2023

Extending the Bayesian Dynamic Linear Model to changepoint problems

https://hdl.handle.net/2144/47986

Boston University
EXTENDING THE BAYESIAN DYNAMIC LINEAR MODEL TO CHANGEPPOINT PROBLEMS

by

RYAN FROST

B.S., Arizona State University, 2016

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2023
Approved by

First Reader
Luis Carvalho, PhD
Associate Professor of Mathematics and Statistics

Second Reader
Alessandro Baccini, PhD
Research Professor of Earth and Environment

Third Reader
Mark Kon, PhD
Professor of Mathematics and Statistics

Fourth Reader
Ashis Gangopadhyay, PhD
Associate Professor of Mathematics and Statistics
Acknowledgments

To my advisor Luis, thank you for your encouragement, advice, and inspiration, and thank you for pushing and challenging me every step of the way. It was truly a privilege to work with you.

To Erin, thank you for your never-ending patience and for always believing in me. I would not have gotten here without you.

To my friends and family, thank you for your love and unfailing support. You bring joy to life!
EXTENDING THE BAYESIAN DYNAMIC LINEAR MODEL TO CHANGEPOINT PROBLEMS

RYAN FROST

Boston University, Graduate School of Arts and Sciences, 2023
Major Professor: Luis Carvalho, PhD
Associate Professor of Mathematics and Statistics

ABSTRACT

In this dissertation we develop a set of related methods for modeling a subset of time series data. Specifically, these models are designed for time series that contain a semi-regular pattern, but subject to outliers in the observations, as well as gradual and abrupt changes in pattern. We motivate these models and algorithms by an application to remote sensing reflectance data, which forces strong attention towards computational complexity due to the overwhelming abundance of data.

The models we introduce are based on the Bayesian Dynamic Linear Model (DLM), and we present them as a progression, in order of increasing model complexity and computational cost.

We start with a standard DLM implementation, with a simple outlier removal procedure. The focus of this work is in the application - we apply this model across tens of millions of time series, using the BU Shared Computing Cluster. We show that it performs favorably in comparison to existing methods in reflectance modeling.

Next, we use variance scales to more robustly accommodate both outliers and abrupt changes in pattern. Further, we develop a characterization of the tradeoff between downweighting outliers and rapidly adapting to changes in pattern, when
using this model.

Finally, we propose a formal changepoint extension for the DLM. We combine the DLM with a Product Partition Model, allowing for rigorous inference on location of changepoints within a series, while preserving the DLM’s capacity for gradual adaptation between changepoints.

We also provide implementations of each of these algorithms in an R package, with the core algorithms written in C++ for efficient computation.
Contents

1 Introduction 1

2 Modeling reflectance with the Dynamic Linear Model 6
   2.1 Landsat Data Challenges ........................................ 10
      2.1.1 Cloud Problems ........................................... 10
      2.1.2 Change ................................................... 10
      2.1.3 Quantity .................................................. 11
   2.2 Landsat processing steps ....................................... 11
      2.2.1 Landsat 5 & Landsat 7 Alignment .......................... 11
   2.3 Previous Work ................................................. 12
   2.4 Dynamic Linear Model ........................................ 15
      2.4.1 Modeling the Observation Variance ....................... 18
      2.4.2 Specifying the Evolution Variance ....................... 18
      2.4.3 Outlier Detection & Removal ............................... 18
   2.5 Applied Discounting Approach ................................ 21
      2.5.1 Further notes on discounting .............................. 24
   2.6 Fitting the Dynamic Linear Model ............................ 25
   2.7 Computational Considerations ................................ 26

3 Robustifying the DLM to Outliers and Soft Changepoints 29
   3.1 Outlier Robustness ............................................. 29
      3.1.1 Outlier Robustness Previous Work ....................... 30
List of Tables

4.1 Model Fit Performance .................................................. 70
4.2 Changepoint Detection Performance ................................. 72
# List of Figures

1·1 The timeseries observations of Shortwave Infrared reflectance values for a representative pixel. We overlay the fit provided by the Continuous Change Detection and Classification (CCDC) model (Zhu & Woodcock, 2014), a method that is widely used in the field of remote sensing. ......................................................... 3

2·1 The area of interest: Landsat Path 227, Row 62. This region covers an area of Pará, Brazil. In the top left, the muddy waters of the Amazon River intersect the darker blue Tapajós River. The red outline shows the area of focus for Figure 2·2. ......................................................... 8

2·2 The top two panels show two captures from the red box outlined in Figure 2·1, separated by about 11 years. The bottom panel gives the timeseries of Shortwave Infrared reflectance observations for the pixel in the center of the region, indicated by the red outline. .......... 9

2·3 The CCDC fit for the representative pixel. CCDC fits a multivariate model using all 7 Landsat bands. Here we show only the predictions for the SWIR band (band 5). ......................................................... 14

2·4 State space diagram for a standard Dynamic Linear Model with constant, unknown observational variance ........................................ 15
2-5  A Dynamic Linear Model fit to the representative pixel. Shaded region represents the 90% symmetric credible interval. The color of each point indicates whether that point was flagged and removed by our outlier detection procedure. .................................................. 20

2-6  DLM fits of the same representative pixel with varying levels of discounting. The rows of the grid are nested with seasonal discounting inside of mean/trend discounting: the first three rows show a low mean/trend discount factor, with progressively larger seasonal discounting, the next three have medium mean/trend discounting, and the last three have a large degree of mean/trend discounting. ........ 23

3-1  State space diagram for the Robust Dynamic Linear Model .............. 31

3-2  A robust Dynamic Linear Model fit to the representative pixel, with \( \tau \) set to 15 using the Expectation Maximization-like method. The solid line represents the EM-like MAP for the mean of the signal. The color of each point shows the weight for that observation (\( q_t \)). ............. 32

3-3  The state space diagram for the Robust Soft-Changepoint Dynamic Linear Model ........................................................................................................... 40

3-4  A Robust Soft-Changepoint DLM fit to the representative pixel, with \( \tau \) set to 4, and \( \gamma \) set to 5. Shaded region again represents the 90% symmetric credible interval. The color of each point shows the weight for that observation (\( q_t \)). .................................................. 41
3·5 A Robust Soft-Changepoint model fit to a simple box function with small noise. The top left shows the series, with four locations of interest marked by color. Bottom left shows the same series, with sample paths of the posterior for $\theta$ drawn overlaid. On the right, the three plots show the density of samples for $\theta$, $\log(\alpha)$, and $\log(q)$ at each of the locations of interest.

3·6 A simplified look into the RSC’s choice when encountering a box function. Either the model adapts to the middle points (red), or it down-weights them (blue), depending on the balance of $\tau$ and $\gamma$.

3·7 A visualization of the $\tau$-$\gamma$ tradeoff in our simplified analysis of the box function. The facetting represents different numbers $n$ of points and different heights $M$ of the middle section of the box function. Color signifies how much $\theta$ adapts within the middle section - dark red shows that $\theta_m \approx M$, dark blue shows that $\theta_m \approx 0$.

4·1 The state space diagram for the PPM-coupled Dynamic Linear Model.

4·2 A PPM-coupled DLM fit to the representative pixel. The bottom figure shows the probability of a changepoint at each location. Solid bars show the probability at that point, while the dotted line gives the 360-day rolling sum - the expected number of changepoints within a $\sim$1 year window.

4·3 A selection of representative time series of Shortwave Infrared Reflectance, shown with the fits of the CCDC, Direct DLM-PPM, and Robust DLM-PPM models. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color (or the hand-annotated ground truth).
4.4 A selection of pixel timeseries where the DLM-PPM models underperform. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color.

4.5 A sample of pixel timeseries where the DLM-PPM models outperform CCDC at detecting change. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color.
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCDC</td>
<td>Continuous Change Detection and Classification</td>
</tr>
<tr>
<td>DLM</td>
<td>Dynamic Linear Model</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation Maximization</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimation</td>
</tr>
<tr>
<td>PPM</td>
<td>Product Partition Model</td>
</tr>
<tr>
<td>RSC</td>
<td>Robust Soft-Changepoint</td>
</tr>
<tr>
<td>SWIR</td>
<td>Shortwave Infrared</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Dynamic Linear Models (DLMs) provide a highly useful Bayesian framework for modeling time series that exhibit a stable pattern subject to gradual changes over time. As a motivating example for this present work, we consider the challenge of modeling surface reflectance across the Earth using massive Landsat satellite imagery data.

Remote-sensed imagery—imagery obtained from satellites orbiting the Earth—provides a wealth of data, useful for many scientific purposes. However, modeling this data poses significant challenges due to four primary concerns: frequent outliers, missing data, changes in pattern (both gradual and abrupt), and sheer quantity of data. Clouds and cloud shadows can cause outliers and missing data, and pattern changes can occur due to natural changes in land cover, though more commonly are a result of man-made disturbances, such as deforestation. The scale of the data is a blessing but also a curse, as with all “big data”. The high resolution images (one pixel represents a 30m by 30m square) captured by the satellites provide an incredible amount of information about the Earth’s surface, but also require significant computation to analyze with statistical models. Figure 1.1 provides the timeseries of Shortwave Infrared reflectance observations for one representative pixel. Note outlying values, gradual changes over time, as well as at least one abrupt change occurring within the series.
The full work of this thesis is motivated from this set of modeling challenges. Addressing these challenges together incurs a trade-off; the first two challenges, outliers and pattern changes, generally benefit from increased model complexity. However, added complexity brings with it a heavier computational burden, making it impractical at scale. As such, we often can’t use the best, most informative model to fit every time series so we must make a sacrifice. In this work we propose a new, comprehensive set of methods for modeling Landsat reflectance data at varying levels of model complexity. With this set of tools, we enable the practitioner to model on a sliding scale: the simplest, fastest methods may be used at the largest scales, and progressively more representative models may be employed for smaller subsets of series that warrant closer attention.

To initially tackle the first two issues, we implement a standard DLM with cross-validated outlier removal in Chapter 2. This approach is computationally fast—a necessity when fitting the proposed model on millions of individual pixels. In this chapter we apply this methodology to study biomass distribution in a region of the Amazon river. We show that the model offers good statistical and computational performance when compared to the state of the art in the field. Importantly, our model provides uncertainty estimates around its predictions, a feature that has been absent from existing methods.

The main methodological challenge here is ensuring that the process appropriately adapts to gradual changes in the underlying signal, without overadapting to outliers. Ideally, the model is only as adaptive as is necessary to capture true changes in signal. This behavior is controlled by tuning the discount factor, a parameter that determines the smoothness of the model. We use cross validation to optimize the discount factor for performance of the model over a large set of timeseries.

Next, in Chapter 3, we introduce a model-based, robust treatment of outliers
Figure 1.1: The timeseries observations of Shortwave Infrared reflectance values for a representative pixel. We overlay the fit provided by the Continuous Change Detection and Classification (CCDC) model (Zhu & Woodcock, 2014), a method that is widely used in the field of remote sensing.
via latent observational variance scales. Further, we add latent scales to the evolution variances, allowing the model to adapt to change points without explicitly detecting them. Together, these model additions form what we call the Robust Soft-Changepoint (RSC) DLM.

While resulting in a flexible and versatile model, the combination of latent variance scales on both the observations and evolutions forces a new trade-off because a contiguous series of seemingly out-of-model observations can be regarded as either outliers or a new segment due to an abrupt change. This trade-off is effectively calibrated by balancing the hyper-prior parameters on these variance scales, observational and evolutional, predisposing the model towards either adapting to (stretches of) outlying observations or downweighting them. We discuss guidelines for eliciting priors that achieve such balance.

To fit the RSC DLM, we derive an Expectation Maximization-like procedure, which may be used when prioritizing a low computational burden. We also provide a Markov chain Monte Carlo Gibbs sampling procedure for fitting the model, which offers enhanced visibility on the outlier-adaptation trade-off. These model extensions give the DLM the flexibility necessary to capture the timeseries patterns common in reflectance data and as a result offer a superior fit than existing methods.

For cases where extra computation is viable, we propose a third and final model in Chapter 4. This extension combines the flexibility of the Dynamic Linear Model with the inferential power of a formal changepoint model. Formulating the model in this way allows us to specify prior knowledge about the location and frequency of changepoints, and furthermore yields interpretable posterior distributions over these changepoint configurations. This allows for rigorous inference on the locations of changepoints.

To preserve the Bayesian cohesiveness of the full model, we choose to layer the
DLM on top of a product partition model (PPM) (Barry & Hartigan, 1993). The PPM is a Bayesian approach where the segmentation of the series is added to the model as a random parameter to be estimated. Barry and Hartigan (1993) introduced this model in the context of independent segments, where within each segment, observations are generated from a common normal distribution. Instead of constant means within segments, we instead construct our model so that each segment (quasi- or fully-) independently follows a dynamic linear model. This approach yields numerous benefits, gracefully combining the strength of the PPM as a changepoint model with the within-segment flexibility of the DLM.

We implement this combination in two separate ways. The first is a direct approach, where the analysis provided by Barry and Hartigan (1993) is followed more closely, but with DLMs instead of normal means within segments. This is typically the faster of the two approaches. The second approach keeps the variance scales of the RSC DLM, using the segmentation of the PPM to determine the hyperprior on the evolution variance scale; segment boundaries predispose the scale to be large, resulting in near-independence across segments. This method requires more significant computation, but offers outlier robustness in exchange.

In summary, these extensions to the DLM provide a set of tools that, in order of increased computational cost, may: robustly handle outliers, accommodate abrupt changes in pattern, and formally detect these changes. We provide the dlmr R package that efficiently implements each of these methods using C++.
Chapter 2
Modeling reflectance with the Dynamic Linear Model

Remote sensing is concerned with studying processes on the surface of the Earth using satellite image data. For example, we often wish to track human generated disturbances such as fires and deforestation leading to significant biomass stock changes affecting global weather. Each image captured by the remote-sensing satellite is spatially calibrated, so that each pixel of the image may be precisely mapped to a particular small region of the Earth’s surface. As the satellite orbits the Earth, it revisits locations on a consistent schedule, providing new observations of the same region. Remote sensed data has different spatial and temporal resolutions depending on the satellite and capturing devices. For instance, Landsat data has a 30m by 30m spatial resolution at each pixel, with data being collected every 16 days per satellite. Landsat 5 and Landsat 7 were offset by 8 days, so that during the overlap of their operational periods, there was an 8 day gap between observations of a single region.

