	25	30	35	40	45	50
2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
3	**6**	3	3	3	3	3	3	3	3	3
4	**8**	5	5	5	5	4	4	4	4	3
5	11	8	**7**	**7**	**7**	6	6	6	6	5
6	19	11	**9**	**8**	**8**	7	7	7	6	5
7	18.8	11.6	9.6	9	**8.6**	8	8	7.6	7	6.4
8	18.4	11.6	9.6	9.4	**8.6**	8.4	8.4	7.6	7.4	7.4

Table 1.3.2: Average number of boundary locations identified on the DAS data for a range of k and γ. The number of boundary locations corresponding to the best performing V-measure for each value of k is highlighted in bold.

In Table 1.3.2 we can see that the the number of boundary locations found does vary with k and γ. However, the actual boundary locations identified are similar across the different parameter settings. In order to demonstrate this, Figure 1.3.2 plots the boundary locations identified using the most suitable value of γ for each value of k.



(a) $k = 2$, $\gamma = 5$, $|B| = 2$

(b) $k = 3$, $\gamma = 5$, $|B| = 6$

(c) $k = 4$, $\gamma = 5$, $|B| = 8$

(d) $k = 5$, $\gamma = 20$, $|B| = 7$

(e) $k = 6$, $\gamma = 20$, $|B| = 8$

(f) $k = 7$, $\gamma = 25$, $|B| = 9$

(g) $k = 8$, $\gamma = 25$, $|B| = 9$

Figure 1.3.2: Each plot shows the boundary locations (red lines) identified under different parameter settings. We show one plot for each value of k, and use the most appropriate value of γ in each setting. The number of boundary locations identified is given by $|B|$.

Only plots for Series 1 are shown in Figure 1.3.2 here but similar performance was observed across all ten series. We can see that in Figure 1.3.2a, for $k = 2$ only two boundary locations are found but these manage to pick out the most obvious data corruption. As k increases, the number of boundary locations increases, but boundary locations only occur in places of change within the signal. For example, in Figure 1.3.2g we can see that $|B| = 9$ although we can only visibly see five red lines in the plot. This means that some of the boundary locations will be very close together, perhaps even neighbouring time points.

1.4 Conclusion

In this chapter we have demonstrated that Clustream can provide an alternative to change-point detection methods for identifying corruption in digital acoustic sensing signals. The fact that Clustream can be run efficiently online means that it can cope well with the high frequency and high dimensionality data created by digital acoustic sensing.

In order to frame the time-series as a clustering problem, we treated each time-series as a different data dimension. Interestingly the initial k-means clustering assignments look visually sensible in a temporal sense, despite no temporal information being given to the k-means algorithm.

We have developed an Algorithm which uses cluster assignments to identify boundary locations within the DAS signal. This algorithm was tested on the DAS data set for a range of values of k and γ and the results were found to be robust. As it is not clear exactly what engineers wish to find when searching for corruption, offering a range of potential boundary locations by varying γ and k can be useful.

When detecting boundary locations, we considered only the time points τ where $\text{sim}(\tau, \gamma) = 0$. It would be possible to instead threshold $\text{sim}(\tau, \gamma)$ to increase the number of boundary locations identified by the algorithm. It would also be possible to consider $\text{sim}(\tau, \gamma)$ as a function of τ and treat identifying boundary locations as a local minima problem.

Bibliography

Charu C. Aggarwal, T. J. Watson, Resch Ctr, Jiawei Han, Jianyong Wang, and Philip S. Yu. A Framework for Clustering Evolving Data Streams. *Proceedings of the 29th international conference on Very large data bases*, pages 81–92, sep 2003.

K. Boone, A. Ridge, R. Crickmore, and D. Onen. Detecting Leaks in Abandoned Gas Wells with Fibre-Optic Distributed Acoustic Sensing. In *International Petroleum Technology Conference*. International Petroleum Technology Conference, jan 2014. ISBN 978-1-61399-322-4.

Jie Chen and Arjun K. Gupta. *Parametric Statistical Change Point Analysis*. Birkhäuser Boston, Boston, 2012. ISBN 978-0-8176-4800-8.

Idris A. Eckley, Paul Fearnhead, and Rebecca Killick. Analysis of changepoint models. In David Barber, A. Taylan Cemgil, and Silvia Chiappa, editors, *Bayesian Time Series Models*, chapter 10, pages 205–224. Cambridge University Press, Cambridge, 2011.

Es Page. Continuous inspection schemes. *Biometrika*, 41(1):100–115, jun 1954.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the 2007 Joint Conference on Empirical Methods*

in Natural Language Processing and Computational Natural Language Learning, 1(June): 410–420, 2007.

Tatiana Silkina. *Application of Distributed Acoustic Sensing to Flow Regime Classification*. Master's thesis, Norwegian University of Science and Technology, 2014.

Juun van der Horst, Hans Den Boer, Peter In 't Panhuis, Brendan Wyker, Roel Kusters, Daria Mustafina, Lex Groen, Nabil Bulushi, Rifaat Mjeni, Kamran Fahmeed Awan, Salma Mohammed Rajhi, Mathieu M Molenaar, Alan Reynolds, Rakesh Paleja, David Randell, Richard Bartlett, and Kevyn Green. Fibre Optic Sensing For Improved Wellbore Production Surveillance. In *International Petroleum Technology Conference*. International Petroleum Technology Conference, jan 2014. ISBN 978-1-61399-322-4.