A Gaussian Process Based Online Change Detection Algorithm for Monitoring Periodic Time Series*

Varun Chandola† Ranga Raju Vatsavai‡

Abstract

Online time series change detection is a critical component of many monitoring systems, such as space and air-borne remote sensing instruments, cardiac monitors, and network traffic profilers, which continuously analyze observations recorded by sensors. Data collected by such sensors typically has a periodic component. Most existing time series change detection methods are not directly applicable to handle such data, either because they are not designed to handle periodic time series or because they cannot operate in an online mode. We propose an online change detection algorithm which can handle periodic time series. The algorithm uses a *Gaussian process* based non-parametric time series prediction model and monitors the difference between the predictions and actual observations within a statistical control chart framework to identify changes. A key challenge in using Gaussian process in an online mode is the need to solve a large system of equations involving the associated covariance matrix which grows with every time step. The proposed algorithm exploits the special structure of the covariance matrix and can analyze a time series of length T in $O(T^2)$ time while maintaining a $O(T)$ memory footprint, compared to $O(T^4)$ time and $O(T^2)$ memory requirement of standard matrix manipulation methods. We experimentally demonstrate the superiority of the proposed algorithm over several existing time series change detection algorithms on a set of synthetic and real time series. Finally, we illustrate the effectiveness of the proposed algorithm for identifying land use land cover changes using *Normalized Difference Vegetation Index* (NDVI) data collected for an agricultural region in Iowa state, USA. Our algorithm is able to detect different types of changes in a NDVI validation data set (with $\approx 80\%$ accuracy) which occur due to crop type changes as well as natural disasters.

*Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725. This research is funded through the LDRD program at ORNL.

†chandolav@ornl.gov Oak Ridge National Laboratory

‡vatsavairr@ornl.gov Oak Ridge National Laboratory

1 Introduction

Online monitoring of periodic time series data for identifying *changes* is a highly relevant problem in many application domains such as biomass monitoring for *land use land cover change detection*, electrocardiogram (ECG) analysis, econometrics, network monitoring, and in other domains where the collected time series data has a periodic (seasonal) component. Accurate and timely detection of changes in the time series can result in identification of significant, and often critical, events, e.g., abnormal conditions in a patient's health [26], network traffic events [17], forest fires [4], etc.

Biomass monitoring[1] for identification of *land use land cover changes* is one such crucial application domain where research in the field of online time series change detection has staggering future implications for multiple reasons. *Forest biomass* monitoring can be used for identification of changes in forest cover due to forest fires [4], logging, natural disasters, etc. Monitoring of *crop biomass* is highly significant as well, as it is directly related to food demands across the world. With recent government emphasis on biofuel development the landscape of United States as well as other countries is going to change dramatically in coming years. There are several preliminary reports [24, 30] that highlight both economic and environmental impacts of growing energy crops, and its impact on climate change. In order to understand the changing landscape and complex interactions between biomass and climate variables on a continuous basis, scalable change detection algorithms are needed to continuously monitor biomass and identify anomalous changes.

Recently, many researchers have proposed methods for detecting changes in biomass [4, 9, 19], using remote sensing data collected from instruments aboard satellites, such as the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument aboard NASA's Terra satellite [15]. These existing methods analyze temporal profiles of one of the several available vegetation indices from MODIS for different geographical locations, and

[1] Biomass is broadly defined as the mass of the living biological organisms in a unit area.

identify changes in them. Figure 1 shows an example of *Normalized Difference Vegetation Index* (NDVI) time series for a particular location in California, USA. In this paper, we demonstrate the applicability of the proposed change detection algorithm to identify changes in NDVI time series data.

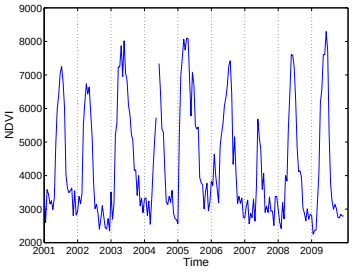


Figure 1: MODIS NDVI Time Series for 2001-2009 for a Southern California location with a known forest fire (*Canyon fire*) in 2007 [*src:* `http://cdfdata.fire.ca.gov/incidents/incidents_archived?archive_year=2007`]. Data has a missing value in 2004 (unreliable MODIS flag).

Most of the existing research in the general area of time series change detection [3] as well as the methods specifically proposed for biomass change detection [4, 9, 19], do not satisfy one or both of the two critical requirements of our problem, i.e., *operating in an online mode* and *handling periodic data*. Methods proposed for online change detection, typically assume that the periodicity (or seasonality) in the time series is removed prior to the analysis. The periodicity is removed using statistical seasonality adjustment methods [7] or by analyzing seasonal aggregates/features [19, 27]. For analyzing NDVI data the latter approach is inapplicable since it would result in coarse annual monitoring of the data. Using statistical methods for adjusting seasonality has limitations due to multiple reasons. First, most of the seasonality adjustment methods operate in a batch mode. Second, removing the seasonality might result in loss of information required to identify changes. For example, in the case of crop biomass monitoring, the change in the NDVI time series might occur because of the change in the seasonality; a farmer alternating between soybean and corn might start growing corn every year for economic reasons. Third, the seasonal behavior of the time series might vary across successive periods due to normal factors. Again, for crop biomass monitoring, the annual NDVI time series for a crop (also known as the *phenology*) might vary across years depending on annual climate, quality of seeds, fertilizers, etc.

Some of the challenges associated with online change detection in periodic time series data, are:

- **Presence of noise, outliers, and missing data** (See Figure 1). These issues are common when dealing with observational data and can adversely affect the data analysis. Missing data is a key issue when dealing with remote sensing observations due to presence of cloud cover.

- **Issues with periodicity of data**. Periodic behavior may vary across successive cycles. For example, in the biomass monitoring domain, the NDVI profile collected from a single location across two years might change due to change in crop phenology. True periodicity might not coincide with the natural cycle length. Often, farmers alternate between two different crops every year, in which the case the corresponding NDVI data for such farms would have a periodicity of two years. Automatic identification of the true periodicity of the time series is a challenging task.

- **Different types of changes**. In a strict statistical sense, a change is defined when the parameter(s) of the generative distribution for the time series changes significantly [3]. When dealing with periodic time series, this general definition can refer to changes in the trend component, seasonal component, or the noise (residual) component. In Section 3 we define four types of changes encountered in the biomass monitoring domain.

- **Scalability**. Data collected by remote sensing satellites and other sensors, such as ECG monitors, is ever increasing. Online change detection methods need to be designed in a computationally and memory efficient manner so that they can scale to massive data sizes.

We propose a non-parametric statistical algorithm which addresses the above challenges to identify changes in periodic time series in an online fashion. We use *Gaussian Process* [23] as the basis for a Bayesian non-parametric predictive model for time series data and use the difference between the predicted and observed values to monitor change in an online fashion. Gaussian process[2] has been traditionally applied to non-periodic time series prediction [10], though it can be applied to handle periodic time series with an appropriate covariance function [5]. To the best of our knowledge this is the first time Gaussian process have been applied for time series change detection. To apply a GP based time series prediction model for online change detection, one needs to address several key issues. How to combine the GP model predictions and actual observations to

[2]Henceforth, referred to as GP.

identify changes in online fashion? What covariance function to use to handle periodic data? How to automatically learn the "true" seasonality of the time series? How to perform the analysis in a scalable manner? The scalability issue is especially significant, since direct application of GP based learning methods to large scale data sets is challenging owing to the inherent $O(t^3)$ computational complexity as well as $O(t^2)$ memory storage requirements, where t is the length of the input time series observed so far. The key bottleneck is the handling of a large $t \times t$ covariance matrix and solving a large system of equations involving the covariance matrix.