Land surface reflectance measures the intensity of light reflected by some region of the Earth’s surface, relative to the amount of light shone on that region. Landsat satellites record the reflectance across the visible and infrared wavelength spectrum, decomposed into several wavelength bands. Effectively, the reflectances across these bands together provide a representation of the overall ‘color’ of the land. In our analysis, we fit the reflectance timeseries for the different wavelength bands separately,
ignoring for simplicity the correlations between spectra.

Analyses of satellite imagery timeseries data face three primary challenges. First, image captures are frequently obscured by clouds, cloud shadows, and snow, causing outliers and missing data. In addition, the actual captured pattern can change over time, for instance, due to phenological changes – trees can senesce and recover differently from year to year – and man-made processes. These changes can either be gradual (e.g. forest degradation or recovery) or abrupt (e.g. forest fires). The third challenge is computational, since there is a large volume of remote sensed data even for moderately sized regions. Our goal is to methodologically address these issues with a dynamic model that is robust to both outliers and abrupt changes.

As a concrete example, consider Figure 2·1 showing two captures of a representative Landsat Tile. The two images were taken about 10 years apart. Of the several hundred images taken between 1984 and 2018, these were two of the cleanest, with the least amount of obscuring cloud cover. Even still, clouds do cover a few parts of the image, providing no usable information about the land reflectance below them.

Figure 2·2 offers a closer look at the red outlined box from Figure 2·1. Comparing the images, we can visually see the impact of deforestation.

The bottom of Figure 2·2 shows the timeseries of Shortwave Infrared (SWIR) reflectance observations for the pixel at the center of the top two panels (the pixel outlined in red). The deforestation occurring within this pixel manifests as a clear changepoint in the timeseries around the year 2010, and possibly a smaller change near the year 2000. Before the deforestation, the pixel contains thick forest, with low reflectance in the SWIR band and low seasonal variation. After the change, the area appears to contain low density vegetation. This results in a higher base level of SWIR reflectance, as well as more seasonal variation.
Figure 2-1: The area of interest: Landsat Path 227, Row 62. This region covers an area of Pará, Brazil. In the top left, the muddy waters of the Amazon River intersect the darker blue Tapajós River. The red outline shows the area of focus for Figure 2-2.
Figure 2-2: The top two panels show two captures from the red box outlined in Figure 2-1, separated by about 11 years. The bottom panel gives the timeseries of Shortwave Infrared reflectance observations for the pixel in the center of the region, indicated by the red outline.
2.1 Landsat Data Challenges

2.1.1 Cloud Problems

Landsat satellites orbit the earth in a 16-day cycle, meaning they pass over any individual region of the earth once every 16 days. This provides, at maximum, 23 captures of the region’s surface per year. However, when there are clouds in the sky on the day of the capture, the observation is not useful. The FMASK image processing algorithm (Zhu & Woodcock, 2012; Zhu et al., 2015) is used to detect clouds from the satellite image captures, and flag the affected pixels as invalid. When this is successful, the result is missing data in the time series. Worse is when the algorithm fails to detect a cloud, and the white reflectance value makes it into the series, causing an outlier. These outliers provide little useful information about the surface reflectance, so the goal is to detect them and remove them from the timeseries.

2.1.2 Change

The reflectance time series of an individual pixel can reveal much about the nature of the land. There are three general types of change that appear in reflectance time series. The first type is predictable, pattern-based change such as a trend of steady increase or decrease over time, or consistent seasonal variation. The second type is gradual deviations in this stable pattern, such as small changes to the shape of the seasonal signal, or slow changes in the trend itself. This sort of change is observed in pixels where the land begins to change gradually. One way this behavior can occur is from forest degradation, where a forest is not completely removed, but a few trees are harvested over a period of time. This lowers the density of the tree canopy, gradually shifting the reflectance profile. The final type of change is rapid, extreme change where the existing pattern is entirely discarded. This can happen in forest fires and
complete deforestation, where between two observations just days apart, the land cover in an affected pixel is completely changed.

2.1.3 Quantity

Each pixel of a Landsat image covers a 30x30 meter square of the Earth’s surface. Since the Earth’s surface is about 510 million km², there are approximately 660 billion Landsat pixels covering the Earth, each with its own timeseries of observations. Each timeseries spans about 35 years of data (and counting). At the daily level that comes to over 12000 time points. Further, Landsat satellites record the reflectance in several separate wavelength bands. 5 of these bands are of interest in biomass prediction. With these numbers in mind, a modeling solution to this data will need to be fast to be feasible even for retrospective analyses, and require efficient model updates for online surveillance of anomalies.

2.2 Landsat processing steps

2.2.1 Landsat 5 & Landsat 7 Alignment

The time series observations come from two separate satellites, Landsat 5 and Landsat 7. Landsat 5 launched on March 1, 1984 and ceased recording on June 5, 2013. Landsat 7 launched on April 15, 1999, and continues to produce observations as of 2021. The two satellites have a 16-day orbit period with an 8 day offset, meaning any individual pixel is observed once every 8 days.

The two satellites both record surface reflectance in segmented wavelength bands. These bands match up across the two satellites closely, but not perfectly. Due to differences in sensor hardware, observations of the same pixel for Landsat 7 have a noticeable upward level shift compared to Landsat 5.

Before fitting the time series, we align the observations. The broad idea is to
produce a model that, given a Landsat 7 reflectance value, can predict the corresponding Landsat 5 value. To build this model we require Landsat 5 and Landsat 7 observations of the same pixel, close together in time. We sample pixels from across the scene, with a large variety of land cover types. We guarantee this by stratifying the sample across quantiles of biomass.

Then, for each pixel we select all pairs of Landsat 5 and Landsat 7 observations that occur within 8 days of each other. Effectively, we choose each Landsat 7 observation, and match it with the Landsat 5 observations from 8 days before and 8 days after. With this data, we fit a robust linear regression.

2.3 Previous Work

Since the Landsat data archive was made freely available to the public in 2008, numerous approaches to modeling reflectance timeseries have emerged. Many of these approaches were designed with land cover classification in mind, and as a result often favor year-to-year stability of estimates at the expense of over-smoothing. Most traditional methods use a combination of mean, trend, and seasonal components to capture the pattern of the reflectance time series. Some, such as CCDC, also give provision for changepoints, fitting separate mean, trend, seasonal models to each segment of the series.

Landtrendr (Kennedy et al., 2010) temporally segments the time series, fitting a piecewise linear model, but including only one observation per year to dispense with seasonal effects. Zhu and Woodcock (2014) introduced Continuous Change Detection and Classification (CCDC), wherein they segment the time series and fit a robust linear model with mean, trend, and seasonal components to each segment:
\[ Y_t = a + bt + \sum_{h=1}^{H} c_h \sin \left( \frac{2\pi ht}{T} \right) + d_h \cos \left( \frac{2\pi ht}{T} \right) + e_t \] (2.1)

where \( H \) is the number of harmonics and \( T \) is the period, the number of time points spanned by a single oscillation of the first harmonic.

CCDC is perfectly suitable for some downstream processes, such as landcover classification. The CCDC method achieved an overall classification accuracy of 90.20%, suggesting that classification performance is less sensitive to the dynamic, year-to-year fluctuations of reflectance. However, the underlying signal of the reflectance can change in ways that are not well-represented by this piecewise linear/seasonal approach. Figure 2·3 shows the CCDC fit for our representative pixel. The fit recognizes a change in pattern near late 2009 which appears by eye-test to be accurate. However, the fits within the two segments leave much to be desired. In the first segment, the model seems to capture the pattern in the beginning, but is unable to adjust as the series begins to gradually drift upwards, and instead compromises by adding an overall trend to the full length of the segment. The allowance for changes in the signal over time is overly rigid, as the only choice the model has is to either continue the current pattern, or start a new one. Two problems can directly arise from this rigidity. First, small fluctuations in reflectance can be classified as changes in land cover erroneously (commission errors). Second, in order to avoid these commission errors, the threshold for a change can be set higher, though this leads to missing true changes in land cover (omission errors). The threshold can be tuned to balance these two errors, but the rigidity of the all-or-nothing approach to change forces this tradeoff. A solution, then, is a model that matches the structure of CCDC’s mean-trend-seasonal linear model, but where the model coefficients may gradually change over time. Giving the model the freedom to follow the evolving pattern dynamically
allows it to capture the signal with greater fidelity.

Ye et al. (2021) modeled reflectance data with a Kalman filter approach. This approach brings many of the benefits of state space modeling. However, the statistical focus of Dynamic Linear Models offers considerable advantages over the control-theory, optimization based Kalman filter. Most importantly in this context, Dynamic Linear Models provide measures of uncertainty around every part of the model, allowing for rigorous statistical inference. This makes it possible to make probabilistic claims for hypotheses about the individual pixel series. In addition, we may carry the uncertainties from the reflectance model through when performing downstream analysis. For example, if we use the dynamic mean of the reflectance to predict biomass,
we also have a rigorous measure of the uncertainty associated with the dynamic mean. We can propagate this uncertainty through, combining it with the uncertainty inherent in the biomass model, to form an overall measure of uncertainty in the estimate for biomass.

2.4 Dynamic Linear Model

With these motivating constraints, we propose a Dynamic Linear Model (West & Harrison, 1997). Dynamic Linear Models (DLMs) are state space models with linearity in both the observation emission process and the state evolution process.

We start with the constant variance dynamic linear model

\[ \theta_t = G \theta_{t-1} + \omega_t, \quad \omega_t \sim N(0, W_t) \]  
\[ Y_t = F^t \theta_t + \nu_t, \quad \nu_t \sim N(0, V) \]  

where \( \theta_t \) is the state at time \( t \), in our case consisting of mean, trend, and seasonal components that capture the underlying pattern of reflectance at time \( t \). \( G \) is a matrix
that determines the pattern of predictable evolution exhibited by the state, and $F$ determines how the different components of the state contribute to the observed data point at each time $t$. A cleverly chosen $G$ and $F$ allow the state to dynamically capture mean, trend, and seasonal effects from the time series. In effect, this model can be made equivalent to CCDC (Equation (2.1)) with time-varying regularized coefficients, i.e.

$$Y_t = a_t + b_t t + \sum_{h=1}^{H} c_{ht} \sin \left( \frac{2\pi h t}{T} \right) + d_{ht} \cos \left( \frac{2\pi h t}{T} \right) + \epsilon_t$$  \hspace{1cm} (2.4)

The $F$ and $G$ that make this equivalence are as follows (with $H = 2$):

$$F = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$ \hspace{1cm} (2.5)

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos f & \sin f & 0 & 0 \\ 0 & 0 & -\sin f & \cos f & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos 2f & \sin 2f \\ 0 & 0 & 0 & 0 & -\sin 2f & \cos 2f \end{pmatrix}$$ \hspace{1cm} (2.6)

with $f = \frac{2\pi}{365.25 \times 8}$, the frequency of the first harmonic (assuming 8 days per time step).

and $\theta_t$ is a 6-element vector:
The traditional model is based on two sources of normally distributed error. The first, \( \omega_t \), is in the evolution step. Here, the error specifies the amount of deviation in the state between time steps, beyond the consistent pattern enforced by the evolution matrix \( G \). The second source, \( \nu_t \), is in the observation step, acting as a measurement noise between the value that is observed and the true signal generated by the state.

The Dynamic Linear Model is a natural match for capturing the patterns found in reflectance data. The mean, trend, and seasonal components are allowed to adapt smoothly over time, rather than only abruptly at changepoints. This gives the model far more flexibility to capture meaningful behavior of the series that would be missed by more rigid methods. Importantly, the added flexibility of the DLM does not come at the cost of interpretability. As mentioned, the mean, trend, seasonal DLM is equivalent to the base CCDC model where each of the coefficients can change over time, and in fact is exactly equivalent if we force the DLM coefficients to be static. Thus, the model provides estimates that describe the instantaneous pattern at every point in time. From these estimates, we may derive metrics such as the instantaneous harmonic amplitudes and phases, which can be informative towards the land cover of the pixel.

In addition, the DLM provides statistically rigorous measures of uncertainty for its estimates. Such measures of uncertainty have largely been missing from the reflectance modeling literature. This allows us to make far more informed inferences about the reflectance pattern. For example, the model deals with missing data in a
perfection in a perfectly natural way. When encountering stretches of missing data, the pattern is modeled to remain unchanged through this stretch, since there is no evidence to the contrary. However, the uncertainty about the state of the pattern grows larger the further away we are from the last observed data point.

2.4.1 Modeling the Observation Variance

For our modeling purposes, we will assume that the observation variance ($V$) is constant over time, but unknown. West and Harrison (1997) offer a comprehensive treatment of this model, and we follow their approach of modeling $V$ as Inverse Gamma distributed, or equivalently the precision $\phi = \frac{1}{V}$ as Gamma distributed. With this formulation, the estimation remains analytically tractable, an important consideration for our requirements of large-scale implementation.

2.4.2 Specifying the Evolution Variance

When fitting a Dynamic Linear Model, there are a number of parameters that may be tuned (or informed with prior knowledge) that affect the behavior of the fit. Perhaps the most important of these is the discount factor, $\delta$, which determines the smoothness of the model, essentially how quickly the state may adapt to changes in signal. Following West and Harrison (1997), we specify the discount factor as a number between 0 and 1, representing the proportion of the state vector’s precision that is retained through each time step. In other words, at each time step, the variance of our estimate of the state vector increases in scale by a factor of $\delta^{-1}$.

2.4.3 Outlier Detection & Removal

In order to accommodate outliers, we include a three-step procedure. First, we fit the model including all data. Then, we use the leave-one-out residuals of the fitted
model to flag outliers. Essentially, for each observation we ask, what is the probability of the model producing a value at least this extreme given the rest of the observations? Then, we filter out all of the observations that fall below a chosen probability threshold. Finally, we refit the model with these observations excluded.

J. Harrison and West (1991) developed a method to compute the leave-one-out distributions

\[
p(Y_t | Y_1, \ldots, Y_{t-1}, Y_{t+1}, \ldots, Y_T)
\]

for each \(t\), all in a single pass through the smoothed estimates of the model. In other words, it is not necessary to refit the model \(T\) times, leaving out one observation each time. Without this development, it would be infeasible to filter outliers in this way.

