Title

Change Point Detection for Image, Graph and Network Data

Author

Xu, Cong

Publication Date

2021

Change Point Detection for Image, Graph and Network Data

By

CONG XU
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Statistics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____
Thomas C.M. Lee, Chair

____
Alexander Aue

____
Jane-Ling Wang

Committee in Charge

2021

For the previous days and nights.

CONTENTS

ABSTRACT

Change Point Detection for Image, Graph and Network Data

Given a set of time series data, the goal for change point detection is to locate, if any, those time points at which some characteristic of the data changes. The need for change point detection arises in many contexts, including stock market prediction, weather forecast, and air pollution monitoring. This thesis studies change point detection for three types of complex data: time series of astronomical images, sequences of undirected graphs, and time-evolving dynamic networks. From each of these three problems, the minimum description length principle is invoked to derive a model selection criterion, which is shown to yield statistically consistent estimates for the change points and other model parameters. Practical, tailored algorithms are also developed to compute these estimates.

ACKNOWLEDGMENTS

First of all, I am extremely grateful to my supervisor, Prof. Thomas C. M. Lee, for his invaluable advice and continuous support during my Ph.D. study. I am indebted to other members of my dissertation committee, Prof. Alexander Aue, Prof. Jane-Ling Wang, Prof. Can Le, and Prof. James Sharpnack, for their time and valuable suggestions. I would also like to thank Dr. Hans Moritz Günther, Dr. Vinay L. Kashyap, and Prof. Andreas Zezas for many valuable discussions on the scientific background for the problem in Chapter 2. Lastly, I would like to thank Prof. Rex Cheung for sending some of the computing codes used in Chapter 4.

Chapter 1

Introduction

In the analysis of many real world complex data, a common task is to identify and study the changes of states over time. The goal is to extract meaningful information about different states, as well as to understand the transition processes between states.

Very often it is known as the change point detection problem. Methods for change point detection can be divided into two categories: “online” methods that aim to detect change points as soon as they occur, and “offline” methods that retrospectively detect changes after all samples have been collected. This thesis focuses on the offline methods.

Many offline change point detection problems can be solved via a model selection approach. That is, an objective function is derived for choosing the number and locations of the change points, as well as the best fitting model between each pair of successive change points.

For the model selection approach, typically three major components are involved:

1. A model selection criterion to derive the objective function.
2. A practical algorithm for optimizing the objective function.
3. A theoretical analysis on the objective function.

Component 1: Model Selection. This thesis uses the minimum description length (MDL) principle as the model selection criterion. Loosely speaking, MDL defines the best model as the one that produces the shortest code length of the data. In general, to apply the MDL principle, one needs to derive a code length formula for the task at hand. This formula, which is the objective function mentioned before, depends on, amongst others, the model assumptions and the format of the data. An advantage of applying the MDL principle is that, once the code length formula is constructed, no free hyper-parameter needs to be chosen.

Component 2: Optimization. The computational complexity for minimizing the MDL criterion changes for different tasks. As usually exact solutions are impractical for complicated settings, fast and approximate solutions are adopted instead. This thesis studies two types of optimization procedures for change point problems:

- One-stage procedures: simultaneously search for change points and best fitting models between adjacent change points.

- Two-stage procedures: first search for change points and then find the best fitting model for each time interval.

Component 3: Theoretical Analysis. This thesis studies the consistency properties of the estimated change point as well as the fitted model parameters in each time interval. In some cases, it is assumed that the number of samples (or number of time points) T goes to infinity. This asymptotic setting can be interpreted as follows: an observation is viewed as one realization of a continuous-time process on an equal-spaced grid of size $1/T$, and $T \to \infty$ means that the size of the grid converges to 0 (Truong *et al.*, 2020). Another case that this thesis covers is that the number of observations at each time point goes to infinity, while the number of time points T remains fixed.

The thesis focuses on three change point problems:

1. Change point detection and image segmentation for time series of astronomical images.

2. Change point detection and partitioning of a sequence of structured signals (graphs).

3. Change point detection and community detection of time-evolving dynamic networks.

These problems will be discussed in Chapters 2, 3 and 4, respectively.

1.1 Problem I: Time Series of Astronomical Images

In X-ray astronomy, the data are obtained as a list of photons, each with four attributes: the 2D spatial coordinates, the times they were recorded, and their energies (or wavelengths). After binning the data, we obtain a 4D table of photon counts indexed by the 2D coordinates (x, y), time index t and energy band w.

Our method assumes that at each time point, the corresponding multi-band image is an unknown 3D piecewise constant function corrupted by Poisson noise. It also assumes that all images between any two adjacent change points (in the time domain) share the same unknown piecewise constant function. An MDL criterion is derived given the model assumptions. It is shown that the consistency of the MDL-based results hold when the number of time points T goes to infinity.

The optimization procedure is a typical two-stage process. Bottom-up search, a greedy algorithm, is used to search for change points. The seeded region growing method and region merging are used for the image segmentation task for the images in each time interval.

1.2 Problem II: Time Series of Graphs

Suppose what we observed is a time-evolving graph, with the number of nodes and the node connectivity (i.e., edges) keep fixed over time. It is assumed that the measurements observed at these nodes are Normally distributed and they share the same variance. The means of the Normal distributions correspond to the true signals for each node. After making assumptions on spatial and temporal smoothness for the true signals, MDL principle is derived for this task. In terms of the theoretical property, it can be shown that the MDL-based solution is consistent, when the numbers of observations in each node go to infinity.

Drawing the idea from generalized total variation denoising (Bleakley and Vert, 2011) and graph-guided-fused-lasso (Chen *et al.*, 2010; Kim *et al.*, 2009), a sequence of candidate models can be generated by solving a convex optimization problem. Smoothing proximal gradient method (Chen *et al.*, 2012) can be used for this problem when the objective function is not smooth. After that, the best model based on MDL will be selected from these candidates. Therefore, the optimization procedure belongs to the one-stage procedure we mentioned, because it simultaneously detects change points and fits models.

1.3 Problem III: Time-evolving Dynamic Networks

In this part we study the change point and community detection for time-evolving dynamic networks. A time-evolving dynamic networks is composed by a sequence of undirected unweighted networks which share the same nodes. It is assumed that all the networks follow the stochastic block mode (SBM) and the networks within the same time interval have the same community assignment and the link probabilities. After deriving MDL based on the model assumptions, the consistency of the MDL-based result when the number of time points T goes to infinity is proved.

The two-stage optimization procedure is composed of the bottom-up search, which is used for searching change points, and the variational approximation method motivated by Daudin *et al.* (2008), which estimates the community assignment as well as the link probabilities.

Chapter 2

Change point detection and image segmentation for time series of astrophysical images

Many astrophysical phenomena are time-varying, in the sense that their intensity, energy spectrum, and/or the spatial distribution of the emission suddenly change. This paper develops a method for modeling a time series of images. Under the assumption that the arrival times of the photons follow a Poisson process, the data are binned into 4D grids of voxels (time, energy band, and x-y coordinates), and viewed as a time series of non-homogeneous Poisson images. The method assumes that at each time point, the corresponding multi-band image stack is an unknown 3D piecewise constant function including Poisson noise. It also assumes that all image stacks between any two adjacent change points (in time domain) share the same unknown piecewise constant function. The proposed method is designed to estimate the number and the locations of all the change points (in time domain), as well as all the unknown piecewise constant functions between any pairs of the change points. The method applies the minimum description length (MDL) principle to perform this task. A practical algorithm is also developed to solve the corresponding complicated optimization problem. Simulation experiments and applications to real datasets show that the proposed method enjoys very promising empirical properties. Applications to two real datasets, the XMM observation of a flaring star and an emerging solar coronal loop, illustrate the usage of the proposed method and the scientific insight gained from it.

2.1 Introduction

Many phenomena in the high-energy universe are time-variable, from coronal flares on the smallest stars to accretion events in the most massive black holes. Often, this variability can just be seen "by-eye" but at other times, we need to use robust methods founded in statistics to distinguish random noise from significant variability. Realizing where the change has occurred is critical for subsequent scientific analyses, e.g., spectral fitting and light curve modeling. Such analyses must focus on those intervals in data space which are properly tied to the changes in the physical processes that generate the observed photons. Therefore, it is of importance to identify sources as well as to locate their spatial boundaries. Our goal is to detect change points in the time direction; that is, the times at which sudden changes happened during the underlying astrophysical process.

Change point detection in time series is well studied, and several algorithms employing different philosophies have been developed. For example, Aue and Horváth (2013) employed hypothesis testing to study structural break detection, using both non-parametric approaches like cumulative sum (CUSUM) and parametric methods like likelihood ratio statistic to deal with different kinds of structural breaks. Another likelihood-based approach commonly used to analyze astronomical time series is Bayesian Blocks (Scargle *et al.*, 2013) which finds change points by fitting piecewise constant models between change points. A good example of the model driven approach is the Auto-PARM procedure developed by Davis *et al.* (2006). By modeling the piecewise-stationary time series, the procedure is able to simultaneously estimate the number of change points, their locations and the parametric model for each piece. Here the minimum description length (MDL) principle by Rissanen (1989b, 2007a) is applied in the model selection procedure. Davis and Yau (2013) proved the strong consistency of the MDL-based change point detection procedure. Another example is Automark by Wong *et al.* (2016), who developed an MDL-based methodology which detects the changes in observed emission from astronomical sources in 2-D time-wavelength space.

Dey *et al.* (2010) loosely classified image segmentation techniques into two categories: (i) image driven approaches and (ii) model driven approaches. Image driven segmentation techniques are mainly based on the discrete pixel values of the image. For example, the graph-based algorithm by Felzenszwalb and Huttenlocher (2004) treats pixels as vertices, and the weights of the edges are based on the similarity between the features of pixels. The evidence for a boundary between two regions can be measured based on the graph. Such methods work in many complicated cases as there is no underlying model for images. Model driven approaches rely upon the information of the structure of the image. These methods are based on the assumption that the pixels in the same region have similar characteristics. The Blobworld framework by Carson *et al.* (1999) assumes that the features of pixels are from a underlying multivariate Gaussian mixture model. Neighboring pixels whose features are from the same Gaussian distribution are grouped into the same region.

Here we follow the model driven approach. In order to develop a change point detection method for image time series data, we begin by specifying an underlying statistical model for the images between any two consecutive change points. In doing so we also study the statistical properties of the change point detection method. We assume the underlying Poisson rate for each of the images follows a piecewise constant function. Therefore, the region growing algorithm developed by Adams and Bischof (1994) for greyvalue images can be naturally applied.

Given the previous successes of applying the MDL principle (Rissanen, 1989b, 2007a) to other time series change point detection and image segmentation problems (e.g., Davis *et al.*, 2006; Lee, 2000; Wong *et al.*, 2016), here we also use MDL to tackle our problem of joint change point detection and image segmentation for time series of astronomical images. Briefly, MDL defines the best model as the one that produces the best lossless compression of the data. There are different versions of MDL, and the one we use is the so-called two-part code; a gentle introduction can be found in Lee (2001). When comparing with other versions of MDL such as normalized maximum likelihood, one advantage of the two-part version is that it tends to be more computationally tractable for complex problems such as the one this paper considers. It has also been shown to enjoy excellent theoretical and empirical properties in other model selection tasks (e.g., Aue and Lee, 2011; Lee, 2000; Davis *et al.*, 2006; Davis and Yau, 2013) Based on MDL, we develop a practical algorithm that can be applied to simultaneously estimate the number and locations of the change points, as well as to perform image segmentation on each of the images.

2.2 Methodology

Our method is applied to 4-D data cubes where 2-D spatial slices in several energy passbands are stacked in time. Such data cubes are commonly available, though in high-energy astrophysics, data are usually obtained in the form of a list of photons. The list contains the two-dimensional spatial coordinates where the photons were recorded on the detector, the times they were recorded, and their energies or wavelengths. To facilitate our analysis, we bin these data into a 4-D rectangular grid of boxes. After the binning of the original data, we obtain a 4D table of photon counts indexed by the two-dimensional coordinates (x, y), time index t and energy band w. The dataset is thus a series of multi-band images with counts of photons as the values of the pixels. Since the emission times of photons can be considered a non-homogeneous Poisson process, and the grids do not overlap with each other, the counts in each pixel are independent, and the image slices are also independent.

We first partition these images into a set of non-overlapping region segments using a seeded region growing (SRG) method, and then merging adjacent segments to minimize MDL (see Section 2.2.1). The counts in each segment are modeled as Poisson counts (see Section 2.2.2; the implementation details of the algorithm are described in Section 2.2.3). We minimize the MDL criterion across the images by iteratively removing change points along the time axis and applying the SRG segmentation onto the images in each of the time intervals. Key pixels that are influential in how the segmentations and change points are determined are then identified through searching for changes in the fitted intensities (see Section 2.2.4). Such regions are the focus of follow-up analyses.

We list the variables, parameters, and notations used here in Table 2.1.

2.2.1 Region Growing and Merging

As a first step in the analysis, a suitable segmentation method must be applied to the images to delineate regions of interest (ROIs). For this, we use the seeded region growing (SRG) method of Adams and Bischof (1994) to obtain a segmentation of the image. We choose SRG over other image segmentation algorithms for its speed and reliability (Fan and Lee, 2015). Also, it can be straightforwardly incorporated to the Poisson setting.

At the beginning of SRG, we select a set of seeds, manually or automatically, from the

Notation	Definition
N_{I}	number of pixels in each 2-D spatial image
N_{T}	number of time bins
N_{W}	number of energy bands
ΔT_t	duration of the t^{th} time bin
$y_{i,t,w}$	photon counts within the i^{th} spatial pixel, the t^{th} time interval and the w^{th} energy range
$\lambda_{i,t,w}$	Poisson rate for the i^{th} spatial pixel, the t^{th} time interval and the w^{th} energy range
K	number of change points
τ_k	location of the k^{th} change point
$m^{(k)}$	number of region segments for the k^{th} interval between two consecutive change points
$a_h^{(k)}$	the area (number of pixels) of the h^{th} region segment of the k^{th} interval between two consecutive change points
$b_h^{(k)}$	the "perimeter" (number of pixel edges between this and neighboring regions) of the h^{th} region of the k^{th} interval
$\mu_{h,w}^{(k)}$	Poisson rate for the h^{th} region segment and the w^{th} energy range of the k^{th} interval
$\hat{\mu}_{h,w}^{(k)}$	fitted Poisson rate for the h^{th} region segment and the w^{th} energy range of the k^{th} interval

Table 2.1: Major notations.

image. Each seed can be a single pixel or a set of connected pixels. A seed comprises an initial region. Then each region starts to grow outward until the whole image is covered. (See Section 2.2.3.2 offers some suggestions on the selection of initial seeds.)

At each step, the unlabelled pixels which are neighbors to at least one of the current regions comprise the set of candidates for growing the region. One of these candidates is selected to merge into the region, based on the Poisson likelihood that measures the similarity between a candidate pixel and the corresponding region. We repeat this process until all the pixels are labeled, thus producing an initial segmentation by SRG.

At the end of the SRG process, we are left with an oversegmentation, i.e., with the image split into a larger than optimal number of segments. We then merge these segments based on the largest reduction or smallest increase in the MDL criterion (see below). From this sequence of segmentations, we select the one that gives the smallest value of the MDL criterion as the final ROIs.

2.2.2 Modeling a Poisson Image Series

2.2.2.1 Input Data Type

We require that the data are binned into photon counts in an $N_{\mathrm{I}} \times N_{\mathrm{T}} \times N_{\mathrm{W}}$ tensor $\{y_{i,t,w}\}, i = 1, ..., N_{\mathrm{I}}, t = 1, ..., N_{\mathrm{T}}, w = 1, ..., N_{\mathrm{W}}$, where $y_{i,t,w}$ is the photon counts within the i^{th} spatial rectangular region, the t^{th} time interval $[T_{t-1}, T_t)$ and the w^{th} energy range $[W_{w-1}, W_w)$. After binning, the data can be viewed as a time series of images in different energy bands. The values of each pixel are the photon counts in the different bands in the corresponding spatial region.

Notice that compared with Automark (Wong *et al.*, 2016), we incorporate 2-D spatial information into the model, thus extending the analysis from two (wavelength/energy and time) to four dimensions (wavelength/energy, time, and projected sky location). We also relax the restriction that the bin sizes along any of the axes are held fixed. Thus, sharp changes are more easily detected.

As the data in high-energy astrophysics are photon counts, we use a Poisson process to model the data,

$$y_{i,t,w} \overset{i.i.d.}{\sim} \text{Poisson}(\lambda_{i,t,w} \Delta T_t), \tag{2.1}$$

where $\Delta T_t = (T_t - T_{t-1})$.

Our goal is to infer model intensities $\lambda_{i,t,w}$ from the observed counts data $\{y_{i,t,w}\}$. We are especially interested in detecting significant changes of $\lambda_{i,t,w}$ over time. If there are changes, we also want to estimate the number and locations of the change points.

To simplify the presentation, we first develop a time-homogeneous model, i.e., one where there are no change points and $\lambda_{i,t,w}$ is unchanging with t (Section 2.2.2.2). We will then consider more complex cases, where change points are added to the model so that $\lambda_{i,t,w}$ is allowed to change over time (Section 2.2.2.3)

2.2.2.2 Piecewise Constant Model

First consider a temporally homogeneous Poisson model without any change points. Then each image can be treated as an independent Poisson realization of the same, unknown, true image.

We model the image as a 3-dimensional piecewise constant function. That is, the 2-dimensional space of x-y coordinates is partitioned into m non-overlapping regions such that all the pixels in a given region have the same Poisson intensity. Different energy bands share the same spatial partitioning. Rigorously, the Poisson parameter $\lambda_{i,t,w}$ can be written as a summation of region-specific Poisson rates $\mu_{h,w}$ times the corresponding indicator functions of regions ($I_{\{i \in R_h\}}$ is 1 for pixel i in region R_h and 0 otherwise) in the following format:

$$\lambda_{i,t,w} = \sum_{h=1}^{m} \mu_{h,w} I_{\{i \in R_h\}}. \tag{2.2}$$

Here $i \in R_h$ means "the i^{th} pixel in the h^{th} region" and I is the indicator function. R_h is the index set of the pixels within the h^{th} region, with $R_h \subseteq \{1, ..., N_{\text{I}}\}$. Also, $\mu_{h,w}$ is the Poisson rate for the w^{th} band of the h^{th} region. The partition of the image is specified by $\boldsymbol{R} = \{R_h | h = 1, ..., m\}$.

2.2.2.3 Adding Change Points to the Model

Now we allow the underlying Poisson parameter $\lambda_{i,t,w}$ to change over time t. We model $\lambda_{i,t,w}$ as a piecewise constant function of t.

Suppose these N_{T} images can be partitioned into $K+1$ homogeneous intervals by K change points

$$\tau = \{\tau_0 = 0,\ \tau_1, \tau_2, ..., \tau_K,\ \tau_{K+1} = N_{\text{T}}\}$$

For the t^{th} image, suppose that it belongs to the k^{th} time interval; i.e., $t \in (\tau_{k-1}, \tau_k]$. For each given t, let λ be a two-dimensional piecewise constant function with $m^{(k)}$ constant regions. Then λ can be represented by

$$\lambda_{i,t,w} = \sum_{k=1}^{K+1} I_{\{t \in (\tau_{k-1}, \tau_k]\}} \sum_{h=1}^{m^{(k)}} \mu_{h,w}^{(k)} I_{\{i \in R_h^{(k)}\}}, \tag{2.3}$$

where $m^{(k)}$ is the number of regions within the k^{th} interval. Let $\mathcal{M} = \{m^{(k)} | k = 1, 2, ..., K+1\}$. The partition of the images within interval k is specified by $\boldsymbol{R}^{(k)} = \{R_h^{(k)} | h = 1, ..., m^{(k)}\}$. And the overall partition is $\mathcal{R} = \{\boldsymbol{R}^{(k)} | k = 1, 2, ..., K+1\}$. The Poisson rates $\mu_{h,w}^{(k)}$ is the value for the w^{th} band in the h^{th} region of the k^{th} interval. Let $\boldsymbol{\mu}^{(k)} = \{\mu_{h,w}^{(k)} | h = 1, ..., m^{(k)}, w = 1, ..., N_{\text{W}}\}$. And let $\boldsymbol{\mu} = \{\boldsymbol{\mu}^{(k)} | k = 1, ..., K+1\}$, and $i \in R_h^{(k)}$ means "the i^{th} pixel is in the h^{th} region of the k^{th} interval".

2.2.2.4 Model Selection Using MDL

Given the observed images $\{y_{i,t,w}\}$, we aim to obtain an estimate of $\lambda_{i,t,w}$. In other words, we want an estimate of the image partitions and the Poisson rates of the regions for each band. It is straightforward to estimate the Poisson intensities given the region partitioning, but the partitioning is a much more complicated model selection problem.

We will apply MDL to select the best fitting model. Loosely speaking, the idea behind MDL for model selection is to first obtain a MDL criterion for each possible model, and then define the best fitting model as the minimizer of this criterion. MDL defines the best model as the one that produces the best compression of the data. The criterion can be treated as the code length, or amount of hardware memory required to store the data.

First we present the MDL criterion for the homogeneous Poisson model, then follow it by the MDL criterion for the general case (i.e., with change points).

Following similar arguments as in Lee (2000) (see their Appendix B), the MDL criterion for segmenting N_{T} homogeneous images is

$$\begin{aligned} \mathrm{MDL}(m, R, \hat{\mu}) &= m \log(N_{\mathrm{I}}) + \frac{\log(3)}{2} \sum_{h=1}^{m} b_h \\ &\quad + \frac{N_{\mathrm{W}}}{2} \sum_{h=1}^{m} \log(N_{\mathrm{T}} a_h) - \sum_{w=1}^{N_{\mathrm{W}}} \sum_{t=1}^{N_{\mathrm{T}}} \sum_{h=1}^{m} \sum_{i \in R_h} y_{i,t,w} \log(\hat{\mu}_{h,w}), \end{aligned} \tag{2.4}$$

where a_h and b_h are, respectively, the "area" (number of pixels) and "perimeter" (number of pixel edges) of region R_h, and

$$\hat{\mu}_{h,w} = \frac{1}{\sum_{t=1}^{N_{\mathrm{T}}} \Delta T_t a_h} \sum_{t=1}^{N_{\mathrm{T}}} \sum_{i \in R_h} y_{i,t,w} \tag{2.5}$$

is the maximum likelihood estimate of the Poisson rate in the corresponding region. Note that the indices of $\hat{\boldsymbol{\mu}} = \{\hat{\mu}_{hw}\}$ run over the region segments $h = 1..m$ and the passbands $w = 1..N_{\mathrm{W}}$.

For the Poisson model with change points, once the number of change points K and the locations $\boldsymbol{\tau} = \{\tau_1, ..., \tau_K\}$ are specified, for each $k \in (1, 2, ..., K+1)$, $m^{(k)}$ and $\boldsymbol{R}^{(k)}$ can be estimated independently. Using the previous argument, the MDL criterion for images within the same homogeneous interval is

$$\begin{aligned} & \mathrm{MDL}(\tau_{k-1}, \tau_k, m^{(k)}, \boldsymbol{R}^{(k)}, \hat{\boldsymbol{\mu}}^{(k)}) \\ = \quad & m^{(k)} \log(N_{\mathrm{I}}) + \frac{\log(3)}{2} \sum_{h=1}^{m^{(k)}} b_h^{(k)} + \frac{N_{\mathrm{W}}}{2} \sum_{h=1}^{m^{(k)}} \log((\tau_k - \tau_{k-1}) a_h^{(k)}) \\ & - \sum_{w=1}^{N_{\mathrm{W}}} \sum_{t=\tau_{k-1}+1}^{\tau_k} \sum_{h=1}^{m^{(k)}} \sum_{i \in R_h^{(k)}} y_{i,t,w} \log(\hat{\mu}_{h,w}^{(k)}). \end{aligned} \tag{2.6}$$

Then the overall MDL criterion for the model with change points is

$$\begin{aligned} & \mathrm{MDL}_{\mathrm{overall}}(K, \tau, \mathcal{M}, \mathcal{R}, \hat{\boldsymbol{\mu}}) \\ = \quad & K \log(N_{\mathrm{T}}) + \sum_{k=1}^{K+1} \mathrm{MDL}(\tau_{k-1}, \tau_k, m^{(k)}, \boldsymbol{R}^{(k)}, \hat{\boldsymbol{\mu}}^{(k)}). \end{aligned} \tag{2.7}$$

To sum up, using the MDL principle, the best-fit model is defined as the minimizer of the criterion (2.7). The next subsection presents a practical algorithm for carrying out this minimization.

2.2.2.5 Statistical Consistency

An important step to demonstrating the efficacy of our method is to establish its statistical consistency. That is, if it is shown that as the size of the data increases, the differences between the estimated model parameters and the true values decrease to zero, then the method can be said to be free of asymptotic bias, can be applied in the general case, and is elevated above a heuristic. We prove in Section 2.6 that the MDL-based model selection to choose the region partitioning, as well as the corresponding Poisson intensity parameters, is indeed strongly statistically consistent under mild assumptions of maintaining the temporal variability structure of $\lambda_{i,t,w}$.

2.2.3 Practical Minimization

2.2.3.1 An Iterative Algorithm

Given its complicated structure, global minimization of $\mathrm{MDL}_{\mathrm{overall}}(K, \tau, \mathcal{M}, \mathcal{R}, \hat{\mu})$ (Equation 2.7) is virtually infeasible when the number of images N_T and the number of pixels N_I are not small, because the time complexity of the exhaustive search is of order $2^{N_{\mathrm{T}} N_{\mathrm{I}}}$.

We iterate the following two steps to (approximately) minimize the MDL criterion (2.7).

1. Given a set of change points, apply the image segmentation method to all the images belonging to the first homogeneous time interval and obtain the MDL best-fitting image for this interval. Repeat this for all remaining intervals. Calculate the MDL criterion (2.7).

2. Modify the set of change points by, for example, adding or removing one change point. In terms of what modification should be made, we use the greedy strategy to select the one that achieves the largest reduction of the overall MDL value in (2.7).

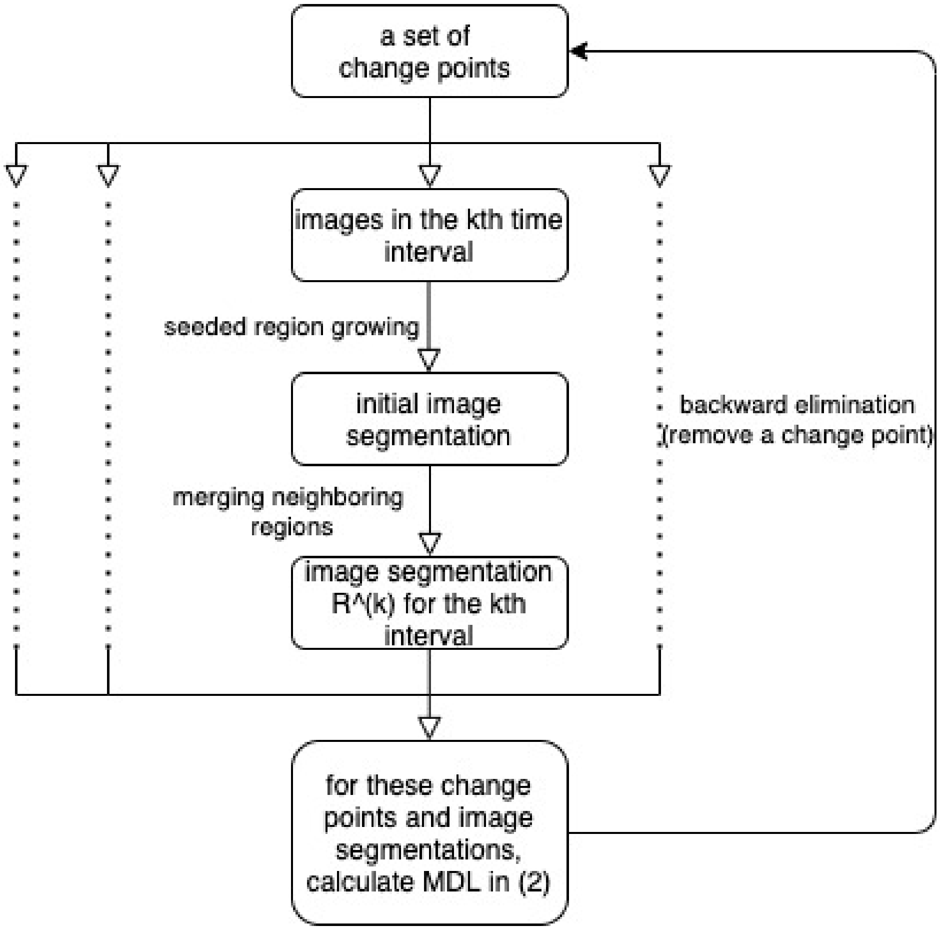

a set of change points
images in the kth time interval
seeded region growing
initial image segmentation
merging neighboring regions
image segmentation R^(k) for the kth interval
backward elimination (remove a change point)
for these change points and image segmentations, calculate MDL in (2)

Figure 2.1: Schematic illustration of the minimization algorithm

For Step 1 we begin with a large set of change points (i.e., an over-fitted model). In Step 2 we remove one change point (i.e., merge two neighboring intervals) to maximize the reduction of the MDL value. The procedure stops and declares the optimization is done if no further MDL reduction can be achieved by removing change points. This is similar to backward elimination for statistical model selection problems. See Figure 2.1 for a flowchart of the whole procedure.

2.2.3.2 Practical Considerations

Here we list some practical issues that are crucial to the success of the above minimization algorithm.

Initial seed allocation for SRG: The selection of the initial seeds in SRG plays an important role in the performance of the algorithm. To obtain a good initial oversegmentation, there must be at least one seed within each true region. Currently we use all the local maxima as well as a subset of the square lattice as the initial seeds. Based on simulation results (see Section 2.3.2), when the number of initial seeds is inadequate, the SRG will underfit the images, which will in turn lead to an overfitting of the change points and will lead to an increased false positive rate. On the other hand, it could be time-consuming when the number of initial seeds is very large, especially for high resolution images. We developed an algorithm that allocates initial seeds automatically based on locating local maxima. However, an optimal selection of initial seeds almost certainly requires expert intervention, as it depends on the type of data that we work with. To reduce the chance of obtaining a poor oversegmentation with SRG, in practice one could try using different sets of initial seeds and select the oversegmentation that gives the smallest MDL value.

Counts per bin: The photon counts in the image pixels cannot be too small, otherwise the algorithm could fail to produce meaningful output; see Section 2.3.2. In some sense, small photon counts can be seen as low signal level, which means that the proposed method requires a minimum level of signal to operate with. Therefore, care must be exercised when deciding the size for the bin. As a rule of thumb, it should be enough to have around 100 counts for each pixel belonging to an astronomical object, while pixels from the background can have very low or even zero count.

Initial change point selection: Although the stepwise greedy algorithm is capable of saving a significant amount of computation time, it could still be time consuming if the initial set of change points is too large, as it might need many iterations to reach a local minimum. It is recommended to select the initial change points based on prior knowledge, if available, in order to accelerate the algorithm.

Computation Time: In each iteration, the main time-consuming part is to apply the SRG. When the total number of pixels and the number of seeds are large, during the process, the number of candidates is large. Therefore, the comparison among all the candidates and the following updating manner lead to most of the computation burden. As an example, we found that it takes about 40 minutes for applying SRG and merging once on 64×64 images with about 200 seeds on a Linux machine with an octa-core 2.90 GHz Intel Xeon processor.

2.2.4 Highlighting the Key Pixels

After the change points are located, it is necessary to locate the pixels or regions that contribute to the estimation of the change points. The manner by which such *key pixels* are identified depends on the scientific context. Below we present two methods that are applicable to the real-world examples we discuss in Section 2.4

We focus here on images in a single passband, i.e., grey-valued images. For multi-band images, one can first transform a multi-band image into a single-band image by, for example, summing the pixel values in different bands, or by using only first principal component image of the multi-band image. Alternatively, one can also apply the method to each band individually, and merge the results from each band.

2.2.4.1 Based on Pixel Differences

The first method is to highlight key pixels based on the distribution of pixel differences before and after change points. The rationale is that a pixel with different fitted values before and after a change point is a strong indicator that it is a key pixel.

Suppose the fitted values for pixel i in time intervals k and $(k+1)$ are $\hat{\lambda}_i^{(k)}$ and $\hat{\lambda}_i^{(k+1)}$, respectively. Given the Poisson nature of the data, we first apply a square-root transformation to normalize the fitted values. Define the difference d_i for pixel i as

$$d_i = \sqrt{\hat{\lambda}_i^{(k+1)}} - \sqrt{\hat{\lambda}_i^{(k)}}. \tag{2.8}$$

A pixel is labelled as a key pixel if its d_i is far away from the mean of all the differences. To be specific, pixel i is labelled as a key pixel if

$$\left| \frac{d_i - \hat{\mu}}{\hat{\sigma}} \right| > \Phi^{-1}\left(1 - \frac{1}{2}p\right), \tag{2.9}$$

where $\hat{\mu} = \frac{1}{N_\mathrm{I}} \sum_1^{N_\mathrm{I}} d_i$ and $\hat{\sigma} = \frac{1}{\Phi^{-1}(3/4)}\mathrm{MAD}$. Here $\mathrm{MAD} = \mathrm{median}(|d_i - \tilde{d}|)$ is the median absolute deviation with $\tilde{d} = \mathrm{median}(d_i)$, and is used to obtain a robust estimate (i.e., a measure that minimizes the effect of outliers) of the standard deviation of the d_i's (see e.g., Rousseeuw and Croux, 1993). $\Phi^{-1}(\cdot)$ is the quantile of the standard normal distribution, and p is the pre-specified significance level[1]. Notice that by checking the sign of $d_i - \hat{\mu}$, we

[1] Φ^{-1} is related to the standard Normal error function, with exemplar values $\Phi^{-1}(\{0.75, 0.841, 0.977, 0.9986, 1 - \frac{10^{-5}}{2}, 1 - \frac{10^{-10}}{2}, 1 - \frac{10^{-15}}{2}\}) = \{0.6745, 1, 2, 3, 4.417, 6.467, 8.014\}$. We typically choose $p = 1 - \frac{10^{-15}}{2}$ as our threshold.

can deduce if pixel i has increased or decreased after this change point.

2.2.4.2 Based on Region Differences

Another method to locate key pixels is to compare pairs of regions. For any region in the time interval k, there must exist at least one region in time interval $(k+1)$ such that these two regions have overlapping pixels. We then test if the difference between the means of the pixels from these two regions is significant or not.

As before, we apply the square-root transformation to the pixels within each of the regions. Then we calculate the sample means $\hat{\mu}_1$ and $\hat{\mu}_2$ and sample variances $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ of these two groups of square-rooted values. Then we can for example test whether the difference between $\hat{\mu}_1$ and $\hat{\mu}_2$ is large enough with

$$\left| \frac{\hat{\mu}_2 - \hat{\mu}_1}{\sqrt{\hat{\sigma}_1^2 + \hat{\sigma}_2^2}} \right| > \Phi^{-1}\left(1 - \frac{1}{2}p\right). \tag{2.10}$$

See Section 2.4 for the applications of these two methods on some real data sets.

Lastly we note that the selection of p, the significance level, deserves a more careful consideration. As in reality one may need to do comparisons for many change points and energy bands, this becomes a multiple-testing problem where the number of tests is large. Therefore, one should adjust the value of p in order to control false positives.

2.3 Simulations

Two groups of simulations were conducted to evaluate the empirical performance of the proposed method. A specially designed $\lambda_{i,t,w}$ was used for each of the experiments. For each experiment, we tested 13 signal levels, defined as the average number of photon counts per pixel. For each signal level, 100 datasets were generated according to (2.1), with $\Delta T = 1$. The number of spectral bands $N_W = 3$ so $w = 1, 2, 3$ and the number of time points $N_T = 60$.

2.3.1 Group 1: Single Pixel

The first group of experiments were designed to evaluate the ability of the proposed method for detecting change points, under the condition that there are no spatial variations. To be more specific, the size of the images is 1×1; i.e., only one pixel. In other words, $N_{\mathrm{I}} = 1$ and the i in $\lambda_{i,t,w}$ is a dummy index. Three $\lambda_{i,t,w}$'s of increasing complexity were used:

1. $\lambda_{i,t,w}$ is constant; i.e., no change point (as depicted in Figure 2.2 (a)).

2. $\lambda_{i,t,w}$ shows intensity changes but all three bands are identical at any given t (see Figure 2.2 (b)).

3. $\lambda_{i,t,w}$ shows spectral changes (see Figure 2.2 (c)).

The first $\lambda_{i,t,w}$ was used to study the level of false positives, while the remaining two were used to study false negatives.