Our GP based algorithm combines the predictive distribution estimated using GP at time t and the actual observation to obtain a normalized change score which is then monitored using an online *Exponential Weighted Moving Average* (EWMA) model to detect changes. The predictive distribution is also used for simultaneously filtering outliers and imputing missing data. The use of EWMA model allows detection of abrupt as well as slow evolving changes in the time series. We use a *exponential periodic* covariance function coupled with a stationary noise model to handle periodic and short range Markovian dependencies in the data. The hyper-parameters for the covariance function and the true periodicity of the input time series are estimated using a conjugate descent based optimizer. The only other parameter associated with the algorithm are related to setting the control limits for the EWMA monitoring which are estimated from a data using standard methods developed within the statistical quality control literature [18].

To address the scalability requirements, we exploit the special structure of the associated covariance matrix. The proposed method can estimate the exact predictive distribution at time t in $O(t)$ time, while requiring only $O(t)$ memory, which is a significant improvement over traditional $O(t^3)$ update methods which require $O(t^2)$ memory. The same fast method is used within the conjugate descent based optimizer for hyper-parameter estimation.

We provide results on real ECG and NDVI data and synthetic data to demonstrate the ability of the proposed algorithm in identification of changes in presence of noise, outliers, and missing data. The results show that the proposed algorithm can identify changes corresponding to unexpected changes in crop cover in NDVI data and abnormal heart conditions in ECG data. Empirical comparisons with several existing online change detection methods and with an online adaptation of an offline method specifically proposed for NDVI time series, show that the proposed GP based method significantly outperforms existing methods in identifying changes. We also evaluate a similar algorithm which uses a *seasonal autoregressive and integrated moving average* (SARIMA) model as the time series model instead of GP, but experimental results show that the SARIMA model is not able to identify the changes in NDVI time series as effectively as the GP based model.

2 Related Work

Change detection for time series data is widely researched in different research communities such as statistics [14], signal processing [2], and process control [20]. Most of the existing change techniques can be grouped into three categories, viz., *parameter change based techniques* [14, 22], *segmentation based techniques* [21, 25] and *forecasting based techniques* [9, 17]. Most change detection methods proposed for periodic data typically operate in a batch mode, i.e., they require the entire time series to be present [4, 12]. While there have been several methods proposed for online change detection, including statistical methods like CUSUM (cumulative sum) [22] and GLR (Generalized Likelihood Ratio) [13], as well as other methods developed in data mining and machine learning disciplines [1, 8, 9, 16, 17], these methods assume that the periodicity in the time series is removed prior to the analysis.

Recently, several techniques have been proposed that identify changes in NDVI time series data by applying time series change detection techniques [4, 9, 19]. Most of these techniques remove the seasonality from the time series before analysis, except for the *recursive merging* algorithm [4].

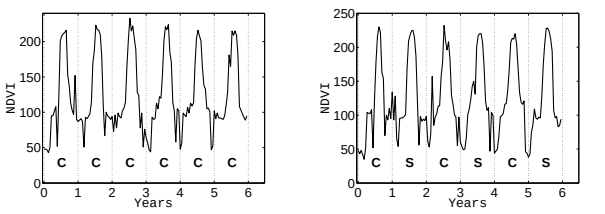
GP have not been explicitly used for change detection in time series. Several papers have used GP for time series modeling and prediction [5, 10] but typically deal with non-periodic time series. Brewer and Stello [5] use GP to model periodic astero-seismic data using a different periodic covariance function.

Several approximation based methods for GP analysis have been proposed in the literature to scale GP to large datasets (See [23, Chapter 8] for a detailed overview), which fall under the general purview of sparse and approximate kernel methods. We focus on scaling GP analysis while obtaining the exact solution. In our earlier work [6], we adapted the algorithm originally proposed by Trench [29, 32] for scalable estimation of the GP hyper-parameters. In this paper we extend it for GP based online change detection.

3 Defining Change for Periodic Time Series

Traditionally, a change occurs in a time series when its characteristics change appreciably. These characteristics are typically translated to statistical properties,

as used by parametric change detection techniques, or structural properties, as used by several segmentation based techniques.



(a) Same Crop Cycles (b) Alternating Crop Cycles

Figure 2: Examples of NDVI time series with annual crop labels (*C - corn, S - soybean*).

Here we define change from the perspective of periodic time series. A periodic time series can have two types of cycles. First is the *natural cycle* in a time series corresponding to a natural temporal duration, such as a day, month, or year. The natural cycle is typically assumed to be a region in the time series which is generated by one "process", e.g., temperature time series has an annual cycle, a building power consumption data has weekly cycle. Biomass observational data, such as NDVI derived from MODIS remote sensors, have an annual cycle. If the land is agricultural, then data in each cycle corresponds to the growing crop. The second type of cycle is caused due to a repetitive pattern in the time series. Thus a *pattern* can be defined as the smallest portion of a time series which repeats itself. Often, the natural cycle length and pattern length do not coincide. For example, Figures 2(a) and 2(b) show two NDVI time series (6 years with 23 observations per year) recorded for two different geographic locations. In the first location, a single crop is grown every year while in the second location, crops are alternated every year. Thus, for the first time series, the natural cycle length and pattern length are equal, while for the second time series, the pattern length is twice that of natural cycle length. The difference between the natural cycle length and the pattern length in a time series is an important characteristic in biomass monitoring. Many time series modeling techniques that use a fixed cycle length cannot adequately model the biomass time series in which pattern length and cycle length are different.

A change in a periodic time series can occur in four ways:

Type 1 - A change occurs within a cycle, e.g., a crop is destroyed in mid-season (See Fig. 3(a)).

Type 2 - A change occurs in the annual cycle, e.g., if a different crop is grown at a given location as compared to previous year (See Fig. 3(b)).

Type 3 - A change occurs in the cyclical pattern, e.g., instead of alternating, a same crop is grown for two or more consecutive years (See Fig. 3(c)).

Type 4 - Similar to the Type 2 change, but in this case the observations are completely different from the previous cycles, e.g., no crop is grown (See Fig. 3(d)).

While the last two change types listed above are more relevant for biomass monitoring, the first two change types are encountered in many domains that deal with time series data. A change detection algorithm is expected to detect all of the above change types for biomass monitoring to be successful.

4 Gaussian Process Based Prediction for Periodic Time Series

We first provide a brief introduction to GP in the context of non-linear time series prediction[3]. We assume that we are interested in learning a nonlinear model $f(x)$, with known inputs, x using observed outputs y, such that the observations are corrupted versions of the function values by an additive noise, ε. Given a set of inputs, $\mathbf{x} = x_1, x_2, \ldots x_n$ and corresponding outputs, $\mathbf{y} = y_1, y_2, \ldots y_n$, we can write

$$y_i = f(x_i) + \varepsilon_i, \ \forall i = 1 \ldots n \tag{4.1}$$

We assume that the additive noise belongs to a stationary Gaussian, i.e., $\varepsilon_i \sim \mathcal{N}(0, \sigma^2), \forall i = 1 \ldots n$.

