BAYESIAN TIME SERIES FORECASTING WITH CHANGE POINT AND ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting plays a crucial role in marketing, finance and many other quantitative fields. A large amount of methodologies has been developed on this topic, including ARIMA, Holt–Winters, etc. However, their performance is easily undermined by the existence of change points and anomaly points, two structures commonly observed in real data, but rarely considered in the aforementioned methods. In this paper, we propose a novel state space time series model, with the capability to capture the structure of change points and anomaly points, as well as trend and seasonality. To infer all the hidden variables, we develop a Bayesian framework, which is able to obtain distributions and forecasting intervals for time series forecasting, with provable theoretical properties. For implementation, an iterative algorithm with Markov chain Monte Carlo (MCMC), Kalman filter and Kalman smoothing is proposed. In both synthetic data and real data applications, our methodology yields a better performance in time series forecasting compared with existing methods, along with more accurate change point detection and anomaly detection.

1 INTRODUCTION

Time series forecasting has a rich and luminous history, and is essentially important in most of business operations nowadays. The main aim of time series forecasting is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model which could describe the inherent structure of the series, in order to generate future values. For instance, the internet companies are interested in the number of daily active users (DAU), say, what is DAU after certain period of time, or when will reach their target DAU goal.

Time series forecasting is a fruitful research area with many existing methodologies. The most popular and frequently used time series model might be the Autoregressive Integrated Moving Average (ARIMA) (Box et al., 2015; Zhang, 2003; Cochrane, 2005; Hipel & McLeod, 1994). Taking seasonality into consideration, Box et al. (2015) proposed the Seasonal ARIMA. The Holt–Winters method (Winters, 1960) is also very popular by using exponential smoothing. State space model (Durbin & Koopman, 2012; Scott & Varian, 2014; Brodersen et al., 2015) also attracts much attention, which is a linear function of an underlying Markov process plus additive noise. Exponential Smoothing State Space Model (ETS) (Hyndman et al., 2008) decomposes times series into error, trend, seasonal that change over time. Recently, deep learning is applied for time-series trend learning using LSTM (Tao Lin, 2017), bidirectional dynamic Boltzmann machine (Osogami et al., 2017) is applied for time-series long-term dependency learning, and coherent probabilistic forecast (Taieb et al., 2017) is proposed for a hierarchy or an aggregation-level comprising a set of time series. Orthogonal to these works, this paper focuses on robust ways of time series forecasting in presence of change points and anomalies.

In Internet time series forecasting, Google develops the Bayesian structure time series (BSTS) model (Brodersen et al., 2015; Scott & Varian, 2014) to capture the trend, seasonality, and similar components of the target series. Recently, Facebook proposes the Prophet approach (Taylor & Letham, 2017) based on a decomposable model with interpretable parameters that can be intuitively adjusted by analyst.

However, as in the DAU example, some special events like Christmas Holiday or President Election, newly launched apps or features, may cause short period or long-term change of DAU, leading to weird forecasting of those traditional models. The aforementioned special cases are well known as

- Anomaly points. The items, events or observations that don't conform to an expected pattern or other items in the dataset, leading to a sudden spike or decrease in the series.
- Change points. A market intervention, such as a new product launch or the onset of an advertising (or ad) campaign, may lead to the level change of the original series.

Time series forecasting without change/anomaly point detection and adjustment may also lead to bizarre forecasting since these models might learn the abrupt changes in the past. There are literatures on detecting anomaly or change points individually, examples can be found in Twitter (2017); Netflix (2017); Barry & Hartigan (1993); Killick & Eckley (2014); twitter (2017). However, the aforementioned change point detection models could not support detection in the presence of seasonality, while the presence of trend/change point is not handled by the anomaly detection models. Most importantly, there is a discrepancy between anomaly/change points detection and adjustment, and commonly used manually adjustment might be a bit arbitrary. Unfortunately, the forecasting gap caused by abnormal and change points, to the best of our knowledge, has not been given full attention and no good solution has been found so far. This paper is strongly motivated by bridging this gap.

In this paper, to overcome the limitations of the most (if not all) current models that the anomaly points and change points are not properly considered, we develop a state space time series forecasting model in the Bayesian framework that can simultaneously detect anomaly and change points and perform forecasting. The learned structure information related to anomaly and change points is automatically incorporated into the forecasting process, which naturally enhances the model prediction based on the feedback of state-space model. To solve the resultant optimization problem, an iterative algorithm based on Bayesian approximate inference with Markov chain Monte Carlo (MCMC), Kalman filter and Kalman smoothing is proposed. The novel model could explicitly capture the structure of change points, anomaly points, trend and seasonality, as also provide the distributions and forecasting intervals due to Bayesian forecasting framework. Both synthetic and real data sets show the better performance of proposed model, in comparison with existing baseline. Moreover, our proposed model outperforms state-of-the-art models in identifying anomaly and change points.

To summarize, our work has the following contributions.

- We proposed a robust [1] Bayesian state-space time series forecasting model that is able to explicitly capture the structures of change points and anomalies (which are generally ignored in most current models), and therefore automatically adapt for forecasting by incorporating the prior information of trend, seasonality, as well as change points and anomalies using state space modeling. Due to the enhancement of model description capability, the results of model prediction and abnormal and change points detection are mutually improved.
- To solve the resultant optimization problem, an effective algorithm based on approximate inference using Markov chain Monte Carlo (MCMC) is proposed with theoretical guaranteed forecasting paths.
- Our proposed method outperforms the state-of-the-art methods in time series forecasting in presence of change points and anomalies, and detects change points and anomalies with high accuracy and low false discovery rate on both tasks, outperforming popular change point and anomaly detection methods. Our method is flexible to capture the structure of time series under various scenarios with any component combinations of trend, seasonality, change points and anomalies. Therefore our method can be applied in many settings in practice.

[1] We call it "robust" because it can accommodate "noises" much better since change points and anomalies can be viewed as a type of "noise".

2 MODEL OVERVIEW

State space time series model (Hangos et al., 2014) has been one of the most popular models in time series analysis. It is capable of fitting complicated time series structure including linear trend and seasonality. However, times series observed in real life are almost all prevailed with outliers. Change points, less in frequency but are still widely observed in real time series analysis. Unfortunately, both structures are ignored in the classic state space time series model. In the section, we aim to address this issue by introducing a novel state space time series model.

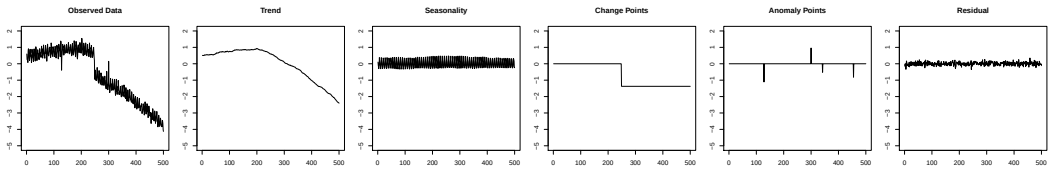

Figure 1: Demostration of Decompositions.

Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be a sequence of time series observations with length n. The ultimate goal is to forecast $(y_{n+1}, y_{n+2}, \ldots)$. The accuracy in forecasting lies in a successful decomposition of $\boldsymbol{y}$ into existing components. Apart from the residuals, we assume the time series is composed by trend, seasonality, change points and anomaly points. In a nutshell, we have an additive model with

$$\text{time series} \;=\; \text{trend} \;+\; \text{seasonality} \;+\; \text{change point} \;+\; \text{anomaly point} \;+\; \text{residual}.$$

Figure 1 provides a demonstration of desired decomposition of time series. In Figure 1, the left panel shows the observed time series. And it can be decomposed into the remaining five panels. The shift in the change point panel shows where the change point lies. And the spikes in the last panel reveals the anomaly points.

As the classical state space model, we have observation equation and transition equations to model $\boldsymbol{y}$ and hidden variables. We use $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$ to model trend, and use $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ to model seasonality. We use a binary vector $\boldsymbol{z^a} = (z_1^a, z_2^a, \ldots, z_n^a)$ to indicate anomaly points. Then we have

$$\text{Observation equation: } y_t = \mu_t + \gamma_t + \begin{cases} \epsilon_t, \text{ if } z_t^a = 0 \\ o_t, \text{ if } z_t^a = 1 \end{cases} . \tag{1}$$

The deviation between the observation y_t and its "mean" $\mu_t + \gamma_t$ is modeled by ϵ_t and o_t, depending on the value of z_t^a. If $z_t^a = 1$, then y_t is an anomaly point; otherwise it is not. Distinguished from the residues $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$, the anomaly is captured by $\boldsymbol{o} = (o_1, o_2, \ldots, o_n)$ which has relative large magnitude.

The hidden state variable $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ have intrinsic structures. There are two transition equations, for trend and seasonality separately

$$\text{Transition Equations:} \quad \text{Trend:} \qquad \mu_t = \mu_{t-1} + \delta_{t-1} + \begin{cases} u_t, \text{ if } z_t^c = 0 \\ r_t, \text{ if } z_t^c = 1 \end{cases} , \tag{2}$$

$$\delta_t = \delta_{t-1} + v_t,$$

$$\text{Seasonality: } \gamma_t = -\sum_{s=1}^{S-1} \gamma_{t-s} + w_t. \tag{3}$$

In Equation (2), $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_n)$ can be viewed as the "slope" of the trend, measuring how fast the trend changes over time. The change point component is also incorporated in Equation (2) by a binary vector $\boldsymbol{z^c} = (z_1^c, z_2^c, \ldots, z_n^c)$. If $z_t^c = 1$, it means the t-th point is a change point, with μ_t differs from $\mu_{t-1} + \delta_{t-1}$ (which can be interpreted as the "momentum" from the previous status) by r_t; otherwise it is not a change point and they differ by u_t. We model the change points in a way such that $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)$ have larger magnitude compared $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$. The "slope" part $\boldsymbol{\delta}$ also has its own noise $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$.

A first look on Equation (2) may bring up with the question that it is not presented in an exactly the same way as shown in Figure 1. In Figure 1, the change points component is a step function, and it

is one of the five additive components along with trend, seasonality, anomaly points and residuals. Here we model the change point directly into the trend component. Though differing in formulation, they are equivalent to each other. We choose to model in as in Equation (2) due to simplicity, and its similarity with the definition of anomaly points in Equation (1).

The seasonality component is presented in Equation (3). Here S is the length of one season and $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ is the noise for seasonality. The seasonality component is assumed to have almost zero average in each season.

The observation equation and transition equations (i.e., Equation (1,2,3)) define how $\boldsymbol{y}$ is generated from all the hidden variables including change points and anomaly points. We continue to explore this new model, under a Bayesian framework.

3 BAYESIAN FRAMEWORK

Bayesian methods are widely used in many data analysis fields. It is easy to implement and interpret, and it also has the ability to produce posterior distribution. The Bayesian method on state space time series model has been investigated in Scott & Varian (2014); Brodersen et al. (2015). In this section, we also consider Bayesian framework for our novel state space time series model. We assume all the noises are normally distributed

$$\{\epsilon_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad \{o_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_o^2), \quad \{u_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_u^2),$$
$$\{r_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_r^2), \quad \{v_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_v^2), \quad \{w_t\}_{t=1}^n \overset{iid}{\sim} \mathcal{N}(0, \sigma_w^2),$$

where $\sigma_\epsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w$ are parameters for standard deviation. As binary vectors, a natural choice is to model anomaly point indicator $\boldsymbol{z^a}$ and change point indicator $\boldsymbol{z^c}$ to the model them as Bernoulli random variables

$$\{z_t^a\}_{t=1}^n \overset{iid}{\sim} \text{Ber}(p_a), \quad \{z_t^c\}_{t=1}^n \overset{iid}{\sim} \text{Ber}(p_c),$$

where p_a, p_c are probabilities for each point to be an anomaly or change point.

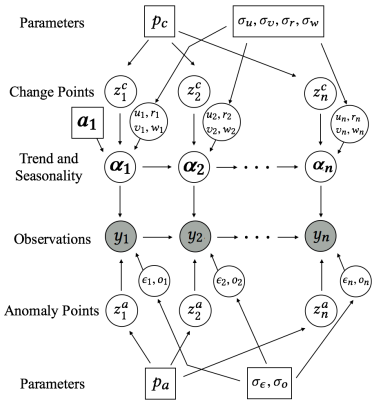

Figure 2: Graphical presentation of our model. Note that $\boldsymbol{y}$ is observed, highlighted by gray background, distinguished from all the remaining ones that are hidden. Among the hidden ones, squares indicate fixed parameters, and circles indicate random variables.

For simplicity, we denote $\boldsymbol{\alpha_t} = (\mu_t, \delta_t, \gamma_t, \gamma_{t-1}, \ldots, \gamma_{t-(S-2)})$ to include the main hidden variables (except z_t^a and z_t^c) in the transition equations. All the $\boldsymbol{\alpha_t}$ are well defined and can be generated from the previous status, except $\boldsymbol{\alpha_1}$. We denote $\boldsymbol{a_1}$ to be the parameter for $\boldsymbol{\alpha_1}$, which can be interpreted as the "mean" for $\boldsymbol{\alpha_1}$.

With Bayesian framework, we are able to represent our model graphically as in Figure 2. As shown in Figure 2, the only observations are $\boldsymbol{y}$ and all the others are hidden. In this paper, we assume there is no additional information on all the hidden states. If we have some prior information, for example, some points are more likely to be change points, then our model can be easily modified to incorporate such information, by using proper prior.

In Figure 2, we use squares and circles to classify unknown variables. Despite all being unknown, they actually behave differently according to their own functionality. For those in squares, they behave like turning parameters. Once they are initialized or given, those in circles behaves like latent variables. We call the former "parameters" and the latter "latent variable", as listed in Table 1.