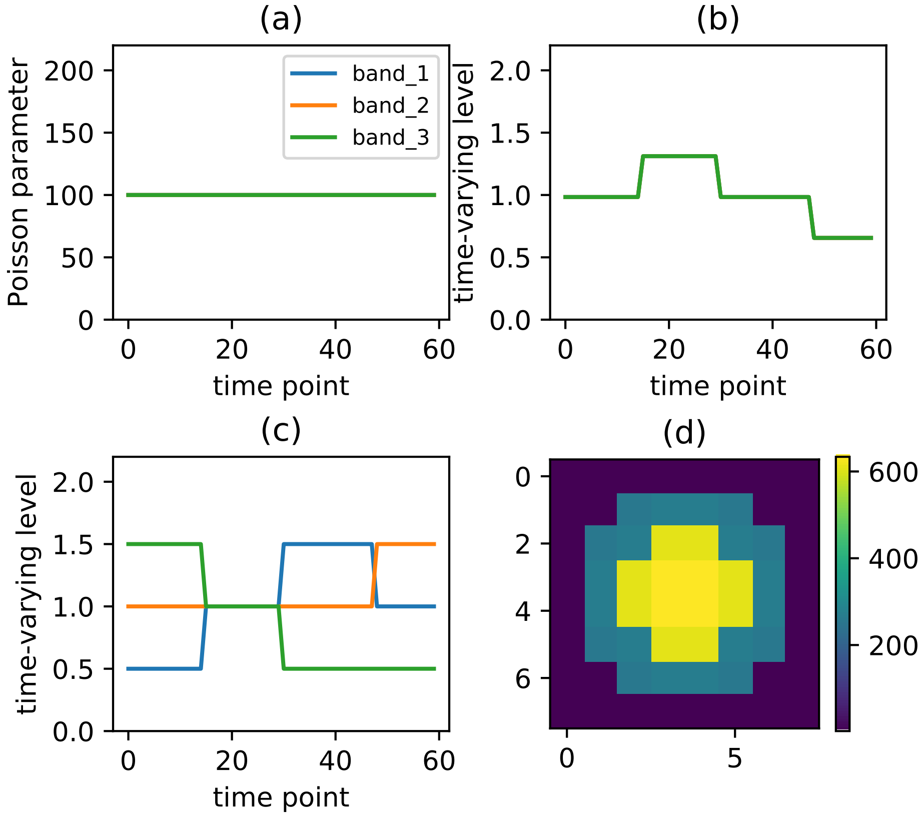


Figure 2.2: The Poisson rate functions $\lambda_{i,t,w}$ used in the simulation experiments. (a): $\lambda_{i,t,w}$ used in the single pixel experiment without change point (Section 2.3.1.1). The x-axis denotes the time points, while the y-axis shows the values of $\lambda_{i,t,w}$ for different band w. (b): the $\lambda_{i,t,w}$ relative to the no-change case (top left), used in the single pixel simulation with changing intensity (Section 2.3.1.2). (c): the $\lambda_{i,t,w}$ for different passbands (marked in blue, orange and green) relative to the no-change case (top left), used in the single pixel simulation with changing intensity and spectra (Section 2.3.1.3). (d): the spatial structure used for the second group of experiments. Size of the image is $m = n = 8$.

2.3.1.1 No Change Point

As there is no change point in $\lambda_{i,t,w}$, this experiment is ideal for studying the relationship between the false positive rate and the signal level; recall the latter is defined as the average number of photon counts per pixel.

The results of this experiment (together with the next two experiments) are summarized as the blue curves in Figure 2.3. The figure captures how well the simulation recovers the location of the change points (top left), the number of change points (top right), the excess number of change points (false positives; bottom left), and the deficit in change points (false negatives; bottom right). The top left plot reports the fraction of simulated datasets for which the set of the fitted change points $\hat{\boldsymbol{\tau}}$ is identical to the set of true change points $\boldsymbol{\tau}$. The top right plot presents the fraction of simulated datasets for which the fitted number of change points $\hat{K}$ equals to the true number of change points K. The bottom left plot shows the average false positive rate, which is defined as the average number of falsely detected change points per possible location. The bottom right plot presents the fraction of simulated datasets for which $\hat{\boldsymbol{\tau}}$ contains $\boldsymbol{\tau}$, i.e., $\boldsymbol{\tau} \subseteq \hat{\boldsymbol{\tau}}$. One can see that the false discovery rate seems to be quite stable across different signal levels. Notice that the last curve is always 1 because $\boldsymbol{\tau}$ is empty for this experiment.

2.3.1.2 Varying Intensity

In this experiment we introduced variation in $\lambda_{i,t,w}$ by multiplying (a) and (b) in Figure 2.2 together. The results are reported as the green curve in Figure 2.3. One can see that when the signal level is small ($\log_{10}$(average signal level) < 1.0), increasing the signal level leads to more false positives: this range of signals levels are too small to provide enough information for the detection of the true change points.

When $\log_{10}$(average signal level) is between 1.0 and 2.0, the proposed method starts to be able to detect the true change points as the signal level increases. Also, the false positive rate begins to drop. One possible explanation is that for any two consecutive homogeneous time intervals, the location of the change point might not be clear when the signal level is relatively small.

As the signal level continues to increase, the proposed method becomes more successful in detecting the true change point locations. For example, when the average number of counts

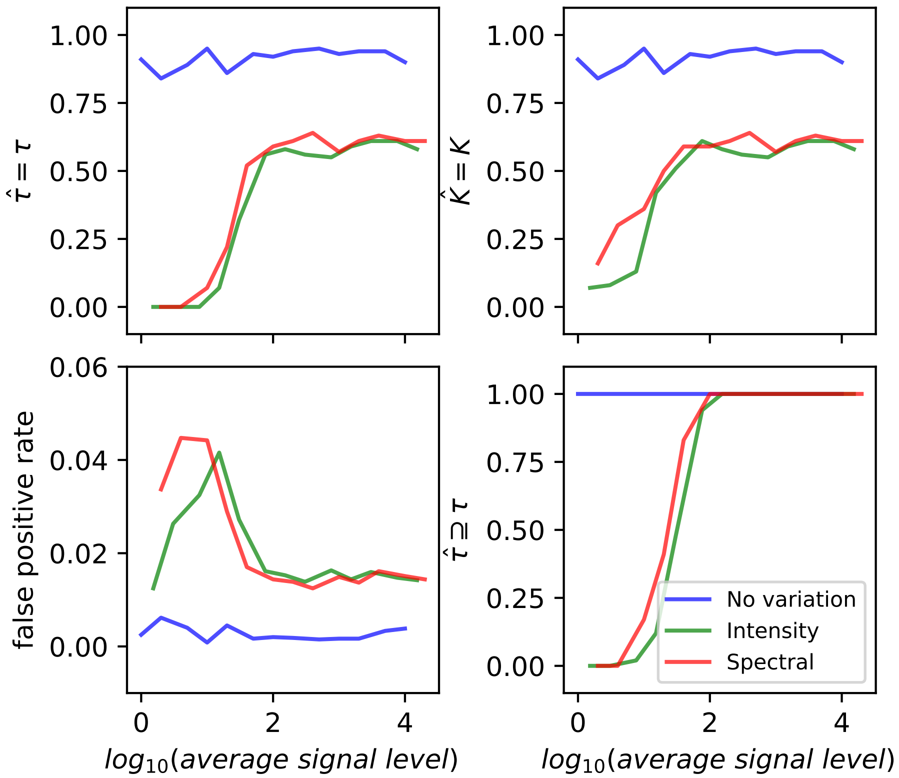


Figure 2.3: Simulation results for Group 1 experiments (single pixel). For all four plots, the x-axes denote the logarithm of the average counts of photons over all time points and all bands. *Top left:* fraction of the fitted change points $\hat{\boldsymbol{\tau}}$ that are identical to the true change points $\boldsymbol{\tau}$. *Top right:* fraction of the fitted number of change point $\hat{K}$ that equals to the true number of change point K. *Bottom left:* false positive rate. *Bottom right:* fraction of fitted change points contains true change points; i.e., $\boldsymbol{\tau} \subseteq \hat{\boldsymbol{\tau}}$. Note that the legend in the bottom right plot holds for all four plots.

in each bin is greater than 100 (i.e., $\log_{10}$(average signal level) > 2.0), the signal is strong enough so that all the true change points can be detected successfully. We note that there are always some false positives due to the Poisson randomness, and it seems that the false positive rate stabilizes as the signal level increases.

2.3.1.3 Varying Spectrum

Here we allow different bands w to change differently at the change points. The rate $\lambda_{i,t,w}$ was obtained by multiplying (a) and (c) in Figure 2.2 together. The results, which are about the same as the previous experiment (varying intensity), are reported as the red curve in Figure 2.3. When the signal level is small ($\log_{10}$(average signal level) < 1.0), the proposed method fails to detect the true change points. When $\log_{10}$(average signal level) is between

1.0 and 2.0, as the signal level increases, the false positive rate begins to decrease while the true positive rate increases. When $\log_{10}$(average signal level) > 2.0, all the true change points can be detected successfully, while the false positive rate stays at the same level as signal level increases.

2.3.2 Group 2: Spatial Structure

Instead of having a constant spatial signal (i.e., single pixel), in this second group of experiments a spatial varying structure is introduced to study the empirical performance of the proposed method. As before, three Poisson rate functions $\lambda_{i,t,w}$ are considered. The size of the image is set to $N_I = 8 \times 8$. To illustrate the importance of initial seed placement, we tested two allocation strategies: (i) we deliberately placed an inadequate number of initial seeds and (ii) we used every pixel as an initial seed.

2.3.2.1 No Change Point

There was no change point in this experiment and the spatial variation of $\lambda_{i,t,w}$ is given in the bottom right plot of Figure 2.2. The results are reported as the blue curves in Figure 2.4. One can see that if the number of initial seeds is inadequate, the false positive rate increases as the signal level increases above $\log_{10}$(average signal level) > 2.5. However, this does not happen when there are a large number of initial seeds; see the blue dotted curves in Figure 2.4. In fact, for this and the following two experiments, our method did not detect any false positive change points. This suggests that when the images are under-segmented, the method tends to place more false change points to compensate for data variability not explainable by image segmentation.

2.3.2.2 Varying Intensity

In this experiment $\lambda_{i,t,w}$ was obtained by multiplying (b) and (d) of Figure 2.2 together, so there are three change points over time. The results are similar to the no change point case, and revsummarized as the green curves in Figure 2.4.

2.3.2.3 Varying Spectrum

In this last experiment the energy bands were allowed to be different, and $\lambda_{i,t,w}$ was obtained by multiplying (c) and (d) of Figure 2.2 together. The results, reported as the red curves in Figure 2.4, are similar to the previous two experiments.

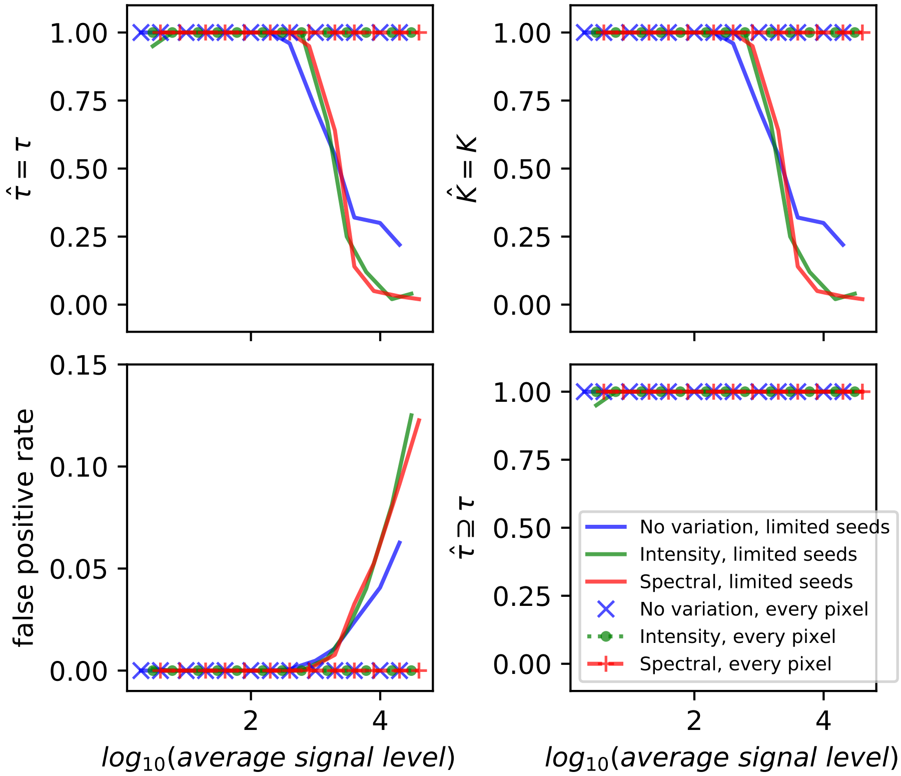


Figure 2.4: Simulation results for Group 2 experiments (with spatial structure). For all four plots, the x-axes denote the logarithm of average count of photons over all time points and all bands. Solid curves denote the results when the number of initial seeds was inadequate, while the dotted curves show the results when every pixel was assigned as a initial seed. *Top left:* fraction of the fitted change points $\hat{\boldsymbol{\tau}}$ that are identical to the true change points $\boldsymbol{\tau}$. *Top right:* fraction of the fitted number of change points $\hat{K}$ that equals to the true number of change points K. *Bottom left:* false positive rates. *Bottom right:* fraction of fitted change points contains the true change points; i.e., $\boldsymbol{\tau} \subseteq \hat{\boldsymbol{\tau}}$. The legend in the bottom right plot holds for all these four plots.

2.3.3 Empirical Conclusions

The following empirical conclusions can be drawn from the above experimental results.

- The method works well in all cases when the signal level is sufficiently large. As a rule of thumb, for binning of the original data, it would be ideal to have 100 counts or more for each bin covering an astronomical source.

- It is important to place enough initial seeds when applying SRG; otherwise the false positive rate will increase with the signal level. See the second paragraph of Section 2.2.3.2 for some practical guidelines for initial seed selection.

2.4 Applications to Real Data

To illustrate the usage in the astrophysics field, we apply the proposed method on two real datasets, which are more complicated than those in the previous section. Specifically, we select these datasets with some obvious time-evolving variations to demonstrate the performance of our method.

2.4.1 XMM-Newton Observations of Proxima Centauri

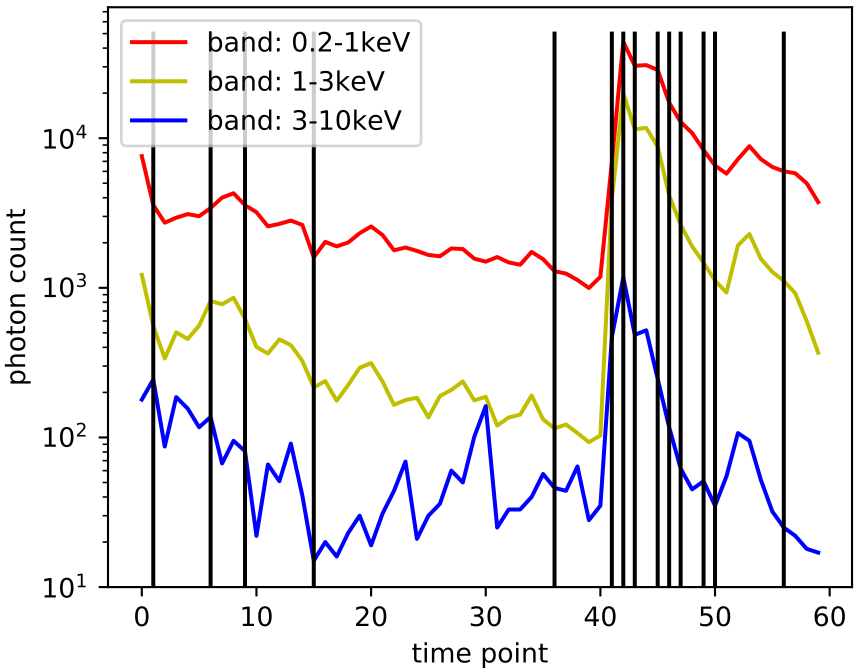


Figure 2.5: Light curves of Proxima Centauri in different bands. Each curve denotes the number of photons within the corresponding band at a given time point index. Vertical black lines denote the locations of the detected change points.

Proxima Centauri is the nearest star to the Sun and as such is well suited for studies of coronal activity. Like our Sun, Proxima Centauri operates an internal dynamo, which generates a stellar magnetic field. In the standard model for stellar dynamos, the magnetic field lines wind up through differential rotation. When some of the magnetic field lines reconnect, the energy is released in the stellar flare. Such flares typically show a sudden rise in X-ray emission and a more gradual decay over several hours. In flares, flux and temperature are correlated such that a higher X-ray flux corresponds to a higher temperature and thus a higher energy of the average detected photons (see Güdel, 2004, for a review of X-ray emission in stellar coronae and further references).

Despite its proximity, Proxima Centauri and its corona are unresolved in X-ray observations; it is just the point-spread function (PSF) of the telescope that distributes the incoming flux over many pixels on the detector.

2.4.1.1 Data

We use a dataset from XMM-Newton (Obs.ID 0049350101), where Proxima Centauri was observed for 67 ks on 2001-08-12. Because of the high flux, the MOS cameras on XMM-Newton are highly piled-up and we restrict our analysis to the data from the PN camera. We obtained the data from the XMM-Newton science archive hosted by the European Space Agency (ESA)[2]. The data we received was processed by ODS version 12.0.0. Our analysis is based on the filtered PN event data from the automated reduction pipeline (PPS). Güdel *et al.* (2002) presented a detailed analysis and interpretation of this dataset.

In our analysis, we only used a subset of photons with spatial coordinates within the region $[25500, 27500] \times [26500, 28500]$, and it was binned as images of size 64×64. We used the temporal bins of width 1100.4 seconds to generate 60 images. We binned the data into three energy bands, $(200, 1000]$, $(1000, 3000]$ and $(3000, 10000]$ in eV.

2.4.1.2 Results

Figure 2.5 presents the light curves for different bands as well as the locations of the detected change points. As there is only a single source of photons with negligible background signal level, the detected change points coincide with the changes of the light curves. Many change points are detected for the abrupt increase and then decrease in brightness for all the bands at the time points between 42 and 50. The time interval between 15 and 36 is detected as a homogeneous time interval, and the variation in light curves within this interval is viewed as common Poisson variations. A few change points are detected for the time interval before time point 15 and the interval after 50. A piecewise constant model is used to fit these gradual changes in intensities.

The fitted images can be found in Figure 2.6. The source of most photons, a point source, is modeled by different piecewise constant models at all these time intervals. The center of

[2] https://www.cosmos.esa.int/web/xmm-newton/xsa

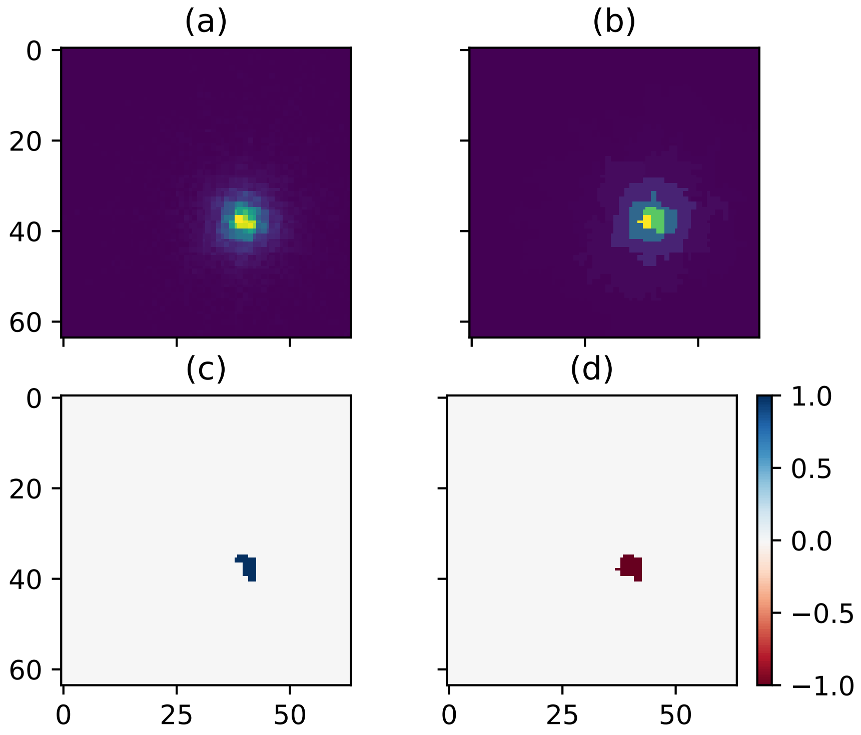


Figure 2.6: Results for Proxima Centauri. (a): the data image at time point 42 for the first band (200, 1000] in eV. (b): the corresponding fitted value $\lambda_{i,t,w}$. (c): regions that show an increase (blue) and decrease (red) in intensity prior to this time point. Compared with the previous time interval, there was a significant increase in the source at this time point. (d): as in panel *c*, but for the epoch after this time point. After this time point, the brightness in the source decreased. Notices that these two bottom plots share the colorbar, where the value 1 denotes increasing and -1 denotes decreasing intensities.

the point source and the wings of the PSF region nearby were fitted by models with shapes like concentric circles.

To test our method for finding the regions of significant change, we also apply it here because we know what the answer should be. For this dataset, as the observations for different bands change simultaneously, we combine all the three bands to highlight the key pixels. That is, we highlight the regions for each band and take the intersection of these regions. Examples of the results based on the method in Section 2.2.4.2 can be found in Figure 2.6. The method indeed picks out the point source. With significance level $p = 10^{-2}$, the abrupt increase in brightness of the source at time point 41 and 42, as well as the sudden decrease at time point 43 can be detected successfully. By modifying the significance level, different sensitivity can be achieved.

2.4.2 Isolated Evolving Solar Coronal Loop

Images of the solar corona constitute a legitimate Big Data problem. Several observatories have been collecting images in extreme ultra-violet (EUV) filters and in X-ray passbands for several decades, and analyzing them to pick out interesting changes using automated routines have been largely unsuccessful. Catalogs like the HEK (Heliophysics Events Knowledgebase Hurlburt *et al.*, 2012; Martens *et al.*, 2012) can detect and mark features of particular varieties, though these compilations remain beset by incompleteness (see, e.g., Aggarwal *et al.*, 2018; Hughes *et al.*, 2019; Barnes *et al.*, 2017). In this context, our method provides a way to model solar features without limiting it to a particular feature set to identify and locate regions in images where something interesting has transpired. As a proof of concept, we apply the method to a simple case of an isolated coronal loop filling with plasma, as observed with the Solar Dynamics Observatory's Atmospheric Imaging Assembly (SDO/AIA) filters (Pesnell *et al.*, 2012; Lemen *et al.*, 2012). Considerable enhancements must still be made in order to lower the computational cost before the method can be applied to full size images at faster than observed cadence; however, we demonstrate here that a well-defined region of interest can be selected without manual intervention for a dataset that consists of images in several filters.

2.4.2.1 Data

In particular, here we consider AIA observations carried out on 2014-Dec-11 between 19:12 UT and 19:23 UT, and focus on a 64×64 pixel region located $(+1'', -271'')$ from disk center, in which a small, isolated, well-defined loop appeared at approximately 19:19 UT. This region was selected solely as a test case to demonstrate our method; the appearance of the loop is clear and unambiguous, with no other event occurring nearby to confuse the issue; see Figure 2.7. We apply our method to these data, downloaded using the SDO AIA Cutout Service,[3] and demonstrate that the loop (and it alone) is detected and identified; see Figures 2.8, 2.9 and 2.10. AIA data are available in 6 filter bands, centered at 211, 94, 335, 193, 131, 171 Å. Here, we have limited our analysis to 3 bands: 94, 335, and 131 Å in which the isolated loop is easily discernible to the eye (a full analysis including all the filters does not change the results). Each filter consists of a sequence of 54 images, and while they are

[3] `https://www.lmsal.com/get_aia_data/`

not obtained simultaneously, the difference in time between the bands is ignorable on the timescale over which the loop evolves.

2.4.2.2 Results

The fitted images for the 3-band case can be found in Figure 2.8. Notice that there is a loop-shaped object that is of interest. Based on the fitted result, this object starts to appear at time point c.36 and becomes brighter after that for the first band. In the second band, this object appears at time point c.26 and stays bright throughout the duration considered. However in the third band, the object becomes bright at time point c.36 and vanishes soon after time point c.38. The proposed method is able to catch these changes in different bands and to detect the corresponding change points.

After detecting these change points, we find the key pixels that contribute to the change points using the methods in Section 2.2.4.1. This method is appropriate to highlight the regions that change rapidly after the change point because different bands may not change in the same direction for this dataset. Here we apply this method on a single band, 94, as an example. We use $p = 10^{-15}$. See Figure 2.9 for an illustration. We find that the method could highlight the loop-shape object which starts to appear at time point c.36, and also detect the region that becomes much brighter after time point c.38. We also compute the light curves of the intensities in a region comprised of the set of pixels formed from the union of all key pixels found at all change points in all the filters; see Figure 2.10. Notice that the event of interest is fully incorporated within the key pixels, with no spillover into the background, and the change points are post facto found to be reasonably located from a temporal perspective in that they are located where a researcher seeking to manually place them would do so. The first segment is characterized by steady emission in all three bands, the second segment shows the isolated loop beginning to form, the third segment catches the time when it reaches a peak, and the last segment tracks the slow decline in intensity.

2.5 Summary

We have developed an approach to model photon emissions by astronomical sources. Also, we propose a practical algorithm to detect the change points as well as to segment the astronomical images, based on the MDL principle for model selection. We test this method on a series of simulation experiments and apply it to two real astrophysical datasets. We are able to recover the time-evolving variations.

Based on the results of simulation experiments, it is recommended that the average number of photon counts within each bin should be from 100 to 1000 for pixels belonging to an astrophysical object, so that the proposed method is able to find change points and limit false positives.

For future work, it will be helpful to quantify the evidence of the existence of a change point by deriving a test statistic based on Monte Carlo simulations or other methods. Another possible extension is to relax the piecewise constant assumption and allow piecewise linear/quadratic modeling so that the method is able to capture more complicated and realistic patterns.

2.6 Supplement: Statistical Consistency

Here we prove that the MDL scheme is statistically consistent (see Section 2.2.2.5), thereby ensuring that the estimates of region segmentations and the Poisson intensities are reliable measures of the data. In the following, we assume that the size of the time bins $\Delta T_t = 1, \forall 1 \leq t \leq N_{\mathrm{T}}$, as N_{T} increases to infinity. That is to say, first, we study that as these underlying nonhomogeneous Poisson processes are extending at the same rate, the size of the bins keeps fixed, which leads to increasing number of independent observations for any given part of this Poisson process. And by keeping the size of the bins fixed, we get rid of the case that the Poisson parameters keep varying as N_{T} increases. Second, by setting $\Delta T_t = 1$, the Poisson parameter is numerically equal to the Poisson rate, which ease the arguments. The proof can be extended if we relax this assumption.

Given the above assumption, the photon counts have the following Poisson model,

$$y_{i,t,w} \overset{i.i.d.}{\sim} \text{Poisson}(\lambda_{i,t,w}). \tag{2.11}$$

Given change points $\boldsymbol{\tau} = (\tau_1, \tau_2, ..., \tau_K)$, we set $\tau_0 = 0$ and $\tau_{K+1} = N_{\mathrm{T}}$, and let $\nu_k = \tau_k / N_{\mathrm{T}}, k = 0, 1, ... N_{\mathrm{T}}$ to be the normalized change points. The consistency results are based on ν_k's being fixed as N_{T} increases.

The observed images within the same interval follow the same corresponding piecewise constant model given change points $\boldsymbol{\tau}$. Let $x^{(k)}_{i,t-\tau_{k-1},w} = y_{i,t,w}, \tau_{k-1} + 1 \leq t \leq \tau_k$, $\boldsymbol{X}^{(k)}_{t-\tau_{k-1}} =$

$\boldsymbol{Y}_t = \{y_{i,t,w}|i = 1, ..., N_{\mathrm{I}}, w = 1, ..., N_{\mathrm{W}}\}, \tau_{k-1} + 1 \leq t \leq \tau_k$, and $\boldsymbol{X}^{(k)} = \{\boldsymbol{X}_t^{(k)}|1 \leq t \leq T_k\}$ where $T_k = \tau_k - \tau_{k-1}$.

Let $\lambda_{i,t,w}$'s within the k^{th} interval follow the same corresponding two-dimensional piecewise constant model with $m^{(k)}$ constant regions. The partition of the images within interval k is specified by a region assignment function $R^{(k)}(.)\colon \{1, ..., N_{\mathrm{I}}\} \to \{1, ..., m^{(k)}\}$. The Poisson parameters $\mu_{h,w}^{(k)}$ is the value for the w^{th} band in the h^{th} region of the k^{th} interval. Let $\mu^{(k)} = \{\mu_{h,w}^{(k)}|h = 1, ..., m^{(k)}, w = 1, ..., N_{\mathrm{W}}\}$. That is to say,

$$\lambda_{i,t,w} = \sum_{k=1}^{K+1} I_{\{t \in (\tau_{k-1}, \tau_k]\}} \mu_{R^{(k)}(i),w}^{(k)}. \tag{2.12}$$

In all, for pixel $i \in R_h^{(k)}$,

$$x_{i,t,w}^{(k)} \overset{i.i.d.}{\sim} \text{Poisson}(\mu_{h,w}^{(k)}). \tag{2.13}$$

For each $1 \leq t \leq T_k$, the log-likelihood function for regions assignment $R^{(k)}$ and Poisson parameters $\mu^{(k)}$ is

$$\tilde{l}_k((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)}) = \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^{(k)}} \sum_{i,s.t.R^{(k)}(i)=h} [x_{i,t,w}^{(k)} \log(\mu_{h,w}^{(k)}) - \mu_{h,w}^{(k)} - \log(x_{i,t,w}^{(k)}!)]. \tag{2.14}$$

As some of the terms in the log-likelihood function have nothing to do with the parameters to estimate, we remove these terms and write down the log-likelihood to be

$$l_k((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)}) = \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^{(k)}} \sum_{i,s.t.R^{(k)}(i)=h} [x_{i,t,w}^{(k)} \log(\mu_{h,w}^{(k)}) - \mu_{h,w}^{(k)}]. \tag{2.15}$$

Define $\psi_k = (R^{(k)}, \mu^{(k)})$ to be the parameter set for the k^{th} interval, and $\boldsymbol{\mathcal{M}}$ to be the class of models ψ_k can take value from. Then the log-likelihood for the k^{th} interval can be written as

$$L_T^{(k)}(\psi_k; \boldsymbol{X}^{(k)}) = \sum_{t=1}^{T_k} l_k((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)}). \tag{2.16}$$

Let $\boldsymbol{\nu} = (\nu_1, ..., \nu_K)$ be the normalized change point location vector, and let $\boldsymbol{\psi} = (\psi_1, ..., \psi_{K+1})$ be the parameter vector. Then vector $(K, \boldsymbol{\nu}, \boldsymbol{\psi})$ can specify a model for this sequence of images. The MDL is derived to be

$$\begin{aligned}\mathrm{MDL}(K,\boldsymbol{\nu},\boldsymbol{\psi}) &= K\log(N_{\mathrm{T}})+\sum_{k=1}^{K+1}[m^{(k)}\log(N_{\mathrm{I}})+\frac{\log(3)}{2}\sum_{h=1}^{m^{(k)}}b_h^{(k)}\\ &\quad+\frac{N_{\mathrm{W}}}{2}\sum_{h=1}^{m^{(k)}}\log(([T\nu_k]-[T\nu_{k-1}]+1)a_h^{(k)})]\\ &\quad-\sum_{k=1}^{K+1}L_T^{(k)}(\psi_k;\boldsymbol{X}^{(k)}). \qquad (2.17)\end{aligned}$$

Here the "area" (number of pixels) and "perimeter" (number of pixel edges) of region $R_h^{(k)}$ are denoted by $a_h^{(k)}$ and $b_h^{(k)}$.

In order to make sure that the change points are identifiable, we assume that there exists a $\epsilon_\nu > 0$ such that $\min_{1\le k\le K+1}|\nu_k-\nu_{k-1}| > \epsilon_\nu$. Therefore, the number of change points is bounded by $K \le [1/\epsilon_\nu]+1$. And there exists a constraint $A_{\epsilon_\nu}^K$ of $\boldsymbol{\nu}$ where

$$A_{\epsilon_\nu}^K=\{\boldsymbol{\nu}\in(0,1)^K|0<\nu_1<\ldots<\nu_K<1,\nu_k-\nu_{k-1}>\epsilon_\nu,\forall 1\le k\le K+1\}. \qquad (2.18)$$

Then the estimation of the model based on MDL is given by

$$(\hat{K}_T,\hat{\boldsymbol{\nu}}_T,\hat{\boldsymbol{\psi}}_T)=\arg\min_{K\le[1/\epsilon_\nu]+1,\boldsymbol{\nu}\in A_{\epsilon_\nu}^K,\boldsymbol{\psi}\in\boldsymbol{\mathcal{M}}}\frac{1}{N_{\mathrm{T}}}\mathrm{MDL}(K,\boldsymbol{\nu},\boldsymbol{\psi}). \qquad (2.19)$$

Here $\mathrm{MDL}(K,\boldsymbol{\nu},\boldsymbol{\psi})$ is defined in (2.17), $\hat{\boldsymbol{\nu}}_T=(\hat{\nu}_1,\ldots,\hat{\nu}_{\hat{K}})$ and $\hat{\boldsymbol{\psi}}_T=(\hat{\psi}_1,\ldots,\hat{\psi}_{\hat{K}+1})$, where $\hat{\psi}_k=(\hat{R}^{(k)},\hat{\mu}^{(k)})$. And $\hat{\mu}^{(k)}$ is defined as

$$\hat{\mu}^{(k)}=\arg\max_{\mu^{(k)}\in\Theta_k(\hat{R}^{(k)})}L_T^{(k)}((\hat{R}^{(k)},\mu^{(k)});\hat{\boldsymbol{X}}^{(k)}) \qquad (2.20)$$

with $\hat{\boldsymbol{X}}^{(k)}=\{\boldsymbol{Y}_t|[T\hat{\nu}_{k-1}]<t\le[T\hat{\nu}_k]\}$ denotes the estimated k^{th} interval of the sequence of images.

We further define the log-likelihood formed by a portion of the observations in the k^{th} interval by

$$L_T^{(k)}(\psi_k,\nu_d,\nu_u;\boldsymbol{X}^{(k)})=\sum_{t=[T_k\nu_d]+1}^{[T_k\nu_u]}l_k((R^{(k)},\mu^{(k)});\boldsymbol{X}_t^{(k)}), \qquad (2.21)$$

where $0\le\nu_d<\nu_u\le1$ and $\nu_u-\nu_d>\epsilon_\nu$.

We denote

$$\sup_{\nu_d,\nu_u} := \sup_{0\le\nu_d<\nu_u\le 1,\nu_u-\nu_d>\epsilon_\nu} \tag{2.22}$$

to simplify the notation.

In this setting, an extension need to be made such that ν_d and ν_u can be slightly outside $[0,1]$. It means that the k^{th} estimated interval could cover a part of the observations that belong to the $(k-1)^{\text{th}}$ and $(k+1)^{\text{th}}$ true intervals. Based on the formula (3.4) in Davis and Yau (2013), for a real-value function $f_T(\nu_d,\nu_u)$ on $\mathcal{R}^2$,

$$\sup_{\underline{\nu_d},\overline{\nu_u}} f_T(\nu_d,\nu_u) \xrightarrow{a.s.} 0 \tag{2.23}$$

is used to denote

$$\sup_{-h_T<\nu_d<\nu_u<1+r_T,\nu_u-\nu_d>\epsilon_\nu} f_T(\nu_d,\nu_u) \xrightarrow{a.s.} 0 \tag{2.24}$$

for any pre-specified positive-valued sequences h_T and r_T, which cover to 0 as $N_{\text{T}}\to\infty$.

The following assumptions on true Poisson parameters $\mu^{o(k)}_{h,w}, 1\le h\le m^{(k)}, 1\le w\le N_{\text{W}}, 1\le k\le (K^o+1)$ are necessary.

Assumption 2.1.

$$0 < C_d := \min_{k,h,w}\mu^{o(k)}_{h,w} \le C_u := \max_{k,h,w}\mu^{o(k)}_{h,w} < \infty. \tag{2.25}$$

Assumption 2.2. *For two true neighboring regions $R^{(k)}_p$ and $R^{(k)}_q$ at the k^{th} interval,*

$$\delta_1 := \min_{k,p,q,w} |\mu^{o(k)}_{p,w} - \mu^{o(k)}_{q,w}| > 0. \tag{2.26}$$

Assumption 2.3. *For any two neighboring intervals $(k-1)$ and k*

$$\delta_2 := \min_{k}\max_{i,w} |\mu^{o(k)}_{R^{(k)}(i),w} - \mu^{o(k-1)}_{R^{(k-1)}(i),w}| > 0. \tag{2.27}$$

Proposition 2.1 (v)**.** *For $k=1,...,K+1$ and any fixed $R^{(k)}$, there exists a $\epsilon>0$ such that,*

$$\begin{aligned}
\sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} & E|l_k((R^{(k)},\mu^{(k)});\boldsymbol{X}^{(k)}_t)|^{v+\epsilon} < \infty,\\
\sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} & E|l'_k((R^{(k)},\mu^{(k)});\boldsymbol{X}^{(k)}_t)|^{v+\epsilon} < \infty,\\
\sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} & E|l''_k((R^{(k)},\mu^{(k)});\boldsymbol{X}^{(k)}_t)| < \infty.
\end{aligned} \tag{2.28}$$

This proposition holds for $v = 1, 2, 4$ due to the compactness of parameter space (Assumption 2.1) and bounded $E[(x_{i,t,w}^{(k)})^{v+\epsilon}]$.