In Figure 2-5, we show the DLM fit to our representative pixel, with detected outliers marked in red. Already we see an improvement over the fit offered by CCDC. The model more readily adapts to the gradual changes between 1995 and 2010. Note, however, that the there is still room for further improvement in both the robustness to outliers and accommodation of abrupt changes.

One further advantage of the DLM is its extensibility. As a full statistical model it allows us to directly address issues with the data by adjusting the model, rather than adding a separate procedure to the analysis. Chapters 3 and 4 introduce model extensions for accommodating outliers and extreme, sudden changes. By incorporating these extensions within the model, rather than as separate procedures, we preserve the connected uncertainty structure and permit straightforward statistical inference.
Figure 2-5: A Dynamic Linear Model fit to the representative pixel. Shaded region represents the 90% symmetric credible interval. The color of each point indicates whether that point was flagged and removed by our outlier detection procedure.
2.5 Applied Discounting Approach

Choosing the discount factor can be difficult, especially in the presence of outliers and abrupt changes in signal. There is an inherent tradeoff between how well the model can adapt to sudden changes, and how robust the model is in the face of extreme outliers. The discount factor selection process is further complicated in our case by the scale at which we are fitting the model: it is simply infeasible to check the model fit for every one of the millions of pixels. To solve this problem, we perform cross validation over a set of representative pixels, optimizing the model fits across these pixels over a range of discount factors.

When our state vector consists of multiple components, be they mean, trend, or seasonal, we often would like for different parts of the state vector to be more stable and more resistant to change than other parts. For example, we may have reason to believe that the seasonal pattern of the series should be more stable than the dynamic mean - in other words, small disturbances are better explained by small changes in the dynamic mean, rather than a shift in the seasonal phase or amplitude. To achieve this behavior, we use West and Harrison (1997)'s suggestion of component discounting. With component discounting, we discount the different components of the model with separate discount factors.

In practice, we find it easiest to construct a block diagonal matrix $D$ where each block is filled with the corresponding component’s discount coefficient $d$, with $d = \delta^{-1} - 1$. For example, if we have a mean, trend, and two seasonal components, and we want to discount the mean and trend together with discount factor $\delta_m$, and the two seasonal components separately with discount factors $\delta_s_1$ and $\delta_s_2$, we would construct $D$ as follows:
$$d_m = \delta_m^1 - 1$$
$$d_{s1} = \delta_{s1}^1 - 1$$
$$d_{s2} = \delta_{s2}^1 - 1$$

$$D = \begin{pmatrix}
    d_m & d_m & 0 & 0 & 0 & 0 \\
    d_m & d_m & 0 & 0 & 0 & 0 \\
    0 & 0 & d_{s1} & d_{s1} & 0 & 0 \\
    0 & 0 & d_{s1} & d_{s1} & 0 & 0 \\
    0 & 0 & 0 & 0 & d_{s2} & d_{s2} \\
    0 & 0 & 0 & 0 & d_{s2} & d_{s2}
\end{pmatrix}$$

Then, to apply the component-wise discounting and obtain $W_t$, we perform the operation

$$W_t = D \odot (GC_t \ G')$$

where $\odot$ represents the Hadamard product, also known as element-wise multiplication.

Component discounting offers extra flexibility to the DLM compared to a single discount factor, but it comes with a cost. Instead of only choosing one tuning parameter to determine the smoothness of the model, we now have to tune all of the component discount factors together.

In Figure 2-6, we provide a comparison between different levels of discounting. Note that, for clarity of comparison, this is a simplified view into the discount factor selection process: we do not detect and remove outliers, and we also discount the seasonal components (first and second harmonics) together, with a shared discount factor.
Figure 2.6: DLM fits of the same representative pixel with varying levels of discounting. The rows of the grid are nested with seasonal discounting inside of mean/trend discounting: the first three rows show a low mean/trend discount factor, with progressively larger seasonal discounting, the next three have medium mean/trend discounting, and the last three have a large degree of mean/trend discounting.
The first three rows of the plot correspond to a relatively low discount factor for the mean and trend components of the state vector. All three of these are clearly undersmoothed and overfit - particularly so for the first of the three, where the discount factor for the seasonal components is also low: this model is capable of almost perfectly tracking through every single point.

On the other end, if we look at the bottom three rows, we see the results of a high discount factor for the mean and trend components. In the very bottom row, both the mean/trend and seasonal component discount factors are high, and the resulting fit is oversmoothed. Arguably this fit is passably adequate for capturing the basic pattern of the series, and is certainly better than the overfit model in the first row.

Looking at the first of the bottom three rows (the seventh row), we see a high mean/trend discount factor with a low seasonal discount factor. This results in the seasonal component stretching and rapidly adapting in order to cover for the inflexibility of the mean and trend component. As a general rule, for data such as this where the seasonal pattern is relatively strong and stable, we prefer a seasonal discount factor that is larger than the mean/trend discount factor.

Looking finally to the middle three rows, we see a medium value for discounting the mean and trend components. In particular, the very middle row also has a medium value for seasonal discounting, and arguably offers the most representative fit which best captures the underlying signal. This fit is able to maintain a relatively stable pattern, while also adapting to the change in mean around 2005 and the change in mean and seasonality around 2010.

2.5.1 Further notes on discounting

When encountering missing data, we need to adjust the discounting procedure to prevent our precision from decaying too quickly. Without adjustment, \( W_t \) grows
exponentially over ranges of missing data. To rectify this we follow the suggestion of West and Harrison (1997) to only update $W_t$ after non-missing observations. Thus, if $Y_t$ is missing, we simply set $W_{t+1} = W_t$ instead of updating with the discount factor approach. This solution causes the variance to grow linearly over stretches of missing data, rather than exponentially.

One thing to note about setting the discount factor is that it is dependent on the temporal resolution of the model. For example, with Landsat data, our coarsest possible discretization is one time step per eight days. If we are interested in producing daily estimates, we can fit the model with one time step per day, and fill in all off-cycle days as missing data. However, since the discount factor specifies the proportion of the precision of the state vector that is lost at every time step, our daily model will discard information at eight times the rate of the eight-day increment model if we keep the same discount factor.

To achieve the same degree of discounting at different time resolutions, we provide here a simple formula. Let $\delta$ be the discount factor that we are satisfied with at our current time resolution. If we want to change the time resolution to $n$ times the current observation rate (i.e. $n$ time steps per current time step), we may specify our new discount factor $\delta_n$ as

$$\delta_n = \frac{n}{\frac{1}{\delta} - 1 + n} \quad (2.8)$$

### 2.6 Fitting the Dynamic Linear Model

The procedure for fitting the standard Dynamic Linear Model is detailed in depth by West and Harrison (1997), though we will cover the algorithm here for the purpose of notation and clarity of implementation.
Fitting the model consists of two sequential steps: filtering, then smoothing. Filtering is a forward pass through the data, accumulating the information to produce the online, filtered estimates of the state vector $\theta_t \mid Y_{1:t}$ for each $t \in 1, \ldots, T$.

Once the filtered estimates have been computed, the series may then be smoothed. Smoothing the series is a backward pass through the filtered estimates that carries the information from later observations to update the earlier estimates with the benefit of hindsight. This produces the sequence $\theta_t \mid Y_{1:T}$ for all $t \in 1, \ldots, T$.

Algorithm 1 details the implementation for the filtering procedure. Algorithm 2 outlines the implementation for the smoothing process. Note that the smoothing algorithm requires as input certain parts of the output of the filtering algorithm. In particular, $m_t, C_t, a_t, R_t,$ and $S_t$ for each $t \in 1, \ldots T$ are used.

### 2.7 Computational Considerations

The challenges with pure scale of data outlined in Section 2.1.3 necessitate a high-performance implementation of the DLM algorithm. Petris (2010) provides an R package that implements the core procedures in C, with an R interface. This package was written before the advent of Rcpp (Eddelbuettel & Francois, 2011), and the C code is difficult to read, and therefore difficult to extend. For these reasons, we introduce a new implementation that leverages Rcpp, and crucially RcppArmadillo (Eddelbuettel & Sanderson, 2014) for performant and legible linear algebra operations. In addition, we utilize the BU Shared Computing Cluster to fit the model at scale.
Algorithm 1 Standard Discount DLM Filtering

Input:
Y - The series of observations
F, G - Observation map, evolution matrix
D - Discount matrix
m₀, C₀ - Posterior mean, variance of θ₀
S₀ - Prior point estimate of V
n₀ - Prior weight of S₀ (prior number of observations)

Output:
For each of the below, 0 ≤ t ≤ T
Wₜ - Discounted evolution error covariance matrix at time t
aₜ, Rₜ - Prior mean, variance of θₜ
fₜ, Qₜ - One-step forecast mean, variance at time t
mₜ, Cₜ - Posterior (filtered) mean, variance for θₜ
Sₜ - Posterior point estimate for V up to time t
nₜ - Number of observations seen up to and including time t

1: function FILTERDLM(Y, F, G, D, m₀, C₀, S₀, n₀)
2: T ← length(Y)
3: for t ← 1 to T do
4:     P ← GCᵢ G'
5:     if Yᵢ is not missing then
6:         Wᵢ ← P ⊙ D
7:     else
8:         Wᵢ ← Wᵢ₋₁
9:     end if
10:    aᵢ ← Gmᵢ₋₁
11:    Rᵢ ← P + Wᵢ
12:    fᵢ ← F'aᵢ
13:    Qᵢ ← F'RTF + Sᵢ₋₁
14:    if Yᵢ is not missing then
15:        nᵢ ← nᵢ₋₁ + 1
16:        eᵢ ← Yᵢ - fᵢ
17:        A ← 1/nᵢRTF
18:        mᵢ ← aᵢ + Aeᵢ
19:        Sᵢ ← Sᵢ₋₁(1 + 1/nᵢ(Fᵢ/RTF - 1))
20:        Cᵢ ← Sᵢ₋₁(Rᵢ - QᵢAA')
21:    else
22:        mᵢ ← aᵢ, Cᵢ ← Rᵢ, Sᵢ ← Sᵢ₋₁, nᵢ ← nᵢ₋₁
23:    end if
24: end for
25: end function
Algorithm 2 Standard Discount DLM Smoothing

Input:
- filt - The combined outputs from the filtering algorithm
- F - Observation map
- G - Evolution matrix

Output:
- $\bar{a}_t$ - Smoothed mean of $\theta_t$
- $\bar{R}_t$ - Smoothed variance of $\theta_t$
- $\bar{f}_t$ - Smoothed mean response
- $\bar{Q}_t$ - Smoothed variance of the mean response

1: function SmoothDLM(filt, F, G)
2: T ← LENGTH(Y)
3: $\bar{a}_T$ ← m_T
4: $\bar{R}_T$ ← C_T
5: $\bar{f}_T$ ← F' $\bar{a}_T$
6: $\bar{Q}_T$ ← F' $\bar{R}_T$ F
7: for t ← T − 1 to 1 do
8: B ← C_t G'R_{t+1}^{-1}
9: $\bar{a}_t$ ← m_t + B ($\bar{a}_{t+1} - a_{t+1}$)
10: $\bar{R}_t$ ← C_t + B ( $\bar{R}_{t+1} - R_{t+1}$ ) B'
11: $\bar{f}_t$ ← F' $\bar{a}_t$
12: $\bar{Q}_t$ ← $\frac{1}{B}$ F' $\bar{R}_t$ F
13: end for
14: end function
Chapter 3

Robustifying the DLM to Outliers and Soft Changepoints

The simple DLM approach detailed in Chapter 2 is computationally efficient and in most cases offers a significantly improved model fit in comparison to the CCDC model that is often used in practice. However, when there is more computational power available, a more flexible model can more faithfully capture the signal when certain types of behavior are present, in particular large outliers and changepoints.

In this chapter, we detail methods for handling outliers and large, sudden changes in pattern within a single cohesive model by adding latent scales to the observation and evolution variances. Further, we explore the inherent tradeoff introduced when modeling these two types of behavior together.

3.1 Outlier Robustness

The posterior predictive probability - based outlier removal technique used in Section 2.4.3 is generally effective in practice and has a low computational burden. However, the single pass, leave-one-out detection method can result in false positives. For instance, observations that continue the existing pattern but are close in time to an extreme outlier can be flagged and removed along with the outlier itself. This happens because in the leave-one-out residuals, when we evaluate the in-pattern observation, the model is fit with this observation removed, but the outlier is included. Thus, the
model is dragged towards the outlier, and in extreme cases can be dragged far enough that the in-pattern observation appears to be an outlier.

This behavior results from the leave-one-out method operating on each observation individually, without taking into account whether other nearby observations are outliers. There is room then for a more robust method, one that jointly considers all observations together.

3.1.1 Outlier Robustness Previous Work

West and Harrison (1997) introduce a method for accommodating outliers within the model by relaxing the normality assumption on the observational errors. They do so by scaling the observational variance $V$ at each time $t$ by independent, unknown weights $q_t$, $t \in 1:T$.

The new observation equation takes the form:

$$Y_t = F'\theta_t + \nu_t, \quad \nu_t \sim N(0, V q_t^{-1})$$

If, a priori, $q_t \sim \text{Gamma}(\tau_2, \tau_2)$, then, marginal over $q_t$, the likelihood $Y_t \mid \theta_t, V$ becomes T-distributed with $\tau$ degrees of freedom. Fattening the tails in this way provides the desired robustness to outliers. Figure 3·1 details visually the structure of the robust Dynamic Linear Model.

West and Harrison introduce this approach in a simulation-based, Markov chain Monte Carlo context. The sampling approach has many benefits and we will take advantage of this and expand upon it in later chapters. However, sampling also brings a larger computational burden. Therefore we also propose here an Expectation-Maximization-like procedure to estimate this model. The algorithm is computationally manageable, typically taking only a few iterations to converge to a reasonable
Figure 3.1: State space diagram for the Robust Dynamic Linear Model

Ting et al. (2007) propose a similar method rooted in the Kalman filter theory. Their approach assumes multivariate observations $Y_t$, as well as unknown (but constant) evolution error covariance matrix $W$. These assumptions force an approximation in the E-step of their algorithm. Further, since their approach is focused on real-time object tracking, they limit their method to only consider past observations. In other words, they do not revise their estimates of previous time points once new data is observed. While this choice makes sense for their purposes, using the full set of observed information is required for retrospective inference, when we would like to make claims about the state of the pattern at some point in the past.