Table 1: Two Categories for Hidden Variables

Category	Hidden Variable	Definition
Latent Variable	$\boldsymbol{\alpha} = (\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \ldots, \boldsymbol{\alpha_n})$	Trend and seasonality
	$\boldsymbol{z} = (\boldsymbol{z^a}, \boldsymbol{z^c})$	Anomaly and change points
Parameter	$\boldsymbol{a_1}$	The "mean" for the initial trend and seasonality
	$\boldsymbol{p} = (p_a, p_c)$	Probabilities for each point to be anomaly or change point
	$\boldsymbol{\sigma} = (\sigma_\epsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w)$	Standard deviation

The discrepancy between these two categories is clearly captured by the joint likelihood function. From Figure 2, the joint distribution (i.e., the likelihood function) can be written down explicitly as

$$
\begin{aligned}
& L_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{z}) \qquad (4)\\
&= \prod_{\{t:z_t^a=0\}} g(y_t - \mu_t - \gamma_t, \sigma_\epsilon) \times \prod_{\{t:z_t^a=1\}} g(y_t - \mu_t - \gamma_t, \sigma_o) \times \prod_{\{t:z_t^c=0\}} g(\mu_t - \mu_{t-1} - \delta_{t-1}, \sigma_u)\\
&\quad \times \prod_{\{t:z_t^c=1\}} g(\mu_t - \mu_{t-1} - \delta_{t-1}, \sigma_r) \times \prod_{t=1}^{n} g(\delta_t - \delta_{t-1}, \sigma_v) \times \prod_{t=1}^{n} g(-\sum_{s=1}^{S-1} \gamma_{t-s}, \sigma_v) \times \prod_{i=1}^{n} (p_a)^{z_t^a}(1-p_a)^{1-z_t^a}(p_c)^{z_t^c}(1-p_c)^{1-z_t^c},
\end{aligned}
$$

where $g(x_1, x_2) = \frac{1}{\sqrt{2\pi}x_2} \exp\left(-x_1^2/(2x_2^2)\right)$ is the density function for normal distribution with mean x_1 and standard deviation x_2. Here we slightly abuse the notation by using $\mu_0, \delta_0, \gamma_0, \gamma_{-1}, \ldots, \gamma_{2-S}$, which are actually the corresponding coordinates of $\boldsymbol{a_1}$.

As long with other probabilistic graphical models, our model can also be viewed as a generative model. Given the parameters $\boldsymbol{a_1}, \boldsymbol{p}, \boldsymbol{\sigma}$, we are able to generate time series. We present the generative procedure as follows.

Algorithm 1: Generative Procedure

Input: Parameters $\boldsymbol{a_1}, \boldsymbol{\sigma} = (\sigma_\epsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w)$ and p_a, p_c, length of time series to generate m
Output: Time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$

1 Generate the indexes where anomalies or change points occur

$$\{z_t^a\}_{t=1}^n \overset{iid}{\sim} \text{Ber}(p_a), \quad \{z_t^c\}_{t=1}^n \overset{iid}{\sim} \text{Ber}(p_c);$$

2 Generate all the noises $\boldsymbol{\epsilon}, \boldsymbol{o}, \boldsymbol{u}, \boldsymbol{r}, \boldsymbol{v}, \boldsymbol{w}$ as independent normal random variables with mean zero and standard deviation $\sigma_\epsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w$ respectively;
3 Generate $\{\boldsymbol{\alpha_t}\}_{t=1}^m$ sequentially by the transition functions in Equation (2) and (3);
4 Generate time series $\{y_t\}_{t=1}^m$ by the observation function in Equation (1).

4 INFERENCE

This section is about inferring unknown variables from $\boldsymbol{y}$, given the Bayesian setting described in the previous section. The main framework here is to sequentially update each hidden variable by fixing the remaining ones. As stated in the previous section, there are two different categories of unknown variables. Different update schemes need to be used due to the difference in their functionality. For the latent variables, we implement Markov chain Monte Carlo (MCMC) for inference. Particular, we use Gibbs sampler. We will elaborate the details of updates in the following sections.

4.1 UPDATES ON TREND AND SEASONALITY

In this section, we focus on updating $\boldsymbol{\alpha}$ assuming all the other hidden variables are given and fixed. The essence of Gibbs sampler is to obtain posterior distribution $p_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{\alpha}|\boldsymbol{y},\boldsymbol{z})$. This can be achieved by a combination of Kalman filter, Kalman smoothing and the so-called "fake-path" trick. We provide some intuitive explanation here and refer the readers to Durbin & Koopman (2012) for detailed implementation.

Kalman filter and Kalman smoothing are classic algorithms in signal processing and pattern recolonization for Bayesian inference. It is well related to other algorithms especially message passing algorithm. Kalman filter collects information forwards to obtain $\mathbb{E}(\boldsymbol{\alpha_t}|y_1, y_2, \ldots, y_t)$; while Kalman smoothing distribute information backwards to achieve $\mathbb{E}(\boldsymbol{\alpha_t}|\boldsymbol{y})$.

However, the combination of Kalman filter and Kalman smoothing is not enough, as it only gives the the expectations of marginal distributions $\{\mathbb{E}(\boldsymbol{\alpha_t}|\boldsymbol{y})\}_{t=1}^n$, instead of the joint distribution required for Gibbs sampler. To address this issue, we can use the "fake-path" trick described in Brodersen et al. (2015); Durbin & Koopman (2012). The main idea underlying this trick lies on the fact that the covariance structure of $p(\boldsymbol{\alpha_t}|\boldsymbol{y})$ is not dependent on the means. If we are able to obtain the covariance by some other way, then we can add it up with $\{\mathbb{E}(\boldsymbol{\alpha_t}|\boldsymbol{y})\}_{t=1}^n$ to obtain a sample from $p(\boldsymbol{\alpha}|\boldsymbol{y})$. This trick involves three steps. Note that all the other hidden variables $\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{\sigma}$ are given.

1. Pick some vector $\tilde{\boldsymbol{a}}_\mathbf{1}$, and generate a sequence of time series $\tilde{\boldsymbol{y}}$ from it by Algorithm 1. In this way, we also observe $\tilde{\boldsymbol{\alpha}}$.
2. Obtain $\{\mathbb{E}(\tilde{\boldsymbol{\alpha}}_\boldsymbol{t}|\tilde{\boldsymbol{y}})\}_{t=1}^n$ from $\tilde{\boldsymbol{y}}$ by Kalman filter and Kalman smoothing.
3. We use $\{\tilde{\boldsymbol{\alpha}}_\boldsymbol{t} - \mathbb{E}(\tilde{\boldsymbol{\alpha}}_\boldsymbol{t}|\tilde{\boldsymbol{y}}) + \mathbb{E}(\boldsymbol{\alpha_t}|\boldsymbol{y})\}_{t=1}^n$ as our sampling from the conditional distribution.

4.2 CHANGE POINT AND ANOMALY DETECTION

In this section, we update $\boldsymbol{z}$ by Gibbs sampler, assuming $\boldsymbol{\alpha}, \boldsymbol{a_1}, \boldsymbol{p}, \boldsymbol{\sigma}$ are all given and fixed. We need to obtain the conditional distribution $p_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\alpha})$. Note that in the graphical model described in Section 2, $\{z_t^a\}_{t=1}^n$ and $\{z_t^c\}_{i=1}^n$ are all Bernoulli random variables and independent of each other. Then the conditional distribution $p_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{\alpha})$ can also be decomposed into product of Bernoulli density functions. In other words, conditioned on $\boldsymbol{y}, \boldsymbol{\alpha}$, $\{z_t^a\}_{t=1}^n$ and $\{z_t^c\}_{i=1}^n$ are still independent Bernoulli random variables, but possibly with different success probabilities. Thus, we can take the calculation point by point. For example, for the anomaly detection for the t-th point, we have

$$z_t^a = 0 : y_t - \mu_t - \gamma_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$$
$$z_t^a = 1 : y_t - \mu_t - \gamma_t \sim \mathcal{N}(0, \sigma_o^2).$$

