BAYESIAN TIME SERIES FORECASTING WITH CHANGE POINT AND ANOMALY DETECTION

A PREPRINT

Daria Demidova **Marina Gomtsyan**

October 26, 2018

ABSTRACT

Time series analysis has many applications in different fields, including finance, healthcare, and computer system monitoring. There is a variety of methods developed for time series analysis, however most of them perform purely when there are anomaly and change points in the data. The purpose of our work is to study the performance of a novel method for forecasting time series based on state space time series model and Bayesian framework. In addition, the method detects these points. To test the performance, we do experiments on simulated and real datasets.

***K*eywords** Bayesian time series · anomaly detection · change point detection

1 Model and Problem Formulation

There are many existing methodologies for time series analysis. One of the most general and widely used methods is Autoregresive Integrated Moving Average (ARIMA) which has different extensions, such as adding seasonality to the model (1). Another frequently used approach is the Holt-Winter's method (2) that uses exponential smoothing. Another method based on this approach is Exponential State Space smoothing (3), that decomposes time series into noise. State space models are used in other approaches as well, for example in Bayesian State Time Series (BSTS) (4) that captures the trend, seasonality, and similar other components of target times series. Some novel methods also include Deep Learning approaches such as Long-short term memory recurrent neural network (LSTM) based one by Lin $et.al.(5)$.

However most of the methods do not take into account the presence of anomaly and change points, which can lead to pure forecasting results. In addition, anomaly and change point detection can be considered as a separate problem to solve, since in many cases such points can contain crucial information. Authors of the method (6) that we are going to study in the scope of this project claim that their approach overcomes the limitations of the current widely used approaches whose performance is very sensitive to anomalies and change points. They developed a a state space time series model in the Bayesian framework that can simultaneously detect anomaly and change points and perform forecasting.

The model considers a sequence of time series $\mathbf{y} = (y_1, y_2, ..., y_n)$ of length n and the aim is to predict $(y_{n+1}, y_{n+2}, ...)$. As a classical state space equation, the model has observation equation

$$y_t = \mu_t + \gamma_t + z_t^a o_t + (1 - z_t^a)\varepsilon_t,$$

where $\mu = (\mu_1, \mu_2, ..., \mu_n)$ is trend, $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$ is seasonality of the model, and $\mathbf{z^a} = \{z_t^a\}_{t=1}^n \sim Ber(p_a)$ is an $i.i.d$ binary vector indicating anomaly points. μ and γ are hidden variables with transition equations

$$\mu_t = \mu_{t-1} + \delta_{t-1} + z_t^c r_t + (1 - z_t^c)u_t$$

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

for trend where δ can be viewed as the slope of trend and

$$\gamma_t = -\sum_{s=1}^{S-1} \gamma_{t-s} + w_t.$$

$\mathbf{z^a} = \{z_t^a\}_{t=1}^n \sim Ber(p_a)$ is an $i.i.d$ binary vector indicating change points. We assume all noises $o_t, \varepsilon_t, r_t, u_t, v_t, w_t$ are independent zeros mean normally distributed.

We denote $\{\alpha_\mathbf{t}\}_{t=1}^n = (\mu_t, \delta_t, \gamma_t, \dots, \gamma_{t-S+2})_{t=1}^n$ to include main hidden variables and $\{\mathbf{z_t}\}_{t=1}^n = \{(z_t^a, z_t^c)\}_{t=1}^n$, with parameters $a_1 = (\mu_o, \delta_0, \gamma_0, \dots, \gamma_{2-S})$, $p = (p_a, p_c)$, and $\sigma = (\sigma_\varepsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w)$, where σ's are standard deviations of noises $o_t, \varepsilon_t, r_t, u_t, v_t, w_t$.

2 Implementation

The implementation of method is about inferring unknown variables from $\mathbf{y}$, given the Bayesian setting described in the previous section. The idea is to sequentially update each hidden variable by fixing the remaining ones. Since 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.

For updating α we use Gibbs sampler to obtain posterior distribution $p_{\alpha_1,p,\sigma}(\alpha|\mathbf{y},\mathbf{z})$. This can be achieved by a combination of Kalman filter, Kalman smoothing and "fake-path" trick. Kalman filter forwards collected information to obtain $\mathbb{E}(\alpha_t|y_1, y_2, ..., y_t)$ while Kalman smoothing distributes information backwards to achieve $\mathbb{E}(\alpha_t|y)$. The purpose of "fake-path" trick is to obtain posterior distribution $p_{\alpha_1,p,\sigma(\alpha|y,z)}$. All hidden variables z, p, σ are given:

1. Pick some vector $\tilde{a}_1$ and generate a sequence of time series $\tilde{y}$ from it. We also observe $\tilde{\alpha}$.
2. Obtain $\{\mathbb{E}(\tilde{\alpha}_t|\tilde{y})\}_{t=1}^n$ from $\tilde{y}$ by Kalman filter and Kalman smoothing.
3. Use $\{\tilde{\alpha}_t - \mathbb{E}(\tilde{\alpha}_t|\tilde{y}) + \mathbb{E}(\alpha_t|y)\}_{t=1}^n$ as sampling distribution from the conditional distribution.