By comparison, our method makes use of the variance discounting procedure to set $W_t$ and thus does not require an approximation. We also estimate the full smoothed posteriors, using all observations in the series. In addition, the algorithm is computationally manageable, typically taking only a few iterations to converge to a reasonable estimate.

Figure 3.2 shows a Robust DLM fit on our representative pixel, fit using the Expectation-Maximization-like procedure. Note that the model is able to more con-
Figure 3.2: A robust Dynamic Linear Model fit to the representative pixel, with $\tau$ set to 15 using the Expectation Maximization-like method. The solid line represents the EM-like MAP for the mean of the signal. The color of each point shows the weight for that observation ($q_i$).

sistently disregard points that clearly (to the eye-test) stray from the signal. However, there is a drawback – the model is now more resistant to accommodating the abrupt change in signal that occurs around the start of 2010. This resistance to accommodating change arises inherently from adding flexibility to downweight observations that depart from the current pattern.

3.1.2 Estimating the Robust DLM

After adding the observational variance scales to the model, we can no longer exploit conjugacy to estimate $\theta$ analytically. We instead turn to two approaches for estimation: an expectation-maximization-like algorithm, and a Gibbs-sampling procedure.
Expectation-maximization-like algorithm

In the EM-like approach, we aim to produce a joint maximum-a-posteriori (MAP) estimate for \( \theta_{1:T} \) and \( \phi \), marginal over \( q_{1:T} \). Starting from the joint posterior over all parameters:

\[
p(\theta_0, \ldots, \theta_T, q_1, \ldots, q_T, \phi \mid Y_1, \ldots, Y_T) \propto p(\theta_0)p(\phi) \prod_{t=1}^{T} p(Y_t \mid \theta_t, q_t, \phi)p(\theta_t \mid \theta_{t-1})p(q_t) \]

\[
= \frac{1}{\sqrt{(2\pi)^r |C_0|}} \exp \left( -\frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \right) = \phi^{(m_0 S_0^{-1})} \exp(-S_0 \phi) \]  

In A.1 we derive the M- and E-steps of the algorithm, and see that, for the M-step, maximizing \( Q(s) (\theta, \phi) \) is equivalent to maximizing the posterior for \( \theta \) and \( \phi \), conditional on the previous values of \( q \) computed in the E-step. Since \( \theta \) is Normal, the mode is equal to the mean.

In practice, for the M-step, the procedure is to compute the smoothed posterior means for \( \theta \) using the standard DLM algorithm with a single small change. In the filtering step, at each \( t \), the observation variance scale \( q_t^{(s)} \) enters into the one-step forecast variance \( Q_t \) so that instead of

\[ Q_t \leftarrow F'R_t F + S_t^{-1} \]

we have:

\[ Q_t \leftarrow F'R_t F + \frac{S_t}{q_t^{(s)}}^{-1} \]

The rest of the filtering algorithm and the smoothing algorithm remain unchanged. Then, from the output of this procedure, we take the smoothed posterior means and
they become our MAP estimate for $\theta$ at iteration $(s)$. Similarly, we take the posterior for $\phi$ and extract the mode as our MAP estimate $\phi^{(s)}$. It is important to note that each of these estimates are in fact marginal MAP estimates and not the joint maximum. Ideally, we will need the joint maximizers $\theta$ and $\phi$ for the M step.

To compute the true joint argmax, we can perform coordinate ascent on $\theta$ and $\phi$, conditional on the current set of $q$ weights. The basic process would be to fix $\phi$ at some starting point, then fit the standard DLM with known $q$ weights, and also known global observational precision $\phi$. Fitting this model results in $\theta$ multivariate-normally distributed, so we may take the means of the posterior ($m_t$) as our maximizers. Then, conditional on these fixed $\theta$s, $\phi$ is Gamma-distributed:

$$\phi \mid Y_{1:T}, \theta_{1:T}, q_{1:T} \sim \mathcal{G} \left( \frac{n_T}{2}, \frac{n_0 S_0 + \sum_{t=1}^{T} q_t (Y_t - F^t \theta_t)^2}{2} \right)$$

and therefore we may maximize at the mode of $\frac{n_T}{n_0 S_0 + \sum_{t=1}^{T} q_t (Y_t - F^t \theta_t)^2}$. Thus, to find the joint MAP of $\theta, \phi \mid q^{(s)}$, we may iterate between these two steps until convergence of $\theta$ and $\phi$. At convergence, the M-step is complete and we have our $\theta^{(s)}$ and $\phi^{(s)}$.

In practice, we may choose to forego the coordinate ascent approach, and instead simply take the marginal posterior modes of $\theta$ and $\phi$ as our maximizers in the M-step as an approximation. Performing the coordinate ascent requires fitting the DLM multiple times per M-step, and often is not worth the cost, as the approximation is generally fairly accurate.

For the E-step, the posterior for each $q_t$ is explicitly Gamma distributed conditional on $\theta^{(s)}$ and $\phi^{(s)}$: 
\[ q_t \mid Y_t, \theta^{(s)}, \phi^{(s)} \sim G \left( \frac{\tau + 1}{2}, \frac{\tau + \phi^{(s)} \left( Y_t - F' \theta^{(s)}_t \right)^2}{2} \right) \]

so that

\[ \mathbb{E}_{q_t|Y_t,\theta^{(s)},\phi^{(s)}} [q_t] = \frac{\tau + 1}{\tau + \phi^{(s)} \left( Y_t - F' \theta^{(s)}_t \right)^2} \]

Thus, we have our M-step and our E-step, and the algorithm is complete. Convergence for this model is generally rapid, especially with the approximate M-step, typically taking only a few iterations.

**Algorithm 3** Robust Discount DLM M-Step

```plaintext
1: function ROBUSTDLMMSTEP(Y, F, G, D, m_0, C_0, S_0, n_0, q)
2:   filt ← FILTERROBUSTDLM(Y, F, G, D, m_0, C_0, S_0, n_0, q)
3:   smooth ← SMOOTHDLM(filt, F, G)
4:   \( \theta_{\text{max}} \leftarrow \text{smooth} \cdot \hat{a} \)
5:   \( V_{\text{max}} \leftarrow \text{filt} \cdot S_T \)
6:   return \( \theta_{\text{max}}, V_{\text{max}} \)
7: end function
```

**Algorithm 4** Robust Discount DLM E-Step

```plaintext
1: function ROBUSTDLMESTEPQ(\( \tilde{Y}, \tau, V \))
2:   for \( t \leftarrow 1 \) to \( T \) do
3:     if \( Y_t \) is missing then
4:       \( q_t \leftarrow 1 \)
5:     else
6:       \( \text{sq err} \leftarrow (\tilde{Y}_t - Y_t)^2 \)
7:       \( \text{scl err} \leftarrow \frac{\text{sq err}}{V} \)
8:       \( q_t \leftarrow \frac{\tau + 1}{\tau + \text{scl err}} \)
9:     end if
10:   end for
11: end function
```

35
Algorithm 5 Robust Discount DLM EM

Input:
Y, F, G, D, m₀, C₀, S₀, n₀ - Same as in standard filtering algorithm
τ - Hyper prior for q

Output:
θ_{max} - MAP estimate for θ
V_{max} - MAP estimate for V
q - Final expected value for each qₜ, useful for measuring weighting

1: function ROBUSTDLMEM(Y, F, G, D, m₀, C₀, S₀, n₀, τ)
2: q ← [1, ..., 1]  \triangleright initialize all qₜ at 1
3: while θ_{max}, V_{max} not converged do
4: θ_{max}, V_{max} ← ROBUSTDLMSTEP(Y, F, G, D, m₀, C₀, S₀, n₀, q)
5:  \bar{f} ← F'θ_{max}
6: q ← ROBUSTDLMSTEPQ(\bar{f}, Y, τ, V_{max})
7: end while
8: return θ_{max}, V_{max}, q
9: end function

3.1.3 Gibbs Sampling Procedure

In each of our Gibbs sampling methods, we make use of the forward-filtering, backward sampling algorithm to draw sample paths of θ. This algorithm is explained in detail in West and Harrison (1997). The algorithm begins with the general filtering step (optionally altered to accommodate variance scales), and randomly samples a path backwards, rather than smoothing. This generates valid samples from the joint posterior (θ₁, ..., θₜ | Y).

The robust DLM Gibbs sampling procedure then consists of three steps: sampling from \theta, \phi, and q. We use the forward filtering, backward sampling approach to sample from \theta, conditional on \phi and q. Next, we may sample \phi conditional on \theta and q, via the following Gamma distribution:

\phi | \theta, Y, q, \alpha \sim G \left( \frac{n₀}{2}, \frac{n₀S₀ + \sum_{t=1}^{T} qₜ (Yₜ - F'θₜ)^2}{2} \right)

Finally, we sample from qₜ | Y, \theta, \phi, t ∈ 1, ..., T from the following Gamma
distribution:

\[ q_t | \theta, Y, \phi \sim G\left( \frac{\tau + 1}{2}, \frac{\phi (Y_t - F'\theta_t)^2 + \tau}{2} \right) \]

Sampling from each of these three conditional distributions sequentially, we are able to draw from the full joint posterior \( p(\theta, \phi, q | Y) \).

### 3.2 Soft changepoints

The robust DLM effectively handles our first challenge motivated by satellite imagery data - outliers arising from cloud cover and cloud shadows. The next challenge is to empower the model with the capability to adjust to large, abrupt changes in pattern. Being a dynamic model, the DLM is already equipped with the machinery to adapt to changes in the state vector. However, these changes are forced to be small - otherwise the model will be unstable and overadaptive.

The presence of an instantaneous changepoint in the series greatly exaggerates the tradeoff in choosing a proper discount factor. When the adaptability is set high, the model can accommodate large changes in a very short period of time. However, the model will also chase every noisy whim of the series and call it signal. When the adaptability is set lower, the model offers a stable, more credible representation of the underlying signal. At a changepoint location, though, the model is very slow to adapt, resulting in significant loss of fidelity near the changepoint.

The root of the problem is the fact that, in the standard dynamic linear model, the model is typically forced to be uniformly adaptive across the full series.
3.2.1 Changepoint Previous Work

P. J. Harrison and Stevens (1971) have one such solution for this problem: multi-process dynamic linear models (P. J. Harrison & Stevens, 1971; P. J. Harrison & Stevens, 1976). The multi-process extension to the dynamic linear model is essentially a mixture model of dynamic linear models. The core idea is to have multiple dynamic linear models, each capable of capturing a different pattern or process. For example, the set of models could consist of a default model, an outlier model, and a change model. The default model would fit well over most portions of the series. The outlier model would, by virtue of an inflated observational variance, be able to better model extreme observations that are less informative about the true signal. Lastly, the change model would be set with an inflated evolution variance, allowing for rapid adaptation to sudden changes in the pattern of observations.

It is then assumed that at any given time \( t \), only one of these models is active and determines the process at that time step. Typically the transitions between these models over time is assumed Markov, or fully independent with constant probabilities for the different models at each time point.

This mixture approach has the benefit that different patterns of the series (regular, outlier, or changepoint) can be explicitly modeled with different behavior. However, the structure of the model as a mixture leads to a combinatoric explosion in the number of models that must be fit, and also imposes a certain rigidity that can handicap the robustness of the overall model. For example, in the case of an extreme outlier, the “best” the model can do is to consider that observation as having been produced by the ‘outlier’ model. While the ‘outlier’ model has an inflated observational variance compared to the standard model, it is still normally distributed. Thus, an extreme outlier can result in an arbitrarily large error, pulling the model towards the outlier.
One natural way to address this within a DLM is to relax the assumption that the evolution errors are normally distributed. Several previous methods have been introduced following this idea. A common approach is to replace the normal distribution with a mixture of normal distributions – typically two zero mean normals, one with larger variance than the other.

Carlin et al. (1992) introduced a general framework and sampling procedure for nonlinear, nonnormal state space models. Frühwirth-Schnatter (1994) investigated in particular $d$-inverse-gamma-models, where variances of the observation and evolution errors are all scaled by a shared, unknown inverse-gamma-distributed scale. For estimating the model, she proposed an approximation based on the data augmentation algorithm of Tanner and Wong (1987), and importantly for our purposes, introduced a method for sampling posterior draws from the sequence of state vectors $\theta_{1:T}$.

### 3.2.2 Robust Soft-Changepoint DLM

Instead of using a mixture for the evolution variances, we propose to mirror the robust observational error method, adding scales to each of the evolution variances. By fattening the tails of the evolution error, we can keep most of the evolution errors small while still allowing for the possibility that some errors can be arbitrarily large.

In a similar way that adding the $q$s allows the model to accommodate nonnormal errors in the observation process, we add an unknown scale $\alpha_t$ to each evolution variance to accommodate nonnormal errors in the evolution process.

$$\theta_t = G\theta_{t-1} + \omega_t, \quad \omega_t \sim N(0, \alpha_t^1W_t)$$

Further, we specify independent Gamma priors on each $\alpha_t$: 

$$\theta_t = G\theta_{t-1} + \omega_t, \quad \omega_t \sim N(0, \alpha_t^1W_t)$$
\[ \alpha_t \sim \text{Gamma}(\frac{\tau}{2}, \frac{\tau}{2}) \] (3.4)

The resulting model combines variance scales on both the observational and evolution errors with the discounting approach for determining the base evolution error. We call this model the Robust Soft-Changepoint Dynamic Linear Model (RSC DLM). Figure 3.3 shows the structure of the model.