Proposition 2.2. *For $k = 1, ..., K+1$ and any fixed $R^{(k)}$,*

$$\begin{aligned}
\sup_{\mu^{(k)} \in \Theta_k(R^{(k)})} |\frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T^{(k)}((R^{(k)}, \mu^{(k)}); \boldsymbol{X}^{(k)}) - L_k((R^{(k)}, \mu^{(k)}))| &\xrightarrow{a.s.} 0, \\
\sup_{\mu^{(k)} \in \Theta_k(R^{(k)})} |\frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T'^{(k)}((R^{(k)}, \mu^{(k)}); \boldsymbol{X}^{(k)}) - L_k'((R^{(k)}, \mu^{(k)}))| &\xrightarrow{a.s.} 0, \qquad (2.29)\\
\sup_{\mu^{(k)} \in \Theta_k(R^{(k)})} |\frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T''^{(k)}((R^{(k)}, \mu^{(k)}); \boldsymbol{X}^{(k)}) - L_k''((R^{(k)}, \mu^{(k)}))| &\xrightarrow{a.s.} 0,
\end{aligned}$$

where

$$\begin{aligned}
L_k((R^{(k)}, \mu^{(k)})) &\coloneqq E(l_k((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)})), \\
L_k'((R^{(k)}, \mu^{(k)})) &\coloneqq E(l_k'((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)})), \qquad (2.30)\\
L_k''((R^{(k)}, \mu^{(k)})) &\coloneqq E(l_k''((R^{(k)}, \mu^{(k)}); \boldsymbol{X}_t^{(k)})).
\end{aligned}$$

The estimated locations of change points are used to define the likelihood in practice. Therefore, the two ends of the k^{th} interval might contain observations from the $(k-1)^{\text{th}}$ and $(k+1)^{\text{th}}$ true intervals, though the estimated change points are close to the true change points. It is necessary to control the effect at the two ends of the fitted interval.

Proposition 2.3 (w)**.** *For $k = 1, ..., K+1$ and any fixed ψ and any sequence of integers $\{g(N_{\mathrm{T}})\}_{N_{\mathrm{T}} \geq 1}$ that satisfies $g(N_{\mathrm{T}}) > cN_{\mathrm{T}}^w$ for some $c > 0$ when N_{T} is large enough, then*

$$\begin{aligned}
\frac{1}{g(N_{\mathrm{T}})} \sum_{t=N_{\mathrm{T}}-g(N_{\mathrm{T}})+1}^{N_{\mathrm{T}}} l_k(\psi; \boldsymbol{X}_t^{(k)}) &\xrightarrow{a.s.} E(l_k(\psi; \boldsymbol{X}_t^{(k)})), \\
\frac{1}{g(N_{\mathrm{T}})} \sum_{t=N_{\mathrm{T}}-g(N_{\mathrm{T}})+1}^{N_{\mathrm{T}}} l_k'(\psi; \boldsymbol{X}_t^{(k)}) &\xrightarrow{a.s.} E(l_k'(\psi; \boldsymbol{X}_t^{(k)})).
\end{aligned} \qquad (2.31)$$

Based on Lemma 1 in Davis and Yau (2013), Proposition 2.3 holds when Proposition 2.1(2) holds and the Assumption 4* in Davis and Yau (2013) is satisfied. And Assumption 4* is satisfied because an independent process, like the current setting, must be mixing.

It is necessary to discuss the identifiability of models in $\boldsymbol{\mathcal{M}}$. First we define R^b an oversegmentation compared with R^s if $R^b(i) = R^b(j)$ leads to $R^s(i) = R^s(j)$.

Proposition 2.4. *For the k^{th} interval, the true model $\psi_k^o \in \boldsymbol{\mathcal{M}}$ satisfies the condition $\psi_k^o = \arg\max_{\psi \in \boldsymbol{\mathcal{M}}} E(l_k(\psi; \boldsymbol{X}_t^{(k)}))$. Also, ψ_k^o is uniquely identifiable, which means that if there exists a μ^* such that $l_k((R^o, \mu^o); \boldsymbol{X}_t^{(k)}) = l_k((R^o, \mu^*); \boldsymbol{X}_t^{(k)})$ almost everywhere for $\boldsymbol{X}_t^{(k)}$, then $\mu^o = \mu^*$. And suppose there exists another model $\psi_k^b = (R^b, \mu^b)$ such that $l_k(\psi_k^b; \boldsymbol{X}_t^{(k)}) = l_k(\psi_k^o; \boldsymbol{X}_t^{(k)})$ almost everywhere, then R^b must be an oversegmentation compared with R^o. And μ^b satisfies $\mu^b_{R^b(i),w} = \mu^o_{R^o(i),w}, \forall i, w$.*

Proof. Suppose on the contrary there exist a model $\psi^* = (R^*, \mu^*)$ that satisfies $\psi^* = \arg\max_{\psi \in \boldsymbol{\mathcal{M}}} E(l_k(\psi; \boldsymbol{X}_t^{(k)})))$, and ψ^* is neither the true model nor an oversegmentation of the true model. Then there exist two pixels i_0 and j_0, such that they are neighboring pixels, $R^{o(k)}(i_0) \neq R^{o(k)}(j_0)$ and $R^*(i_0) = R^*(j_0)$. Therefore, by Assumption 2.2, we have

$$|\mu^{o(k)}_{R^{o(k)}(i_0),w} - \mu^{o(k)}_{R^{o(k)}(j_0),w}| \geq \delta_1 > 0. \tag{2.32}$$

Define

$$\bar{\mu}_{h,w}(R) := \frac{1}{a_h(R)} \sum_{i,R(i)=h} \mu^{o(k)}_{R^o(i),w}, \tag{2.33}$$

where $a_h(R)$ denotes the number of pixels in region h given segmentation R. And in a special case,

$$\bar{\mu}_{h,w}(R^{o(k)}) = \mu^{o(k)}_{h,w}. \tag{2.34}$$

Then for all possible $\mu^* \in \Theta_k(R^*)$, we have

$$\begin{aligned} E(l_k((R^*, \mu^*); \boldsymbol{X}_t^{(k)})) &= E(\sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^*} \sum_{i,s.t.R^*(i)=h} [x^{(k)}_{i,t,w} \log(\mu^*_{h,w}) - \mu^*_{h,w}]) \\ &= \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^*} \sum_{i,s.t.R^*(i)=h} [\mu^{o(k)}_{R^{o(k)}(i),w} \log(\mu^*_{h,w}) - \mu^*_{h,w}] \\ &= \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^*} a_h(R^*)[\bar{\mu}_{h,w}(R) \log(\mu^*_{h,w}) - \mu^*_{h,w}] \\ &\leq \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^*} a_h(R^*) \max_{\mu_{h,w}} [\bar{\mu}_{h,w}(R) \log(\mu_{h,w}) - \mu_{h,w}] \\ &= \sum_{w=1}^{N_{\mathrm{W}}} \sum_{h=1}^{m^*} a_h(R^*)[\bar{\mu}_{h,w}(R^*) \log(\bar{\mu}_{h,w}(R^*)) - \bar{\mu}_{h,w}(R^*)] \\ &= E(l_k((R^*, \bar{\mu}(R^*)); \boldsymbol{X}_t^{(k)})). \end{aligned} \tag{2.35}$$

Also, we have

$$\begin{aligned}
E(l_k((R^*,\bar{\mu}(R^*));\boldsymbol{X}_t^{(k)})) =&\sum_{w=1}^{N_{\mathrm{W}}}\sum_{h=1}^{m^*} a_h(R^*)\max_{\mu_{h,w}}[\bar{\mu}_{h,w}(R^*)\log(\mu_{h,w})-\mu_{h,w}] \\
\leq&\sum_{w=1}^{N_{\mathrm{W}}}\sum_{i=1,i\notin\{i_0,j_0\}}^{N_{\mathrm{I}}}\max_{\lambda_{i,w}}[\mu^{o(k)}_{R^{o(k)}(i),w}\log(\lambda_{i,w})-\lambda_{i,w}] \\
&+\sum_{w=1}^{N_{\mathrm{W}}}[\mu^{o(k)}_{R^{o(k)}(i_0),w}\log(\mu^*_{R^*(i_0),w})-\mu^*_{R^*(i_0),w} \\
&+\mu^{o(k)}_{R^{o(k)}(j_0),w}\log(\mu^*_{R^*(j_0),w})-\mu^*_{R^*(j_0),w}] \\
<&\sum_{w=1}^{N_{\mathrm{W}}}\sum_{i=1,i\notin\{i_0,j_0\}}^{N_{\mathrm{I}}}\max_{\lambda_{i,w}}[\mu^{o(k)}_{R^{o(k)}(i),w}\log(\lambda_{i,w})-\lambda_{i,w}] \\
&+\sum_{w=1}^{N_{\mathrm{W}}}\max_{\lambda_{i_0,w}}[\mu^{o(k)}_{R^{o(k)}(i_0),w}\log(\lambda_{i_0,w})-\lambda_{i_0,w}] \\
&+\sum_{w=1}^{N_{\mathrm{W}}}\max_{\lambda_{j_0,w}}[\mu^{o(k)}_{R^{o(k)}(j_0),w}\log(\lambda_{j_0,w})-\lambda_{j_0,w}] \\
=&\sum_{w=1}^{N_{\mathrm{W}}}\sum_{i=1}^{N_{\mathrm{I}}}[\mu^{o(k)}_{R^{o(k)}(i),w}\log(\mu^{o(k)}_{R^{o(k)}(i),w})-\mu^{o(k)}_{R^{o(k)}(i),w}] \\
=&\sum_{w=1}^{N_{\mathrm{W}}}\sum_{h=1}^{m^{o(k)}} a_h^{o(k)}[\mu^{o(k)}_{R^{o(k)}(i),w}\log(\mu^{o(k)}_{R^{o(k)}(i),w})-\mu^{o(k)}_{R^{o(k)}(i),w}] \\
=&E(l_k((R^{o(k)},\mu^{o(k)});\boldsymbol{X}_t^{(k)})).
\end{aligned} \tag{2.36}$$

Here the strict inequities must hold because of (2.32)

Finally combining (2.35) and (2.36), we have

$$E(l_k((R^*,\mu^*);\boldsymbol{X}_t^{(k)}))<E(l_k((R^{o(k)},\mu^{o(k)});\boldsymbol{X}_t^{(k)})), \tag{2.37}$$

which is a contradiction. This finishes the proof. □

Lemma 2.1. *For any fixed* $R^{(k)}$,

$$\begin{aligned}
&\sup_{\underline{\nu_d},\overline{\nu_u}} \sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} \Big|\frac{1}{N_{\mathrm{T}}(\nu_k-\nu_{k-1})} L_T^{(k)}((R^{(k)},\mu^{(k)}),\nu_d,\nu_u;\boldsymbol{X}^{(k)}) - (\nu_u-\nu_d)L_k((R^{(k)},\mu^{(k)}))\Big| \\
&\qquad \xrightarrow{a.s.} 0, \\
&\sup_{\underline{\nu_d},\overline{\nu_u}} \sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} \Big|\frac{1}{N_{\mathrm{T}}(\nu_k-\nu_{k-1})} L_T'^{(k)}((R^{(k)},\mu^{(k)}),\nu_d,\nu_u;\boldsymbol{X}^{(k)}) - (\nu_u-\nu_d)L_k'((R^{(k)},\mu^{(k)}))\Big| \\
&\qquad \xrightarrow{a.s.} 0, \\
&\sup_{\underline{\nu_d},\overline{\nu_u}} \sup_{\mu^{(k)}\in\Theta_k(R^{(k)})} \Big|\frac{1}{N_{\mathrm{T}}(\nu_k-\nu_{k-1})} L_T''^{(k)}((R^{(k)},\mu^{(k)}),\nu_d,\nu_u;\boldsymbol{X}^{(k)}) - (\nu_u-\nu_d)L_k''((R^{(k)},\mu^{(k)}))\Big| \\
&\qquad \xrightarrow{a.s.} 0.
\end{aligned} \tag{2.38}$$

See Proposition 1 and 2 in Davis and Yau (2013) for the proof.

Lemma 2.2. *Suppose the true parameters for interval* k *is* $\psi^{o(k)} = (R^{o(k)},\mu^{o(k)})$. *And suppose a region segmentation* $R^{(k)}$ *is specified for estimation. Let*

$$\begin{aligned}
\hat\mu_T = \hat\mu_T^{(k)}(\nu_d,\nu_u) &:= \arg\max_{\mu^{(k)}\in\Theta_k(R^{(k)})} L_T^{(k)}((R^{(k)},\mu^{(k)}),\nu_d,\nu_u;\boldsymbol{X}_k), \\
\mu^{*(k)} &:= \arg\max_{\mu^{(k)}\in\Theta_k(R^{(k)})} L_k((R^{(k)},\mu^{(k)})).
\end{aligned} \tag{2.39}$$

Then

$$\sup_{\underline{\nu_d},\overline{\nu_u}} \Big|\frac{1}{N_{\mathrm{T}}(\nu_k-\nu_{k-1})} L_T^{(k)}((R^{(k)},\hat\mu_T),\nu_d,\nu_u;\boldsymbol{X}^{(k)}) - (\nu_u-\nu_d)L_k((R^{(k)},\mu^{*(k)}))\Big| \xrightarrow{a.s.} 0, \tag{2.40}$$

where the supremum is defined in (2.23). *And if* $R^{(k)} = R^{o(k)}$, *we further have*

$$\sup_{\underline{\nu_d},\overline{\nu_u}} |\hat\mu_T^{(k)}(\nu_d,\nu_u) - \mu^{o(k)}| \xrightarrow{a.s.} 0. \tag{2.41}$$

If $R^{(k)}$ *is an oversegmentation than* $R^{o(k)}$, *then we have*

$$\sup_{\underline{\nu_d},\overline{\nu_u}} |\hat\mu_{T,R^{(k)}(i),w}(\nu_d,\nu_u) - \mu^{o(k)}_{R^{o(k)}(i),w}| \xrightarrow{a.s.} 0 \ \forall i, w. \tag{2.42}$$

Proof.

$$
\begin{aligned}
&(\nu_u - \nu_d)(L_k((R^{(k)}, \mu^{*(k)})) - L_k((R^{(k)}, \hat{\mu}_T))) \\
\leq &\sup_{\underline{\nu_d}, \overline{\nu_u}} |(\nu_u - \nu_d)L_k((R^{(k)}, \mu^{*(k)})) - \frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T^{(k)}((R^{(k)}, \mu^{*(k)}), \nu_d, \nu_u; \boldsymbol{X}^{(k)}) \\
&+ \frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T^{(k)}((R^{(k)}, \hat{\mu}_T), \nu_d, \nu_u; \boldsymbol{X}^{(k)}) - (\nu_u - \nu_d)L_k((R^{(k)}, \hat{\mu}_T))| \\
\leq &2 \sup_{\underline{\nu_d}, \overline{\nu_u}} \sup_{\mu^{(k)} \in \Theta_k(R^{(k)})} |\frac{1}{N_{\mathrm{T}}(\nu_k - \nu_{k-1})} L_T^{(k)}((R^{(k)}, \hat{\mu}_T), \nu_d, \nu_u; \boldsymbol{X}^{(k)}) - (\nu_u - \nu_d)L_k((R^{(k)}, \mu^{(k)}))| \\
\xrightarrow{a.s.} &0.
\end{aligned}
\tag{2.43}
$$

The first inequity is obtained by the definition of maximum likelihood estimator, and the last convergence comes from Lemma 2.1. As $\mu^{*(k)}$ maximizes $L_k((R^{(k)}, \mu^{(k)}))$ and $\nu_u - \nu_d > 0$, we have

$$
|L_k((R^{(k)}, \mu^{*(k)})) - L_k((R^{(k)}, \hat{\mu}_T))| \xrightarrow{a.s.} 0. \tag{2.44}
$$

Combining (2.43), (2.44) and Proposition 2.1(1), 2.40 holds. If $R^{(k)} = R^{o(k)}$, by Proposition 2.4, $L_k((R^{(k)}, \mu^{(k)}))$ has a unique maximizer at $\mu^{o(k)}$, so (2.41) holds. If $R^{(k)}$ is an oversegmentation compared with $R^{o(k)}$, by Proposition 2.4, (2.42) holds. □

Now we give a preliminary result of the convergence when the number of change points is known.

Theorem 2.1. *(Theorem 1 in Davis and Yau (2013)) Let $\{\mathbf{Y}_t | t = 1, ..., N_{\mathrm{T}}\}$ be the observed images specified by $(K^o, \boldsymbol{\nu}^o, \boldsymbol{\psi}^o)$. And suppose the number of change points K^o is known. The change points and parameters are estimated by*

$$
(\hat{\boldsymbol{\nu}}_T, \hat{\boldsymbol{\psi}}_T) = arg \min_{\boldsymbol{\lambda} \in A_{\epsilon_\lambda}^m, \boldsymbol{\psi} \in \boldsymbol{\mathcal{M}}} \frac{1}{N_{\mathrm{T}}} MDL(K^o, \boldsymbol{\nu}, \boldsymbol{\psi}). \tag{2.45}
$$

Then $\hat{\boldsymbol{\nu}}_T \xrightarrow{a.s.} \boldsymbol{\nu}^o$ and for each interval, the estimated $\hat{R}^{(k)}$ must be an oversegmentation comparing to the true region segmentation.

We skip the proof of this theorem because it is quite similar to the proof of Theorem 1 in Davis and Yau (2013). Notice that we need to use Assumption 2.3 in the proof.

Corollary 2.1. *(Corollary 1 in Davis and Yau (2013)) Under the conditions of Theorem 2.1, if the number of change- points is unknown and is estimated from the data , then*

1. *The number of change points cannot be underestimated. That is to say, $\hat{K} \geq K^o$ almost surely when N_{T} is large enough.*
2. *When $\hat{K} > K^o$, $\boldsymbol{\nu}^o$ must be a subset of the limit of $\hat{\boldsymbol{\nu}}_T$ for large enough N_{T}.*
3. *In each fitted interval, the region segmentation must be equal to or be an oversegmentation comparing with the corresponding true region segmentation.*

See Corollary 1 in Davis and Yau (2013) for more details.

Theorem 2.2. *(Theorem 2 in Davis and Yau (2013)) Let $\boldsymbol{\nu}^o = (\nu_1^o, \nu_2^o, ..., \nu_{m^o}^o)$ be the true change points. And $(\hat{K}, \hat{\boldsymbol{\nu}}_T, \hat{\boldsymbol{\psi}}_T)$ is the MDL-based result. Then $\forall k = 1, 2, ..., K^o$, there exists a $\hat{\nu}_{t_k} \in \hat{\boldsymbol{\nu}}_T$ where $1 \leq t_k \leq \hat{K}$ such that*

$$|\hat{\nu}_{t_k} - \nu_k^o| = o(N_{\mathrm{T}}^{-\frac{1}{2}}) \ a.s. \ . \tag{2.46}$$

See the proof of Theorem 2 in Davis and Yau (2013).

Lemma 2.3. *Suppose the true region segmentation $R^{o(k)}$ is specified for the k^{th} interval, then*

$$\hat{\mu}_T^{(k)}(\hat{\nu}_{k-1}, \hat{\nu}_k) - \mu^{o(k)} = O(\sqrt{\frac{\log\log(N_{\mathrm{T}})}{N_{\mathrm{T}}}}) \ a.s. \ . \tag{2.47}$$

When the specific region segmentation $R^{(k)}$ is an oversegmentation compared with $R^{o(k)}$, then we have

$$\hat{\mu}_{T,R^{(k)}(i),w}^{(k)}(\hat{\nu}_{k-1}, \hat{\nu}_k) - \mu_{R^{o(k)}(i),w}^{o(k)} = O(\sqrt{\frac{\log\log(N_{\mathrm{T}})}{N_{\mathrm{T}}}}) \ \ a.s. \ \ \forall i, w. \tag{2.48}$$

See Lemma 2 in Davis and Yau (2013) for more details.

Then we come to the main result.

Theorem 2.3. *Let $\{\boldsymbol{Y}_t | t = 1, ..., N_{\mathrm{T}}\}$ be the observed images specified by $(K^o, \boldsymbol{\nu}^o, \boldsymbol{\psi}^o)$. The estimator $(\hat{K}_T, \hat{\boldsymbol{\nu}}_T, \hat{\boldsymbol{\psi}}_T)$ is defined by (2.19). Then we have*

$$\begin{aligned} \hat{K}_T &\xrightarrow{a.s.} K^o, \\ \hat{\boldsymbol{\nu}}_T &\xrightarrow{a.s.} \boldsymbol{\nu}^o, \\ \hat{\boldsymbol{\psi}}_T &\xrightarrow{a.s.} \boldsymbol{\psi}^o. \end{aligned} \tag{2.49}$$

See Theorem 3 in Davis and Yau (2013) for more details.

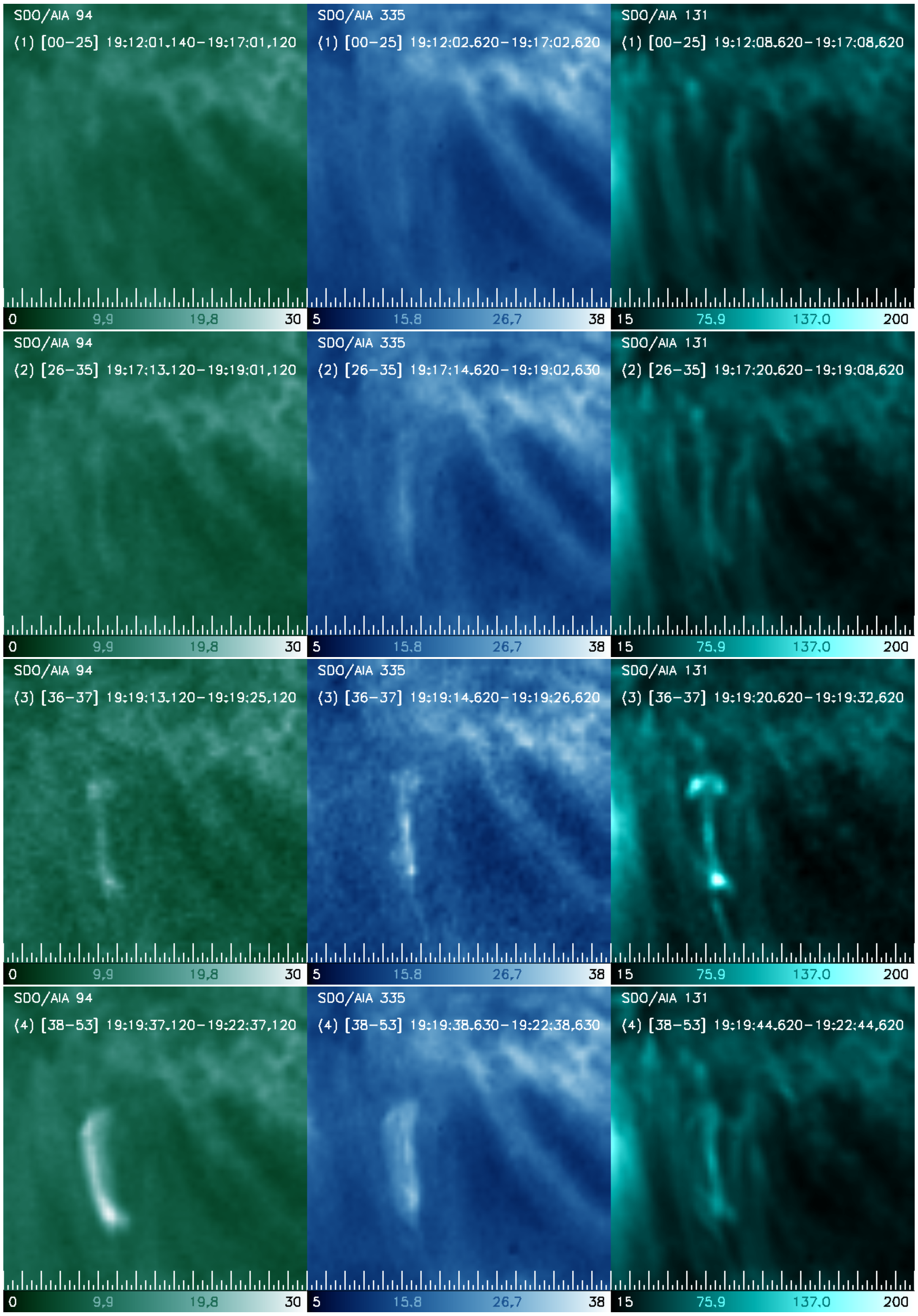


Figure 2.7: An isolated loop structure shown lighting up in 3 SDO/AIA passbands. Each row corresponds to the intensities in AIA filter images, averaged over the time duration found by our method, going from interval 1 (top) to interval 4 (bottom). The columns, going from left to right, show the 94, 335, and 131 Å filter band images. The filter name, time duration, and the image sequence indices are marked at the top of the image and the intensity scale is marked at the bottom. The grid at the bottom of each image denotes the pixelation, with each image having a size of 64×64 pixels. Notice that the isolated loop becomes bright enough for detection in the 3$^{\text{rd}}$ interval.

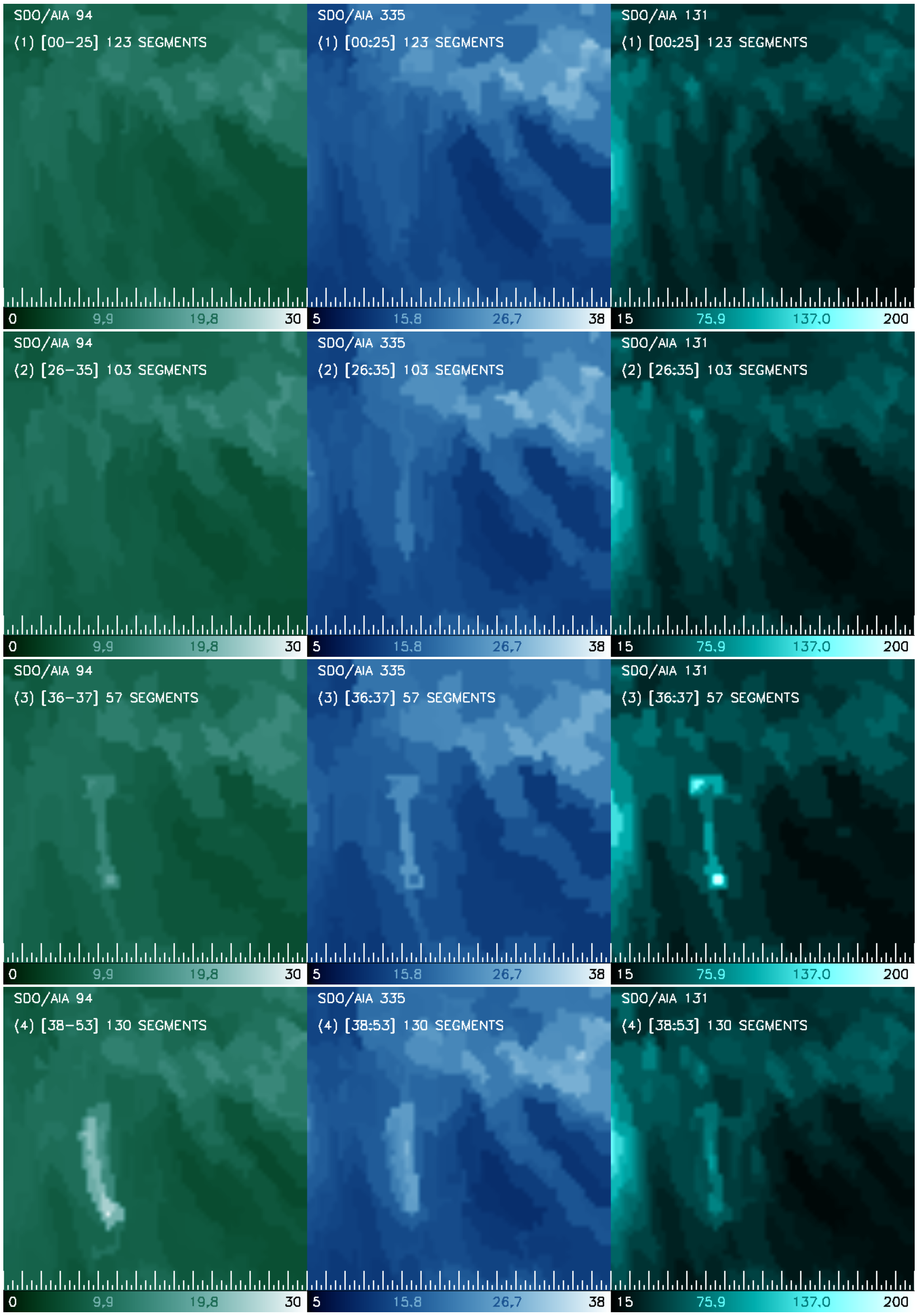


Figure 2.8: Intensities $\lambda_{i,t,w}$ as fit to the data from Figure 2.7 in spatial segments. The images are arranged in the same manner, and demonstrate that the loop structure is locatable and identifiable. The number of region segments found are also marked.

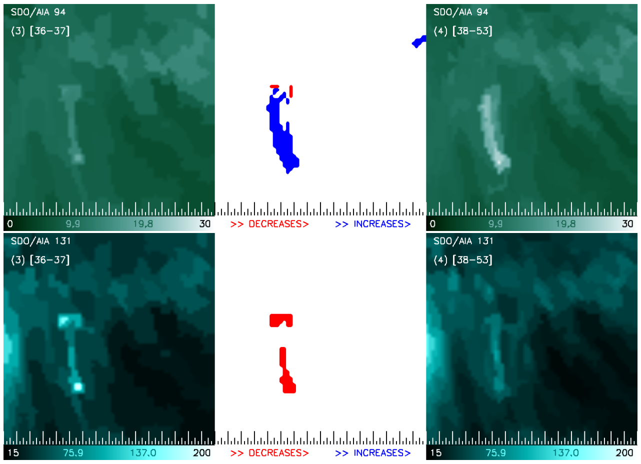


Figure 2.9: Demonstrating the isolation of key pixels of interest. Each set of three shows the fitted intensity in one passband in the 3rd interval (*left*), followed by a bitmap of pixels (*middle*) showing where intensity increases (blue) and decreases (red), followed by the fitted intensity image in the same filter in the 4th time interval (*right*). The upper row shows the transition in the 94 Å filter, and the lower row shows the transition in the 131 Å filter. Notice that the loop continues to brighten at 94 Å, even as it starts to fade at 131 Å.

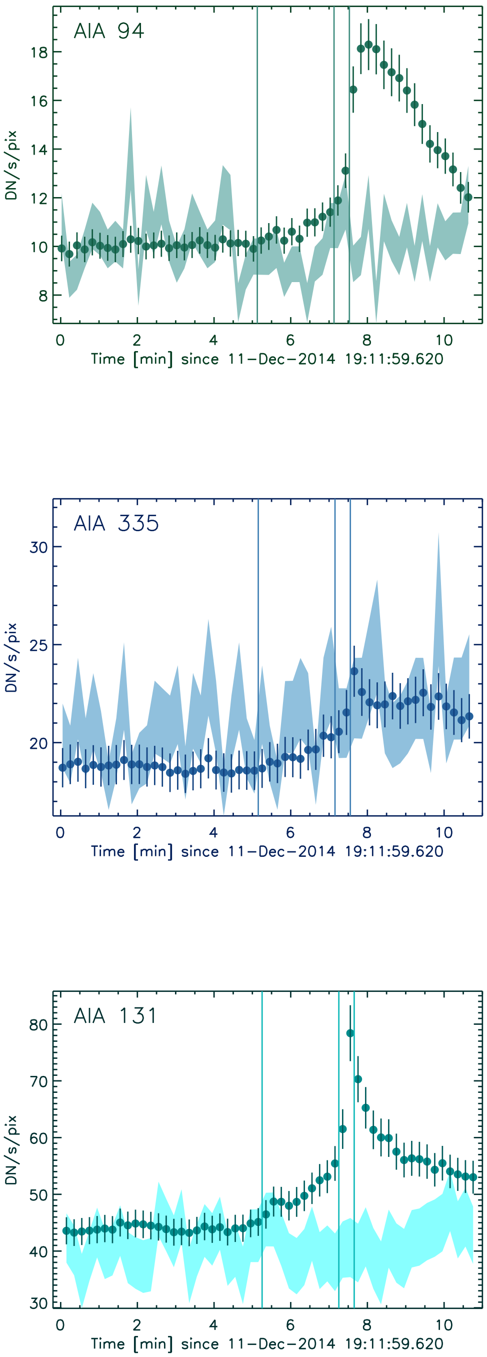


Figure 2.10: Light curves of the key pixels where changes are found, for the three filters used in the analysis: 94 Å (*top*), 335 Å (*middle*), and 131 Å (*bottom*). The average of the observed intensities, weighted by the number of times each pixel is flagged as a key pixel, are shown as dots, along with the similarly weighted sample standard deviation as vertical bars. The shaded regions represent the envelope of the sample standard deviation seen *outside* the flagged pixels. The vertical lines denote the change points found by our algorithm.

Chapter 3

Consistent Change Point Detection and Node Clustering for Time Series of Graphs

Suppose an undirected graph is observed over time. Its structure (i.e., nodes and edges) remains the same but the measurements taken at the nodes may vary over time. This paper proposes a method that simultaneously performs the following two tasks: (i) it detects change points in the time domain, and (ii) for each time interval between any two consecutive detected change points, it partitions the nodes into different clusters of similar measurements. The method begins with recasting the problem into a model selection problem, and employs the minimum description length principle to construct a selection criterion for which the best fitting model is defined as its minimizer. A practical algorithm is developed to (approximately) locate this minimizer. It is shown that the model selection criterion leads to statistically consistent estimates, while numerical experiments show that the method enjoys promising empirical properties. To the best of the authors' knowledge, the proposed method is one of the first that performs simultaneous change point detection and node clustering for time series of graphs.

3.1 Introduction

Consider the following situation. Suppose we would like to study criminal activities in a certain region over time. We could first partition the region into different administrative districts, where each district is represented by a node in a graph. Two nodes are connected if their corresponding districts share a common border. For each node weekly measurements are taken over a time period. These measurements can be the weekly total numbers of reported crime incidents in the district, or they can be the numbers of a certain crime incidents such as burglary. With this set up, we can model the crime measurements as a time-evolving graph, and see if the crime activities change over time, or if they are spatially correlated in the sense that adjacent districts have similar patterns.

This problem can be formalized as follows. Suppose a time-evolving graph is observed at time $t = 1, \ldots, T$. The number of nodes and the node connectivity (i.e., edges) remain unchanged over time, although the noisy measurements observed at the nodes may change. We use p to denote the number of nodes, $n_{t,i}$ to denote the number of measurements observed in the ith node at time t, and $x_{t,i,j}$ to denote the jth measurement (i.e., $j = 1, \ldots, n_{t,i}$) of the ith node at time t, where $i = 1, \ldots, p$ and $t = 1, \ldots, T$. The values of the $n_{t,i}$'s are typically small, and could even be zero for some t, i; i.e., no measurement. For all $\{t, i, j\}$, we model the measurements $x_{t,i,j}$ as

$$x_{t,i,j} \overset{i.i.d.}{\sim} \mathcal{N}(\beta_{t,i}, \sigma^2),$$

where $\beta_{t,i}$ is the true signal value for the ith node at time t, and σ^2 is the noise variance. The goal is to, given the data $x_{t,i,j}$, estimate the signal $\beta_{t,i}$. Of course, an unbiased estimator for $\beta_{t,i}$ is $\sum_{j=1}^{n_{t,i}} x_{t,i,j}/n_{t,i}$ (if $n_{t,i} > 0$), the sample average. However, this estimator is of high variance if $n_{t,i}$ is small, which is quite common for many real data problems where it is typical to have $n_{t,i} = 1$ for some $\{t, i\}$. Thus, we impose two additional assumptions to the problem so that improved estimates for $\beta_{t,i}$ can be obtained.

First we assume that the underlying signal is temporally smooth. Specifically, we assume that there exists a sequence of M time points $1 < t_1 < \ldots < t_M \leq T$, called change points, such that *all* the signal $\beta_{t,i}$'s are the same between any two consecutive change points. Write $\boldsymbol{\beta}_t = (\beta_{t,1}, \beta_{t,2}, ..., \beta_{t,p})^\top$, $t_0 = 1$ and $t_{M+1} = T + 1$. This assumption implies that $\boldsymbol{\beta}_s = \boldsymbol{\beta}_t$ if $t_k \leq s, t < t_{k+1}$ for all $k = 0, \ldots, M$.