And the prior on z_t^a is $\mathbb{P}(z_t^a = 1) = p_a$ and $\mathbb{P}(z_t^a = 0) = p_1$. Let $p_t^a = \mathbb{P}(z_t^a = 1|\boldsymbol{y}, \boldsymbol{\alpha})$. Directly calculation leads to

$$p_t^a = \frac{\frac{p_a}{\sigma_o}\exp\left[-\frac{(y_t-\mu_t-\gamma_t)^2}{2\sigma_o^2}\right]}{\frac{1-p_a}{\sigma_\epsilon}\exp\left[-\frac{(y_t-\mu_t-\gamma_t)^2}{2\sigma_\epsilon^2}\right] + \frac{p_a}{\sigma_o}\exp\left[-\frac{(y_t-\mu_t-\gamma_t)^2}{2\sigma_o^2}\right]}. \tag{5}$$

This equality holds for all $t = 1, 2, \ldots, n$. Similarly for change point detection, let $p_t^c = \mathbb{P}(z_t^c = 1|\boldsymbol{y}, \boldsymbol{\alpha})$, and we have

$$p_t^c = \frac{\frac{p_c}{\sigma_r}\exp\left[-\frac{(\mu_t-\mu_{t-1}-\delta_{t-1})^2}{2\sigma_r^2}\right]}{\frac{1-p_c}{\sigma_u}\exp\left[-\frac{(\mu_t-\mu_{t-1}-\delta_{t-1})^2}{2\sigma_u^2}\right] + \frac{p_c}{\sigma_r}\exp\left[-\frac{(\mu_t-\mu_{t-1}-\delta_{t-1})^2}{2\sigma_r^2}\right]}. \tag{6}$$

As mentioned above, all the coordinates in $\boldsymbol{z}$ are still independent Bernoulli random variables conditioned on $\boldsymbol{y}, \boldsymbol{\alpha}$. Thus, for Gibbs sampler, we can generate $\boldsymbol{z}$ by sampling independently with

$$\{z_t^a\}_{t=1}^n \sim \text{Ber}(p_t^a), \quad \{z_t^c\}_{t=1}^n \sim \text{Ber}(p_t^c).$$

For change point detection here, we have an additional segment control step. After obtaining $\{z_t^c\}_{t=1}^n$ as mentioned above, we need to make sure that the change points detected satisfy some additional requirement on the length of segment among two consecutive change points. This issue

arises from the ambiguity between the definitions of change point and anomaly points. For example, consider a time series with value $(0, 0, 0, 0, 1, 1, 1, 0, 0, 0)$. We can view it with two change points, one increases the trend by 1 and the other decreases it by 1. Alternatively, we can also argue the three 1s in this time series are anomalies, though next to each other. One way to address this ambiguity is by defining the minimum length of segment (denoted as ℓ). In this toy example, if we set the minimum length to be 4, then they are anomaly points; if we set it to be 3, then we regard them to be change points. But a more complicated criterion is needed than using minimum length as the time series usually own much more complex structure than this toy example. Consider time series $(0, 0, 0, 0, -1, -1, 1, 1, 1, 1)$ and the minimum time series parameter $\ell = 3$. It is reasonable to view it with one change point with increment 1, and the two -1s should be regarded as anomalies. As a combination of all these factors, we propose the following segment control method. A default value for the parameter ℓ is the length of seasonality, i.e., $\ell = S$.

Algorithm 2: Segment control on change points

Input: change point binary vector $\boldsymbol{z^c}$,trend $\boldsymbol{\mu}$, standard deviation for outliers σ_r, change point minimum segment ℓ

Output: change point binary vector $\boldsymbol{z^c}$

Denote $t_1 < t_2 < \ldots$ to be all the indexes such that $z^c_{t_i} = 1$;

while *there exists* i *such that* $|t_{i+1} - t_i| < \ell$ **do**

  Check if $|\mu_{t_i - 1} - \mu_{t_{i+1}+1}| \leq \sigma_r/2$. If so, exclude both them from change points by setting $z^c_{t_i} = z^c_{t_{i+1}} = 0$. Otherwise, randomly exclude one of them by setting the corresponding coordinate in $\boldsymbol{z^c}$ to be 0;

  Update all the indexes of change points in $\boldsymbol{z^c}$.

end

4.3 Initialization and Updates on Parameters

The parameters $\boldsymbol{\sigma}, \boldsymbol{a_1}$ and p need both initialization and update. We have different initializations and update schemes for each of them.

For all the standard deviations, once we obtain $\boldsymbol{\alpha}$ and $\boldsymbol{z}$, we update them by taking the empirical standard deviation correspondingly. For σ_δ and σ_γ, the calculation is straightforward as they only involve $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ respectively. For $\sigma_\epsilon, \sigma_o, \sigma_u$ and σ_r, it is a bit more involved due to $\boldsymbol{z}$. Nevertheless, we can obtain the following update equations for all of them:

$$\sigma_\epsilon = \sqrt{\sum_{\{t: z^a_t = 0\}} \frac{(y_t - \mu_t - \gamma_t)^2}{|\{t : z^a_t = 0\}|}}, \sigma_o = \sqrt{\sum_{\{t: z^a_t = 1\}} \frac{(y_t - \mu_t - \gamma_t)^2}{|\{t : z^a_t = 1\}|}}, \sigma_u = \sqrt{\sum_{\{t: z^c_t = 0\}} \frac{(\mu_t - \mu_{t-1} - \delta_{t-1})^2}{|\{t : z^c_t = 0\}|}}, \quad (7)$$

$$\sigma_r = \sqrt{\sum_{\{t: z^c_t = 1\}} \frac{(\mu_t - \mu_{t-1} - \delta_{t-1})^2}{|\{t : z^c_t = 0\}|}}, \sigma_\delta = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (\delta_t - \delta_{t-1})^2}, \sigma_\gamma = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (\sum_{s=0}^{S-1} \gamma_{t-s})^2}. \quad (8)$$

Note that in some iterations, when there is no change point or anomaly detected in $\boldsymbol{z}$, then the updates above for σ_o, σ_r are not well-defined. In those cases, we simply let them remain the same. To initialize $\boldsymbol{\sigma}$, we let them all equal to the standard deviation of $\boldsymbol{y}$.

For $\boldsymbol{a_1}$, we initialize it by letting its first coordinate to be equal to the average of $y_1, y_2, \ldots, y_S$, and all the remaining coordinates to be equal to 0. Since $\boldsymbol{a_1}$ can be interpreted as the mean vector of α_1, in this way the trend is initialized to be matched up with average of the first season, and the slope and seasonality are initialized to be equal to 0. We update a_1 by using information of $\boldsymbol{\alpha}$. We let the first two coordinates (trend and slope) of $\boldsymbol{a_1}$ to be equal to those of $\boldsymbol{\alpha_1}$, and we let the remaining coordinates (seasonality) of $\boldsymbol{a_1}$ to be equal to those of $\boldsymbol{\alpha_{S+1}}$. The reason why we do not let $\boldsymbol{a_1}$ to be equal to $\boldsymbol{\alpha_1}$ entirely is due to the consideration on convergence and robustness. Since we initialize the seasonality part in $\boldsymbol{a_1}$ as 0, it will remain 0 if we let $\boldsymbol{a_1}$ equals $\boldsymbol{\alpha_1}$ entirely (due to the mechanism how we update $\boldsymbol{\alpha_1}$ as described in Section 4.1. We can avoid such trouble via using $\boldsymbol{\alpha_{S+1}}$.