It is important to note here that each \( \alpha_t \) is a scalar quantity, even though \( W_t \) is often a matrix (whenever the state vector \( \theta \) has two or more elements). \( W_t \) specifies the direction and magnitude of the random evolution error introduced to the state vector between time \( t - 1 \) and time \( t \). Thus, \( \alpha_t \) scales the magnitude of these errors, without affecting the direction. Therefore when \( \alpha_t \) is very small, the \( \theta_t \) can vary widely from \( \theta_{t-1} \), across all components of the state vector. One limitation of a scalar \( \alpha_t \) is that we cannot selectively retain or drop information about individual components of the state vector. In other words, it’s not possible to model a level-shift, where the mean component experiences a sudden change, but the seasonality is unaffected.
Figure 3.4: A Robust Soft-Changepoint DLM fit to the representative pixel, with $\tau$ set to 4, and $\gamma$ set to 5. Shaded region again represents the 90% symmetric credible interval. The color of each point shows the weight for that observation ($q_k$).

However, in most cases this information can be relearned relatively quickly.

In Figure 3.4 we return once more to our representative pixel and fit this time a Robust Soft-Changepoint DLM, fit with a Gibbs sampling procedure. We see a great improvement over the basic Robust DLM - the new model manages to both accommodate the abrupt changes, while also remaining robust to the outlying observations.

3.2.3 Estimating the RSC

As with the outlier-robust model, we can estimate the parameters using either Gibbs sampling, or an EM-like procedure.

In cases where speed is valued over fullness of information, an expectation maximization-
like procedure can be used to fit the model. When more computational resources can be spared, a Markov-chain monte carlo approach can be employed to render a fully Bayesian estimate of the model.

In many cases, the optimization approach is sufficient. However, there are some cases where sampling can offer a much fuller picture into the multi-modal structure that can arise when modeling changepoints and outliers. Section 3.3 explores this issue in greater depth.

**Expectation-Maximization-like procedure**

The EM-like approach for the RSC model follows the mold of the robust DLM. The M-step again maximizes over \( \theta \) and \( \phi \), but our E-step now includes both \( q \) and \( \alpha \) as well. We start once again from the joint posterior over all parameters:

\[
p(\theta_1, \ldots, \theta_T, q_1, \ldots, q_T, \alpha_1, \ldots, \alpha_T, \phi \mid Y_1, \ldots, Y_T) \quad (3.5)
\]

In fitting the DLM during the M-step, we maintain the change from the robust DLM, where we scale the contribution of the observation variance to the one-step forecast variance \( Q_t \). In addition, we now also scale the evolution variance that contributes to the one-step prior variance, \( R_t \). So, in place of

\[
R_t \leftarrow GC_t^{-1} G' + W
\]

we instead have

\[
R_t \leftarrow GC_t^{-1} G' + \frac{1}{\alpha_t} W
\]

The remainder of the filtering algorithm and smoothing algorithm remain unchanged. Algorithm 6 shows the complete filtering procedure. The M-step of the
algorithm, shown in Algorithm 7, is nearly identical to the M-step for the robust DLM, with the only difference being the slight alteration to the filtering step.

As with the robust DLM M-step, we may either use coordinate ascent to maximize over \( \theta \) and \( \phi \), or we may approximate with the marginal posterior modes.

### 3.2.4 Gibbs Sampling Procedure

The Gibbs sampling procedure for the Robust Soft-Changepoint DLM follows nearly exactly the procedure used for the Robust DLM. The only changes are the use of the RSC filtering algorithm (adjusting the evolution variances with the \( \alpha \) scales) during the forward-filtering, backward sampling step for \( \theta \), and sampling the vector \( \alpha \) itself. Note in particular that the process for sampling \( \phi \) and \( q \) remain completely unchanged from the Robust DLM sampling procedure.

We may sample \( \alpha \) conditional on \( \theta \) via the following Gamma distribution:

\[
\alpha_t \mid \theta_t, \theta_{t-1} \sim \mathcal{G} \left( \frac{\gamma + r}{2}, \frac{\gamma + (\theta_t - G\theta_{t-1})' W_t^{-1} (\theta_t - G\theta_{t-1})}{2} \right)
\]

Thus, the full sampling procedure starts with sampling from the posterior for \( \theta \) conditional on \( \alpha \), \( q \), and \( \phi \) using the forward-filtering, backward sampling algorithm where the RSC filtering approach is used in the filtering step. Next we may sample \( \phi \) conditional on \( \theta \) and \( q \). Then, \( q \) may be sampled conditional on \( \theta \) and \( \phi \). Finally we may sample \( \alpha \) conditional on \( \theta \). By continually sampling from these four conditionals, we have a proper Gibbs sampler that produces draws from the joint posterior.

### 3.3 Tau-Gamma Tradeoff

In the standard DLM, tuning the discount factor is critically important to ensure the model appropriately fits the data. The proper tuning of \( \tau \) and \( \gamma \) is likewise necessary.
Algorithm 6 RSC Discount DLM Filtering

Input:
Y, F, G, D, m₀, C₀, S₀, n₀ - Same as in standard filtering algorithm
For each of the below, 0 ≤ t ≤ T
- qₜ - Observational variance scale at time t, produced from the E-step
- αₜ - Evolution variance scale at time t, produced from the E-step

Output:
For each of the below, 0 ≤ t ≤ T
- aₜ, Rₜ, Wₜ, fₜ, Qₜ, mₜ, Cₜ, Sₜ, nₜ - Same as the standard filtering algorithm

1: function FILTERRSCDLM(Y, F, G, D, m₀, C₀, S₀, n₀, q, α)
2:   T ← LENGTH(Y)
3:   for t ← 1 to T do
4:     P ← GC⁻¹G'
5:     if Yₜ is not missing then
6:       Wₜ ← P ⊙ D
7:     else
8:       Wₜ ← Wₜ⁻¹
9:     end if
10:    aₜ ← Gmₜ⁻¹
11:    Rₜ ← P + 1/αₜWₜ
12:    fₜ ← F'aₜ
13:    Qₜ ← F'RₜF + 1/qₜSₜ⁻¹
14:    if Yₜ is not missing then
15:      nₜ ← nₜ⁻¹ + 1
16:      eₜ ← Yₜ - fₜ
17:      A ← 1/qₜRₜF
18:      mₜ ← aₜ + Aeₜ
19:      Sₜ ← Sₜ⁻¹(1 + 1/nₜ(eₜ²/qₜ - 1))
20:      Cₜ ← Sₜ⁻¹(Rₜ - QₜAA’)
21:    else
22:      mₜ ← aₜ, Cₜ ← Rₜ, Sₜ ← Sₜ⁻¹, nₜ ← nₜ⁻¹
23:    end if
24:   end for
25: end function
Algorithm 7 RSC Discount DLM M-Step

1: function RscDlmMStep(Y, F, G, D, m₀, C₀, S₀, n₀, q, α)
2: filt ← FilterRscDlm(Y, F, G, D, m₀, C₀, S₀, n₀, q, α)
3: smooth ← SmoothDlm(filt, F, G)
4: θ_max ← smooth.α
5: V_max ← filt.Śₜ
6: return θ_max, V_max
7: end function

Algorithm 8 RSC Discount DLM E-Step (α)

1: function RscDlmEStepα(ā, Y, W, γ, G)
2: for t ← 1 to T do
3: d ← āⱭ − G āⱭ₋₁
4: maha ← d′Wₜ₋¹d
5: αₜ ← γ+ nrows(G) γ+maha
6: end for
7: return α
8: end function

Algorithm 9 RSC Discount DLM EM

1: function RscDlmEM(Y, F, G, D, m₀, C₀, S₀, n₀, τ, γ)
2: q ← [1, ... , 1]
3: α ← [1, ... , 1]
4: while θ_max, V_max not converged do
5: θ_max, V_max ← RscDlmMStep(Y, F, G, D, m₀, C₀, S₀, n₀, q, α)
6: ñ ← F′θ_max
7: q ← RobustDlmEStepQ(ñ, Y, τ, V_max)
8: α ← RscDlmEStepα(θ_max, Y, W, γ, G)
9: end while
10: return α
11: end function
These two new parameters add flexibility in competing ways. In a broad sense, a low value for $\tau$ allows the model to downweight and ignore observations that break the pattern of surrounding observations. A low value for $\gamma$, on the other hand, encourages the model to instead upweight these incongruous observations, to adapt to a potential change in signal.

The balance of these two parameters, then, determines how the model will tend to respond to these types of events. When the ratio $\tau/\gamma$ is small, the model will lean towards downweighting the observation(s). When large, the model will favor rapid adaptation. Lastly, if both $\tau$ and $\gamma$ are large, the model will tend to neither downweight nor upweight the incongruous observations, and instead will adapt gradually using the standard DLM machinery.

Figure 3.5 illustrates this tradeoff with a simple box function. For this example, we purposefully choose $\tau$ and $\gamma$ in a manner that encourages multimodality in the posterior. At this point, the model is ‘on the fence’: both scenarios (adapting abruptly or downweighting) are tenable.

Finding the optimal balance of these parameters can be difficult. To address this, we provide guidelines for eliciting the priors on $\tau$ and $\gamma$ from a practitioner based on desired behavior of the model under specific circumstances with incongruous observations.

### 3.3.1 Guidelines for hyperprior elicitation

Looking at the box function,

$$Y_t = B(t), \quad t \in 1, 2, \ldots, T$$

where
Figure 3.5: A Robust Soft-Changepoint model fit to a simple box function with small noise. The top left shows the series, with four locations of interest marked by color. Bottom left shows the same series, with sample paths of the posterior for $\theta$ drawn overlaid. On the right, the three plots show the density of samples for $\theta$, $\log(\alpha)$, and $\log(q)$ at each of the locations of interest.
Figure 3.6: A simplified look into the RSC’s choice when encountering a box function. Either the model adapts to the middle points (red), or it downweights them (blue), depending on the balance of $\tau$ and $\gamma$. 
The sequence \( Y = (Y_1, \ldots, Y_T) \) then consists of three parts: \( a - 1 \) consecutive zero values, followed by \( n = b - a \) points with value \( M \), followed by \( T - b + 1 \) more zero values. Assume that \( n \ll a - 1 \) and \( n \ll T - b + 1 \), so that a reasonable model will estimate \( \theta_t \approx 0 \) for the majority of the series.

A traditional DLM will model this series by attempting to split the difference between 0 and \( M \) through the middle \( n \) points. If \( M \) is extreme (very large relative to \( V \) and \( W \)), this will be a suboptimal fit to the data. The crucial issue is that the jump is too large relative to the behavior of the rest of the series. With \( M \gg V \), the points are outside of the reasonable window expected from normally-distributed observational errors centered near zero. Thus, the model must adapt to these points. With \( M \gg W \), the jump from 0 to \( M \) is too large to be adapted to from a single normally-distributed evolution error. Thus, the gaps between \( Y_{a-1} \) and \( Y_a \), and \( Y_{b-1} \) and \( Y_b \) cannot be quickly adapted to, and therefore the model must spread the adaptation out over a significant stretch of points. This disrupts the fit near \( a \) and \( b \) by forcing the model to cut corners in order to adapt to the middle points.

An apparent solution to this problem would be to raise either \( V \) or \( W \) (via the prior on \( V \) or the discount factor), so that the gap falls more comfortably within the normally-distributed window of either the observational or evolution errors. The issue with this of course is that the original \( V \) and \( W \) were appropriate for the rest of the series. Thus, a raised \( V \) will result in the model being underadaptive over the majority of the series, and thus unable to capture smaller changes in pattern outside

\[
B(t) = \begin{cases} 
0 & 1 \leq t < a \\
M & a \leq t < b \\
0 & b \leq t \leq T 
\end{cases}
\]
of the middle $n$ points. Raising $W$ will allow the model to rapidly adapt from 0 up to $M$ at time $a$, and then back down at time $b$, but at the cost of overadaptivity across the rest of the series. This is a serious cost, as it weakens our confidence in the model as a stable representation of the underlying pattern.

This example is suited well to the RSC approach, where we replace the normally-distributed errors (observation and evolution) with $t$-distributed errors. Introducing the observational variance scales $q$ gives the model the option to ignore the middle section, so that it may better fit the edges - making the choice that the points in the middle are outliers or otherwise less informative of the state $\theta$. The evolution variance scales $\alpha$ offer the alternative path: abandoning the built-up knowledge that $\theta$ is close to zero, to more closely track through the middle points at value $M$.

Our interest is to characterize the behavior of the model as a function of $\tau$ and $\gamma$.

We consider the simplest case, where $\theta_t$ is the one-dimensional dynamic mean, $F = G = 1$, and $V$ and $W$ are known, constant, and equal to each other, $W_t = W = V = V_t$. Then, for tractability, we make the simplifying assumption that the evolution errors, $\omega_t$, are zero everywhere but $a$ and $b$, where $\omega_a = \theta_m$ and $\omega_b = -\theta_m$. That is, $\theta_t = 0$ for all $t < a$ or $t \geq b$, and $\theta_t = \theta_m$ for $t \in \{a, \ldots, b-1\}$

Consider the two extreme cases: either the model fully adapts, with $\theta_m = M$, or the model fully downweights the middle box observations, with $\theta_m = 0$.

$$
\sum_{t=1}^{T} \left\{ \left( \frac{\tau}{2} - 1 \right) \log q_t - \frac{\gamma}{2} q_t + \left( \frac{\gamma}{2} - 1 \right) \log \alpha_t - \frac{\gamma}{2} \alpha_t \\
+ \frac{1}{2} \log (q_t) - \frac{1}{2} \frac{q_t}{V} (Y_t - F'\theta_t)^2 \\
+ \frac{1}{2} \log \alpha_t - \frac{1}{2} \frac{\alpha_t}{W} (\theta_t - G\theta_t 1)^2 \right\}
$$

With the restrictions imposed on our model, this simplifies to:

50
At the end, this boils down to weighing the cost of $n$ outliers versus 2 changepoints. The priors on $q$ and $\alpha$ ($\tau$ and $\gamma$, respectively) determine the cost of outliers and changepoints, and therefore determine the behavior that minimizes these costs.

To characterize the impact of the $\tau - \gamma$ tradeoff on the behavior of the model, we optimize (3.6) with respect to $\theta_m$, $q_m$, and $\alpha_m$, as a function of $n$, $M$, $\tau$, and $\gamma$.

In Figure 3.7 we see the results of the optimization. We set $V = W = 0.8$ for this demonstration, though in practice $V$ and $W$ may be set according to expert knowledge about the degree of observation error and evolution variance. The color within each subplot illustrates the optimal behavior of the model across varying levels of $\tau$ and $\gamma$, at particular values of $M$ and $n$. Dark blue indicates that the model prefers to treat the middle section of points as outliers, while red indicates that the model prefers strong adaptation.