In addition to temporal smoothness, we also assume the signal is "spatially" smooth, in the sense that two nodes that are connected by an edge tend to have more similar values of $\beta_{t,i}$ than nodes that are not. We formalize this idea by assuming that, at any time point

t, the nodes can be partitioned into different connected subgraphs in such a way that all the nodes within the same cluster share the same signal value. In below we shall call these subgraphs *clusters*. In other words, if at time t the ith and lth nodes are in the same cluster, then $\beta_{t,i} = \beta_{t,l}$. Note that the clusters may change at the change points $t_1, \ldots, t_M$.

It is straightforward to estimate the underlying signal $\beta_{t,i}$'s if the change points and the cluster structure are known; it will simply be the average of the relevant $x_{t,i,j}$'s; see (3.17) below. In this paper, however, we do not assume the change points nor the cluster structure are known, and we will estimate them as well as the $\beta_{t,i}$'s. We first recast this problem as a model selection problem and invoke the minimum description length (MDL) principle (Rissanen, 1989a, 2007b) to select a best fitting model as our final answer. As a model selection criterion, MDL defines the best model as the model that compresses the data into the shortest code length for storage. Among different versions of MDL, we shall use the so-called "two-part" variant which has been shown to produce excellent results in other model selection problems, such as image segmentation (Lee, 2000) and structural break detection (Davis *et al.*, 2006).

To the best of the authors' knowledge, the current paper is one of the first that considers the problem of simultaneous change point detection and node clustering for time series of graphs, although various authors have considered other different but similar problems. For example, Cheung *et al.* (2020) used the MDL principle for change point detection and community detection in time series of networks. Notice that the focus of their work is to model the edge behavior of the networks and no theoretical results are provided. Sharpnack *et al.* (2013) derived a so-called Spectral Scan Statistic to test whether the signal over a given graph is constant, or is piecewise constant over two subgraphs. Lastly, a commonly studied problem is change point detection for time-varying Gaussian graphical models. A popular approach is to impose different kinds of l_1 type penalties to encourage sparsity and smoothness across time so that the entries of the precision matrix are piecewise constant or slow varying over time; e.g., see Kolar and Xing (2012); Gibberd and Nelson (2017); Hallac *et al.* (2017) and Yang and Peng (2020).

3.2 Methodology

To make the presentation more digestible, we begin with deriving in Section 3.2.1 the MDL solution for the case when there is no change point; i.e., the homogeneous case. Then in Section 3.2.2 we will extend to the general case that allows for change points.

3.2.1 Homogeneous Case

This subsection assumes the cluster structure stays the same across different times. That is, there is no change point and $\beta_{t,i} = \beta_i$ for all $\{t, i\}$. The task is to estimate the cluster structure, which includes the number of clusters as well as the cluster membership for each node; i.e., which cluster the node belongs to. Let d be the number of clusters (so $1 \leq d \leq p$) and write the cluster membership for the ith node as c_i; i.e., the ith node belongs to the c_ith cluster, where $1 \leq c_i \leq d$. Let $\boldsymbol{c} = \{c_1, c_2, ..., c_p\}$ and $\mathcal{P} = \{\beta_1, \beta_2, ..., \beta_p\}$. For the homogeneous case the goal is to estimate d, $\boldsymbol{c}$ and $\mathcal{P}$.

As mentioned before, MDL defines the best fitting model as the one that enables the best compression of the data, or in other words, the one that produces the shortest code length of the data. This idea can be formalized as follows. If we write $\mathrm{CL}(z)$ as the code length of z, then the code length CL("data") of the observed data can be decomposed into two parts, a model $\mathcal{F}$ plus the corresponding residuals $\hat{\mathcal{E}}$:

$$\mathrm{CL}(\text{“data”}) = \mathrm{CL}(\mathcal{F}) + \mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F}), \tag{3.1}$$

and the best model is the one that minimizes CL("data"). Here $\mathcal{F} = \{d, \boldsymbol{c}, \mathcal{P}\}$ and note that the dependence of $\hat{\mathcal{E}}$ on $\mathcal{F}$ is stressed in the notation of the last term.

To minimize (3.1) we need computable expression for CL("data") and we begin by calculating $\mathrm{CL}(\mathcal{F})$, which can be further decomposed into

$$\mathrm{CL}(\mathcal{F}) = \mathrm{CL}(d) + \mathrm{CL}(\boldsymbol{c}) + \mathrm{CL}(\mathcal{P}). \tag{3.2}$$

According to Rissanen (1989a), it takes approximately $\log_2(I)$ bits to encode an integer I with upper bound unknown, and approximately $\log_2(I_u)$ bits with a known upper bound I_u. To encode the number of clusters d with an upper bound p, the smaller one, $\log_2(d)$ is used. So

$$\mathrm{CL}(d) = \log_2(d). \tag{3.3}$$

For $\boldsymbol{c}$, it takes $\log_2(d)$ bits to encode each c_i. Then we have

$$\mathrm{CL}(\boldsymbol{c}) = \sum_{i=1}^{p} \log_2(d) = p\log_2(d). \tag{3.4}$$

Next we calculate $\mathrm{CL}(\mathcal{P})$, and we need the maximum likelihood estimate (MLE) of the β_i's. If the ith node belongs to the rth cluster (i.e., $c_i = r$), the MLE of β_i is

$$\hat{\beta}_i = \frac{\sum_{t=1}^{T}\sum_{q,c_q=r}\sum_{j=1}^{n_{t,q}} x_{t,q,j}}{\sum_{t=1}^{T}\sum_{q,c_q=r} n_{t,q}}, \tag{3.5}$$

which is simply the average of all the observations of all the nodes belonging to the rth cluster at all time. By Rissanen (1989a), to encode an MLE, the code length is $\frac{1}{2}\log_2 N$ if N observations are used for the estimation. For $\hat{\beta}_i$, this number is given by the denominator of (3.5), and hence

$$\mathrm{CL}(\mathcal{P}) = \sum_{r=1}^{d} \frac{1}{2}\log_2(\sum_{t=1}^{T}\sum_{i,c_i=r} n_{t,i}). \tag{3.6}$$

Notice that although there are p of the $\hat{\beta}_i$'s, there are only d distinct values of them, as there are only $d \le p$ clusters around. Therefore the upper limit of the first summation in (3.6) is d not p.

Now substitute (3.3), (3.4) and (3.6) into (3.2), and replace all the $\log_2(.)$ with $\log(.)$, (because all the terms have logarithm to the base 2) we have

$$\mathrm{CL}(\mathcal{F}) = (p+1)\log_2(d) + \sum_{r=1}^{d}\frac{1}{2}\log_2(\sum_{t=1}^{T}\sum_{i,c_i=r} n_{t,i}). \tag{3.7}$$

Lastly we calculate the last term $\mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F})$ of (3.1), which, according to Rissanen (1989a), is given by the negative log (base 2) of the likelihood of the fitted model. With the Gaussianity assumption $x_{t,i,j} \sim \mathcal{N}(\beta_i, \sigma^2)$ for all $\{t,i,j\}$, the negative log-likelihood (natural logarithm) is

$$\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_{t=1}^{T}\sum_{i=1}^{p}\sum_{j=1}^{n_{t,i}}(x_{t,i,j} - \beta_i)^2, \tag{3.8}$$

where $n = \sum_{t=1}^{T}\sum_{i=1}^{p} n_{t,i}$ is the total number of observations. The MLE $\hat{\beta}_i$ for β_i is given by (3.5), while the MLE for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{r=1}^{d}\mathrm{SSE}_r,$$

where $\text{SSE}_r = \sum_{t=1}^{T} \sum_{i,c_i=r} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}_i)^2$ is the sum of squared errors of the rth cluster.

Plugging these MLEs $\hat{\beta}_i$ and $\hat{\sigma}^2$ into (3.8), we obtain the code length of the residuals $\hat{\mathcal{E}}$

$$\text{CL}(\hat{\mathcal{E}}|\mathcal{F}) = \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\frac{1}{n}\sum_{r=1}^{d}\text{SSE}_r) + \frac{n}{2}, \tag{3.9}$$

and from (3.1), (3.7) and (3.9), the overall code length is

$$\begin{aligned}&\text{CL}(\text{“data”}) = \text{CL}(\mathcal{F}) + \text{CL}(\hat{\mathcal{E}}|\mathcal{F})\\ =&(p+1)\log(d) + \sum_{r=1}^{d}\frac{1}{2}\log(\sum_{t=1}^{T}\sum_{i,c_i=r} n_{t,i}) + \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\frac{1}{n}\sum_{r=1}^{d}\text{SSE}_r) + \frac{n}{2}.\end{aligned}$$

Ignoring constant terms we arrive at the following MDL criterion for the homogeneous case, and the best fitting model is defined as its minimizer:

$$(p+1)\log(d) + \sum_{r=1}^{d}\frac{1}{2}\log(\sum_{t=1}^{T}\sum_{i,c_i=r} n_{t,i}) + \frac{n}{2}\log(\frac{1}{n}\sum_{r=1}^{d}\text{SSE}_r). \tag{3.10}$$

3.2.2 Heterogeneous Case

This subsection considers the heterogeneous case where the cluster structure and the signal values are allowed to change at change points. The number M and the locations $\mathcal{T} = \{t_1, t_2, ..., t_M\}$ of such change points are unknown and need to be estimated, and we will continue to use MDL. With M change points, the time line is partitioned into $M+1$ intervals, where the mth interval is $[t_{m-1}, t_m)$ for $m = 1, \ldots, M+1$. We write the number of clusters in the mth interval as $d^{(m)}$ and the cluster membership as $\boldsymbol{c}^{(m)} = \{c_1^{(m)}, c_2^{(m)}, ..., c_p^{(m)}\}$; i.e., in the mth interval the ith node belongs to the $c_i^{(m)}$th cluster. We write $\mathcal{C} = \{\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, ..., \boldsymbol{c}^{(M+1)}\}$ and $\mathcal{P} = \{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_T\}$, and hence the model is $\mathcal{F} = \{\mathcal{T}, \mathcal{C}, \mathcal{P}\}$, which leads to the code length decomposition:

$$\text{CL}(\mathcal{F}) = \text{CL}(\mathcal{T}) + \text{CL}(\mathcal{C}) + \text{CL}(\mathcal{P}). \tag{3.11}$$

To encode $\mathcal{T}$, we first need to encode the number of the change points and then the actual locations of the change points. As there are M change points, the code length is $\log(M+1)$, where the additional 1 is used to distinguish $M = 0$ and $M = 1$. The locations of change points $\mathcal{T}$ can be encoded by using the length of the time intervals $(t_m - t_{m-1})$'s. Therefore combining the two we have

$$\text{CL}(\mathcal{T}) = \log(M+1) + \sum_{m=1}^{M}\log(t_m - t_{m-1}). \tag{3.12}$$

Once $\mathcal{T}$ is encoded, it becomes the homogeneous case for each time interval. Using similar arguments as before, we have

$$\mathrm{CL}(\mathcal{C}) = \sum_{m=1}^{M+1}(p+1)\log(d^{(m)}) \tag{3.13}$$

and

$$\mathrm{CL}(\mathcal{P}) = \sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\frac{1}{2}\log(\sum_{t=t_{m-1}}^{t_m-1}\sum_{i,c_i^{(m)}=r} n_{t,i}). \tag{3.14}$$

Combining (3.11) to (3.14), we have

$$\begin{aligned}\mathrm{CL}(\mathcal{F}) =& \log(M+1) + \sum_{m=1}^{M}\log(t_m - t_{m-1}) + \sum_{m=1}^{M+1}(p+1)\log(d^{(m)}) \\ &+ \sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\frac{1}{2}\log(\sum_{t=t_{m-1}}^{t_m-1}\sum_{i,c_i^{(m)}=r} n_{t,i}).\end{aligned} \tag{3.15}$$

Similarly, the code length of the residuals $\hat{\mathcal{E}}$ is

$$\mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F}) = \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\mathrm{SSE}_r^{(m)}) + \frac{n}{2}, \tag{3.16}$$

where

$$\mathrm{SSE}_r^{(m)} = \sum_{t=t_{m-1}}^{t_m-1}\sum_{i,c_i^{(m)}=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j} - \hat{\beta}_{t,i})^2$$

with $\hat{\beta}_{t,i}$ being the MLE of $\beta_{t,i}$

$$\hat{\beta}_{t,i} = \frac{\sum_{s=t_{m-1}}^{t_m-1}\sum_{q,c_q^{(m)}=r}\sum_{j=1}^{n_{s,q}} x_{s,q,j}}{\sum_{s=t_{m-1}}^{t_m-1}\sum_{q,c_q^{(m)}=r} n_{s,q}}. \tag{3.17}$$

Now adding (3.15) and (3.16) together and omitting constant terms, the MDL criterion for the heterogeneous case is

$$\begin{aligned}\mathrm{MDL}(\mathcal{T},\mathcal{C}) =& \log(M+1) + \sum_{m=1}^{M}\log(t_m - t_{m-1}) + \sum_{m=1}^{M+1}(p+1)\log(d^{(m)}) \\ &+ \sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\frac{1}{2}\log(\sum_{t=t_{m-1}}^{t_m-1}\sum_{i,c_i^{(m)}=r} n_{t,i}) + \frac{n}{2}\log(\frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\mathrm{SSE}_r^{(m)}).\end{aligned} \tag{3.18}$$

Note that in the notation of the above MDL criterion, $\mathcal{P}$ is dropped from its argument list. It is because once $\mathcal{T}$ and $\mathcal{C}$ are specified, $\mathcal{P}$ can be uniquely estimated by (3.17). Note also that when there is no change point, $\mathrm{MDL}(\mathcal{T},\mathcal{C})$ reduces to (3.10).

To sum up, we propose to estimate the change points $\mathcal{T}$ and the cluster structures $\mathcal{C}$ (as well as the signal $\mathcal{P}$) as the minimizer of MDL($\mathcal{T},\mathcal{C}$):

$$\{\hat{\mathcal{T}},\hat{\mathcal{C}}\} = \arg\min_{\mathcal{T},\mathcal{C}} \frac{1}{n}\text{MDL}(\mathcal{T},\mathcal{C}). \tag{3.19}$$

3.2.3 Theoretical Properties

This subsection establishes the statistical consistency of the MDL solution $\{\hat{\mathcal{T}},\hat{\mathcal{C}}\}$ defined by (3.19). The proofs of the following results can be found in Section 3.7.1.

We need the following regularity conditions. First, for all $N > 0$, it is assumed that there exist an $N_0 > 0$ such that whenever $n > N_0$,

$$n_{t,i} > N \quad \text{for all} \quad 1 \le i \le p, 1 \le t \le T. \tag{3.20}$$

This condition guarantees that the numbers of observations in all node $n_{t,i}$'s go to infinity when the total number of observations n goes to infinity. Second, it is assumed that

$$\lim_{n\to\infty} \frac{n_{t,i}}{n} = \gamma_{t,i} \quad \text{for all} \quad 1 \le i \le p, 1 \le t \le T, \tag{3.21}$$

where the $\gamma_{t,i}$'s are some non-negative constants that sum to one. This condition ensures that the numbers of observations $n_{t,i}$'s for the nodes grow at the same linear rate.

We also assume the conditions that were listed in the beginning of Section 3.2 for the change points and signal. We denote the true model as $\{\mathcal{T}^0,\mathcal{C}^0\}$:

$$\mathcal{T}^0 = (t_1^0, t_2^0, ..., t_{M^0}^0) \quad \text{and} \quad \mathcal{C}^0 = \{\boldsymbol{c}^{0(1)}, \boldsymbol{c}^{0(2)}, ..., \boldsymbol{c}^{0(M+1)}\},$$

where $\boldsymbol{c}^{0(m)} = \{c_1^{0(m)}, c_2^{0(m)}, ..., c_p^{0(m)}\}$. We have the following lemma.

Lemma 3.1. *Suppose the total number of clusters $\sum_{m=1}^{M+1} d^{(m)}$ is known. Under the model assumptions and Conditions (3.20) and (3.21), the MDL criterion (3.19) gives*

$$\hat{\mathcal{T}} \to \mathcal{T}^0 \textit{ a.s.} \quad \textit{and} \quad \hat{\mathcal{C}} \to \mathcal{C}^0 \textit{ a.s.}$$

Lemma 3.1 is based on the assumption that the total number of clusters is known, which can be unrealistic for many real data problems. This assumption can be relaxed.

Theorem 3.1. *Assume the conditions of Lemma 3.1 with the exception that the total number of clusters $\sum_{m=1}^{M+1} d^{(m)}$ is unknown. The MDL criterion (3.19) gives*

$$\hat{\mathcal{T}} \to \mathcal{T}^0 \textit{ a.s.} \quad \textit{and} \quad \hat{\mathcal{C}} \to \mathcal{C}^0 \textit{ a.s.}$$

3.3 Practical Minimization of MDL($\mathcal{T}$,$\mathcal{C}$)

Even for moderate sizes of p and T, direct minimization of (3.18) is by no means a trivial task. This section develops a practical procedure to tackle this task. The idea is to first construct a function that can be used to generate a set of good candidate models relatively quick, and then select the final model from these candidate models as the one that gives the smallest value of MDL($\mathcal{T}$,$\mathcal{C}$). We shall call such a function a *candidate model generating function*. The idea is similiar to, in the context of variable selection in linear models, first apply lasso to quickly generate a set of candidate models on its solution path, and then use a model selection criterion such as BIC to select the best model from these candidate models.

We need some notation to proceed. Let $y_{t,i}$ be the average of all the observations within the ith node at time t; i.e., $y_{t,i} = \bar{x}_{t,i} = \frac{1}{n_{t,i}} \sum_{j=1}^{n_{t,i}} x_{t,i,j}$, and write $\boldsymbol{Y}_t = (y_{t,1}, y_{t,2}, ..., y_{t,p})^\top$ for $t = 1, ..., T$, and $\boldsymbol{Y} = (\boldsymbol{Y}_1^\top, \boldsymbol{Y}_2^\top, \dots, \boldsymbol{Y}_T^\top)^\top$; hence $\boldsymbol{Y}$ is a vector of length $p \times T$. Let $\boldsymbol{n} = (n_{1,1}, \dots, n_{1,p}, n_{2,1}, \dots n_{2,p}, \dots, n_{T,1}, \dots, n_{T,p})^\top$ be the vector of the numbers of observations for the nodes, and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \dots, \boldsymbol{\beta}_T^\top)^\top$ be the vector of the true signal, where $\boldsymbol{\beta}_t = (\beta_{t,1}, \beta_{t,2}, ..., \beta_{t,p})^\top$ for $t = 1, ..., T$. Then the goal is to retrieve the underlying signal $\boldsymbol{\beta}$ from its noisy version $\boldsymbol{Y}$

$$y_{t,i} = \beta_{t,i} + e_{t,i}, \quad e_{t,i} \overset{i.i.d.}{\sim} \mathcal{N}(0, \frac{\sigma^2}{n_{t,i}})$$

under the assumptions of temporal and spatial smoothness.

3.3.1 Construction of A Candidate Model Generating Function

The goal of a candidate model generating (CMG) function is to quickly generate a set of good candidate models with small values of MDL($\mathcal{T}$,$\mathcal{C}$). Thus for the current problem a good CMG function should produce models that are both temporally and spatially smooth, and yet maintain good data fidelity. One naturally way to construct such a function is to combine three terms together: a penalty term that encourages temporal smoothness, a second penalty term that encourages spatial smoothness, and lastly a loss term that measure data fidelity.

We begin with the temporal smoothness assumption, which prefers signals close in time to have similar values (except at the change points); i.e., $\boldsymbol{\beta}_{t+1} \approx \boldsymbol{\beta}_t$. This suggests the following penalty term

$$\Omega_1(\boldsymbol{\beta}) = \lambda_1 \sum_{t=1}^{T-1} \|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|_2, \tag{3.22}$$

where λ_1 is a tuning parameter and $\|.\|_2$ is the vector l_2-norm. This is in similar spirit as the penalty used in the fused lasso of Tibshirani *et al.* (2005) and the generalized total variation denoising method of Bleakley and Vert (2011).

For the spatial smoothness assumption, we borrow the idea from graph-guided-fused-lasso (Chen *et al.*, 2010; Kim *et al.*, 2009) to construct the penalty term. First, let $\boldsymbol{E}$ be the set of all connected edges in the graph:

$$\boldsymbol{E} = \{(i,k) : \text{the } i\text{th and } k\text{th nodes are connected}, \ 1 \leq i,k \leq p\}.$$

Recall that the node connectivitiy of our graph is assumed constant over time, so $\boldsymbol{E}$ does not change over time. Next, define a matrix $\boldsymbol{G}$ in such way that if $(i,k) \in \boldsymbol{E}$, then one row of $\boldsymbol{G}$ is all zeros except the ith entry is 1 and the kth entry is -1. Note that $\boldsymbol{G}$ is of size $|\boldsymbol{E}| \times p$, and is not unique as its rows can be permuted, but it will not affect the final results. Here we suggest using the following penalty term for spatial smoothness:

$$\Omega_2(\boldsymbol{\beta}) = \lambda_2 \sum_{t=1}^{T} \|\boldsymbol{G}\boldsymbol{\beta}_t\|_1, \tag{3.23}$$

where λ_2 is a tuning parameter and $\|.\|_1$ is the vector l_1-norm.

Lastly we need a data fidelity term and a natural candidate is the loss

$$l(\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{n}) = \sum_{t=1}^{T}\sum_{i=1}^{p} \frac{n_{t,i}}{2}(y_{t,i} - \beta_{t,i})^2. \tag{3.24}$$

Combining (3.22), (3.23) and (3.24), our CMG function is

$$\begin{aligned} f(\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{n}) &= l(\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{n}) + \Omega_1(\boldsymbol{\beta}) + \Omega_2(\boldsymbol{\beta}) \\ &= l(\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{n}) + \Omega(\boldsymbol{\beta}), \end{aligned} \tag{3.25}$$

where $\Omega(\boldsymbol{\beta}) = \Omega_1(\boldsymbol{\beta}) + \Omega_2(\boldsymbol{\beta})$. Thus, given a pair of (λ_1, λ_2), one can generate a good candidate model by minimizing (3.25).

3.3.2 Generating Candidate Models with the CMG Function

Although the penalty $\Omega(\boldsymbol{\beta})$ is not smooth, (3.25) can still be approximately minimized in the following manner. First, using the smoothing proximal gradient method of Chen *et al.* (2012), we obtain a smooth approximation of $\Omega(\boldsymbol{\beta})$ so that its gradient with respect to $\boldsymbol{\beta}$ can be derived. Then we apply the fast iterative shrinkage-thresholding algorithm (FISTA) of Beck and Teboulle (2009) to carry out the minimization. This procedure is summarized in Algorithm 1, and technical details such as the smooth approximation of $\Omega(\boldsymbol{\beta})$ are deferred to Section 3.7.2.

We note that the output from Algorithm 1 does not produce exactly the same value for $\beta_{t,i}$'s that belong to the same time interval and cluster. For example, suppose for a certain node Algorithm 1 returns $\tilde{\boldsymbol{\beta}} = (1.0, 1.1, 0.9, 2.3, 2.2, 2.3, 2.2)$ for $t = 1, \ldots 7$, which signifies there is a change point at $t = 4$. To circumvent this issue, we conduct a fast scanning operation that will adjust the values to $\hat{\boldsymbol{\beta}} = (1.0, 1.0, 1.0, 2.25, 2.25, 2.25, 2.25)$. Details of the scanning operation are given in Algorithms 2 and 3.

Thus, by performing the above steps, we can quickly obtain a good candidate model $\{\hat{\mathcal{T}}, \hat{\mathcal{C}}\}$ for a given pair of (λ_1, λ_2). As an optional step, we can quickly generate more good candidate models by perturbing $\{\hat{\mathcal{T}}, \hat{\mathcal{C}}\}$, such as removing a change point in $\hat{\mathcal{T}}$.

Lastly we comment on the choice of (λ_1, λ_2), for which in practice depends on the scale of the observations. Specifically, a large T usually requires large values of λ_1, while λ_2 depends on the number of edges $|\boldsymbol{E}|$ of the prespecified graph $\boldsymbol{G}$.

3.3.3 Summary

The minimization for $\text{MDL}(\mathcal{T}, \mathcal{C})$ defined by (3.18) can be summarized by the following steps:

1. Given (λ_1, λ_2), apply Algorithm 1 to minimize (3.25) to obtain $\tilde{\boldsymbol{\beta}}$.

2. Apply Algorithms 2 and 3 to $\tilde{\boldsymbol{\beta}}$ to obtain a good candidate model $\{\hat{\mathcal{T}}, \hat{\mathcal{C}}\}$.

3. (Optional) Perturb $\hat{\mathcal{T}}$ to generate more $\{\hat{\mathcal{T}}, \hat{\mathcal{C}}\}$'s.

4. Repeat Steps 1 to 3 with different values of (λ_1, λ_2) to obtain more $\{\hat{\mathcal{T}}, \hat{\mathcal{C}}\}$'s.

```
Algorithm 1: FISTA for minimizing (3.25)
Input: $\boldsymbol{Y}$, $\boldsymbol{n}$, $\boldsymbol{C}$ derived by (3.43), (3.44) and (3.45), $\boldsymbol{\beta}^{[0]}$, Lipschitz constant $L$
derived by (3.49) or (3.50), $D$ derived by (3.46), desired accuracy $\varepsilon$, $\lambda_1$, $\lambda_2$ ;
Initialize: $\mu = \frac{\varepsilon}{2D}$, $\theta_0 = 1$ ;
for $k$ = 0, 1, 2,..., until $\boldsymbol{\beta}^{[k]}$ converges do
    Compute $\boldsymbol{\alpha}^{*[k]}$ based on $\boldsymbol{\beta}^{[k]}$ by (3.47) and (3.48) ;
    $\nabla h(\boldsymbol{w}^{[k]}) \leftarrow \boldsymbol{n}(\boldsymbol{w}^{[k]} - \boldsymbol{X}) + \boldsymbol{C}^\top \boldsymbol{\alpha}^{*[k]}$ ;
    $\boldsymbol{\beta}^{[k+1]} \leftarrow \boldsymbol{w}^{[k]} - \frac{1}{L}\nabla h(\boldsymbol{w}^{[k]})$ ;
    $\theta_{k+1} \leftarrow \frac{2}{k+3}$ ;
    $\boldsymbol{w}^{[k+1]} \leftarrow \boldsymbol{\beta}^{[k+1]} + \frac{1-\theta_k}{\theta_k}\theta_{k+1}(\boldsymbol{\beta}^{[k+1]} - \boldsymbol{\beta}^{[k]})$ ;
end
$\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{[k+1]}$;
Output: $\tilde{\boldsymbol{\beta}}$
```

5. Calculate the MDL($\mathcal{T}$,$\mathcal{C}$) values for all $\{\hat{\mathcal{T}},\hat{\mathcal{C}}\}$'s obtained from Step 4. Take the one that gives the smallest value as the minimizer of MDL($\mathcal{T}$,$\mathcal{C}$).

3.4 Simulation Experiments

3.4.1 Setting 1: Regular Grid

In this first experiment the graph structure was a square image of size 8×8. That is, there were $p = 64$ nodes arranged as a 8×8 two-dimensional grid, and each node was connected to its 4 neighboring nodes, except for those nodes at the edges and corners of the grid, where they were connected to, respectively, 3 and 2 neighboring nodes. We set $T = 100$ and had change points at t = 25, 50, 60 and 90. The nodes were partitioned into two groups and for each time segment, all the nodes within the same group share the same true signal $\beta_{t,i}$ value. The true signal values are reported in Table 3.1 and they are visually displayed in Figure 3.1. All the $n_{t,i}$'s were set to 1.

Gaussian noise with variance $\sigma^2 \in \{0.1^2, 0.2^2, 0.3^2, 0.4^2\}$ was added to the true signal to generate the noisy observations $x_{t,i,j}$, with 100 repetitions for each value of σ^2. For each noisy data set, 25 combined values of $\lambda_1 \in \{0.5, 1, 2, 4, 8, 16\}$ and $\lambda_2 \in \{0.5, 1, 2, 4, 8, 16\}$

segment	interval	cluster sizes	values
1	[0, 25)	16, 48	2, 1
2	[25, 50)	16, 48	2.2, 1
3	[50, 60)	26, 38	2.1, 1
4	[60, 90)	26, 38	2.4, 1
5	[90, 100)	35, 29	2.4, 1

Table 3.1: True signal values used for Experimental Setting 1.

were used in Algorithm 1 to obtain the MDL solution.

Figure 3.2 presents the results of this numerical experiment. The histograms show the locations of all the detected change points for the 100 repetitions. As expected, the larger the noise variance, the more difficult to detect the change points. This phenomenon is more obvious for those change points where the changes of the true signal values were small: $t = 25$ and 35. To be more specific, the only difference in the true signal before and after the change point at $t = 25$ was the value for the top-left region. As the noise level increases, it becomes more difficult to detect this change point. Similar phenomenon was observed for the change point at $t = 60$.

Apart from reporting the histograms of the detected change points, we also evaluated the quality of the signal estimates $\hat{\beta}_{t,i}$ in terms of mean squared error (MSE):

$$\text{MSE} = \frac{1}{\sum_{t=1}^{T}\sum_{i=1}^{p} n_{t,i}} \sum_{t=1}^{T}\sum_{i=1}^{p} n_{t,i}(\hat{\beta}_{t,i} - \beta_{t,i})^2.$$

We report the MSE results in a similar fashion as Wang *et al.* (2016). First define the negative signal-to-noise ratio (SnR) as

$$10\log_{10}(\sum_{t=1}^{T}\sum_{i=1}^{p}\frac{\sigma^2}{n_{t,i}} / \sum_{t=1}^{T}\sum_{i=1}^{p}(\beta_{t,i} - \bar{\beta})^2).$$

Thus, the negative SnR increases as the noise level increases. Next define the denoised negative SnR as

$$10\log_{10}(\text{MSE}/\frac{1}{Tp}\sum_{t=1}^{T}\sum_{i=1}^{p}(\beta_{t,i} - \bar{\beta})^2),$$

and hence the smaller the denoised negative SnR is, the better the estimates $\beta_{t,i}$'s are. We compared the results obtained from the proposed method with their corresponding saturated models: here a saturated model was the model with a separate parameter $\beta_{t,i}$ fitted for each node. The results are also reported in Figure 3.2. As the noise level increases, the denoised negative SnRs for both the MDL fitted model and the saturated model increase. Compared with the saturated models, the denoised negative SnRs for the MDL fitted models are smaller, even more so for those cases with high noise levels.

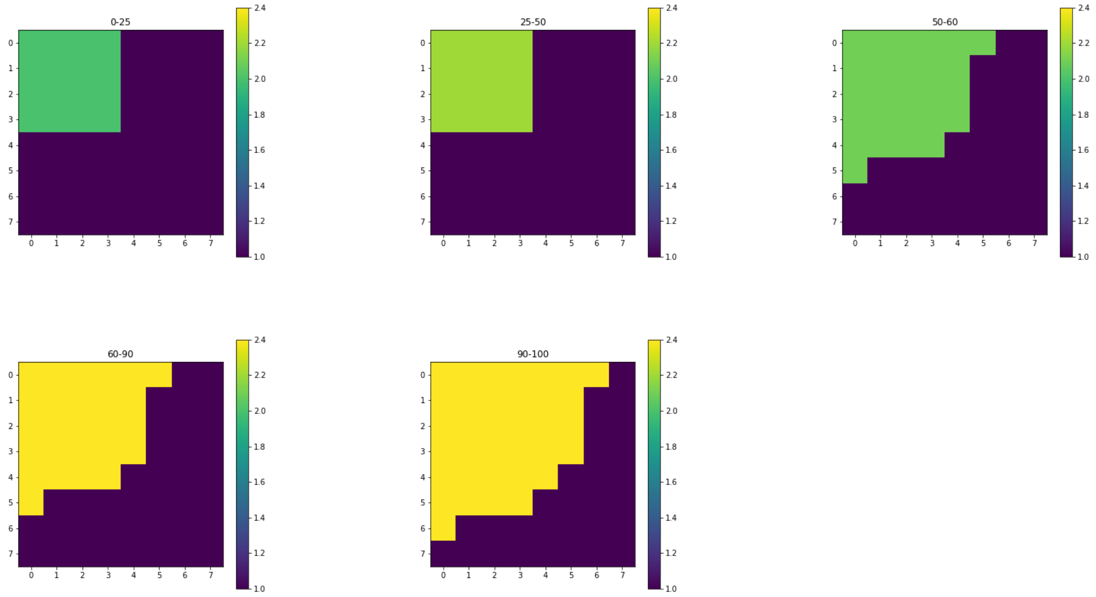
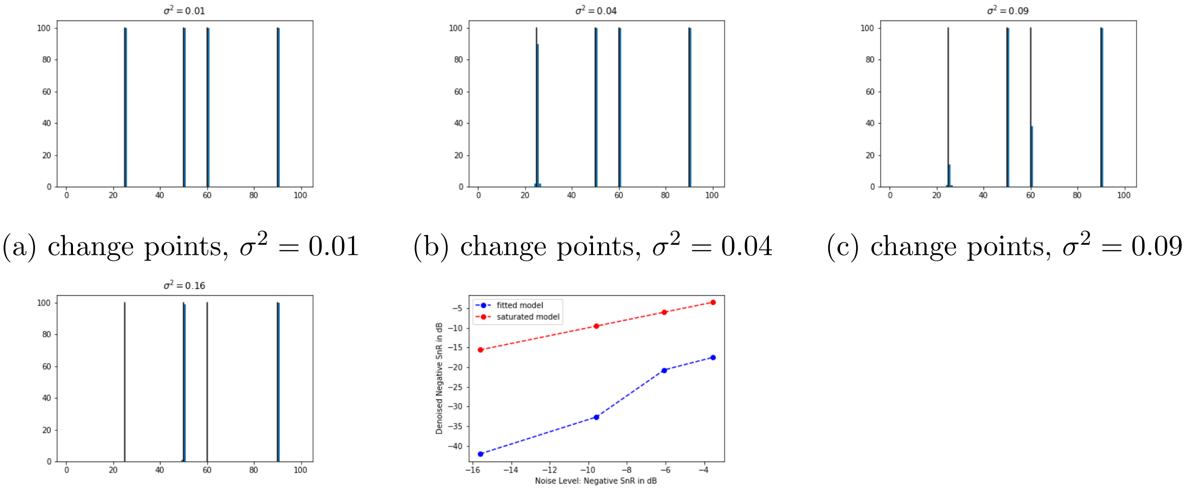


Figure 3.1: True (but unknown) signal values for Experimental Setting 1.

3.4.2 Setting 2: Graph based on California Counties

In this second experiment the graph structure was defined by the 58 counties in California. Each county was a node, and two nodes were connected if the two corresponding counties share a common border. So there were 58 nodes and 136 edges; see Figure 3.3. We partitioned the nodes into 4 groups, and the number of time points was $T = 60$ with change points at $t = 10, 20, 35$ and 45. For each time segment, all the nodes within the same group share the same true signal $\beta_{t,i}$ value; see Table 3.2 and Figure 3.4. Note that these signal values were selected so that the overall signal averages were the same for all the time intervals. Consequently, any univariate change point detection method will fail when it is applied to the (univariate) time series of combined signal values for all time points, as the important graph structure information is ignored.



(a) change points, $\sigma^2 = 0.01$ (b) change points, $\sigma^2 = 0.04$ (c) change points, $\sigma^2 = 0.09$

(d) change points, $\sigma^2 = 0.16$ (e) negative SnR

Figure 3.2: (a)-(d) Histograms of the detected change points under different noise levels. (f) Denoised negative SnRs for different noise levels. Recall that a saturated model is a model with a separate parameter fitted for each node.

segment	interval	cluster sizes	values
1	[0, 10)	10, 17, 12, 19	1, 2, 3, 4
2	[10, 20)	10, 17, 12, 19	1, 3.12, 4, 2
3	[20, 35)	10, 17, 12, 19	2, 3.12, 3.17, 2
4	[35, 45)	10, 17, 12, 19	2, 4, 2, 1.95
5	[45, 50)	10, 17, 12, 19	3, 2.53, 3.17, 2

Table 3.2: True signal values used for Experimental Setting 2.

We tested the proposed method with 6 difference noise variance $\sigma^2 \in \{1^2, 2^2, 3^2, 4^2, 6^2, 8^2\}$ and 36 combined values of $\lambda_1 \in \{2, 4, 8, 16, 32, 64\}$ and $\lambda_2 \in \{0.5, 1, 2, 4, 8, 16\}$. As before, the number of repetitions was 100. The histograms of the deteced change points are given in Figure 3.5, as well as the denoised negative SnRs. Similar empirical conclusions can be drawn as before: the larger the noise level, the more difficult to detect the change points.

3.5 Real Data Applications

3.5.1 Violent Crime in Cincinnati, OH

The data set in this subsection concerns with reported crime incidents in Cincinnati, OH. It contains dates, times, locations and other information of the reported events. We considered

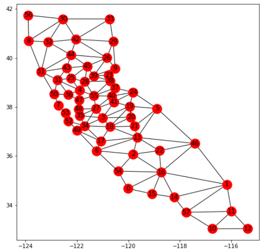


Figure 3.3: The graph structure defined by the counties in California.