4.1 GP Prior The random function, $f(x)$ is defined as a GP, with mean function $m(x)$, and covariance function, $k(x_i, x_j)$, if the function values at any number of inputs are jointly Gaussian, i.e.:

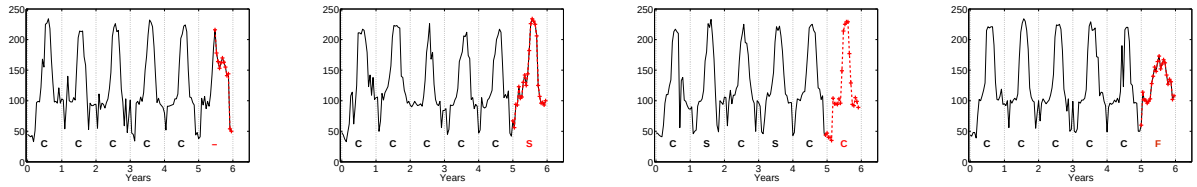
$$f(x_1), f(x_2), \ldots, f(x_n) \sim \mathcal{N}(m(x), K) \tag{4.2}$$

K is a $n \times n$ covariance matrix, where $K_{ij} = k(x_i, x_j)$ is the covariance between function values $f(x_i)$ and $f(x_j)$, which is a function of the inputs, x_i and x_j. Typically, f is assumed to be a zero mean process, i.e., $m(x) = 0$. The covariance is measured as a function of the distance between the pair of inputs.

4.2 Regression with GP Given the prior on f in (4.2) and the relationship between the inputs and outputs in (4.1), GP can be used to obtain a predictive distribution for the function value $f(x^*)$ at a new input x^*. It can be shown that $f(x^*)$ is also Gaussian distributed with mean and variance given as:

$$\hat{f}(x^*) = k(x^*, \mathbf{x})^{\mathsf{T}} (K + \sigma^2 I)^{-1} \mathbf{y} \tag{4.3}$$

[3]We direct the reader to excellent literature on GP [23] for more details about the topic



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

Figure 3: Examples of changes (last year) in NDVI data (*C - corn, S - soybean, F - fallow*).

$$\hat{v}(x^*) = k(x^*, x^*) - k(x^*, \mathbf{x})^\mathsf{T}(K + \sigma^2 I)^{-1} k(x^*, \mathbf{x}) \tag{4.4}$$

where $k(x^*, \mathbf{x}) = [k(x^*, x_1), k(x^*, x_2), \ldots, k(x^*, x_n)]^\mathsf{T}$ is the $n \times 1$ vector of covariances between the new point and the training inputs and $k(x^*, x^*)$ is the self covariance for the test input. Based on the relationship between y^* and $f(x^*)$ in (4.1) the predictive distribution for the unseen target y^* is also Gaussian with mean $\hat{f}(x^*)$ and variance, $(\hat{v}(x^*) + \sigma^2)$.

4.3 Time Series Prediction GP can be used for time series prediction/forecasting by replacing the input x with time index $t = 1, 2, \ldots, T$. As shown above, the output at time t, denoted as y_t, can be predicted as a Gaussian distribution ($\sim \mathcal{N}(\hat{y}_t, \hat{v}_t)$), using the historical observations till $(t-1)$, denoted as $\mathbf{y_{t-1}} = [y_1, y_2, \ldots, y_{t-1}]^\mathsf{T}$, using the following equations:

$$\hat{y}_t = k(t, \mathbf{t-1})^\mathsf{T}(K_{\mathbf{t-1}} + \sigma^2 I)^{-1}\mathbf{y_{t-1}} \tag{4.5}$$

$$\hat{v}_t = k(t,t) - k(t, \mathbf{t-1})^\mathsf{T}(K_{\mathbf{t-1}} + \sigma^2 I)^{-1} k(t, \mathbf{t-1}) \tag{4.6}$$

where $K_{\mathbf{t-1}}$ is a $|\mathbf{t-1}| \times |\mathbf{t-1}|$ kernel matrix defined over inputs $(1, 2, \ldots, t-1)$. Similarly, $k(t, \mathbf{t-1})$ is a $(t-1)$ length column vector with covariance between t and each index in $(\mathbf{t-1})$ and $\mathbf{y_{t-1}}$ is the time series observed till time $(t-1)$.

4.4 Covariance Function for Periodic Time Series The covariance function $k(,)$ is the critical component for the GP based predictive model. For time series, the covariance function determines how high or low is the covariance between the observations, y_{t_1} and y_{t_2}. For example, if the widely used *squared exponential* (*se*) function[4] is used, the observation at time t will have a higher covariance with observations in immediate history, and hence these observations will have a higher impact in estimating the prediction at time t, while the impact of older historical observations will exponentially decay with time.For periodic time series, the observation at time t is strongly correlated with observation at $t-1$ as well as with the observation at time $t-\omega$, where ω is the length of a single cycle in the time series. Such periodic dependency cannot be modeled using the *se* function for any value of l.

To account for the periodic dependency, we use a periodic covariance function, called *Exponential Periodic* (*ep*) function:

$$k(t_1, t_2) = \sigma_f^2 exp(-\frac{\Delta t^2}{2l^2\omega^2}) exp(-\frac{(1 - \cos\frac{2\pi\Delta t}{\omega})}{a}) \tag{4.7}$$

where $\Delta t = |t_1 - t_2|$.

The hyper-parameter l controls how much weight should be given to observations of past cycles when predicting the current observation, while a controls the weight on observations within a cycle. While the parameter ω is often set equal to the number of observations per cycle, it can also be treated as a hyper-parameter if the periodicity of the time series is not known.

4.5 Automatically Estimating Hyper-parameters and Periodicity The *unknown* hyper-parameters associated with the *ep* covariance function are $\Theta = \langle l, a, \sigma_f^2, \sigma_n^2, \omega \rangle$[5]. These hyper-parameters can either be specified by the user based on domain knowledge or can be estimated using a training time series by maximizing the joint log-likelihood for the entire time series. The predictive distribution for each y_t given by (4.5) and (4.6) shows that y_t is dependent on $\mathbf{y_{t-1}}$ and the conditional distribution of y_t is Gaussian. The conditional log-likelihood of y_t, denoted as l_t, can be calculated as:

$$l_t = \log P(y_t | \mathbf{y_{t-1}}) = -\frac{1}{2}\left[\log 2\pi + \log \hat{v}_t + \frac{(y_t - \hat{y}_t)^2}{\hat{v}_t}\right] \tag{4.8}$$

The log-likelihood of the time series, $\mathbf{l_T}$ can be written in terms of the conditional likelihoods l_t calculated in

[4] $k(t_1, t_2) = \sigma_f^2 exp(-\frac{\Delta t^2}{2l^2})$ where $\Delta t = t_1 - t_2$

[5] If periodicity is known then ω is fixed.