As one might expect, the model strongly prefers the outlier path in the scenario of a single out-of-pattern observation (the leftmost column of subplots). The model behavior in this case is largely insensitive to the values chosen for $\tau$ and $\gamma$. In order for the model to favor adaptation to the single point, $\gamma$ must be set fairly low, and $\tau$
**Figure 3.7:** A visualization of the $\tau$-$\gamma$ tradeoff in our simplified analysis of the box function. The facetting represents different numbers $n$ of points and different heights $M$ of the middle section of the box function. Color signifies how much $\theta$ adapts within the middle section - dark red shows that $\theta_m \approx M$, dark blue shows that $\theta_m \approx 0$.  

---

52
must be significantly larger than $\gamma$.

With two points in the middle section of the box function (shown in the second column of subplots), the tradeoff becomes symmetric. The balance of $\tau$ and $\gamma$ now directly determine the preferred model behavior: $\tau$ larger than $\gamma$ results in a preference for adaptation, while $\gamma$ larger than $\tau$ causes the model to prefer downweighting.

Once three points are in the middle section, the model begins to generally prefer to adapt, rather than downweight. At this point the model will require coaxing (in the form of a large $\gamma$-to-$\tau$ ratio) in order to ignore the points within the middle section.

With more than three points in the middle section, heavy-handed prior favoritism ($\gamma \gg \tau$) is required if the desired model behavior is downweighting the middle observations.

An extension of this analysis would be to consider a step function instead of a box function. In this case, we would have a large number of points at 0, followed by $n$ points at value $M$. This results in the exact tradeoff as (3.6), but with a coefficient of 1 attached to the second portion of the sum, rather than 2, since only one change is necessary to adapt to the points after the step occurs. Conceptually, this halves the cost of the changepoint pathway as compared to the box function. In effect, this means that the model more readily adapts to outliers at the end of the series.

This comes about because these scenarios are fundamentally different. With the box function, the tradeoff between adapting and downweighting is essentially a definition problem. Given that we know that the series should be at zero before and after the middle section, we must make a choice about whether these middle points are informative about the underlying state.

With the step function, we must make a similar decision – are these points at the end a deviation of the signal, or just a sequence of outliers? However, this decision is burdened by an element absent from the box function scenario: uncertainty of the
future.
Chapter 4

Formal changepoints

The soft changepoint method of Chapter 3 reliably allows the model to accommodate
sharp changes in mean, trend, or seasonality. However, it is an informal changepoint
method. The $t$-distributed observation errors make the model robust to outliers, but
they do not explicitly provide a probability that any individual observation is in fact
an outlier. In the same way, the $t$-distributed evolution errors allow for changepoints
in a practical sense, but do not explicitly provide a probability that any particular
point is in fact a changepoint.

There is a desire for a formal changepoint framework compatible with the DLM,
one that directly assigns probability to individual points being true changepoints.
This framework will open the door for more rigorous inference about the structure of
the modeled timeseries, allowing us to make probabilistic claims about the location
and quantity of changepoints in the series. To build this formal framework, we will
layer a product partition model on top of the DLM.

Changepoint methods seek to identify structural breaks in a time series - locations
where the underlying pattern generating the observed data changes in some manner.
There exist many approaches to changepoint problems, and, as noted by Burg and
Williams (2020), the approaches can be separated by three general criteria: online
vs offline algorithms, univariate vs multivariate, and parametric vs non-parametric.
Online algorithms operate at the end of the time series, generally seeking to answer
the question of whether a change point has occurred within the last few observations, given all of the data observed before then. This is most useful when attempting to quickly detect change points happening in real-time, so that action may be taken to correct the change, or respond to it. Offline algorithms instead seek to look back over the entire series, and, informed by all observations, determine where changepoints may have occurred. This is a retrospective analysis, seeking to best answer what happened in the past, rather than what is happening at the present. The methods that we will present reside squarely in the offline, univariate, parametric category.

4.1 Product Partition Model

The product partition model (Barry & Hartigan, 1993) is a natural approach to a rigorous Bayesian changepoint model. It starts with a prior distribution on any given segmentation of a time series into contiguous chunks. Then, by treating the segments as independent from one another, we get that the likelihood of the given partition is the product of the independent likelihoods within each segment. From there follows the posterior, an allotment of probability on the configuration of changepoints within the series, informed by the prior and the observed data.

Practically, the model is implemented by adding an unknown partition vector, \( \rho = (u_1, u_2, \ldots, u_n) \) where for each time point \( i \) in the series, \( u_i \in \{0, 1\} \). The values of each \( u_i \) determine the location of changepoints: \( u_i = 0 \) means that the observation at time \( i, Y_i \), is not generated from the same process as \( Y_{i-1} \). Conversely, \( u_i = 1 \) means that \( Y_i \) belongs to the same segment as \( Y_{i-1} \). Then, under certain assumptions about the structure of dependence between the elements of \( \rho \), we may compute \( p(u_i = 0 \mid Y, u_{\{j, j \neq i\}}) \), and therefore we can sample \( u_i \) conditional on the rest of the partition and the data itself. Further, given the partition, \( \rho \), we may also
estimate the models within each segment defined by $\rho$.

Barry and Hartigan (1993) consider the case where observations within segment $b$ are assumed to be independent, identically distributed samples from a normal distribution with mean $\mu_b$. So, each segment has its own mean, fully independent from the means of the other segments. Then, given the partition $\rho$, the posterior segment mean $\mu_b \mid Y_b$ is easily attainable as the simple posterior mean of the observations $Y_b$ within segment $b$. Thus, we can also easily sample the posterior mean at any time point, $\mu_i \mid \rho, Y$. This structure allows for a Gibbs sampling procedure, where we alternate sampling first the partition $\rho$, then the segment means $\mu$, each conditional on the other. Sampling repeatedly in this way yields estimates of two important quantities. First, we obtain posterior estimates of the mean $\mu_i$ at every $i$, marginalized over every possible partition. Second, we also obtain the posterior probabilities of any time point $i$ being a changepoint.

This is an effective procedure for many changepoint problems. However, for our use it suffers from a similar problem as the CCDC algorithm: all change must happen at segment boundaries.

### 4.2 PPM-coupled DLM

We offer here a major methodological contribution: instead of modeling each segment with a single mean, we allow each segment to follow an independent DLM. By combining the product partition model with the machinery of the Dynamic Linear Model, we marry the strengths of the PPM as a changepoint model with the within-segment flexibility of the DLM. The primary appeal of this approach is that we may model changes in the latent state through two complementary pathways: small, gradual changes may be captured within segments by the adaptivity of the traditional DLM,
while large, sudden changes can be explicitly modeled as changepoints.

By separating the modeling of change in this way, we reap two benefits. First, by freeing the traditional DLM from the responsibility of adapting to large changes, we may tune the model to be more stable and robust within each segment. On the other side, since the changepoint model is no longer responsible for adapting to small changes (as it is in Barry and Hartigan (1993)), it can be tuned to be less sensitive so that it only engages when encountering large, sudden changes in pattern. Thus, when the model makes use of this pathway, we are given a more clearly interpretable signal of large and sudden change to the latent state.

4.2.1 A direct approach

The first approach for combining these models is to implement the PPM directly, fitting independent DLMs to every possible segment of the series. This requires computing \( p(Y_{ij}) \) for every \( (i, j : 0 < i < j \leq T) \), where \( Y_{ij} = (Y_i, Y_{i+1}, \ldots, Y_j) \). By removing redundant computation, we can acquire all of these segment likelihoods with just \( T \) applications of the DLM algorithm.

\[
\begin{align*}
p (Y_{ij}) &= p (Y_i, \ldots, Y_j) = p (Y_j | Y_{i,j-1}) p (Y_{i,j-1}) \\
&= p (Y_j | Y_{i,j-1}) p (Y_{j-1} | Y_{i,j-2}) p (Y_{i,j-2}) \\
&\quad \vdots \\
&= p (Y_i) \prod_{k=i+1}^{j} p (Y_k | Y_{i,k-1})
\end{align*}
\]

Now, \( p (Y_k | Y_{i,k-1}) \) is directly available \( \forall k \in (i + 1, \ldots, j) \) when fitting the DLM over the observations \((Y_i, \ldots, Y_j)\): these quantities are simply the one-step forecast densities at each time \( k \). Therefore, by fitting the DLM just once over \((Y_i, \ldots, Y_j)\), we
may procure each of $p(Y_{ik}), \forall k \in (i, \ldots, j)$. With this in mind, after fitting the DLM over $(Y_1, \ldots, Y_T)$,

$$(p(Y_{1,1}), p(Y_{1,2}), \ldots, p(Y_{1,T}), p(Y_{1,T}))$$

is readily acquired as the cumulative product of the one-step forecast likelihoods. Fitting over $(Y_2, \ldots, Y_T)$ likewise produces

$$(p(Y_{2,2}), p(Y_{2,3}), \ldots, p(Y_{2,T}), p(Y_{2,T}))$$

Continuing on, the fit over $(Y_{T-1}, Y_T)$ gives

$$(p(Y_{T-1,T}), p(Y_{T,T}))$$

Practically, we seek to fill in the upper triangle of a $T \times T$ matrix $E$, where the element in the $i$th row and $j$th column of $E$ contains $p(Y_{ij})$, the likelihood of the DLM fit on the observations $(Y_i, \ldots, Y_j)$.

Fitting the full series allows us to fill in the full first row of $E$. Then, fitting all but the first observation, $(Y_2, \ldots, Y_T)$, provides the second row. Next, we leave out the first two observations, fitting $(Y_3, \ldots, Y_T)$, which provides the third row of $E$ (excluding the first and second columns). We continue this process all the way to time $T$, where fitting the DLM on just the observation $Y_T$ gives the final, bottom-right element of $E$, $p(Y_{T,T})$.

The partition likelihood is given by

$$p(Y | \rho) = \prod_{ij \in \rho} p(Y_{ij})$$
Given the precomputed segment likelihoods, we may obtain the MLE for $\rho$ by maximizing this product over all possible partitions. A brute force approach, testing every combination one by one, is impossible, requiring computation that scales combinatorially in $T$. We introduce here an algorithm that shares some similarity with Viterbi’s algorithm that computes the MLE for $\rho$ in $O(T^2)$ time. The idea of the algorithm is to work backwards from the end of the series at $t = T$, to progressively find the maximum likelihood path from $t$ to $T$, which we will call $P(t)$. We will call $P(t)$’s associated likelihood $M(t)$. The main idea, then, is that

$$M(t) = \max_{s=t,t+1,\ldots,T} \{ p(Y_{t,s}) \times M(s + 1) \}$$

where at the edge case, we define $M(T + 1) = 1$.

That is, to find the optimum path from $t$ to $T$, all we need to consider are the paths we can take from $t$ to $s$, $s \geq t$, combined respectively with the optimum path from $s + 1$ to $T$. This requires on the order of $T - t + 1$ operations per step, with $T$ total steps, amounting to $\frac{T(T+1)}{2} = O(T^2)$ operations.

While the MLE for $\rho$ could be useful in some circumstances, the true power and utility of the PPM comes from incorporating prior information to regularize the frequency of changepoints. Therefore, what we really seek is the maximum a posteriori (MAP) estimate for $\rho$.

The posterior for $\rho$, marginal over $\theta$, is given by
\[ p(\rho | Y) = \frac{p(Y | \rho)p(\rho)}{p(Y)} \]

\[ \propto p(Y | \rho)p(\rho) \]

\[ = \prod_{ij \in \rho} \{p(Y_{ij})c_{ij}\} \]

The prior cohesions \( c_{ij} \) determine how costly each changepoint is to the model. We follow the prior specification of Barry and Hartigan (1993) in effect, though the notation is slightly different. We define

\[ c_{ij} = p^{i(i>1)}(1-p)^{j-i} \]

with \( p \) random, between 0 and 1. Further, we follow the prior specification for \( p \) used by Loschi et al. (2003) - \( p \sim \text{Beta}(\alpha, \beta) \). This requires conditioning on \( p \), so that we have

\[ p(\rho | Y, p) = \frac{1}{p} \prod_{ij \in \rho} \left\{ p(Y_{ij}) \times (1-p)^{j-i}p \right\} \]

With this setup, we can in fact follow the very same algorithm as for the MLE, this time weighting each segment likelihood by the cost induced by the prior. The recursive update is then
\[ M(t) = \max_{s=t,t+1,\ldots,T} \{ p(Y_{t,s}) \times M(s+1) \times c_{t,s} \} \]

\[ M(t) = \max_{s=t,t+1,\ldots,T} \{ p(Y_{t,s}) \times M(s+1) \times p^{1(s>1)} (1-p)^s \} \]

Following this algorithm to its conclusion at \( t = 1 \), we obtain the posterior maximum for \( \rho \), conditional on \( p \). To obtain the unconditional MAP for \( \rho \), we can apply a simple expectation maximization procedure. We will iteratively maximize the expected log joint posterior:

\[ Q^{(s)}(\rho) = \mathbb{E}_{p|Y,\rho^{(s)}} [\log p(p, \rho | Y)] \]

Starting from the joint posterior over \( p \) and \( \rho \), we have

\[ p(p, \rho | Y) \propto p(Y | \rho, p) p(\rho, p) \]

\[ = p(Y | \rho) p(\rho | p) p(p) \]

\[ = \prod_{r=1}^{b} \{ f(Y_{i_r,i_{r+1}}) c_{i_r,i_{r+1}} 1 \} p(p) \]

\[ \propto \frac{1}{p} \prod_{r=1}^{b} \left\{ f(Y_{i_r,i_{r+1}}) p(1-p)^{(i_{r+1} - i_r)} \right\} p^{\alpha-1} (1-p)^{\beta-1} \]

where \( b \) is the number of segments in the partition \( \rho \). As such, \( i_r \) is the location of the start of the \( r \)th segment, with \( i_1 = 1 \) and \( i_{b+1} = T + 1 \). Taking the log, we arrive at
\[
\log p (\rho, p \mid Y) = \sum_{r=1}^{b} \left\{ \log f(Y_{i_r, i_{r+1}} 1) + \log p + (i_{r+1} - 1 - i_r) \log (1 - p) \right\} \\
+ (\alpha - 1) \log p + (\beta - 1) \log (1 - p) - \log p + L
\]