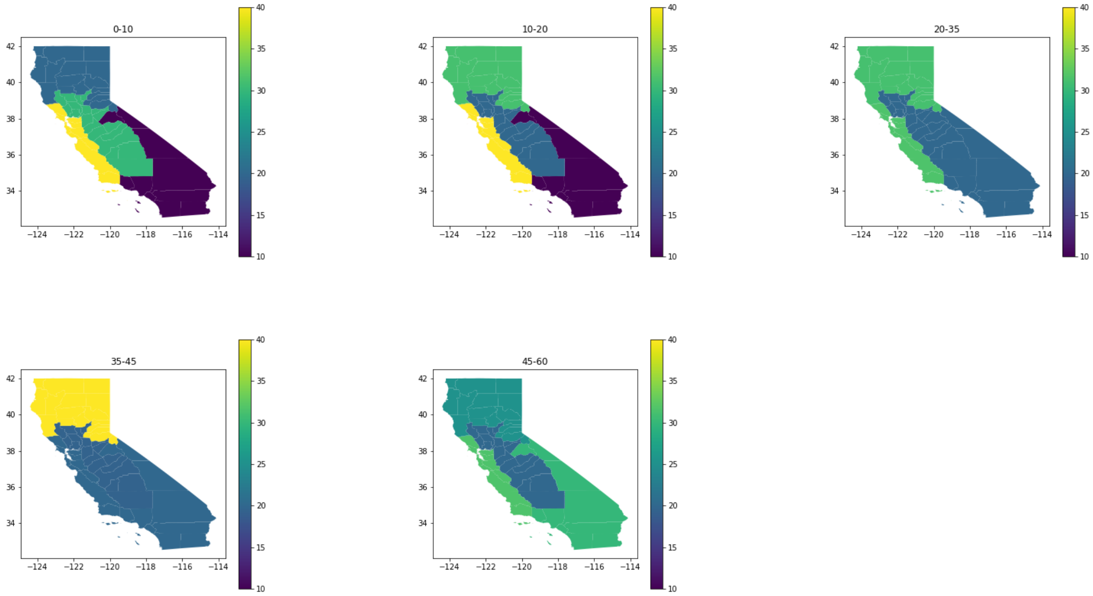
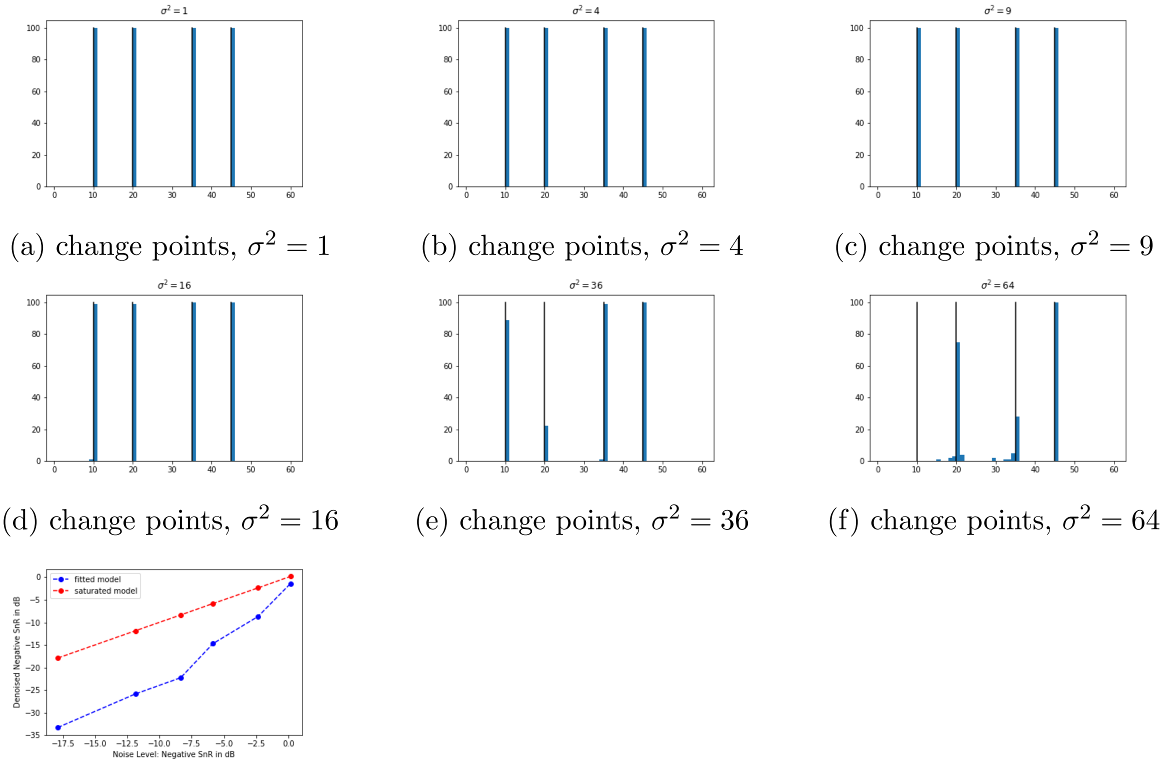


Figure 3.4: True (but unknown) signal values for Experimental Setting 2.

weekly crime rates from December 31, 2018 to December 29, 2019; i.e., $T = 52$.

Each crime event has a FBI Uniform Crime Reporting code that describes its type. As similar to Taddy (2010), we used this code to classify each crime event into violent crime or non-violent crime: a violent crime can be homicide, rape, aggravated assault or robbery, while all the other types of crimes are non-violent.

The nodes were defined by ZIP Code Tabulation Areas in Cincinnati, and edges were defined by geographically neighborhoods. There were 31 nodes and 77 edges in the graph; see Figure 3.6(a). During the t-th week and at the i-th node, the number of observations $n_{t,i}$ was the total number of reported crime events, while the j-th measurement $x_{t,i,j}$ was 1 if the j-th crime was violent, and 0 otherwise. Thus, the data was in fact binomial and we



(a) change points, $\sigma^2 = 1$

(b) change points, $\sigma^2 = 4$

(c) change points, $\sigma^2 = 9$

(d) change points, $\sigma^2 = 16$

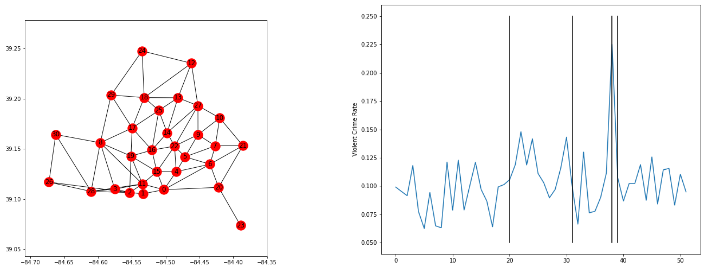
(e) change points, $\sigma^2 = 36$

(f) change points, $\sigma^2 = 64$

(g) negative SnR

Figure 3.5: (a)-(f) Histograms of the detected change points under different noise levels. (g) Denoised negative SnRs for different noise levels.

modified the likelihood function in the MDL criterion (3.18) to reflect this.



(a)

(b)

Figure 3.6: (a) The graph structure defined by the ZIP Code Tabulation Areas in Cincinnati, OH. (b) Violent crime rate for each week from 2018-12-31 to 2019-12-29 in Cincinnati, OH. Vertical lines denote detected change point locations .

Change points were detected at 2019-05-20, 2019-08-05, 2019-09-23 and 2019-09-30. The weekly overall violent crime rates, together with these 4 change points, are displayed in Figure 3.6(b). Ranson (2014) studied the relationships between temperature and different kinds of crimes. The author concluded that, higher temperatures lead to statistically significant increases in all types of crimes. However, the rate of increase is approximately constant for violent crimes, while for non-violent crimes, the rate of increase starts to slow down around 50 °F. Therefore, the first detected change point (late May) signifies the beginning of summer and hence an increased rate for violent crime. The second detected change point (early August) was close to the end of the peak travel season which may explain the drop of violent crime rates. The last two change points together actually suggest that the week in between was an outlier. In fact, that week included the last weekend before Halloween, and it is known that violent crime rate (e.g., robbery and sexual assault) increases shortly before or at Halloween.

3.5.2 Temperatures in Counties in California

The data set is the output of PRISM (parameter-elevation regressions on independent slopes model), a combination of statistical and human-expert methods for climate mapping (Daly *et al.*, 1994). It contains different readings such as temperatures and precipitation. In this study we considered mean annual temperatures from 1960 to 2019 in 58 counties in California. We collected data at the grids of 0.2×0.2 degrees of longitude/latitude. [1] The graph structure was defined by the 58 counties in California, in the same manner as in Section 3.4.2; see Figure 3.3. The proposed method was applied and detected one change point at the year 2012. We plotted the average temperature of the whole California in Figure 3.7, together with the detected change point. It seems that the mean annual temperatures after the change points are higher than those before the change point, hence supporting a warming trend in California (Anderson, 2016).

3.6 Concluding Remarks

This paper proposed a method for simultaneous change point detection and node clustering for time-evolving graphs. The method is composed of two major components: (i) an MDL criterion for which the best fitting model is defined as its minimizer, and (ii) a practical algorithm for finding this minimizer. It is shown that the MDL criterion yields statistically

[1] Obtained from `http://www.prism.oregonstate.edu/explorer/map.php`

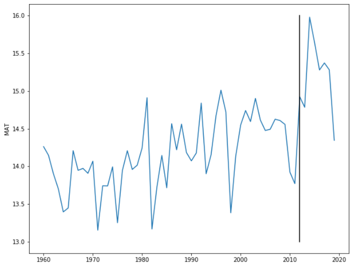


Figure 3.7: Mean annual temperatures (°C) of California in 1960-2019. The vertical line indicates the detected change point at 2012.

consistent estimates, while simulation results suggest that the method also enjoys highly desirable empirical properties.

Future work includes extending the piecewise constant assumption to piecewise linear or even quadratic fitting, for accomodating more signal trends. Another possible extension is to relax the iid noise assumption. For example, different time intervals can have different noise levels, or the noise can be temporally and/or spatially correlated. One could also allow for outliers in the observations, or placing different weights on the nodes. It should be relatively straightforward to derive a tailored MDL criterion for any of these extensions. The major challenge is then, how to practically minimize the criterion.

3.7 Supplement: Technical Details

3.7.1 Proofs of Theoretical Results

Here we provide the proofs of the theoretical results presented in Section 3.2.3.

3.7.1.1 Proof of Lemma 3.1

Let $\mathcal{B}$ be a probability 1 set. For each $\omega \in \mathcal{B}$, suppose on the contrary $\hat{\mathcal{T}} \nrightarrow \mathcal{T}^0$ or $\hat{\mathcal{C}} \nrightarrow \mathcal{C}^0$. As the numbers of time points and nodes are finite, the possible values for $\mathcal{T}$ and $\mathcal{C}$ are finite. Therefore, there exists a subsequence $\{n_k\}$ such that $\hat{\mathcal{T}} \to \mathcal{T}^*$ and $\hat{\mathcal{C}} \to \mathcal{C}^*$ for some $\mathcal{T}^*$ and $\mathcal{C}^*$ as k increases.

It is convenient to define the set $R^*(m,r)$ that collects all the time and node indices belonging to the m-th interval and r-th cluster:

$$R^*(m,r) = \{(t,i) | t^*_{m-1} + 1 \leq t \leq t^*_m, c_i^{*(m)} = r\}. \tag{3.26}$$

Therefore if during the interval $\{t^*_{m-1}+1, \leq t^*_m\}$ the ith node belongs to the rth cluster (i.e., $c_i^{*(m)} = r$), then its signal estimate $\hat{\beta}_i^{*(m)}$ is given by the sample mean of all the observations $x_{t,i,j}$'s such that $(t,i) \in R^*(m,r)$. We denote this sample mean as $\hat{\beta}(R^*(m,r))$, and we have

$$\hat{\beta}_i^{*(m)} = \hat{\beta}(R^*(m,r)) = \frac{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} \sum_{j=1}^{n_{t,i}} x_{t,i,j}}{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} n_{t,i}}. \tag{3.27}$$

To simplify notations, we replace n_k by n. For large enough n,

$$\begin{aligned} \frac{1}{n}\mathrm{MDL}(\hat{\mathcal{T}}, \hat{\mathcal{C}}) &= \frac{1}{n}\log(M+1) + \frac{1}{n}\sum_{m=1}^{M}\log(t^*_m - t^*_{m-1}) + \frac{1}{n}\sum_{m=1}^{M+1}(p+1)\log(d^{(m)}) \\ &\quad + \frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\frac{1}{2}\log(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r} n_{t,i}) + \frac{1}{n}\frac{n}{2}\log(\frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\mathrm{SSE}_r^{*(m)}) \\ &= c_n + \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}}\mathrm{SSE}_r^{*(m)}). \end{aligned} \tag{3.28}$$

In the above c_n is of order $O(\log(n)/n)$ and

$$\mathrm{SSE}_r^{*(m)} = \sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j} - \hat{\beta}_i^{*(m)})^2. \tag{3.29}$$

As $(\mathcal{T}^*, \mathcal{C}^*) \neq (\mathcal{T}^0, \mathcal{C}^0)$, for each $R^*(m,r)$, there are two possible cases, to be examined below.

Case 1

If $R^*(m,r) \subseteq R^0(s,l)$, that is to say, $R^*(m,r)$ is totally within a true $R^0(s,l) = \{(t,i)|t^0_{s-1}+1 \leq t \leq t^0_s, c_i^{0(s)} = l\}$, then $\forall (t,i) \in R^*(m,r) \subseteq R^0(s,l)$, $x_{t,i,j} \sim \mathcal{N}(\beta_{(s)}^{(l)}, \sigma^2)$ *i.i.d.* ($\beta_{(s)}^{(l)}$ denotes the common mean shared by all the nodes in $R^0(s,l)$). Then from (3.27),

$$\hat{\beta}_i^{*(m)} = \hat{\beta}(R^*(m,r)) \to \beta_{(s)}^{0(l)} \ a.s.$$

by strong law of large number. Also, from (3.29)

$$\frac{1}{n}\mathrm{SSE}_r^{*(m)} = \frac{1}{n}\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j} - \hat{\beta}_i^{*(m)})^2 \to \sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r}\gamma_{t,i}\sigma^2 \ a.s., \tag{3.30}$$

where $\gamma_{t,i}$ is defined in (3.20).

Case 2

If $R^*(m,r) \subseteq \cup_{(s,l)\in\boldsymbol{S}} R^0(s,l)$ and $R^*(m,r) \cap R^0(s,l) \neq \emptyset, \forall (s,l) \in \boldsymbol{S}$, which is that same as saying $R^*(m,r)$ has nontrivial intersection with more than one true $R^0(s,l)$, then

$$\begin{aligned}
\hat{\beta}_i^{*(m)} = \hat{\beta}(R^*(m,r)) &= \frac{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} \sum_{j=1}^{n_{t,i}} x_{t,i,j}}{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} n_{t,i}} \\
&= \frac{\sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \sum_{j=1}^{n_{t,i}} x_{t,i,j}}{\sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} n_{t,i}} \\
&\to \frac{\sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \gamma_{t,i}\beta_{(s)}^{0(l)}}{\sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \gamma_{t,i}} \quad a.s.
\end{aligned}$$

And

$$\begin{aligned}
\frac{1}{n}\mathrm{SSE}_r^{*(m)} &= \frac{1}{n} \sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}_i^{*(m)})^2 \\
&= \frac{1}{n} \sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}_i^{*(m)})^2 \\
&\geq \frac{1}{n} \sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}_{(l)}^{0(s)})^2 \\
&\to \sum_{(s,l)\in\boldsymbol{S}} \sum_{t=\max\{t^*_{m-1},t^*_{s-1}\}}^{\min\{t^*_m,t^*_s\}-1} \sum_{i,c_i^*=r,c_i^0=l} \sum_{j=1}^{n_{t,i}} \gamma_{t,i}\sigma^2 \quad a.s. \qquad (3.31)
\end{aligned}$$

Here the strict inequalities hold for at least one (m,r) because $(\mathcal{T}^*, \mathcal{C}^*) \neq (\mathcal{T}^0, \mathcal{C}^0)$ and the total number of clusters $\sum_{m=1}^{M+1} d^{(m)}$ is known.

Therefore, combining (3.28), (3.30) and (3.31), for large enough n,

$$\begin{aligned}
\frac{1}{n}\mathrm{MDL}(\hat{\mathcal{T}}, \hat{\mathcal{C}}) &= c_n + \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M+1}\sum_{r=1}^{d^{(m)}} \mathrm{SSE}_r^{*(m)}) \\
&> c_n + \frac{1}{2}\log(\sigma^2) \\
&= \frac{1}{n}\mathrm{MDL}(\mathcal{T}^0, \mathcal{C}^0) \\
&\geq \frac{1}{n}\mathrm{MDL}(\hat{\mathcal{T}}, \hat{\mathcal{C}}),
\end{aligned}$$

which is a contradiction. This comes to the conclusion that $(\hat{\mathcal{T}}, \hat{\mathcal{C}}) \to (\mathcal{T}^0, \mathcal{C}^0)$ *a.s.* when the total number of clusters $\sum_{m=1}^{M+1} d^{(m)}$ is known.

3.7.1.2 Lemma 3.2 and Its Proof

Lemma 3.2. *Assume the setting of Lemma 3.1 with the exception that the total number of clusters $\sum_{m=1}^{M+1} d^{(m)}$ is unknown. If the change points and the cluster structures are estimated by (3.19), then*

1. *The number of change points cannot be underestimated; i.e., $\hat{M} \geq M^0$ for large enough n.*
2. *The true change points $\mathcal{T}^0$ are a subset of the estimated $\hat{\mathcal{T}}$; i.e., the true change points can be identified for large enough n.*
3. *For large enough n and each $1 \leq m \leq \hat{M}$ with its corresponding s such that $t_{s-1}+1 \leq t_{m-1}+1 < t_m \leq t_s$, there exists a true $R^0(s,l)$ such that*

$$\hat{R}(m,r) \subseteq R^0(s,l)$$

for any of the fitted $\hat{R}(m,r)$. (Here $\hat{R}(m,r)$ and $R^0(s,l)$ are defined in the similar manner as (3.26).) In other words, the cluster structure cannot be underestimated.

The proof of Lemma 3.2 follows the proof of Lemma 3.1. If Case 2 applies, there will be a contradiction. This finishes the proof.

3.7.1.3 Lemma 3.3 and Its Proof

Lemma 3.3. *For k independent $\hat{U}_i \sim \mathcal{N}(\mu, \frac{\sigma^2}{n_i})$, let $\hat{U} = \frac{1}{n}\sum_{i=1}^{k} n_i\hat{U}_i$, where $n = \sum_{i=1}^{k} n_i$. We have*

$$\sum_{i=1}^{k} n_i(\hat{U}_i - \hat{U})^2 \sim \sigma^2\chi^2_{k-1}.$$

Proof: Let $V_i = \sqrt{n_i}\hat{U}_i \sim \mathcal{N}(\sqrt{n_i}\mu, \sigma^2)$ and $\boldsymbol{V} = (V_1, ..., V_k)^\top$. Define a orthonormal matrix $\boldsymbol{A} = (a_1, ..., a_k)^\top$ with $a_1^\top = (\frac{\sqrt{n_1}}{\sqrt{n}}, ..., \frac{\sqrt{n_k}}{\sqrt{n}})$. Therefore, $\hat{U} = \frac{1}{\sqrt{n}}a_1^\top\boldsymbol{V} \sim \mathcal{N}(\mu, \frac{\sigma^2}{n})$. By the property of orthonormal matrices, $E(a_i^\top\boldsymbol{V}) = a_i^\top(\sqrt{n_1}, ..., \sqrt{n_k})^\top = a_i^\top\sqrt{n}a_1 = 0$, for $i = 2, ..., k$. Hence $\boldsymbol{W} = \boldsymbol{AV} \sim \mathcal{N}(\mu(\sqrt{n}, 0, ..., 0)^\top, \sigma^2\boldsymbol{I}_k)$. By the definition of χ^2 distribution, $\sum_{i=2}^{k} W_i^2 \sim \sigma^2\chi^2_{k-1}$ and $\sum_{i=1}^{k} W_i^2 = \boldsymbol{W}^\top\boldsymbol{W} = (\boldsymbol{AV})^\top\boldsymbol{AV} = \boldsymbol{V}^\top\boldsymbol{V} = \sum_{i=1}^{k} V_i^2$. Then,

$$\sum_{i=1}^{k} n_i(\hat{U}_i - \hat{U})^2 = \sum_{i=1}^{k} n_i\hat{U}_i^2 - n\hat{U}^2 = \sum_{i=1}^{k} V_i^2 - (a_1^\top\boldsymbol{V})^2 = \sum_{i=2}^{k} W_i^2 \sim \sigma^2\chi^2_{k-1},$$

which completes the proof.

3.7.1.4 Lemma 3.4 and Its Proof

Lemma 3.4. *For large enough n, if $(\hat{\mathcal{T}},\hat{\mathcal{C}}) \neq (\mathcal{T}^0,\mathcal{C}^0)$, then the difference Δ between the penalty terms in MDL$(\hat{\mathcal{T}},\hat{\mathcal{C}})$ and that in MDL$(\mathcal{T}^0,\mathcal{C}^0)$ is positive and of order $O(\log n)$.*

Proof: Let $\mathcal{B}$ be a probability 1 set. For each $\omega \in \mathcal{B}$, suppose on the contrary $\hat{\mathcal{T}} \nrightarrow \mathcal{T}^0$ or $\hat{\mathcal{C}} \nrightarrow \mathcal{C}^0$. For large enough n, The penalty term of the MDL for the fitted model is

$$\begin{aligned}&\log(M^*+1) + \sum_{m=1}^{M^*} \log(t^*_m - t^*_{m-1}) + \sum_{m=1}^{M^*+1} (p+1)\log(d^{*(m)}) \\ &+ \sum_{m=1}^{M^*+1} \sum_{r=1}^{d^{*(m)}} \frac{1}{2} \log(\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c_i^{*(m)}=r} n_{t,i}),\end{aligned} \tag{3.32}$$

and the penalty term of the MDL for the true model is

$$\begin{aligned}&\log(M^0+1) + \sum_{m=1}^{M^0} \log(t^0_m - t^0_{m-1}) + \sum_{m=1}^{M^0+1} (p+1)\log(d^{0(m)}) \\ &+ \sum_{m=1}^{M^0+1} \sum_{r=1}^{d^{0(m)}} \frac{1}{2} \log(\sum_{t=t^0_{m-1}}^{t^0_m-1} \sum_{i,c_i^{0(m)}=r} n_{t,i}).\end{aligned} \tag{3.33}$$

Define Δ as the difference between (3.32) and (3.33).

As $M^0 \leq M^* \leq T$, $d^{0(m)} \leq p$, $\forall m$ and $d^{*(m)} \leq p$, $\forall m$, the first part of Δ

$$\begin{aligned}&[\log(M^*+1) + \sum_{m=1}^{M^*} \log(t^*_m - t^*_{m-1}) + \sum_{m=1}^{M^*+1} (p+1)\log(d^{*(m)})] \\ -&[\log(M^0+1) + \sum_{m=1}^{M^0} \log(t^0_m - t^0_{m-1}) + \sum_{m=1}^{M^0+1} (p+1)\log(d^{0(m)})]\end{aligned} \tag{3.34}$$

is finite.

By Lemma 3.2, for large enough n, for each of the fitted $\hat{R}(m,r) = R^*(m,r)$, there must exist a true $R^0(s,l)$, such that $R^*(m,r) \subseteq R^0(s,l)$. Without loss of generality, we assume that there exists a true set $R^0(s,l) = \cup_{(m,r)\in S} R^*(m,r)$, which means that this set is over segmented. And for all the other true sets, we have $R^0(s',l') = R^*(m',r')$, that is to say, the fitted model is the same as the true model in all the other sets.

Therefore, the second part of Δ can be written in the following format:

$$\begin{aligned}
&\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\frac{1}{2}\log(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r} n_{t,i}) - \sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\frac{1}{2}\log(\sum_{t=t^0_{m-1}}^{t^0_m-1}\sum_{i,c_i^{0(m)}=r} n_{t,i})\\
= \quad &\sum_{(m,r)\in S}\frac{1}{2}\log(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r} n_{t,i}) - \frac{1}{2}\log(\sum_{t=t^0_{s-1}}^{t^0_s-1}\sum_{i,c_i^{0(s)}=l} n_{t,i}). \qquad (3.35)
\end{aligned}$$

Here we have

$$\sum_{(m,r)\in S}\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c_i^{*(m)}=r} n_{t,i} = \sum_{t=t^0_{s-1}}^{t^0_s-1}\sum_{i,c_i^{0(s)}=l} n_{t,i}. \qquad (3.36)$$

As n is large enough, combining (3.36) with the assumption (3.21), it can be seen that the second part of Δ defined by (3.35) is positive and of order $O(\log(n))$. As in Δ, the other part (3.34) is finite, the second part dominates Δ, which finishes the proof.

3.7.1.5 Proof of Theorem 3.1

By Lemma 3.4, $\frac{1}{n}\Delta$ is positive and of order $O(\log(n)/n)$. The difference between the log-likelihood terms in $\frac{1}{n}\mathrm{MDL}(\mathcal{T}^0,\mathcal{C}^0) - \frac{1}{n}\mathrm{MDL}(\hat{\mathcal{T}},\hat{\mathcal{C}})$ is

$$\frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\mathrm{SSE}_r^{(m)}) - \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}).$$

By Lemma 3.2, this difference is positive. To prove the theorem, it is sufficient to show that the difference is of order $o(\log(n)/n)$. We begin with calculating

$$\begin{aligned}
&\frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\mathrm{SSE}_r^{(m)}) - \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)})\\
= \quad &\frac{1}{2}\log(\frac{\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\mathrm{SSE}_r^{(m)}}{\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}})\\
= \quad &\frac{1}{2}\log(1+\frac{\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\mathrm{SSE}_r^{(m)} - \sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}}{\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}})\\
\leq \quad &\frac{1}{2}\frac{\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}}\mathrm{SSE}_r^{(m)} - \sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}}{\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}}\mathrm{SSE}_r^{*(m)}}. \qquad (3.37)
\end{aligned}$$

Without loss of generality, we use the same idea in the proof of Lemma 3.4. Let

$$\mathrm{SSE}^0_{s,l} = \sum_{t=t^0_{s-1}}^{t^0_s-1} \sum_{i,c^0_i=l} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}(R^0(s,l)))^2,$$

$$\mathrm{SSE}^*_{m,r} = \sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c^*_i=r} \sum_{j=1}^{n_{t,i}} (x_{t,i,j} - \hat{\beta}(R^*(m,r)))^2,$$

where

$$\begin{aligned} \hat{\beta}(R^0(s,l)) &= \frac{\sum_{t=t^0_{s-1}}^{t^0_s-1} \sum_{i,c^0_i=l} \sum_{j=1}^{n_{t,i}} x_{t,i,j}}{\sum_{t=t^0_{s-1}}^{t^0_s-1} \sum_{i,c^0_i=l} n_{t,i}}, \\ \hat{\beta}(R^*(m,r)) &= \frac{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c^*_i=r} \sum_{j=1}^{n_{t,i}} x_{t,i,j}}{\sum_{t=t^*_{m-1}}^{t^*_m-1} \sum_{i,c^*_i=r} n_{t,i}}. \end{aligned} \tag{3.38}$$

Then the numerator of (3.37), $\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}} \mathrm{SSE}_r^{(m)} - \sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}} \mathrm{SSE}_r^{*(m)}$, can be written as

$$\begin{aligned}
& \mathrm{SSE}_{s,l}^0 - \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* \\
= & \sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j}-\hat{\beta}(R^0(s,l)))^2 - \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* \\
= & \sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j}-\hat{\beta}(R^*(m,r))+\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l)))^2 \\
& - \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* \\
= & \sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j}-\hat{\beta}(R^*(m,r)))^2 \\
& +2\sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(x_{t,i,j}-\hat{\beta}(R^*(m,r)))(\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l))) \\
& +\sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l)))^2 \\
& - \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* \\
= & \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* + 0 \\
& +\sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l)))^2 - \sum_{(m,r)\in S} \mathrm{SSE}_{m,r}^* \\
= & \sum_{(m,r)\in S}\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r}\sum_{j=1}^{n_{t,i}}(\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l)))^2 \\
= & \sum_{(m,r)\in S}(\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r} n_{t,i})(\hat{\beta}(R^*(m,r))-\hat{\beta}(R^0(s,l)))^2. \qquad (3.39)
\end{aligned}$$

By (3.38)

$$\hat{\beta}(R^*(m,r)) \sim \mathcal{N}(\beta_{(l)}^{0(s)}, \frac{\sigma^2}{\sum_{t=t_{m-1}^*}^{t_m^*-1}\sum_{i,c_i^*=r} n_{t,i}}) \qquad (3.40)$$

are independent for different $(m, r) \in S$. Also

$$\hat{\beta}(R^0(s,l)) = \frac{\sum_{(m,r)\in S}(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c^*_i=r} n_{t,i})\hat{\beta}(R^*(m,r))}{\sum_{(m,r)\in S}(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c^*_i=r} n_{t,i})}. \tag{3.41}$$

By (3.40), (3.41) and Lemma 3.3, we have

$$\sum_{(m,r)\in S}(\sum_{t=t^*_{m-1}}^{t^*_m-1}\sum_{i,c^*_i=r} n_{t,i})(\hat{\beta}(R^*(m,r)) - \hat{\beta}(R^0(s,l)))^2 \sim \sigma^2\chi^2_{|S|-1}.$$

As $|S| \leq Tp$, we can conclude that (3.39) is of order $O(1)$.

In addition, the denominator of (3.37), $\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}} \text{SSE}_r^{*(m)}$, satisfies

$$\frac{1}{n}\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}} \text{SSE}_r^{*(m)} \to \sigma^2 \quad a.s.$$

Furthermore, we can show that the numerator and the denominator of (3.37) are independent. That is to say,

$$0 < \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}} \text{SSE}_r^{(m)}) - \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}} \text{SSE}_r^{*(m)}) = o(\log(n)/n). \tag{3.42}$$

Then for large enough n, combining Lemma 3.4 and (3.42), we have

$$\begin{aligned}\frac{1}{n}\text{MDL}(\mathcal{T}^0, \mathcal{C}^0) - \frac{1}{n}\text{MDL}(\hat{\mathcal{T}}, \hat{\mathcal{C}}) =& -\frac{1}{n}\Delta + \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^0+1}\sum_{r=1}^{d^{0(m)}} \text{SSE}_r^{(m)}) \\ & - \frac{1}{2}\log(\frac{1}{n}\sum_{m=1}^{M^*+1}\sum_{r=1}^{d^{*(m)}} \text{SSE}_r^{*(m)}) \\ <& 0,\end{aligned}$$

which is a contradiction. This finishes the proof.

3.7.2 Details of Smoothing Proximal Gradient Descent

This part provides details required for the minimization of (3.25).

3.7.2.1 An Alternative Expression for $\Omega_1(\cdot)$ in (3.22)

Let $\boldsymbol{\alpha}_{(1)} = (\boldsymbol{\alpha}_1^\top, \boldsymbol{\alpha}_2^\top, ..., \boldsymbol{\alpha}_{T-1}^\top)^\top$ and $\mathcal{Q}_1 = \{\boldsymbol{\alpha}_{(1)} | \|\boldsymbol{\alpha}_t\|_2 \leq 1, t = 1, ..., T-1\}$. Notice that for any vector $\boldsymbol{v}$, $\|\boldsymbol{v}\|_2 = \max_{\|\boldsymbol{\alpha}\|_2 \leq 1} \alpha^\top \boldsymbol{v}$, where $\boldsymbol{\alpha}$ is a vector that has the same dimension as $\boldsymbol{v}$.

Then $\Omega_1(\boldsymbol{\beta})$ can be written as

$$\begin{aligned}
\Omega_1(\boldsymbol{\beta}) &= \lambda_1 \sum_{t=1}^{T-1} \|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|_2 \\
&= \lambda \sum_{t=1}^{T-1} \max_{\|\boldsymbol{\alpha}_t\|_2 \le 1} \boldsymbol{\alpha}_t^\top (\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t) \\
&= \lambda \max_{\boldsymbol{\alpha}_{(1)} \in \mathcal{Q}_1} \sum_{t=1}^{T-1} \boldsymbol{\alpha}_t^\top (\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t) \\
&= \max_{\boldsymbol{\alpha}_{(1)} \in \mathcal{Q}_1} \alpha_{(1)}^\top \boldsymbol{C}_1 \boldsymbol{\beta},
\end{aligned}$$

where the matrix $\boldsymbol{C}_1 \in \mathcal{R}^{(T-1)p \times Tp}$ is defined as

$$\boldsymbol{C}_1 = \lambda_1 \begin{pmatrix} -\boldsymbol{I} & \boldsymbol{I} & & & \\ & -\boldsymbol{I} & \boldsymbol{I} & & \\ & & \ddots & \ddots & \\ & & & -\boldsymbol{I} & \boldsymbol{I} \end{pmatrix} \tag{3.43}$$

with $\boldsymbol{I} = \boldsymbol{I}_p$ being the p-dimensional identity matrix.

3.7.2.2 An Alternative Expression for $\Omega_2(\cdot)$ in (3.23)

Let $\boldsymbol{\alpha}_{(2)} \in \mathcal{R}^{T|E|}$ and $\mathcal{Q}_2 = \{\boldsymbol{\alpha} | \|\boldsymbol{\alpha}\|_\infty \le 1\}$, and notice that $\|\boldsymbol{v}\|_1 = \max_{\|\boldsymbol{\alpha}\|_\infty \le 1} \boldsymbol{\alpha}^\top \boldsymbol{v}$. Then the penalty $\Omega_2(\boldsymbol{\beta})$ can be written as

$$\Omega_2(\boldsymbol{\beta}) = \lambda_2 \sum_{t=1}^{T} \|\boldsymbol{G}\boldsymbol{\beta}_t\|_1 = \|C_2\boldsymbol{\beta}\|_1 = \max_{\alpha_{(2)} \in \mathcal{Q}_2} \alpha_{(2)}^\top \boldsymbol{C}_2 \boldsymbol{\beta},$$

where

$$\boldsymbol{C}_2 = \lambda_2 \begin{pmatrix} \boldsymbol{G} & & \\ & \ddots & \\ & & \boldsymbol{G} \end{pmatrix}. \tag{3.44}$$

3.7.2.3 A Smooth Approximation of $\Omega(\cdot) = \Omega_1(\cdot) + \Omega_2(\cdot)$

Let

$$\boldsymbol{C} = (\boldsymbol{C}_1^\top, \boldsymbol{C}_2^\top)^\top \tag{3.45}$$

and $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{(1)}^\top, \boldsymbol{\alpha}_{(2)}^\top)^\top$. The penalty term $\Omega(\boldsymbol{\beta})$ can be written as

$$\Omega(\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha}_{(1)} \in \mathcal{Q}_1} \boldsymbol{\alpha}_{(1)}^\top \boldsymbol{C}_1 \boldsymbol{\beta} + \max_{\boldsymbol{\alpha}_{(2)} \in \mathcal{Q}_2} \boldsymbol{\alpha}_{(2)}^\top \boldsymbol{C}_2 \boldsymbol{\beta} = \max_{\boldsymbol{\alpha} \in \mathcal{Q}} \boldsymbol{\alpha}^\top \boldsymbol{C} \boldsymbol{\beta},$$

where $\mathcal{Q} = \{\boldsymbol{\alpha} = (\boldsymbol{\alpha}_{(1)}^\top, \boldsymbol{\alpha}_{(2)}^\top)^\top | \boldsymbol{\alpha}_{(1)} \in \mathcal{Q}_1 \text{ and } \boldsymbol{\alpha}_{(2)} \in \mathcal{Q}_2\}$.

By Nesterov (2005), The smooth approximation of $\Omega(\boldsymbol{\beta})$ can be constructed as

$$g_\mu(\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha} \in \mathcal{Q}} (\boldsymbol{\alpha}^\top \boldsymbol{C} \boldsymbol{\beta} - \mu d(\boldsymbol{\alpha})),$$

where μ is a positive smoothness parameter and $d(\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2$. Therefore, the original penalty term $\Omega(\boldsymbol{\beta})$ can be viewed as $g_0(\boldsymbol{\beta})$.