For $\boldsymbol{p}$, we initialize them to be equal to $1/n$. If we have additional information on the number of change points or anomaly points, we can initiate them with different values, for example, $0.1/n$, or $10/n$. We can update $\boldsymbol{p}$ after obtaining $\boldsymbol{z}$, but we choose not to, also for the sake of robustness. In the early iterations when the algorithm is far from convergence, it is highly possible that $\boldsymbol{z^a}$ or $\boldsymbol{z^c}$

may turn out to be all 0. If we update $\boldsymbol{p}$, say, by taking the proportion of change point or anomaly points in $\boldsymbol{z}$. Then p_a or p_c might be 0, and it may get stuck in 0 in the remaining iterations.

5 FORECASTING

Once we infer all the latent variables $\boldsymbol{\alpha}, \boldsymbol{z}$ and tune all the parameters $\boldsymbol{p}, \boldsymbol{a_1}, \boldsymbol{\sigma}$, we are able to forecast the future time series $\boldsymbol{y}_{\text{future}}$. From the graphical model described in Section 3, the future forecasting only involves $\boldsymbol{\alpha_n}$ instead of the whole $\boldsymbol{\alpha}$. Note that we assume that there exists no change point and anomaly point in the future. This is reasonable as in most cases we have no additional information on the future time series. Given $\boldsymbol{\alpha_n}$ and $\boldsymbol{\sigma}$ we can use our predictive procedures (i.e., Algorithm 1) to generate future time series $\boldsymbol{y}_{\text{future}}$. We can further integrate out $\boldsymbol{\alpha_n}$ to have the posterior predictive distribution as $p_\sigma(\boldsymbol{y}_{\text{future}}|\boldsymbol{y})$.

The forecasting on future time series is not deterministic. There are two sources for the randomness in $\boldsymbol{y}_{\text{future}}$. One comes from the inference of $\boldsymbol{\alpha_n}$ (and also $\boldsymbol{\sigma}$) from $\boldsymbol{y}$. Under the Bayesian framework in Section 3, we have a posterior distribution over $\boldsymbol{\alpha_n}$ rather than a single point estimation. The second one comes from the forecasting function itself. The forecasting involves intrinsic noise like ϵ_t, u_t, v_t and w_t. Thus, the predictive density function $p_\sigma(\boldsymbol{y}_{\text{future}}|\boldsymbol{y}, \boldsymbol{\alpha_n})$ will lead to different path even with fixed $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha_n}$. In this way we are able to obtain distribution and predictive interval for forecasting. We also suggest to take the average of multiple forecasting paths, as the posterior mean for the forecasting.

The average of multiple forecasting paths (denoted as $\bar{\boldsymbol{y}}_{\text{future}}$), if the number of paths is large enough, always takes the form as a combination of linear trend and seasonality. This can be observed in both our synthesis data (Section 7) and real data analysis (Section 8). This seems to be surprising at the first glance, but makes some sense intuitively. Under our assumption, we have no information on the future, and thus a reliable way to forecast the future is to use the information collected at the end of observed time series, i.e., trend μ_n, slope δ_n and seasonality structure. Theorem 1 gives mathematical explanation of the linearity of $\bar{\boldsymbol{y}}_{\text{future}}$, in both mean and standard deviation.

Theorem 1. *Let N be the number of future time series paths we generate from Algorithm 1). Let m be the number of points we are going to forecast. Denote $\{y_{n+j}^{(1)}\}_{j=1}^m, \{y_{n+j}^{(2)}\}_{j=1}^m, \ldots, \{y_{n+j}^{(N)}\}_{j=1}^m$ to be the future paths. Define $\bar{\boldsymbol{y}}_{future} = (\bar{y}_{n+1}, \bar{y}_{n+2}, \ldots, \bar{y}_{n+m})$ to be the average such that*

$$\bar{y}_{n+j} = \frac{1}{N}\sum_{i=1}^{N} y_{n+j}^{(i)}.$$

Then for all $j = 1, 2, \ldots, N$, we have $\bar{y}_{n+j}$ as a normal distribution with mean and variance as

$$\mathbb{E}[\bar{y}_{n+j}] = \mu_n + j\delta_n + \gamma_{n-S+(j \bmod S)}$$
$$Var\,[\bar{y}_{n+j}] = \frac{1}{N}\left(j(j+1)\sigma_v^2/2 + j(\sigma_u^2 + \sigma_w^2) + \sigma_\epsilon^2\right).$$

Consequently, for all $j = 1, 2, \ldots, m$, $\mathbb{E}[\bar{y}_{n+j}]$ is in a linear form with respect to j, and the standard deviation of $\bar{y}_{n+j}$ also takes a approximately linear form with respect to j.

Proof. Recall that $\boldsymbol{\alpha_n}, \boldsymbol{\sigma}$ are given and fixed, and we assume there is no change point or anomaly in the future time series. The Equation (2) leads to $\delta_{n+j} = \delta_n + \sum_{l=1}^{j} v_{n+l}$, which implies that

$$\mu_{n+j} = \mu_n + j\delta_n + \sum_{l=1}^{j}(j+1-l)v_{n+l} + \sum_{l=1}^{j} u_{m+l}.$$

For the seasonality part, simple linear algebra together with Equation 3 leads to $\gamma_{n+j} = \gamma_{n-S+(j \bmod S)} + \sum_{l=1}^{j} w_{n+l}$. Thus,

$$\bar{y}_{n+j} = \frac{1}{N}\sum_{i=1}^{N}\left[\mu_n + j\delta_n + \gamma_{n-S+(j \bmod S)} + \sum_{l=1}^{j}(j+1-l)v_{n+l}^{(i)} + \sum_{l=1}^{j} u_{m+l}^{(i)} + \sum_{l=1}^{j} w_{n+l}^{(i)} + \epsilon_{n+j}^{(i)}\right].$$

Due to the independence and Gaussian distribution of all the noises, $\bar{y}_{n+j}$ is also normally distributed and its means and variance can be calculated accordingly. □

6 ALGORITHM

Our proposed method can be divided into three parts: initialization, inference, and forecasting. Section 4 and Section 5 provide detailed explanation and reasoning for each of them. We present a whole picture of our proposed methodology in Algorithm 3.

Algorithm 3: Proposed Algorithm

Input: Observed time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, seasonality length S, length of time series for forecasting m, number of predictive paths N, change point minimum segment l

Output: Change point detection $\boldsymbol{z^c}$, anomaly points $\boldsymbol{z^a}$, forecasting result $\boldsymbol{y}_{\text{future}} = (y_{n+1}, y_{n+1}, \ldots, y_{n+m})$ and its distribution or predictive intervals

Part I: *Initialization*;

Initialize $\sigma_\epsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w$ all with the empirical standard deviation of $\boldsymbol{y}$;

Initialize $\boldsymbol{a_1}$ such that its first coordinate equals to the average of $(y_1, y_2, \ldots, y_S)$ and all the remaining S coordinates with 0;

Initialize p_a and p_c by $1/n$. Then generate $\boldsymbol{z^a}$ and $\boldsymbol{z^c}$ as independent Bernoulli random variables with success probability p_a and p_c respectively;

Part II: *Inference*;

while *the likelihood function* $L_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{z})$ *not converges* **do**

  Infer $\boldsymbol{\alpha}$ by Kalman filter, Kalman smoothing and "fake-path" trick described in Section 4.1;

