CHALMERS | UNIVERSITY OF GOTHENBURG

MASTER'S THESIS

Joint segmentation of multiple features with application to analysis of epilepsy and music

JÓHANNA SIGMUNDSDÓTTIR

Department of Mathematical Sciences
Division of Mathematical Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2012

Thesis for the Degree of Master of Science

Joint segmentation of multiple features with application to analysis of epilepsy and music

Jóhanna Sigmundsdóttir

Department of Mathematical Sciences
Division of Mathematical Statistics
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, October 2012

Matematiska vetenskaper
Göteborg 2012

Abstract

Multivariate data is segmented into parts, called segments, with common characteristics. The segments are assumed to have an underlying model structure. It is of interest to see whether the characteristic changes originate from a subset of features or a unique feature. Various approaches can be taken when constructing such a method for multiple features. Three different methods are devised and compared via a simulation study. The methods use a penalized likelihood in different ways to estimate the number of segments. Two of these methods exhibit positive results and are examined further. They are shown to have different capabilities. One method favors the detection of coordinated segment changes at the expense of finding those that originate in a unique feature. The other method has an overall better performance, i.e., it is better at locating each and every characteristic change. The two methods are applied in two real life settings, one measuring physical changes in coordination with various music and the other measuring a range of physical changes in epilepsy patients. The methods capture the trends in the data but are not able to detect precisely when the music changes or the beginning and the end of a seizure.

Acknowledgement

I want to thank my supervisor, Rebecka Jörnsten, for her teachings, bright ideas and for turning my spirits towards the positive in our meetings. Johan Stigwall deserves credit for his helpful comments and assistance with the programs. I thank Magnús for his patience and moral support. In the end I express my gratitude to my parents for always being there for me. My mother deserves special mention for her endless patience and our comforting talks when all seemed so bleak.

Contents

1 Introduction

1.1 Objective

Our goal is to segment multivariate data into parts, called segments, with common characteristics. It is of interest to see whether these characteristics alter simultaneously over multiple features or if the change occurs in only one feature. There are various ways to construct methods for multivariate segmentation. Here, we devise three different methods and compare them via a simulation study.

We examine two real life datasets that are assumed to consist of segments. The first dataset contains measurements of different physical features, skin conductance and finger temperature, from a volunteer listening to music. It is of interest to see whether features alter characteristics simultaneously as the music changes. The second data is an epilepsy dataset which includes multiple features from a patient, containing both seizures and non seizures. We examine three time intervals containing a seizure and 60 seconds around it with non-seizure data. It is both of interest to see if the segments catch the start and the end of the seizure as well as whether it captures the trend in the data.

Two of the three different methods show positive results in the simulation study. One method favors the detection of coordinated segment changes at the expense of finding those that originate in a unique feature. The other method is better at locating each and every characteristic change. The methods are then applied to the real life datasets and are found to be able to capture the trend in the data but they cannot detect precisely when the music changes or the beginning and the end of a seizure.

1.2 Future work

Future work may include applying the joint segmentation methods further on the epilepsy data. It could be interesting to use the parameters from the obtained segments to classify seizures from non seizures. Two master thesis exist about the epilepsy data ([5], [10]). Both cover the topic of classifying seizures from non seizures. Neither of the previous theses use segmentation methods for the classification. A next step for the epilepsy data could be to combine the segmentation from this thesis to the classification used in the other theses.

Future work could also include improving the methods derived in this thesis by looking at ways to select the adjusting parameter when the segmentation algorithm is used on-line on large datasets.

2 Data

2.1 Music data

The data was collected from a volunteer at the Sahlgrenska Center for Brain repair and rehabilitation. The volunteer was asked to lie blindfolded on a bed with soundproof headphones on. In a 40 minute session, the volunteer first experienced a quiet 10 minute period, followed by 10 minutes of "terror" music (e.g. theme to "Jaws"), followed by 10 minutes of peaceful music and finally another 10 minute period of quiet. During the session, the subject's finger temperature and skin conductance were recorded. The dataset studied here (data not published) is a part of a larger study of 20 volunteers and different music [1]. The purpose of study is to see how subjects physically react to different types of music.

2.2 Epilepsy data

The data was collected by attaching three censor boxes (specially developed by IMEGO for this purpose) to epilepsy patients. The devices sample acceleration 50 times per seconds from three dimensions. A total of 51 features have been derived from the raw data by Wipenmyr [10] in a previous master thesis. All features are calculated each second and are based on a windowing over a 1 or 4 second interval. The DC-values are based on a 4 seconds windowing but other features on a one second windowing. This makes each DC value correlated to its adjacent values. Table 1 contains a short description of all the features. For a more thorough description of the features and the equipment we refer to Hildeman [5] and Wipenmyr [10].

All the features are standardized and smoothed with a moving average smoother of degree 5 [6], defined by:

$$y_{smoothed}(3) = (y(1) + y(2) + y(3) + y(4) + y(5))/5$$

The data examined in this thesis are from patient 7. That patient suffered from 11 seizures in approximately 2 days.

3 The segmentation method for one feature

The segmentation methods in this thesis are built on a method presented by Frank Pickard et al [9]. Their work is applicable to one feature data only. The data that is analyzed in [9] comprises comparative genomic hybridization data (CGH) that includes jumps in the mean and variance between

Feature name	Nr. of features	Description	Formula
DC values	9	Captures the gravitational acceleration	Average acceleration over 4 seconds
Signal magnitude area	3	Overall activity	$\frac{\sum_i w_i \frac{\lvert x_i\rvert+\lvert y_i\rvert+\lvert z_i\rvert}{3}}{\sum_i w_i}$, w_i is the weight for time instance i, and x,y,z are acceleration from each dimension
Vector magnitude	3	Overall activity	$\frac{\sum_i w_i \sqrt{x_i^2+y_i^2+z_i^2}}{\sum_i}$
Mean absolute magnitude difference	3	Overall activity	$\frac{\sum_i w_i \lvert x_i^2+y_i^2+z_i^2-1\rvert}{\sum_i}$
Periodicity	3	Compare the max largest frequency component to the average frequency magnitude	$\sum_{s=x,y,z} \frac{\max_{\omega\in\Omega(\lvert F_s(\omega)\rvert)}}{\text{mean}_{\omega\in\Omega(\lvert F_s(\omega)\rvert)}}$ F_s is the Fourier transform of the acceleration in the s direction during a time window ω
Frequency bands	18	0.75-2.25 Hz 2.25-3.75 Hz 3.75-5.25 Hz 5.25-8.25 Hz 8.25-13.25 Hz 13.25-25 Hz	FFT transforms acceleration in a 1 second window
Correlation	12	The linear correlation measures how well the sensors move in a phase. The circular correlation measures how similar the sensors move in a 90 degree phase shift	$\text{lincorr}(f,g) = \sum_{k=\kappa} \sum_{i=\{x_f,y_f,z_f\}} \sum_{j=\{x_g,y_g,z_g\}} \frac{\mathbf{a}_i[k]\mathbf{a}_j[k]}{\lvert\mathbf{a}_i[k]\rvert\lvert\mathbf{a}_j[k]\rvert}$ $\text{circorr}(f,g) = \sum_{k=\kappa} \sum_{i=\{x_f,y_f,z_f\}} \sum_{j=\{x_g,y_g,z_g\}} = = \frac{\mathbf{a}_i[k]\alpha_j[k]}{\lvert\mathbf{a}_i[k]\rvert\lvert\alpha_j[k]\rvert}$ $\mathbf{a}_j[k]$ is the acceleration in direction i at time instance k and $\alpha_j[k]$ is the same as $\mathbf{a}_j[k]$ but with a 90 degree shift

Table 1: The table contains a short description of the features. Different amount of features result from that they can both be calculated for each direction of each sensor separately or derived from all the directions at once.

segments. To apply this method to music and epilepsy data, we propose several extensions:

- Allow for more complex and different models in each segment.
- Formulate an on-line algorithm to handle large amounts of data.
- Allow for the segmentation of multiple features jointly.

We begin by reviewing the method [9] for one feature, and then extend it to multiple features.

3.1 Segmentation

The feature is assumed to consist of segments, each with an underlying segment specific model. The underlying segment structure is unknown but assumed to be of a certain model class

$$y_t = f_t + \epsilon_t,$$

where f_t is a known function, e.g., $f_t = \alpha_k$ (mean) or $f_t = \alpha_k + \beta_k t$ (line), $t \in (t_{k-1}, t_k]$ is a segment (time interval), k is the segment index and t_k are the unknown change-point locations. The random components ϵ_t are assumed to be independent and identically distributed, i.e., $\varepsilon_t \sim N(0, \sigma_k^2)\ \forall t \in (t_{k-1}, t_k]$. Given the interval $(t_{k-1}, t_k]$, estimates of the model parameters for segment k and the corresponding fitted value, $\hat{y}_t = \hat{f}_t$, are obtained through maximum likelihood. Because the errors are independent the total likelihood for the interval is obtained if we sum the log likelihoods for each segment. The log likelihood for the interval then becomes $L = \sum_{k=1}^{K} l_k$ where

$$l_k = -\frac{1}{2} \sum_{t=t_{k-1}+1}^{t_k} (\log(2\pi\sigma_k^2) + \frac{(y_t - \hat{y}_t)^2}{\sigma_k^2})$$

and K is the number of segments.

In reality though, neither the change-point locations nor the number of segments in each interval are known beforehand. We ask the following two questions:

1. How many segments are there?

2. Where are the change-points located?

These questions are solved by first assuming that the number of segments is known. Then, it is relatively easy to estimate where the change-points are located. Next, an appropriate number of segments is chosen by using a penalized log likelihood approach.

3.2 Constructing the cost matrix

The key component of the segmentation method [9] is the construction of a so called cost matrix, G. The entries in the cost matrix are obtained from the log likelihoods of each possible segment. The log likelihood for segment

k with known change-points t_k is:

$$\begin{aligned} l_k &= -\frac{1}{2}\sum_{t=t_{k-1}+1}^{t_k}(\log(2\pi\hat{\sigma}_k^2) + \frac{(y_t-\hat{y}_t)^2}{\hat{\sigma}_k^2}) \\ &= -\frac{1}{2}\sum_{t=t_{k-1}+1}^{t_k}(\log(2\pi) + \log(\hat{\sigma}_k^2)) - \frac{1}{2}\sum_{t=t_{k-1}+1}^{t_k}\frac{(y_t-\hat{y}_t)^2}{\hat{\sigma}_k^2} \\ &= -\frac{1}{2}\sum_{t=t_{k-1}+1}^{t_k}(\log(2\pi) + \log(\hat{\sigma}_k^2)) - \frac{1}{2}(t_k - t_{k-1}) \\ &= -\frac{1}{2}\sum_{t=t_{k-1}+1}^{t_k}\log(2\pi) - \frac{1}{2}(t_k - t_{k-1}) - \frac{1}{2}(t_k - t_{k-1})\log(\hat{\sigma}_k^2). \end{aligned}$$

where

$$\hat{\sigma}_k^2 = \frac{1}{t_k - t_{k-1}}\sum_{t=t_{k-1}+1}^{t_k}(y_t - \hat{y}_t)^2.$$

The full likelihood is

$$L = \sum_{k=1}^{K} l_k = \frac{-n}{2}(\log(2\pi)+1) - \frac{1}{2}\sum_{k=1}^{K}(t_k - t_{k-1})\cdot\log(\hat{\sigma}_k^2), \tag{1}$$

where n is the length of the interval. The log likelihood is dependent on the change-point locations through $(t_k - t_{k-1})\cdot\log(\hat{\sigma}_k^2)$. We now review the formula of the cost matrix G. It is defined as

$$G_{i,j} = \begin{cases} (i-j+1)\cdot\log(\hat{\sigma}_{ij}^2) & \text{if } j \geq i + l_{min} - 1 \\ \infty & \text{else} \end{cases} \tag{2}$$

where

$$\hat{\sigma}_{ij}^2 = \frac{1}{i-j+1}\sum_{t=i}^{j}(y_t - \hat{y}_t)^2. \tag{3}$$

Notice how element $[i,j]$ of G contains components of the second term of the likelihood in equation (1). That is, element $[i,j]$ contains the "cost" of having a segment in $[i,j]$. G will be the upper diagonal $n \times n$ matrix, where n is the length of the interval. Calculating G is computationally heavy and puts a limit on the n we can consider. We can set a minimum segment length l_{min} to speed up the calculation time.

3.3 Switching between models

We want to allow the segments to be characterized by different models. For example, a segment consisting of a mean and error can be followed by a segment that consists of a line and error. We select each segment model class using BIC (Bayesian information criteria) [4]. BIC is defined by:

$$\text{BIC} = -2 \cdot L + p \log(n)$$

where L is the log likelihood of the data points, p the number of model parameters and n is the number of data points. We calculate BIC_k for all model classes and each possible segment. Let us call $n_k = t_k - t_{k-1}$. For a segment from $t_{k-1} + 1$ to t_k BIC becomes:

$$\begin{aligned}
\text{BIC}_k &= -2 \cdot \frac{-1}{2} \sum_{t=t_{k-1}+1}^{t_k} (\log(2\pi\hat{\sigma}_k^2) + \frac{(y_t - \hat{y}_t)^2}{\hat{\sigma}_k^2}) + p_k \log(n_k) \\
&= \sum_{t=t_{k-1}+1}^{t_k} (\log(2\pi\hat{\sigma}_k^2) + \frac{(y_t - \hat{y}_t)^2}{\hat{\sigma}_k^2}) + p_k \log(n_k) \\
&= n_k(\log(2\pi) + \log(\hat{\sigma}_k^2) + 1) + p_k \log(n_k)
\end{aligned}$$

where

$$\hat{\sigma}_k^2 = \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} (y_t - \hat{y}_t)^2.$$

We construct a G matrix for each model class since different residuals are obtained from different models. We reduce the number of G matrices to one again by choosing the model that gives the lowest BIC value for each possible segment k.

The amount of model classes is restricted by the computational burden of the G matrix. Three different types of modeling are tested on artificial datasets:

1. Mean model, $y_t = \alpha$.
2. Choosing between mean ($y_t = \alpha$) and line ($y_t = \alpha + \beta t$) models.
3. Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha - \beta x_t$. All autoregressive parameters are estimated with the yule-walker equations [8].

Several other models are also applied to the music and epilepsy data:

4. Choosing between mean ($y_t = \alpha$) and line ($y_t = \alpha + \beta t$) model, using robust estimate of the regression parameters [6].

5. Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha$.

6. Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha - \beta t$, using robust estimate of the regression parameters.

7-9. Polynomial models of order 3 to 5, $y_t = \sum_{i=0}^{s} p_i t^i, s = 3 : 5$.

3.4 Finding the location of the change-points

After the cost matrix is obtained, a dynamical algorithm is used to find the optimal change-point locations [2], [9]. We define K_{max} as the maximum allowed number of change-points on the interval. $J_1(1, j)$ is the log likelihood for the interval $[1, j]$ and $J_{k+1}(1, j)$ is the optimal log likelihood for the same interval when allowing for $k + 1$ segments. Similarly, $J_1(i, j)$ is the log likelihood for one segment on the interval $[i, j]$. $J_{k+1}(1, j)$ can be obtained from $J_k(1, i)$ and $J_1(i+1, j)$ recursively. We denote $\hat{\sigma}^2_{ij}$ to be the variance on $[i, j]$, see equation (3). The recursion formula is:

$$J_1(i, j) = \sum_{t=i}^{j} (\log(2\pi\hat{\sigma}^2_{ij}) + \frac{\varepsilon_t^2}{\hat{\sigma}^2_{ij}}), 1 \leq i < j \leq n$$
$$J_{k+1}(1, j) = \min_h \{J_k(1, h) + J_1(h + 1, j)\}, \forall k \in [1, K_{max}].$$

The G matrix contains the part of the likelihood that is dependent on the location of the change-points, see equations (1) and (2). To be able to use columns and lines from the G matrix directly, we define $\tilde{J}$:

$$\tilde{J}_1(i, j) = (j - i + 1) \cdot \log(\hat{\sigma}^2_{ij}),\ \hat{\sigma}^2_{ij} \text{ is the variance on } [i, j]$$
$$\tilde{J}_{k+1}(1, j) = \min_h \{\tilde{J}_k(1, h) + \tilde{J}_1(h + 1, j)\}, \forall k \in [1, K_{max}].$$

No information is lost when using $\tilde{J}_k$ instead of J_k since $\tilde{J}_k$ will always be proportional to J_k. We notice that $\tilde{J}_1(1, j)$ is simply element $(1, j)$ in the G matrix whereas $\tilde{J}_1(h + 1, j)$ is column $h + 1$, line 1 to j in G matrix. It is the part of the log likelihood for one segment on the interval $[h + 1, j]$.

To calculate $\tilde{J}_{k+1}(1, j)$ it is convenient to build up the matrix $I(k+1, j) = \tilde{J}_{k+1}(1, j)$. The lines in I denote the number of allowed segments whereas

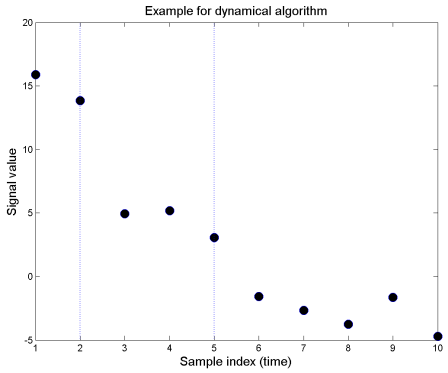


Figure 1: The data consists of three segments. The vertical lines denote the location of the change-points.

column j denotes point j in the data. Element (k, j) in I holds the part of the optimal log likelihood when allowing for k segments in the interval $(1, j)$. The I matrix is built up using the recursion formula for $\tilde{J}_{k+1}(1, j)$. We obtain $I(k, j)$ by using the previous line in the I matrix which holds the part of the log likelihood for k segments up to each point, h, as well as looking at the part of the log likelihood for one segment from $h + 1$ to j.

Since $\tilde{J}_1(1, j)$ is element $(1, j)$ of the G matrix, the first line of the I matrix is the same as the first line of the G matrix by definition. We use an example to demonstrate how the other elements are obtained.

Figure 1 contains the data used in the example. The data consists of three jumps in the mean with added random noise to each segment. A mean value is fitted and a G matrix constructed:

$$G = \begin{bmatrix} \infty & 1.46 & 10.58 & 13.96 & 17.65 & 22.77 & 27.40 & 31.87 & 35.39 & 39.59 \\ \infty & \infty & 7.37 & 9.75 & 12.59 & 17.22 & 21.35 & 25.35 & 28.31 & 32.10 \\ \infty & \infty & \infty & -6.78 & 0.92 & 9.12 & 13.03 & 16.62 & 18.57 & 21.96 \\ \infty & \infty & \infty & \infty & 1.65 & 7.43 & 10.54 & 13.54 & 15.14 & 18.09 \\ \infty & \infty & \infty & \infty & \infty & 4.73 & 6.67 & 8.78 & 9.56 & 11.96 \\ \infty & \infty & \infty & \infty & \infty & \infty & -0.96 & 0.55 & 0.27 & 3.12 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & -1.08 & 0.36 & 2.30 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1.63 & 2.73 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 3.12 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

The I matrix is:

$$I = \begin{bmatrix} \infty & 1.46 & 10.58 & 13.96 & 17.65 & 22.77 & 27.40 & 31.87 & 35.39 & 39.59 \\ \infty & \infty & \infty & -5.33 & 2.37 & 10.57 & 14.49 & 18.08 & 17.92 & 20.77 \\ \infty & \infty & \infty & \infty & \infty & -0.59 & 1.34 & 2.93 & 2.64 & 5.49 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & -1.67 & -0.23 & 1.71 \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1.44 \end{bmatrix}$$

To show how the elements of the I matrix are obtained, we calculate $I(2,4)$, i.e., the part of the log likelihood when having two segments on the interval $[1,4]$. Here the values colored red in G and I are used for the calculation.

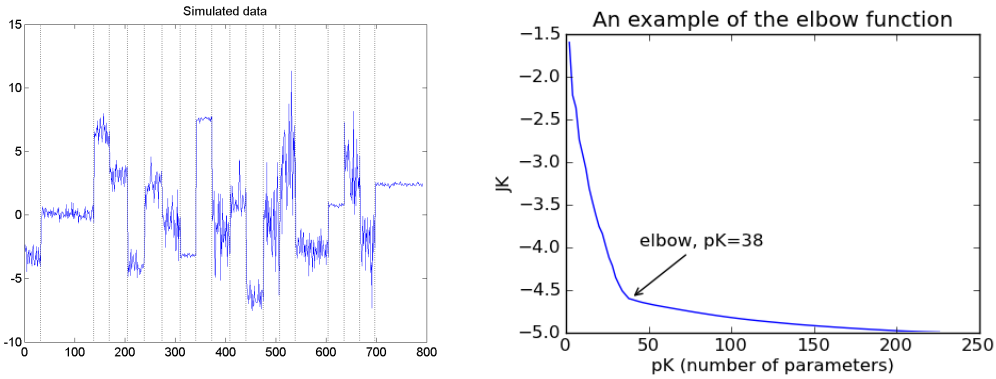
$$\begin{aligned} I(2,4) &= \tilde{J}_2(1,4) \\ &= \min_{h\in[1,4]} \{\tilde{J}_1(1,h) + \tilde{J}_1(h+1,4)\} \\ &= \min_{h\in[1,4]} \{I(1,h) + G(h+1,4)\} \\ &= \min\{[\infty, 1.46, 10.58, 13.96] + [9.75, -6.78, \infty, \infty]\} \\ &= \min\{[\infty, -5.33, \infty, \infty]\} \\ &= -5.33. \end{aligned}$$

The value of $I(2,4)$ is minimal when $h = 2$, i.e the optimal change-point location is point 2 for 2 segments on the interval $[1,4]$. We obtain infinity for other values of h since $l_{min} = 2$. We get the part of the log likelihood for the whole interval in the last column of I. The line number of I represents how many segments are used. We save the values of h to keep track of where the optimal change-points are located.

3.5 Choosing the number of segments

Many methods have been proposed for choosing the number of segments, for example BIC and AIC [4], but as has been proposed in [7], those methods tend to overestimate the number of segments. In [9], the authors compare a few different methods to choose the number of segments and recommend Lavielle's method [7]. This method is based on a penalized version of the log likelihood (also called roughness penalty) [4]. From now on we refer to this method as the "elbow" method. Let J_K be the likelihood for K segments estimated to be optimally located and p_K the number of model parameters in these K segments. $\tilde{J}_K$ is proportional to the optimal log likelihood J_K, see equation (1).

Let us consider an example. Figure 2a shows a feature consisting of 19 segments. Each segment has two parameters, a mean and a variance. Figure



(a) Simulated feature (b) Elbow

Figure 2: Figure 2a shows a feature consisting of 19 segments. Each segment has two parameters, a mean and a variance. Figure 2b depicts the trade off between number of segments and the fit of the model. It has the number of parameters, p_K, on the x axis, and $\tilde{J}_K$ on the y axis which is proportional to the log likelihood. When 38 parameters are reached $\tilde{J}_K$ does not decrease as drastically.

2b depicts the number of parameters on the x axis and $\tilde{J}_K$, on the y axis. We aim to find the point when adding more parameters does not result in a much lower $\tilde{J}_K$, that is the elbow of the curve $(p_K, \tilde{J}_K)$. In figure 2b this point is reached after 38 parameters.

Lavielle [7] does not allow for different number of parameters in different segments. We have therefore adjusted the method to allow for switching models between segments.

We build up a function that balances J_K (which has lower value as the number of segments increase) and p_K (which increases as the number of segments increases). A penalization constant β is also included:

$$\hat{K}(\beta) = \arg\min(J(K) + \beta p_K).$$

Here $\hat{K}$ is the estimator for the number of segments, K, and is dependent on β. Following Lavielle, we calculate β as follows:

"**Proposition [from [7]]**
There exist a sequence $K_1 = 1 < K_2 < ...$ and a sequence $\beta_0 = \infty > \beta_1 > ...$ with

$$\beta_i = \frac{J_{K_i} - J_{K_{i+1}}}{p_{K_{i+1}} - p_{K_i}}$$

such that $\hat{K}(\beta) = K_i, \forall \beta \in (\beta_i, \beta_{i-1})$. The subset $\{(p_{K_i}, J_{K_i}, i \geq 1)\}$ is the convex hull of the set $\{(p_K, J_K, K \geq 1)\}$."

β is the derivative of the curve (p_K, J_K). We look at the difference between consecutive β's to obtain how the slope is changing. Lavielle [7] proposes an automatic procedure to find this point. We adopt this procedure here to allow for a different number of model parameters in each segment:

1. Construct the normalized sequence. This was not presented in the example but is necessary for the automatic procedure.

$$\overline{J}_K = \frac{J_{K_{MAX}} - J_K}{J_{K_{MAX}} - J_1}(K_{MAX} - 1) + 1, \ 1 \leq K \leq K_{MAX}.$$

   Then $\overline{J}_1 = K_{MAX}$, $\overline{J}_{MAX} = 1$ and the slope is of average -1.

