Aalto University
School of Science
Bachelor's Programme in Science and Technology

Utilisation of viewing statistics in video recording credits detection

Bachelor's thesis

April 19, 2022

Saskia Kivistö

Aalto University
School of Science
Bachelor's Programme in Science and Technology

ABSTRACT OF
BACHELOR'S THESIS

Author:	Saskia Kivistö
Title of thesis:	Utilisation of viewing statistics in video recording credits detection
Date:	April 19, 2022
Pages:	27
Major: **Code:**	Computer Science SCI3027
Supervisor: **Advisor:**	Assistant Professor Parinya Chalermsook Wanchote Jiamjitrak, M.Sc. (Tech) (Department of Computer Science)

The goal of this thesis was to study whether it is possible to use user viewing behaviour statistics to distinguish program content from surplus content from television broadcasts, which have been recorded with a network personal video recorder. Recorded television programs tend to include surplus content, since programs are not broadcast strictly according to the electronic program guide. In order to ensure that the entire program is recorded, recording must start a few minutes earlier and end a few minutes later than what is stated as the programs broadcast time in the electronic program guide. The incentive for identifying surplus content is that it consumes storage space, and fast forwarding over it is tedious for the person watching the recording.

The research question was treated as a signal processing problem for offline change point detection, more specifically as a minimisation problem for the sum of costs of segments. The optimal segmentation was computed with dynamic programming, using least squares deviation as the cost function.

The position of opening and closing credits was correctly identified for most of the recordings in the sample of 279 recordings. The last change point was typically within a few seconds from the beginning of closing credits, but for recordings with program content during the closing credits, the last change point was predicted closer to the end of closing credits. There was more variance in the position of the first change point within opening credits.

The precise start and end time of program content could not be detected solely with user viewing behaviour statistics, but the data could be useful for more precise detection combined with other analytics and information about the typical program structure.

Keywords:	change point detection, segmentation, statistical signal processing, NPVR, behavioural analytics
Language:	English

Aalto-yliopisto
Perustieteiden korkeakoulu
Teknistieteellinen kandidaattiohjelma

KANDIDAATINTYÖN
TIIVISTELMÄ

Tekijä:	Saskia Kivistö
Työn nimi:	Videonauhoitteiden alku- ja lopputekstien paikannus katsomistilastojen perusteella
Päiväys:	19. huhtikuuta 2022
Sivumäärä:	27
Pääaine: **Koodi:**	Tietotekniikka SCI3027
Vastuuopettaja: **Työn ohjaaja:**	Apulaisprofessori Parinya Chalermsook DI Wanchote Jiamjitrak (Tietotekniikan laitos)

Tässä kandidaatintyössä tutkittiin, onko katsomiskäyttäytymistilastojen avulla mahdollista erottaa ohjelmasisältö ylimääräisestä sisällöstä televisiolähetyksissä, jotka on nauhoitettu verkkopohjaisella digitaalisella mediatallentimella (engl. network personal video recorder). Nauhoitteiden alkuun ja loppuun jää ylimääräistä sisältöä, koska nauhoituksen tallennusaika on muutaman minuutin pidempi kuin ohjelman ohjelmaoppaan mukainen kesto. Tällä varmistetaan, että ohjelma tallentuu kokonaisuudessaan, vaikka sen lähetysaika poikkeaisikin hieman ohjelmaoppaasta. Kannuste ylimääräisen sisällön tunnistamiselle on, että se vie tallennustilaa ja sen ohi kelaaminen on puuduttavaa.

Tutkimuskysymystä lähestyttiin signaalinkäsittelyn kautta signaalin segmentointiongelmana. Mahdollisten segmenttien homogeenisyyttä mitattiin neliöllisellä tappiofunktiolla. Tappiofunktion minimoivan segmenttiyhdistelmän laskemiseen sovellettiin kahta algoritmia, joista ensimmäinen löytää optimaalisen ratkaisun, kun segmenttien määrä on määritelty ennalta, ja toinen, kun segmenttien määrä ei ole tiedossa.

Alku- ja lopputekstien sijainti löytyi oikein suurimmassa osassa 279 tallenteen otoksesta. Lopun muutoskohta paikantui tyypillisesti muutaman sekunnin päähän lopputekstien alkukohdasta, mutta jos lopputekstien aikana tai päätteeksi oli vielä varsinaista ohjelmasisältöä, muutoskohta paikantui lähemmäs lopputekstien loppua. Alun muutoskohdan sijainnissa alkutekstien sisällä oli enemmän vaihtelua.

Alku- ja lopputekstien tarkkaa alku- ja päätöskohtaa ei pysty määrittämään pelkkien katsomistilastojen avulla. Katsomistilastot voisivat kuitenkin olla hyödyllisiä tarkan alun ja lopun paikantamisessa yhdistettynä muuhun analytiikkaan ja metatietohin ohjelmien tyypillisestä rakenteesta.

Avainsanat:	muutospisteiden havaitseminen, segmentaatio, tilastollinen signaalinkäsittely, NPVR, käyttäytymisanalytiikka
Kieli:	englanti

Contents

1 Introduction

Watching television used to be a very time-sensitive activity. If one wanted to see a TV program, one had to watch it when it was broadcast. Videocassette recorder brought the freedom to store programs and to choose freely when to watch them. Even though newer technologies have replaced the videocassette recorder, one major inconvenience still remains. Programs are not broadcast strictly according to the schedule of the Electronic program guide (EPG). Sometimes programs start earlier or end later than what is stated in the EPG. To ensure that the entire program is recorded, recording must start before the EPG start time and end after the EPG end time. This usually results in some non-program content being included in the recording. Skipping over the non-program content can be frustrating for the person watching the recording.

Knowing when the program truly starts and ends would solve the issue, but this information is not generally available. This leaves the option to detect the start and end of the program from the recording. Various artificial intelligence solutions have been devised to solve the problem, for example in [3] [4] [6] [10], but as program content can be quite varied it is difficult to find a universal solution.

Network personal video recorder (NPVR) is a type of service for recording broadcast TV programs for later viewing. Instead of storing recordings on the users' local device, NPVR stores recordings on the content provider's server. The start and end times of recordings are determined by the scheduling information given in the EPG, and a fixed amount of margin is added on both ends to ensure that the entire program is recorded. This leads to non-program content being included in recordings, which the users typically skip. However, statistics of which parts of the recording users watch and which parts they skip can be collected and analysed.

The goal of this thesis was to study whether user viewing behaviour can be used to detect when the actual program content begins and ends in an NPVR recording. This thesis was written on behalf of an NPVR service provider company. From the perspective of an NPVR service provider, detecting the location of core program content is useful for the following reasons. Firstly, less storage space is needed if the surplus content is discarded. Secondly, it is convenient for the customers if the relevant content of a recording is pre-identified, and they do not have to search for it.

This thesis is largely based on the ”Selective review of offline change point detection methods” by Truong et al. [14]. The program content detection task was formulated as an offline signal change point detection problem, and the suitable algorithms for solving the problem were chosen with the help of the typology established in the research paper. The empirical calculations were done with the Python scientific library `ruptures`, which is associated with the aforementioned research paper.