We update z by Gibbs sampler, assuming α, a_1, p, σ are all given and fixed. We need to obtain the conditional distribution $p_{\alpha_1,p,\sigma}(\mathbf{z}|\mathbf{y}, \alpha)$. We sample

$$z_t^a \sim p_{a_1,p,\sigma}(z_t^a|y, \alpha) = Ber(p_t^a) \quad z_t^c \sim p_{a_1,p,\sigma}(z_t^c|y, \alpha) = Ber(p_t^c),$$

where $\{p_t^a\}_{t=1}^n = p(z_t^a = 1|y, \alpha)$ and $\{p_t^c\}_{t=1}^n = p(z_t^c = 1|y, \alpha)$.

Having all update equations we can introduce the main algorithm, which looks as follows:

Part I: Initialization

1. Initialize $\sigma_\varepsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w$ with the empirical standard deviation.
2. Initialize a_1: $a_1[0] = \frac{1}{S}\sum\limits_{t=1}^{S} y_t$, $a_1[1:] = 0$.
3. Initialize $p_a = p_c = \frac{1}{n}$, $\{z_t^a\}_{t=1}^n \sim Ber(p_a)$, $\{z_t^c\}_{t=1}^n \sim Ber(p_c)$

Part II: Inference

while $L_{a_1,p,\sigma}(\mathbf{y}, \alpha, \mathbf{z})$ does not converges:

1. $\alpha \sim p_{a_1,p,\sigma}(\alpha|\mathbf{y},\mathbf{z})$ by Kalman filter, Kalman smoothing and “fake-path” trick
2. Update z_t^a and z_t^c by Gibbs sampler
3. Segment control on z_c: requirement on the length of segment among two consecutive change points
4. Using α and z, update σ by the empirical standard deviation
5. Update a_1. $a_1[:2] = \alpha_1[:2]$, $a_1[2:] = \alpha_{S+1}[2:]$
6. Calculate $L_{a_1,p,\sigma}(y, \alpha, z)$

Part III: Forecasting

1. With a_n and σ, generate future time series y_{future} with length m. Repeat generative procedure to obtain future paths $\mathbf{y_{future}}^{(1)}, \mathbf{y_{future}}^{(2)}, ..., \mathbf{y_{future}}^{(N)}$.
2. Combine all the predictive paths and give the distribution for the future time series forecasting. Calculate the point-wise quantile intervals.

The joint likelihood function can be calculated as

$$L_{a_1,p,\sigma}(y,\alpha,z) = \prod_{t:z_t^a=0} \mathcal{N}(y_t|\mu_t+\gamma_t,\sigma_\varepsilon)\cdot \prod_{t:z_t^a=1} \mathcal{N}(y_t|\mu_t+\gamma_t,\sigma_o)\cdot \prod_{t:z_t^c=0} \mathcal{N}(\mu_t|\mu_{t-1}+\delta_{t-1},\sigma_u)\cdot \prod_{t:z_t^c=1} \mathcal{N}(\mu_t|\mu_{t-1}+\delta_{t-1},\sigma_r)\cdot$$

$$\cdot \prod_{t=1}^{n} \mathcal{N}(\delta_t|\delta_{t-1},\sigma_v)\cdot \prod_{t=1}^{n} \mathcal{N}(\gamma_t|-\sum_{s=1}^{S-1}\gamma_{t-s},\sigma_w)\cdot \prod_{t=1}^{n}(p_a)^{z_t^a}(1-p_a)^{1-z_t^a}(p_c)^{z_t^c}(1-p_c)^{1-z_t^c}.$$

Segment control on change points is performed by the following procedure:

Denote $t_1 < t_2 <$ ... to be all the indexes such that $z_{t_i}^c = 1$

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

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

Finally, the generative procedure from which we obtain y_{future} is the following:

1. Generate the indexes of anomalies or change points occur
$$\{z_t^a\}_{t=1}^n \sim Ber(p_a), \quad \{z_t^c\}_{t=1}^n \sim Ber(p_c).$$
2. Gnerate $\varepsilon, o, u, r, v, w$ as independent normal r.v.'s with zero mean and standard deviations $\sigma_\varepsilon, \sigma_o, \sigma_u, \sigma_r, \sigma_v, \sigma_w$.
3. Generate $\{\alpha_t\}_{t=1}^m$ by transition functions.
4. Generate time series $\{\mathbf{y}_t\}_{t=1}^m$ by the observation function.