2. Calculate:

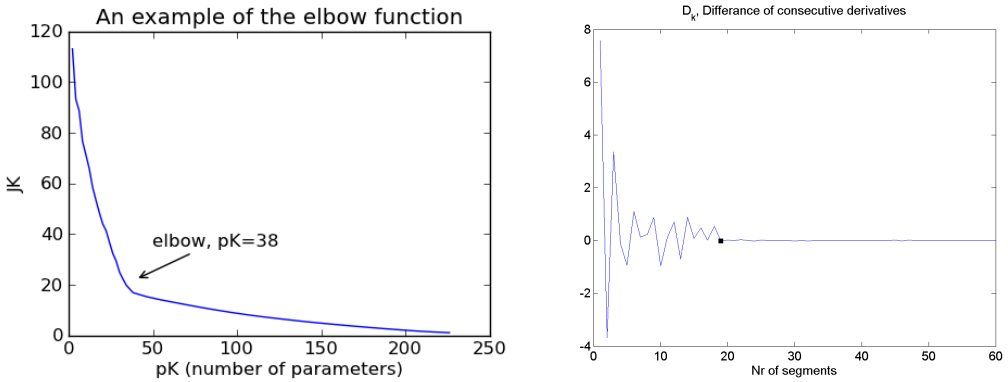
$$D_K = \frac{\overline{J}_{K-1} - \overline{J}_K}{p_K - p_{K-1}} - \frac{\overline{J}_K - \overline{J}_{K+1}}{p_{K+1} - p_K}, \ 2 \leq K \leq K_{MAX} - 1$$

   and set $D_1 = \infty$. Then, the estimate of K is

$$\hat{K} = \max\{1 \leq K \leq K_{MAX} - 1 \text{ such that } D_K > S\},$$

where S is an adjusting parameter. The distribution of $\max_K D_K$ is not known. Low values of S usually lead to an overestimate of the number of segments, while high values tend to underestimate them. To estimate S we examine the values of D_K and see when the difference between consecutive derivatives has leveled off. When we allow for a switch between model types this procedure is an approximation since we ignore the change in number of parameters between segments. In that case, we may be biasing slightly towards more segments.

When this method is applied to the data we use $\tilde{J}$ instead of J. $\tilde{J}$ is proportional to J. This is only done because it is convenient to calculate it. It has no effects on the results. Figure 3a shows the plot of $(p_K, \bar{J}_K)$ where $\tilde{J}$ is used in the calculation instead of J. The data used is the same as in



(a) Simulated feature

(b) Elbow

Figure 3: Figure 3b shows $(p_K, \bar{J}_K)$ where $\tilde{J}$ is used in the calculation instead of J. There is a clear point where adding segments does not lower $\bar{J}_k$ considerably. Figure 3b shows the curve (K, D_K) again where again $\tilde{J}$ is used in the calculation instead of J. We see that at point 19 the values of D_K are lower than before. This K tells us how many segments the dataset contains.

figure 2a. Figure 3b shows the corresponding (K, D_K). We clearly detect the location where the values of D_K have leveled off. The black box located at $\hat{K} = 19$ shows this point. We choose S so this point is reached. This is consistent with the dataset which truly 19 has segments. Since this data is simulated with clear segments, the plot of (p_K, D_K) is easily interpreted. When these curves are plotted for real data this point may not be as clear. For a more thorough discussion we refer to Lavielle [7].

3.6 On-line method

Because the G matrix is computationally heavy, one cannot apply this method to a large dataset. We solve this problem by applying the method to consecutive intervals. However, the maximum length of a segment is then limited to the length of the interval chosen and parameters are forced to change at the interval endpoints. To resolve this we use the information of the previous segmented interval and construct an on-line segmentation algorithm that moves through the dataset. Two questions arise when extending the algorithm in this way:

- Where should we construct the next G matrix when sliding through the data?

- How can we construct segments that are longer than the length of the interval chosen?

The answer to these questions are discussed below.

Extensions

When we move to an on-line algorithm we still need to allow for a minimum segment length l_{min}. Therefore, if we are segmenting some limited interval there will be no change-points in the l_{min} first and last points. This is taken into consideration in the on-line algorithm so that any point in the data series can be a change-point. Let us call the starting point of our interval t_1 and the endpoint t_n. We consider two cases.

- **Change-point detected in the interval** $[t_n - 2l_{min}, t_n]$.
  If there is a change-point t_x detected in the interval $[t_n - 2l_{min}, t_n]$ we want to construct the G matrix at $t_x - l_{lmin}$. If we start constructing the G matrix somewhere on the interval $[t_x - l_{min} + 1, t_n]$ we will get an interval that never allows for a jump. We could also start constructing the G matrix even earlier but that only results in a slower algorithm. $t_x - l_{min}$ is therefore the optimal place to start the next G matrix. See figure 4.

- **No change-point detected in the interval** $[t_n - 2l_{min}, t_n]$.
  If no change-point was detected in the interval, we start constructing the G matrix at $t_n - 2l_{min}$. Starting any later we again get an interval that never allows for a jump. See figure 5.

Updated version of the G matrix

When constructing an updated G matrix which starts from a previous interval, we only need to change the first line of the G matrix. That line keeps information about the segments that have a starting point at the beginning of the interval. Let us call the last change-point found t_{last}. The segments need to be able to start at $t_{last} + 1$. The first line of the updated G matrix becomes:

$$g_{1,j} = \begin{cases} (t_{last} - j + 1) \cdot \log(\hat{\sigma}_{t_{last}j}) & \text{if } j \geq l_{min} \\ \infty & \text{else} \end{cases}$$

where

$$\hat{\sigma}^2_{t_{last}j} = \frac{1}{t_{last} - j + 1} \sum_{t=t_{last}}^{j} (y_t - \hat{y}_t)^2.$$

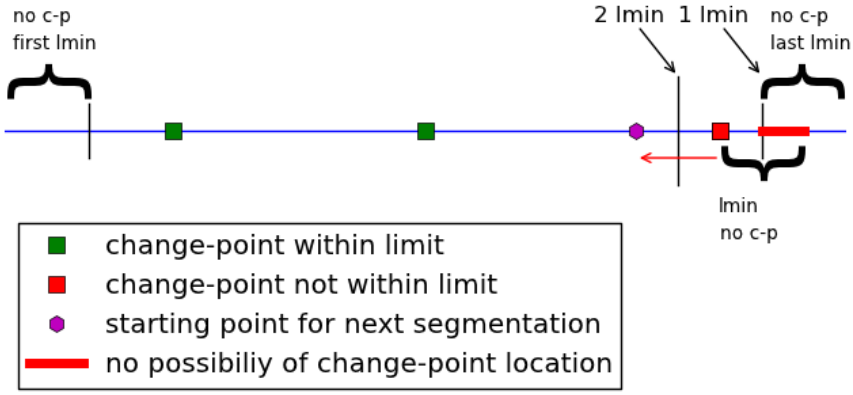


Figure 4: Starting point location when a change-point is found in the interval $[t_n - 2l_{min}, t_n]$.

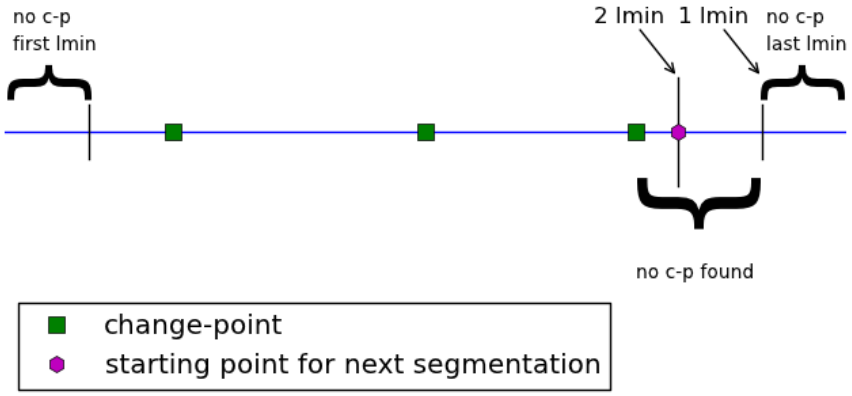


Figure 5: Starting-point location when no change point is found in the interval $[t_n - 2l_{min}, t_n]$.

Updating the G matrix in this fashion allows us to construct segments that are longer than the size of each consecutive interval. The result is an on-line algorithm that can segment one feature where the segments can be from different models.

The on-line method makes the code applicable to larger datasets. However, there are also drawbacks. When choosing the number of segments in each interval we use the elbow method with the same adjusting parameter S for all the intervals. We need to assume that the β's in each interval are of the same order of magnitude. If they are not, the adjusting parameter S might not be the same for each consecutive interval which results in an incorrect estimation of the number of segments in some intervals. Simulation studies are needed to find this adjusting parameter S. The ability to plot (p_k, J_k) for the whole interval and find the elbow visually is not possible when segmenting on-line.

4 Joint segmentation

We consider three approaches to extend the segmentation algorithm from one to multiple features. Our goal is not only to find the optimal change-point locations but also to determine whether these change-points are coordinated over multiple features. For the first objective (optimal change-points) we will use the likelihood ratio test [3] and the elbow method. For the second objective (coordination of change-points) we will construct three different methods. Before we discuss these approaches, we describe how the on-line method is generalized to multiple features.

4.1 On-line method for multiple features

In the multivariate setting all features are segmented for the same intervals. To extend the on-line procedure to multiple features, we need to adjust for different optimal starting points between features (see section 3.6). We calculate a starting point for each feature separately and start the new interval from the starting point furthest behind. If no change-points are detected in $[n-2l_{min}, n-l_{min}]$ the new starting point becomes $n-2l_{min}$ for all features. Complications arise when there is a change-point, t_f, detected in the interval $[n-2l_{min}, n-l_{min}]$ and the new starting point becomes $t_f - l_{min}$. Figure 6 shows that the interval $[t_f - l_{min}, n-2l_{min}]$ will not allow for any change-points since the minimum segment length is l_{min}. We solve this by changing the minimum segment length l_{min} of the first segment of all features that

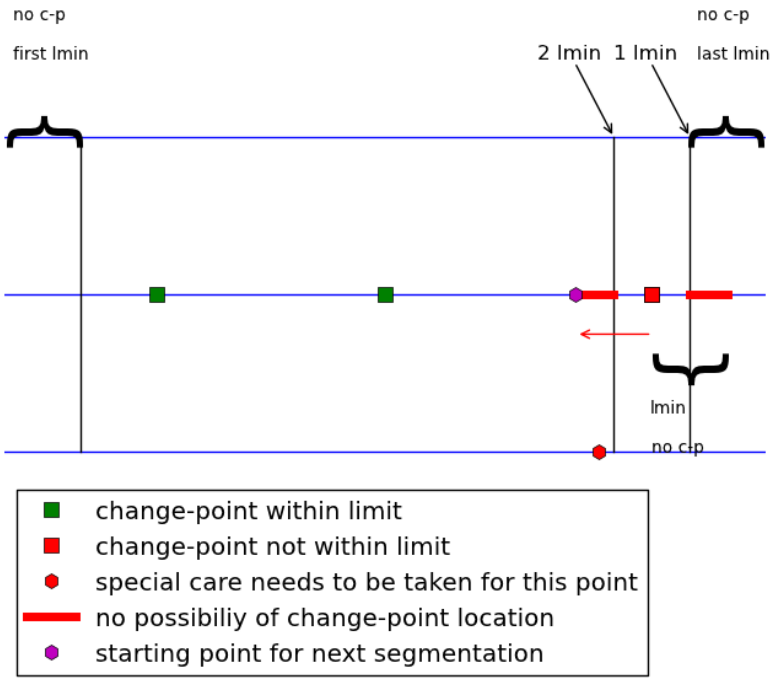


Figure 6: Complications arise when we expand the on-line algorithm to handle multiple features. We find a new starting point for each feature (see section 3.6) and choose the one furthest behind to start the segmentation of all features. However, we then obtain an interval that never allows for a change-point. This is solved by allowing the first segment of all features that have a change-point in the interval $[t_f - l_{min}, n - 2l_{min}]$ to have a lower segment length. The change-points can then be detected again.

have a change-point in the interval $[t_f - l_{min}, n - l_{min}]$. The change-points can then be detected again.

4.2 Joining and removing change-points with a likelihood ratio test

We use the likelihood ratio test to both join (coordinate) change-points across features and remove change-points.

Joining the change-points

When multiple features are segmented separately, change-points in different features may lie close to each other. Joining these change-points results in a reduced number of model parameters as well as giving further knowledge on how the features are linked. We use the likelihood ratio test to decide whether this model simplification is justifiable.

Let $t^f_{j_f}$ be a change-point number j_f in feature f. To coordinate $t^f_{j_f}$ with all the change-points of other features that lie within a certain radius, it has been moved to $\tau^f_{j_f}$. The new change-point, $\tau^f_{j_f}$, is the mean of all the change-points within this radius. We state the following hypothesis that the change-points are:

$$
\begin{aligned}
H_0 &: \text{The change-points, } \tau^f_{j_f}, \text{ are coordinated} \\
&\quad \text{(maximum one change-point in each feature).} \\
H_A &: \text{The change-points, } t^f_{j_f}, \text{ are not coordinated.}
\end{aligned}
$$

Only the likelihood of segments whose endpoints move get affected by the change-point joining. We use that fact and equation (1) (see section 3.2) to obtain the test statistic Λ_f for feature f:

$$
\begin{aligned}
\Lambda_f &= -2\log\left(\frac{\text{likelihood for the null model}}{\text{likelihood for the alternative model}}\right) \\
&= -2\cdot\frac{-1}{2}\sum_{k=1}^{K}\sum_{t=\tau^f_{k-1}+1}^{\tau^f_k}\log\Big(\frac{1}{\tau^f_k-\tau^f_{k-1}}\sum_{t=\tau^f_{k-1}+1}^{\tau^f_k}\varepsilon_t^2\Big) \\
&\quad + 2\cdot\frac{-1}{2}\sum_{k=1}^{K}\sum_{t=t^f_{k-1}+1}^{t^f_k}\log\Big(\frac{1}{t^f_k-t^f_{k-1}}\sum_{t=t^f_{k-1}+1}^{t^f_k}\varepsilon_t^2\Big) \\
&= \sum_{k=j_f}^{j_f+1}\sum_{t=\tau^f_{k-1}+1}^{\tau^f_k}\log\Big(\frac{1}{\tau^f_k-\tau^f_{k-1}}\sum_{t=\tau^f_{k-1}+1}^{\tau^f_k}\varepsilon_t^2\Big) \\
&\quad - \sum_{k=j_f}^{j_f+1}\sum_{t=t^f_{k-1}+1}^{t^f_k}\log\Big(\frac{1}{t^f_k-t^f_{k-1}}\sum_{t=t^f_{k-1}+1}^{t^f_k}\varepsilon_t^2\Big)
\end{aligned}
$$

We calculate Λ_f for all features f that have an altered change-point. The test statistic $\sum_f \Lambda_f$ is compared with a χ^2-distribution with degrees of freedom equal to the total number of parameters dropped. If the obtained p-value is below a certain adjusting parameter α the null hypothesis is rejected and we do not join the change-points.

We use backward selection [4] to decide in which order the change-points are joined. We join the best candidates first and then continue to the second best. The best candidates are the ones that obtain the highest p-value when tested. For each change-point we calculate the p-value for joining it with its neighbors. The change-points are joined for the highest p-value above an adjusting parameter α_1. Then all the p-values are updated and this is repeated until α_1 is reached.

Removing change-points

We remove change-points in a similar manner as joining them. We call the change-point that we consider removing t_r. We want to test whether t_r can be removed from all features f it exists in. We state the following null hypothesis:

$$H_0 : t_r \text{ can be removed from the features it exists in.}$$
$$H_A : t_r \text{ cannot be removed from those features.}$$

We call τ^f the set of change-points for feature f where t_r has been removed and t^f the original set of change-points. We calculate the test statistic for H_0 in the same way as in the previous chapter. Instead of only having change-points moved between the null model and the alternative model we have

dropped one change-point. Λ_f is:

$$\begin{aligned}
\Lambda_f &= -2\log\left(\frac{\text{likelihood for the null model}}{\text{likelihood for the alternative model}}\right) \\
&= -2\cdot\frac{-1}{2}\sum_{k=1}^{K}\sum_{t=\tau_{k-1}^f+1}^{\tau_k^f}\log(\frac{1}{\tau_k^f-\tau_{k-1}^f}\sum_{t=\tau_{k-1}^f+1}^{\tau_k^f}\varepsilon_t^2) \\
&+ 2\cdot\frac{-1}{2}\sum_{k=1}^{K}\sum_{t=t_{k-1}^f+1}^{t_k^f}\log(\frac{1}{t_k^f-t_{k-1}^f}\sum_{t=t_{k-1}^f+1}^{t_k^f}\varepsilon^2) \\
&= \sum_{t=\tau_{r-1}^f+1}^{\tau_{r+1}^f}\log(\frac{1}{\tau_k^f-\tau_{k-1}^f}\sum_{t=\tau_{k-1}^f+1}^{\tau_k^f}\varepsilon_t^2) \\
&- \sum_{k=r}^{r+1}\sum_{t=t_{k-1}^f+1}^{t_k^f}\log(\frac{1}{t_k^f-t_{k-1}^f}\sum_{t=t_{k-1}^f+1}^{t_k^f}\varepsilon_t^2)
\end{aligned}$$

We calculate Λ_f for all f that have t_r. $\sum_f \Lambda$ is compared with a χ^2 distribution with degrees of freedom equal to the number of parameters dropped. Like when we join change-points, we use backward selection and stop removing change-points when all p-values are below a certain adjusting parameter α_2.

We now describe and compare three different joint segmentation methods defined by the order of removing and joining of change-points.

4.3 Sequential method 1 (SeqM1)

Sequential method 1 (SeqM1) is built up in the following way:

1. We segment each feature, using the elbow method to obtain the optimal number and location of change-points for each feature.

2. The likelihood ratio test is then applied to join the change-points across features afterwards. It is done in a backward selection setting as described in section 4.2

The method has one adjusting parameter for joining the change-points. The S parameter for the elbow can however be different for each feature.

4.4 Sequential method 2 (SeqM2)

Sequential method 2 (SeqM2) is in some sense the mirror image of SeqM1. For each feature the method proceeds in the following way on each interval:

1. Each feature is standardized to $[-1, 1]$.
2. A G matrix is constructed for each feature.
3. All the G matrices are summed into one matrix.
4. We proceed as if the G matrix in 3. comes from one feature. The change-point locations are estimated.
5. It is not known which features contribute to each change-point. That is solved by using a likelihood test. We use backward selection to remove the change-point for each feature separately.

Since we work with the sum of all the G matrices we are never able to find change-points less than l_{min} apart between features. The method has two adjusting parameters, S for the "elbow" and α for the likelihood ratio test.

4.5 All subset selection method (ASM)

A computationally heavy alternative is the all subset selection method (ASM). The method looks good in theory but in practice it needs to be restricted due to computational load and is therefore not likely to be competitive. ASM consists of taking the following steps:

1. Construct the G matrix for each feature.
2. Estimate the optimal change-point locations and log likelihoods for each feature when we allow for 1 to K_{max} segments.
3. The total log likelihood is the sum of the log likelihoods of each feature. We calculate the total likelihood for each combinations of number of segments. See table 2 for clarification.
4. For each combination of segments there may be several change-points that can be joined across features. Since it could be optimal to join only some of these change-points we need to calculate the likelihood for each combination of these unions, see table 3.

5. We obtain all these different combination of segmented features. We note that often these models have the same number of parameters, e.g., allowing for 1 segment in the first feature, 2 segments in the second and vice versa. For each number of parameters we choose the best model, i.e., the model with the lowest negative log likelihood.

6. When we have the total log likelihood for different numbers of parameters we use the elbow method to select the number of parameters. Thereby we obtain the number of segments for each features as well as which change-points are joined.

Comb. of segments	Nr of segments											
Feature 1	1	2	3	1	2	1	1	2	3	1	2	3...
Feature 2	1	1	1	2	2	2	3	3	3	1	1	1...
Feature 3	1	1	1	1	1	1	1	1	1	2	2	2...
Total likelihood	$\sum$	..	..	..	..	..	..	..	..	..	..	...

Table 2: For each feature we calculate the likelihood of having k segments, where $k = 1, ..., K_{max}$. Then the total likelihood is calculated for each combination of different segment numbers.

Combine change-point group 1	1	0	1	0	1	0	1
Combine change-point group 2	0	1	1	0	0	1	1
Combine change-point group 3	0	0	0	1	1	1	1

Table 3: When looking at each combination of segments there may be several change-points that have neighbors in other features. We need to calculate the likelihood of joining all these groups as well as the likelihood of joining some of them. The combination for joining three change-point groups can be seen above.

Since this method is computationally heavy, we cannot allow for short segments because the joining possibilities will become too many. The method has only one adjusting parameter S for the elbow.

4.6 Comparing the joint segmentation methods

We compare the three methods in terms of:

1. Overall ability to find the correct change-points in the right features.

2. Ability to find unique change-points, i.e., change-points that only appear in one feature.

3. Ability to find common change-points, i.e., change-points that are coordinated over multiple features.

All change-points found are either located where there is a true change-point or not. We consider each found point to be correctly located if it is within 1 point radius from the true change-point. In part 1 of table 4 we see this classification.

1	Nr of true found = M1	Nr of false found = M2	Total nr found = N M1+M2=N
2	Nr of true common found = A	Nr of false common found = B	Total nr of common found = E A+B=E
	Nr of true unique found = C	Nr of false unique found = D	Total nr of unique found = F C+D=F
			Total nr of found jumps = N E+F=N
3	Nr of true common = G	Nr of true unique = H	Total nr of true jumps = I G+H=I

Table 4: Each found point is either correctly located or not.

This classification does not take into consideration whether the change-points are classified as common to some features or unique. Each point found is either a correct or false common change-point or it is a correct or false unique change-point. The common and unique classification of the found change-points is displayed in part 2 of table 4. Points found in 1 point radius of true unique points are considered true. Change-points within 1 point radius of the correct change-point are considered true common if all the change-points under consideration are at the exact same place.

A true common change-point can be defined in different ways. The most strict definition is detecting the common change-point in the true features and nowhere else. We can also detect a change-point in all the true features as well as in extra features. Another possibility is finding a change-point in some of the true features but not all. We use two types of definitions of a true common change-point:

1. The exact true features have a change-point (strict definition).

2. All the true features have change-points, as do possibly additional features (loose definition).

Part 3 of table 4 contains the classification of the true change-points.

Using table 4, the true and false positive rates can be defined. In the following definitions a stands for all change-points, i.e., finding a change-point at the right place for the right feature, c the common change-points and u the unique change-points.

$$\begin{aligned} \mathrm{TPR}_a &= \frac{\mathrm{M1}}{\mathrm{I}} & \mathrm{FPR}_a &= \frac{\mathrm{M2}}{\mathrm{N}} \\ \mathrm{TPR}_u &= \frac{\mathrm{C}}{\mathrm{H}} & \mathrm{FPR}_u &= \frac{\mathrm{D}}{\mathrm{F}} \\ \mathrm{TPR}_c &= \frac{\mathrm{A}}{\mathrm{G}} & \mathrm{FPR}_c &= \frac{\mathrm{B}}{\mathrm{E}} \end{aligned}$$

In addition to these definitions we define true and false positive rates for the overall detection of finding unique and common change-points. A found change-point is true if it is either a true unique or a true common.

$$\mathrm{TPR} = \frac{\mathrm{A}+\mathrm{C}}{\mathrm{I}} \qquad \mathrm{FPR} = \frac{\mathrm{B}+\mathrm{D}}{\mathrm{N}} \tag{4}$$

We see that this definition is dependent on how we define a true common. The subscript *ucs* will stand for using the strict definition of common true change-point and *ucl* will use the loose definition. Since these true and false positive rates are directly related to the true and false positive rates of unique and common change-points, results from their calculation are placed in appendix A.1.

We compare the methods by fixing the adjusting parameters so that $\mathrm{FPR}_a \in [0, 0.1]$ and compare the highest corresponding TPR_a for these fixed FPR_a. With the adjusting parameters fixed, we compare the other false and true positive rates to examine the performance in finding unique and common change-points.

5 Results

We compare the methods on both simulated and real datasets. The methods are implemented in Matlab. The programs are built on freely available Matlab code from [9]. The code for simulating the artificial datasets is by Johan Stigwall.

5.1 Simulated datasets

The characteristics of the simulated datasets are controlled by a set of parameters. Those parameters are: number of change-points, minimum segment length, maximum segment length, number of features, variance of noise for each segment and variance of step height for each segment. We can also choose which kind of models the segments are allowed to have.

The datasets are simulated with the following procedure:

- Simulate change-point location, based on minimum and maximum length of segments as well as the number of change-points.
- Select which features jump at those change-point locations.
- Generate the segments for each feature. The same model type can be selected for all segments or we can allow for a switch between model types. The mean and variance are randomized by using variance of noise and variance of step hight. Those parameters represent the maximum noise and step height for each segment.
- There are a few parameters that are fixed constants in the simulation. The minimum step size = 2, which means that consecutive segments must have mean difference of at least 2. This minimum is set to obtain clear jumps in mean. The maximum parameter for slope, $h_\beta = 5$. h_β is the maximum height gained in a segment. The slope is therefore constricted by the segment length.
- When we use more models than only a mean and a line, more parameters are needed.

We analyze three sets of simulated data, SIM1, SIM2 and SIM3. Each set contains 100 datasets which have common characteristics:

1. SIM1: Three features, long segments, each segment has its own mean and variance.
2. SIM2: Three or four features, both long and short segments, each segment has its own mean and variance.
3. SIM3: Three or four features, both long and short segments, each segment can either consist of its own mean and variance or a slope and variance.

	SIM1	SIM2	SIM3
Model types	α	α	α or $\alpha + \beta x$
Nr of steps	From 9 to 25	From 9 to 25	From 9 to 25
Min segment length	From 20 to 30	From 5 to 20	From 5 to 20
Max segment length	From the min segment length+5 to 50	From the min segment length+5 to 40	From the min segment length+5 to 40
Nr of features	3	3 or 4	3 or 4
Variance of noise	From 0.5 to 2	0.5 to 2	0.5 to 2
Variance of step height	From 3 to 10	From 3 to 10	From 3 to 10

Table 5: Parameters used for simulating datasets.

The parameter values are different for each of the 100 datasets but lie within the same range for each of the three groups. Table 5 contains the parameters used for the simulation of the three sets. Since the parameters are allowed to vary between each of the 100 datasets we obtain large variability within each of the three sets (SIM1, SIM2 and SIM3). This is done to see which method has better performance given data of broad variety. We do not know the exact structure of the real dataset and we therefore want to find the method that has the best overall performance. Figures 7, 8 and 9 show examples of a dataset from each simulation set.