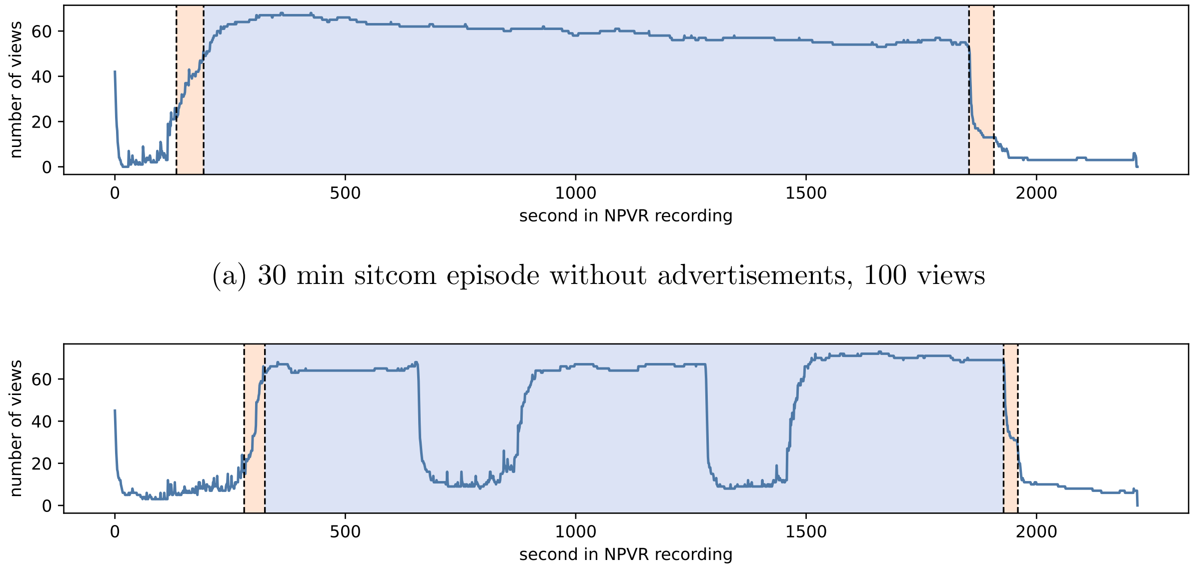
The thesis is structured as follows. Section 2 gives an overview on the characteristics of the viewing behaviour data. Section 3 discusses the theoretical background of signal change point detection from the perspective of this specific use case. Section 4 discusses the sample data and the `ruptures` library, and how the library was used in this thesis to detect change points. The results are addressed in Section 5. Section 6 considers the viability of using user viewing behaviour for change point detection, based on the previous sections. Lastly, the main points of this thesis are summarised in Section 7.

2 User viewing behaviour data

Whenever an NPVR video is watched by a user, certain metrics about the viewing event are saved. The main reason for collecting viewing metrics is the monitoring of the amount of views and the user experience quality. It can be calculated from the metrics of one view which parts of the video the user watched and which parts they skipped. Aggregating this data from multiple views of the same recording allows acquiring an overview of what parts of the recording users typically watch. This type of recording specific aggregated view count is referred to as user behaviour data in this thesis.

The content of an NPVR recording can be divided into program and non-program content. Non-program content is the surplus material at the very beginning and end of a recording, that is a byproduct of ensuring that the entire program is recorded even if its broadcast time deviates slightly from the EPG schedule. Program content can be categorised into opening and closing credits, core program content and advertisement breaks. Advertisement breaks occur only in programs that are broadcast on commercial channels.

User interest for the content categories varies. Core program content is what the average user watches the recordings for, and non-program content is irrelevant for them. This is reflected in the viewing behaviour. Users also tend to skip over advertisement breaks. Viewing behaviour regarding the opening and closing credits is not as clear-cut, but generally users are content with starting watching anywhere on the opening credits. Closing credits do not typically interest users, given there is no extra content included in them, such as a preview of the next episode.



(a) 30 min sitcom episode without advertisements, 100 views

(b) 30 min soap opera episode with two advertisement breaks, 100 views

Figure 1: Visualisation of user viewing behaviour for two example recordings.

Two examples of user viewing behaviour data are illustrated in Figure 1. The data in both figures consists of a sample of 100 user views, but the views are from two different TV programs. Figure 1a views are from a sitcom episode and Figure 1b views are from an episode of a soap opera. The episodes are divided into one second-long segments, and calculated for each segment is the number of user views in which the segment was watched. The segments are plotted on the horizontal axis, and the number of user views per segment is plotted on the vertical axis. For example, 63 users from the sample of 100 users watched the part of the sitcom episode between 0:10:00 - 0:10:01 (600 on the horizontal axis in Figure 1a).

The background colour in the figure represents the content type of the time segment: non-program content is white, core program content is blue and beige indicates the opening and closing credits. The change points are marked with a vertical dashed line. There is a distinct increase in the number of views during the opening credits in both figures. Likewise, there is a steep decrease in the number of views during closing credits. The program visualised in Figure 1b has two advertisement breaks, which appear as recesses in the number of views during the core program content. Figure 1a episode has no advertisement breaks.

2.1 Data cleaning

This section explains how the sample views were chosen. A uniform sample size was used for all recordings in order to simplify result evaluation and comparison. The characteristic pattern for viewing behaviour, described at the end of the previous section, is usually visible from a dozen views. Sample size of 100 views was chosen because it should be large enough to ensure that it is very unlikely that the characteristic pattern does not emerge.

As the first step, all views of a recording are sorted from oldest to newest. The views will be added to the sample in this order, but potentially incorrect or irrelevant views are filtered out. This means that the sample views are not simply the 100 earliest ones. Firstly, views which end before the recording process ends are discarded. Also views with no skips are discarded, since the change point detection relies on inspecting which parts the viewers skip, so views without skips are useless. Lastly, views with implausible player source duration are removed. This includes the cases where player source duration does not exist, it is negative or close to maximum value for unsigned 32-bit integer.

3 Theoretical background

3.1 Signal change detection

Locating the start and end of programs with user viewing behaviour time series data can be formulated as a signal processing problem, more precisely as a change point detection problem. Signal change point detection is quite widely researched topic, since it has applications in multiple fields, such as network traffic data analysis [7] [9], bio-informatics [8] [17] and climatology [13] [16]. However, signal processing is not the only possible approach for time series change point detection. Statistical approaches, such as Bayesian methods, are also widely used [1] [12], but in order to narrow down the scope of this thesis only signal processing methods will be examined.

The signal in this case is the user viewing behaviour time series data, as described in Section 2. The assumption in signal change point detection is that the signal consists of segments with different statistical properties, for example a different central tendency. The data in each segment is approximately homogeneous in respect to the statistical property that varies between segments. Change points are the transition points between segments.

Signal change point detection problems can be divided into two main categories, depending on whether the change detection must be done for incoming data in near real-time, or not. Methods solving the former case are referred to as online algorithms. Offline algorithms solve the latter case, and they differ from the online algorithms by getting the entire dataset as input and typically being computationally more complex, but also by detecting the changes more accurately. The user viewing behaviour data is aggregated from multiple views after the views have ended, meaning that all the data to be analysed is available from the start, allowing offline methods to be used.

Offline change point detection methods can be divided further into two categories, based on whether the number of change points in the signal, K, is known beforehand, or not. If K is unknown, a constraint which restricts the number of change points must be added to prevent overfitting. This is referred to as the *complexity penalty.* There are multiple approaches for choosing the complexity penalty.