Let $D = \max_{\alpha \in \mathcal{Q}} d(\boldsymbol{\alpha})$, then by Nesterov (2005),

$$g_0(\boldsymbol{\beta}) - \mu D \leq g_\mu(\boldsymbol{\beta}) \leq g_0(\boldsymbol{\beta}),$$

which means that $g_\mu(\boldsymbol{\beta})$ is an approximation of $g_0(\boldsymbol{\beta})$ with a maximum gap of μD. Chen *et al.* (2012) suggested that $\mu = \frac{\varepsilon}{2D}$ achieves the best convergence rate for the given desired accuracy ε. For the current problem

$$D = \max_{\alpha \in \mathcal{Q}} d(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}_{(1)} \in \mathcal{Q}_1} \frac{1}{2}\|\boldsymbol{\alpha}_{(1)}\|_2^2 + \max_{\boldsymbol{\alpha}_{(2)} \in \mathcal{Q}_2} \frac{1}{2}\|\boldsymbol{\alpha}_{(2)}\|_2^2 = \frac{1}{2}(T-1) + \frac{1}{2}T|\boldsymbol{E}|. \tag{3.46}$$

Also, by Theorem 1 in Chen *et al.* (2012), for $\mu > 0$, $g_\mu(\boldsymbol{\beta})$ is convex and continuously-differentiable with respect to $\boldsymbol{\beta}$. And the gradient is

$$\nabla g_\mu(\boldsymbol{\beta}) = \boldsymbol{C}^\top \boldsymbol{\alpha}^*,$$

where $\boldsymbol{\alpha}^* = \arg\max_{\alpha \in \mathcal{Q}} \alpha^\top \boldsymbol{C}\boldsymbol{\beta} - \mu d(\boldsymbol{\alpha})$. And $\nabla g_\mu(\boldsymbol{\beta})$ is Lipschitz continuous with Lipschitz constant $L_\mu = \frac{1}{\mu}\|\boldsymbol{C}\|^2$, where $\|.\|$ is the matrix spectral norm. ($\|\boldsymbol{C}\| \equiv \max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{C}\boldsymbol{v}\|_2$).

As $\boldsymbol{\alpha}^* = ((\boldsymbol{\alpha}_{(1)}^*)^\top, (\boldsymbol{\alpha}_{(2)}^*)^\top)^\top$, by Chen *et al.* (2012), we have

$$\begin{aligned} \boldsymbol{\alpha}_{(1)}^* &= (\boldsymbol{\alpha}_{(1),1}^*, ..., \boldsymbol{\alpha}_{(1),(T-1)}^*)^\top \\ \boldsymbol{\alpha}_{(1),t}^* &= S_1(\frac{\lambda_1}{\mu}(\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t)), \quad t = 1, \ldots, T-1, \end{aligned} \tag{3.47}$$

where S_1 is the projection operator that projects a vector onto l_2 unit ball:

$$S_1(\boldsymbol{u}) = \begin{cases} \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_2} & \|\boldsymbol{u}\|_2 \geq 1, \\ \boldsymbol{u} & \|\boldsymbol{u}\|_2 < 1. \end{cases}$$

In addition,

$$\begin{aligned} \boldsymbol{\alpha}_{(2)}^* &= (\boldsymbol{\alpha}_{(2),1}^*, ..., \boldsymbol{\alpha}_{(2),T}^*)^\top \\ \boldsymbol{\alpha}_{(2),t}^* &= S_2(\frac{\lambda_2}{\mu} G\boldsymbol{\beta}_t), \quad t = 1, \ldots, T, \end{aligned} \tag{3.48}$$

where S_2 is the projection operator defined as

$$S_2(x) = \begin{cases} x & x \in [-1, 1] \\ -1 & x < -1 \\ 1 & x > 1. \end{cases}$$

And for any vector $\boldsymbol{u}$, the projection $S_2(\boldsymbol{u})$ is defined as applying S_2 element-wise. So the operator can be viewed as the projection operator that projects a vector onto l_∞ unit ball.

3.7.2.4 Smoothing Proximal Gradient Descent

By replacing the penalty term $\Omega(\boldsymbol{\beta})$ with $g_\mu(\boldsymbol{\beta})$, we obtain the following optimization problem

$$\min_{\boldsymbol{\beta}} h(\boldsymbol{\beta}) \equiv l(\boldsymbol{\beta}|X, \boldsymbol{n}) + g_\mu(\boldsymbol{\beta}).$$

The gradient of $h(\boldsymbol{\beta})$ is

$$\nabla h(\boldsymbol{\beta}) = \boldsymbol{n}(\boldsymbol{\beta} - \boldsymbol{X}) + \boldsymbol{C}^\top \boldsymbol{\alpha}^*,$$

which is Lipschitz continuous with the Lipschitz constant

$$L = \boldsymbol{n}_{\max} + L_\mu = \boldsymbol{n}_{\max} + \frac{1}{\mu}\|\boldsymbol{C}\|^2, \tag{3.49}$$

where $\boldsymbol{n}_{\max}$ is the largest element of vector $\boldsymbol{n}$.

3.7.2.5 Computation of the Lipschiz Constant

To use the smoothing proximal gradient descent algorithm, one needs to compute the Lipschiz constant L (3.49). However, it is difficulty to calculate the spectral norm $\|\boldsymbol{C}\|$ when the dimension of $\boldsymbol{C}$ is high. Therefore, following Chen *et al.* (2012), we replace it with an upper bound. We begin by calculating

$$\begin{aligned} \|\boldsymbol{C}\|^2 = \left\| \begin{pmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \end{pmatrix} \right\|^2 &= \max_{\|\boldsymbol{v}\|_2 \le 1} \left\| \begin{pmatrix} \boldsymbol{C}_1\boldsymbol{v} \\ \boldsymbol{C}_2\boldsymbol{v} \end{pmatrix} \right\|_2^2 = \max_{\|\boldsymbol{v}\|_2 \le 1} \|\boldsymbol{C}_1\boldsymbol{v}\|_2^2 + \|\boldsymbol{C}_2\boldsymbol{v}\|_2^2 \\ &\le \max_{\|\boldsymbol{v}\|_2 \le 1} \|\boldsymbol{C}_1\boldsymbol{v}\|_2^2 + \max_{\|\boldsymbol{v}\|_2 \le 1} \|\boldsymbol{C}_2\boldsymbol{v}\|_2^2. \end{aligned}$$

Let $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_1^\top & \boldsymbol{v}_2^\top & \dots & \boldsymbol{v}_T^\top \end{pmatrix}^\top$ where $\boldsymbol{v}_t = (v_{t,1}, v_{t,2}, ..., v_{t,p})^\top$, $t = 1, ..., T$. Then for the first term $\max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{C}_1\boldsymbol{v}\|_2^2$,

$$\begin{aligned}
\|\boldsymbol{C}_1\boldsymbol{v}\|_2^2 &= \lambda_1^2 \sum_{t=1}^{T-1} \|\boldsymbol{v}_{t+1} - \boldsymbol{v}_t\|_2^2 = \lambda_1^2 \sum_{t=1}^{T-1} (\|\boldsymbol{v}_{t+1}\|_2^2 - 2\boldsymbol{v}_{t+1} \cdot \boldsymbol{v}_t + \|\boldsymbol{v}_t\|_2^2) \\
&\leq \lambda_1^2 \sum_{t=1}^{T-1} 2(\|\boldsymbol{v}_{t+1}\|_2^2 + \|\boldsymbol{v}_t\|_2^2) \\
&\leq \lambda_1^2 \sum_{t=1}^{T} 4\|\boldsymbol{v}_t\|_2^2 = 4\lambda_1^2 \|\boldsymbol{v}\|_2^2.
\end{aligned}$$

Therefore, $\max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{C}_1\boldsymbol{v}\|_2^2 \leq 4\lambda_1^2$. For the second term $\max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{C}_2\boldsymbol{v}\|_2^2$,

$$\|\boldsymbol{C}_2\boldsymbol{v}\|_2^2 = \lambda_2^2 \sum_{t=1}^{T} \|\boldsymbol{G}\boldsymbol{v}_t\|_2^2 \leq \lambda_2^2 \sum_{t=1}^{T} d_1^2 \|\boldsymbol{v}_t\|_2^2 = \lambda_2^2 d_1^2 \|\boldsymbol{v}\|_2^2,$$

where d_1 is the largest (non-negative) singular value of $\boldsymbol{G}$, or $d_1 = \|\boldsymbol{G}\|$. So $\max_{\|\boldsymbol{v}\|_2 \leq 1} \|\boldsymbol{C}_2\boldsymbol{v}\|_2^2 = \lambda_2^2 d_1^2$.

Finally we have

$$L = \boldsymbol{n}_{\max} + \frac{1}{\mu}\|\boldsymbol{C}\|^2 \leq \boldsymbol{n}_{\max} + \frac{1}{\mu}(4\lambda_1^2 + \lambda_2^2\|\boldsymbol{G}\|^2). \tag{3.50}$$

3.7.3 Processing Output from Algorithm 1

As mentioned in Section 3.3.2, the output from Algorithm 1 does not produce exactly the same signal values $\beta_{t,i}$'s for nodes belonging to the same time interval and cluster. To circumvent this issue, we apply Algorithm 2 to the output from Algorithm 1. Briefly, Algorithm 2 compares the fitted signal values (from Algorithm 1) between any two time points with a pre-set threshold to determine if a change point exists, and if yes, sets all the relevant fitted signal values to the same value. It employs Algorithm 3 recursively to compare connected nodes, in a depth-first manner. Nodes with very similar fitted signal values are assigned to the same cluster.

Algorithm 2: To convert output from Algorithm 1 into a final fitted model

```
Input: fitted coefficients $\tilde{\boldsymbol{\beta}}$, threshold $\epsilon$, edges of the graph $\boldsymbol{E}$, tolerance $\gamma$ ;
Initialize: $\hat{\mathcal{T}} \leftarrow \emptyset$, $\hat{\mathcal{C}} \leftarrow \emptyset$;
$c_1 \leftarrow \gamma\sqrt{p(2\epsilon)^2}$;
for $t$ = 1,..,T-1 do
    if $\|\tilde{\boldsymbol{\beta}}_{t+1} - \tilde{\boldsymbol{\beta}}_t\| > c_1$ then
        Add $t$ to $\hat{\mathcal{T}}$;
    end
end
for $m = 1, ..., |\hat{\mathcal{T}}| + 1$ do
    $t_m \leftarrow m$th element in $\hat{\mathcal{T}}$, $(t_{|\hat{\mathcal{T}}|+1} \leftarrow T + 1)$;
    $t_{m-1} \leftarrow (m-1)$th element in $\hat{\mathcal{T}}$, $(t_0 \leftarrow 1)$;
    $c_2 \leftarrow \gamma\sqrt{(t_k - t_{k-1})(2\epsilon)^2}$;
    $l \leftarrow (-1, -1, ..., -1) \in \mathcal{R}^p$;
    $c \leftarrow 0$;
    for $i$ = 1,...,p do
        if $l_i = -1$ then
            apply Algorithm 3 with $i, \tilde{\boldsymbol{\beta}}, c_2, \boldsymbol{E}, l, c, (t_{m-1}, t_m)$;
            $c \leftarrow c + 1$;
        end
    end
    Add $l$ to $\hat{\mathcal{C}}$;
end
Output: fitted change points $\hat{\mathcal{T}}$, set of fitted membership vectors $\hat{\mathcal{C}}$
```

Algorithm 3: Use a depth-first search strategy to compare connected nodes, and nodes with similar fitted signal values to the coefficients are labelled the same.

```
**Input:** current index $i$, fitted coefficients $\tilde{\boldsymbol{\beta}}$, threshold $c_2$, edges of the graph $\boldsymbol{E}$,
current membership vector $l$,, current label $c$, time interval $(t_{m-1}, t_m)$;
$l_i \leftarrow c$;
$\tilde{\boldsymbol{\beta}}_{(t_{m-1},t_m),i} \leftarrow (\tilde{\boldsymbol{\beta}}_{t_{m-1},i}, \tilde{\boldsymbol{\beta}}_{t_{m-1}+1,i}, ..., \tilde{\boldsymbol{\beta}}_{t_m-1,i})^\top$;
$\tilde{\boldsymbol{\beta}}_{(t_{m-1},t_m),j} \leftarrow (\tilde{\boldsymbol{\beta}}_{t_{m-1},j}, \tilde{\boldsymbol{\beta}}_{t_{m-1}+1,j}, ..., \tilde{\boldsymbol{\beta}}_{t_m-1,j})^\top$;
**for** $j$=1,...,$p$ **do**
    **if** $(i, j) \in \boldsymbol{E}$ **and** $l_i = -1$ **and** $\|\tilde{\boldsymbol{\beta}}_{(t_{m-1},t_m),i} - \tilde{\boldsymbol{\beta}}_{(t_{m-1},t_m),j}\| < c_2$ **then**
        apply Algorithm 3 with $j, \tilde{\boldsymbol{\beta}}, c_2, \boldsymbol{E}, l, c, (t_{m-1}, t_m)$;
    **end**
**end**
**Output:** updated membership vector $l$
```

Chapter 4

Statistical Consistency for Change Point Detection and Community Estimation in Time-Evolving Dynamic Networks

Suppose a time sequence of networks are observed. It is known that the probabilistic behaviors of the networks do not change over time, except at a few time points. These time points are usually called change points, whose number and locations are unknown. This paper proposes a method for automatically estimating such change points, and the community structures of the networks. The proposed method invokes the minimum description length principle to derive a model selection criterion, where the best estimates are defined as its minimizer. It is shown that this selection criterion yields consistent estimates for the change points as well as the community structures. For practical minimization of the selection criterion, a bottom-up search algorithm that combines the EM-Algorithm with variational approximation is developed. The promising empirical properties of the proposed method are illustrated via a sequence of numerical experiments and applications to some real datasets. To the best of the authors' knowledge, this method is one of the earliest that provides consistent estimates in the context of change point detection for time-evolving networks.

4.1 Introduction

Modeling relational information among objects can often be successfully achieved through a network representation, where nodes and edges of the network together form a succinct summary of the relationships among the objects. Depending whether the relationship between any two objects is two-way or one-way, networks can be classified into undirected and directed networks. Also, a network is a weighted network if there exists strength (or some other attribute) for each connection, and unweighted otherwise. In this paper, we focus on the undirected unweighted networks.

Various probabilistic models for networks have been proposed; a good survey is given by Goldenberg *et al.* (2010). A well-studied model is the stochastic block model (SBM) (e.g., Bickel and Chen, 2009; Choi *et al.*, 2012; Bickel *et al.*, 2013). For most of these models, it is assumed that what can be observed is a single network.

In many practical situations, what one observes is a collection of networks that share a common set of nodes. This is referred as multi-graph, which can be further divided into two sub-categories. One such common sub-category is the so-called multi-layer networks. Here, the networks capture different types of relationships, such as the networks of connections through email, messaging, and/or social media among a set of users. Another sub-category is time-evolving dynamic networks; e.g., a sequence of time-stamped social networks of interactions among people.

For community detection in multi-layer networks, Holland *et al.* (1983) extended the standard SBM to the multi-layer setting, and call the resulting model the multi-layer stochastic block model (MLSBM). Han *et al.* (2015) proved the consistency of the maximum likelihood estimates (MLEs) in this model when the number of layers grows. Later, Paul and Chen (2016) proposed a modified model called the restricted multi-layer stochastic block model (RMLSBM). In RMLSBM, contraints are imposed on the link probabilities, and the MLEs for an RMLSBM become regularized estimates. Compared with Han *et al.* (2015), this regularized approach improves the performance of MLEs when either the number of communities grows fast or the network layers are sparse on average. In addition, Stanley *et al.* (2016) proposed the strata multi-layer stochastic block model (sMLSBM). This model agglomerates sets of layers into structurally similar groups called “strata”, and simultaneously clusters nodes into communities.

For time evolving dynamic networks, a common goal is change point detection; i.e., to locate the time points at which community structures change. Several methods have been proposed. The Multi-step method of Aynaud and Guillaume (2011) uses an agglomerative hierarchical clustering approach for change point detection. At every iteration, the most similar segments (i.e., time intervals) are merged together. The similarity between segments is quantified by an average modularity, and the community assignments are estimated by maximizing the average modularity with a modified Louvain algorithm. However, as the output of this method is a hierarchical tree indicating where and when the merges occur, it does not provide an estimate for the number of change points. Therefore, it is suitable only if the number of change points is known or pre-specified.

The SCOUT method of Hulovatyy and Milenković (2016) is another method for detecting change points and community assignments for time evolving dynamic networks. It works if the number of time intervals is pre-specified by the user, and it is also capable of choosing the number of change points automatically by invoking the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). To practically locate the change point locations, three search strategies are combined: exhaustive search, top-down search, and bottom-up search. Add three clustering methods are used for community detection: sum graph, Average-Louvain, and consensus matrix.

More recently, Cheung *et al.* (2020) developed a method that explicitly recasts the network change point detection problem into a statistical model selection problem, and uses the minimum description length (MDL) principle (Rissanen, 1989a, 2007b) to construct a selection criterion that automatically determines the number and locations of the change points, as well as the number and structures of the communities. In practice, a top-down search strategy is used to search for change points while a modified Louvain algorithm is used to estimate the communities.

While these change point detection methods perform well in practice when their assumptions are satisfied by the data, their theoretical properties are largely unknown. The main contribution of this paper is the development of a new method for simultaneous change point detection and community estimation that is shown to possess desirable theoretical properties.

More specifically, this method produces statistically consistent estimates for the number and locations of the change points, as well as the number and structures for the communities. The proposed method also uses the MDL principle, but there are some major differences when comparing to the method of Cheung *et al.* (2020): (i) it assumes a different model so the resulting selection criterion is different, (ii) it uses a different algorithm to estimate the communities, and (iii) it enjoys statistical consistency (while it is shown below that the method of Cheung *et al.* (2020) is not consistent). To the best of our knowledge, this paper is one of the earliest to provide a statistically consistent method for the current problem.

The rest of this paper is organized as follows. Section 4.2 formulates the precise problem that this paper addresses and presents the model for the problem. Section 4.3 uses the MDL principle to derive the a selection criterion for the problem and establishes its consistency properties. Sections 4.4 develops a practical algorithm for optimizing the MDL selection criterion. Simulation results are reported in Section 4.5 while the applications to several real datasets are provided in Section 4.6. Lastly, remarks and discussions are offered in Section 4.7 and technical details are delayed to Section 4.8.

4.2 Problem Statement and Modeling

This section defines the problem that this paper considers, and presents the statistical model that we use. We first begin with the homogeneous case; i.e., when there is no change point.

4.2.1 Homogeneous Case

Consider a sequence of graphs that can be denoted by a sequence of binary adjacency matrices $\{\boldsymbol{A}_t | t = 1, \dots, T\}$ of the same fixed $N \times N$ size. Here $\boldsymbol{A}_{t,ij} = 1$ means that node i and node j are connected at time t, while $\boldsymbol{A}_{t,ij} = 0$ means otherwise. It is assumed that the networks are undirected so that $\boldsymbol{A}_{t,ij} = \boldsymbol{A}_{t,ji}, \forall i, j, t$. It is also assumed that there is no self-loop in the networks so $\boldsymbol{A}_{t,ii} = 0, \forall i, t$. The edges are formed following a Bernoulli distribution independently, with probabilities depending on the nodes and the layer. That is,

$$\boldsymbol{A}_{t,ij} \sim Bernoulli(\Omega_{t,ij}) \quad \forall i \neq j.$$

This subsection assumes that $\Omega_{t,ij}$ does not change over time, and models it with a SBM. That is, the nodes are partitioned into Q blocks (Q unknown) and the linkage probabilities are modeled as follows. Let $\boldsymbol{c} = \{c_1, c_2, ..., c_N\}$ be the community assignment indicator vector, with $c_i \in \{1, 2, ..., Q\}$; i.e., $c_i = q$ means that node i is in block q. The probabilities depend only on the community assignment:

$$\Omega_{t,ij}|(c_i = q, c_j = l) = \pi_{ql}. \tag{4.1}$$

The total number of free parameters is $Q(Q+1)/2$. The parameter set is denoted by $\boldsymbol{\gamma} = \{\pi_{ql}, 1 \leq q \leq l \leq Q\}$. We use $\boldsymbol{\psi} = \{\boldsymbol{c}, \boldsymbol{\gamma}\}$ to denote the whole SBM.

4.2.2 Heterogeneous Case

This subsection considers the heterogeneous case where the community assignments and the parameters are allowed to change at change points. Suppose these T networks can be partitioned into $M+1$ homogeneous intervals with M distinct change points $\mathcal{T} = \{t_1, t_2, ..., t_M\}$. Set $t_0 = 0$ and $t_{M+1} = T$. Given change points $\mathcal{T}$, the observed networks within the same interval follow the homogeneous model presented in the previous subsection. Let $\boldsymbol{B}^{(m)}_{t-t_{m-1}} = \boldsymbol{A}_t, t_{m-1}+1 \leq t \leq t_m$, and $\boldsymbol{B}^{(m)} = \{\boldsymbol{B}^{(m)}_t | 1 \leq t \leq T^{(m)}\}$, where $T^{(m)} = t_m - t_{m-1}$, then given the community assignment vector $\boldsymbol{c}^{(m)}$ and parameters $\boldsymbol{\gamma}^{(m)}$ for interval m,

$$\boldsymbol{B}^{(m)}_{t,ij}|(\boldsymbol{c}^{(m)}, \boldsymbol{\gamma}^{(m)}) \sim Bernoulli(\Omega^{(m)}_{t,ij}) \quad \text{and} \quad \Omega^{(m)}_{t,ij}|(c^{(m)}_i = q, c^{(m)}_j = l) = \pi^{(m)}_{ql} \tag{4.2}$$

if $1 \leq t \leq T^{(m)}$. Here $c^{(m)}_i$ and $c^{(m)}_j$ are community assignments of node i and node j at the mth interval. Therefore, given the change points $\mathcal{T}$, each interval is modeled by a community assignment vector $\boldsymbol{c}^{(m)}$ and parameters $\boldsymbol{\gamma}^{(m)}$.

The precise problem that this paper addresses is to, given model (4.2), estimate $\mathcal{T}$, $\boldsymbol{c}^{(m)}$ and $\boldsymbol{\gamma}^{(m)}$ for all m.

Here we mention the model settings for other two change point detection methods with which we are going to compare with. The model of SCOUT (Hulovatyy and Milenković, 2016) is actually the same as the proposed heterogeneous model. However, the model of the method proposed by Cheung *et al.* (2020) allows the link probabilities to vary over time, that is to say,

$$\Omega^{(m)}_{t,ij}|(c^{(m)}_i = q, c^{(m)}_j = l) = \pi^{(m)}_{t,ql}.$$

It means that within the same interval, the model is actually MLSBM by Han *et al.* (2015). Therefore, the proposed method and SCOUT define change points on which community assignments and/or link probabilities change, while Cheung *et al.* (2020) defines change points as the time points on which community assignments change.

4.3 Model Selection using MDL

Once the change points and the community assignments are given, link probabilities can be easily estimated. However, estimating the change points as well as the community assignments is not a trivial task. This section uses the minimum description length (MDL) principle as the model selection criterion.

The MDL principal defines the "best" model as the one that achieves the best lossless compression of the data. That is to say, based on the best model, the data can be stored in the hardware memory with the shortest code length. Here we use the "two-part" version of MDL, where the first part is the code length of encoding the model, and the second part is the code length of encoding the residuals that can not be explained by the model.

We use $\mathrm{CL}(z)$ to denote the code length of z, then the code length CL("data") of the observed data can be decomposed into two parts, a model $\mathcal{F}$ plus the corresponding residuals $\hat{\mathcal{E}}$:

$$\mathrm{CL}(\text{“data”}) = \mathrm{CL}(\mathcal{F}) + \mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F}),$$

and the "best" model is the one that minimizes CL("data").

4.3.1 Homogeneous Case

Now we assume that there is no change point, that is to say, these T observed network follow the proposed homogeneous model 4.1. In this case, $\mathcal{F} = \{\boldsymbol{\psi}\}$. So $\mathrm{CL}(\mathcal{F})$ can be written as

$$\mathrm{CL}(\mathcal{F}) = \mathrm{CL}(\boldsymbol{\psi}).$$

For a given class assignment $\boldsymbol{c}$, let $n_q(\boldsymbol{c}) = \#\{i|c_i = q\}$ be the number of nodes in class q. Then the number of possible pairs in each block can be denoted as

$$N_{ql}(\boldsymbol{c}) = \begin{cases} n_q n_l & q \neq l, \\ n_q(n_q - 1)/2 & q = l. \end{cases} \tag{4.3}$$

Notice the dependency on $\boldsymbol{c}$ is dropped when there is no ambiguity.

The explicit form of the MDL in homogeneous case is

$$\begin{aligned}\mathrm{MDL}_{homo}(\boldsymbol{\psi};\boldsymbol{A}) =&(N+1)\log(Q)+\sum_{q\leq l}\frac{1}{2}\log_2(N_{ql}(\boldsymbol{c})T)\\ &-\sum_{t=1}^{T}\sum_{i<j}\boldsymbol{A}_{t,ij}\log(\hat{\Omega}_{t,ij})+(1-\boldsymbol{A}_{t,ij})\log(1-\hat{\Omega}_{t,ij}),\end{aligned} \tag{4.4}$$

where $Q=|\boldsymbol{c}|$ denotes the number of communities, and $\hat{\Omega}_{t,ij}$ is the MLE given the community assignment. The details of the derivation of the MDL can be found in Section 4.8.1.1

4.3.2 Heterogeneous Case

In this case, the number of change points M and the locations $\mathcal{T}=\{t_1,t_2,...,t_M\}$ of these change points are unknown and need to be estimated based on MDL.

The overall MDL is

$$\mathrm{MDL}(\mathcal{T},\boldsymbol{\psi})=\log(M+1)+\sum_{m=1}^{M+1}\log(t_m-t_{m-1})+\sum_{m=1}^{M+1}\mathrm{MDL}_{homo}(\boldsymbol{\psi}^{(m)};\boldsymbol{B}^{(m)}). \tag{4.5}$$

The details of the derivation can be found in Section 4.8.1.2. We propose to estimate $(\mathcal{T},\boldsymbol{\psi})$ as the minimizer of (4.5).

4.3.3 Statistical Consistency

This subsection establishes the statistical consistency of the MDL based result $\{\hat{\mathcal{T}},\hat{\boldsymbol{\psi}}\}$ defined by (4.5) when $T\to\infty$. The proofs of the following results can be found in Section 4.8.2.

Let $\lambda_m=t_m/T,\forall m=0,1,...M+1$ be the normalized change points and let $\boldsymbol{\lambda}=(\lambda_1,...,\lambda_M)$ be the normalized change point location vector. The asymptotic results are based on λ_m's being fixed when T increases. The main result is given below:

Theorem 4.1. *Let $\{\boldsymbol{A}_t|t=1,...,T\}$ be the observed adjacency matrices specified by parameters $(M^o,\boldsymbol{\lambda}^o,\boldsymbol{\psi}^o)$. The estimator are defined by*

$$(\hat{M}_T,\hat{\boldsymbol{\lambda}}_T,\hat{\boldsymbol{\psi}}_T)=\underset{M,\boldsymbol{\lambda}\in A^M_{\epsilon_\lambda},\boldsymbol{\psi}\in\boldsymbol{\mathcal{M}}}{\arg\min}\frac{1}{T}MDL(M,\boldsymbol{\lambda},\boldsymbol{\psi})$$

Then we have

$$\hat{M}_T\xrightarrow{a.s.}M^o$$

$$\hat{\boldsymbol{\lambda}}_T\xrightarrow{a.s.}\boldsymbol{\lambda}^o$$

$$\hat{\boldsymbol{\psi}}_T\xrightarrow{a.s.}\boldsymbol{\psi}^o$$

The details of the proof can be found in Section 4.8.2. Besides, the consistency of other change point detection methods for time-evolving dynamic networks are briefly discussed in Section 4.8.3.

4.4 Practical Minimization

This section develops a practical algorithm for minimizing (4.5). The algorithm consists of two components: a bottom-up search procedure for identifying change points and a EM-Algorithm paired with variational approximation to estimate model parameters between any two adjacent change points. We first describe the EM-Algorithm.

4.4.1 EM-Algorithm with Variational Approximation

To obtain an algorithm of estimating parameters and community assignment, we follow the framework by Daudin *et al.* (2008), and view our model as a mixture model with latent discrete indicator variables Z. Suppose the number of communities Q is known, and Z_i follows a multinomial distribution $Multi(1, (\alpha_1, \alpha_2, ..., \alpha_Q))$. Let $Z_{i,q}$'s be the indicator variables that denote whether node i belongs to the qth community. Therefore, we have

$$\begin{aligned} &Z_{i,q} \in \{0, 1\} \quad i = 1, ..., N, q = 1, ..., Q \\ &\sum_{q=1}^{Q} Z_{i,q} = 1 \quad i = 1, ..., N \end{aligned}$$

The complete data log-likelihood can be derived as

$$\begin{aligned} l(A, Z) =& l(Z) + l(A|Z) \\ =& \sum_i \sum_q Z_{i,q} \log(\alpha_q) \\ &+ \frac{1}{2} \sum_{i \neq j} \sum_{q,l} \sum_t Z_{i,q} Z_{j,l} [A_{t,ij} \log(\Omega_{t,ij}) + (1 - A_{t,ij}) \log(1 - \Omega_{t,ij})] \\ =& \sum_i \sum_q Z_{i,q} \log(\alpha_q) \\ &+ \frac{1}{2} \sum_{i \neq j} \sum_{q,l} \sum_t Z_{i,q} Z_{j,l} [A_{t,ij} \log(\pi_{ql}) + (1 - A_{t,ij}) \log(1 - \pi_{ql})] \end{aligned}$$

The observed data log-likelihood $l(A)$ can be calculated by summing $l(A, Z)$ over all possible values of Z. However, it is not tractable even for moderate N (Paul and Chen, 2016). Therefore, EM algorithm will be used and unobserved Z will be treated as missing values. The problem is that in EM algorithm, it is necessary to calculate $P(Z|A)$, which is also intractable in this case.

We follow the argument of Daudin *et al.* (2008) and apply the variational approximation that aims to maximize a lower bound of $l(A)$, which is denoted as

$$J(R_A) = l(A) - KL(R_A(.), P(.|A)).$$

Here the second term denotes the Kullback-Liebler (KL) divergence between the true conditional distribution of $P(Z|A)$ and its variational approximation $R_A(.)$. $J(R_A)$ is equal to $l(A)$ if and only if $P(Z|A) = R_A(Z)$.

As $P(Z|A)$ is not tractable, we will look for its best approximation $R_A(.)$ from a certain class of distributions. Specifically, we constrain that $R_A(.)$ has the form of the product of multinomial distributions. That is to say,

$$R_A(Z) = \prod_i \prod_q \tau_{i,q}^{Z_{i,q}}.$$

For this $R_A(.)$, the objective function is

$$\begin{aligned} J(R_A) =& \sum_Z R_A(Z)\log(P(A,Z)) - \sum_Z R_A(Z)\log(R_A(Z)) \\ =& \sum_i \sum_q \tau_{i,q}\log(\alpha_q) + \frac{1}{2}\sum_{i\neq j}\sum_{q,l}\sum_t \tau_{i,q}\tau_{j,l}[A_{t,ij}\log(\pi_{ql}) + (1-A_{t,ij})\log(1-\pi_{ql})] \\ & - \sum_i \sum_q \tau_{i,q}\log(\tau_{i,q}) \end{aligned} \tag{4.6}$$

In the E-step, given the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$, we need to update the variational parameter $\boldsymbol{\tau}$ by maximizing $J(R_A)$, under the constraint that $\sum_q \tau_{i,q} = 1, \forall i$. It can be derived that the following fixed point relation satisfies.

$$\hat{\tau}_{i,q} \propto \alpha_q exp(\sum_{j\neq i}\sum_l \sum_t \hat{\tau}_{j,l}[A_{t,ij}\log(\pi_{ql}) + (1-A_{t,ij})\log(1-\pi_{ql})]).\ i = 1, ..., N, q = 1, ..., Q \tag{4.7}$$

In the M-step, we need to maximize $J(R_A)$ with respect to the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. $\hat{\boldsymbol{\alpha}}$ has the following closed form solution.

$$\hat{\alpha}_q = \frac{1}{N}\sum_{i=1}^{N} \tau_{i,q}.\ q = 1, ..., Q \tag{4.8}$$

And the close form for $\hat{\pi}_{ql}$'s are given by

$$\hat{\pi}_{ql} = \frac{\sum_{i \neq j} \sum_t \tau_{q,i} \tau_{j,l} A_{t,ij}}{\sum_{i \neq j} \sum_t \tau_{q,i} \tau_{j,l}}. \ q = 1, ..., Q, l = 1, ..., Q \tag{4.9}$$

The algorithm based on the methodology is summarized in Algorithm 4. Notice that this algorithm assumes that the number of communities is known. Therefore it is necessary to apply this algorithm for different numbers of communities, and choose the one based on the MDL principle (4.4) for the homogeneous case.

4.4.2 Search Strategy

We use backward elimination or bottom-up search to explore the space for possible change points sets. We start with M change points. At each subsequent iteration, two existing adjacent time intervals are merged together in a locally optimal way (with respect to MDL (4.5)), until there is only a single time interval left, in another word, no change point. The homogeneous model for each time interval can be estimated by Algorithm 4. The overall computational time complexity for backward elimination is $O(M)$. See Hulovatyy and Milenković (2016) Supplementary 2 for more details.

4.5 Simulation Experiments

We evaluate the performance of the proposed method on simulated datasets. We compare the proposed method with two other methods. One is by Cheung *et al.* (2020), and the other one is the SCOUT algorithm (with BIC as the criterion) by Hulovatyy and Milenković (2016). We choose these two methods for comparison because they outperform many other methods, as reported in Hulovatyy and Milenković (2016) and Cheung *et al.* (2020). Like the proposed method, these two methods are able to automatically select the number of change points.

Apart from comparing the performance of change point detection, we also want to evaluate the result of community detection. Normalized mutual information (NMI) can be used as a criterion to evaluate the result of clustering for one single network. NMI ranges from 0 to 1, where 1 means the fitted community assignment and the true assignment are perfectly matched. Specifically, for true community assignment $\boldsymbol{c}$ and fitted community assignment

$\hat{\boldsymbol{c}}$, the NMI is defined as

$$\mathrm{NMI}(\hat{\boldsymbol{c}}, \boldsymbol{c}) = \frac{I(\hat{\boldsymbol{c}}, \boldsymbol{c})}{[H(\hat{\boldsymbol{c}}) + H(\boldsymbol{c})]/2}$$

We denote

$$V_q = \{1 \leq i \leq N | c_i = q\}$$

to be the set of nodes that belong to community q. And

$$\hat{V}_l = \{1 \leq i \leq N | \hat{c}_i = l\}$$

to be the set of nodes that are estimated to be in community l. Then $H(.)$ (entropy) and $I(.)$ (mutual information) are defined as

$$\begin{aligned} H(\boldsymbol{c}) &= -\sum_q P(V_q) \log(P(V_q)) \\ &= -\sum_q \frac{|V_q|}{N} \log(\frac{|V_q|}{N}) \end{aligned}$$

$$\begin{aligned} I(\hat{\boldsymbol{c}}, \boldsymbol{c}) &= \sum_q \sum_l P(V_q \cap \hat{V}_l) \log(\frac{P(V_q \cap \hat{V}_l)}{P(V_q) \cap P(\hat{V}_l)}) \\ &= \sum_q \sum_l \frac{|V_q \cap \hat{V}_l|}{N} \log(\frac{N|V_q \cap \hat{V}_l|}{|V_q||\hat{V}_l|}) \end{aligned}$$

For a sequence of networks, two different metrics are developed based on NMI. The first metric is defined as the overall average NMI of all the networks.

$$\mathrm{NMI}_{\mathrm{avg}} = \frac{1}{T} \sum_t \mathrm{NMI}(\hat{\boldsymbol{c}}_t, \boldsymbol{c}_t)$$

where $\boldsymbol{c}_t = \boldsymbol{c}^{(m)}$ when $t_{m-1} + 1 \leq t \leq t_m$ and $\hat{\boldsymbol{c}}_t = \hat{\boldsymbol{c}}^{(k)}$ when $\hat{t}_{k-1} + 1 \leq t \leq \hat{t}_k$. However, if a fitted interval contains two or more true intervals, the fitted community assignment is not comparable with these true assignments. Therefore, we only consider the time points such that the fitted interval is totally within a true interval.

$$\mathcal{T}^* = \{t \in [\hat{t}_{k-1} + 1, \hat{t}_k], \forall k | \exists m, s.t. [\hat{t}_{k-1} + 1, \hat{t}_k] \subseteq [t_{m-1} + 1, t_m]\}$$

And the adjusted average NMI is defined as

$$\mathrm{NMI}_{\mathrm{adj}} = \frac{1}{|\mathcal{T}^*|} \sum_{t \in \mathcal{T}^*} \mathrm{NMI}(\hat{\boldsymbol{c}}_t, \boldsymbol{c}_t)$$

4.5.1 Setting 1

The number of node is set to be $N = 300$. And $T = 30$ for this and the following settings. For each time interval, the community assignment for each node is drawn from a multinomial distribution with given community size ratio. The link probabilities tell the probabilities of the existence of edges in each network within the interval, and the networks are independent. Here P_W denotes the probability for an edge within a community, and P_B denotes that for an edge between two communities. See Table 4.1 for more details.