5.2 Simulation studies

The methods are applied to the simulated datasets. The adjusting parameter for the elbow in SeqM1 is kept the same for all features as otherwise we need a complicated search for the optimal adjusting parameters. We then compare the false and true positive rates described in section 4.6. The overall true and false positive rates of finding both unique and common change-points can be found in the appendix A.1. This comparison is carried out for all the simulation studies. In the first simulation study we compare the three methods. The second simulation study compares two methods, SeqM1 (Sequential method 1), and SeqM2 (Sequential method 2). The third simulation study compares SeqM1 and SeqM2 on a simulated set with a more complicated structure. Further descriptions of the simulated datasets can be found in section 5.1 and table 5.

5.2.1 Simulation 1

We segment each dataset in SIM1 with all methods. The adjusting parameters are fixed so that $FPR_a \in [0, 0.1]$ (see section 4.6). Figure 10 shows the true and false positive rates.

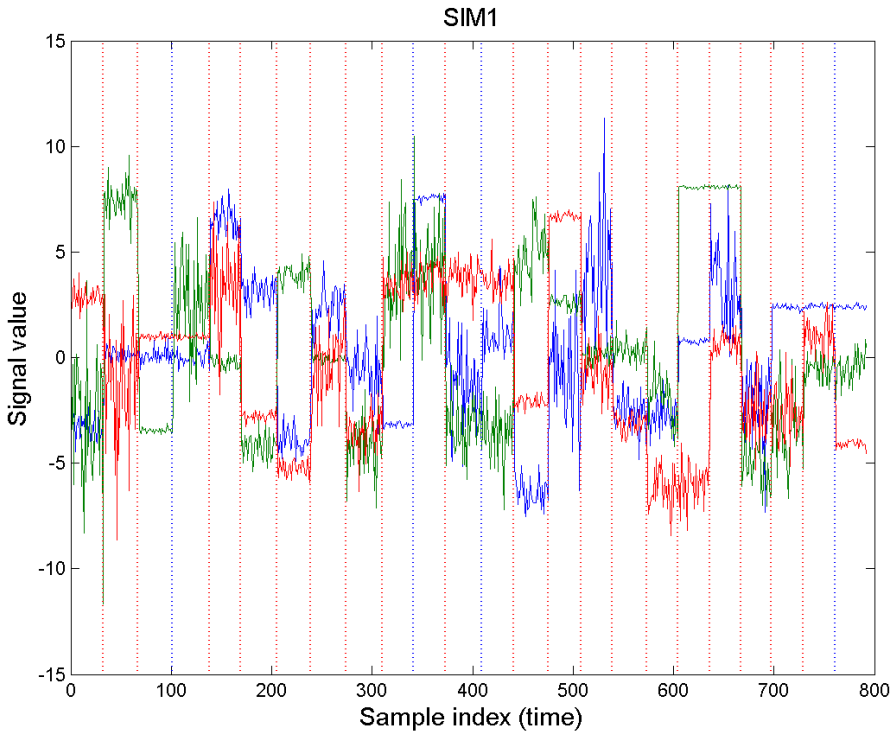


Figure 7: SIM 1: An example of the dataset used for comparing all the methods. The parameters used for the simulation of this dataset can be found in table 5. Here the blue dotted lines represent unique change-points while the red dotted lines represent common change-points. As we see from the figure the jumps are clear but the variance can vary significantly making the change-points harder to detect.

From figure 10a, we notice that SeqM1 and SeqM2 outperform ASM slightly. SeqM1 does have a large tail in the true positive rates for the whole dataset which indicates that although the method has overall good performance it can fail occasionally. Figure 10b shows how SeqM1 outperforms the other methods in finding the unique change-points. This is expected since SeqM1 focuses on each feature when segmenting. We also notice how SeqM1 and SeqM2 have lower false positive rates than ASM for finding unique change-points. From figures 10c and 10d we see how methods SeqM1 and SeqM2 also outperform ASM in terms of common change-points. In figure 10d we use a more generous definition of a true common change-point. If we join more features than we should, we still consider the change-point true. We notice that SeqM2 performs considerably better when the loose definition

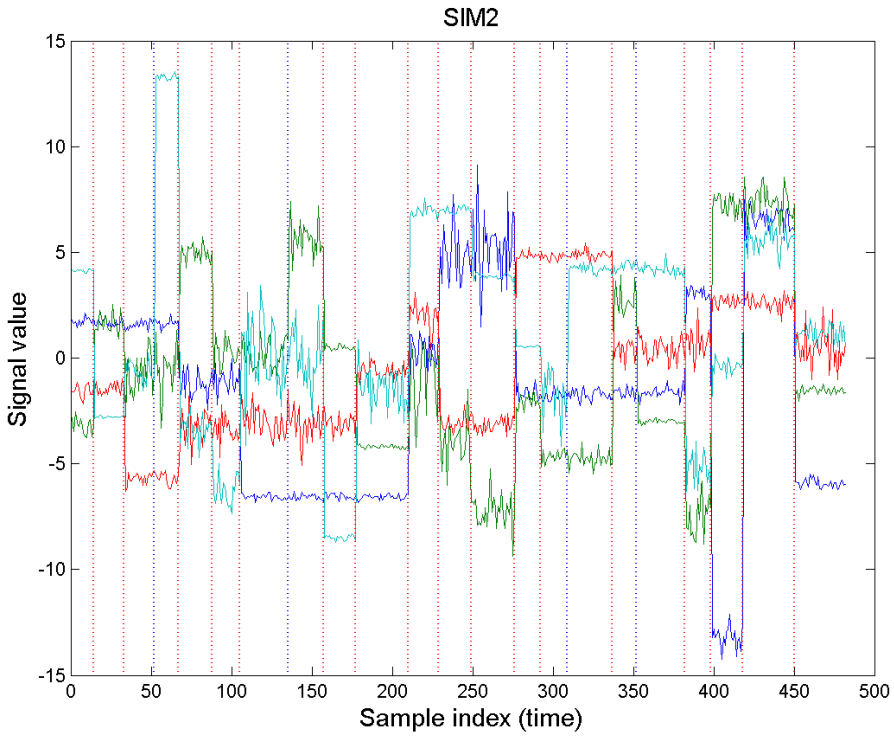


Figure 8: SIM 2: An example of the dataset used for comparing SeqM1 and SeqM2. We see that the segments can be both short and somewhat longer. The parameters used for the simulation of this dataset can be found in table 5. Here the blue dotted lines represent unique change-points while the red dotted lines represent common change-points. As we see from the figure the jumps are clear but the variance can vary significantly making the change-points harder to detect. Despite that the variances seem to be lower here than in figure 7 datasets from SIM2 can also contain segments that have higher variances. This figure is only one example of 100.

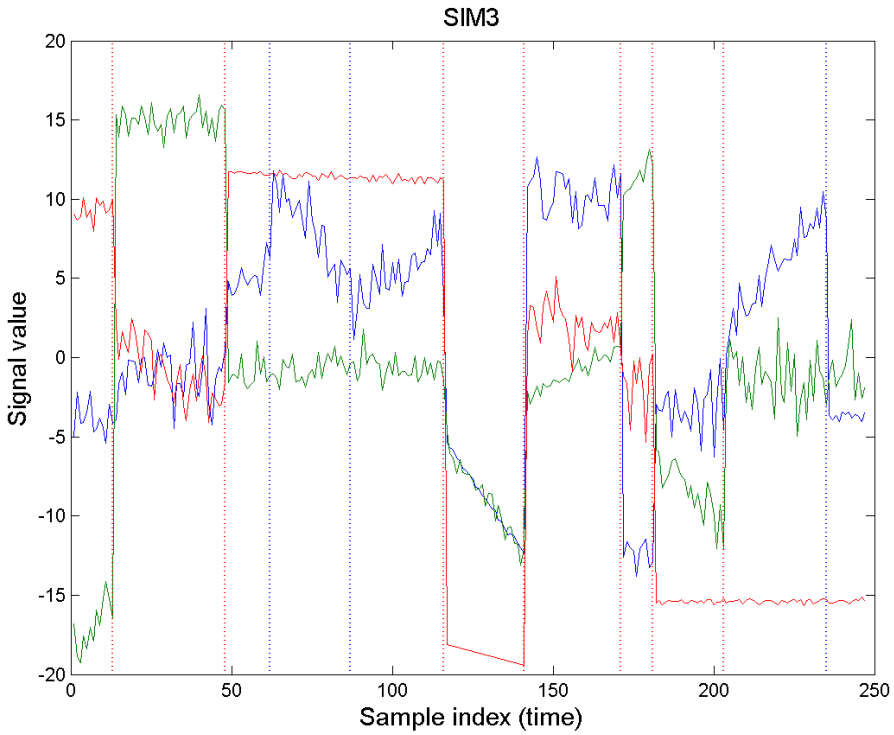


Figure 9: SIM 3: An example of the dataset used to see how SeqM1 and SeqM2 perform when the data has a more complicated structure. The parameters used for the simulation of this dataset can be found in table 5. Each segment has its own variance and a model type mean or a line. Here the blue dotted lines represent unique change-points while the red dotted lines represent common change-points. As we see from the figure the change-points are rather clear but it can be harder to detect changes when segments are allowed to have different model types.

is used.

Since ASM is computationally heavier and does not perform better than the other methods, we exclude it from further study.

5.2.2 Simulation 2

We compare SeqM1 and SeqM2 on a simulated dataset which contains both long and short segments (SIM2). The results from the segmentation of datasets are presented in figure 11.

We notice from 11a that there is no clear difference in TPR_a between SeqM1 and SeqM2. However, we learn from 11b that SeqM1 outperforms SeqM2 for finding the unique change-points. Figure 11c shows us that there is not much difference between the methods in terms of finding the common change-points. However, if we use a more loose definition of true common change-point SeqM2 outperforms SeqM1, see figure 11d. This is consistent with the findings in figure 10.

5.2.3 Simulation 3

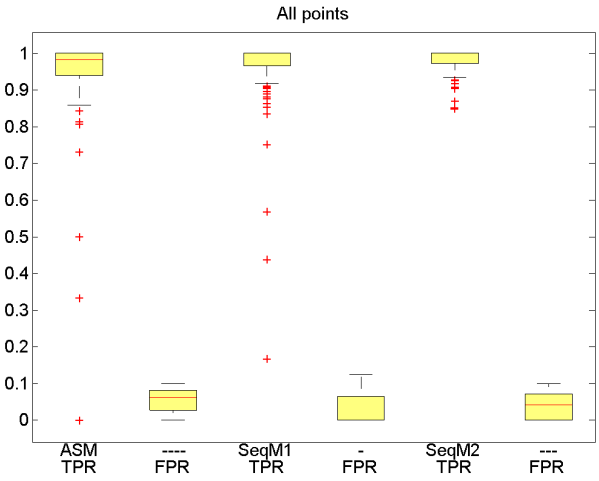
Figure 12 shows the results of SeqM1 and SeqM2 performance on SIM3. Figure 12a displays how SeqM1 outperforms SeqM2 overall. Again, we see from 12b that SeqM1 is better at detecting unique change-points while we see from 12c that SeqM2 no longer has a clear advantage for finding the common change-points.

5.2.4 Overall findings

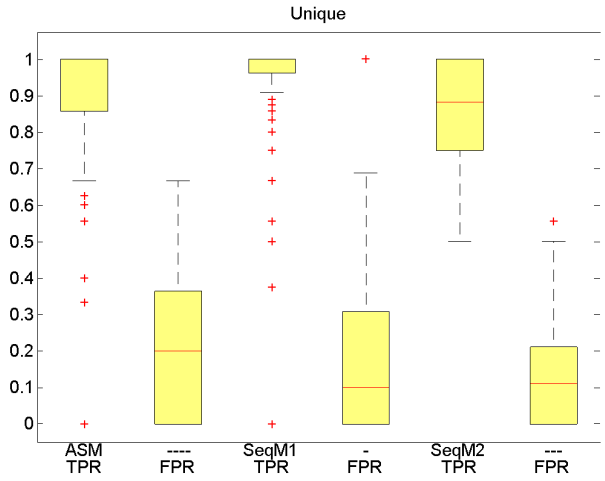
From figures 10, 11 and 12 we learn that SeqM1 and SeqM2 are far better than ASM. ASM is limited to datasets containing few long segments due to its computational burden and does not have any capabilities that exceed those of SeqM1 and SeqM2.

SeqM1 and SeqM2 have different properties. SeqM2 favors common change-points at the expense of finding unique change-points and it tends to find more common change-points than there are. SeqM1 does not perform as well as SeqM2 for finding the common change-points but does better overall.

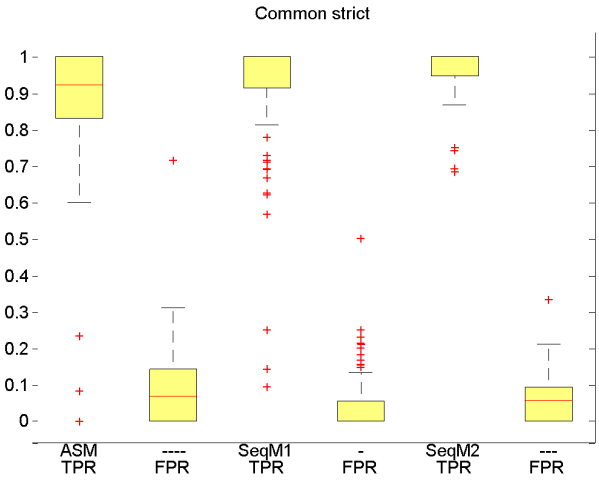
SIM3 has a more complicated structure, since we allow for different models between segments. Segmenting SIM3 is a more difficult task and therefore we see a reduction in performance when we compare figure 12 to figures 10 and 11.



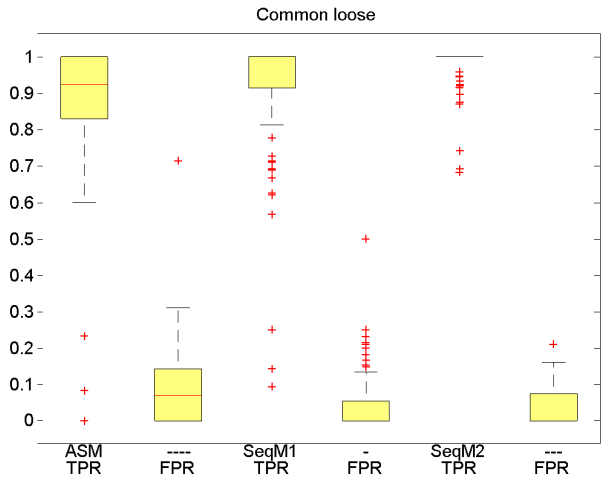
(a) All change-points



(b) Unique change-point

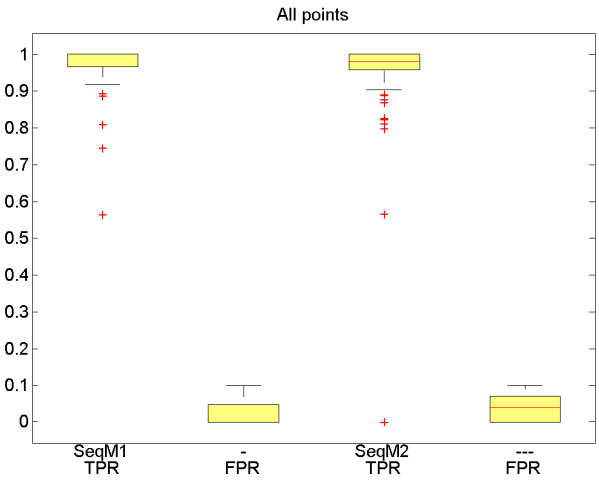


(c) Common change-points strict def.

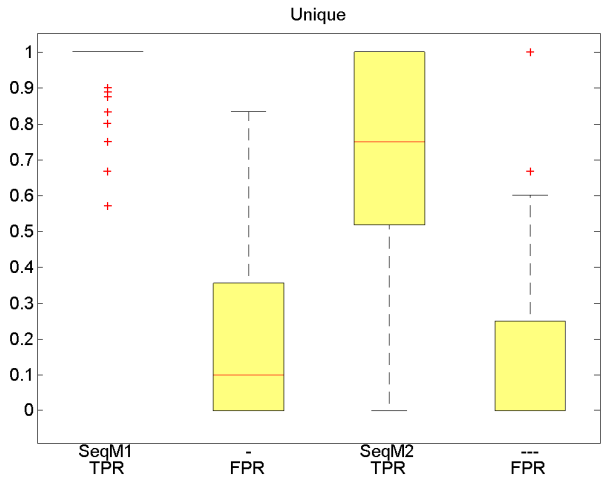


(d) Common change-points loose def.

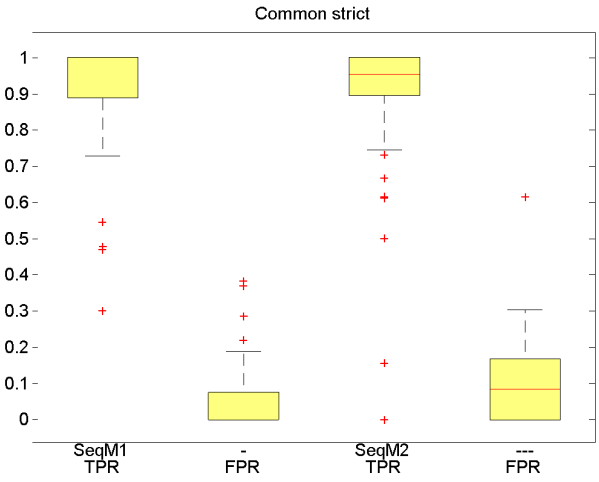
Figure 10: SIM 1: Box plots for the true and false positive rates for all the three methods. Figure a) contains the TPR_a while FPR_a is fixed on the interval $[0, 0.1]$. By using the result that keep FPR_a on this interval we calculate TPR and FPR for common and unique points. Definitions of TPR and FPR can be found in section 4.6. We see from the figures above that ASM is also worse than SeqM1 and SeqM2 in every aspect. SeqM2 seems to be doing slightly better in finding the overall points, TPR_a. It is also interesting to compare SeqM1 and SeqM2 when looking at the common and unique change-points. SeqM1 seems to do a better job in finding the unique change-points while SeqM2 outperforms the other methods in finding the common change-points.



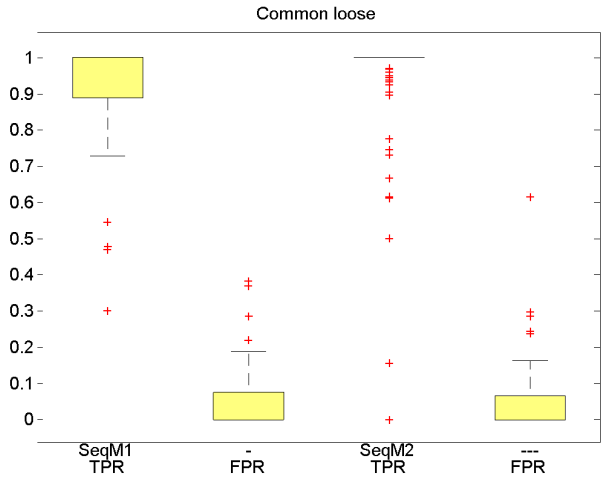
(a) All change-points



(b) Unique change-point

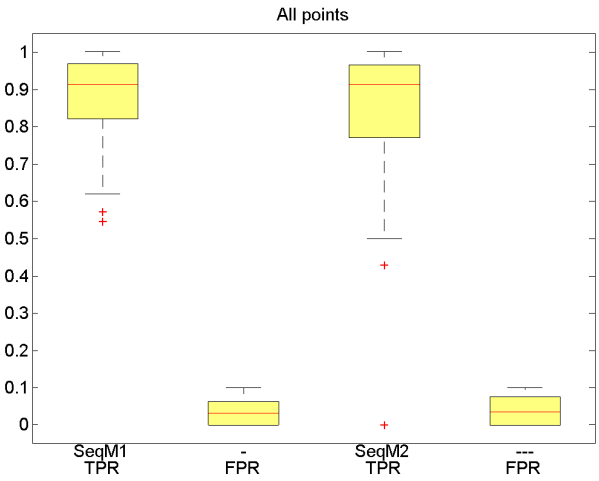


(c) Common change-points strict def.

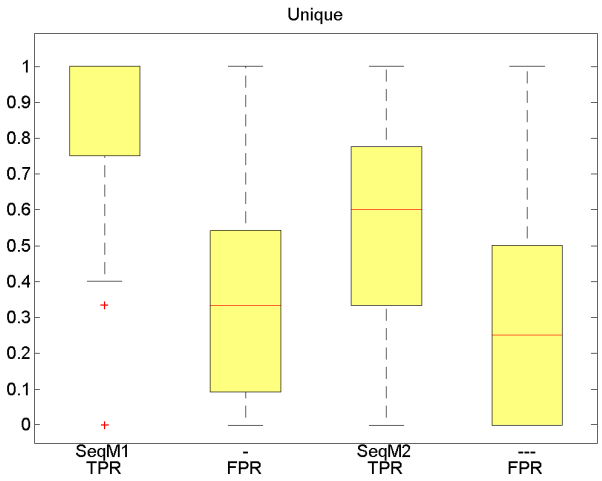


(d) Common change-points loose def.

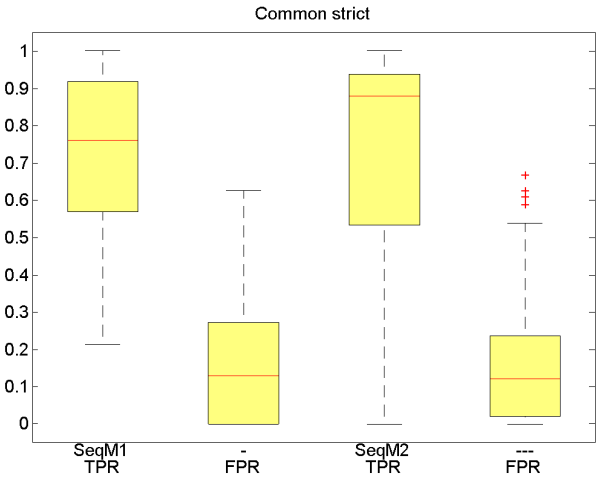
Figure 11: SIM 2: Box plots for the true and false positive rates for SeqM1 and SeqM2. Figure a) contains the TPR_a while FPR_a is fixed on the interval $[0, 0.1]$. We again see how methods SeqM1 and SeqM2 perform at finding the common and unique change-points. This is consistent with the results presented in figure 10.



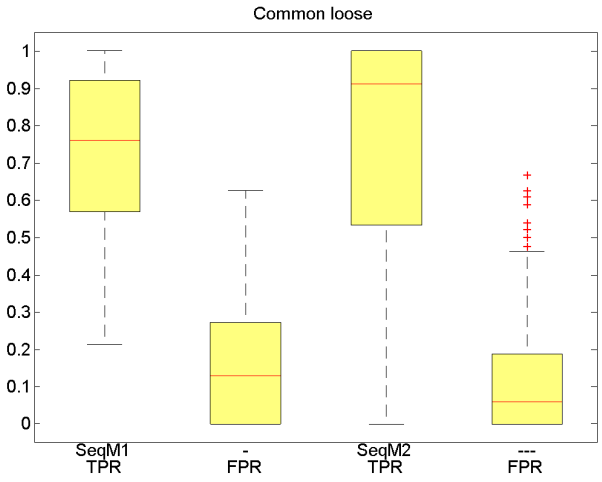
(a) All change-points



(b) Unique change-point



(c) Common change-points strict def.



(d) Common change-points loose def.

Figure 12: SIM 3: Box plots for the true and false positive rates for SeqM1 and SeqM2. Figure a) contains the TPR_a while FPR_a is fixed on the interval $[0, 0.1]$. SeqM1 has higher TPR_a. SeqM2 does not seem to have as clear an advantage over SeqM1 for finding common change-points as in figures 10 and 11. SeqM1 outperforms SeqM2 at finding unique change-points.

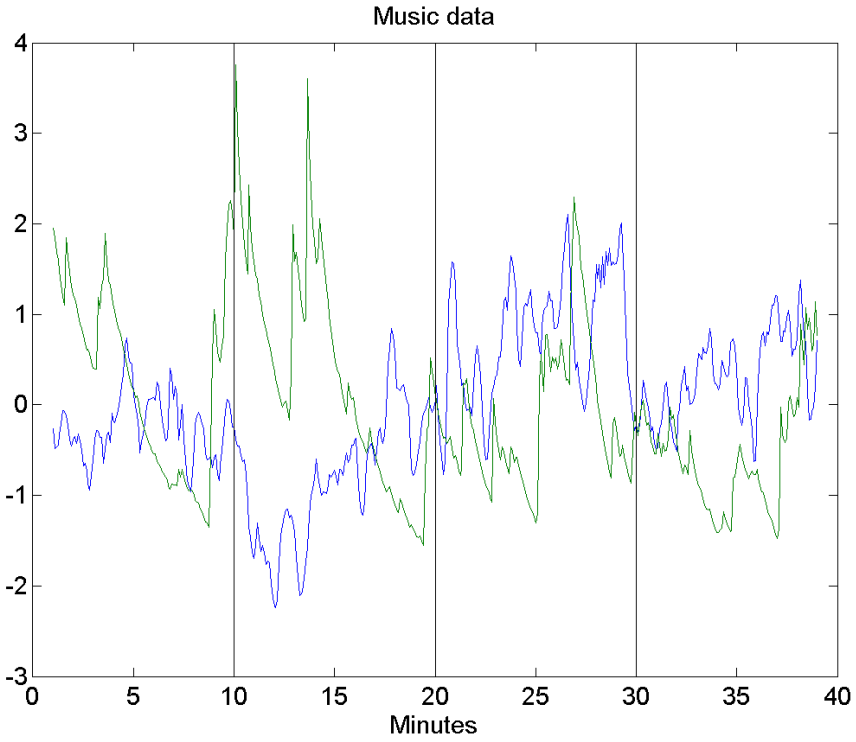


Figure 13: A figure of the music data. Finger temperature is noted by the blue color while the skin conductance is noted by the green color. The vertical lines denote the time instances when the music changes.