In the paper there are no details about Kalman Filter and Kalman Smoothing. Since the model contains change points and anomaly points, it is not standard. Thus we derived to be able to do the implementation, which can be found in Appendix.

The paper does not contain information about p for the derivation of $p_{a_1,p,\sigma}(\alpha|y,z)$. We tested both $p = (p^a, p^c)$ and $p = \{(p_t^a, p_t^c)\}_{t=1}^n$ and noticed, that the last one works better.

It is important to notice that there are several important misprints in the paper. For example, in $L_{a_1,p,\sigma}(y,\alpha,z)$ there should be $g(-\sum_{s=1}^{S}\gamma_{t-s}, \sigma_v)$.

3 Experiments and Results

For the first experiment we generated an artificial data using generative procedure and initial parameters described in the previous section. We noticed that results are very sensitive to the initial values of σ. If sigma is overestimated then Kalman filter and smoothing will oversmooth time series. When new time series is re-estimation based on this oversmoothed time series, σ is obtained to be bigger and the series even more oversmoothed. Also, we noticed that we should lower initial value of change point probability because of the segment control procedure.

For the simulated data, the algorithm founds most of change points and anomalies, but sometimes it created an anomaly point instead of a change point, or first created a set of anomaly points and after that created a change point ($i.e.$ the last change point).

In Figure 1 the red line is the initial time series, the blue one is the result of Kalman Filter, Kalman Smoothing and Fake Path Trick, and green lines are the expectation and confidence intervals for forecasting. The mean squared error (MSE) on predicted path is about 18.66.

We also observe that over the iterations of the the algorithm anomaly points remain the same, which means that there are not sensitive to the algorithm (Figure 2).

As the second experiment we used time series of a CPU workload of virtual machine. Results are shown in Figure 3. As it can be seen, anomaly and change points are not predicted and the confidence intervals are very big. We suppose that this pure performance is due to fact that we did not know the actual value of σ and as we observed for the simulated data, with big σ time series is oversmoothed, in the result we obtain bigger σ and end up with highly fluctuating series.

Our implementations can be accessed by clicking here.