  Update $\boldsymbol{z^a}$ and $\boldsymbol{z^c}$ by sampling from

$$\{z_t^a\}_{t=1}^n \sim \text{Ber}(p_t^a), \quad \{z_t^c\}_{t=1}^n \sim \text{Ber}(p_t^c),$$

  where the success probability $\{p_t^a\}_{t=1}^n$ and $\{p_t^c\}_{t=1}^n$ are defined in Equation (5) and (6);

  Segment control on $\boldsymbol{z^c}$ by Algorithm 2;

  Update $\boldsymbol{\sigma}$ by Equation (7) to (8);

  Update $\boldsymbol{a_1}$ such that its first two coordinates equal to the those of $\boldsymbol{\alpha_1}$ and the remaining $(S-1)$ coordinates equals to those of $\boldsymbol{\alpha_{S+1}}$;

  Calculate the likelihood function $L_{\boldsymbol{a_1},\boldsymbol{p},\boldsymbol{\sigma}}(\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{z})$ given in Equation (4);

end

Part III: *Forecasting*;

With $\boldsymbol{a_n}$ and $\boldsymbol{\sigma}$, use the generate procedure in Algorithm 1 to generate future time series $\boldsymbol{y}_{\text{future}}$ with length m. Repeat the generative procedure to obtain multiple future paths $\boldsymbol{y}_{\text{future}}^{(1)}, \boldsymbol{y}_{\text{future}}^{(2)}, \ldots, \boldsymbol{y}_{\text{future}}^{(N)}$;

Combine all the predictive paths give the distribution for the future time series forecasting. If needed, calculate the point-wise quantile to obtain predictive intervals. Use the point-wise average as our final forecasting result.

It is worth mentioning that our proposed methodology is downward compatible with many simpler state space time series models. By letting $p_c = 0$, we assume there is no change point in the time series. By letting $p_a = 0$, we assume there is no anomaly point in the time series. If both p_c and p_a are set to be 0, then our model is reduced to the classic state space time series model. Also, the seasonality and slope can be removed from our model, if we know there exists no such structure in the data.

7 SIMULATION

In this section, we study the synthetic data generated from our model. We let $S = 7$ and provide values for σ and a_1. The change points and anomaly points are randomly generated. We use our generative procedure (Algorithm 1) to generate time series with total length 500 by fixed parameters. The first 350 points will be used as training set and the remaining 150 points will be used to evaluate the performance of forecasting.

When generating, we let the time series have weekly seasonality with $S = 7$. For $\boldsymbol{\sigma}$ we have $\sigma_\epsilon = 0.1, \sigma_u = 0.1, \sigma_v = 0.0004, \sigma_w = 0.01, \sigma_r = 1, \sigma_o = 4$. For $\boldsymbol{\alpha_1}$ we have value for μ as 20, value for δ as 0, and value for seasonality as $(1, 2, 4, -1, -3, -2)/10$. For $\boldsymbol{p}$ we have $p_c = 4/350$ and $p_a = 10/350$. Despite that, to make sure that at least one change point is in existence, we force $z^c_{330} = 1$ and $r_{330} = 2$. That is, for each time series we generate, its 330th point is a change point with the mean shifted up by 3. Also to be consistence with our assumption, we force $z^c_i = z^a_i = 0, \forall 351 \leq i \leq 500$ so there exists no change point or anomaly point in the testing part.

The top panel of Figure 3 shows one example of synthesis data. The blue line marks the separation between training and testing set. The blue dashed line indicates the locations for the change point, while the yellow dots indicate the positions of anomaly points. Also see Figure 3 for illustration on the results returned by implementing our proposed algorithm on the same dataset. The red line gives the fitting results in the first 350 points and forecasting results in the last 150 points. The change points detected are marked with vertical red dotted line, and the anomaly detected are flagged with purple squares. Figure 3 shows that on this dataset, our proposed algorithm yields perfect detection on both change points and anomaly points. In Figure 3, the gray part indicates the 90% predictive interval for forecasting.

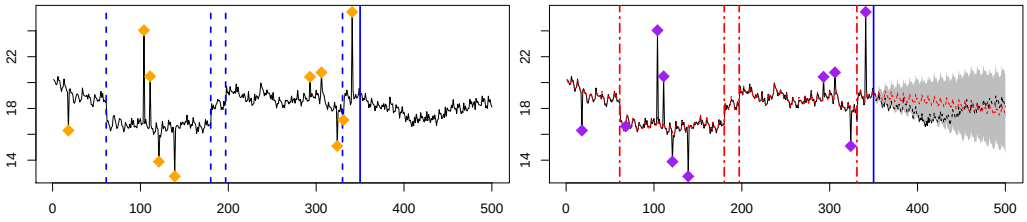

Figure 3: An example of synthesis data (left), and the result after applying our algorithm (right).

We run our generative model 100 times to produce 100 different time series, and implement multiply methods on each of them, and aggregate the results together for comparison. We include the following methodologies. For time series forecasting, we compare our method against Bayesian Structural Time Series (BSTS) (Scott & Varian, 2014; Brodersen et al., 2015)), Seasonal Decomposition of Time Series by Loess (STL) (Cleveland et al., 1990)), Seasonal ARIMA (Box et al., 2015), Holt–Winters (Holt, 2004), Exponential Smoothing State Space Model (ETS) (Hyndman et al., 2008)), and the Prophet R package by Taylor & Letham (2017). We evaluate the performances by mean absolute percentage error (MAPE), mean square error (MSE) and mean absolute error (MAE) on forecasting set. The mathematical definition of these three criterion is given as follows. Let $x_1, x_2, \ldots, x_n$ be the true value and $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ be the estimation or predictive values. Then we have

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - \hat{x}_i|}{x_i}, \text{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}, \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i|.$$

The comparison of our proposed algorithm and the aforementioned algorithms are included below in Table 2. As we mentioned in Section 6, our algorithm is downward compatible with the cases ignoring the existence of change point or anomaly, by setting $p_c = 0$ or $p_a = 0$. We also run proposed algorithm on the synthetic data with $p_c = 0$ (no change point), or $p_a = 0$ (no anomaly point), or $p_c = p_a = 0$ (no change and anomaly point), for the purpose of numeric comparison.

From Table 2 it turns out that our proposed algorithm achieves the best performance compared to other existing methods. Our proposed algorithm also performs better compared with the cases ignoring change point or anomaly point. This is a convincing evidence on the importance of incorporating both change point structure and anomaly point structure when modeling, for time series forecasting.