5.3 Music data

Figure 13 shows the music data. Finger temperature is noted by the blue color while green is the skin conductance. At 10, 20 and 30 minutes, the music changes. The first and last time periods consist of silence while in the second period horror music is played and peaceful music in the third period. The aim is to see whether we can detect physiologically where the music changes. It is also of interest to see whether the features change segments coordinated or with a delay [1] (a preprint will appear on the website shortly).

We fit multiple models to the music data. We start out by looking at how the mean and line models perform. We then look at autoregressive models, and finally polynomial segment models. For each segmentation we examine three plots. The first one contains the plotted result of the segmentation. The second plot contains the plotted elbow, that is the normalized sequence of the part of the log likelihood and third plot will display the difference between each consecutive derivative of the elbow curve. Further description

of how these plot are obtained can be found in section 3.5. The complete list of models is given in the appendix A.2.

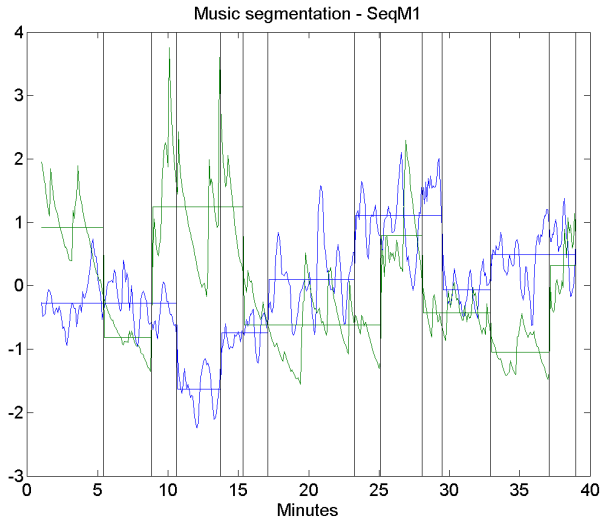
5.3.1 A mean and line

Figure 14a shows the segmentation using SeqM1. We use a mean and variance to model each segment. Figures 14b and 14c suggest an appropriate number of segments has been chosen. Figure 15 shows the same using SeqM2. We now see a clear difference between these methods. SeqM2 does not perform as well for unique change-points. From both methods we note that a specific mean and variance for each segment is a too simple model to describe this dataset.

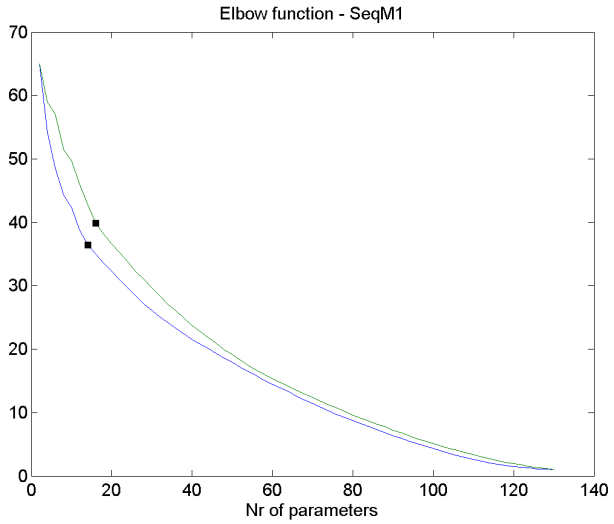
Figure 16 and 17 show the data segmented by SeqM1 and SeqM2, respectively, when we allow the segments to switch between a mean and a line. The segments catch the trends in the data. We note that, for SeqM1 in figure 16a, the change-points are not coordinated over features. We see that the finger temperature changes after the skin conductance, see further [1] (a preprint will appear on the website shortly). This Figure 17a, however, has only coordinated change-points. SeqM2 restricts every change-point of all features to have a minimum length between change-points. Therefore, it misses the delay between the features which seems to appear in figure 16. A mean and line does appear to be useful for modeling the data since it manages to capture changes in the trend. SeqM1 especially detects changes in finger temperature shortly after the music changes at minute 10, see figure 16.

5.3.2 Autoregressive model

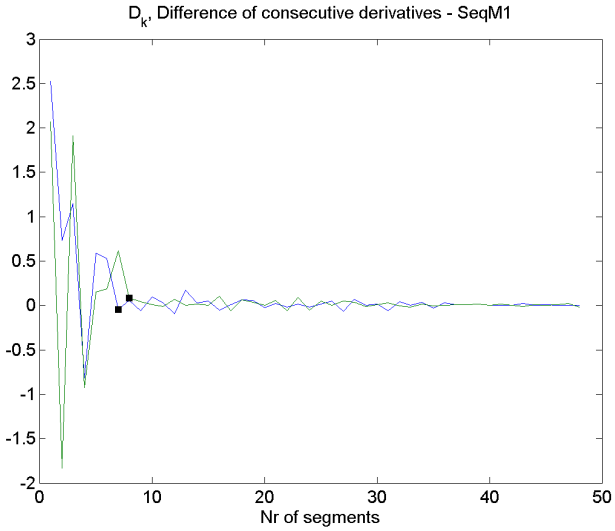
Multiple autoregressive models are fitted to the data. Only an autoregressive model with a trend is discussed here. Additional autoregressive models can be found in the appendix A.2. Figure 18 and 19 show the results for SeqM1 and SeqM2. We see that neither methods identify change-points. We notice that both an autoregressive model with a trend and a model that switches between a mean and a line are capable of describing the data well. The autoregressive model is a rich and more complex model and catches the quickly changing trends of the data. However, a model that switches between a mean and a line detects the change-points where the trend changes. We know that the music changes after each 10 minutes. It can therefore be justifiable to use a model with a mean and line if the aim is to interpret where those change-points are located.



(a) Segmentation

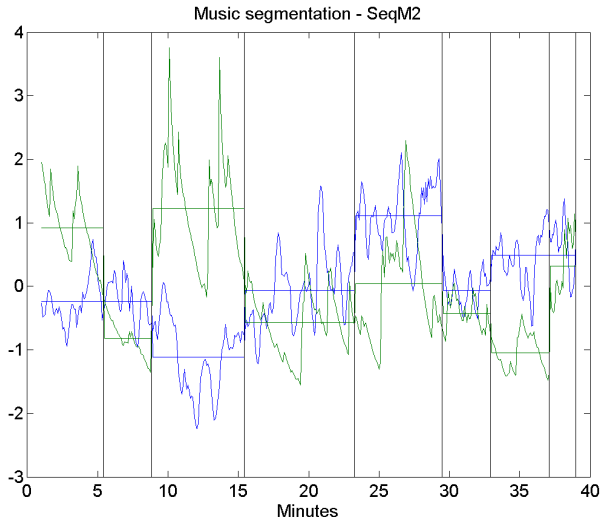


(b) Elbow function

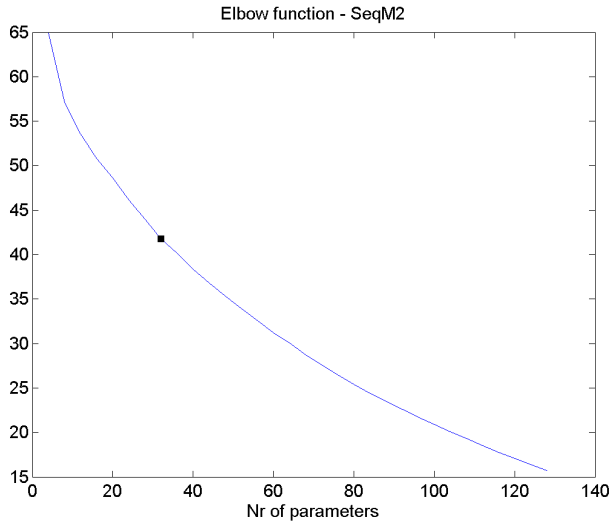


(c) Slope changes

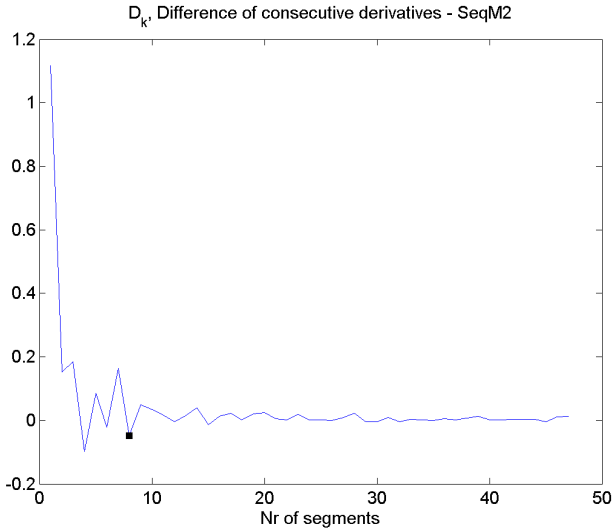
Figure 14: We segment finger temperature and skin conductance with SeqM1. Each segment has its own mean and variance. Figure 14a shows us that this model is too simple for these features.



(a) Segmentation

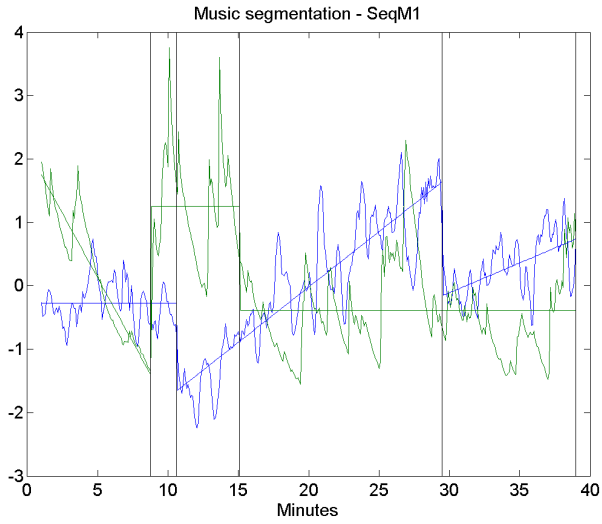


(b) Elbow function

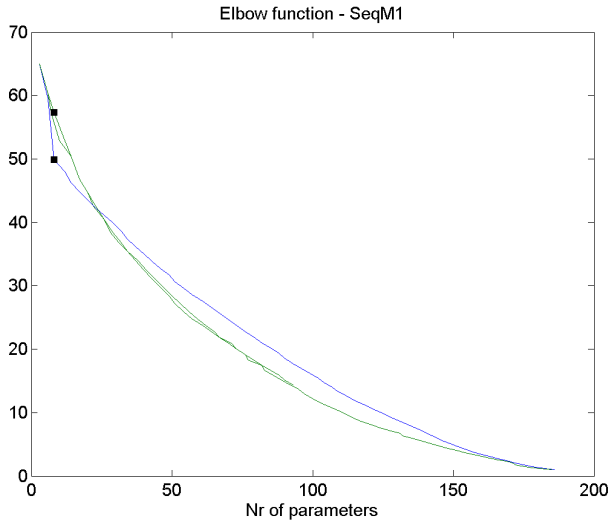


(c) Slope changes

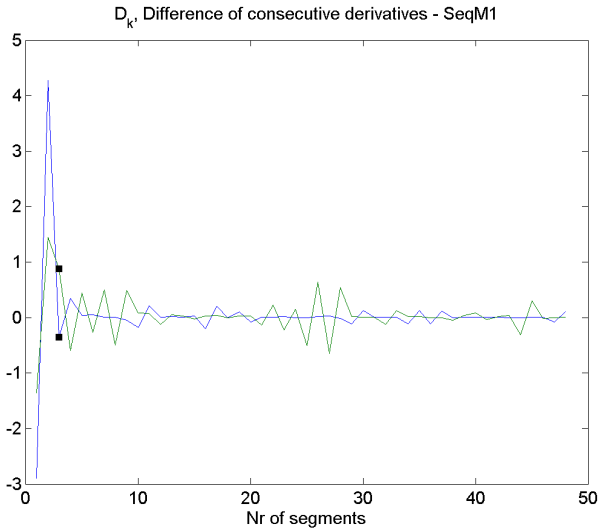
Figure 15: We segment finger temperature and skin conductance with SeqM2. Each segment has its own mean and variance. Figure 14a shows us that this model is too simple for these features. From figure 15a we see that SeqM2 favors common change-points.



(a) Segmentation

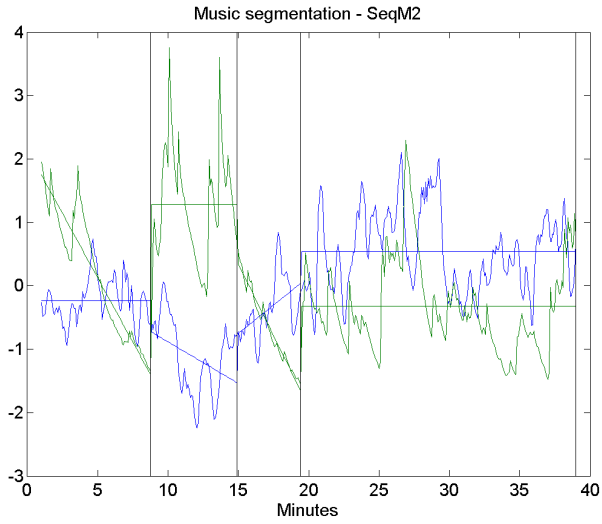


(b) Elbow function

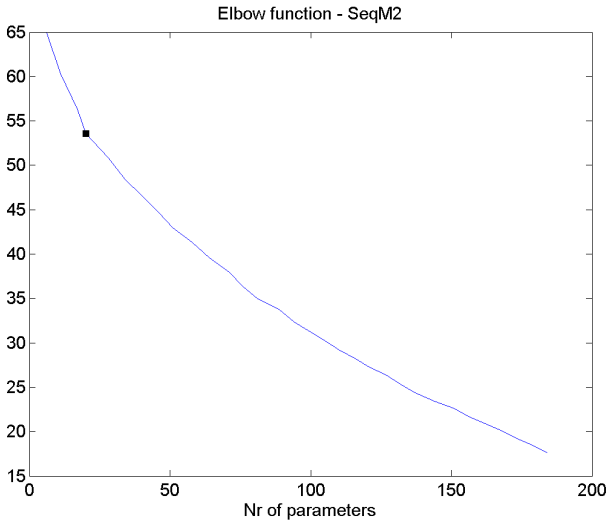


(c) Slope changes

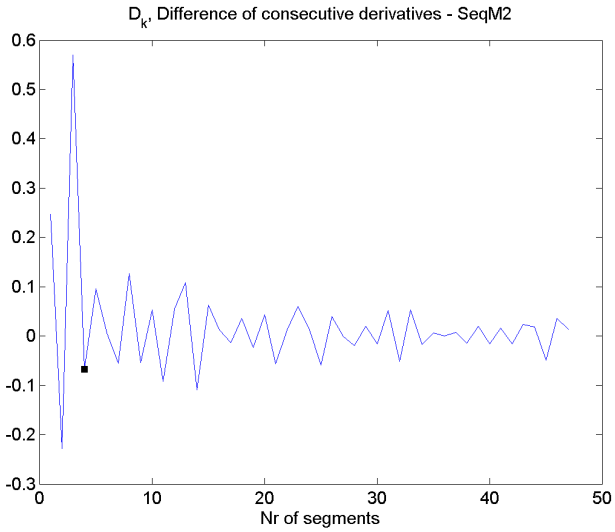
Figure 16: We segment the music data with SeqM1 where each segment can switch between a mean and a line. We use figures 16b and 16c to decide the number of segments. From figure 16a we see that the model captures the trend changes. We do not see change-points at the exact time points where the music is changed. However, we see a trend in the finger temperature shortly after the music is changed at minute 10.



(a) Segmentation

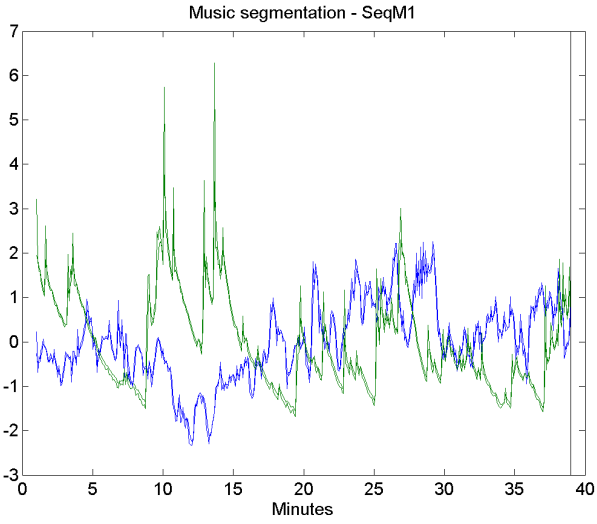


(b) Elbow function

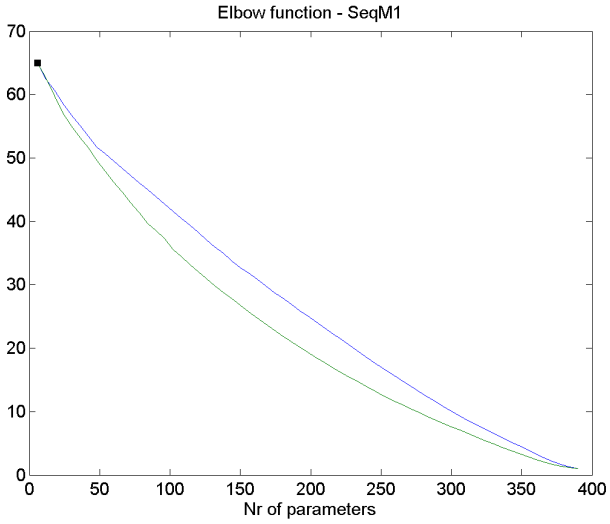


(c) Slope changes

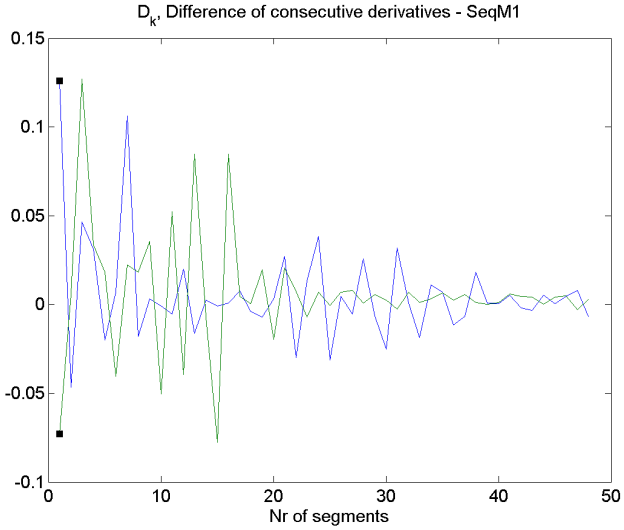
Figure 17: We segment the music data using SeqM2. The segments can switch between a mean and a line and we use 17b and 17c to choose the number of segments. In contrast with figure 16a we see from 17a that each change-point is common. SeqM2 favors common change-points. Due to that, we loose the information that the finger temperature seems to change shortly after the skin conductance around minute 10, see figure 16a.



(a) Segmentation

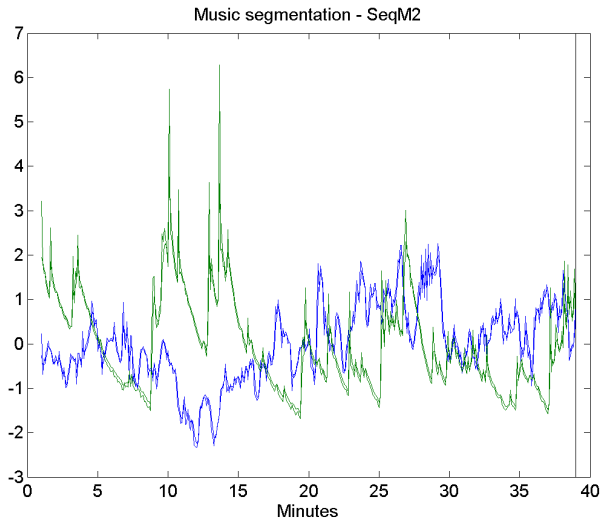


(b) Elbow function

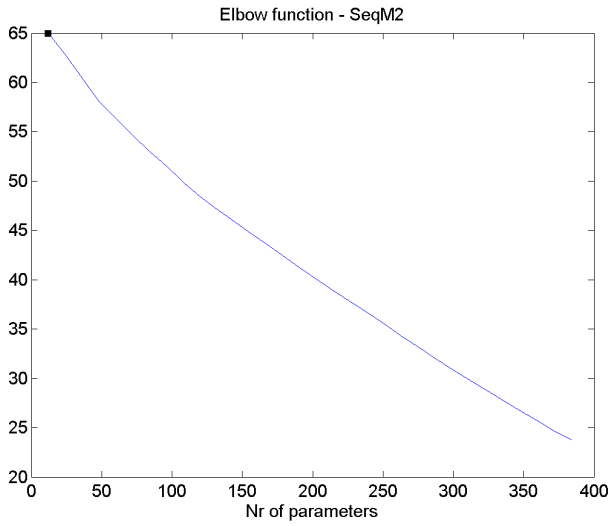


(c) Slope changes

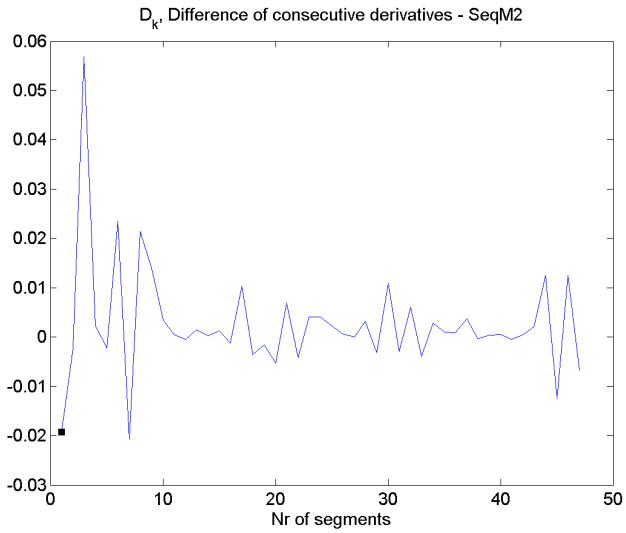
Figure 18: Here SeqM1 is used to segment the data. We use an autoregressive model with a trend, which is a richer and a more complex model than a mean and a line. The autoregressive model, models the data without detecting change-points. For interpretation purpose, the model is too rich since we cannot deduce anything about the music changes.



(a) Segmentation



(b) Elbow function



(c) Slope changes

Figure 19: Here SeqM2 is used to segment the data. We use an autoregressive model with a trend, which is a richer and a more complex model than a mean and a line. The autoregressive model, models the data without detecting change-points. For interpretation purpose, the model is too rich since we cannot derive anything about the music changes.

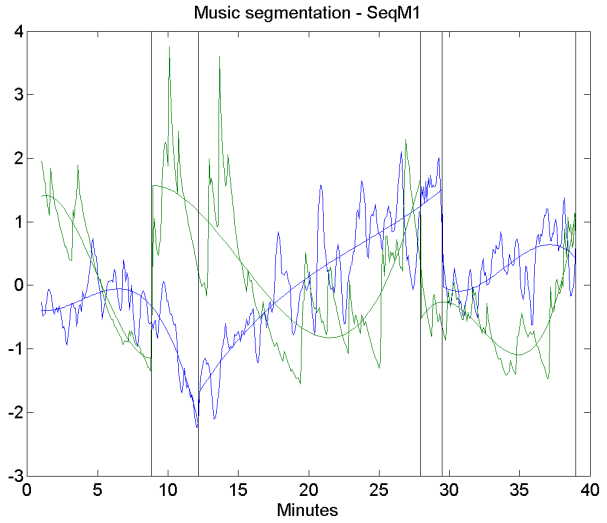
5.3.3 Polynomials of order 3 and 5

Polynomials of orders 3 and 5 are fitted to the music data using methods SeqM1 and SeqM2. Results from fitting a polynomial of order 4 are in appendix A.2. Figures 20 and 21 show the segmented features using SeqM1 and SeqM2 when we fit with a polynomial of order 3. We see from figures 20d and 21d that we can justify choosing multiple different values for the number of segments. The black boxes indicate the number of segments when we are rather conservative, i.e., we choose the number of segments where the difference in consecutive derivatives has become significantly lower but has not leveled off yet. The red boxes show how many segments we end up with if we choose to let the consecutive derivatives level off. In figure 20a there are change-points around minute 10 and 30 which are not coordinated. Figure 20b is not easy to interpret. We see that for the finger temperature (blue), we choose a reasonable amount of segments while for the skin conductance we may need another model type if we insist on the consecutive derivatives to level off. In figures 21a and 21b we see the same results but now using SeqM2. For SeqM2 the skin conductance dominates where the change-points are located.