Only the terms within the sum are affected by the partition, so we may lump the remaining terms into the constant \(L\) for our purpose of maximizing \(Q(\rho)\). Therefore, we have

\[
Q^{(s)}(\rho) = \mathbb{E}_{p \mid Y, \rho^{(s)}} \left[ \sum_{r=1}^{b} \log f(Y_{i_r, i_{r+1}} 1) + \log p + (i_{r+1} - 1 - i_r) \log (1 - p) \right] \\
+ \mathbb{E}_{p \mid Y, \rho^{(s)}} [L]
\]

\[
= \sum_{r=1}^{b} \{ \log f(Y_{i_r, i_{r+1}} 1) + \mathbb{E} [\log p] + (i_{r+1} - 1 - i_r) \mathbb{E} [\log (1 - p)] \}
\]

Note that we suppress the subscripts of the expectations for brevity only. This brings us to our M step:

\[
\rho^{(s+1)} = \underset{\rho}{\operatorname{argmax}} \ Q^{(s)}(\rho)
\]

\[
= \underset{\rho}{\operatorname{argmax}} \sum_{r=1}^{b} \log f(Y_{i_r, i_{r+1}} 1) + \mathbb{E} [\log p] + (i_{r+1} - 1 - i_r) \mathbb{E} [\log (1 - p)] \quad (4.1)
\]

To maximize this function, we'll need to compute two quantities in our E step:
\( \mathbb{E}_{p|Y,\rho}(\log p) \) and \( \mathbb{E}_{p|Y,\rho}(\log (1 - p)) \). With our chosen prior for \( p, p \sim B(\alpha, \beta) \), the conditional distribution \( (p \mid Y, \rho) \) is also Beta-distributed:

\[
p(p \mid Y, \rho) = p(p \mid \rho)
\]

\[
\propto p(\rho \mid p)p(p)
\]

\[
\propto \frac{1}{p} \prod_{r=1}^{b} \left\{ p(1 - p)^{i_r + 1} \right\} p^\alpha (1 - p)^\beta
\]

\[
= p^b 1 + \alpha (1 - p)^T (b + \beta)
\]

\[
\sim B(\alpha + b, \beta)\]

The required expectations are then well defined,

\[
\mathbb{E}_{p|Y,\rho}(\log p) = \psi(\alpha + b(s) - 1) - \psi(\alpha + \beta + T - 1)
\]

\[
\mathbb{E}_{p|Y,\rho}(\log (1 - p)) = \psi(\beta + T - b(s)) - \psi(\alpha + \beta + T - 1)
\]

where \( \psi \) is the digamma function.

Algorithms 10, 11, and 12 put this method into practice.

**Algorithm 10 PPM MAP E-Step**

1: function PPM ESTEP(\( \alpha, \beta, T, b \))

2: \( \text{elogp} \leftarrow \psi(\alpha + b - 1) - \psi(\alpha + \beta + T - 1) \)

3: \( \text{elogimp} \leftarrow \psi(\beta + T - b) - \psi(\alpha + \beta + T - 1) \)

4: return elogp, elogimp

5: end function

This approach to compute the MAP for the direct implementation of the PPM DLM is useful in some cases, and often benefits from being fully deterministic. However, this still requires total computation that scales quadratically in \( T \). In addition,
Algorithm 11 PPM MAP M-Step

1: function ppm mstep(E, elogp, elog1mp)
2: P[T + 1] ← []
3: log M[T + 1] ← 0
4: for i = T − 1, . . . , 1 do
5: \[ g = \{ \log E[i, j] + \log M[j + 1] + elogp + (j - i) * elog1mp, j \geq i \} \]
6: \[ \log M[i] \leftarrow \max_j(g) \]
7: \[ j_m \leftarrow \text{argmax}_j(g) \]
8: \[ P[i] \leftarrow [j_m, P[j_m + 1]] \]
9: end for
10: return P[1]
11: end function

Algorithm 12 PPM MAP EM

1: function ppm em(E, \alpha, \beta)
2: T ← nrow(E)
3: elogp, elog1mp ← ppm estep(\alpha, \beta, 1, 1) \quad \triangleright \text{Initialize}
4: b ← 0
5: b prev ← −1
6: while b \neq b prev do
7: \quad cps ← ppm mstep(E, elogp, elog1mp) \quad \triangleright \text{changepoint locations}
8: \quad b prev ← b
9: \quad b ← \text{length(cps)}+1
10: \quad elogp, elog1mp ← ppm estep(\alpha, \beta, T, b)
11: end while
12: return cps
13: end function
we cannot use the outlier-robust model, since this eliminates the possibility of removing redundant model fits, and would require sampling or performing EM for every possible \((i, j)\) segment, requiring computation on the order of \(sT^2\), where \(s\) is either the number of samples or iterations of EM. Importantly, this would even require recomputing all segment likelihoods for each iteration (which is typically considerably more computationally expensive than finding the MAP, even though both operations are \(O(T^2)\)).

Since this form of the algorithm is unable to provide robustness to outliers, we propose next an alternative approach.

### 4.2.2 The Robust PPM Dynamic Linear Model

To sidestep this issue, we change our concept of what a changepoint means to the model. Instead of making the segments fully independent, we simply allow the information carryover between segments to decay to arbitrarily small. To implement this, we go back to the robust, soft-changepoint model. By making the evolution error scale \(\alpha_t\) dependent on whether \(t\) is a segment boundary, we can allow changepoints much more freedom to “disregard” information about \(\theta_t\) coming from observations prior to time \(t\). Practically, we introduce a changepoint deflation on our hyperprior \(\gamma\):

\[
(\alpha_t \mid \rho) \sim \text{Gamma}\left(\frac{\gamma}{2K_t}, \frac{\gamma}{2K_t}\right)
\]

where,

\[
K_t = \begin{cases} 
1 & u_t = 1 \quad (t \text{ is not a changepoint}) \\
K & u_t = 0 \quad (t \text{ is a changepoint}) 
\end{cases}
\]
Figure 4.1: The state space diagram for the PPM-coupled Dynamic Linear Model.

Figure 4.1 shows the structure of the model. One advantage of formulating the model this way is that there is a layer of separation between the partition $\rho$ and the observations $Y$. By squeezing the layer of $\alpha_i$'s in between, we simplify the conditionals considerably, and are able to completely avoid computing the data likelihood for every possible segment.

In addition, while we adjust the meaning of changepoint, we retain the rest of the useful structure that the product partition model provides. This structure allows us to specify prior knowledge about the location and frequency of changepoints.

Adding the PPM layer to the DLM brings greater flexibility and explanatory power to the model. However, with this great flexibility of course comes great responsibility. The tradeoffs between the different behavioral modes of the model become more
Figure 4.2: A PPM-coupled DLM fit to the representative pixel. The bottom figure shows the probability of a changepoint at each location. Solid bars show the probability at that point, while the dotted line gives the 360-day rolling sum - the expected number of changepoints within a ~1 year window.

complex, and we must be careful in specifying our priors to ensure the model responds desirably to different patterns of data.

In Figure 4.2, we show the PPM-coupled changepoint DLM fit to our representative pixel. Again we see that the model gracefully handles gradual changes, outliers, and abrupt changes - and now also permits probabilistic inference on the location of these abrupt changes.
4.3 Case Study

4.3.1 Data Preparation

In order to assess the performance of our model in comparison to CCDC, we make use of a publicly-available dataset (Zhu et al., 2020) of Landsat time series with ground-truth annotations of changepoint locations.

For ease of computation, we filter the set of pixels to only include the time series where all gaps between observations are an integer multiple of 8. Some time series within the dataset have slightly offset observation dates, and the gaps between Landsat 5 and Landsat 7 observations oscillate between 7 and 9 days. We remove these from consideration as they would require on the order of 64 times as much computation. We detect these time series by finding the greatest common denominator between the finite differences between observations, and discard any time series that has a greatest common denominator less than 8.

Further, we subset to only those time series that have annotations. Every annotated time series has at least two annotated dates - the beginning and the end of the observed annotation period. Annotated dates between these signify human-identified changes in landcover occurring at these dates.

We also only include three specific types of annotated changepoints: harvests, mechanical disturbances, and fires. We limit our analysis to these annotation types because they most clearly correspond to large, sudden changes in pattern. After these filtering steps, we are left with a total of 2210 annotated pixel time series.

4.3.2 Evaluation

To evaluate the DLM-PPM methods, we will assess the performance on two separate grounds: changepoint detection, and overall fit to the observations themselves. As a
### Table 4.1: Model Fit Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Median Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCDC</td>
<td>0.0193 (0.0189, 0.0197)</td>
</tr>
<tr>
<td>Direct DLM-PPM</td>
<td>0.0168 (0.0164, 0.0172)</td>
</tr>
<tr>
<td>Robust DLM-PPM</td>
<td>0.0133 (0.0130, 0.0136)</td>
</tr>
</tbody>
</table>

benchmark comparison, we will measure the performance of our DLM-PPM models against that of the CCDC model.

**Fit comparison**

To assess the overall fit to the data, we use the median absolute error as a measure of fit to each individual time series. Then we take the average of these median absolute errors across the full set of time series to produce the fit performance metric.

We see from Table 4.1 that the Direct DLM-PPM strongly outperforms the CCDC model, and further that the Robust DLM-PPM performs even better. In the table we show the mean metric next to the 90% confidence interval produced via a non-parametric bootstrap over the set of median absolute errors.

**Changepoint comparison**

Evaluating the performance of a changepoint detection algorithm can be a challenging task. To evaluate the changepoint detection, we follow the recommendations of Burg and Williams (2020), and measure performance with two metrics that stem from separate approaches.

The first approach draws motivation from the idea of a partition being a clustering of the observations, where each contiguous segment of the partition is seen as a cluster. Then, we may measure the fidelity of the given partition $\rho'$ to the ground-truth partition $\rho$ with the metric...
\[
C (\rho', \rho) = \frac{1}{T} \sum_{S \in \rho} |S| \max_{S' \in \rho'} J (S, S')
\]

where \(J (S, S')\) represents the Jaccard Index (or Intersection over Union) of the segment \(S\) from the ground truth partition and the segment \(S'\) from the candidate partition, defined as

\[
J (S, S') = \frac{|S \cap S'|}{|S \cup S'|}
\]

Burg and Williams (2020) call \(C (\rho', \rho)\) the covering metric of the ground-truth partition \(\rho\) by the partition \(\rho'\).

The second metric Burg and Williams (2020) recommend is the F1 score, motivated from a classification perspective. This approach treats each time point in the series as a binary classification problem: changepoint or no changepoint. The F1 score then weighs together the precision and recall of the candidate partition’s set of changepoints when compared to the ground-truth partition.

Because there is ambiguity in the annotation data around when a changepoint truly occurred, we use a margin of error of 23 8-day timesteps. This means that if a candidate partition detects a changepoint within the 368-day window around a ground-truth changepoint, it will be considered a true positive.

For the direct implementation PPM-DLM, the estimated partition is straightforward - we simply use the MAP estimate. For the Robust PPM-DLM, we must choose a mapping from our posterior samples to a single partition that we may compare to the ground truth. For simplicity, we simply threshold at 90%: if 90% of the sample partitions detect a changepoint at time \(t\), we include \(t\) in the estimated partition for that pixel.

Table 4.2 shows the changepoint detection performance of each of our models over
Table 4.2: Changepoint Detection Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Covering Metric</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCDC</td>
<td>0.905 (0.899, 0.911)</td>
<td>0.886 (0.879, 0.894)</td>
</tr>
<tr>
<td>Direct DLM-PPM</td>
<td>0.902 (0.896, 0.908)</td>
<td>0.876 (0.868, 0.883)</td>
</tr>
<tr>
<td>Robust DLM-PPM</td>
<td>0.857 (0.850, 0.864)</td>
<td>0.839 (0.831, 0.847)</td>
</tr>
</tbody>
</table>

the dataset. Again we show the mean metrics next to the 90% confidence intervals produced via a nonparametric bootstrap over the set of per-time-series metrics. We see that, by the numbers, the CCDC model marginally outperforms the Direct DLM-PPM, and the Robust DLM-PPM lags slightly behind both.

4.3.3 Example Fits

To give context and a better look into the performance of the different models, we show here a collection of representative time series, with the model fits overlaid.

In Figure 4-3, we show a few pixels where all three models perform well. These pixels have very clear changepoint locations, and as a result the three models are all able to identify the change.

In Figure 4-4, we show a sampling of pixels where the fits for the DLM-PPM models are suboptimal. For the top two series, the Direct DLM-PPM mistakes an outlier for a single-observation changepoint. This happens as a result of the lack of outlier robustness. In the bottom three series, the Robust DLM-PPM overcommits to changes in the series. An argument could be made that there is legitimate underlying signal behind these changes, and that the model is simply adapting to this true change in pattern. However, in many downstream applications a more smoothed fit would be more useful.

Figure 4-5 shows a representative group of series where both the CCDC and even the hand-annotated ground-truth miss clear changes in pattern. These pixels point to a challenge with the dataset. The threshold for classifying a change in pattern as
Figure 4.3: A selection of representative time series of Shortwave Infrared Reflectance, shown with the fits of the CCDC, Direct DLM-PPM, and Robust DLM-PPM models. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color (or the hand-annotated ground truth).
Figure 4.4: A selection of pixel timeseries where the DLM-PPM models underperform. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color.
a changepoint or not can be ambiguous, even to a human annotator. As a result, the changepoint detection performance is fully dependent on the judgement calls made by human annotators, as well as possible mistakes by the annotators themselves.

For example, a pixel that undergoes a change may have a period between the onset of the change and return to stability where there is a clear pattern. In this case, there could arguably be two changepoints, one at the onset, and one at the end of the change. However, the annotator may simply mark a changepoint in the middle of the change period. This results in a reported misclassification by the model, even though it better represents the pattern of the series.