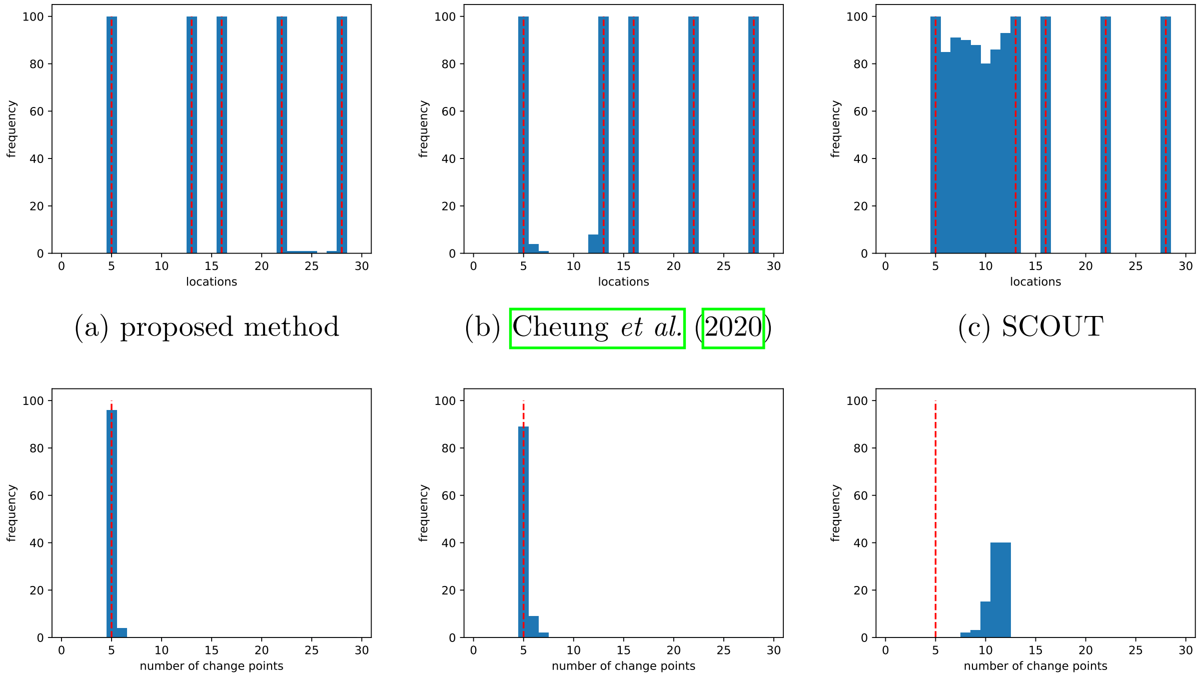
Figure 4.1 show the histograms of the estimated change point locations for Settings 1, as well as the numbers of fitted change points.

segment	interval	community size ratio	link probabilities
1	[0, 5)	1/3, 1/3, 1/3	$P_W = 0.9$, $P_B = 0.1$
2	[5, 13)	1	$P_W = 0.7$, $P_B = 0.2$
3	[13, 16)	1/4, 1/4, 1/4, 1/4	$P_W = 0.85$, $P_B = 0.15$
4	[16, 22)	2/3, 1/3	$P_W = 0.84$, $P_B = 0.2$
5	[22, 28)	3/10, 2/10, 2/10, 2/10, 1/10	$P_W = 0.8$, $P_B = 0.15$
6	[28, 30)	4/10, 3/10, 3/10	$P_W = 0.9$, $P_B = 0.1$

Table 4.1: Specification for Setting 1.

4.5.2 Setting 2

In this setting, the number of node is $N = 300$. Here we allow that within a time interval, the link probabilities can be different among networks. For all networks, P_W and P_B follow Uniform distributions. See Table 4.2 for more details of the setting. See Figure 4.2 for histograms about the fitted change points.



(a) proposed method (b) Cheung *et al.* (2020) (c) SCOUT

(d) proposed method (e) Cheung *et al.* (2020) (f) SCOUT

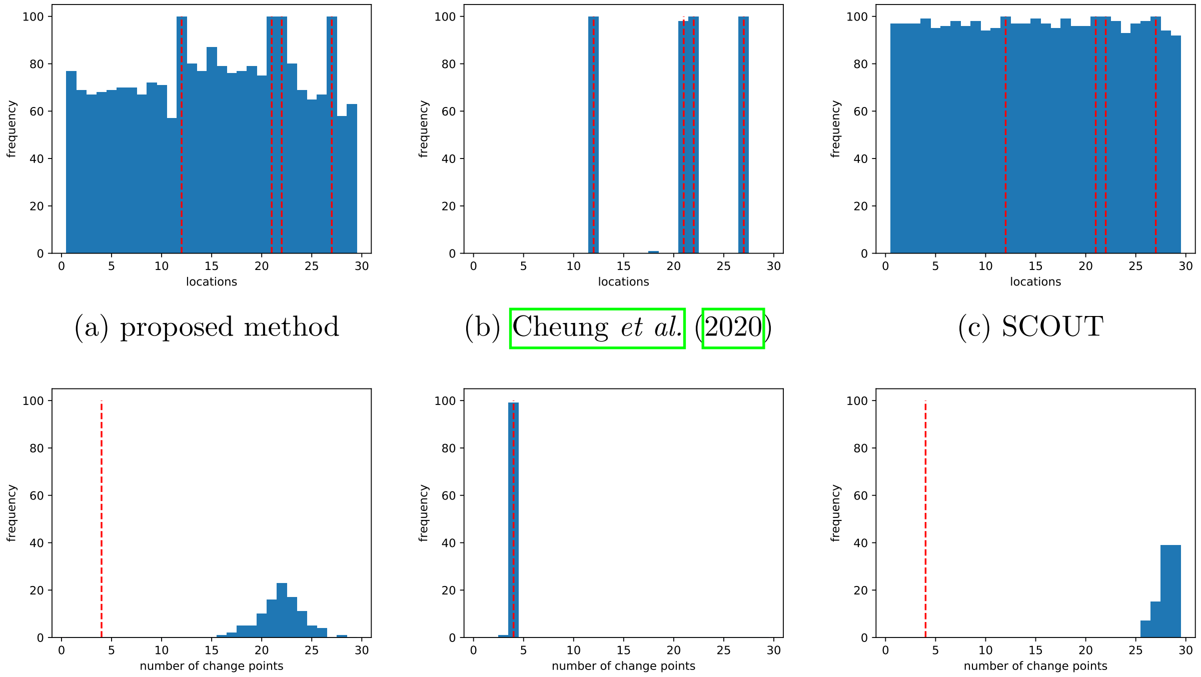
Figure 4.1: (a)-(c) Histograms of the frequency of detected change points with setting 1 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

segment	interval	community size ratio	link probabilities
1	[0, 12)	1/3, 1/3, 1/3	$P_W \sim U(0.7, 0.95),\ P_B \sim U(0.05, 0.3)$
2	[12, 21)	2/3, 1/3	$P_W \sim U(0.7, 0.95),\ P_B \sim U(0.05, 0.3)$
3	[21, 22)	3/4, 1/4	$P_W \sim U(0.7, 0.95),\ P_B \sim U(0.05, 0.3)$
4	[22, 27)	4/10, 3/10, 3/10	$P_W \sim U(0.7, 0.95),\ P_B \sim U(0.05, 0.3)$
5	[27, 30)	3/10, 3/10, 2/10, 2/10	$P_W \sim U(0.7, 0.95),\ P_B \sim U(0.05, 0.3)$

Table 4.2: Specification for Setting 2.

4.5.3 Setting 3

In this setting, the number of node is $N = 400$. Compared with the previous setting, we decrease the variation of link probabilities. See Table 4.3 for more details of the setting. See Figure 4.3 for histograms about the fitted change points.



(a) proposed method (b) Cheung *et al.* (2020) (c) SCOUT

(d) proposed method (e) Cheung *et al.* (2020) (f) SCOUT

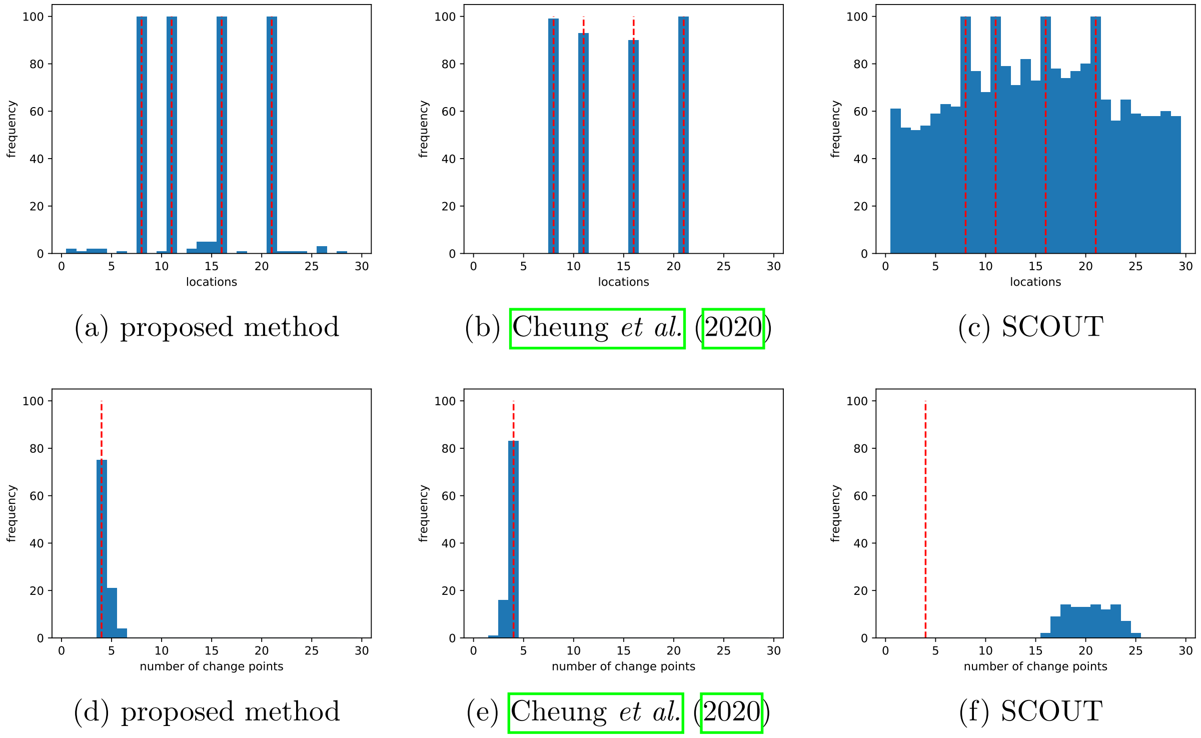
Figure 4.2: (a)-(c) Histograms of the frequency of detected change points with setting 2 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

segment	interval	community size ratio	link probabilities
1	[0, 8)	1/3, 1/3, 1/3	$P_W \sim U(0.35, 0.4),\ P_B \sim U(0.05, 0.1)$
2	[8, 11)	3/4, 1/4	$P_W \sim U(0.35, 0.4),\ P_B \sim U(0.05, 0.1)$
3	[11, 16)	1/2, 1/2	$P_W \sim U(0.35, 0.4),\ P_B \sim U(0.05, 0.1)$
4	[16, 21)	3/4, 1/4	$P_W \sim U(0.35, 0.4),\ P_B \sim U(0.05, 0.1)$
5	[21, 30)	4/10, 3/10, 3/10	$P_W \sim U(0.35, 0.4),\ P_B \sim U(0.05, 0.1)$

Table 4.3: Specification for Setting 3.

4.5.4 Setting 4

We set the number of node to be $N = 400$. The link probabilities are modified such that the separations between communities are less clear. See Table 4.4 for more details of the setting. See Figure 4.4 for histograms about the fitted change points.



(a) proposed method (b) Cheung *et al.* (2020) (c) SCOUT

(d) proposed method (e) Cheung *et al.* (2020) (f) SCOUT

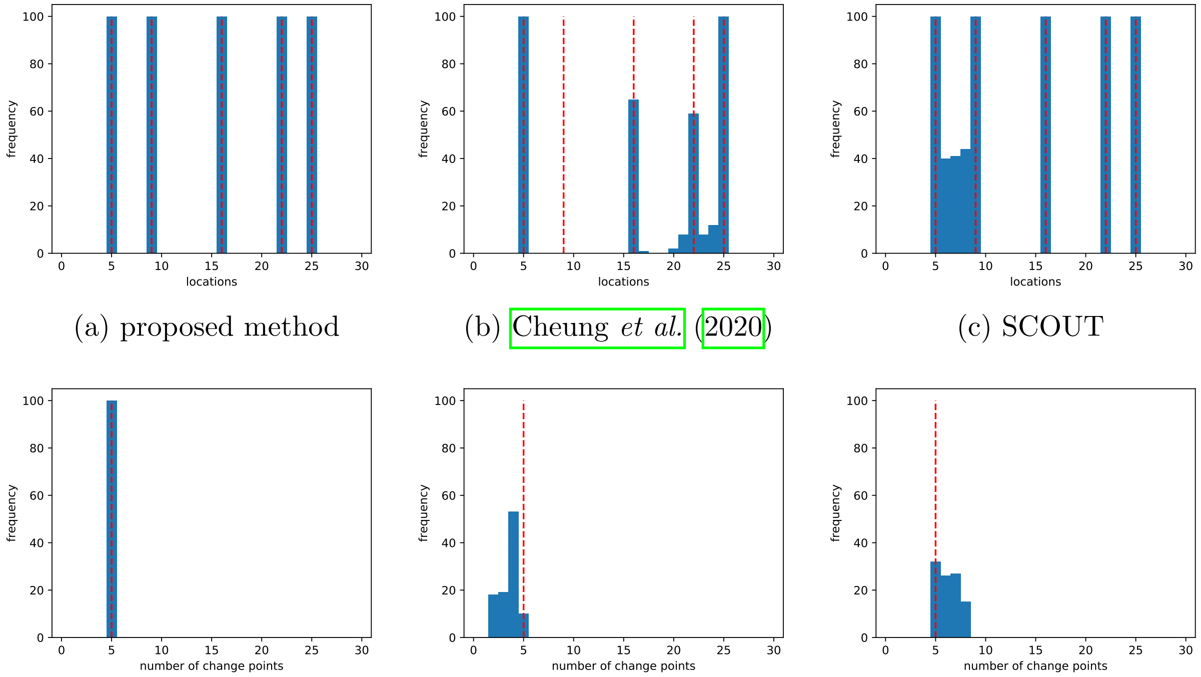
Figure 4.3: (a)-(c) Histograms of the frequency of detected change points with setting 3 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

segment	interval	community size ratio	link probabilities
1	[0, 5)	1/3, 1/3, 1/3	$P_W = 0.7,\ P_B = 0.6$
2	[5, 9)	3/4, 1/4	$P_W = 0.2,\ P_B = 0.1$
3	[9, 16)	1/4, 1/4, 1/4, 1/4	$P_W = 0.5,\ P_B = 0.3$
4	[16, 22)	1/2, 1/2	$P_W = 0.2,\ P_B = 0.1$
5	[22, 25)	1/5, 1/5, 1/5, 1/5, 1/5	$P_W = 0.4,\ P_B = 0.15$
6	[25, 30)	1/2, 1/2	$P_W = 0.7,\ P_B = 0.55$

Table 4.4: Specification for Setting 4.

4.5.5 Setting 5

The number of node is $N = 400$. Compared with the previous setting, we add some variations into the link probabilities. See Table 4.5 for more details of the setting. See Figure 4.5 for



(a) proposed method

(b) Cheung *et al.* (2020)

(c) SCOUT

(d) proposed method

(e) Cheung *et al.* (2020)

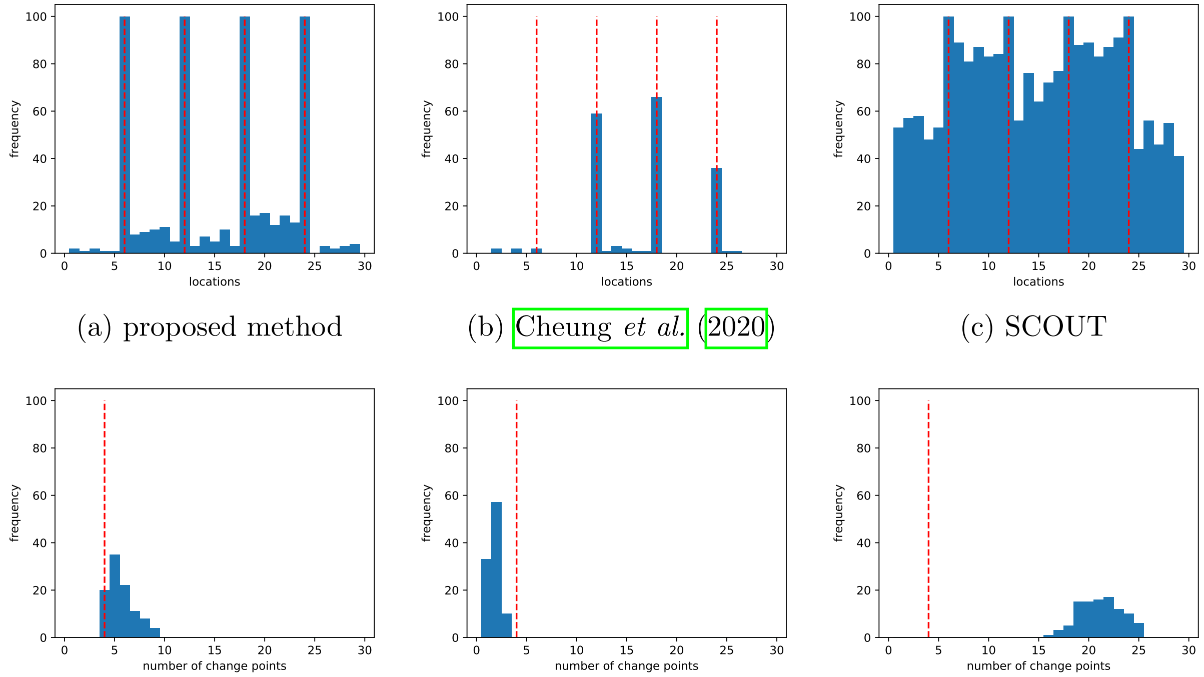
(f) SCOUT

Figure 4.4: (a)-(c) Histograms of the frequency of detected change points with setting 4 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

histograms about the fitted change points.

segment	interval	community size ratio	link probabilities
1	[0, 6)	1/4, 1/4, 1/4, 1/4	$P_W \sim U(0.2, 0.3)$, $P_B \sim U(0.05, 0.1)$
2	[6, 12)	1/2, 1/2	$P_W \sim U(0.45, 0.55)$, $P_B \sim U(0.25, 0.35)$
3	[12, 18)	2/4, 1/4, 1/4	$P_W \sim U(0.15, 0.25)$, $P_B \sim U(0.05, 0.1)$
4	[18, 24)	2/3, 1/3	$P_W \sim U(0.4, 0.5)$, $P_B \sim U(0.2, 0.3)$
5	[24, 30)	1/4, 1/4, 1/4, 1/4	$P_W \sim U(0.15, 0.25)$, $P_B \sim U(0.05, 0.1)$

Table 4.5: Specification for Setting 5.



(a) proposed method (b) Cheung *et al.* (2020) (c) SCOUT

(d) proposed method (e) Cheung *et al.* (2020) (f) SCOUT

Figure 4.5: (a)-(c) Histograms of the frequency of detected change points with setting 5 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

4.5.6 Setting 6

The number of node is $N = 400$. This time the link probabilities are deterministic functions of time t. Specifically,

$$P_W = 0.8\text{expit}(-1 + 5\text{expit}(0.1t - 2))$$

$$P_B = 0.8\text{expit}(-2.5 + 5\text{expit}(0.1t - 2))$$

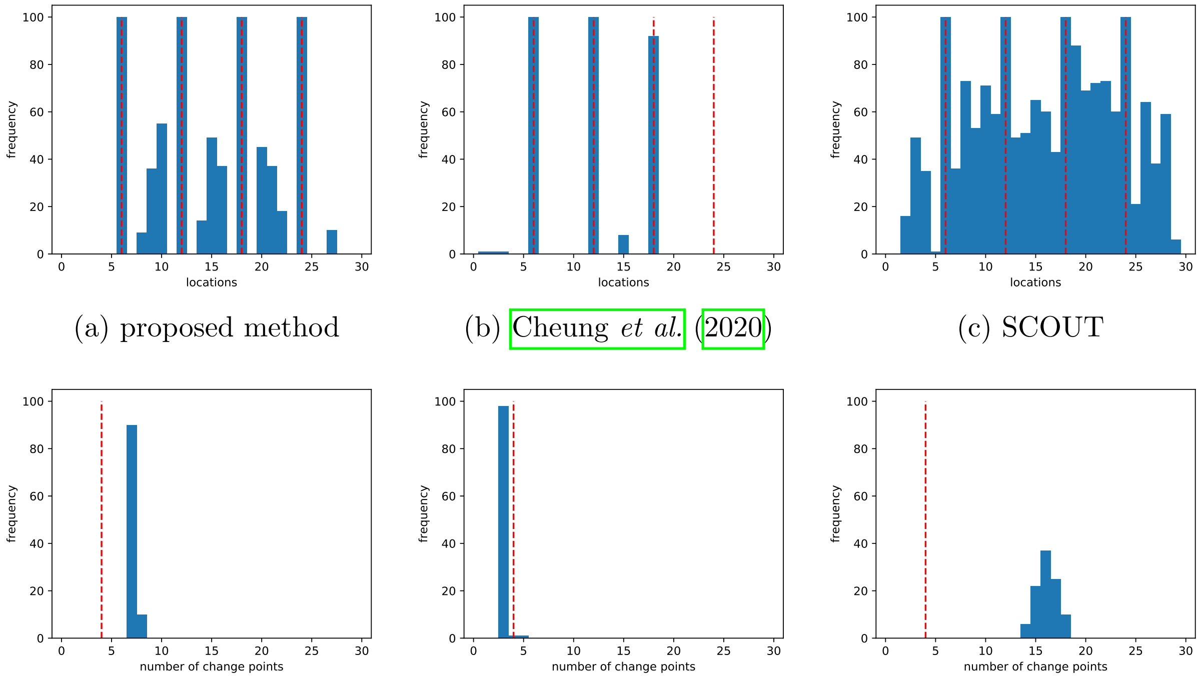
where

$$\text{expit}(x) = \frac{e^x}{1 + e^x}.$$

See Table 4.6 for more details of the setting. See Figure 4.6 for histograms about the fitted change points.

segment	interval	community size ratio
1	[0, 6)	1/4, 1/4, 1/4, 1/4
2	[6, 12)	1/2, 1/2
3	[12, 18)	2/4, 1/4, 1/4
4	[18, 24)	2/3, 1/3
5	[24, 30)	1/4, 1/4, 1/4, 1/4

Table 4.6: Specification for Setting 6.



(a) proposed method

(b) Cheung *et al.* (2020)

(c) SCOUT

(d) proposed method

(e) Cheung *et al.* (2020)

(f) SCOUT

Figure 4.6: (a)-(c) Histograms of the frequency of detected change points with setting 8 over 100 trails, with red dotted vertical lines representing true change points. (d)-(f) Histograms of the frequency of the number of detected change points over 100 trails, with red dotted vertical lines representing true number of change points.

4.5.7 Summary

Notice that both the proposed method and the SCOUT algorithm assume that the link probabilities keep the same for networks within the same interval, while the method proposed by Cheung *et al.* (2020) allows the link probabilities to vary over time, which means within the same interval, the model is actually MLSBM by Han *et al.* (2015). Therefore, the definitions of change point have some subtle differences.

One can find that our proposed method outperforms the rest two methods when the simulated datasets are generated from the setting that the link probabilities keep the same within an interval. (Setting 1, 4) When the variations of the link probabilities are limited (Setting 3,5,6), the proposed method performs better than the SCOUT algorithm as it has much less false positive (fake change points). Compared with Cheung *et al.* (2020), the proposed method is able to detect all the true change points with acceptable false positive, while the method by Cheung *et al.* (2020) tends to miss the true change points. While when the variations of the link probabilities are quite large (Setting 2), the method by Cheung *et al.* (2020) performs better than the rest methods because its assumptions are concordant to the setting.

In terms of the performance of community detection, based on Table 4.7, one can find that the performed method outperforms the rest two methods.

Settings	average NMI			adjusted average NMI		
1	0.998	0.996	0.733	0.998	0.996	0.733
2	1.000	0.999	1.000	1.000	1.000	1.000
3	1.000	0.494	1.000	1.000	0.504	1.000
4	0.999	0.551	0.980	0.999	0.809	0.980
5	0.993	0.193	0.965	0.993	0.299	0.965
6	1.000	0.583	1.000	1.000	0.552	1.000

Table 4.7: The average NMI and adjusted average NMI for all the settings. The first column denote the settings. The columns 2-4 show the average NMI for the proposed method, the method by Cheung *et al.* (2020), and the SCOUT. The columns 5-7 present the adjusted average NMI for the proposed method, the method by Cheung *et al.* (2020), and the SCOUT.

4.6 Applications

We compare the three methods on these real datasets. For some of the datasets, the ground truths of the change points are available. In that case, we use precision, recall and F_1 score to evaluate the performances for different methods.

4.6.1 Enron Email Network

This dataset contains email communication of about 150 employees of the Enron corporation from May 1999 to June 2002. [1] The nodes correspond to employees, and there is an edge between two nodes if one employee sent at least one email to another. After the data cleaning process based on the strategy in Zhou *et al.* (2007), we apply the proposed method on the monthly snapshots. The ground truth of change points can be found as a list of company-related events in Peel and Clauset (2015).

Method	Precision	Recall	F1 Score
Proposed Method	1	0.174	0.296
Cheung *et al.* (2020)	0	0	0
SCOUT	0	0	0

Table 4.8: Result for Enron.

The method by Cheung *et al.* (2020) and SCOUT fail to detect any change point, while the proposed method is able to detect some true change points.

4.6.2 AMD Hope

This dataset contains information about time-location of 748 attendees of The Last HOPE conference in 2008. [2] There is an edge between two attendees if they were at the same location at that snapshot (every 15 minutes.) The ground truth of change points that correspond to the talks and social events is available. [3]

Method	Precision	Recall	F1 Score
Proposed Method	0.191	0.971	0.319
Cheung *et al.* (2020)	0	0	0
SCOUT	0.3	0.088	0.136

Table 4.9: Result for AMD Hope.

[1] The cleaned version was obtained from `https://data.world/brianray/enron-email-dataset`, and the original dataset can be downloaded from `www.cs.cmu.edu/~enron/`

[2] The information of the conference can be found in `https://vii.hope.net/`, and the dataset is available on `https://crawdad.org/hope/amd/20080807/`

[3] The schedule of the conference can be found in `https://vii.hope.net/Schedule-FullPage.pdf`

The method by Cheung *et al.* (2020) detects no change point. The number of change points detected by SCOUT is much less than the true number, while the proposed method has many false positive cases.

4.6.3 Reality Mining Network

This dataset contains information about proximity inferred from repeated Bluetooth scans among university students and faculty during 2004-2005 academic year (Eagle *et al.*, 2009). [4] Weekly snapshots are used for analysis. The ground truth change points can be found in Peel and Clauset (2015) as a list of events from the academic calendar.

Method	Precision	Recall	F1 Score
Proposed Method	0.417	0.333	0.370
Cheung *et al.* (2020)	0.333	0.067	0.111
SCOUT	0.667	0.133	0.222

Table 4.10: Result for Reality Mining.

The proposed method has lower precision than SCOUT, but it has the best recall. Based on the F_1 score, the proposed method performed best among all these three methods.

4.7 Conclusion

In this paper, we proposed a method for simultaneous change point detection and community assignment detection for time-evolving dynamic networks. We developed an MDL criterion for model selection, and developed a practical algorithm for finding this minimizer. It is shown that the MDL criterion yields statistically consistent estimates, and simulation experiments suggest that the method also performs well in practice.

For the future work, one possible extension is to relax the constant assumption of link probabilities. By allowing link probabilities to vary over time, the model will be able to accommodate more kinds of linkage trends. The main challenge is to verify the statistical properties of the relaxed model.

[4] The dataset was downloaded from `http://realitycommons.media.mit.edu/realitymining4.html`

4.8 Supplement: Technical Details

4.8.1 Derivation of MDL

4.8.1.1 Homogeneous Case

When there is no change point, these T observed network follow the proposed homogeneous model (4.1). In this case, $\mathcal{F} = \{\boldsymbol{\psi}\}$. So $\mathrm{CL}(\mathcal{F})$ can be written as

$$\mathrm{CL}(\mathcal{F}) = \mathrm{CL}(\boldsymbol{\psi}).$$

The parameter set $\boldsymbol{\psi}$ is composed of the community assignments, as well as the parameters that determine the link probabilities for each homogeneous time interval. That is to say,

$$\mathrm{CL}(\boldsymbol{\psi}) = \mathrm{CL}(\boldsymbol{c}) + \mathrm{CL}(\boldsymbol{\gamma}|\boldsymbol{c}).$$

According to Rissanen (1989a), it takes approximately $\log(I)$ bits to encode an integer I with upper bound unknown, and approximately $\log(I_u)$ bits with a known upper bound I_u. To partition the node set of size N into non-overlapping communities, we need

$$\mathrm{CL}(\boldsymbol{c}) = \log_2(Q) + N\log_2(Q),$$

where the first term encodes the number of communities and the second term encodes the community assignment for each node. Here $Q = |\boldsymbol{c}|$ denotes the number of communities.

The code length of encoding a maximum likelihood estimate of a parameter computed from n observation is shown to be $\frac{1}{2}\log_2(n)$ (Rissanen, 1989a). In this case,

$$\mathrm{CL}(\boldsymbol{\gamma}|\boldsymbol{c}) = \sum_{q\leq l}\frac{1}{2}\log_2(N_{ql}(\boldsymbol{c})T)$$

.

Now combining the previous parts, we have

$$\mathrm{CL}(\mathcal{F}) = (N+1)\log_2(Q) + \sum_{q\leq l}\frac{1}{2}\log_2(N_{ql}(\boldsymbol{c})T)$$

Finally, we calculate the last term $\mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F})$, which is given by the negative log (base 2) of the likelihood of the fitted model (Rissanen, 1989a). With the assumption that given the model $\mathcal{F}$, $A_{t,ij}$ follows a Bernoulli distribution,

$$\mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F}) = -\sum_{t=1}^{T}\sum_{i<j}\boldsymbol{A}_{t,ij}\log_2(\hat{\Omega}_{t,ij}) + (1-\boldsymbol{A}_{t,ij})\log_2(1-\hat{\Omega}_{t,ij}),$$

where $\hat{\Omega}_{t,ij}$ is determined by equation (4.1) given $\mathcal{F}$.

The overall code length is

$$\begin{aligned}\mathrm{CL}(\text{“data”}) =& \mathrm{CL}(\mathcal{F}) + \mathrm{CL}(\hat{\mathcal{E}}|\mathcal{F})\\ =&(N+1)\log_2(Q) + \sum_{q\leq l}\frac{1}{2}\log_2(N_{ql}(\boldsymbol{c})T)\\ &-\sum_{t=1}^{T}\sum_{i<j}\boldsymbol{A}_{t,ij}\log_2(\hat{\Omega}_{t,ij}) + (1-\boldsymbol{A}_{t,ij})\log_2(1-\hat{\Omega}_{t,ij})\end{aligned}$$

Because all the terms have logarithm to the base 2, we can replace all of the $\log_2$'s with log's, which gives the MDL in homogeneous case (4.4).

4.8.1.2 Heterogeneous Case

To encode change points $\mathcal{T}$, one need to first encode the number of the change points and then the actual locations of them. To encode M, the number of change points, the code length is $\log(M+1)$, where the additional 1 is used to distinguish $M=0$ and $M=1$. The locations of change points $\boldsymbol{\tau}$ can be encoded by using the length of each of the time intervals. Therefore combining these two parts we have

$$\mathrm{CL}(\mathcal{T}) = \log_2(M+1) + \sum_{m=1}^{M+1}\log_2(t_m - t_{m-1}).$$

Once $\mathcal{T}$ is encoded, for each time interval, it becomes the homogeneous case. Using similar arguments as before, we can derive the overall MDL (4.5).

4.8.2 Statistical Consistency

For a given class assignment $\boldsymbol{c}$, let

$$E_{t,ql}^{(m)}(\boldsymbol{c}^{(m)}) = \{\begin{matrix}\sum_{c_i^{(m)}=q,c_j^{(m)}=l}\boldsymbol{B}_{t,ij}^{(m)} & q\neq l\\ \sum_{c_i^{(m)}=q,c_j^{(m)}=l}\boldsymbol{B}_{t,ij}^{(m)}/2 & q=l\end{matrix} \tag{4.10}$$

be the number of observed edges in a block in the mth interval.

For each $1\leq t\leq T_m$, the log-likelihood function for community assignment vector $\boldsymbol{c}^{(m)}$ and parameters $\boldsymbol{\pi}^{(m)}$ is

$$
\begin{aligned}
l_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_t^{(m)}) &= \sum_{i<j} \boldsymbol{B}_{t,ij}^{(m)} \log(\Omega^{(m)}_{t,c_i^{(m)} c_j^{(m)}}) + (1 - \boldsymbol{B}_{t,ij}^{(m)}) \log(1 - \Omega^{(m)}_{t,c_i^{(m)} c_j^{(m)}}) \\
&= \sum_{p\leq q} E_{t,ql}^{(m)}(\boldsymbol{c}^{(m)}) \log(\pi_{ql}^{(m)}) + (N_{ql}(\boldsymbol{c}^{(m)}) - E_{t,ql}^{(m)}(\boldsymbol{c}^{(m)})) \log(1 - \pi_{ql}^{(m)})
\end{aligned}
$$

Notice that $1 \leq p \leq q \leq |\boldsymbol{c}^{(m)}|$, where $|\boldsymbol{c}^{(m)}|$ denotes the number of communities in the mth interval.

Define $\psi_m = (\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)})$ to be the parameter set for the mth interval, and $\boldsymbol{\mathcal{M}}$ to be the class of models ψ_m can take value from. Then the log-likelihood for the mth interval can be written as

$$
L_T^{(m)}(\psi_m; \boldsymbol{B}^{(m)}) = \sum_{t=1}^{T_m} l_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_t^{(m)})
$$

Denote $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_M)$ as the normalized change point location vector, and $\boldsymbol{\psi} = (\psi_1, ..., \psi_{M+1})$ to be the parameter vector. Then vector $(M, \boldsymbol{\lambda}, \boldsymbol{\psi})$ can specify a model for this sequence of networks. The MDL is derived to be

$$
\begin{aligned}
\text{MDL}(M, \boldsymbol{\lambda}, \boldsymbol{\psi}) =& \log(M+1) + \sum_{m=1}^{M+1} \log([T\lambda_m] - [T\lambda_{m-1}]) + \sum_{k=1}^{m+1} (N+1) \log(|\boldsymbol{c}^{(m)}|) \\
&+ \sum_{m=1}^{M+1} \sum_{p\leq q} \frac{1}{2} \log(([T\lambda_m] - [T\lambda_{m-1}]) N_{ql}(\boldsymbol{c}^{(m)})) - \sum_{m=1}^{M+1} L_T^{(m)}(\psi_m; \boldsymbol{B}^{(m)})
\end{aligned}
$$

In order to make sure that the change points are identifiable, we assume that there exists a $\epsilon_\lambda > 0$ such that $\min_{1\leq m\leq M+1} |\lambda_m - \lambda_{m-1}| > \epsilon_\lambda$. Therefore, the number of change points is bounded by $M \leq M_u = [1/\epsilon_\lambda] + 1$. And there exists a constraint $A^M_{\epsilon_\lambda}$ of $\boldsymbol{\lambda}$ where

$$
A^M_{\epsilon_\lambda} = \{\boldsymbol{\lambda} \in (0,1)^M | 0 < \lambda_1 < ... < \lambda_M < 1, \lambda_m - \lambda_{m-1} > \epsilon_\lambda, \forall 1 \leq m \leq M+1\}
$$

Then the estimation of the model based on MDL is given by

$$
(\hat{M}_T, \hat{\boldsymbol{\lambda}}_T, \hat{\boldsymbol{\psi}}_T) = \arg \min_{M\leq M_u, \boldsymbol{\lambda}\in A^M_{\epsilon_\lambda}, \boldsymbol{\psi}\in\boldsymbol{\mathcal{M}}} \frac{1}{T} \text{MDL}(M, \boldsymbol{\lambda}, \boldsymbol{\psi})
$$

Here $\hat{\boldsymbol{\lambda}}_T = (\hat{\lambda}_1, ..., \hat{\lambda}_{\hat{M}})$ and $\hat{\boldsymbol{\psi}}_T = (\hat{\psi}_1, ..., \hat{\psi}_{\hat{M}+1})$, where $\hat{\psi}_m = (\hat{\boldsymbol{c}}^{(m)}, \hat{\boldsymbol{\pi}}_T^{(m)})$. And $\hat{\boldsymbol{\pi}}_T^{(m)}$ is defined as

$$\hat{\boldsymbol{\pi}}_T^{(m)} = \arg \max_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\hat{\boldsymbol{c}}^{(m)})} L_T^{(m)}((\hat{\boldsymbol{c}}^{(m)}, \boldsymbol{\pi}^{(m)}); \hat{\boldsymbol{B}}^{(m)})$$

with $\hat{\boldsymbol{B}}^{(m)} = \{\boldsymbol{A}_t | [T\hat{\lambda}_{m-1}] \le t < [T\hat{\lambda}_m]\}$ denotes the estimated mth interval of the sequence of graphs, and $\Theta_m(\hat{\boldsymbol{c}}^{(m)})$ is the parameter space of $\boldsymbol{\pi}^{(m)}$ given $\hat{\boldsymbol{c}}^{(m)}$.