(4.8) as:

(4.9)
$$\mathbf{l_T} = \sum_{t=1}^{T} l_t = -\frac{T}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{T}\left[\log \hat{v}_t + \frac{(y_t - \hat{y}_t)^2}{\hat{v}_t}\right]$$

The hyper-parameters are estimated by maximizing the log likelihood using *gradient descent*. The partial derivatives of $\mathbf{l_T}$ with respect to a hyper-parameter, $\theta \in \Theta$, can be calculated as:

(4.10)
$$\frac{\partial \mathbf{l_T}}{\partial \theta} = -\frac{1}{2}\sum_{t=1}^{T}\left[\frac{1}{\hat{v}_t}\frac{\partial \hat{v}_t}{\partial \theta}(1 - \frac{(y_t - \hat{y}_t)^2}{\hat{v}_t}) - \frac{2}{\hat{v}_t^2}(y_t - \hat{y}_t)\frac{\partial \hat{y}_t}{\partial \theta}\right]$$

The partial derivatives of $\hat{p}_t$ and $\hat{v}_t$ can be computed from (4.5) and (4.6) as follows. For notational simplicity we drop $(t-1)$ subscripts and denote $\hat{y}_t = \mathbf{k}^\mathsf{T} K^{-1}\mathbf{y}$ and $\hat{v}_t = k - \mathbf{k}^\mathsf{T} K^{-1}\mathbf{k}$. The partial derivatives can be written as:

$$\frac{\partial \hat{y}_t}{\partial \theta} = \frac{\partial \mathbf{k}^\mathsf{T}}{\partial \theta}K^{-1}\mathbf{y} - \mathbf{k}^\mathsf{T}K^{-1}\frac{\partial K}{\partial \theta}K^{-1}\mathbf{y} \quad (4.11)$$

$$\frac{\partial \hat{v}_t}{\partial \theta} = \frac{\partial k^\mathsf{T}}{\partial \theta} - (2\frac{\partial \mathbf{k}^\mathsf{T}}{\partial \theta} - \mathbf{k}^\mathsf{T}K^{-1}\frac{\partial K}{\partial \theta})K^{-1}\mathbf{k} \quad (4.12)$$

In this paper, we treat ω as a hyper-parameter. The estimated value of ω is rounded to the nearest multiple of the natural periodicity of the time series, e.g., for a time series with monthly data, i.e., natural periodicity of 12, if ω is estimated to be 20.3, then we round it off to 24.

5 GP Based Online Change Detection Algorithm

In this section we describe GPChange, an online change detection algorithm which uses the GP based time series predictive model as discussed in Section 4. The key assumption behind GPChange is that the observation at time t is assumed to be generated from a parametric reference distribution (Gaussian), denoted as $\mathcal{P}_t$, whose parameters (θ_t) are estimated using the data observed so far, using the GP based predictive model described in Section 4. Let the expected value of y_t be denoted as $\hat{y}_t = \mathbb{E}_{\mathcal{P}_t}(y_t|\hat{\theta}_t)$ and the uncertainty associated with the prediction, or the variance be denoted as $\hat{v}_t = \mathbb{E}_{\mathcal{P}_t}((y_t - \hat{y}_t|\hat{\theta}_t)^2)$. Let p_t denote the p-value for the observation y_t under the reference distribution $\mathcal{P}_t$, i.e.,

$$p_t = \begin{cases} \mathcal{P}_t(\hat{y}_t > y_t) & \text{if } y_t > \hat{y}_t \\ \mathcal{P}_t(\hat{y}_t < y_t) & \text{otherwise} \end{cases} \quad (5.13)$$

Note that, in the early stages of monitoring, the estimate of $\hat{y}_t$ might not be accurate due to lack of sufficient history. But the GP predictive model will also estimate large value for $\hat{v}_t$ and hence the p-value for y_t will not be alarmingly low. Similarly, if the historical data is highly noisy, the GP model will estimate large variability in the predictions and hence will not result in false alarms.

It is well known that if $\mathcal{P}_t$ is a continuous distribution, then in absence of any change, the distribution of p_t is uniform$(0,1)$. Hence the normal score for y_t, denoted as $z_t = \Phi^{-1}(p_t)$, will have an approximately normal distribution ($\sim \mathcal{N}(0,1)$) [9] (Φ^{-1} is the inverse normal cumulative distribution function). Note that if the observation y_t is very far from the mean of the predictive distribution, $\hat{y}_t$, p_t will be low and hence z_t will also be low.

Since z_t is expected to be normally distributed, we can monitor z_t to identify when a change occurs in a time series. This approach is also known as Q-charting in statistical quality control literature [18]. A running change score s_t, estimated as the EWMA of z_t, is maintained and is updated at every time instance t as:

$$s_t = \lambda z_t + (1-\lambda)s_{t-1} \quad (5.14)$$

where $\lambda \in (0,1]$. The initial score s_0 is the in-control score and is typically set to 0. If the process is *in control*, i.e., the score s_t is expected to be approximately normal $\sim \mathcal{N}(0, \sigma_{s_t}^2)$ [20], where, $\sigma_{s_t}^2 = \frac{\lambda}{2-\lambda}$. A change is declared when the change score falls below a threshold, i.e.,$s_t < M\sigma_{s_t} = M\sqrt{\frac{\lambda}{2-\lambda}}$. M defines the control limits in terms of number of standard deviations away from the mean. Choice of λ in (5.14) determines if one is interested in identifying sudden large changes (high λ), or small gradual changes (low λ). Typically, the choice of M and λ are related to the *average run length* (ARL), which is the average time between successive changes in an in-control time series and can be set based on the expected ARL [18].

The individual steps of GPChange are shown in Algorithm 1. The input to the algorithm are the time series to be monitored starting from observation $(n+1)$ $(n > 1)$, the hyper-parameters associated with the covariance function (Θ), the periodicity of the time series (ω), a parameter α to identify outliers in the time series, and parameters λ and M related to the EWMA component of the algorithm. The output of the algorithm is a vector of alarms which contains 1 at locations where a change was identified. The algorithm can be easily turned into a continuous monitoring algorithm, i.e., $T = \infty$.

Handling Outliers and Missing Data An outlier is typically a single observation which is different from rest of the time series and occurs independent of the change. To detect such outliers, the p-value, p_t is compared to a threshold α. If $p_t < \alpha$, the observation

Algorithm 1 GPChange($\mathbf{y_T}, n, \Theta, \omega, \alpha, \lambda, M$)

$s_1 \dots s_n \leftarrow 0$
$a_1 \dots a_n \leftarrow 0$
for $t = n+1$ to T **do**
  Compute $\hat{y}_t$ and $\hat{v}_t$ (See (4.5) and (4.6))
  if y_t is *missing* **then**
    $y_t = \hat{y}_t$
  end if
  $p_t \leftarrow p$-value for y_t under $\mathcal{N}(\hat{y}_t, \hat{v}_t)$
  if $p_t < \alpha$ **then**
    $y_t \leftarrow$ Random sample from tail of $\mathcal{N}(\hat{y}_t, \hat{\sigma}_t^2)$ {y_t is outlier}
  end if
  $z_t \leftarrow \frac{|y_t - \hat{y}_t|}{\sqrt{\hat{v}_t}}$
  $s_t \leftarrow \lambda z_t + (1-\lambda) s_{t-1}$
  if $|s_t| > M\sqrt{\frac{\lambda}{2-\lambda}}$ **then**
    $a_t \leftarrow 1$ {Raise Alarm}
  end if
end for
return $\mathbf{a_T}$