Overall, the DLM-PPM models tend to capture signal significantly better than CCDC. This can occasionally come at the expense of overadaptation. However, in many cases what may look like overadaptation can in fact be the proper representation of a rapidly changing signal.
Figure 4.5: A sample of pixel timeseries where the DLM-PPM models outperform CCDC at detecting change. The vertical dashed lines represent detected changepoint locations for the model of the corresponding color.
Chapter 5

Conclusion

5.1 Overview of contributions

We start with a simple, computationally efficient implementation of the DLM for modeling Landsat reflectance time series at huge scale.

Next we introduced the Robust Soft-Changepoint DLM, a more flexible model capable of downweighting the effect of outlying observations, while also being to adapt rapidly to sudden, large changes in the pattern of observations. Further, we characterized the tradeoff that accompanies this added flexibility, and outlined a procedure for eliciting an appropriate set of prior distributions to match example patterns of observations with the corresponding desired model behavior.

Lastly, we proposed a major novel contribution to the Bayesian changepoint literature: a model that gracefully combines the formal changepoint detection framework of the Product Partition Model with the flexibility of the Dynamic Linear Model within segments. We develop two models with slightly different approaches. The first directly implements the PPM on top of the DLM, and permits deterministic computation of the MAP estimate for the optimal segmentation of the series in $O(T^2)$ time (with $T$ the number of observations in the series). The second approach layers the PPM on top of the Robust Soft-Changepoint DLM, which provides robustness to outliers.
5.2 Recommendations for future work

- Make the direct DLM-PPM more robust to outliers
- Further characterize the prior tradeoffs in the DLM-PPM
- Improve mixing in the Gibbs sampling for the Robust DLM-PPM
Appendix A

Appendix A

A.1 Robust DLM EM

\[
p (\theta_0, \ldots, \theta_T, q_1, \ldots, q_T, \phi \mid Y_1, \ldots, Y_T) \\
\propto p (\theta_0) p (\phi) \prod_{t=1}^{T} p (Y_t \mid \theta_t, q_t, \phi) p (q_t) p (\theta_t \mid \theta_{t-1})
\]

with,

\[
\begin{align*}
\theta_0 & \sim \mathcal{N} (m_0, C_0), \\
\phi & \sim \mathcal{G} \left( \frac{m_0}{2}, \frac{n_0 S_0}{2} \right), \\
Y_t \mid \theta_t, q_t, \phi & \sim \mathcal{N} \left( F' \theta_t, \frac{1}{q_t \phi} \right), \\
q_t & \sim \mathcal{G} \left( \frac{\tau_2}{2}, \frac{\tau_2}{2} \right), \\
\theta_t \mid \theta_{t-1} & \sim \mathcal{N} (G \theta_{t-1}, W_t)
\end{align*}
\]

thus,
\begin{align*}
p (\theta_0, \ldots, \theta_T, q_1, \ldots, q_T, \phi \mid Y_1, \ldots, Y_T)
\propto & \frac{1}{\sqrt{(2\pi)^p |C_0|}} \exp \left( -\frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \right) \phi \left( \frac{n_0}{2} \right) \exp \left( -\frac{n_0 S_0}{2} \phi \right) \\
& \times \prod_{t=1}^{T} \left\{ \left( \frac{q_t \phi}{2\pi} \right)^{\frac{1}{2}} \exp \left( \frac{-q_t \phi}{2} (Y_t - F' \theta_t)^2 \right) q_t \left( \frac{\tau}{2} \right)^{\frac{1}{2}} \exp \left( -\frac{\tau}{2} q_t \right) \\
& \quad \times \frac{1}{\sqrt{(2\pi)^{p|W_t|}}} \exp \left( -\frac{1}{2} (\theta_t - G \theta_t)' W_t^{-1} (\theta_t - G \theta_t) \right) \right\} (A.1)
\end{align*}

\begin{align*}
= & -\frac{p}{2} \log 2\pi - \frac{1}{2} \log |C_0| - \frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \\
& + \left( \frac{n_0}{2} - 1 \right) \log \phi - \frac{n_0 S_0}{2} \phi \\
& + \sum_{t=1}^{T} \left\{ \frac{1}{2} \log (q_t \phi) - \frac{1}{2} \log 2\pi - \frac{1}{2} q_t \phi (Y_t - F' \theta_t)^2 \\
& \quad + \left( \frac{\tau}{2} - 1 \right) \log q_t - \frac{\tau}{2} q_t \\
& \quad - \frac{p}{2} \log 2\pi - \frac{1}{2} \log |W_t| - \frac{1}{2} (\theta_t - G \theta_t)' W_t^{-1} (\theta_t - G \theta_t) \right\} (A.2)
\end{align*}

\[Q^{(s)} (\theta, \phi) = \mathbb{E}_{q|Y, \theta^{(s)}, \phi^{(s)}} [\log p (\theta, \phi, q \mid Y)] \]

\begin{align*}
= & - (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) + \left( \frac{n_0}{2} - 1 \right) \log \phi - \frac{n_0 S_0}{2} \phi \\
& + \sum_{t=1}^{T} \left\{ \frac{1}{2} \log (\phi) - \mathbb{E}_{q|Y, \theta^{(s)}, \phi^{(s)}} [q_t] \frac{1}{2} \phi (Y_t - F' \theta_t)^2 \\
& \quad - \frac{1}{2} (\theta_t - G \theta_t)' W_t^{-1} (\theta_t - G \theta_t) \right\}
\end{align*}

+ L

with L a constant term, not dependent on \( \theta \) or \( \phi \).
\[
p(q_t \mid Y, \theta^{(s)}, \phi^{(s)})
\]

\[
= p(q_t \mid Y_t, \theta_t^{(s)}, \phi^{(s)})
\]

\[
\propto p(Y_t \mid q_t, \theta_t^{(s)}, \phi^{(s)}) p(q_t \mid \theta_t^{(s)}, \phi^{(s)})
\]

\[
= p(Y_t \mid q_t, \theta_t^{(s)}, \phi^{(s)}) p(q_t)
\]

\[
\propto \sqrt{q_t} \exp \left( -\frac{1}{2} \phi^{(s)} \left( Y_t - F'\theta_t^{(s)} \right)^2 \right) \times q_t^{\tau} \exp \left( -\frac{\tau}{2} q_t \right)
\]

\[
= q_t^{\frac{\tau + 1}{2}} \exp \left( -\frac{1}{2} \left( \phi^{(s)} \left( Y_t - F'\theta_t^{(s)} \right)^2 + \tau \right) q_t \right)
\]

Thus,

\[
q_t \mid Y_t, \theta^{(s)}, \phi^{(s)} \sim \mathcal{G} \left( \frac{\tau + 1}{2}, \frac{\tau + \phi^{(s)} \left( Y_t - F'\theta_t^{(s)} \right)^2}{2} \right)
\]

and therefore,

\[
\mathbb{E}_{q_t \mid Y, \theta^{(s)}, \phi^{(s)}}[q_t] = \frac{\tau + 1}{\tau + \phi^{(s)} \left( Y_t - F'\theta_t^{(s)} \right)^2}
\]

(A.3)

A.2 RSC DLM EM

\[
p(\theta_0, \ldots, \theta_T, q_1, \ldots, q_T, \alpha_1, \ldots, \alpha_T, \phi \mid Y_1, \ldots, Y_T)
\]

\[
\propto p(\theta_0) p(\phi) \prod_{t=1}^{T} p(Y_t \mid \theta_t, q_t, \phi) p(q_t) p(\theta_t \mid \theta_{t-1}, \alpha_t) p(\alpha_t)
\]

with,
\[
\phi \sim \mathcal{G} \left( \frac{n_0}{2}, \frac{n_0 S_0}{2} \right), \\
Y_t \mid \theta_t, q_t, \phi \sim \mathcal{N} \left( F' \theta_t, \frac{1}{q_t \phi} \right), \\
q_t \sim \mathcal{G} \left( \frac{\tau}{2}, \frac{1}{2} \right), \\
\theta_t \mid \theta_{t-1}, \alpha_t \sim \mathcal{N} \left( G \theta_{t-1}, \frac{1}{\alpha_t} W_t \right), \\
\alpha_t \sim \mathcal{G} \left( \frac{\gamma}{2}, \frac{1}{2} \right)
\]

Then, taking the log, we have

\[
p (\theta_0, \ldots, \theta_T, q_1, \ldots, q_T, \alpha_1, \ldots, \alpha_T, \phi \mid Y_1, \ldots, Y_T) \\
\propto \frac{1}{\sqrt{(2\pi)^p |C_0|}} \exp \left( -\frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \right) \phi^{\left( \frac{n_0}{2} - 1 \right)} \exp \left( -\frac{n_0 S_0}{2} \phi \right) \\
\times \prod_{t=1}^{T} \left\{ \left( \frac{q_t \phi}{2\pi} \right)^{\frac{1}{2}} \exp \left( -\frac{q_t \phi}{2} (Y_t - F' \theta_t)^2 \right) q_t^{\left( \frac{1}{2} - 1 \right)} \exp \left( -\frac{1}{2} q_t \right) \right\}^{\frac{1}{\alpha_t} W_t} \left( \frac{1}{\alpha_t} W_t \right)^{\frac{1}{2}} (\theta_t - G \theta_{t-1}) \\
\times \exp \left( -\frac{1}{2} (\theta_t - G \theta_{t-1})' \left( \frac{1}{\alpha_t} W_t \right)^{\frac{1}{2}} (\theta_t - G \theta_{t-1}) \right)
\]

(A.4)

\[
\begin{align*}
&- \frac{p}{2} \log 2\pi - \frac{1}{2} \log |C_0| - \frac{1}{2} (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) \\
&+ \left( \frac{n_0}{2} - 1 \right) \log \phi - \frac{n_0 S_0}{2} \phi \\
&+ \sum_{t=1}^{T} \left\{ \frac{1}{2} \log (q_t) + \frac{1}{2} \log (\phi) - \frac{1}{2} \log 2\pi - \frac{1}{2} q_t \phi (Y_t - F' \theta_t)^2 \\
&\quad + \left( \frac{\tau}{2} - 1 \right) \log q_t - \frac{\tau}{2} q_t + \left( \frac{\gamma}{2} - 1 \right) \log \alpha_t - \frac{\gamma}{2} \alpha_t \\
&\quad - \frac{p}{2} \log 2\pi - \frac{1}{2} \log \left| \frac{1}{\alpha_t} W_t \right| - \frac{1}{2} (\theta_t - G \theta_{t-1})' \left( \frac{1}{\alpha_t} W_t \right)^{\frac{1}{2}} (\theta_t - G \theta_{t-1}) \right\}
\end{align*}
\]
Thus, the expectation we seek to maximize is:

\[
Q^{(s)}(\theta, \phi) = \mathbb{E}_{q, \alpha | Y, \theta^{(s)}, \phi^{(s)}} \left[ \log p(\theta, \phi, q | Y) \right]
\]

\[
= - (\theta_0 - m_0)' C_0^{-1} (\theta_0 - m_0) + \left( \frac{n_0}{2} - 1 \right) \log \phi - \frac{n_0 S_0}{2} \phi
\]

\[
+ \sum_{t=1}^{T} \left\{ \frac{1}{2} \log (\phi) - \mathbb{E}_{q, \alpha | Y, \theta^{(s)}, \phi^{(s)}} [q_t] \right\} \frac{1}{2} \phi (Y_t - F' \theta_t)^2
\]

\[
- \mathbb{E}_{q, \alpha | Y, \theta^{(s)}, \phi^{(s)}} [\alpha_t] \frac{1}{2} (\theta_t - G \theta_{t-1})' W_t^{-1} (\theta_t - G \theta_{t-1}) \right\}
\]

\[
+ L
\]

with \( L \) a constant term, not dependent on \( \theta \) or \( \phi \).

Now,

\[
\mathbb{E}_{q, \alpha | Y, \theta^{(s)}, \phi^{(s)}} [q_t] = \mathbb{E}_{q | Y, \theta^{(s)}, \phi^{(s)}} [q_t]
\]

and therefore the E-step for \( q \) matches that of (A.3).

For \( \alpha \),
\[ p(\alpha_t \mid Y, \theta(s), \phi(s)) \]
\[ = p(\alpha_t \mid \theta_t(s), \theta_{t-1}) \]
\[ \propto p(\theta_t(s) \mid \theta_{t-1}(s), \alpha_t) p(\alpha_t \mid \theta_{t-1}(s)) \]
\[ = p(\theta_t(s) \mid \theta_{t-1}(s), \alpha_t) p(\alpha_t) \]
\[ \propto \frac{1}{\sqrt{(2\pi)^p |\alpha_t W_t|}} \exp \left( -\frac{1}{2} \left( \theta_t(s) - G\theta_{t-1}(s) \right)' \left( \frac{1}{\alpha_t} W_t \right)^{-1} \left( \theta_t(s) - G\theta_{t-1}(s) \right) \right) \times \alpha_t^{\frac{p+1}{2}} \exp \left( -\frac{\gamma}{2} \alpha_t \right) \]
\[ \propto \alpha_t^{\frac{p+1}{2}} \exp \left( -\frac{\gamma}{2} \alpha_t \right) \times \alpha_t \exp \left( -\frac{1}{2} \alpha_t \left( \theta_t(s) - G\theta_{t-1}(s) \right)' W_t^{-1} \left( \theta_t(s) - G\theta_{t-1}(s) \right) \right) \]
\[ = \alpha_t^{\frac{p+1}{2}} \exp \left( -\frac{1}{2} \alpha_t \left( \gamma + \left( \theta_t(s) - G\theta_{t-1}(s) \right)' W_t^{-1} \left( \theta_t(s) - G\theta_{t-1}(s) \right) \right) \right) \]

Thus,

\[ \alpha_t \mid Y_t, \theta(s), \phi(s) \sim \mathcal{G} \left( \frac{\gamma + p}{2}, \frac{\gamma + \left( \theta_t(s) - G\theta_{t-1}(s) \right)' W_t^{-1} \left( \theta_t(s) - G\theta_{t-1}(s) \right)}{2} \right) \]

and therefore,

\[ \mathbb{E}_{q,\alpha \mid Y, \theta(s), \phi(s)} [\alpha_t] = \frac{\gamma + p}{\gamma + \left( \theta_t(s) - G\theta_{t-1}(s) \right)' W_t^{-1} \left( \theta_t(s) - G\theta_{t-1}(s) \right)} \]

(A.5)
Bibliography