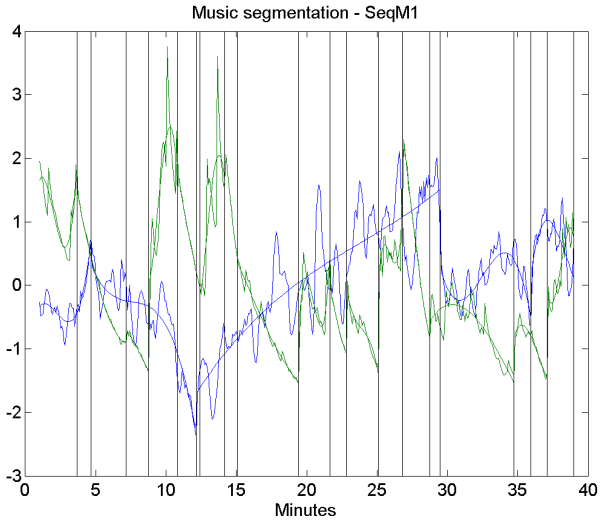
Figures 22 and 23 shows the segmented data where each segment is fitted with a polynomial of order five. We note from figures 22b, 22c, 23b and 23c that there is not a clear change-point located in the data. This is different from figures 20 and 21. A polynomial of order 5 is rich enough to capture all the trends in the model, so no change-points are needed. It is then a matter of interpretation which of the models are more appropriate. A model that assumes that the data is consisting of segments that change characteristics, where each segment has a simpler model or fitting it right away with a more complex model.

5.3.4 Simulated data on autoregressive and polynomial models

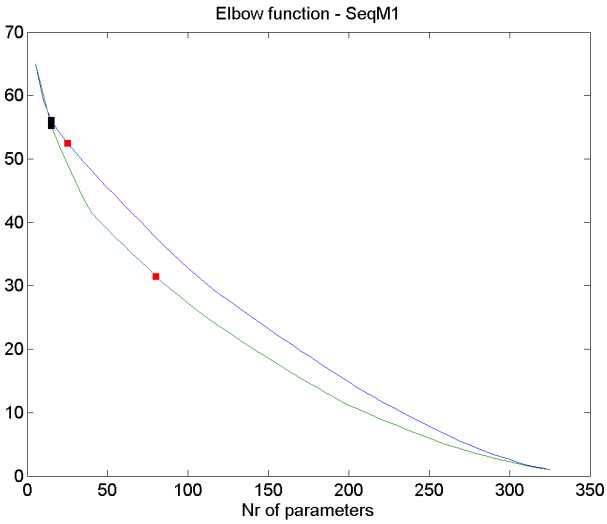
We perform a small simulation study to demonstrate the performance of autoregressive and polynomial fitting in a more controlled setting. The previous simulation studies focused on a mean and line models. Table 6 shows the parameters used for the simulation of the two datasets. We want to take a snapshot to see how the methods perform when the datasets have complicated structure. The maximum variance of each segment is allowed to be high for each segment. This makes the change-points harder to detect. Figure 24 displays how the number of segments is chosen for AR(3)+line dataset for both method SeqM1 and SeqM2, see section 3.5 for information about the plots. Figure 25 displays the segmented data. Table 7 shows the true



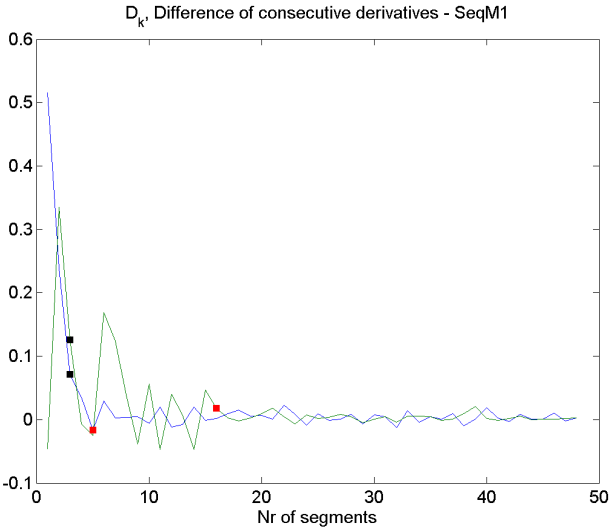
(a) Segmentation



(b) Segmentation

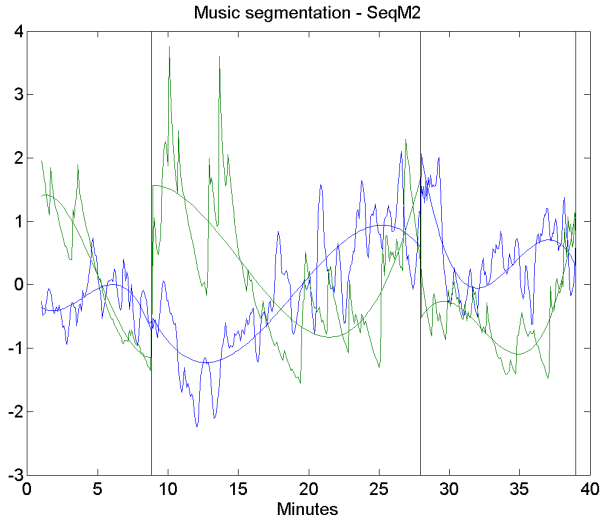


(c) Elbow function

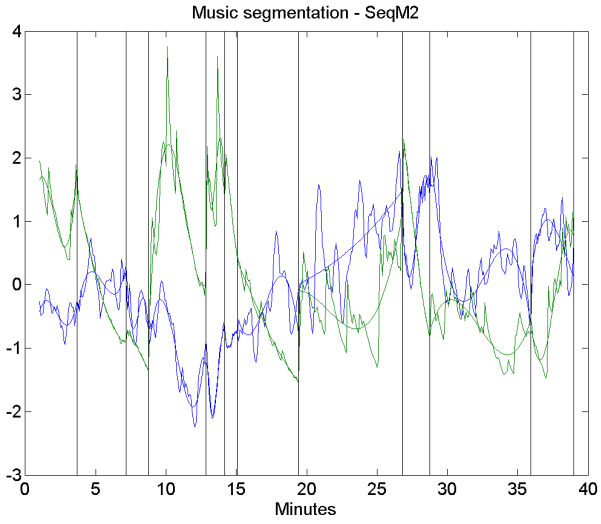


(d) Slope changes

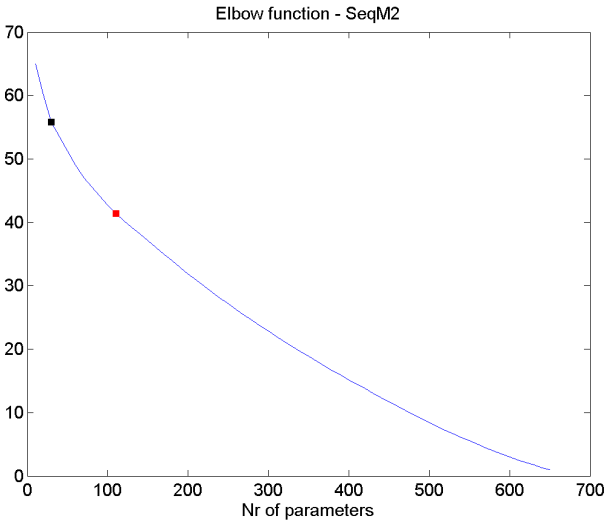
Figure 20: Here SeqM1 is used to segment the data. The models that are used are polynomials of order 3. We can reason for choosing different number of segments. The black boxes correspond to figure 20a and the red boxes correspond to 20b. In figure 20a we detect change-points around minute 10 and 30 and the polynomials follow the trend well. Figure 20b is not as easily interpreted. The finger temperature has a reasonable number of segments while the skin conductance may need another model type if we insist on the consecutive derivatives to level off.



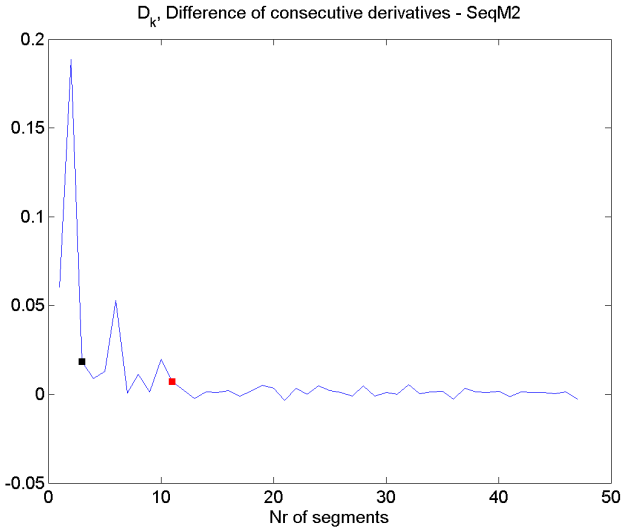
(a) Segmentation



(b) Segmentation

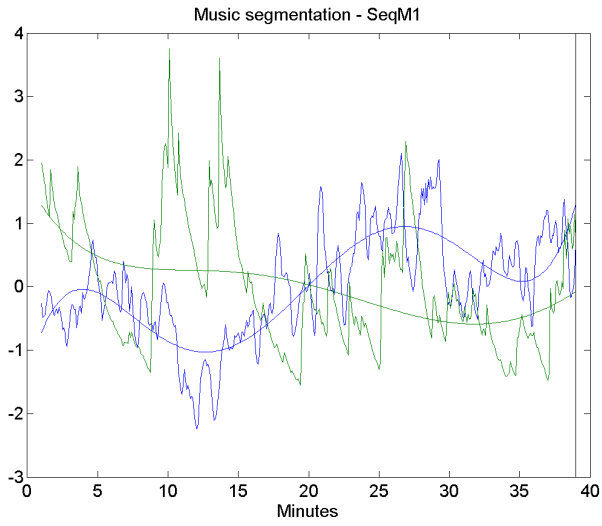


(c) Elbow function

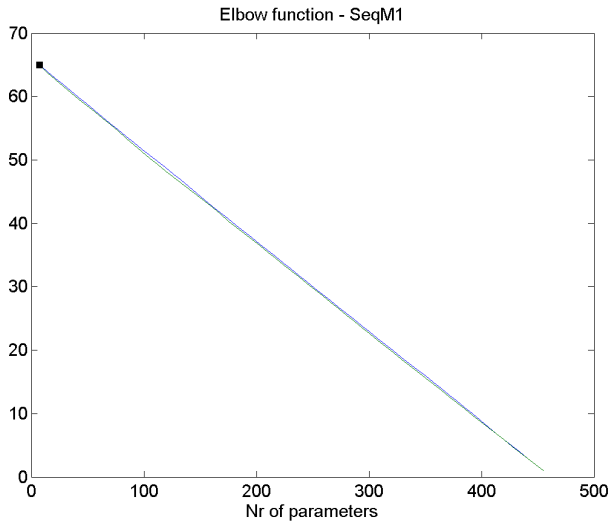


(d) Slope changes

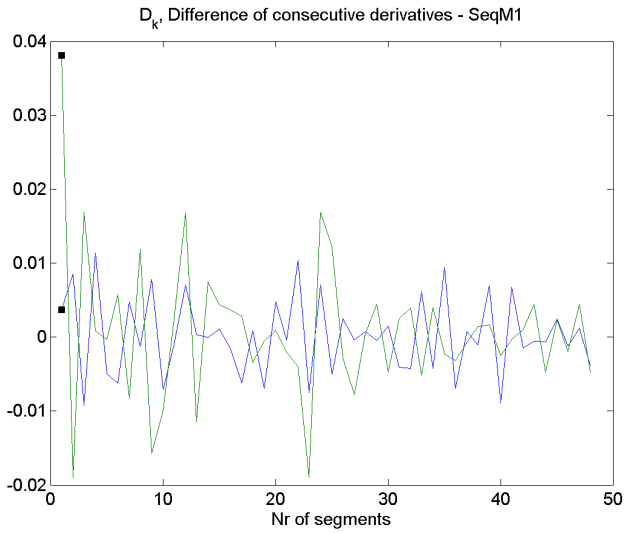
Figure 21: Here SeqM2 is used to segment the data. The models that are used are polynomials of order 3. We can reason for choosing different number of segments. The black boxes correspond to figure 21a and the red boxes correspond to 21b. In figure 21a we detect change-points around minute 10 and 30. From both figure 21a and 21b we see that the skin conductance dominates the finger temperature. SeqM1 is therefore more appropriate if we do not want to miss that information.



(a) Segmentation

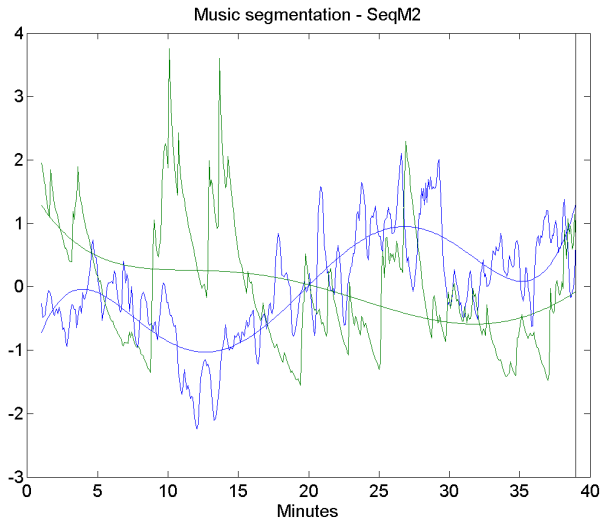


(b) Elbow function

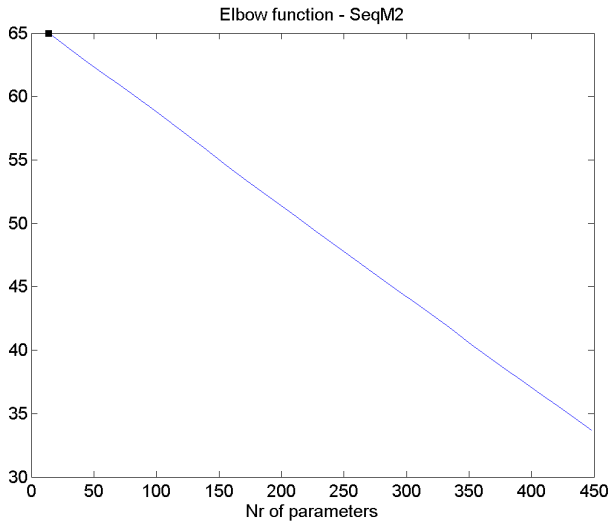


(c) Slope changes

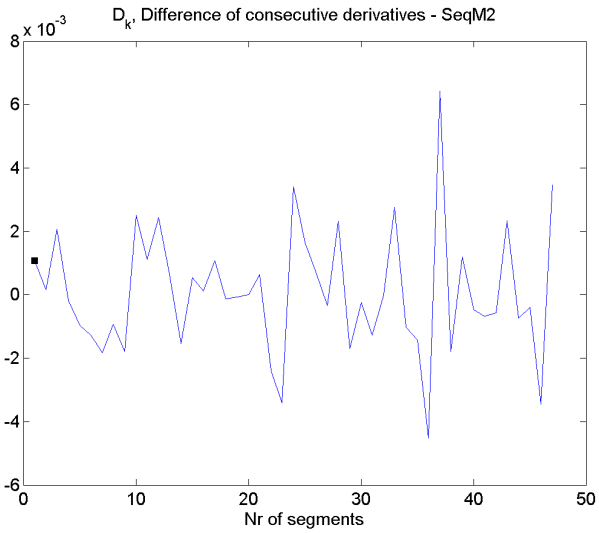
Figure 22: The segmentation with SeqM1 when using a polynomial of order 5. Polynomial of order 5 can be used to model the data without a change-point. It depends on the assumption of the data whether a more complicated model should be used or if it is justifiable to model it with independent segments that change characteristics.



(a) Segmentation



(b) Elbow function



(c) Slope changes

Figure 23: The segmentation with SeqM2 when using a polynomial of order 5. Polynomial of order 5 can be used to model the data without a change-point. It depends on the assumption of the data whether a more complicated model should be used or if it is justifiable to model it with independent segments that change characteristics.

	Autoregressive	Polynomial
Model types	AR(3)+$\alpha + \beta x$	$\sum_{i=0}^{3} p_i x^i$
Model parameters	$\phi_1 = [-1.17, 1.17]$, $\phi_2 = 0.16$, $\phi_3 = 0.0016$	$p_1 = 10^{-6} \cdot [-1, 1]$, $p_2 = 10^{-4} \cdot [-1, 1]$, $p_3 = 10^{-2} \cdot [-1, 1]$, $p_4 = [-2, 2]$
Nr of steps	10	10
Min length of segment	28	63
Max length of segment	62	18
Nr of features	3	3
Variance of noise	2.33	1.17
Variance of step height	2	1

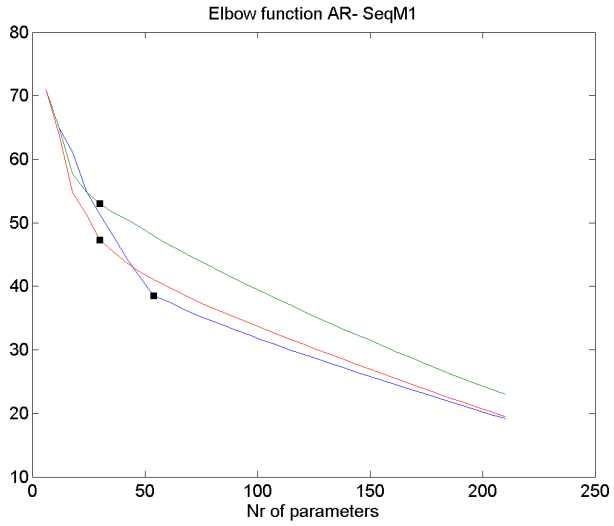
Table 6: Parameters used for simulating the datasets.

	SeqM1 tpr	SeqM1 fpr	SeqM2 tpr	SeqM2 frp
All points	0.7895	0.0625	0.9474	0.1000
Common loose def	0.6000	0.1818	1.0000	0.1111
Common strict	0.6000	0.1818	0.8667	0.2778
Unique	0.7500	0.4000	0.5000	0
Overall unique and common strict	0.6316	0.2500	0.7895	0.2500
Overall unique and common loose	0.6316	0.2500	0.8947	0.1000

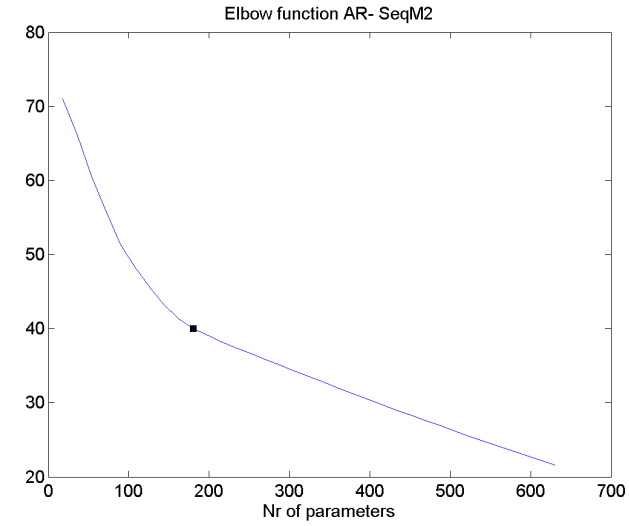
Table 7: The table shows the result for the AR(3)+line simulation. The plotted results are displayed in figure 25. Taking into account how complicated this dataset is, both methods perform well. SeqM2 performs overall better except for finding the unique change-points, there SeqM1 has an edge on SeqM2. The false and positive ratios in the last two rows give indication on the overall ability to find common and unique change-points, see equation (4).

and false positive rates. SeqM2 performs better except for finding unique change-points.

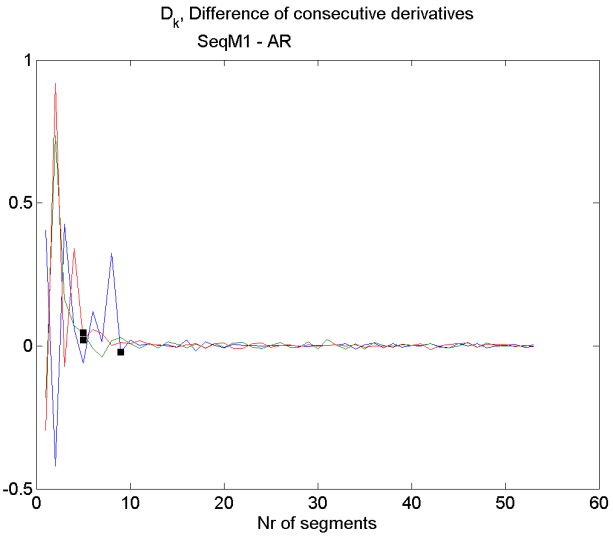
Figure 26 shows how we choose the number of segments. From the elbow plots we detect clearly a point where adding more segments does not result in a significantly lower log likelihood. The plotted results from that segmentation are displayed in figure 27. Table 8 contains the true and false positive rates. We see that SeqM2 again has an edge on SeqM1, when finding common change-points. When finding unique change-points, SeqM2 is conservative while SeqM1 has a high false positive rate. The reason for this high false positive rate is that it tends to find a unique change-point where there truly is a common-change point. That results in a low true positive rate for the common change-points and high false positive rate for the unique change-points. We keep in mind that the dataset has a complex structure,



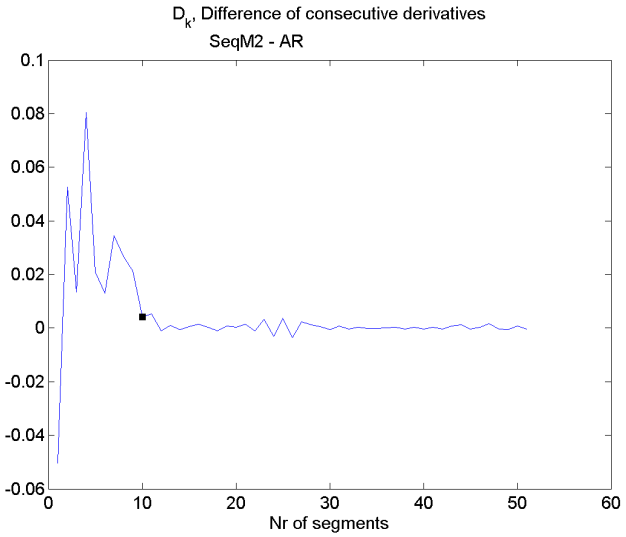
(a) SeqM1-Elbow



(b) SeqM2-Elbow

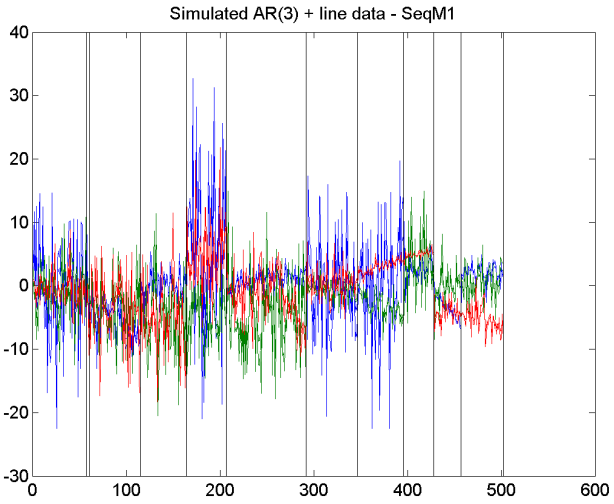


(c) SeqM1-Slope changes

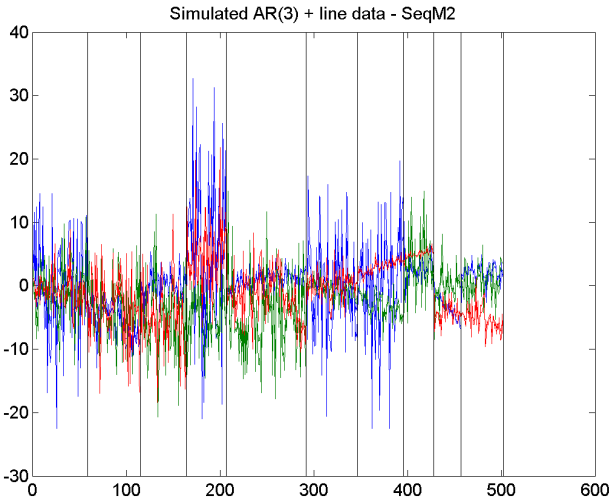


(d) SeqM2-Slope changes

Figure 24: The figures show how the number of segments are chosen for the autoregressive dataset. More information about these plots can be found in section 3.5. Here we see from both figures that there is a value where the log likelihood decreases more slowly.



(a) SeqM1



(b) SeqM2

Figure 25: The figures show the segmentation of a simulated dataset. The simulated dataset is built up from segments with different kinds of AR(3)+line structures. Figures 25a and 25b show how SeqM1 and SeqM2 segment the data respectively. True and false positive rates in table 7 give an indication of how the methods perform.

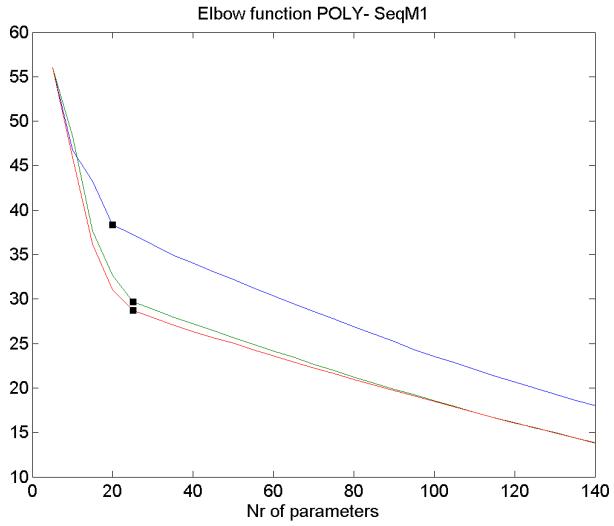
and the performance is expected to reduce according to that. Further work is needed to select the adjusting parameter for these methods.