Algorithm 2 ToeplitzSolve($\mathbf{k_t}, \mathbf{y_t}$)

if $k_1 != 1$ **then**
  $\mathbf{k_t} \leftarrow \mathbf{k_t}/k_1$, $\mathbf{y_t} \leftarrow \mathbf{y_t}/k_1$
end if
$\mathbf{z_1} \leftarrow y_1$, $\mathbf{g_1} \leftarrow -k_2$, $\lambda_1 \leftarrow 1 - k_2^2$
for $i = 1$ to $t-2$ **do**
  $\theta_i \leftarrow y_{i+1} - \mathbf{z_i}^\mathsf{T} \hat{\mathbf{k}}_{\mathbf{2:i+1}}$
  $\gamma_i \leftarrow -k_{i+2} - \mathbf{g_i}^\mathsf{T} \hat{\mathbf{k}}_{\mathbf{2:i+1}}$
  $\mathbf{z_{i+1}} \rightarrow \begin{bmatrix} \mathbf{z_i} + \frac{\theta_i}{\lambda_i}\hat{\mathbf{g}}_\mathbf{i} \\ \frac{\theta_i}{\lambda_i} \end{bmatrix}$
  $\mathbf{g_{i+1}} \rightarrow \begin{bmatrix} \mathbf{g_i} + \frac{\gamma_i}{\lambda_i}\hat{\mathbf{g}}_\mathbf{i} \\ \frac{\gamma_i}{\lambda_i} \end{bmatrix}$
  $\lambda_{i+1} \rightarrow \lambda_i - \frac{\gamma_i^2}{\lambda_i}$
end for
$\theta_{t-1} \rightarrow y_t - \mathbf{z_{t-1}}^\mathsf{T} \hat{\mathbf{k}}_{2:t}$
$\mathbf{z_t} \rightarrow \begin{bmatrix} \mathbf{z_{t-1}} + \frac{\theta_{t-1}}{\lambda_{t-1}}\hat{\mathbf{g}}_{t-1} \\ \frac{\theta_{t-1}}{\lambda_{t-1}} \end{bmatrix}$
return $\mathbf{z_t}, \mathbf{g_{t-1}}, \lambda_{t-1}$

is considered to be an outlier. But since y_t is used to predict the values downstream, the outlier can bias the predictions. Hence y_t is replaced with a random value sampled from the tail of the distribution $\mathcal{P}_t$. If the data at any time point t is *missing*, the algorithm replaces it with the predicted value, $\hat{y}_t$.

6 Addressing Scalability Issues

The computational bottleneck for using GP based prediction in Algorithm 1 is computing $\hat{y}_t = \mathbf{k}^\mathsf{T} K^{-1}\mathbf{y}$ and $\hat{v}_t = k - \mathbf{k}^\mathsf{T} K^{-1}\mathbf{k}$ at every time step. The key operations required for obtaining $\hat{y}_t$ and $\hat{v}_t$ are solving the systems of equations, $K\mathbf{z} = \mathbf{y}$ and $K\mathbf{z} = \mathbf{k}$ to obtain $K^{-1}\mathbf{y}$ and $K^{-1}\mathbf{k}$, respectively. Same operations are required for hyper-parameter estimation in (4.11) and (4.12). At each time step, solving the system of equations involving K will require $O(T^3)$ time, resulting in a $O(T^4)$ complexity to monitor a time series of length T. Moreover, the memory requirement for storing K (and $\frac{\partial K}{\partial \theta}$ during hyper-parameter estimation) is quadratic in t, which can become a critical issue when dealing with long time series.

We address this issue by exploiting the fact that the covariance matrix obtained using the *ep* covariance function has a special structure, i.e., it is a symmetric positive definite Toeplitz matrix, K, as shown below:

$$K = \begin{pmatrix} k_0 & k_1 & \dots & k_{t-1} \\ k_1 & k_0 & \dots & k_{t-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{t-1} & k_{t-2} & \dots & \dots & k_0 \end{pmatrix} \tag{6.15}$$

One can straightaway note that K in (6.15) can be represented using just the first row (or column) of K. This characteristic immediately provides a way to reduce the memory requirements of the algorithms involving K from $O(T^2)$ to $O(T)$.

6.1 Using Toeplitz Matrix Operations

In our earlier work [6], we adapted the algorithm originally proposed by Trench [29, 32] for scalable estimation of the GP hyper-parameters[6] For completeness, we provide the fast algorithm for solving a symmetric Toeplitz system of equations in Algorithm 2. The algorithm takes the first row of the input Toeplitz matrix, $\mathbf{k_t} = \{k_1, k_2, \dots, k_t\}$, such that $K = Toeplitz(\mathbf{k_t})$, and the target vector $\mathbf{y_t}$ as inputs[7]. Note that this algorithm is $O(t^2)$ and requires $O(t)$ memory only. More details can be found in our earlier work [6].

If the scalable adaptation [6] is directly applied at every time step in Algorithm 1, one can only reduce the complexity of obtaining $K^{-1}\mathbf{y}$ and $K^{-1}\mathbf{k}$ to $O(T^2)$, resulting in the overall complexity of $O(T^3)$ for monitoring a T length time series, which is still prohibitive for long time series.

Careful analysis of Algorithm 2 shows that the solution vector $\mathbf{z_t}$ is incrementally obtained in the for loop using results from the previous iteration. Inside the for loop, the operations take linear time in computing two dot products. We utilize this fact to come up with an incremental solver (shown as Algorithm 3 which uses results obtained from solving $\mathbf{y_t} = Toeplitz(\mathbf{k_t})\mathbf{z_t}$ to solve $\mathbf{y_{t+1}} = Toeplitz(\mathbf{k_{t+1}})\mathbf{z_{t+1}}$. The inputs to the incremental algorithm are $\mathbf{k_{t+1}}$, the last entry in

[6] Zhang et al. [31] also derived similar results for Gaussian process regression on time series.

[7] In the algorithm $\hat{\mathbf{x}}$ denotes a vector obtained by reversing the vector $\mathbf{x}$. A portion of a vector is denoted as $\mathbf{x}_{i:j}$.

Algorithm 3 ToeplitzSolveInc($\mathbf{k_{t+1}}, y_{t+1}, \mathbf{z_t}, \mathbf{g_{t-1}}, \lambda_{t-1}$)

if k_1 != 1 **then**
  $\mathbf{k_{t+1}} \leftarrow \mathbf{k_{t+1}}/k_1, y_{t+1} \leftarrow y_{t+1}/k_1$
end if
$\theta_t \rightarrow y_{t+1} - \mathbf{z_t}^\top \hat{\mathbf{k}}_{\mathbf{2:t}}$
$\gamma \leftarrow -k_t - \mathbf{g_{t-1}}^\top \hat{\mathbf{k}}_{\mathbf{2:t-1}}$
$\lambda_t \leftarrow \lambda_{t-1} - \frac{\gamma^2}{\lambda_{t-1}}$
$\mathbf{g_t} \leftarrow \begin{bmatrix} \mathbf{g_{t-1}} + \frac{\gamma}{\lambda_{t-1}} \hat{\mathbf{g}}_{\mathbf{t-1}} \\ \frac{\gamma}{\lambda_{t-1}} \end{bmatrix}$
$\mathbf{z_{t+1}} \rightarrow \begin{bmatrix} \mathbf{z_t} + \frac{\theta_t}{\lambda_t} \hat{\mathbf{g}}_\mathbf{t} \\ \frac{\theta_t}{\lambda_t} \end{bmatrix}$
return $\mathbf{z_{t+1}}, \mathbf{g_t}, \lambda_t$

the target vector, y_{t+1}, and $\mathbf{z_t}, \mathbf{g_{t-1}}, \lambda_{t-1}$ which are obtained by calling *ToeplitzSolve* or *ToeplitzSolveInc* for $\mathbf{y_t} = Toeplitz(\mathbf{k_t})\mathbf{z_t}$. The output is the new solution vector $\mathbf{z_{t+1}}$ as well as outputs that can be used by next call of *ToeplitzSolveInc*.