Literature review by Truong et al. [14] compared a selection of offline change point detection algorithms. It classified the algorithms according to whether the number of change points is known in advance, how the homogeneity within signal segments is measured, and how the signal is divided into segments. The measure of homogeneity is referred to as *cost function* and the method for choosing segments is referred to as *search method.* A more detailed definition of the cost function is in Section 3.2 and of the search function in Section 3.3. The objective of the change point detection algorithms discussed in the literature review is to find a segmentation that minimises the sum of cost functions for the segments. The function to be minimised can be expressed as follows:

$$\sum_{k=0}^{K} \text{cost}(y[t_k + 1, t_{k+1}]) \tag{1}$$

where $y[t_k + 1, t_{k+1}]$ is a segment of signal y from index $t_k + 1$ to t_{k+1} and t_{k+n} is the index of the nth change point, cost is the cost function, and K is the number of change points.

3.2 Cost function

What is a good cost function depends on what kind of change occurs in the signal on change points. Looking at the user viewing behaviour data discussed in Section 2, it seems that what best describes what happens to the viewer count during credits is a shift in the distribution mean. A cost function that works well for mean shift detection is squared deviation, cost_{L_2}. It measures the variance and is defined as follows:

$$\text{cost}_{L_2}(y[a, b]) = \sum_{t=a}^{b} \|y[t] - \overline{y}[a, b]\|_2^2 \,, \tag{2}$$

where $y[a, b]$ is a segment of signal y from index a to b, and $\overline{y}[a, b]$ is the empirical mean of $y[a, b]$.

Another possibly suitable cost function for the data is the absolute deviation, cost_{L_1}. It detects changes in the median and can be used also for detecting shifts in other central tendencies, such as mean and mode. It is defined as follows:

$$\text{cost}_{L_1}(y[a, b]) = \sum_{t=a+1}^{b} \|y[t] - \overline{y}[a, b]\|_1 \,. \tag{3}$$

The key difference between squared deviation and absolute deviation is that the squared deviation penalises larger deviations from the mean more. As there might be some fluctuation in the viewer count during the core program content, and drastic changes in the viewer count typically occur during content type changes, it can be assumed that for this use case Equation 2 should produce better results than Equation 3.

3.3 Search method

Truong et al. [14] divided search methods into two categories, by whether they always provide an optimal solution for change point location with the chosen cost function, or if they provide an approximate solution. This thesis will examine only optimal methods. Optimal methods produce inherently more reliable and accurate results than approximate ones, with the trade-off of typically higher computational complexity. The approximate methods are excluded on the basis that the goal of this thesis is to study whether it is at all possible to detect credits from solely viewing behaviour statistics, not determining the most viable approach to do it in practice. Therefore, high reliability and accuracy outweighs low computational complexity.

Two optimal search methods were presented in the literature review, `Opt` and `Pelt`. `Opt` was first introduced by Bellman [2] for an unrelated problem. `Pelt`, short for Pruned Exact Linear Time, was first introduced by Killick et al. [5]. The main difference between the methods is that `Opt` requires the number of change points K to be known beforehand, and `Pelt` can be used when K is unknown.

Both of the methods use dynamic programming to find the change points. Dynamic programming is a mathematical optimisation method, where the problem is solved by recursively dividing it into smaller sub-problems. In this case the problem is minimising Equation 1. The optimal segmentation is defined as follows:

$$\mathrm{OPT}([a,b],K) : \text{Optimal segmentation for } y[a,b] \text{ with } K \text{ change points} \tag{4}$$

When K is known in advance the optimal segmentation can be calculated as follows:

The recursive step:

$$\mathrm{OPT}([a,b],K) = \min_{a<k<b} \begin{cases} \mathrm{cost}(y[a,k]) + \\ \mathrm{OPT}([k+1,b],K-1) \end{cases} \tag{5}$$

The base case:

$$\text{For all } 1 \leq a < b \leq m : \mathrm{OPT}([a,b],0) \leftarrow \mathrm{cost}(y[a,b]), \tag{6}$$

where m is the number of elements in y.

Intuitively, the above equations can be understood as follows. The base case in Equation 6 occurs when there are no change points and thus no subsegments, that is $K = 0$. For a signal with no change points, Equation 1 is simply the output of the cost function for that signal. Finding the optimal segmentation for a signal with $K > 0$ change points means finding a set of segments that minimises the sum of costs. As described in Equation 5, the minimisation task can be expressed as finding the optimal combination of the first segment and optimal segmentation for the remaining signal.

The concept of the algorithm is represented as top-down, but in practice the implementation is done bottom-up, by first computing the possible base cases and building more complex optimal segmentations on top of the simpler ones with less change points.

In principle, finding the optimal set of change points works similarly when K is unknown. It would be possible to choose a sufficiently large maximum value $K = K_{\max}$ and iterate `Opt` for all $K = 1, ..., K_{\max}$ and choose the result which produced the best minimisation of Equation 1. However, the computational complexity of this method is very high. A more viable solution is to add a complexity penalty term into Equation 1 to restrict the number of predicted change points. With the complexity penalty pen Equation 1 becomes:

$$\sum_{k=0}^{K} [\text{cost}(y[t_k + 1, t_{k+1}]) + \text{pen}] \tag{7}$$

When pen is linear the `Pelt` algorithm can be used. In principle, `Pelt` follows the dynamic programming approach of Equation 5 and 6, but there is a pruning rule which discards some indices which cannot be change points, reducing the computational complexity. [5]

4 Methods

4.1 Sample data and ground truth

In order to evaluate how well a method detects change points, a ground truth is needed for comparison. Ground truth can be obtained by having a person look at a video recording and having them write down the timestamps of the change points.

The start and end time of both opening and closing credits were collected by hand from 279 NPVR recordings with a margin of error of ± 1 seconds. The sample consists of episodes from 16 different TV series. 138 of the sample recordings were broadcast on non-commercial channels, and therefore they do not contain advertisement breaks. Some general information about these recordings is listed in Table 1. The 141 recordings with advertisement breaks are listed in Table 2. Every recording in the sample has at least one hundred user views, after the data cleaning process described in Section 2.1.

The reason for selecting exclusively serial programs in the sample is that series tend to have a predictable program structure that stays consistent between episodes. This makes it faster to collect the ground truth, as the program structure has to be deciphered only once per series. For the rest of the episodes, the person collecting the timestamps knows what to look for.

There were a couple of factors that contributed to a series being included in the sample. These were the perceived likelihood of the series being popular enough to have a least one hundred suitable user views per episode, a large amount of episodes, and a conventional program structure in which credits are clearly distinct from the program content. Nevertheless, some series were included in the sample mostly by chance.

The sample series were categorised by program structure. The categories are defined in Table 3. Conventional program structure refers to both of the credits being clearly distinct from the core program content, and in between the non-program and core program content. For example, the program in Figure 1a has conventional structure. The other categories refer to some anomaly in the program structure, compared to the conventional structure.