5.4 Epilepsy data

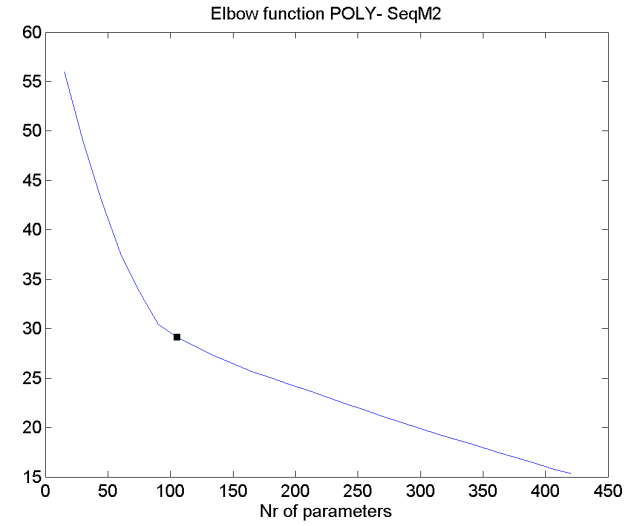
We segment three different seizures from the same patient. We keep 60 seconds around the seizures to obtain a baseline. Every seizure therefore starts at second 61 and ends 60 seconds before the interval is over. It is important to notice that when we look at parts of the data we can investigate the elbow function and how the consecutive derivatives behave. We can then choose an adjusting parameter according to section 3.5. However, we are not able to do that when segmenting the whole data on-line. Each segment is fitted using an autoregressive model with a trend.

Figure 28 shows the segmentation of seizure 1 with method SeqM1 for the vector magnitudes. The vector magnitudes capture the overall activity for each sensor. The segments capture the activity changes but not the start of the seizure.

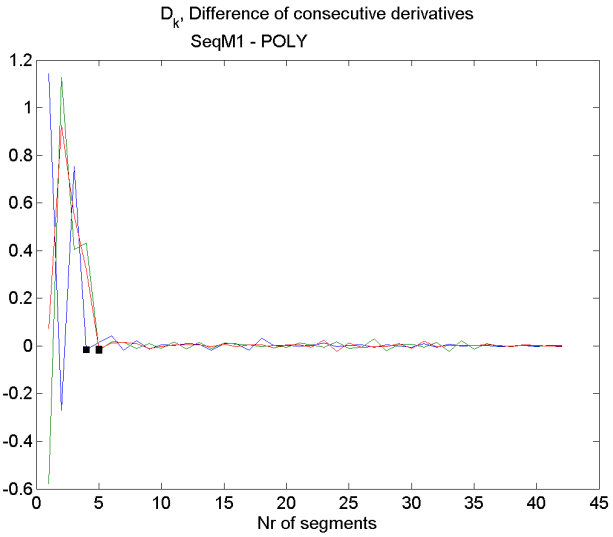
Figure 29 contains same data segmented with SeqM2. There is a clear



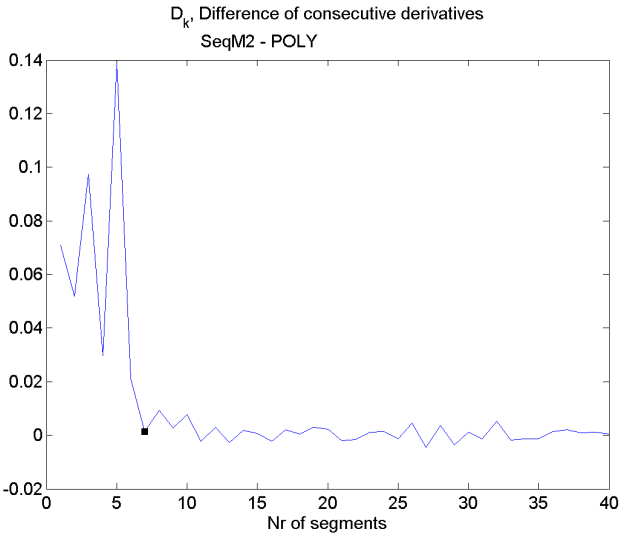
(a) SeqM1-Elbow



(b) SeqM2-Elbow

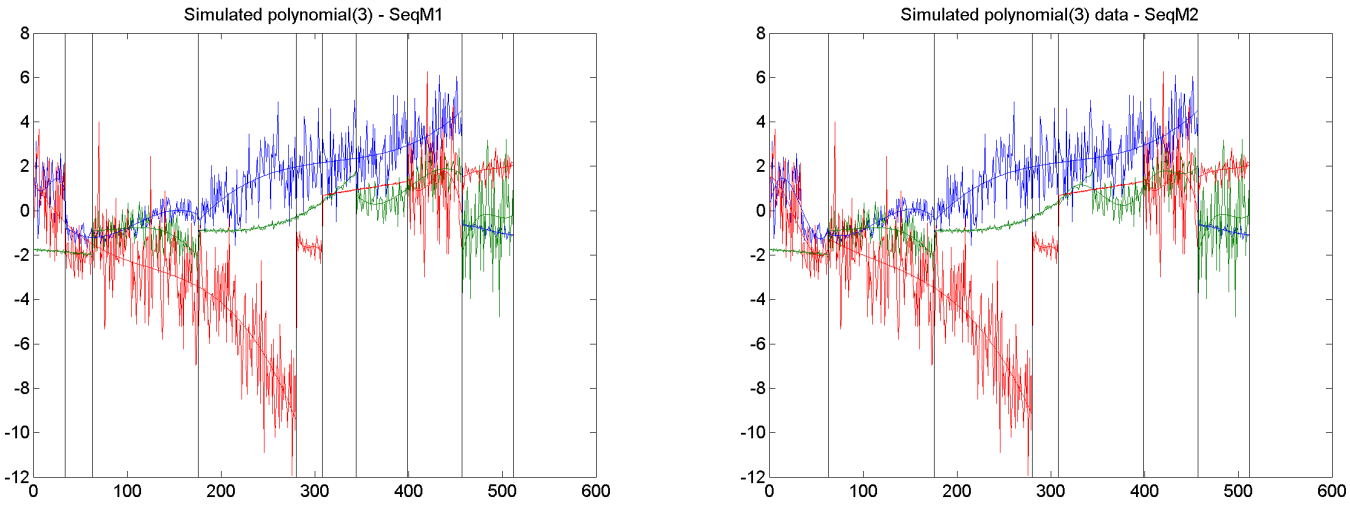


(c) SeqM1-Slope changes



(d) SeqM2-Slope changes

Figure 26: We see how the number of segments are chosen for the polynomial dataset. More information about these plots can be found in section 3.5. Here we see from both figures that there is a value where the log likelihood decreases more slowly.



(a) SeqM1

(b) SeqM2

Figure 27: We see the segmentation results for SeqM1 and SeqM2 of the simulated polynomial dataset. It looks like the segmentation manages to detect the data changes. The true and false positive rates for this segmentation can be found in table 8.

	SeqM1 3 tpr	SeqM1 3 fpr	SeqM2 tpr	SeqM2 frp
All points	0.6471	0	0.7059	0.0769
Common loose def	0.4167	0	0.8333	0.1667
Common strict	0.4167	0	0.8333	0.1667
Unique	0.6000	0.5000	0.2000	0
Overall unique and common strict	0.4706	0.2727	0.6471	0.1538
Overall unique and common loose	0.4706	0.2727	0.6471	0.1538

Table 8: The true and false positive rates for the polynomial simulated data. We see that SeqM2 has an edge on SeqM1, when finding common change-points. When finding unique change-points, SeqM2 is conservative while SeqM1 has a high false positive rate. The reason for the high false positive rate is that it tends to find a unique change-point where there truly is a common change-point. That results in a low true positive rate for the common change-points and high false positive rate for the unique change-points. The false and positive ratios in the last two rows give indication on the overall ability to find common and unique change-points, see equation (4).

difference between the methods. Since it favors common change-points we obtain clear change-points where the overall activity changes. SeqM1 however focuses on each feature separately, we therefore detect more changes-points than when using SeqM1. Neither of the methods detect the start of the seizure. However, they both detect the activity changes.

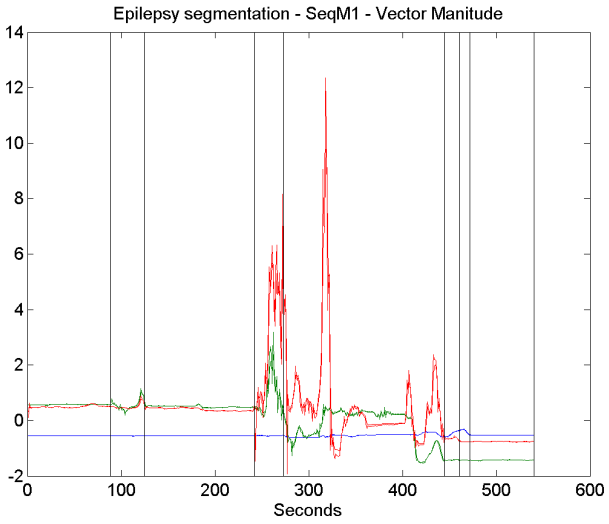
In figure 30, we see the segmentation of the same seizure with method SeqM1 but now we include all the features. We use the same adjusting parameter for all the 51 features since handpicking each of them is time consuming. Some of the features may therefore get over-segmented while other can lack segments. The correlation features have few segments compared to the other features, while the signal magnitude area contains more or up to 8 segments. The frequency bands also contain many segments (up to 9). The other groups of features, see table 1, have a more varying number of segments. From figure 30a we see that the data has many segments. Perhaps it is not suitable to segment all the features at once. It may be more effective to segment the features in subsets. We also notice that there are change-points near to each other that have not been joined which suggest that there is a delay in activity between groups of features.

Figure 31 show the same results for SeqM2. There we have much fewer segments and the plot is similar to the one we got in figure 29. We obtain a clear elbow in figure 31b and the segments do seem to capture activity changes. Comparing figures 30 to 31 we easily detect the different abilities of the methods. SeqM1 focuses on each feature while SeqM2 favors common change-points.

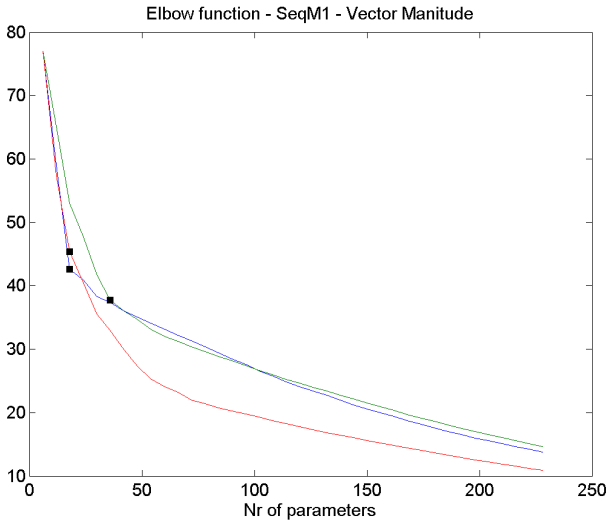
In figure 32, we look at the segmentation of seizure 3 for method SeqM1. This seizure is shorter than the previous one and we do see a lot of segment changes where the seizure starts at second 61. The change-points at seconds 7, 13, 21 and 27 all come from the same feature, which is one of the DC values. This suggest that this particular feature is over-segmented. We see from figure 32b, that there are groups of features with the same number of change-points. 10 of the 12 correlation features have only one segments and 17 of the 18 frequency bands have 2 segments where most of the jumps are around second 60 and few of them at second 38. Other feature groups have a more varying number of segments. The change-points around second 61 are common change-points. SeqM1 does not join these change-points. This suggest that features change in groups at different time-points.

Figure 33 shows the segmentation of the same seizure with method SeqM2. The model catches the seizure start at second 61. However, it does not detect when the seizure ends. Comparing figures 32 and 33 we see how SeqM2 detects fewer change-points.

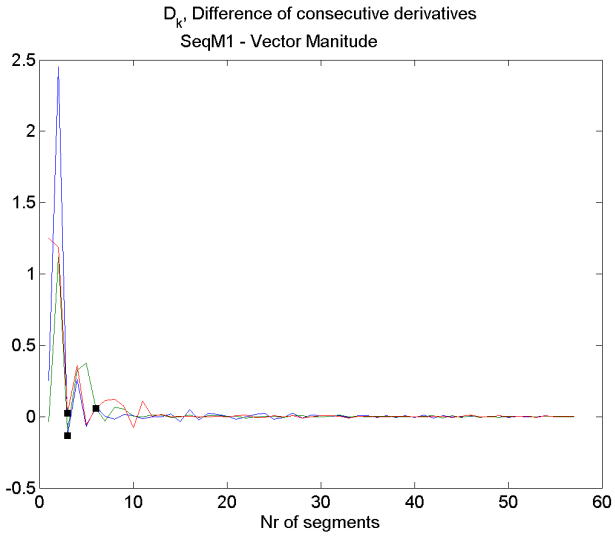
At last, figures 34 and 35 show seizure number 7. Seizure 3 had a clear



(a) Segmentation

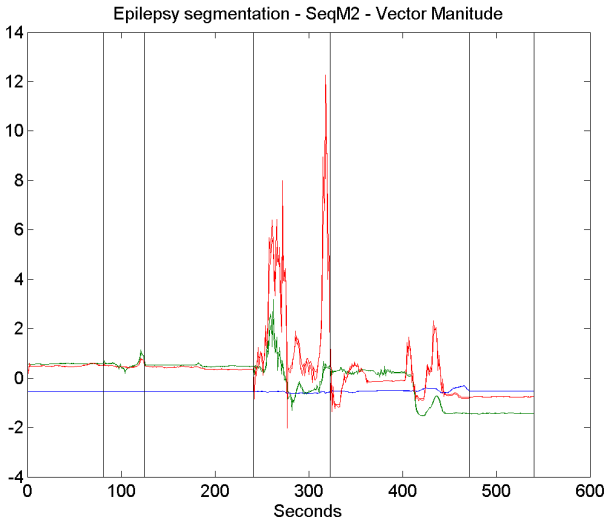


(b) Elbow function

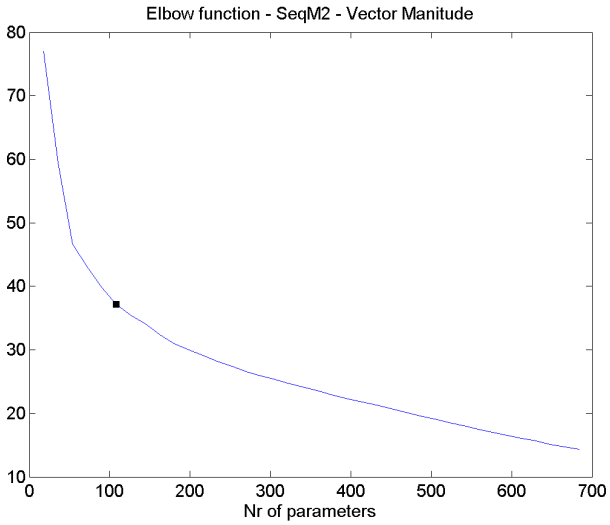


(c) Slope changes

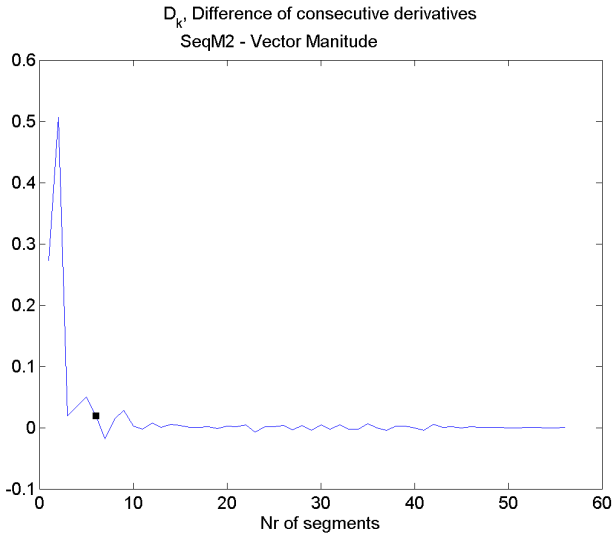
Figure 28: The segmentation of vector magnitudes for seizure number 1 using SeqM1. Each segment is fitted using an autoregressive model with a trend. The segments capture the overall activity of each sensor. We choose the number of segments according to figures 28b and 28c. We do not see a change-point at second 60 where the seizure starts. However, we do see that the segments capture the activity changes of the features.



(a) Segmentation

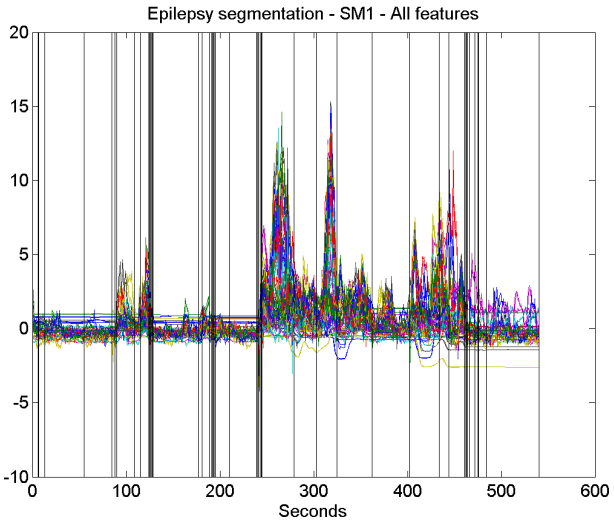


(b) Elbow function

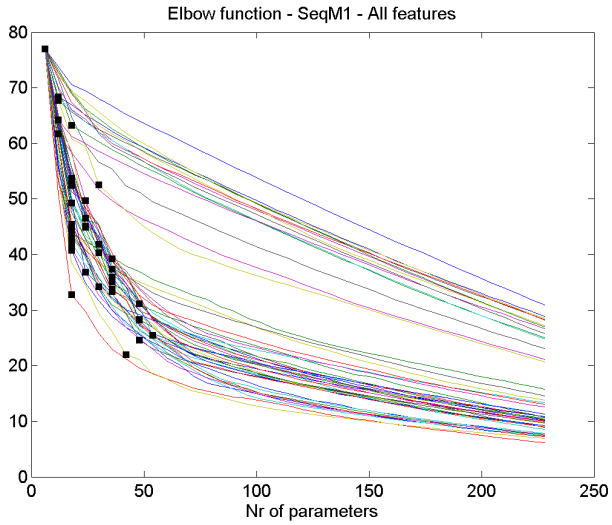


(c) Slope changes

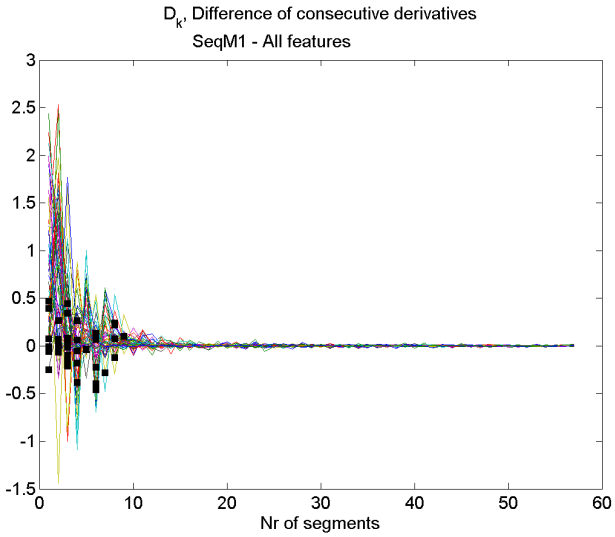
Figure 29: The segmentation of vector magnitudes for seizure number 1 using SeqM2. Each segment is fitted using an autoregressive model with a trend. The segments do capture the activity nicely in separate segments but do not manage to capture the start of the seizure.



(a) Segmentation

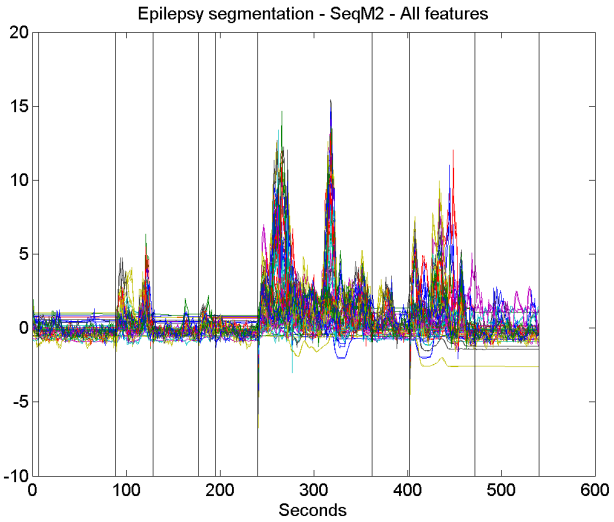


(b) Elbow function

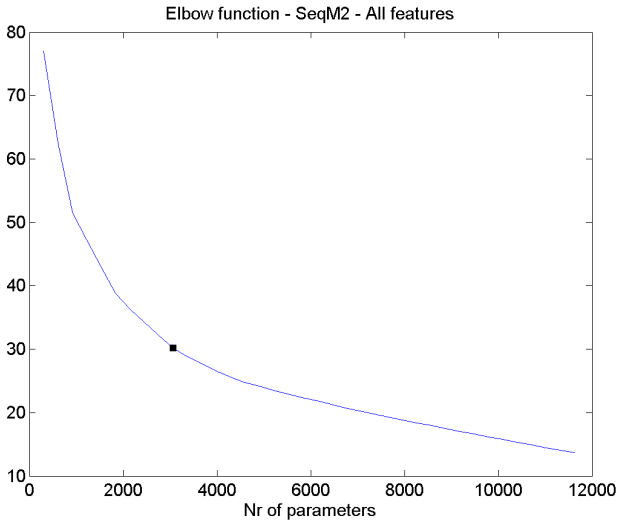


(c) Slope changes

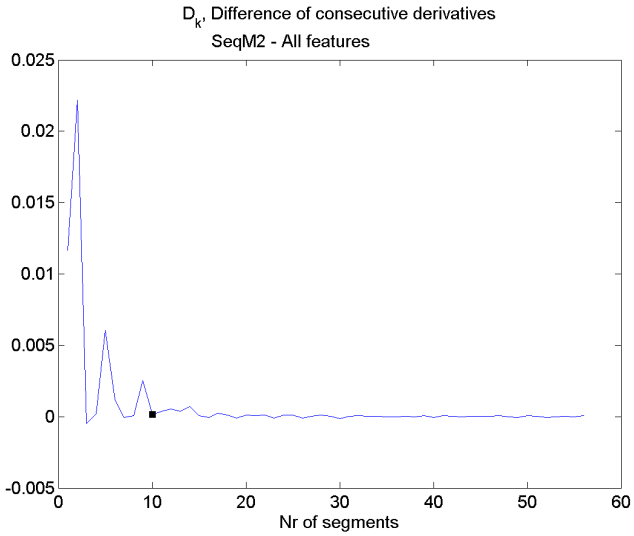
Figure 30: The segmentation of all features of seizure number 1 using SeqM1. Each segment is fitted using an autoregressive model with a trend. The same adjusting parameter is used for all features. This might not be appropriate but is convenient due to the large amount of features we have.



(a) Segmentation

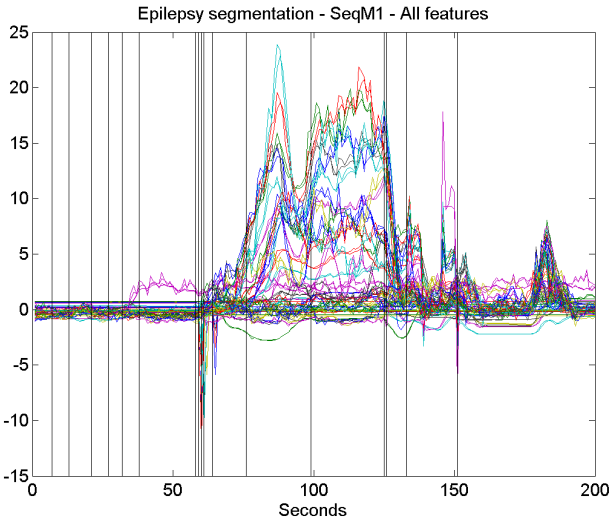


(b) Elbow function

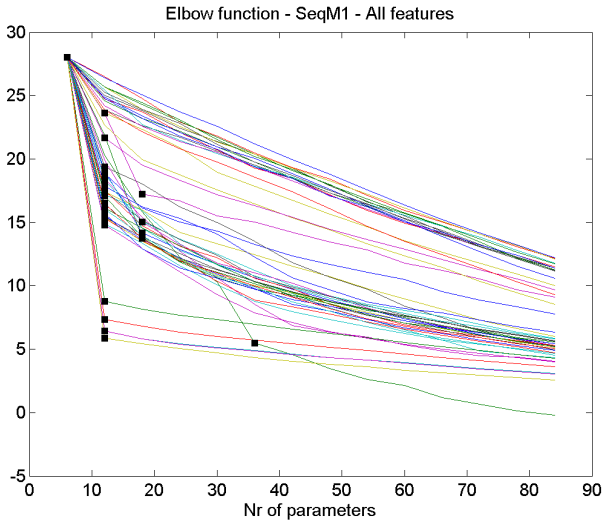


(c) Slope changes

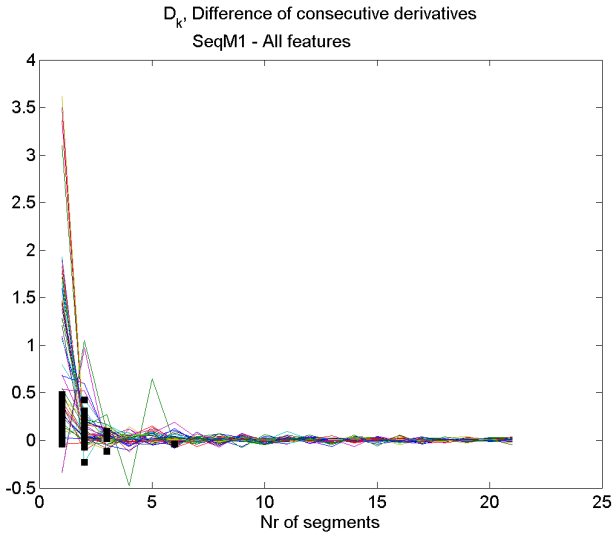
Figure 31: The segmentation of all features of seizure number 1 using SeqM2. Each segment is fitted using an autoregressive model with a trend. We have few segments but they seem to catch the activity changes.