The first three steps in *ToeplitzSolveInc*, i.e., normalization of $\mathbf{k_{t+1}}$ and calculation of θ_t and γ require $O(t)$ time while all other operations are $O(1)$. Thus the algorithm can solve the Toeplitz system of equations in $O(T)$ time resulting in an overall $O(T^2)$ complexity for monitoring a T length time series. The memory requirement for the algorithm is $O(T)$ as well.

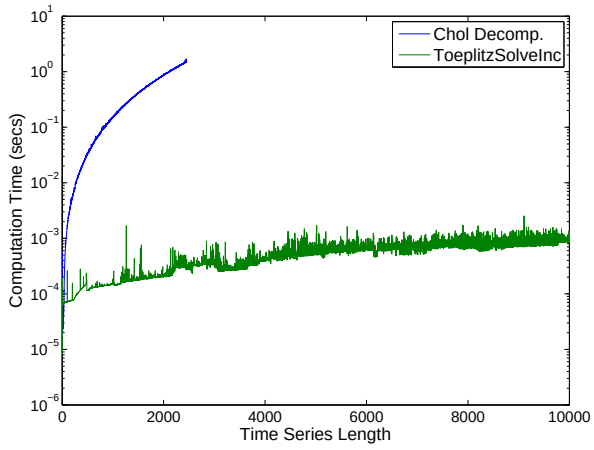


Figure 4: Comparing performance of *ToeplitzSolveInc* against traditional Cholesky decomposition based method for solving a Toeplitz system of equations. Y-axis is on log scale.

To provide a sense of the speedup provided by *ToeplitzSolveInc* over a traditional solution using *Cholesky decomposition*[8] for computing $\mathbf{y_t} = Toeplitz(\mathbf{k_t})\mathbf{z_t}$, we compared the two methods to solve the system of equations for increasing t. We first invoke *ToeplitzSolve* for $t = 1, 2$, and then invoke the incremental *ToeplitzSolveInc* for subsequent values of t. The run time comparisons are shown in Figure 4[9]. The traditional method failed to run beyond length 2452 due to lack of memory. For $t = 2452$, the proposed *ToeplitzSolveInc* method achieves a speedup of 5300 over the traditional Choleksy decomposition based method while obtaining the exact same solution.

7 Experimental Evaluation

In this section we experimentally evaluate the performance of GPChange in identifying changes in periodic time series data. We present results from two sets of experiments. The first set of experiments demonstrate the applicability of the GPChange algorithm in identifying known changes in different types of time series. The second set of experiments demonstrate the precision and recall of GPChange in identifying time series with changes from a data set consisting of time series containing changes and time series without any changes. For each experimental run, we assume that the initial portion of the time series is change free. This training portion is used to train the hyper-parameters, learn the "true" periodicity of the data, and to estimate the control chart parameters, M and λ (using the expected ARL based methods [18]).

7.1 Other Methods For comparison, we evaluated five existing methods which can be directly used or adapted for online change detection in periodic time series:

7.1.1 Seasonal ARIMA (SARIMA) [27] Uses a *Seasonal Auto Regressive Integrated Moving Average* model as the time series model. We adapt this widely used time series model for change detection by substituting it in place of GP based model in Algorithm 1. The SARIMA model requires total 6 parameters, 3 parameters (number of autoregressive terms, number of differences, and number of moving average terms) for the non-seasonal part and corresponding 3 parameters for the seasonal part. The best setting for these parameters is found by evaluating multiple setting using the AIC criterion.

7.1.2 Recursive Merging (RM) [4] Finds most anomalous cycle in a time series by recursively merging the two closest cycles until all cycles are merged into one. The last cycle to be merged is identified as the

[8] Using MATLAB's Cholesky decomposition implementation

[9] Experiments run on an Intel Core 2 Duo Desktop @ 2.40 GHz with 3.00 GB memory.

cycle with change and is assigned a change score. Note that RM assigns a change score to a time series and identifies an entire cycle as containing change. In our experiments, RM is run in retrospective mode. If the cycle with highest score as determined by RM matches the cycle containing the true change, we declare RM to be successful.

7.1.3 Cumulative Sum Control Charts (CUSUM) and Likelihood Ratio Test (LRT) CUSUM charts [22] and LRT [8, 13] are widely used statistical methods for online change detection but are limited to time series without any seasonal component. We apply CUSUM and LRT to periodic data by first separating the trend, seasonality, and noise from the time series using the widely used STL procedure [7] and then removing the seasonal component from the original time series. Both CUSUM and LRT require control limits to identify presence of change which are estimated from the training portion of the data using methods based on the expected ARL.

7.1.4 Bayesian Online Change Detection (BOCD) [1] This method estimates the predictive model over *run length* without change. For a normal time series the value should increase monotonically, but if the estimated run length drops down to 0, a change is declared at that time instance. This method also requires seasonality removal.

7.2 Data Description

7.2.1 Synthetic Data This data was generated by adding random Gaussian noise to a squared sine wave for each cycle, i.e., $y_t = a \sin^2 \frac{\pi t}{\omega} + \sigma^2$. Three such time series were generated (length = 200), SYNC1, in which the last cycle was generated using a different periodicity $\omega' = 2\omega$), SYNC2, in which the amplitude is changed for the last cycle ($a' = 1.2a$), and SYNC3 in which the Gaussian noise covariance is changed in the last cycle ($\sigma' = 2 * \sigma$).

7.2.2 ECG Data We use two ECG time series (ECG1 and ECG2) from the *Creighton University Ventricular Tachyarrhythmia Database* available in the *Physiobank* [11] archives. The first time series (ECG1) is 6 minutes long (length = 1500) ECG recording of a person experiencing *Ventricular Fibrillation* and the second (ECG2) is a 5 minutes long (length = 1050) recording of a person experiencing *Ventricular Tachychardia* (both are life threatening forms of arrhythmia).

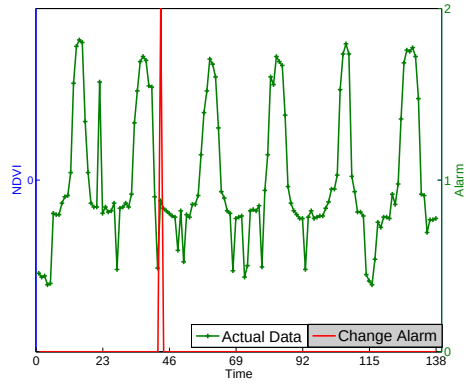
7.2.3 Iowa Crop Biomass NDVI Data We used NDVI time series data available as a MODIS data product for years 2001 to 2006 with observations for every 16 days. MODIS data is highly contaminated by clouds; we treat an observation with an associated unreliable flag as missing, and use the GPChange algorithm to impute the missing values. We collected ground truth for randomly selected locations spread throughout the Iowa state. Time series for 4 such locations (length = 138) were shown earlier in Figure 3, we will refer to these time series as NDVI1, NDVI2, NDVI3, and NDVI4. We also created a labeled validation data set of 98 time series consisting of 49 with a change in the last year and 49 with no change in the last year.