We also compare our proposed method with other existing change point detection methods and anomaly detection algorithm with respect to the performance of detections. We evaluate the performance by two criterions: True Positive Rate (TPR) and False Positive (FP). TPR measures the percentage of change points or anomalies to be correctly detected. FP count the number of points wrongly detected as change points or anomaly points. The mathematical definitions of TPR and FP are as follows. Let $(z_1, z_2, \ldots, z_n)$ be the true binary vector for change points or anomalies, and

Table 2: Comparison of methodologies on Forecasting

Methods	MAPE	MSE	MAE
Proposed	$\mathbf{0.041 \pm 0.027}$	$\mathbf{1.03 \pm 0.59}$	$\mathbf{0.89 \pm 0.53}$
Proposed ($p_a = 0$)	0.069 ± 0.068	1.71 ± 1.61	1.49 ± 1.44
Proposed ($p_c = 0$)	0.065 ± 0.058	1.67 ± 1.53	1.43 ± 1.35
Proposed ($p_a = 0, p_c = 0$)	0.084 ± 0.079	2.15 ± 2.00	1.87 ± 1.77
BSTS	0.162 ± 0.110	4.10 ± 2.81	3.59 ± 2.48
STL	0.047 ± 0.039	1.18 ± 1.06	1.03 ± 0.95
ARIMA	0.076 ± 0.050	1.88 ± 1.38	1.71 ± 1.24
Holt–Winters	0.093 ± 0.082	2.35 ± 2.06	2.05 ± 1.84
ETS	0.054 ± 0.042	1.37 ± 1.05	1.19 ± 0.94
Prophet	0.082 ± 0.055	2.06 ± 1.33	1.78 ± 1.16

$(\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n)$ are the estimated ones. Then

$$\text{TPR} = \frac{|\{i : z_i = 1, \hat{z}_i = 1\}|}{|\{i : z_i = 1\}|}, \quad \text{FP} = |\{i : z_i = 0, \hat{z}_i = 1\}|.$$

From the definition, we can see high TPR and low FP means the algorithm has better performance in detection.

The comparison on change point detection is shown in Table 3. We compare our results against three popular change point detection methods: Bayesian Change Point (BCP) (Barry & Hartigan, 1993), Change-Point (CP) (Killick & Eckley, 2014) and Breakout (twitter, 2017). From Table 3 our proposed method outperforms the most of the others by both TPR and FP. We have smaller TPR compared to CP, but we are better in FP.

Table 3: Comparison of Change Point Detection

Methods	TPR	FP
Proposed	$\mathbf{0.41 \pm 0.26}$	$\mathbf{0.34 \pm 0.57}$
Proposed ($p_a = 0$)	0.14 ± 0.21	0.26 ± 0.60
BCP	0.58 ± 0.22	29.84 ± 8.13
CP	0.29 ± 0.22	1.71 ± 1.15
Breakout	0.01 ± 0.04	0.53 ± 0.86

Table 4: Comparison of Anomaly Detection

Methods	TPR	FP
Proposed	$\mathbf{0.88 \pm 0.12}$	$\mathbf{0.58 \pm 0.96}$
Proposed ($p_c = 0$)	0.87 ± 0.12	2.56 ± 1.49
AnomalyDetection	0.32 ± 0.19	1.03 ± 1.94
RAD	0.88 ± 0.11	19.33 ± 3.58
tsoutlier	0.81 ± 0.14	4.76 ± 4.29

In Table 4, we also compare the performance of our algorithm on anomaly detection with three existing common anomaly detection methods: the AnomalyDetection package by Twitter (2017), RAD by Netflix (2017) and Tsoutlier by Chen & Liu (1993). The comparison is listed in Table 4. We can see our method also outperforms most of the others with respect to anomaly detection, by both TPR and FP. RAD has slightly better TPR but its FP is much worse compared with ours.

8 Real Data Analysis

In this section, we implement our proposed method on real-world datasets. We also compare its performance against other existing time series forecasting methodologies. We consider two datasets, one is a public data called Well-log dataset, and the other is an unpublished internet traffic dataset. The bottom panels of Figure 4 and Figure 5 give the result of our proposed algorithms. The blue line separates the training set and testing set. We use red line to show our fitting and forecasting result, vertical red dashed line to indicate change points and purple dots to indicate anomaly points. The gray part shows 90% predication interval.

8.1 Well-log Data

This dataset (Fearnhead & Clifford, 2003; JK & WJ, 1996) was collected when drilling a well. It measures the nuclear magnetic response, which provides geophysical information to analyze the structure of rock surrounding the well. This dataset is public and available online [2]. It has 4050

[2] `http://hips.seas.harvard.edu/files/well-log-2.dat`

points in total. We split it such that the first 3000 points are used as training set and last 1000 points are used to evaluate the forecasting performance.

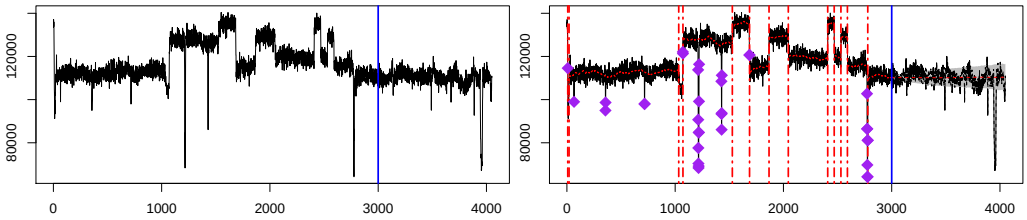

Figure 4: Well-log Data (left). The result of implementing our proposed algorithm (right).

From Figure 4, it is obvious that there exists no seasonality or slope structure in the dataset. This motivates us not to include these two components in our model. We implement our proposed algorithm without seasonality and slope, and compare the forecasting performance with other methods in Table 5. Our method outperforms BSTS, ARIMA, ETS and Prophet. However in Table 5 the performance can be slightly improved if we ignore the existence of anomaly points by letting $p_a = 0$. This may be caused by model mis-specification as the data may not generated in a way not entirely captured by our model. Nevertheless, the performances of our method considering anomaly points or not, are comparable to each other.

Table 5: Comparison of Forecasting in Well-log Data

Methods	MAPE	MSE	MAE
Proposed	**0.031**	**5296**	**3120**
Proposed ($p_a = 0$)	0.029	5252	2957
Proposed ($p_c = 0$)	0.033	5434	3409
Proposed ($p_a = 0, p_c = 0$)	0.038	5703	3908
BSTS	0.250	32030	27210
ARIMA	0.084	10480	8738
ETS	0.037	6071	3860
Prophet	0.159	19530	17480

In this dataset there is no ground-truth of change point and anomaly point on their locations or even existence. However, from bottom panel of Figure 4, there are some obvious changes in the sequence and they all successfully captured by our algorithm.

8.2 Internet Traffic Data

Our second real data is an Internet traffic data acquired from a major Tech company (see Figure 5). It is a daily traffic data, with seasonality $S = 7$. We use the first 800 observations as training set and evaluate the performance of forecasting on the remaining 265 points. The bottom panel of Figure 5 show the result from implementing our algorithm.

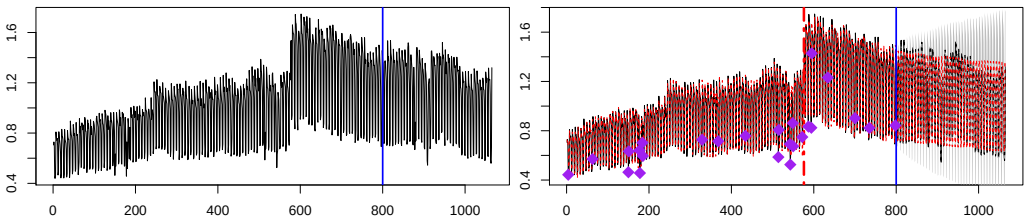

Figure 5: Internet Traffic Data (top); The result of implementing our proposed algorithm (bottom).

We also do the comparison of forecasting performance of our proposed algorithm together with other existing methods, shown in Table 6. We can also see that our algorithm outperforms all the other algorithms with respect to MAPE, MSE and MAE.