(a) Segmentation

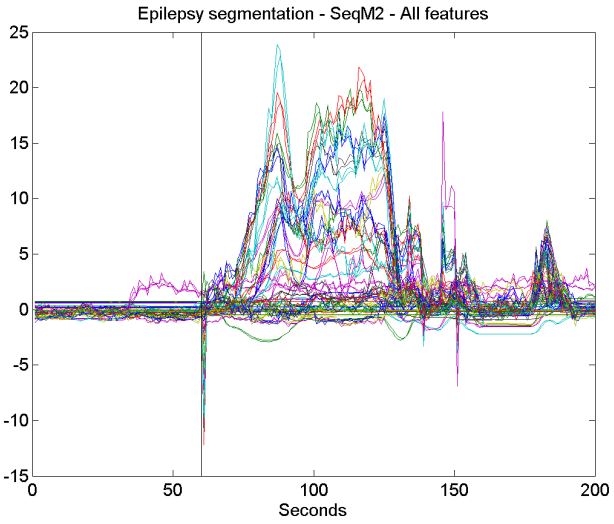


(b) Elbow function

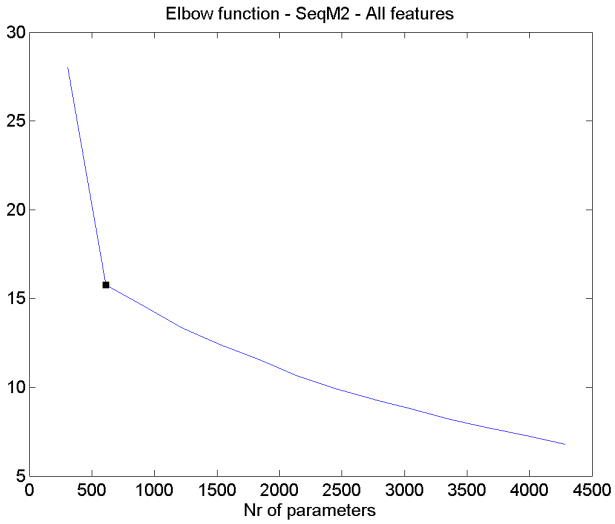


(c) Slope changes

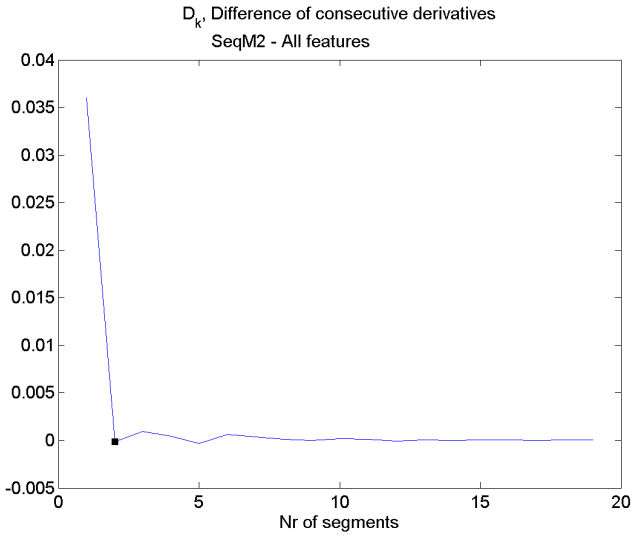
Figure 32: The segmentation of all features of seizure number 3 using SeqM1. Each segment is fitted using an autoregressive model with a trend. We see change-points around the start of the seizure at second 61.



(a) Segmentation



(b) Elbow function



(c) Slope changes

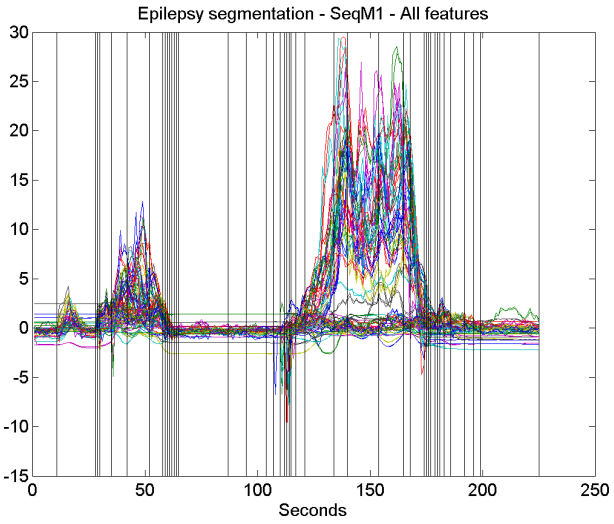
Figure 33: The segmentation of all features of seizure number 3 using SeqM2. Each segment is fitted using an autoregressive model with a trend. We see that we catch the seizure start at second 61.

start at second 60 but seizure 7 does not. In figure 34 we see how SeqM1 performs when segmenting this seizure. Again we have a lot of segment and it is doubtful that we can use the same adjusting parameter for all features. Figure 34b suggest that there are six groups of features that contain the same number of segments. These groups are not the same as the feature groups (see table 1). The correlation features have relatively few segments (2-3), while 16 of the 18 frequency bands have 5 segments. Most of the other features contain more segments. From figure 34a, we observe multiple common change-points near to each other, which are not joined. This suggests that different features change character with a delay and supports the idea of segmenting groups of features together. That can lead to more clear results about the behavior of the features. Figure 35 shows the segmentation performance of SeqM2. The segments clearly capture the activity changes, but not the segment changes of unique features. We do not capture the seizure start but it is interesting to know if it is at all possible using joint segmentation. Classification with the parameters of the segments is an interesting future step.

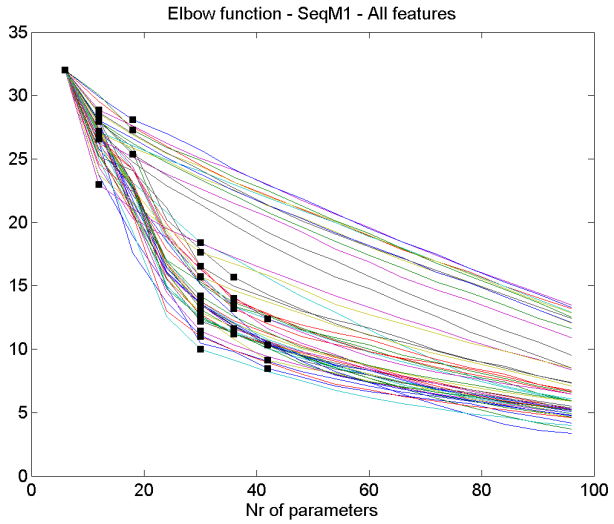
6 Discussion

Three different methods for segmenting multiple features with coordinated change-points are devised and compared via a simulation study. Two of these methods, SeqM1 (Sequential method 1) and SeqM2 (Sequential method 2), exhibit positive results and are examined further. The third method, ASM (All subset method), was discarded early in the study. SeqM1 and SeqM2 perform differently. SeqM1 has better overall performance and it is also better at finding the unique change-points. However, if the goal is to mainly detect common change-points, SeqM2 is preferable. SeqM2 rewards common change-points at the expense of the unique change-points. SeqM1 and SeqM2 are therefore promising methods each in their own way. An interesting future work for those two methods could be to extend them so that the adjusting parameter S is not a constant in each consecutive interval when the methods are used on-line. We could also plot (p_k, J_k) for some or all of the consecutive intervals but an adjusting parameter that adapts to locate the elbow is preferable when segmenting large amounts of data.

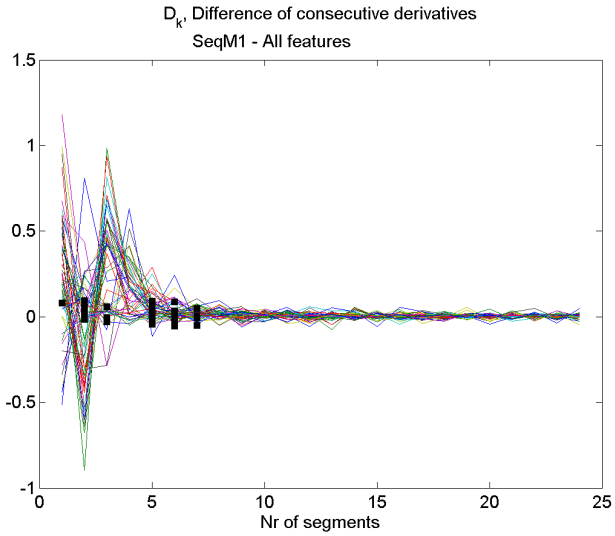
Various models have been applied to the music data. Fitting only a mean and variance to the data results in over-segmentation and does not capture the structure. By allowing for rich and complex models, e.g., polynomials of order 5 and trend models with autoregressive errors, no segments are needed to obtain a good model. For interpretation purposes, we cannot deduce anything about the music changes when fitting such complex models. Allowing



(a) Segmentation

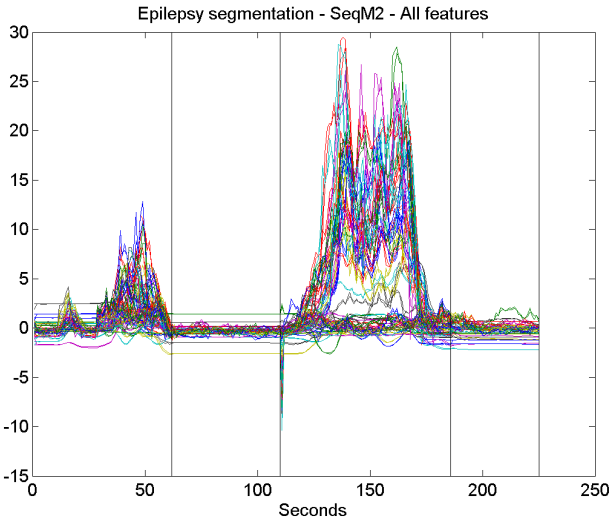


(b) Elbow function

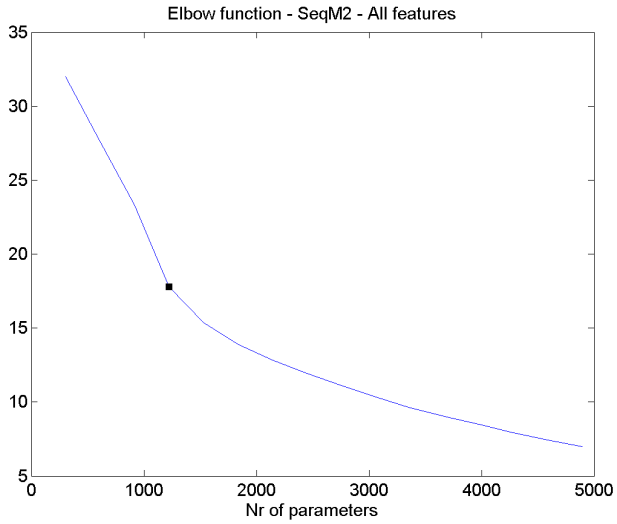


(c) Slope changes

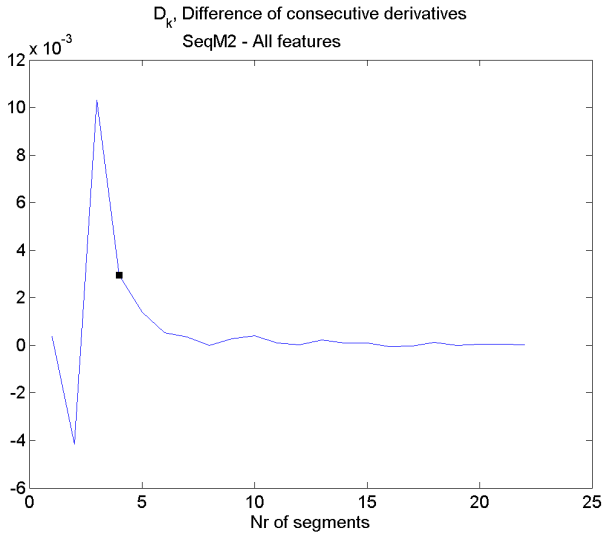
Figure 34: The segmentation of all features of seizure number 7 using SeqM1. Each segment is fitted using an autoregressive model with a trend. SeqM1 may not be appropriate to segment so many features at once.



(a) Segmentation



(b) Elbow function



(c) Slope changes

Figure 35: The segmentation of all features of seizure number 7 using SeqM2. Each segment is fitted using an autoregressive model with a trend. We see that the segments capture the activity changes.

the segments to switch between a mean and line and fitting polynomials of lower order gives interesting results on trend changes in the model. However, we cannot easily use the results to locate where the music changes. We see different performances when using SeqM1 and SeqM2. SeqM1 seems to outperform SeqM2, since SeqM2 favors common change-points at the cost of finding the unique ones. When using SeqM2, it is not possible to detect delay in characteristic changes which seems to occur between the features.

An autoregressive model with a trend was fitted to the epilepsy data. Since handpicking an adjusting parameter for each feature is time consuming, the same adjusting parameter is used for all features when applying SeqM1 on the data. This results in over-segmentation of some features. By using SeqM2 we only catch the most extreme changes. For future work it could be interesting to apply the method on subsets of the epilepsy data with common characteristics. The results might become easier to interpret as well as the choice of an adjusting parameter becomes easier. Another possibility for future work is to segment all the epilepsy data on-line. It could then be interesting to look further at how we can choose an appropriate adjusting parameter for all intervals. The aim with the epilepsy data is to be able to classify seizures from non seizure so future work includes applying classification methods on segmented epilepsy data.

ASM (All subset method), that was discarded early in the study, is restricted to datasets containing long segments due to computational burden which may possibly reduce the quality of the results from the elbow method. It does not have any advantages over SeqM1 and SeqM2. The results for the ASM method are in a way disappointing since it should be the method that takes into account simultaneously the location and the joining of change-points.

7 References

[1] Body score. `http://www.neuro.gu.se/neurovetenskap/bodyscore`. Accessed: 13/10/2012.

[2] I.E. Auger and C.E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989.

[3] G. Casella and R.L. Berger. *Statistical inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002.

[4] T. Haštie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

[5] A. Hildeman. Classification of epileptic seizures using accelerometers. Master's thesis, Chalmers University of Technology, 2011.

[6] M.H. Kutner. *Applied Linear Statistical Models*. The McGraw-Hill/Irwin series operations and decision sciences. McGraw-Hill Irwin, 2005.

[7] M. Lavielle. Using penalized contrasts for the change-point problem. Technical Report RR-5339, INRIA, October 2004.

[8] H. Madsen. *Time Series Analysis*. Texts in Statistical Science. Taylor & Francis, 2007.

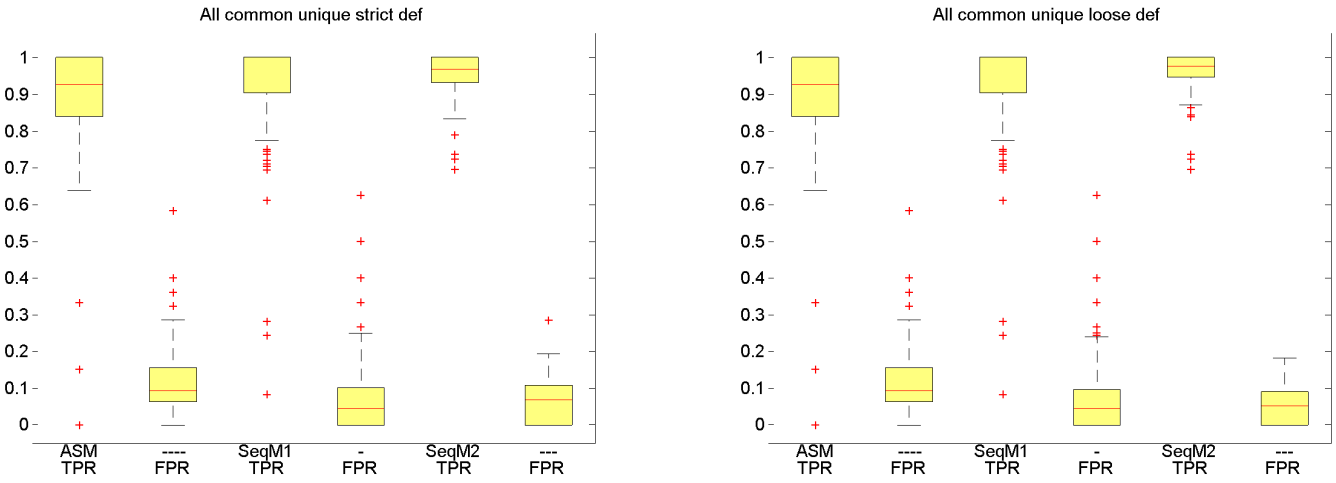
[9] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J-J. Daudin. A statistical approach for array cgh data analysis. *BMC Bioinformatics*, 6(1):27, 2005.

[10] J. Wipenmyr. Utveckling av metoder för diskriminering av epileptiska anfal. Master's thesis, Luleå University of Technology, 2011.

A Appendix

A.1 Additional results from simulation study

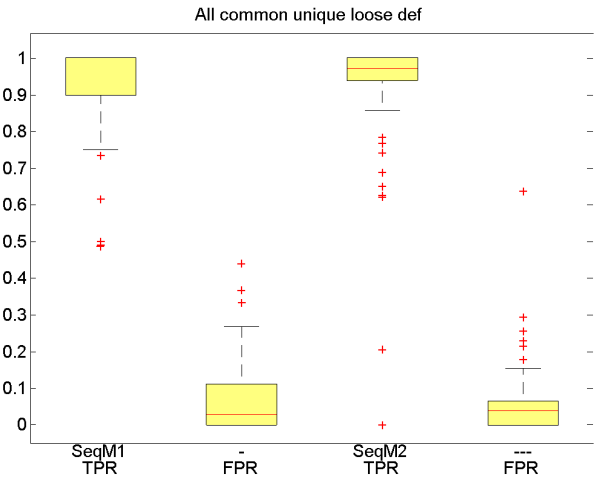
Figures 36, 37 and 38 contain box plots of TPR_{ucs}, TPR_{ucs},TPR_{ucl} and TPR_{ucl}, i.e., the overall true and false positive rates of finding unique and common change-points. See section 4.6 for further information.



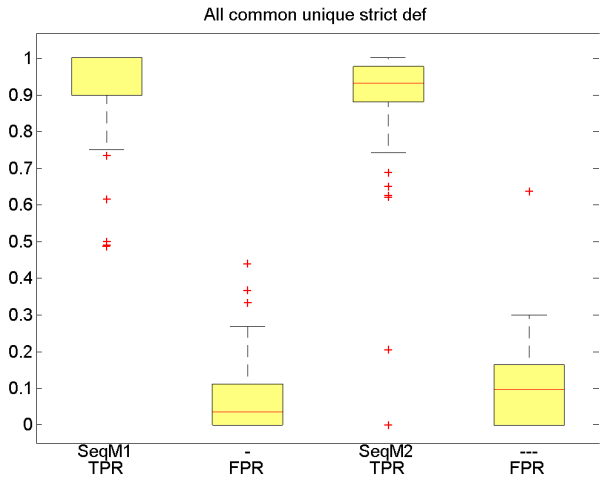
(a) Strict definition

(b) Loose definition

Figure 36: SIM1: The true and false positive rates for the overall detection of unique and common change-points, see section 4.6.

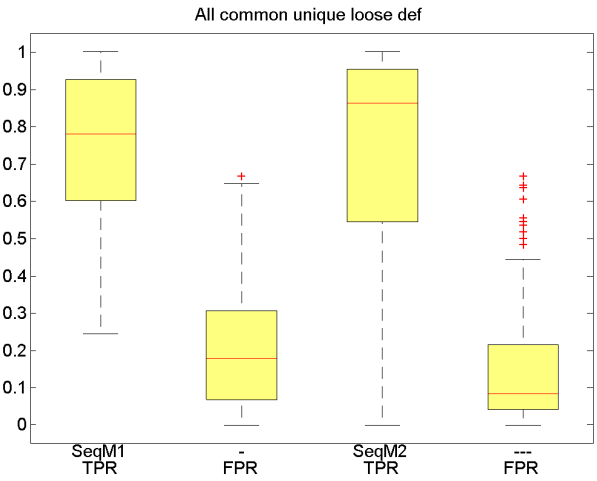


(a) Strict definition

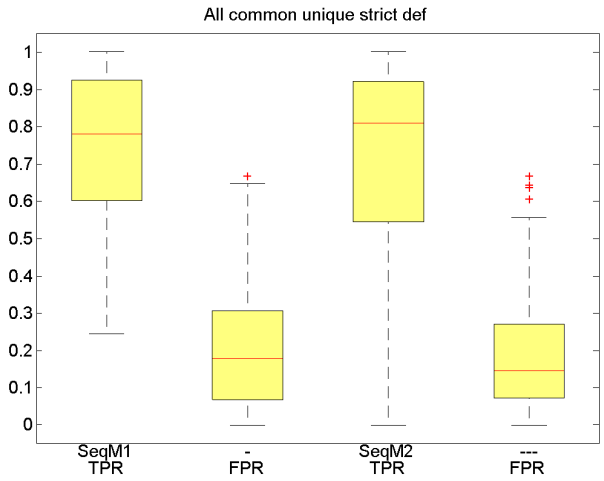


(b) Loose definition

Figure 37: SIM 2: The true and false positive rates for the overall detection of unique and common change-points, see section 4.6.



(a) Strict definition



(b) Loose definition

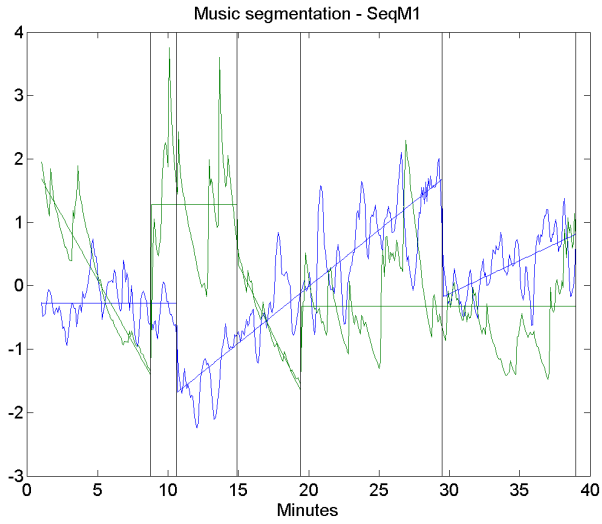
Figure 38: SIM 3: The true and false positive rates for the overall detection of unique and common change-points, see section 4.6.

A.2 Music results

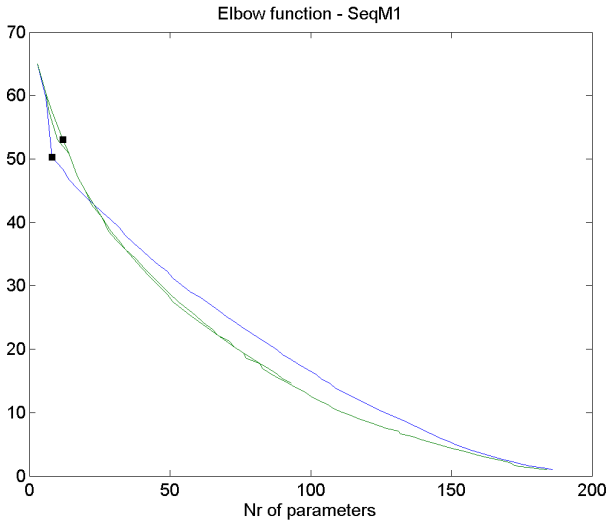
Table 9 contains a complete list of all models we fit to the music data. Figures referred to in the table are both located in the result chapter as well as here below.

Model type of segments	Nr. of Figure SeqM1	Nr. of Figure SeqM2
Mean model, $y_t = \alpha$.	14	15
Choosing between mean ($y_t = \alpha$) and line ($y_t = \alpha + \beta t$) models.	16	17
Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha - \beta x_t$.	18	19
Choosing between mean ($y_t = \alpha$) and line ($y_t = \alpha + \beta t$) model, using robust estimate.	43	44
Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha$.	41	42
Autoregressive errors: $r_t + \phi_1 r_{t-2} + \phi_1 r_{t-2} + \phi_3 r_{t-3} = \varepsilon_t$ where $r_t = y_t - \alpha - \beta t$, using robust estimate.	39	40
Polynomial models of order 3, $y_t = \sum_{i=0}^{s} p_i t^i, s = 3$.	20	21
Polynomial models of order 4, $y_t = \sum_{i=0}^{s} p_i t^i, s = 4$.	45	46
Polynomial models of order 5, $y_t = \sum_{i=0}^{s} p_i t^i, s = 5$.	22	23

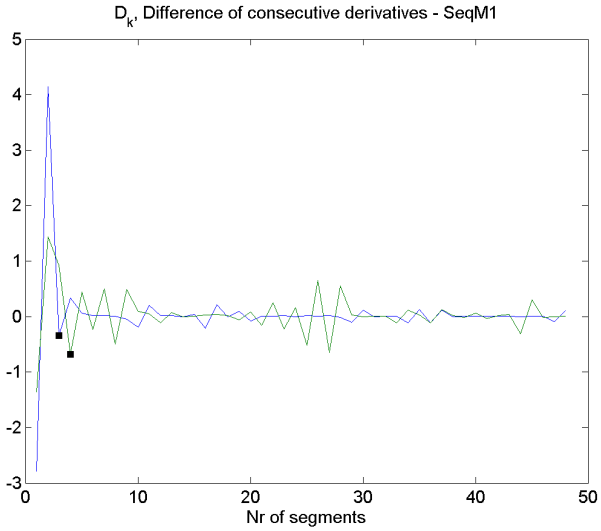
Table 9: A complete list of all models fitted to the music data.