7.3 Performance on Time Series with No Change We first study the performance of the proposed algorithm in monitoring a 6 year NDVI time series with no change and with crop rotation every year ($\omega = 46$). The estimated value of ω using the first four years of data is 39.5 and hence we chose 46 as the periodicity.

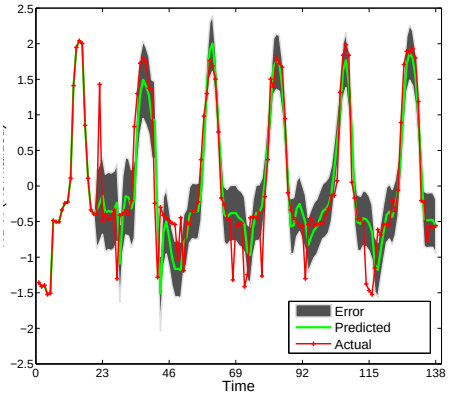
Figures 5(a) shows the control chart obtained after applying GPChange with $n = 1$ and Figure 5(b) shows the predictions (mean and variance). Note that the algorithm handles the outlier observation at $t = 22$ and reduces its influence on the future observations. The initial predictions made by the algorithm are different from the actual observations but after sufficient time, the algorithm models the actual data effectively. The algorithm identifies one change, during the beginning of the year. The reason for this behavior is that the NDVI data during the beginning of the year has high variance, but since the predictive variance in GPChange is independent of the observations (See (4.6)) it does not take that information into account.

7.4 Performance on Time Series with Actual Changes We now illustrate the performance of GPChange in identifying changes in the synthetic, ECG, an NDVI time series described in Section 7.2. Table 1 summarizes the performance of GPChange against the existing methods on the different time series. For RM method, we declare that a change is detected if the period containing the actual change is assigned the highest change score. For other methods, including GPChange, we consider that a change is detected only if the number of change points identified in the period containing actual change is greater than the number of change points identified in other periods.

Results in Table 1 show that the proposed GPChange algorithm is able to detect changes in the different types of time series, while other methods, es-



(a) Change Detection Using GPChange



(b) GP Predictions

Figure 5: Output of GPChange for a time series with no change and alternating crop cycle.

pecially the ones that operate on deseasonalized data, are not able to identify changes in NDVI time series. On the other hand, the RM algorithm, which operates on original time series, results in relatively better performance than others, thought it is not suitable to solve the online change detection problem. The SARIMA model based algorithm is also not able to identify changes in two of the time series, indicating that the GP based time series model is a better alternative for the change detection task. We could not fit the SARIMA model on the ECG time series (length > 1050, $\omega = 250$), as it ran out of memory on a PC desktop with 3 GB memory[10].

Figures 6(a)–6(d) shows the performance of the proposed algorithm for NDVI time series with the four types of changes. GPChange is able to identify all four types of change. For type 2 change, the detection happens as soon as the change occurs at the beginning

[10]Using ARIMA implementation in R.

	GPC	SARM	RM	CSUM	LRT	BOCD
SYNC1	√	√	√	×	√	×
SYNC2	√	×	√	×	√	×
SYNC3	√	√	√	×	×	×
ECG1	√	*NA*	√	√	×	√
ECG2	√	*NA*	√	√	×	√
NDVI1	√	×	×	×	×	×
NDVI2	√	√	√	×	√	×
NDVI3	√	√	√	×	×	×
NDVI4	√	√	√	√	√	×

Table 1: Relative performance of different change detection algorithms.

of the year, while for the type 4 change, the threshold is crossed only by middle of the season, but this is expected since the first half of the cycle appears to be similar to the crop for previous years. For type 3 change, the algorithm manages to detect the change only towards the end of the cycle, the reason being that last year's crop (*corn*) is not very different from the expected crop (*soybean*), though a lower λ might result in the threshold getting crossed sooner. For type 1 change, the algorithm identifies a change as soon as it happens in the middle of the season, but also identifies a change in the beginning of the season, possibly due to the absence of a sustained drop in NDVI which happens in all of the previous years.

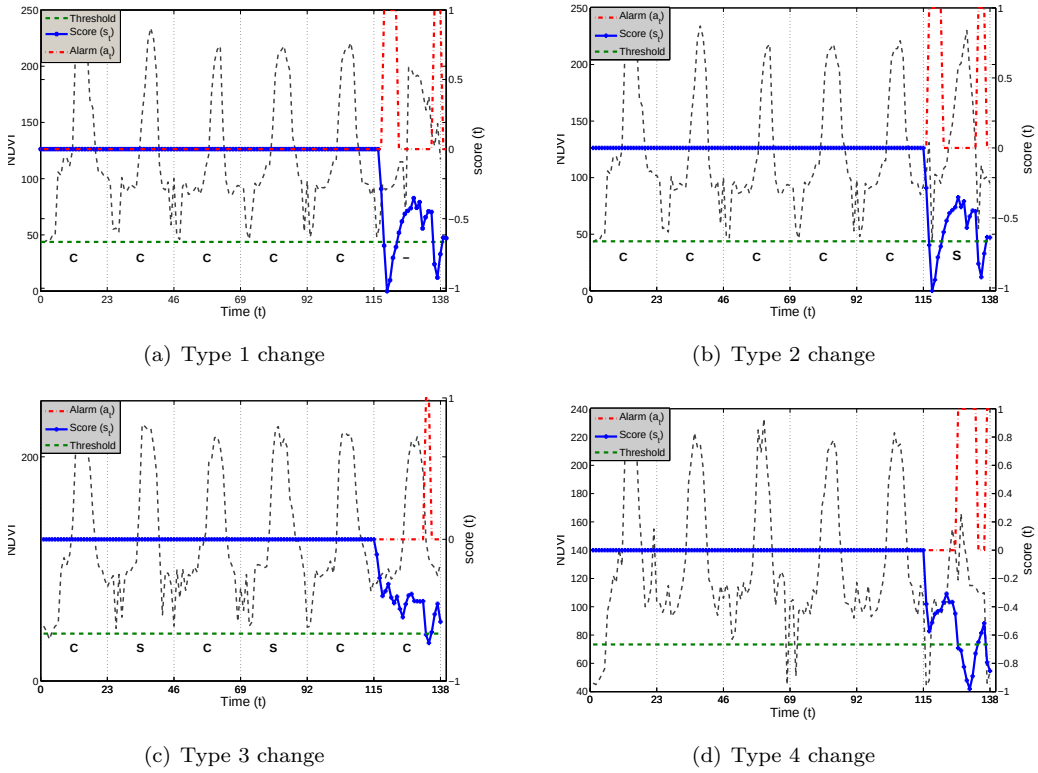
7.5 Sensitivity and Specificity Analysis To understand the ability of the GPChange method to differentiate between time series that contain a real change and time series that do not, we evaluate it and the other existing methods on the Iowa validation data set of NDVI time series. The data set contains 49 time series with change and 49 with no change. The results (accuracy, precision, and recall metrics) are summarized in Table 2. Results in Table 2 shows that the proposed GPChange based algorithm can identify changes with highest accuracy when compared to other methods. The Bayesian online change detection algorithm does not identify any changes, indicating that removing seasonality results in loss of information which adversely affects the performance of methods that operate on deseasonalized data.