We further define the log-likelihood formed by a portion of the mth interval by

$$L_T^{(m)}(\psi_m, \lambda_d, \lambda_u; \boldsymbol{B}^{(m)}) = \sum_{t=[T_m\lambda_d]}^{[T_m\lambda_u]-1} l_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_t^{(m)})$$

where $0 \le \lambda_d < \lambda_u \le 1$ and $\lambda_u - \lambda_d > \epsilon_\lambda$.

We denote

$$\sup_{\lambda_d, \lambda_u} := \sup_{0 \le \lambda_d < \lambda_u \le 1, \lambda_u - \lambda_d > \epsilon_\lambda}$$

to simplify the notation.

In this setting, an extension need to be made such that λ_d and λ_u can be slightly outside $[0, 1]$. It means that the mth estimated interval could cover a part of the observations that belong to the $(m-1)$th and $(m+1)$th true intervals. Based on the formula (3.4) in Davis and Yau (2013), for a real-value function $f_T(\lambda_d, \lambda_u)$ on $\mathcal{R}^2$,

$$\sup_{\underline{\lambda_d}, \overline{\lambda_u}} f_T(\lambda_d, \lambda_u) \xrightarrow{a.s.} 0 \tag{4.11}$$

is used to denote

$$\sup_{-h_T < \lambda_d < \lambda_u < 1 + r_T, \lambda_u - \lambda_d > \epsilon_\lambda} f_T(\lambda_d, \lambda_u) \xrightarrow{a.s.} 0$$

for any pre-specified positive-valued sequences h_T and r_T, which cover to 0 as $T \to \infty$.

The following assumptions on class link probabilities $\boldsymbol{\pi}^{o(m)}, 1 \le m \le (M+1)$ make sure the quality of the community estimation.

Assumption 4.1.

$$C_0 := \inf_{m,q,l} (\pi_{ql}^{o(m)}, 1 - \pi_{ql}^{o(m)}) > 0 \tag{4.12}$$

Assumption 4.2. *Let*

$$\sigma(x) := x \log(x) + (1-x)\log(1-x) \tag{4.13}$$

then

$$\delta := \inf_{m,q,l} \max_r \sigma(\pi_{qr}^{o(m)}) + \sigma(\pi_{lr}^{o(m)}) - 2\sigma(\frac{\pi_{qr}^{o(m)} + \pi_{lr}^{o(m)}}{2}) > 0 \tag{4.14}$$

Based on the assumptions and the format of the log-likelihood function, the following propositions can be derived.

Proposition 4.1 (v). *For* $m = 1, ..., M+1$ *and any fixed* $\boldsymbol{c}^{(m)}$*, there exists a* $\epsilon > 0$ *such that,*

$$\begin{aligned} \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & E|l_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})|^{v+\epsilon} < \infty \\ \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & E|l'_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})|^{v+\epsilon} < \infty \\ \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & E|l''_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})| < \infty \end{aligned} \tag{4.15}$$

This proposition holds for $v = 1, 2, 4$ due to the compactness of parameter space and bounded $E_{t,pq}^{(m)}(\boldsymbol{c}^{(m)})$.

Proposition 4.2. *For* $m = 1, ..., M+1$ *and any fixed* $\boldsymbol{c}^{(m)}$*,*

$$\begin{aligned} \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & |\frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}^{(m)}) - L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0 \\ \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & |\frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{\prime(m)}((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}^{(m)}) - L'_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0 \\ \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_m(\boldsymbol{c}^{(m)})} & |\frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{\prime\prime(m)}((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}^{(m)}) - L''_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0 \end{aligned} \tag{4.16}$$

where

$$\begin{aligned} L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)})) &\coloneqq E(l_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})) \\ L'_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)})) &\coloneqq E(l'_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})) \\ L''_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)})) &\coloneqq E(l''_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}); \boldsymbol{B}_1^{(m)})) \end{aligned} \tag{4.17}$$

The proof is trivial given the assumptions.

The estimated locations of change points are used to define the likelihood in practice. Therefore, the two ends of the mth interval might contain observations from the $(m-1)$th and $(m+1)$th true intervals, though the estimated change points are close to the true change points. It is necessary to control the effect at the two ends of the fitted interval.

Proposition 4.3 (w)**.** *For $m = 1, ..., M+1$ and any fixed ψ and any sequence of integers $\{g(T)\}_{T\geq 1}$ that satisfies $g(T) > cT^w$ for some $c > 0$ when T is large enough, then*

$$\begin{aligned} \frac{1}{g(T)} \sum_{t=T-g(T)+1}^{T} l_m(\psi; \boldsymbol{B}_t^{(m)})) &\xrightarrow{a.s.} E(l_m(\psi; \boldsymbol{B}_t^{(m)}))) \\ \frac{1}{g(T)} \sum_{t=T-g(T)+1}^{T} l'_m(\psi; \boldsymbol{B}_t^{(m)})) &\xrightarrow{a.s.} E(l'_m(\psi; \boldsymbol{B}_t^{(m)}))) \end{aligned} \tag{4.18}$$

Based on the Lemma 1 in Davis and Yau (2013), Proposition 4.3 holds when Proposition 4.1(2) holds and the Assumption 4* in Davis and Yau (2013) is satisfied. And Assumption 4* is satisfied because an independent process, like the current setting, must be mixing.

It is necessary to discuss the identifiability of models in $\boldsymbol{\mathcal{M}}$. We can define $\boldsymbol{c}^b$ a bigger model than $\boldsymbol{c}^s$ if $c_i^b = c_j^b$ leads to $c_i^s = c_j^s$. Equivalently, there exists a function $g\colon c_i^b \to c_i^s$.

Proposition 4.4. *For the mth interval, the true model $\psi_m^o \in \boldsymbol{\mathcal{M}}$ satisfies the condition $\psi_m^o = \arg\max_{\psi\in\boldsymbol{\mathcal{M}}} E(l_m(\psi; \boldsymbol{B}_t^{(m)})))$. Also, ψ_m^o is uniquely identifiable, which means that if there exists a $\boldsymbol{\pi}^*$ such that $l_m((\boldsymbol{c}^o, \boldsymbol{\pi}^o); \boldsymbol{B}_t^{(m)}) = l_m((\boldsymbol{c}^o, \boldsymbol{\pi}^*); \boldsymbol{B}_t^{(m)})$ almost everywhere for $\boldsymbol{B}_t^{(m)}$, then $\boldsymbol{\pi}^o = \boldsymbol{\pi}^*$. And suppose there exists another model $\psi_m^b = (\boldsymbol{c}^b, \boldsymbol{\pi}^b)$ such that $l_m(\psi_m^b; \boldsymbol{B}_t^{(m)}) = l_m(\psi_m^o; \boldsymbol{B}_t^{(m)})$ almost everywhere, then $\boldsymbol{c}^b$ must be a bigger model than $\boldsymbol{c}^o$, i.e. there exists $g\colon c_i^b \to c_i^o$. And ψ_m^o and ψ_m^b satisfies $\pi_{ql}^{b(m)} = \pi_{g(q)g(l)}^{o(m)}$.*

Proof: To lighten notations, we skip some (m)'s when there is no ambiguity in the proof. Let the underlying true model be $\psi^o = (\boldsymbol{c}^o, \boldsymbol{\pi}^o)$. Define

$$\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}) \coloneqq \frac{1}{N_{ql}(\boldsymbol{c})} \sum_{i\neq j, c_i=q, c_j=l} \boldsymbol{\pi}^o_{c_i^o, c_j^o}$$

For a special case, we have $\bar{\boldsymbol{\pi}}(\boldsymbol{c}^o) = \boldsymbol{\pi}^o$.

Let $\boldsymbol{\pi}^*$ be another link probability with $\boldsymbol{c}$ as the aommunity assignment. Then,

$$\begin{aligned} E(l_m((\boldsymbol{c}, \boldsymbol{\pi}^*); \boldsymbol{B}_t^{(m)})) &= E(\sum_{i<j} \boldsymbol{B}_{t,ij}^{(m)} \log(\boldsymbol{\pi}^*_{c_i,c_j}) + (1 - \boldsymbol{B}_{t,ij}^{(m)}) \log(1 - \boldsymbol{\pi}^*_{c_i,c_j})) \\ &= \sum_{q\leq l} \sum_{i\neq j, c_i=q, c_j=l} \boldsymbol{\pi}^o_{c_i^o,c_j^o} \log(\boldsymbol{\pi}^*_{c_i,c_j}) + \sum_{i\neq j, c_i=q, c_j=l} (1 - \boldsymbol{\pi}^o_{c_i^o,c_j^o}) \log(1 - \boldsymbol{\pi}^*_{c_i,c_j}) \\ &= \sum_{q\leq l} N_{ql}(\boldsymbol{c})[\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}) \log(\boldsymbol{\pi}^*_{ql}) + (1 - \bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})) \log(1 - \boldsymbol{\pi}^*_{ql})] \end{aligned}$$

Similarly,

$$E(l_m((\boldsymbol{c},\bar{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{B}_t^{(m)})) = \sum_{q\leq l} N_{ql}(\boldsymbol{c})[\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\log(\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})) + (1-\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))\log(1-\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))]$$

Then

$$\begin{aligned} E(l_m((\boldsymbol{c},\bar{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{B}_t^{(m)})) - E(l_m((\boldsymbol{c},\boldsymbol{\pi}^*);\boldsymbol{B}_t^{(m)})) =& \sum_{q\leq l} N_{ql}(\boldsymbol{c})[\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\log(\frac{\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})}{\boldsymbol{\pi}^*_{ql}}) \\ &+ (1-\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))\log(\frac{1-\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})}{1-\boldsymbol{\pi}^*_{ql}})] \\ =& \sum_{q\leq l} N_{ql}(\boldsymbol{c}) D_{KL}(\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})||\boldsymbol{\pi}^*_{ql}) \\ \geq& 0 \end{aligned} \tag{4.19}$$

Here $D_{KL}(\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})||\boldsymbol{\pi}^*_{ql})$ denotes the Kullback–Leibler divergence of a $Bernoulli(\bar{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))$ distribution from a $Bernoulli(\boldsymbol{\pi}^*_{ql})$ one.

And based on Lemma 1 in Han *et al.* (2015), the following result holds when a label assignment $\boldsymbol{c}$ is not equal to $\boldsymbol{c}^o$ or it is not a bigger model than $\boldsymbol{c}^o$.

$$E(l_m((\boldsymbol{c}^o,\bar{\boldsymbol{\pi}}(\boldsymbol{c}^o));\boldsymbol{B}_t^{(m)})) - E(l_m((\boldsymbol{c},\bar{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{B}_t^{(m)})) \geq \frac{1}{2}\delta \min_q n_q(\boldsymbol{c}^o) \tag{4.20}$$

where $n_q(\boldsymbol{c}^o)$ denotes the number of nodes in community q under label $\boldsymbol{c}^o$. And

$$\delta = \min_{q,l}\max_r \sigma(\boldsymbol{\pi}^o_{qr}) + \sigma(\boldsymbol{\pi}^o_{lr}) - 2\sigma(\frac{\boldsymbol{\pi}^o_{qr}+\boldsymbol{\pi}^o_{lr}}{2}) \tag{4.21}$$

here

$$\sigma(x) \coloneqq x\log(x) + (1-x)\log(1-x)$$

Finally, the following result can be derived

$$E(l_m((\boldsymbol{c}^o,\boldsymbol{\pi}^o);\boldsymbol{B}_t^{(m)})) - E(l_m((\boldsymbol{c},\boldsymbol{\pi});\boldsymbol{B}_t^{(m)})) \geq \frac{1}{2}\delta \min_q n_q(\boldsymbol{c}^o) \tag{4.22}$$

when the label assignment $\boldsymbol{c}$ is not a bigger model than $\boldsymbol{c}^o$. This finishes the proof.

Lemma 4.1. *For any fixed $\boldsymbol{c}^{(m)}$,*

$$\begin{aligned}
&\sup_{\underline{\lambda_d},\overline{\lambda_u}} \sup_{\boldsymbol{\pi}^{(m)}\in\Theta_k(\boldsymbol{c}^{(m)})} |\frac{1}{T(\lambda_m-\lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}),\lambda_d,\lambda_u;\boldsymbol{B}^{(m)}) \\
&\qquad\qquad - (\lambda_u-\lambda_d)L_m((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0 \\
&\sup_{\underline{\lambda_d},\overline{\lambda_u}} \sup_{\boldsymbol{\pi}^{(m)}\in\Theta_k(\boldsymbol{c}^{(m)})} |\frac{1}{T(\lambda_m-\lambda_{m-1})} L_T'^{(m)}((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}),\lambda_d,\lambda_u;\boldsymbol{B}^{(m)}) \\
&\qquad\qquad - (\lambda_u-\lambda_d)L_m'((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0 \\
&\sup_{\underline{\lambda_d},\overline{\lambda_u}} \sup_{\boldsymbol{\pi}^{(m)}\in\Theta_k(\boldsymbol{c}^{(m)})} |\frac{1}{T(\lambda_m-\lambda_{m-1})} L_T''^{(m)}((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}),\lambda_d,\lambda_u;\boldsymbol{B}^{(m)}) \\
&\qquad\qquad - (\lambda_u-\lambda_d)L_m''((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}))| \xrightarrow{a.s.} 0
\end{aligned} \tag{4.23}$$

See Proposition 1 and 2 in Davis and Yau (2013) for the proof.

Lemma 4.2. *Suppose the true parameters for interval k is $\psi^{o(m)} = (\boldsymbol{c}^{o(m)},\boldsymbol{\pi}^{o(m)})$. And suppose a community assignment $\boldsymbol{c}^{(m)}$ is specified for estimation. Let*

$$\begin{aligned}
\hat{\boldsymbol{\pi}}_T = \hat{\boldsymbol{\pi}}_T^{(m)}(\lambda_d,\lambda_u) &:= \arg \max_{\boldsymbol{\pi}^{(m)}\in\Theta_k(\boldsymbol{c}^{(m)})} L_T^{(m)}((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}),\lambda_d,\lambda_u;\boldsymbol{B}^{(m)}) \\
\boldsymbol{\pi}^{*(m)} &:= \arg \max_{\boldsymbol{\pi}^{(m)}\in\Theta_k(\boldsymbol{c}^{(m)})} L_m((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{(m)}))
\end{aligned}$$

Then

$$\sup_{\underline{\lambda_d},\overline{\lambda_u}} |\frac{1}{T(\lambda_m-\lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)},\hat{\boldsymbol{\pi}}_T),\lambda_d,\lambda_u;\boldsymbol{B}^{(m)}) - (\lambda_u-\lambda_d)L_m((\boldsymbol{c}^{(m)},\boldsymbol{\pi}^{*(m)}))| \xrightarrow{a.s.} 0 \tag{4.24}$$

where the supremum is defined in (4.11). And if $\boldsymbol{c}^{(m)} = \boldsymbol{c}^{o(m)}$, we further have

$$\sup_{\underline{\lambda_d},\overline{\lambda_u}} |\hat{\boldsymbol{\pi}}_T^{(m)}(\lambda_d,\lambda_u) - \boldsymbol{\pi}^{o(m)}| \xrightarrow{a.s.} 0 \tag{4.25}$$

If $\boldsymbol{c}^{(m)}$ is a bigger model than $\boldsymbol{c}^{o(m)}$, which means there exist a function g: $c_i^{(m)} \to c_i^{o(m)}$,then we have

$$\sup_{\underline{\lambda_d},\overline{\lambda_u}} |\hat{\boldsymbol{\pi}}_{T,ql}^{(m)}(\lambda_d,\lambda_u) - \boldsymbol{\pi}_{g(q)g(l)}^{o(m)}| \xrightarrow{a.s.} 0 \ \forall q, l \tag{4.26}$$

Proof:

$$\begin{aligned}
&(\lambda_u - \lambda_d)(L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{*(m)})) - L_m((\boldsymbol{c}^{(m)}, \hat{\boldsymbol{\pi}}_T)))\\
\leq &\sup_{\underline{\lambda_d},\overline{\lambda_u}} |(\lambda_u - \lambda_d)L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{*(m)})) - \frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{*(m)}), \lambda_d, \lambda_u; \boldsymbol{B}^{(m)})\\
&+ \frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)}, \hat{\boldsymbol{\pi}}_T), \lambda_d, \lambda_u; \boldsymbol{B}^{(m)}) - (\lambda_u - \lambda_d)L_m((\boldsymbol{c}^{(m)}, \hat{\boldsymbol{\pi}}_T))|\\
\leq &2 \sup_{\underline{\lambda_d},\overline{\lambda_u}} \sup_{\boldsymbol{\pi}^{(m)} \in \Theta_k(\boldsymbol{c}^{(m)})} |\frac{1}{T(\lambda_m - \lambda_{m-1})} L_T^{(m)}((\boldsymbol{c}^{(m)}, \hat{\boldsymbol{\pi}}_T), \lambda_d, \lambda_u; \boldsymbol{B}^{(m)})\\
&- (\lambda_u - \lambda_d)L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))|\\
\xrightarrow{a.s.} &0
\end{aligned} \tag{4.27}$$

The first inequity is obtained by the definition of maximum likelihood estimator, and the last convergence comes from the previous proposition. As $\boldsymbol{\pi}^{*(m)}$ maximizes $L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))$ and $\lambda_u - \lambda_d > 0$, we have

$$|L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{*(m)})) - L_m((\boldsymbol{c}^{(m)}, \hat{\boldsymbol{\pi}}_T))| \xrightarrow{a.s.} 0 \tag{4.28}$$

Combining (4.27), (4.28) and Proposition 4.1(1), (4.24) holds. If $\boldsymbol{c}^{(m)} = \boldsymbol{c}^{o(m)}$, by Proposition 4.4, $L_m((\boldsymbol{c}^{(m)}, \boldsymbol{\pi}^{(m)}))$ has a unique maximizer at $\boldsymbol{\pi}^{o(m)}$, so (4.25) holds. If $\boldsymbol{c}^{(m)}$ is a bigger model than $\boldsymbol{c}^{o(m)}$, by Proposition 4.4, (4.26) holds. This finishes the proof.

Now we give a preliminary result of the convergence when the number of change points is known.

Theorem 4.2. *Let $\{\boldsymbol{A}_t | t = 1, ..., T\}$ be the observed adjacency matrices specified by parameters $(M^o, \boldsymbol{\lambda}^o, \boldsymbol{\psi}^o)$. And suppose the number of change points M^o is known. The change points and parameters are estimated by*

$$(\hat{\boldsymbol{\lambda}}_T, \hat{\boldsymbol{\psi}}_T) = arg \min_{\boldsymbol{\lambda} \in A^M_{\epsilon_\lambda}, \boldsymbol{\psi} \in \boldsymbol{\mathcal{M}}} \frac{1}{T} MDL(M^o, \boldsymbol{\lambda}, \boldsymbol{\psi})$$

Then $\hat{\boldsymbol{\lambda}}_T \xrightarrow{a.s.} \boldsymbol{\lambda}^o$ and for each interval, the estimated $\hat{\boldsymbol{c}}^{(m)}$ must be a bigger model than the true community assignment.

The idea of the proof can be found in Theorem 1 in Davis and Yau (2013). One can complete the proof by verifying the assumptions and propositions we mentioned.

Corollary 4.1. *Under the conditions of Theorem 4.2, if the number of change- points is unknown and is estimated from the data , then*

1. *The number of change points cannot be underestimated. That is to say, $\hat{M} \geq M^o$ almost surely when T is large enough.*
2. *When $\hat{M} > M^o$, $\boldsymbol{\lambda}^o$ must be a subset of the limit of $\hat{\boldsymbol{\lambda}}_T$ for large enough T.*
3. *In each fitted interval, the community assignment must be equal or be a bigger model that the corresponding true community assignment.*

Similarly, one just need to check the requirements for Corollary 1 in Davis and Yau (2013) to finish the proof.

Theorem 4.3. *Let $\boldsymbol{\lambda}^o = (\lambda_1^o, \lambda_2^o, ..., \lambda_{M^o}^o)$ be the true change points. And $(\hat{M}, \hat{\boldsymbol{\lambda}}_T, \hat{\boldsymbol{\psi}}_T)$ is the MDL-based result. Then $\forall m = 1, 2, ..., M^o$, there exists a $\hat{\lambda}_{t_m} \in \hat{\boldsymbol{\lambda}}_T$ where $1 \leq t_m \leq \hat{M}$ such that*

$$|\hat{\lambda}_{t_m} - \lambda_m^o| = o(T^{-\frac{1}{2}}) \text{ a.s.}$$

This theorem can be verified by checking the requirements in Theorem 2 in Davis and Yau (2013).

Lemma 4.3. *Suppose the true community assignment vector $\boldsymbol{c}^{o(m)}$ is specified for the mth interval, then*

$$\hat{\boldsymbol{\pi}}_T^{(m)}(\hat{\lambda}_{m-1}, \hat{\lambda}_m) - \boldsymbol{\pi}^{o(m)} = O(\sqrt{\frac{\log\log(T)}{T}}) \text{ a.s.}$$

When the specific community assignment $\boldsymbol{c}^{(m)}$ is bigger than $\boldsymbol{c}^{o(m)}$, which means there exist a function $g\colon c_i^{(m)} \to c_i^{o(m)}$, then we have

$$\hat{\boldsymbol{\pi}}_{T,ql}^{(m)}(\hat{\lambda}_{m-1}, \hat{\lambda}_m) - \boldsymbol{\pi}_{g(q)g(l)}^{o(m)} = O(\sqrt{\frac{\log\log(T)}{T}}) \text{ a.s.}$$

One can follow the proof of Lemma 2 in Davis and Yau (2013) for the proof of Lemma 4.3

Finally we come to the main result. By following the proof of Theorem 3 in Davis and Yau (2013), given Theorem 4.2, Theorem 4.3 and Lemma 4.3, the proof of Theorem 4.1 can be derived.

4.8.3 Discussion on consistency of other methods

The model setting of SCOUT (Hulovatyy and Milenković, 2016) is the same as the proposed method. By similar arguments, one can show that SCOUT based on BIC is consistent. The main idea is that the consistency depends on the order of the penalty term, instead of the particular format. However, it is interesting to note that the proposed method gives superior empirical performance, suggesting that in addition to the order of the penalty, the exact form of the selection criterion plays an important role in practical performance.

However, the model setting of Cheung *et al.* (2020) is different as the link probabilities can vary over time. Based on the results of Han *et al.* (2015), suppose the change points are known, the community assignment based on MLE is consistent. However, as the penalty term is of order $O(T)$, it grows too fast to ensure consistency. Therefore, the MDL based estimates in Cheung *et al.* (2020) can not be consistent.

Algorithm 4: Variational EM Algorithm

```
Input: $s_{max}$, $k_{max}$, $\hat{\boldsymbol{\tau}}^{(0)}$, $\hat{\boldsymbol{\alpha}}^{(0)}$, $\hat{\boldsymbol{\pi}}^{(0)}$;
Initialize: $k = 0$;
while $k < k_{max}$ and convergence criterion on $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ are not met do
    $\hat{\boldsymbol{\tau}}^{(0)} = \hat{\boldsymbol{\tau}}^{(k)}$, $s = 0$ ;
    while $s < s_{max}$ and convergence criterion on $\tau$ are not met do
        for $i$ = 1,2,..., $N$ do
            for $q$ = 1,2,...,$Q$ do
                $\hat{\tau}_{i,q}^{(s+1)} =$
                $\hat{\alpha}_q^{(k)} exp(\sum_{i\neq j}\sum_l\sum_t \hat{\tau}_{j,l}^{(s)}[A_{t,ij}\log(\hat{\pi}_{q,l}^{(k)}) + (1 - A_{t,ij})\log(1 - \hat{\pi}_{q,l}^{(k)})]);$
            end
        end
        $s = s + 1$;
    end
    for i=1,2,...,$N$ do
        $denom = \sum_{q=1}^{Q} \hat{\tau}_{i,q}^{(s)}$;
        for q=1,...,$Q$ do
            $\hat{\tau}_{i,q}^{(k+1)} = \hat{\tau}_{i,q}^{(s)}/denom$;
        end
    end
    for q=1,...,$Q$ do
        $\hat{\alpha}_q^{(k+1)} = \frac{1}{N}\sum_{i=1}^{N} \hat{\tau}_{i,q}^{(k+1)}$;
        for l=1,...,$Q$ do
            $\hat{\pi}_{ql}^{(k+1)} = \frac{\sum_{i\neq j}\sum_t \hat{\tau}_{q,i}^{(k+1)}\hat{\tau}_{j,l}^{(k+1)} A_{t,ij}}{\sum_{i\neq j}\sum_t \hat{\tau}_{q,i}^{(k+1)}\hat{\tau}_{j,l}^{(k+1)}}$
        end
    end
    $k = k + 1$;
end
Output: $\tilde{\tau} = \hat{\tau}^{(k)}$, $\tilde{\alpha} = \hat{\alpha}^{(k)}$, $\tilde{\pi} = \hat{\pi}^{(k)}$
```

Chapter 5

Concluding Remarks

In this thesis three different change point detection problems have been studied. They are concerned with

- time series of astronomical images,
- sequences of structured signals, and
- time-evolving dynamic networks.

A unified approached has been adopted to solve these problems. First, for each of these problems, a statistical change point model was put forward to model the data. Then the minimum description length principle was applied to derive a consistent model selection criterion for choosing the change points. Lastly, a practical algorithm was developed to solve the minimization problem induce by the model selection criterion.

We believe that substantial contributions have been made to the field of change point detection by this thesis.

References

Adams, R. and Bischof, L. (1994) Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 641–647.

Aggarwal, A., Schanche, N., Reeves, K. K., Kempton, D. and Angryk, R. (2018) Prediction of Solar Eruptions Using Filament Metadata. *The Astrophysical Journal Supplement Series*, **236**, 15.

Anderson, M. (2016) Hydroclimate report water year 2015. *Office of the State Climatologist.*

Aue, A. and Horváth, L. (2013) Structural breaks in time series. *Journal of Time Series Analysis*, **34**, 1–16. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9892.2012.00819.x.

Aue, A. and Lee, T. C. M. (2011) On image segmentation using information theoretic criteria. *The Annals of Statistics*, **39**, 2912–2935.

Aynaud, T. and Guillaume, J.-L. (2011) Multi-Step Community Detection and Hierarchical Time Segmentation in Evolving Networks. In *Fifth SNA-KDD Workshop Social Network Mining and Analysis, in conjunction with the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011).* San Diego, CA, United States. URL https://hal.archives-ouvertes.fr/hal-01286941.

Barnes, G., Schanche, N., Leka, K. D., Aggarwal, A. and Reeves, K. (2017) A Comparison of Classifiers for Solar Energetic Events. In *Astroinformatics* (eds. M. Brescia, S. G. Djorgovski, E. D. Feigelson, G. Longo and S. Cavuoti), vol. 325, 201–204.

Beck, A. and Teboulle, M. (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, **2**, 183–202.

Bickel, P., Choi, D., Chang, X. and Zhang, H. (2013) Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, **41**, 1922 – 1943. URL https://doi.org/10.1214/13-AOS1124.

Bickel, P. J. and Chen, A. (2009) A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, **106**, 21068–21073. URL https://www.pnas.org/content/106/50/21068.

Bleakley, K. and Vert, J.-P. (2011) The group fused lasso for multiple change-point detection. arXiv:1106.4199 [q–bio.QM].

Carson, C., Belongie, S., Greenspan, H. and Malik, J. (1999) Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 167–181.

Chen, X., Kim, S., Lin, Q., Carbonell, J. G. and Xing, E. P. (2010) Graph-structured multi-task regression and an efficient optimization method for general fused lasso. arXiv:1005.3579 [stat.ML].

Chen, X., Lin, Q., Kim, S., Carbonell, J. G. and Xing, E. P. (2012) Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, **6**, 719–752.

Cheung, R. C. Y., Aue, A., Hwang, S. and Lee, T. C. M. (2020) Simultaneous detection of multiple change points and community structures in time series of networks. *IEEE Transactions on Signal and Information Processing over Networks*, **6**, 580–591.

Choi, D. S., Wolfe, P. J. and Airoldi, E. M. (2012) Stochastic blockmodels with a growing number of classes. *Biometrika*, **99**, 273–284. URL http://www.jstor.org/stable/41720691.

Daly, C., Neilson, R. P. and Phillips, D. L. (1994) A statistical-topographic model for mapping climatological precipitation over mountainous terrain. *Journal of Applied Meteorology*, **33**, 140–158.

Daudin, J. J., Picard, F. and Robin, S. (2008) A mixture model for random graphs. *Statistics and Computing*, **18**, 173–183. URL https://doi.org/10.1007/s11222-007-9046-7.

Davis, R. A., Lee, T. C. M. and Rodriguez-Yam, G. A. (2006) Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, **101**, 223–239.

Davis, R. A. and Yau, C. Y. (2013) Consistency of minimum description length model selection for piecewise stationary time series models. *Electronic Journal of Statistics*, **7**, 381–411.

Dey, V., Zhang, Y. and Zhong, M. (2010) A review on image segmentation techniques with remote sensing perspective. *The International Society for Photogrammetry and Remote Sensing Symposium (ISPRS) TC VII Symposium-100 Years ISPRS, Vienna, Austria*, **XXXVIII**, 31–42.

Eagle, N., Pentland, A. S. and Lazer, D. (2009) Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, **106**, 15274–15278. URL `https://www.pnas.org/content/106/36/15274`.

Fan, M. and Lee, T. C. (2015) Variants of seeded region growing. *IET Image Processing*, **9**, 478–485.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004) Efficient graph-based image segmentation. *International Journal of Computer Vision*, **59(2)**, 167–181.

Gibberd, A. J. and Nelson, J. D. B. (2017) Regularized estimation of piecewise constant Gaussian graphical models: The group-fused graphical lasso. *Journal of Computational and Graphical Statistics*, **26**, 623–634.

Goldenberg, A., Zheng, A. X., Fienberg, S. E. and Airoldi, E. M. (2010) A survey of statistical network models. *Found. Trends Mach. Learn.*, **2**, 129–233. URL `https://doi.org/10.1561/2200000005`.

Güdel, M. (2004) X-ray astronomy of stellar coronae. *The Astronomy and Astrophysics Review*, **12**, 71–237.

Güdel, M., Audard, M., Skinner, S. L. and Horvath, M. I. (2002) X-Ray Evidence for Flare Density Variations and Continual Chromospheric Evaporation in Proxima Centauri. *The Astrophysical Journal*, **580**, L73–L76.

Hallac, D., Park, Y., Boyd, S. and Leskovec, J. (2017) Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 205–213.

Han, Q., Xu, K. S. and Airoldi, E. M. (2015) Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 1511–1520. JMLR.org.

Holland, P. W., Laskey, K. B. and Leinhardt, S. (1983) Stochastic blockmodels: First steps. *Social Networks*, **5**, 109–137. URL https://www.sciencedirect.com/science/article/pii/0378873383900217.

Hughes, J. M., Hsu, V. W., Seaton, D. B., Bain, H. M., Darnel, J. M. and Krista, L. (2019) Real-time solar image classification: Assessing spectral, pixel-based approaches. *Journal of Space Weather and Space Climate*, **9**, A38.

Hulovatyy, Y. and Milenković, T. (2016) Scout: simultaneous time segmentation and community detection in dynamic networks. *Scientific Reports*, **6**.

Hurlburt, N., Cheung, M., Schrijver, C., Chang, L., Freeland, S., Green, S., Heck, C., Jaffey, A., Kobashi, A., Schiff, D., Serafin, J., Seguin, R., Slater, G., Somani, A. and Timmons, R. (2012) Heliophysics Event Knowledgebase for the Solar Dynamics Observatory (SDO) and Beyond. *Solar Physics*, **275**, 67–78.

Kim, S., Sohn, K.-A. and Xing, E. P. (2009) A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, **25**, i204–i212.

Kolar, M. and Xing, E. P. (2012) Estimating networks with jumps. *Electronic Journal of Statistics*, **6**, 2069–2106.

Lee, T. C. M. (2000) A minimum description length-based image segmentation procedure, and its comparison with a cross-validation-based segmentation procedure. *Journal of the American Statistical Association*, **95**, 259–270.

Lee, T. C. M. (2001) An introduction to coding theory and the two–part minimum description length principle. *International Statistical Review*, **69**, 169–183.

Lemen, J. R., Title, A. M., Akin, D. J., Boerner, P. F., Chou, C., Drake, J. F., Duncan, D. W., Edwards, C. G., Friedlaender, F. M., Heyman, G. F., Hurlburt, N. E., Katz, N. L., Kushner, G. D., Levay, M., Lindgren, R. W., Mathur, D. P., McFeaters, E. L., Mitchell, S., Rehse, R. A., Schrijver, C. J., Springer, L. A., Stern, R. A., Tarbell, T. D., Wuelser, J.-P., Wolfson, C. J., Yanari, C., Bookbinder, J. A., Cheimets, P. N., Caldwell, D., Deluca, E. E., Gates, R., Golub, L., Park, S., Podgorski, W. A., Bush, R. I., Scherrer, P. H., Gummin, M. A., Smith, P., Auker, G., Jerram, P., Pool, P., Soufli, R., Windt, D. L., Beardsley, S., Clapp, M., Lang, J. and Waltham, N. (2012) The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO). *Solar Physics*, **275**, 17–40.

Martens, P. C. H., Attrill, G. D. R., Davey, A. R., Engell, A., Farid, S., Grigis, P. C., Kasper, J., Korreck, K., Saar, S. H., Savcheva, A., Su, Y., Testa, P., Wills-Davey, M., Bernasconi, P. N., Raouafi, N. E., Delouille, V. A., Hochedez, J. F., Cirtain, J. W., Deforest, C. E., Angryk, R. A., de Moortel, I., Wiegelmann, T., Georgoulis, M. K., McAteer, R. T. J. and Timmons, R. P. (2012) Computer Vision for the Solar Dynamics Observatory (SDO). *Solar Physics*, **275**, 79–113.

Nesterov, Y. (2005) Smooth minimization of non-smooth functions. *Mathematical Programming*, **103**, 127–152.

Paul, S. and Chen, Y. (2016) Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electron. J. Statist.*, **10**, 3807–3870. URL `https://doi.org/10.1214/16-EJS1211`.

Peel, L. and Clauset, A. (2015) Detecting change points in the large-scale structure of evolving networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, 2914–2920. AAAI Press.

Pesnell, W. D., Thompson, B. J. and Chamberlin, P. C. (2012) The Solar Dynamics Observatory (SDO). *Solar Physics*, **275**, 3–15.

Ranson, M. (2014) Crime, weather, and climate change. *Journal of Environmental Economics and Management*, **67**, 274–302.

Rissanen, J. (1989a) *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore.

Rissanen, J. (1989b) *Stochastic Complexity in Statistical Inquiry Theory*. USA: World Scientific Publishing Co., Inc.

Rissanen, J. (2007a) *Information and Complexity in Statistical Modeling*.

Rissanen, J. (2007b) *Information and Complexity in Statistical Modeling*. Springer Science & Business Media.

Rousseeuw, P. J. and Croux, C. (1993) Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, **88**, 1273–1283. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1993.10476408.

Scargle, J. D., Norris, J. P., Jackson, B. and Chiang, J. (2013) STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. VI. BAYESIAN BLOCK REPRESENTATIONS. *The Astrophysical Journal*, **764**, 167. URL https://doi.org/10.1088%2F0004-637x%2F764%2F2%2F167.

Sharpnack, J., Singh, A. and Rinaldo, A. (2013) Changepoint detection over graphs with the spectral scan statistic. In *International Conference on Artificial Intelligence and Statistics*, vol. 16, 545–553.

Stanley, N., Shai, S., Taylor, D. and Mucha, P. J. (2016) Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, **3**, 95–105.

Taddy, M. A. (2010) Autoregressive mixture models for dynamic spatial poisson processes: Application to tracking intensity of violent crime. *Journal of the American Statistical Association*, **105**, 1403–1417.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B*, **67**, 91–108.

Truong, C., Oudre, L. and Vayatis, N. (2020) Selective review of offline change point detection methods. *Signal Processing*, **167**, 107299. URL `https://www.sciencedirect.com/science/article/pii/S0165168419303494`.

Wang, Y.-X., Sharpnack, J., Smola, A. J. and Tibshirani, R. J. (2016) Trend filtering on graphs. *Journal of Machine Learning Research*, **17**, 1–41.

Wong, R. K. W., Kashyap, V. L., Lee, T. C. M. and van Dyk, D. A. (2016) Detecting abrupt changes in the spectra of high-energy astrophysical sources. *Annals of Applied Statistics*, **10**, 1107–1134. URL `https://doi.org/10.1214/16-AOAS933`.

Yang, J. and Peng, J. (2020) Estimating time-varying graphical models. *Journal of Computational and Graphical Statistics*, **29**, 191–202.

Zhou, Y., Goldberg, M., Magdon-ismail, M. and Wallace, W. A. (2007) Strategies for cleaning organizational emails with an application to enron email dataset. In *5th Conf. of North American Association for Computational Social and Organizational Science (NAACSOS 07)*. Emory – Atlanta, GA.