(a) Segmentation

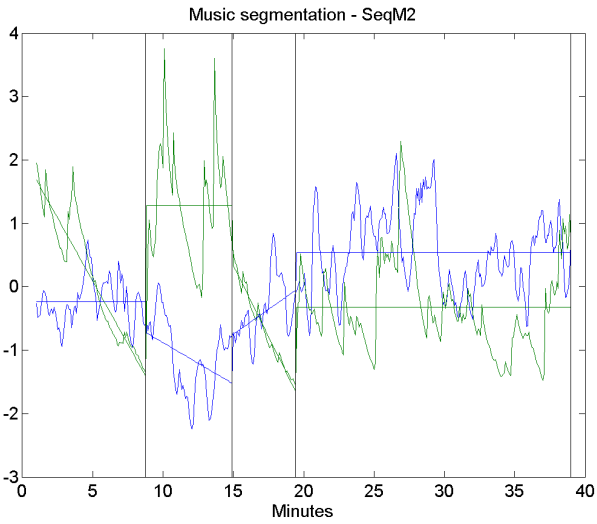


(b) Elbow function

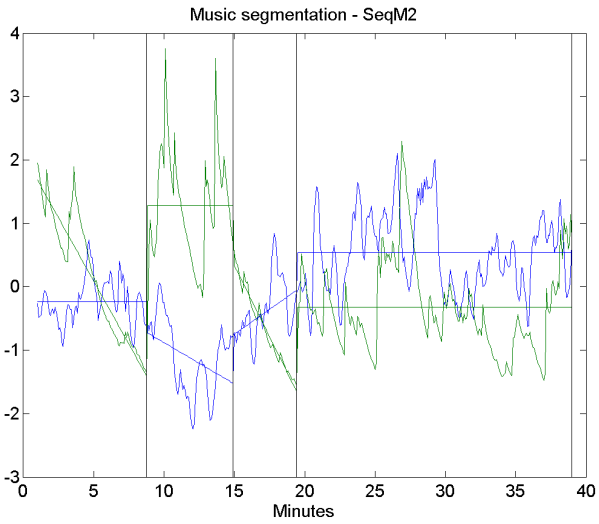


(c) Slope changes

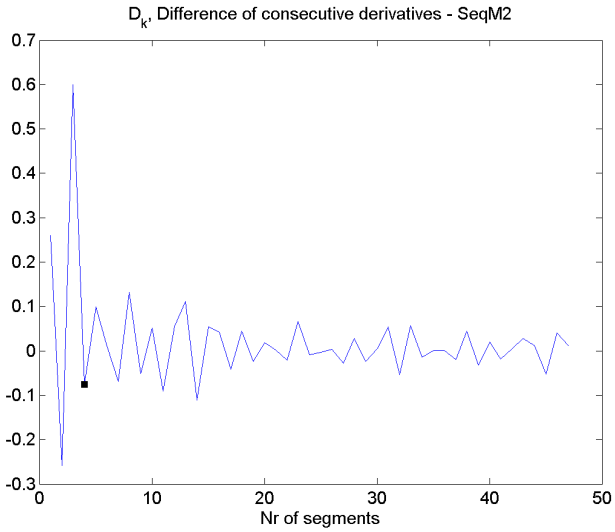
Figure 39: SeqM1, modeled with a mean and line, using robust estimate for the regression parameters.



(a) Segmentation

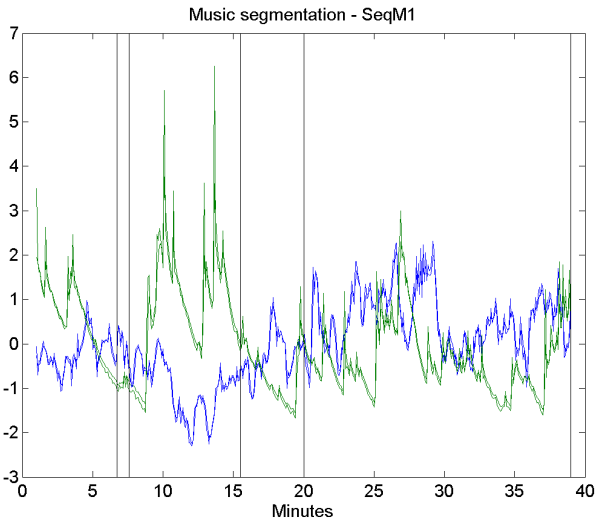


(b) Elbow function

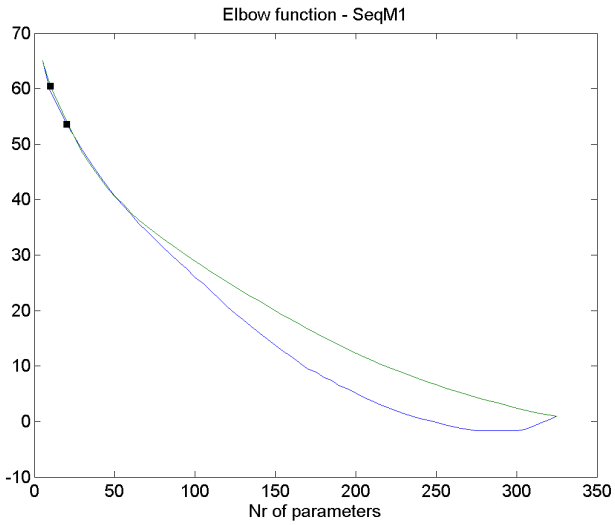


(c) Slope changes

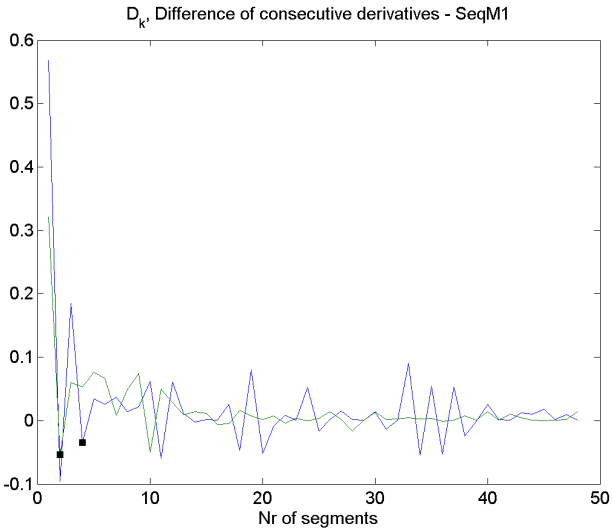
Figure 40: SeqM2, modeled with a mean and line, using robust estimate for the regression parameters.



(a) Segmentation

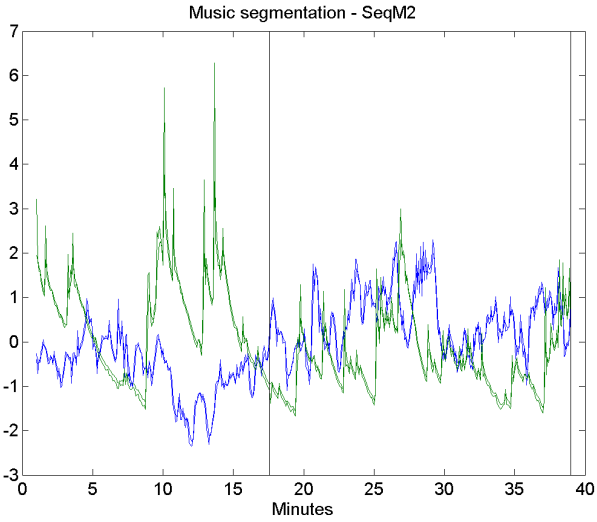


(b) Elbow function

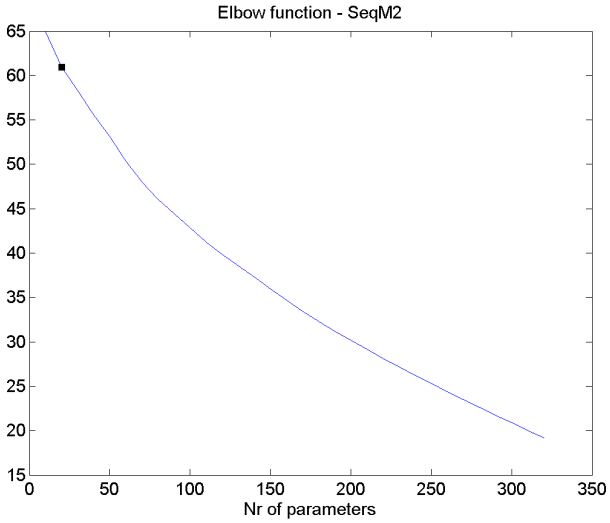


(c) Slope changes

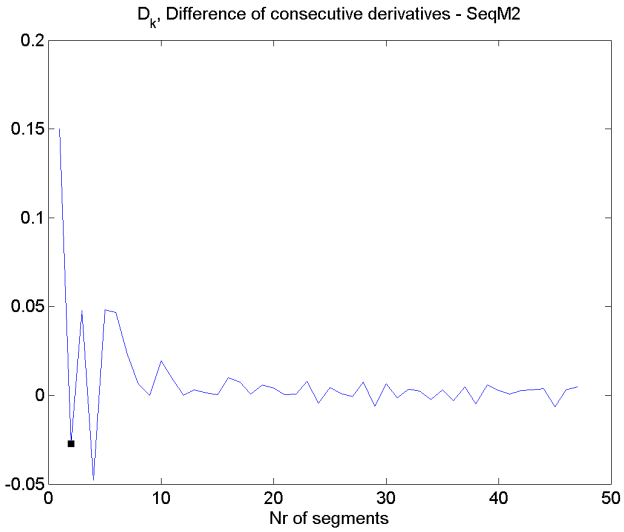
Figure 41: SeqM1, modeled with a mean where the errors are autoregressive of third order.



(a) Segmentation

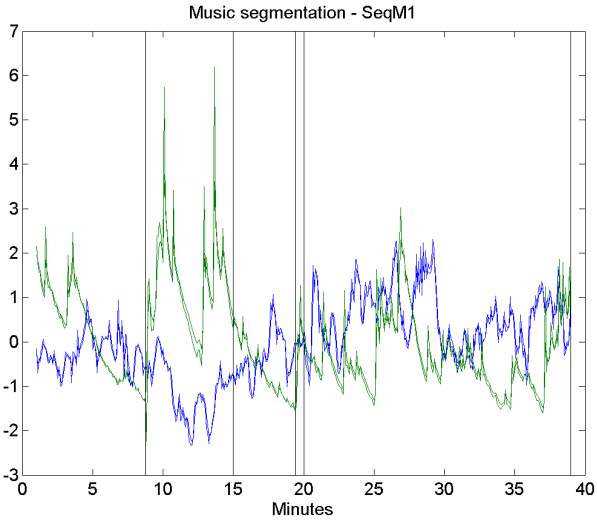


(b) Elbow function

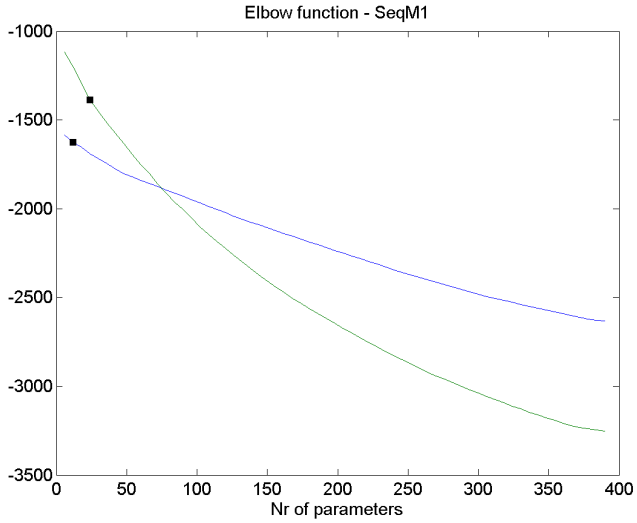


(c) Slope changes

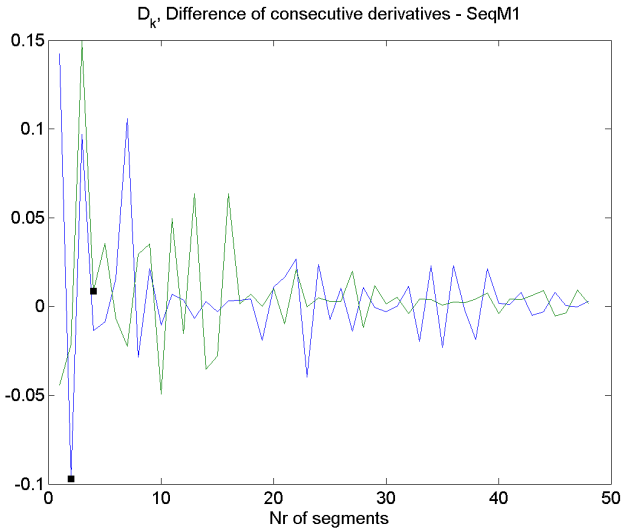
Figure 42: SeqM2, modeled with a mean where the errors are autoregressive of third order.



(a) Segmentation

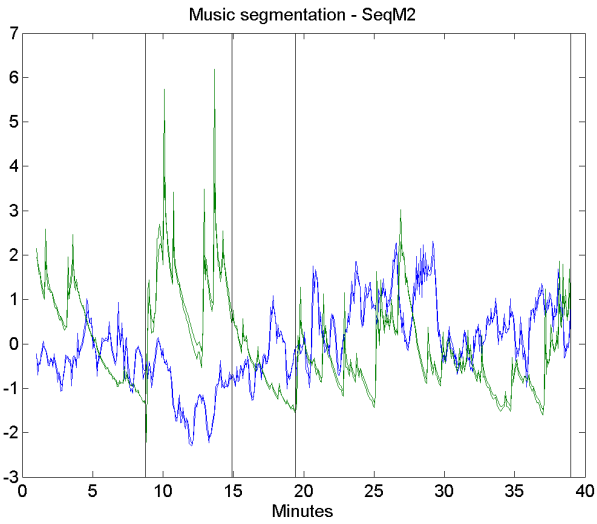


(b) Elbow function

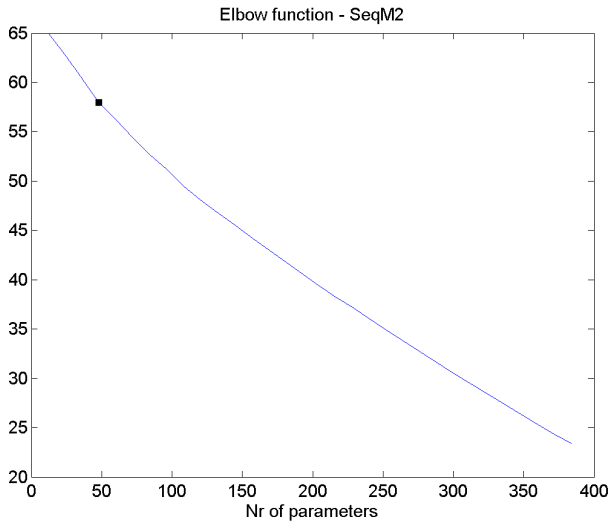


(c) Slope changes

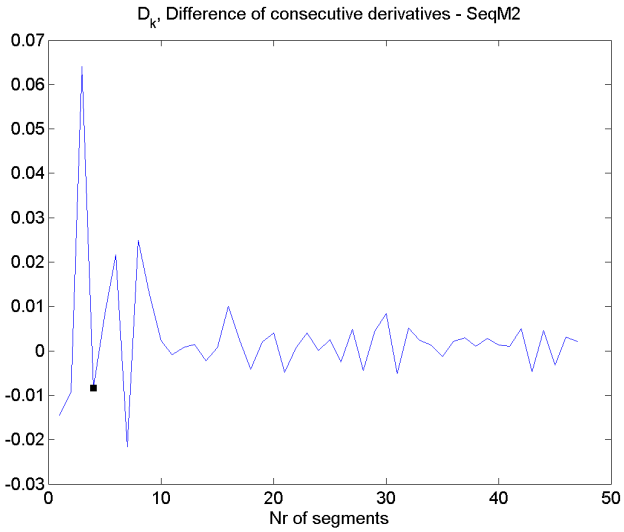
Figure 43: SeqM1, modeled with a line, using robots estimate. The errors are autoregressive of order 3.



(a) Segmentation

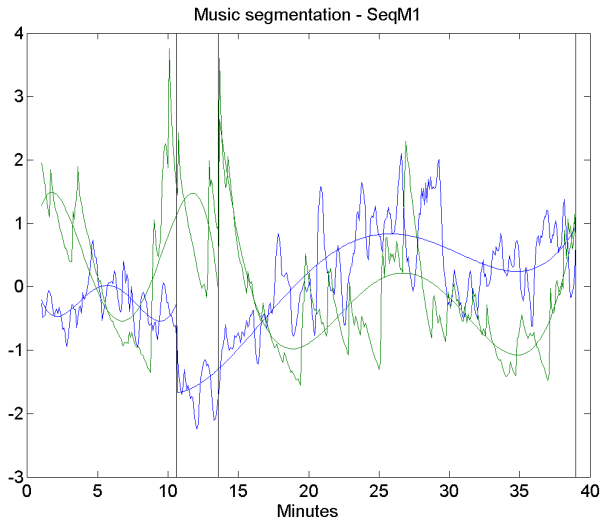


(b) Elbow function

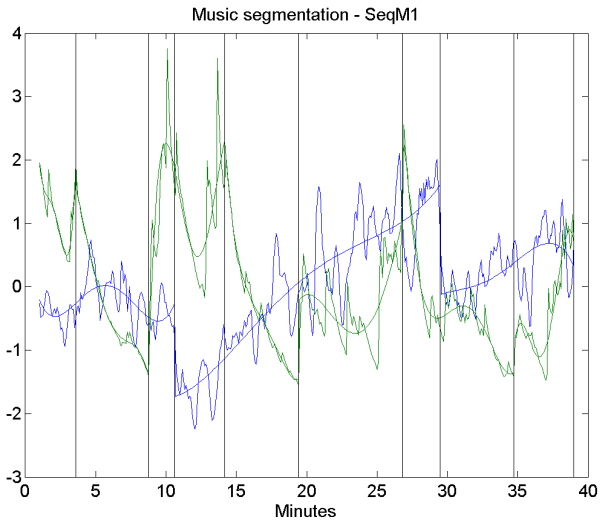


(c) Slope changes

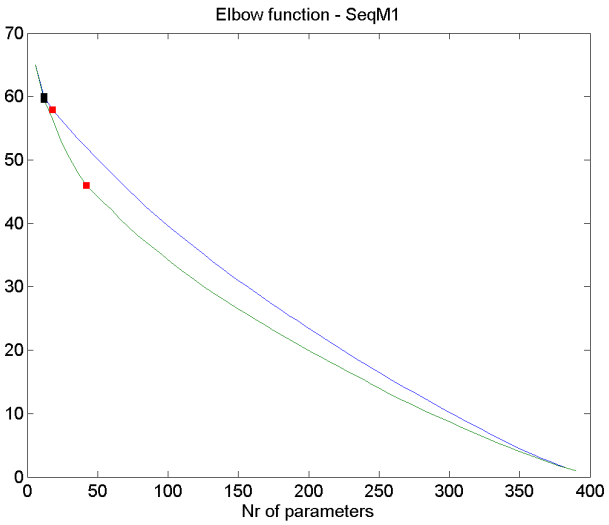
Figure 44: SeqM2, modeled with a line, using robots estimate. The errors are autoregressive of order 3.



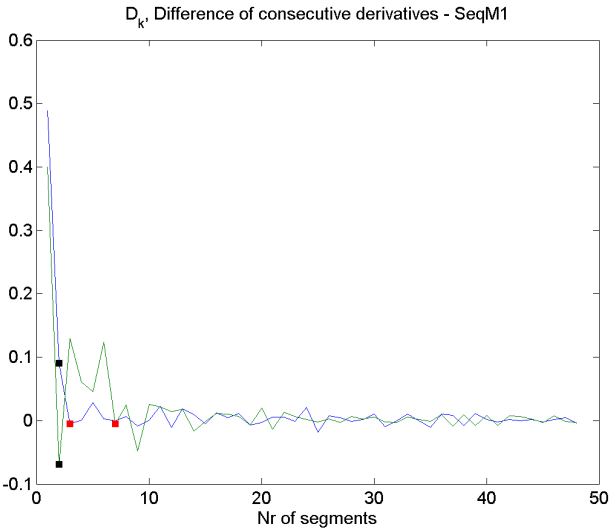
(a) Segmentation



(b) Segmentation

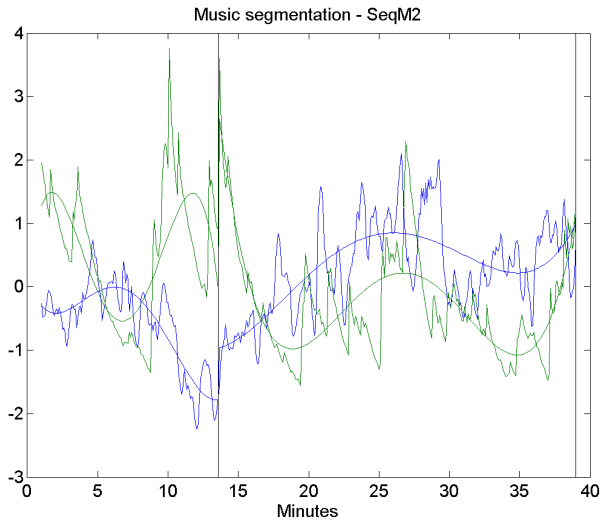


(c) Elbow function

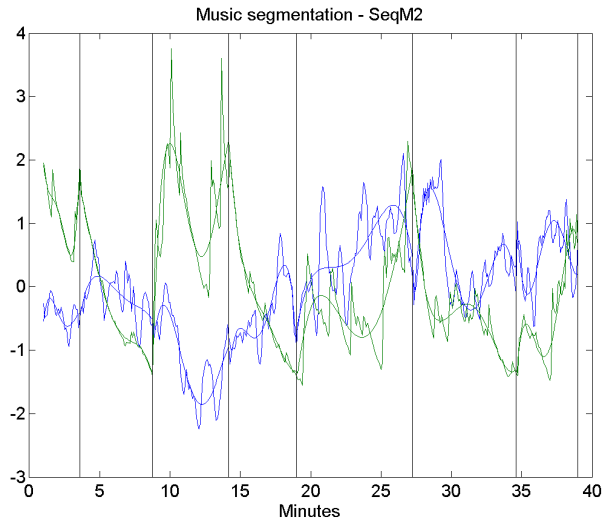


(d) Slope changes

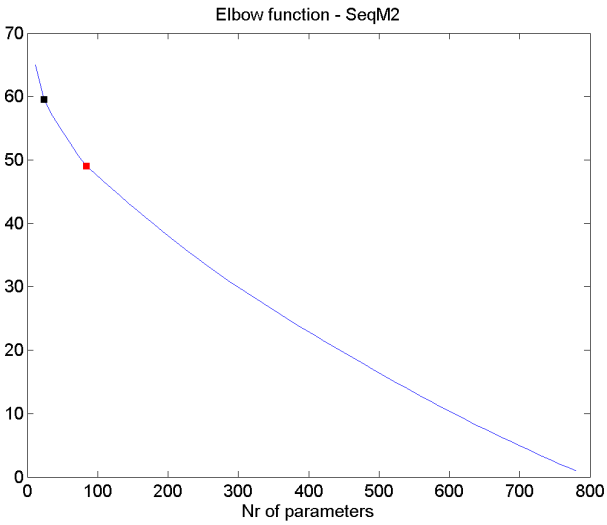
Figure 45: SeqM1, modeled with polynomials of order 4. The black boxes correspond to figure 45a while the red boxes correspond to 45b.



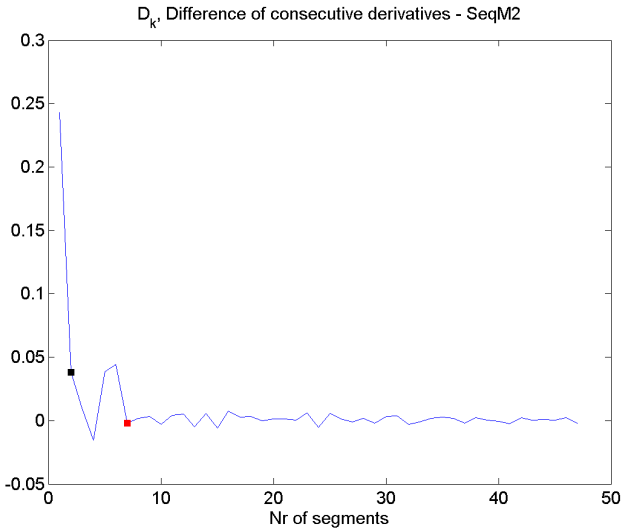
(a) Segmentation



(b) Segmentation



(c) Elbow function



(d) Slope changes

Figure 46: SeqM2, modeled with polynomials of order 4. The black boxes correspond to figure 46a while the red boxes correspond to 46b.

Printed and Bound at
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
2012