Recapitulations of previous episodes and sneak peeks into following episodes were considered as a part of the credits in the ground truth. For programs with a delayed or entwined intro, the ground truth for opening credits was set to consist of the first second of the program, instead of the actual opening credits. The reasoning behind this was the assumption, that an average viewer does not want to miss the first minutes of the core program content before or during the opening credits. For programs with split intro, the ground truth opening credits contains only the first part of the opening credits.

series	episodes	episode length	genre	structure
A	6	30 min	reality television	conventional
B	10	30 min	comedy	conventional
C	32	30 min	comedy	conventional
D	30	45 min	drama	conventional
E	8	50 min	drama	conventional
F	20	50 min	drama	conventional
G	4	45 min	drama	conventional *
H	3	90 min	drama	entwined intro
I	9	40 min	drama	recap, split intro
J	16	45 min	drama	split intro

Table 1: Sample recordings without advertisement breaks categorized by series.

series	episodes	ad breaks	episode length	genre	structure
K	35	1	30 min	soap opera	conventional
L	25	1	30 min	soap opera	delayed intro
M	15	1, 2 or 3	30-60 min	reality television	recap, sneak peek
N	9	4	90 min	reality television	recap, sneak peek
O	31	2	30 min	soap opera	sneak peek
P	26	3	50 min	drama	split intro

Table 2: Sample recordings with advertisement breaks categorized by series.

category	description
conventional	Distinct opening and closing credits.
delayed intro	Opening credits occur a few minutes after the program has begun.
entwined intro	Opening credits are entwined with the core program content.
recap	Program begins with a recapitulation of the previous episode.
sneak peek	Closing credits end with a sneak peek into the following episode.
split intro	Opening credits are divided in two parts. The first part is at the beginning of the program, and the second occurs a few minutes after the program has begun.

Table 3: Descriptions of the program structure categories.

* One episode has delayed intro, rest of the episodes have conventional structure.

4.2 Change point detection with the `ruptures` library

Change points were detected from the sample data with the Python library `ruptures`. The library is based on the findings of the Truong et al. literature review [14]. Selected algorithms examined in the literature review are implemented in `ruptures`.

Suitable algorithms for this use case were chosen by considering the three aspects of change point detection methods discussed in the literature review: cost function, search method and complexity penalty. In this case accuracy is more important than low computational complexity, so optimal method is preferable to an approximate one. Two optimal search methods, `Opt` and `Pelt`, are implemented in `ruptures`, as described in Section 3.3.

As mentioned in Section 2, advertisement breaks in the recording may produce recesses in the viewer count. Change point detection algorithms can interpret these recesses as separate segments. The number of advertisement breaks per program varies, meaning that K is unknown if the recording might contain advertisement breaks. However, programs in non-commercial channels never contain advertisements, and for them, it was assumed that there should be exactly one change point at the beginning and one at the end of the program. This implies that `Opt` should be a viable search method for recordings from non-commercial channels with $K = 2$. `Pelt` can be used for all the recordings, but the complexity penalty needs to be calibrated. The penalty was calibrated heuristically, and the calibration process that was used is described in Section 5.2.

The `ruptures` library has three different cost functions that can be used with `Opt` and `Pelt`. These are absolute deviation cost_{L1} as described in Equation 3, squared deviation cost_{L2} as described in Equation 2, and kernel cost function cost_{rbf}. cost_{rbf} was not taken into further consideration, because it is substantially more complex than the other two cost functions. cost_{L2} is preferable over cost_{L1}, for the reasons discussed in section 3.2.

Standard statistical methods for univariate analysis were used for result evaluation. The statistics include five-number summary first devised by Tukey et al. [15], which includes the median, the 1st and 3rd quartiles and the minimum and maximum.

Following the convention used by Truong et al. the actual change points are denoted by t and the change points produced by `ruptures` are denoted by $\hat{t}$ [14]. Each change point has a subscript, where the first letter denotes the credits the change point corresponds to (o: opening credits, c: closing credits). If there are separate change points for the start and end timestamps of the credits, the subscript will have a second letter that denotes which one of those the change point represents (s: start, e: end). This notation is illustrated in Figure 2.

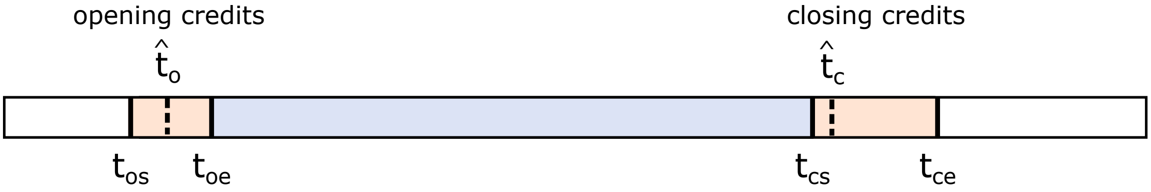


Figure 2: Notation for the actual (t) and predicted ($\hat{t}$) change points.

5 Results

5.1 Non-commercial channels with `Opt` as search method

An example of `Opt` output with cost function cost_{L2} and number of change points $K = 2$ is visualised in Figure 3. The viewing behaviour data is the same as in Figure 1a. The difference between the figures is that in Figure 3 the vertical dashed lines mark the change points determined by `ruptures` instead of the actual change points checked by hand. In the example, both of the predicted change points align with the credits. The change point for the start of the program is in the middle of the opening credits and the change point for the end of the program is close to the beginning of the closing credits.

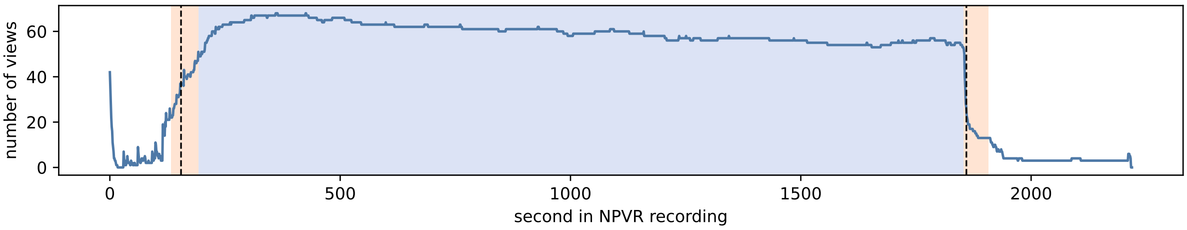


Figure 3: `Opt` cost_{L2} $K = 2$ output for Figure 1a data.

Running `Opt` cost_{L2} with $K = 2$ for Table 1 recordings, it turns out that in most cases the results are quite similar to Figure 3, and the predicted change points typically land somewhere on the credits. Out of the 138 recordings, the first change point landed outside of opening credits in 25 recordings and the second change point was not on the closing credits in one recording. More statistics of the result are listed in Table 4.