Table 6: Comparison of Forecasting in Internet traffic data

Methods	MAPE	MSE	MAE
Proposed	**0.0837**	**0.1216**	**0.08414**
Proposed ($p_a = 0$)	0.0838	0.1215	0.08320
Proposed ($p_c = 0$)	0.0934	0.1332	0.09296
Proposed ($p_a = 0, p_c = 0$)	0.0934	0.1366	0.09223
BSTS	0.2756	0.3087	0.27960
STL	0.1014	0.1258	0.09910
ARIMA	0.1409	0.1653	0.12580
Holt–Winters	0.2495	0.2739	0.25270
ETS	0.0893	0.1199	0.09362
Prophet	0.1015	0.1405	0.11450

From Figure 5 our proposed algorithm identifies one change point (the 576th point, indicated by the vertical red dashed line), which can be confirmed that this is exactly the only one change point existing in this time series caused by the change of counting methods, by some external information. Thus, we give the perfect change point detection in this Internet traffic data.

For this Internet traffic dataset, since we have ground-truth for change point, we can compare the performance of change point detection of different methodologies. BCP returns posterior distribution, which peaks in the the 576th point with posterior probability value 0.5. And it also returns with many other points with posterior probability value around 0.1. CP returns 4 change points, where the 576th point (the only true one) is one of them. Breakout returns 8 change points without including the 576th point. To sum up, our proposed method achieves the best change point detection in this real dataset.

9 Related Work

Parametric models are widely considered in econometric literature for time series forecasting, e.g. Jalles (2009), Commandeur et al. (2011), Gould et al. (2008), Harvey & Peters (1990), Harvey et al. (1998). The general procedure of decomposition method (using trend, seasonal and irregular components) for univariate structural time series modeling is discussed in Harvey & Peters (1990); a unified state space framework is proposed to handle any messy time series in Harvey et al. (1998); and the explicit modeling of both additive and multiplicative seasonalities in Gould et al. (2008); Jalles (2009). Although Kalman filter and MCMC-based approaches are used to sample posterior to estimate hidden components, the changing points and anomalies are not considered and processed in the above works. For example, the irregular component considered in Jalles (2009) is simply the noises. Commandeur et al. (2011) discusses the statistical software for state-space modeling which is designed for generic time series analytic and modeling, which cannot directly be used when changing point and anomalies are in existence. Our proposed approach shares similarity with the aforementioned papers as we have similar additive structure of components. However we are able to incorporate the change points and anomalies, two common structure widely observed in real data, into our model by using Bernoulli indicators. This is non-trivial, and cannot be handled by the aforementioned papers or their variants.

Non-parametric approaches are used for extraction of components from quasi-periodic time-series, e.g., the ensembles of weak detectors using non-parametric measurement are used in Artemov & Burnaev (2016) to detect change-points and anomalies, and the online decomposition algorithm based on per-component is adopted for change-point detection in Alexey Artemov (2015). Different from the above works, to handle the structural brakes and change-points, this paper presents the parametric approach for modeling anomalies and changing points by fitting them in the state-space framework using approximate inference for forecasting path prediction.

Different Bayesian approaches are proposed for change-point detection, e.g., Adams & MacKay (2007) performs Bayesian change point detection from online inference by generating the distribution estimation of the next unseen datum in the sequence given only data already observed, and the Bayesian Online CPD (BOCPD) algorithm proposed by Turner et al. (2009) performs online prediction using hidden variable given the underlying predictive model (UPM) and the hazard function.

Compared to the aforementioned models, our work differs in Bayesian modeling which samples posterior to estimate hidden components given the independent Bernoulli priors of changing point and anomalies.

10 CONCLUSION

We incorporate the change point structure and anomaly point structure into the classic space state time series model. We provide a Bayesian scheme for inference and time series forecasting. We compare the performance of our methodology and state-of-the-art methods on both synthetic data and real datasets. Our method performs the best with respect to forecasting, change point detection, and anomaly detection as well.

REFERENCES

Ryan P. Adams and David J.C. MacKay. Bayesian online changepoint detection. Cambridge, UK, 2007.

Andrey Lokot Alexey Artemov, Evgeny Burnaev. Nonparametric decomposition of quasi-periodic time series for change-point detection, 2015.

Alexey Artemov and Evgeny Burnaev. Detecting performance degradation of software-intensive systems in the presence of trends and long-range dependence. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain.*, pp. 29–36, 2016.

Daniel Barry and John A Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Kay H Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, Steven L Scott, et al. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9 (1):247–274, 2015.

Chung Chen and Lon-Mu Liu. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*, 88(421):284–297, 1993.

Robert B Cleveland, William S Cleveland, and Irma Terpenning. Stl: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3, 1990.

John H Cochrane. Time series for macroeconomics and finance. 2005.

Jacques Commandeur, Siem Koopman, and Marius Ooms. Statistical software for state space methods. *Journal of Statistical Software, Articles*, 41(1):1–18, 2011. ISSN 1548-7660.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

Paul Fearnhead and Peter Clifford. On-line inference for hidden markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899, 2003.

Phillip G. Gould, Anne B. Koehler, J. Keith Ord, Ralph D. Snyder, Rob J. Hyndman, and Farshid Vahid-Araghi. Forecasting time series with multiple seasonal patterns. *European Journal of Operational Research*, 191:207–222, 2008.

Katalin Hangos, Jzsef Bokor, and G Szederknyi. *Analysis and Control of Nonlinear Process Systems*. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London, 2014. doi: 10.1007/b97665.

A. C. Harvey and S. Peters. Estimation procedures for structural time series models. *Journal of Forecasting*, 9(2):89–108, 1990. ISSN 1099-131X.

Andrew Harvey, Siem Jan Koopman, and J Penzer. Messy time series: A unified approach. pp. 103–143, 13 1998.

Keith W Hipel and A Ian McLeod. *Time series modelling of water resources and environmental systems*, volume 45. Elsevier, 1994.

Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

Joao Jalles. Structural time series models and the kalman filter: a concise review. Feunl working paper series, Universidade Nova de Lisboa, Faculdade de Economia, 2009.

OR JK and F WJ. Numerical bayesian methods applied to signal processing, 1996.

Rebecca Killick and Idris Eckley. changepoint: An r package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19, 2014.

Netflix. Rad: Time series anomaly detection. 2017.

Takayuki Osogami, Hiroshi Kajino, and Taro Sekiyama. Bidirectional learning for time-series models with hidden units. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 2711–2720, 2017. URL http://proceedings.mlr.press/v70/osogami17a.html.

Steven L Scott and Hal R Varian. Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1-2):4–23, 2014.

Souhaib Ben Taieb, James W. Taylor, and Rob J. Hyndman. Coherent probabilistic forecasts for hierarchical time series. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3348–3357, 2017. URL http://proceedings.mlr.press/v70/taieb17a.html.

Karl Aberer Tao Lin, Tian Guo. Hybrid neural networks for learning the trend in time series. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2273–2279, 2017. doi: 10.24963/ijcai.2017/316. URL https://doi.org/10.24963/ijcai.2017/316.

S. J. Taylor and Letham. Prophet: forecasting at scale. 2017.

Ryan Turner, Yunus Saatci, and Carl Edward Rasmussen. Adaptive sequential Bayesian change point detection. In *Advances in Neural Information Processing Systems (NIPS): Temporal Segmentation Workshop*, 2009.

Twitter. Anomalydetection: Anomaly detection with r. 2017.

twitter. Breakout detection via robust e-statistics. 2017.

Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.