	GPC	SARM	RM	CSUM	LRT	BOCD
Acc.	78%	61 %	69%	75%	47 %	50 %
Rec.	75%	47 %	49%	75%	47 %	0 %
Prec.	82%	66 %	83%	75%	47 %	-

Table 2: Performance of GPChange and other existing methods on Iowa data set in terms of accuracy, recall, and precision (for identifying time series with change).

8 Conclusions and Future Work

The significance of the GPChange algorithm is immense in the context of biomass monitoring, both for economic



(a) Type 1 change

(b) Type 2 change

(c) Type 3 change

(d) Type 4 change

Figure 6: Output of GPChange for different types of changes in NDVI time series. The periodic gray plot is the NDVI time series.

and energy security reasons. Crops (both food and energy) are susceptible to diseases, natural disasters, and bioterrorism, and hence there has been a growing need for a robust method for online monitoring and rapid identification of changes in biomass using remote sensing as well as other observational data.

The proposed GPChange algorithm is one such analysis tool, that meets the requirements posed by the application domain. Finding changes in periodic time series data poses issues that traditional parametric change detection techniques find challenging to handle. The GP based non-parametric algorithm uses the entire available history, in an intelligent manner, to estimate the predictive distribution at a given time step. A key strength of the algorithm is that it is highly flexible. Besides online change detection, it can also be used for removing noise from data, imputing for missing values, outlier detection, time series smoothing and forecasting. It can use a clean portion of the data for training or it can start the monitoring from the beginning of the time series. While typical change detection in time series moves along the time (left to right), GPChange is not limited to operate in this direction. It can easily be adapted to work in a retrospective mode, i.e., based on the present data identify if a change has occurred in the past. This feature is valuable in the context of land cover monitoring, since there is reliable ground truth available for recently collected data, but not for the past data, and it is important to understand if the land cover had changed in the past at a given geographic location and when the change occurred. The proposed algorithm can be easily adapted to provide such capability.

Another strength of the proposed algorithm is that all associated parameters, including the hyper-parameters of the *ep* covariance function and the periodicity of the time series, can be automatically estimated from a training portion of the time series, as shown in Section 4.5. While monitoring, these hyper-parameters are not recomputed, and hence the algorithm operates under the assumption that the hyper-parameters do not change significantly with time. This is a reasonable assumption, since the hyper-parameters, e.g., l (See (4.7)), are not the true parameters of the predictive model, but only control how much weight should be assigned to historical observations to make the next prediction.

While GP allows us to build a powerful analysis tool, its inherent computational complexity is a major obstacle in handling large scale data sets. Our proposed

algorithm cuts down the analysis time from $O(T^4)$ to $O(T^2)$ for analyzing a T length time series, while maintaining a $O(T)$ memory footprint. This opens up the possibility of applying GPChange to long time series, such as the ECG recordings.

In its current state, the GPChange algorithm seemingly uses all data since start to make predictions at time t. But due to the nature of the covariance function, specifically the effect of the hyper-parameter l (See (4.7)), the impact of older observations is much less compared to that of newer observations. From computational complexity perspective, it might be desirable to completely drop the older observations. It needs to be investigated if the *ToeplitzSolveInc* can be extended to efficiently remove older observations. Another future direction is to incorporate the spatial dependencies across a spatial neighborhood to enhance the change detection performance of GPChange for biomass monitoring.

References

[1] R. P. Adams and D. J. MacKay. Bayesian online changepoint detection. Tech. report, University of Cambridge, UK, 2007.

[2] M. Basseville. Detecting changes in signals and systems–a survey. *Automatica*, 24(3):309 – 326, 1988.

[3] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[4] S. Boriah, et al. Land cover change detection: a case study. *Proc. 14th KDD*, pages 857–865, 2008.

[5] B. J. Brewer and D. Stello. Gaussian process modelling of asteroseismic data. *Monthly Notices of the Royal Astronomical Society*, 395(4):2226–2233, June 2009.

[6] V. Chandola and R. R. Vatsavai. Scalable time series change detection for biomass monitoring using gaussian process. *Proc. NASA CIDU*, September 2010.

[7] R. B. Cleveland, et al. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

[8] F. Desobry, et al. An online kernel change detection algorithm. *Signal Processing, IEEE Trans.*, 53(8):2961–2974, Aug. 2005.

[9] Y. Fang, et al. Online change detection: Monitoring land cover from remotely sensed data. In *ICDMW '06*, pages 626–631, 2006.

[10] A. Girard, et al. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. In *NIPS*, pages 529–53, 2003.

[11] A. L. Goldberger, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23): e215–e220, 2000.

[12] C. M. Gruner and D. H. Johnson. Detection of change in periodic, nonstationary data. In *ICASSP '96*, pages 2471–2474, 1996.

[13] F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, 2000.

[14] C. Inclán and G. C. Tiao. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427):913–923, 1994.

[15] C. O. Justice, et al. The Moderate Resolution Imaging Spectrradiometer (MODIS): Land remote sensing for global change research. *IEEE Trans. Geosciences and Remote Sensing*, 36:1228–1249, 1998.

[16] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *SDM*, pages 389–400, 2009.

[17] D. Lambert and C. Liu. Adaptive thresholds monitoring streams of network counts online. *J. Amer. Stat. Assc.*, 101(473):78–88, 2006.

[18] J. M. Lucas and M. S. Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1):1–29, 1990.

[19] R. S. Lunetta, et al. Land-cover change detection using multi-temporal MODIS NDVI data. *Remote Sensing of Environment*, 105(2):142–154, 2006.

[20] D. C. Montgomery. *Introduction to Statistical Quality Control*. John Wiley and Sons, 4th. edition, 2001.

[21] T. Ogden and E. Parzen. Change-point approach to data analytic wavelet thresholding. *Statistics and Computing*, 6(2):93–99, 1996.

[22] E. S. Page. On problems in which a change can occur at an unknown time. *Biometrika*, 44:248–252, 1957.

[23] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[24] T. Searchinger, et al. Use of U.S. Croplands for Biofuels Increases Greenhouse Gases Through Emissions from Land-Use Change. *Science*, 319(5867):1238–1240, 2008.

[25] M. Sharifzadeh, et al. Change detection in time series data using wavelet footprints. *Advances in Spatial and Temporal Databases*, pages 127–144, 2005.

[26] M. Staudacher, et al. A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep. *Physica A Stat. Mech. and its Appl.*, 349:582–596, Apr. 2005.

[27] B. K. Subhabrata, et al. Sketch-based change detection: Methods, evaluation, and applications. In *In Internet Measurement Conference*, pages 234–247, 2003.

[28] L. A. Tatum. The southern corn leaf blight epidemic. *Science*, 171(3976):1113–1116, 1971.

[29] W. F. Trench. Weighting coefficients for the prediction of stationary time series from the finite past. *SIAM J. Appl. Math.*, 15(6):1502–1510, 1967.

[30] M. Wise, et al. Implications of Limiting CO2 Concentrations for Land Use and Energy. *Science*, 324(5931):1183–1186, 2009.

[31] Y. Zhang, et al. Time-series gaussian process regression based on Toeplitz computation of O(n2) operations and O(n)-level storage. In *CDC-ECC*, 2005.

[32] S. Zohar. The solution of a toeplitz set of linear equations. *J. ACM*, 21(2):272–276, 1974.