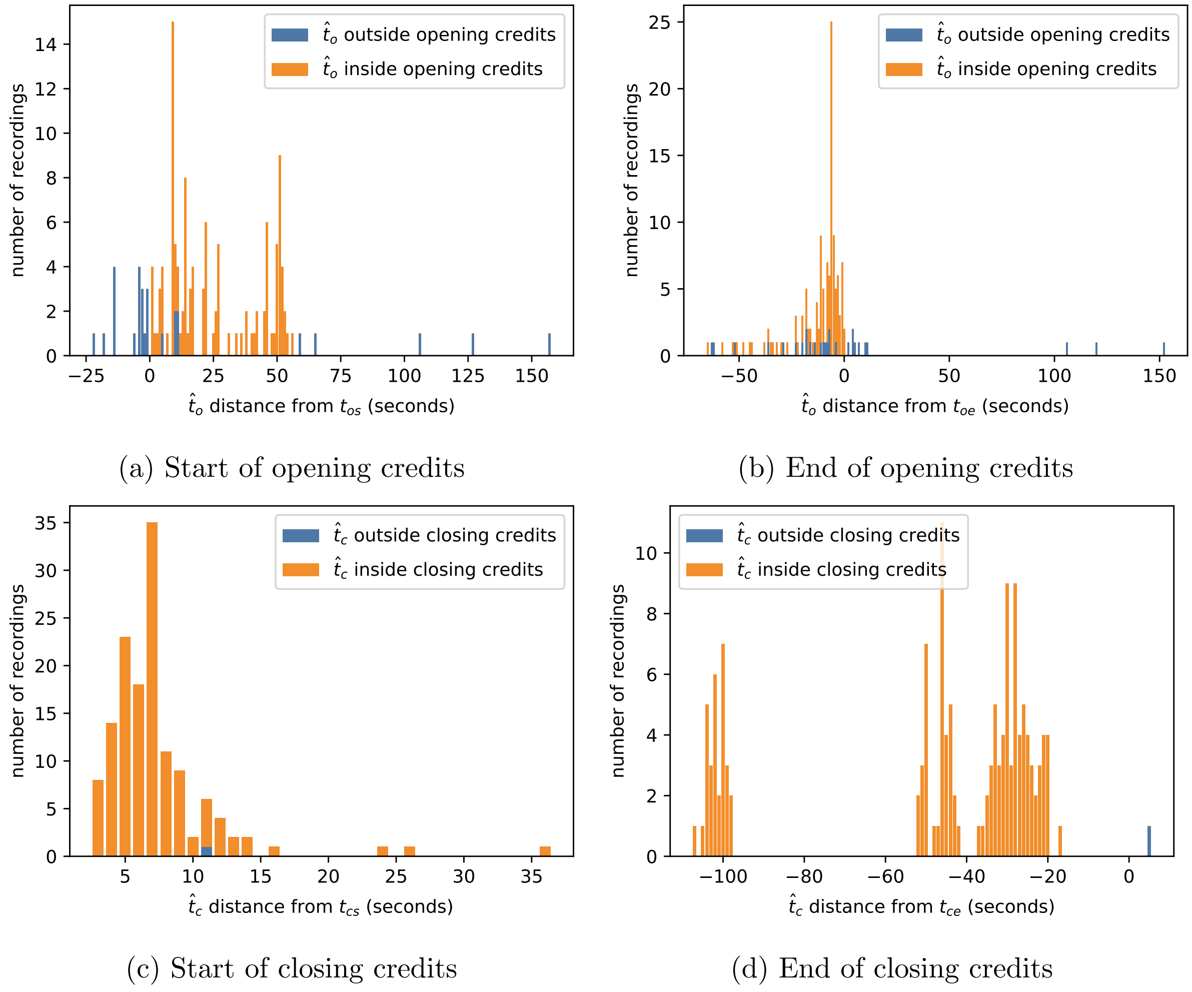
statistic	$\hat{t}_o - t_{os}$	$\hat{t}_o - t_{oe}$	$\hat{t}_c - t_{cs}$	$\hat{t}_o - t_{os}$
minimum	-22	-65	3	-107
1st quartile	9	-17.75	5	-50.75
median	16	-7	7	-35.5
3rd quartile	45,75	-5	8	-28
maximum	157	152	36	5
variance	676	635	17.2	873
standard deviation	26.0	25.2	4.15	29.5

Table 4: Five-number summary and other statistics for `Opt` cost_{L2} $K = 2$.

The assumption stated in Section 4.2, that the lack of advertisement breaks would result in $K = 2$ change points being aligned to the start and end of the program, did not hold for all the recordings. For two episodes of series J and one episode of series H, from Table

1, the first change point is off by a couple of minutes. In series J the change point is on the delayed opening credits and in series H during the opening credits entwined with the core program content. This explains the large maximum value for $\hat{t}_o - t_{oe}$ in Table 4. Without these three outliers the maximum value for $\hat{t}_o - t_{oe}$ would be 11.

The five-number summary of the quartiles in Table 4 gives some insight about the output, but the results can be understood more intuitively by plotting the data. Plotted in Figure 4 are the predicted locations for opening and closing credits, $\hat{t}_o$ and $\hat{t}_c$, compared to the actual start and end times of the credits. Figure 4a has $\hat{t}_o - t_{os}$ values plotted, Figure 4b has $\hat{t}_o - t_{oc}$ values plotted, et cetera.



(a) Start of opening credits

(b) End of opening credits

(c) Start of closing credits

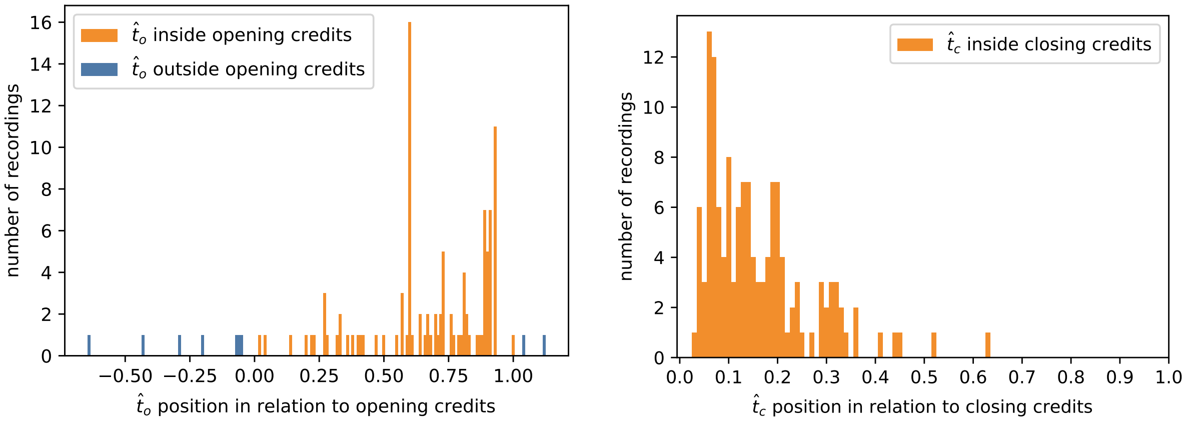
(d) End of closing credits

Figure 4: `Opt` cost$_{L2}$ $K = 2$ output compared to actual change points of credits.

In order to gain more insight about the results, the change points were divided into two categories, depending on whether the predicted change point $\hat{t}$ is within the credits it corresponds to, or not. Both $\hat{t}_o$ and $\hat{t}_c$ are typically somewhere on the credits, but $\hat{t}_o$ has more predictions outside the credits and higher variance in the relative position within the credits. For the closing credits, it is visible in Figure 4c and 4d that most often $\hat{t}_c$ is very close to t_{cs}, although with a delay of a few seconds. Three fourths of the sample $\hat{t}_c$

results are within 8 s from t_{cs}. Another detail worth noting is that it holds for all samples that $\hat{t}_c > t_{cs}$, meaning that none of the $\hat{t}_c$ were predicted to be before the closing credits.

The relative position of the predicted change points within the credits is visualised in Figure 5. The position 0.0 on the x-axis marks the beginning of credits and position 1.0 marks the end of credits. From Figure 5a it can be seen that $\hat{t}_o$ is more often closer to $\hat{t}_{oe}$ than $\hat{t}_{os}$, but it aligns strongly with neither. However, $\hat{t}_c$ has a clear tendency to align with $\hat{t}_{cs}$ with a slight time delay, as seen in Figure 5b. Credits that are ten seconds long or shorter were left out of the visualisation, because for very short credits misprediction by a few seconds is a big error compared to the length of the credits, which easily disturbs the clarity of the visualisation.



(a) Opening credits

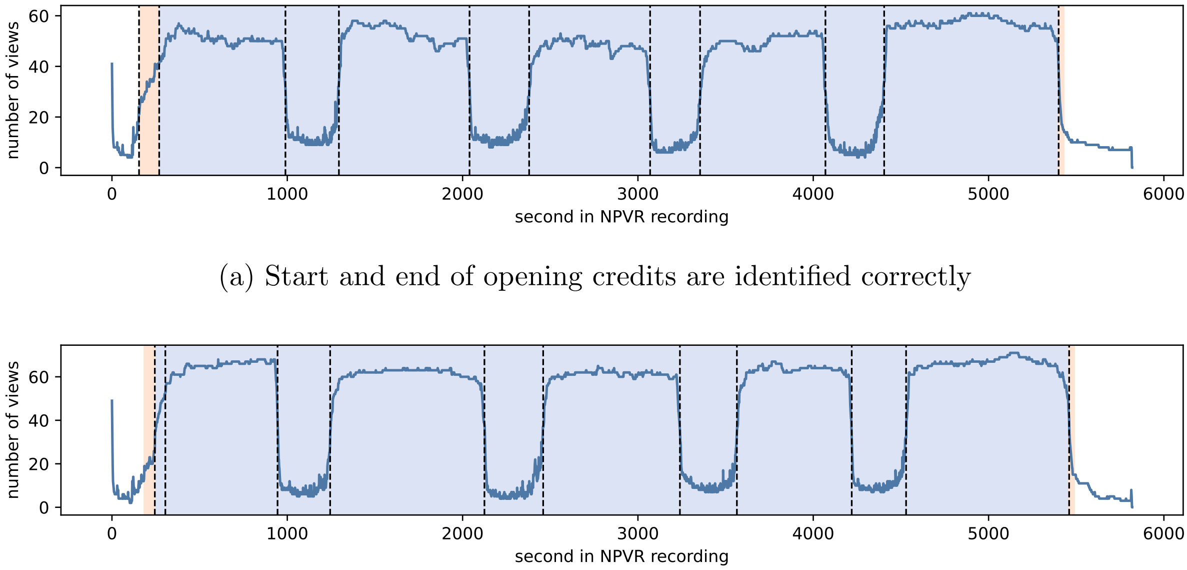
(b) Closing credits

Figure 5: `Opt` cost$_{L2}$ $K = 2$ output change point position in relation to credits.

5.2 All channels with `Pelt` as search method

As the number of change points detected by `Pelt` is controlled by a complexity penalty, the exact amount of detected change points is not known in advance. A common outcome is that one change point is predicted per credits, possibly with additional change points within the program. Occasionally the output is more precise, and both the start and end of credits are found. An example of this is visualised in Figure 6a. However, as can be seen in Figure 6b, sometimes results that seem possibly very accurate can be misleading. Since there is no trivial way to evaluate whether `Pelt` output is more accurate than usually, without knowing the ground truth, only the first and last predicted change points of recordings were examined, and it was assumed that the former corresponded to opening credits, and the latter to closing credits.

The value for the cost$_{L_2}$ complexity penalty was determined heuristically, by trying out different values and evaluating the result with the following criterion: a complexity



(a) Start and end of opening credits are identified correctly

(b) End of opening credits is correctly identified, but with an extra change point after it

Figure 6: `Pelt` cost$_{L2}$, pen = 18000, output for two episodes of series N from Table 2.

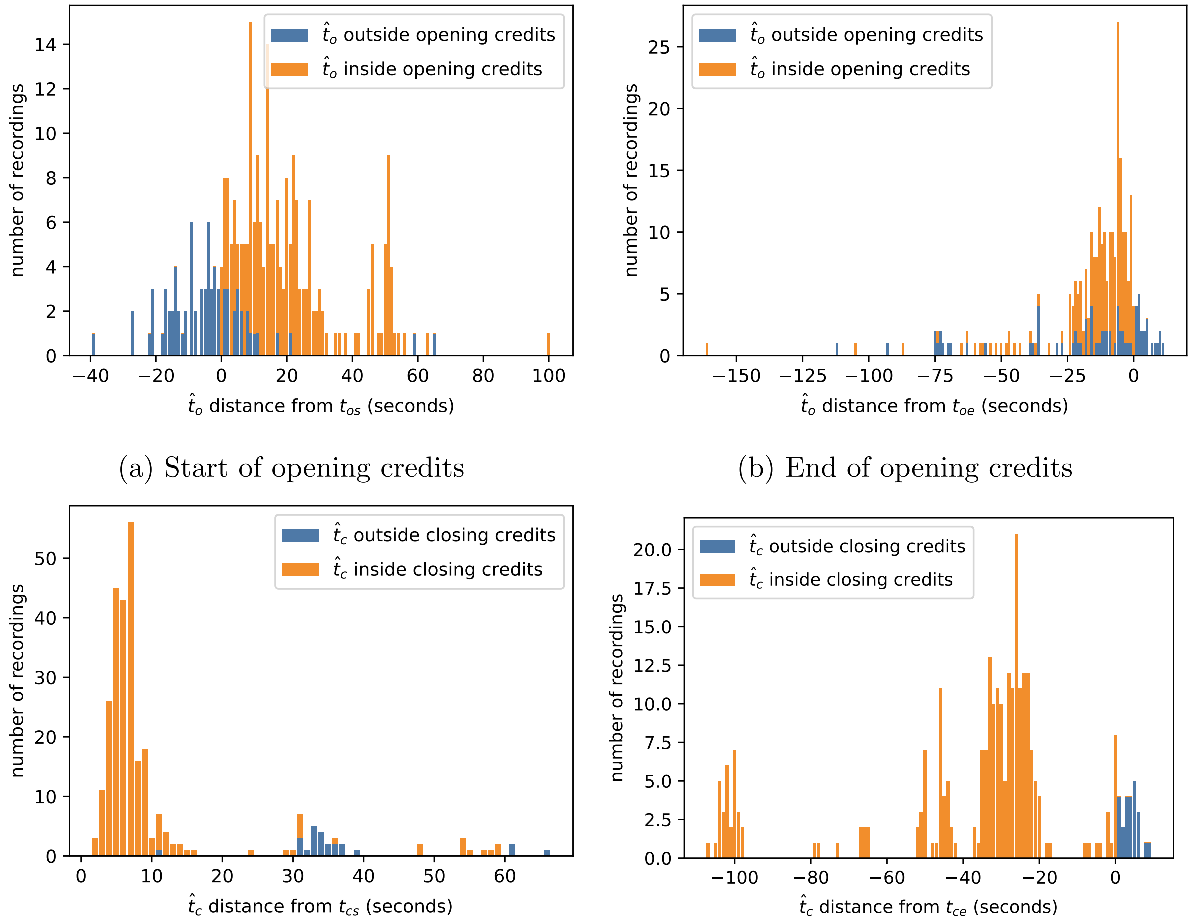
penalty was considered sufficiently good, when for all the 279 sample recordings, both the first and last predicted change point were within 60 seconds of the opening or closing credits, respectively. Penalties between 10000 and 25000 produced good results, when the penalty was increased in increments of one hundred. It can be assumed that, for the sample data, penalties smaller than 10000 are prone to detect too many change points, and likewise penalties larger than 25000 are likely to fail to detect all relevant changes. Penalty of 18000 was chosen as the final value, because it is almost in the middle of 10000 and 25000, and thus it is likely to minimise the risk for both under and oversegmentation.

Statistics of the results with `Pelt` cost$_{L_2}$, pen=180000 are listed in Table 5. The data encompasses all the 279 recordings in Table 1 and 2. The absolute distance from the beginning and end of credits is plotted in Figure 7 similarly as the `Opt` results in Figure 4. The relative distance within credits is plotted in Figure 8. Similar to Figure 5, in order to keep the visualisation informative, credits that are ten seconds long or shorter were excluded from Figure 8.

The results appear to be quite similar to the results with `Opt`. This is not surprising, as both `Opt` and `Pelt` are optimal search methods. The optimal nature of the methods implies that, with the same cost function, the output produced by the methods is identical, if the number of change points predicted by `Pelt` happens to be the same that was given as the parameter K to `Opt`. For example, if `Pelt` cost$_{L_2}$ happened to predict $K = 2$ change points for all Table 1 recordings, the results would be identical to what is presented in Section 5.1. However, as only the first and last change points predicted by `Pelt` are considered, and the complexity penalty was calibrated with this in mind, `Pelt` sometimes predicted more change points than `Opt` for the same recordings. The criterion for choosing the penalty value ensured that the start time of the three recordings that were mispredicted with `Opt`, was predicted approximately correctly.

statistic	$\hat{t}_o - t_{os}$	$\hat{t}_o - t_{oe}$	$\hat{t}_c - t_{cs}$	$\hat{t}_o - t_{os}$
minimum	-39	-161	2	-107
1st quartile	2	-18.5	5	-44
median	12	-10	7	-29
3rd quartile	23	-5	9	-23.5
maximum	100	11	66	9
variance	382	471	169	776
standard deviation	19.6	21.7	13.0	27.9

Table 5: Five-number summary and other statistics for `Pelt` cost$_{L2}$, pen = 18000.



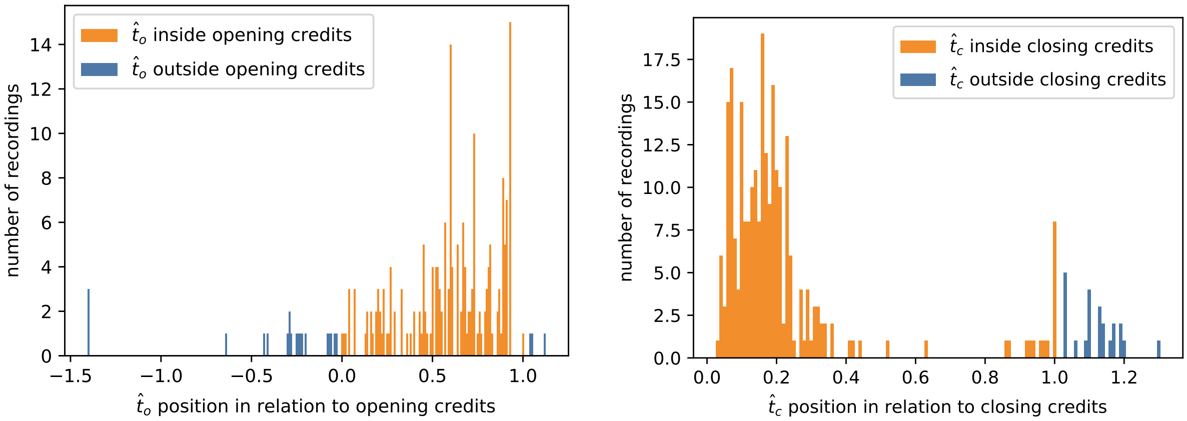
(a) Start of opening credits

(b) End of opening credits

(c) Start of closing credits

(d) End of closing credits

Figure 7: `Pelt` cost$_{L2}$ pen = 18000 output compared to actual change points of credits.



(a) Opening credits

(b) Closing credits

Figure 8: `Pelt` cost$_{L2}$ pen = 18000 output change point position in relation to credits.

One distinct difference between `Opt` and `Pelt` output can be seen in the position of $\hat{t}_c$ in relation to closing credits, visualised in Figure 5b and 8b. For `Pelt`, some $\hat{t}_c$ are clustered around the end of the closing credits, which did not happen for `Opt`. As it happens, this has more to do with the program structure than search method. Plotted in Figure 9 is the same data as in Figure 8b, but the results for the programs with a sneak peek at the end of the closing credits are coloured in red. It can be seen that the cluster around the end of closing credits consists almost solely from these programs. In the sample data all the series with a sneak peek have advertisement breaks, and therefore they were not included in the `Opt` sample.

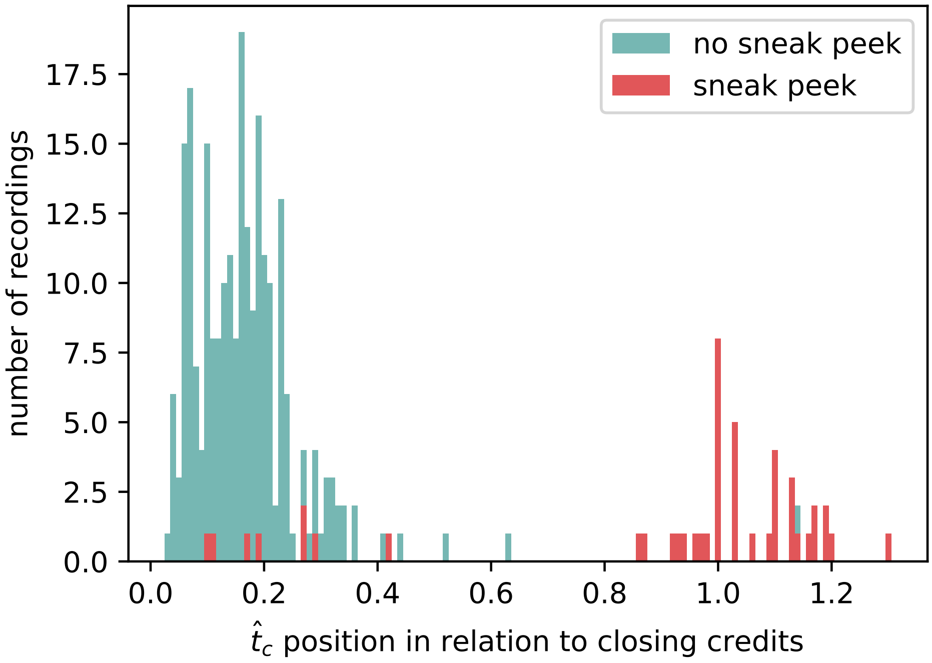


Figure 9: Figure 8b data with episodes with a sneak peek highlighted

6 Discussion

6.1 Interpretation of the results

The results described in Section 5 show that identifying the approximate location of both opening and closing credits is possible merely with viewing behaviour data. However, the results are not always accurate, and usually only the general location of credits was found, instead of the exact start and end time of the credits.

The assumption was proved false that only advertisement breaks would produce notable recesses in the views count within the program content. In Section 5.1 it was shown that with `Opt` as the search method unconventional structure of opening credits might cause the start of program to be detected less accurately. Although the first change point was not within one minute of the start of the program only in three recordings out of the 138 recordings in Table 1, it must be concluded that specifying the number of change points to be detected in advance might lead to impaired accuracy. This is true at least if no heuristics are used to evaluate the accuracy of `ruptures` output.

Results with the `Pelt` search method were more accurate, but as the complexity penalty was adjusted with the dataset itself, it is possible that the penalty value is overfitted for the data and does not produce generally as good results. However, the range of complexity penalties that got the start and end change points for the program approximately correct was quite wide, and the final penalty value was chosen from the middle of this range. This makes it more likely that the chosen penalty value works also for other recordings.

Unconventional program structure might affect the results, as was seen both with the opening credits in `Opt` and the closing credits in `Pelt`. Content that is generally considered interesting is viewed frequently, which is reflected in the location of change points.

6.2 Possible future considerations

There are multiple aspects of the credits detection process that could be improved or investigated further. These aspects can be roughly categorised into studying the viability of the concept, improving the computational speed of the process, or improving the accuracy of the results.

Questions regarding the sample size and data cleaning affect both the computational speed and accuracy. It could be investigated what is a sufficient sample size of user views per recording for the characteristic user behaviour pattern to emerge. The data cleaning procedure could possibly be improved by either doing more filtering, in order to get the data to display the characteristic user behaviour pattern more clearly, or by filtering less in order to gather the needed amount of views faster. The filtering criteria could be reconsidered, and the data could perhaps be preprocessed somehow in order to improve the results.

Approximate search methods could be tried out, in order to study whether they produce significantly worse result than `Pelt`, or if the results are almost as good. Other cost functions than cost_{L2} could be experimented with, and choosing a suitable value for the complexity penalty is a rather vast topic.

Perhaps the biggest question about the viability of the concept is, does the sample data represent all recordings. The sample consisted only of 30-90 min long episodes of TV series, which is a rather narrow subset of all broadcast programs. Short programs, programs that do not belong to a series, and programs from different genres, such as news, sports or documentaries could be tried out. Another variable to consider could be the program structure, although this sample already had programs with opening credits that are not located at the very beginning of the program, recapitulations of previous episodes, and sneak peeks into future episodes at the end of closing credits.

Offline signal change point detection is not the only possible approach for detecting the drastic changes in the viewer count. For example Bayesian methods could also be used, as mentioned in the beginning of Section 3.1. The approach examined in this thesis could perhaps be combined with some other approach, or with heuristics about the expected program structure, in order to get more reliable and accurate results.

6.3 Possible applications

There are a couple of areas where change points detected from user behaviour data could be useful. Firstly, change point detection could be combined with other methods for identifying video content, in order to find the exact position of credits. For instance, text in credits could be detected with optical character recognition (OCR) [11]. Change point detection from user behaviour could be used to ensure that the text found by OCR belongs to the credits of the entirely recorded program, and not to the closing credits of the previous one, which might be included in the beginning of a recording.

Secondly, with user behaviour data and some heuristic assumptions about program structure, it might be possible to monitor whether a recording contains the entire program it is supposed to contain. However, it might be nontrivial to come up with a heuristic that works with different kinds of program structures.

Lastly, if a supervised machine learning model was to be trained to detect the start and end times of recorded programs, user behaviour data might be useful for automatic labeling of training data, and for result validation. Nevertheless, the accuracy of change point detection might be lacking for this to be viable.

7 Conclusions

The goal of this thesis was to study whether opening and closing credits can be identified from an NPVR recording based solely on user viewing behaviour data. The verdict is that in most cases the general location of the credits can be found with reasonable accuracy. However, identifying the exact start and end times of credits is more difficult, and possibly cannot be done relying only on user viewing behaviour data.

The idea of using user viewing behaviour data for credits detection relies on the assumption that an average viewer is interested mainly in the core program content and does not watch the surplus content at the beginning and end of an NPVR recording. In general, this assumption seemed to hold. However, the program structure varies between programs, which affects what parts of the credits viewers consider interesting to watch. This in turn affects the structure of the user viewing behaviour data, and the location of the change points in relation to the credits.

Without more detailed information about the program structure, only the general position of credits can be detected. Nevertheless, user viewing behaviour based change point detection could perhaps be combined with other methods or heuristics for more accurate results.

References

[1] Barry, D. & Hartigan, J. A. *A Bayesian Analysis for Change Point Problems.* Journal of the American Statistical Association. 1993, vol. 88:421. P. 309–319. Available at: doi:10.2307/2290726.

[2] Bellman, R. *On a routing problem.* Quarterly of Applied Mathematics. 1958, vol. 16 :1. P. 87–90. Available at: doi:10.1090/qam/102435.

[3] Berrani, S.-A. & Manson, G. & Lechat, P. *A non-supervised approach for repeated sequence detection in TV broadcast streams.* Signal Processing: Image Communication. 2008, vol. 23:7. P. 525–537. Available at: doi:10.1016/j.image.2008.04.018.

[4] Ibrahim, Z. A. A. & Gros, P. *TV Stream Structuring.* ISRN Signal Processing. 2011, vol. 2011. P. 1–17. Available at: doi:10.5402/2011/975145.

[5] Killick, R. & Fearnhead, P. & Eckley, I. A. *Optimal Detection of Changepoints With a Linear Computational Cost.* Journal of the American Statistical Association. 2012, vol. 107:500. P. 1590–1598. Available at: doi:10.1080/01621459.2012.737745.

[6] Kompatsiaris, Y. & Merialdo, B. & Lian, S. *TV Content Analysis: Techniques and Applications.* CRC Press, 2012. ISBN 978-1-4398-5562-1.

[7] Lévy-Leduc, C. & Roueff, F. *Detection and localization of change-points in high-dimensional network traffic data.* The Annals of Applied Statistics. 2009, vol. 3:2. P. 637–662. Available at: doi:10.1214/08-AOAS232.

[8] Liu, S. & Wright, A. & Hauskrecht, M. *Change-point detection method for clinical decision support system rule monitoring.* Artificial Intelligence in Medicine. 2018, vol. 91. P. 49–56. Available at: doi:10.1016/j.artmed.2018.06.003.

[9] Lung-Yut-Fong, A. & Lévy-Leduc, C. & Cappé, O. *Distributed detection/localization of change-points in high-dimensional network traffic data.* Statistics and Computing. 2012, vol. 22:2. P. 485–496. Available at: doi:10.1007/s11222-011-9240-5.

[10] Manson, G. & Berrani, S.-A. *Automatic TV Broadcast Structuring.* International Journal of Digital Multimedia Broadcasting. 2010, vol. 2010. P. e153160. Available at: doi:10.1155/2010/153160.

[11] Ngo, C.-W. & Chan, C.-K. *Video text detection and segmentation for optical character recognition.* Multimedia Systems. 2005, vol. 10:3. P. 261–272. Available at: doi:10.1007/s00530-004-0157-0.

[12] Rabiner, L. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE. 1989, vol. 77:2. P. 257–286. Available at: doi:10.1109/5.18626.

[13] Reeves, J. & Chen, J. & Wang, X. L. & Lund, R. & Lu, Q. *A Review and Comparison of Changepoint Detection Techniques for Climate Data*. Journal of Applied Meteorology and Climatology. 2007, vol. 46:6. P. 900–915.

[14] Truong, C. & Oudre, L. & Vayatis, N. *Selective review of offline change point detection methods*. Signal Processing. 2020, vol. 167. P. 107299. Available at: doi:10.1016/j.sigpro.2019.107299.

[15] Tukey, J. W. *Exploratory Data Analysis*, volume 2. 1977. ISBN 0-201-07616-0.

[16] Verbesselt, J. & Hyndman, R. & Newnham, G. & Culvenor, D. *Detecting trend and seasonal changes in satellite image time series*. Remote Sensing of Environment. 2010, vol. 114:1. P. 106–115. Available at: doi:10.1016/j.rse.2009.08.014.

[17] Vert, J.-p. & Bleakley, K. Fast detection of multiple change-points shared by many signals using group LARS. *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.