Figure 1: Simulated time series

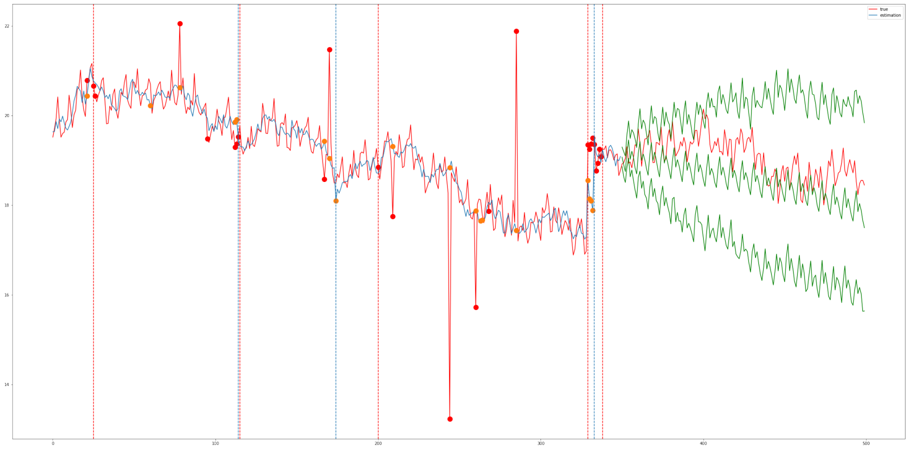


Figure 2: Anomaly points over iterations

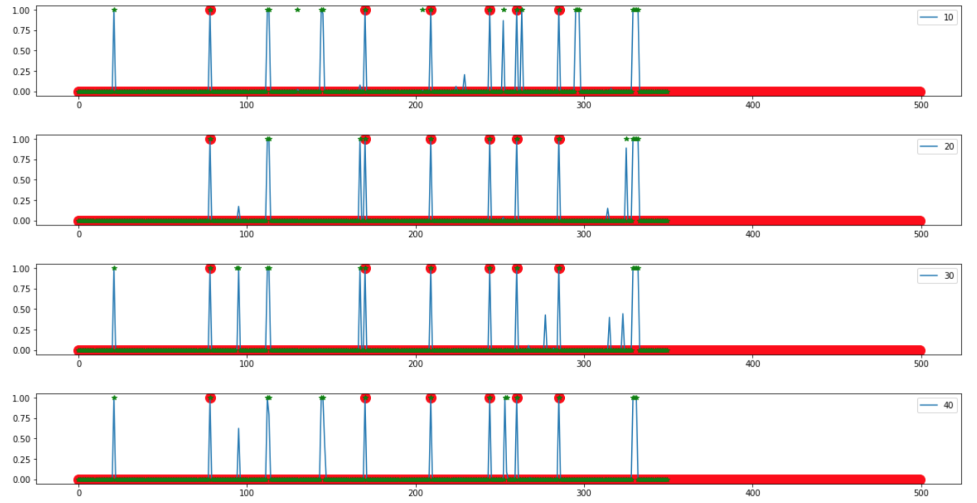
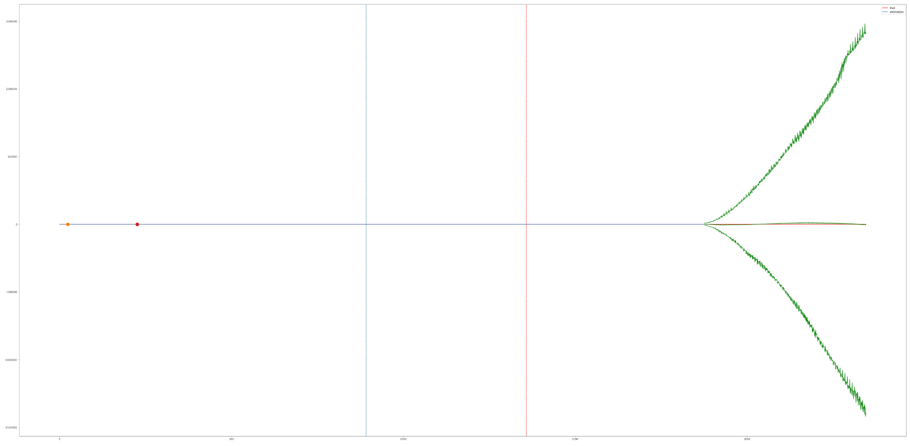


Figure 3: CPU workload data



4 Appendix

State space equations

$y_t = Z_t\alpha_t + A_t o_t + (1 - A_t)\varepsilon_t$

$\alpha_{t+1} = T_t\alpha_t + R_t C_t\eta_t + R_t(1 - C_t)\,\xi_t$

$o_t \sim \mathcal{N}(0, H_t^o)$, $\varepsilon_t \sim \mathcal{N}(0, H_t^\varepsilon)$, $\eta_t \sim \mathcal{N}(0, Q_t^\eta)$, $\xi_t \sim \mathcal{N}(0, Q_t^\xi)$

Kalman Filter

$a_{t|t} = \mathbb{E}(\alpha_t|Y_t)$, $a_{t+1} = \mathbb{E}(\alpha_{t+1}|Y_t)$

$P_{t|t} = Var(\alpha_t|Y_t)$, $P_{t+1} = Var(\alpha_{t+1}|Y_t)$

$v_t = y_t - \mathbb{E}(y_t|Y_{t-1}) = y_t - \mathbb{E}(Z_t\alpha_t + A_t o_t + (1 - A_t)\varepsilon_t|Y_{t-1}) = y_t - Z_t a_t$

$a_{t|t} = \mathbb{E}(\alpha_t|Y_t) = \mathbb{E}(\alpha_t|Y_{t-1}, v_t)$, $a_{t+1} = \mathbb{E}(\alpha_{t+1}|Y_t) = \mathbb{E}(\alpha_{t+1}|Y_{t-1}, v_t)$

Lemma 1

If $(x, y) = \mathcal{N}((\mu_x, \mu_y), (\,\Sigma\,)_{xx}\, \Sigma_{xy}$
$\Sigma_{xy}^T \Sigma_{yy})$, then $p(x|y) = \mathcal{N}(\mathbb{E}(x|y), Var(x|y))$,

$\mathbb{E}(x|y) = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)$, $Var(x|y) = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T$

Apply *Lemma 1* for $x = \alpha_t|Y_{t-1}$ and $y = v_t|Y_{t-1}$, then $a_{t|t} = \mathbb{E}(\alpha_t|T_{t-1}) + Cov(\alpha_t, v_t)[Var(v_t)]^{-1}v_t = a_t + P_t Z_t^T F_t^{-1} v$

$Cov(\alpha_t, v_t) = \mathbb{E}[\alpha_t(Z_t\alpha_t + A_t o_t + (1 - A_t)\varepsilon_t - Z_t a_t)^T|Y_{t-1}] = \mathbb{E}[\alpha_t(\alpha_t - a_t)^T Z_t^T|Y_{t-1}] = P_t Z_t^T$

$P_t = \mathbb{E}[\alpha_t(\alpha_t - a_t)^T|Y_{t-1}] = \mathbb{E}[(\alpha_t - a_t)(\alpha_t - a_t)^T|Y_{t-1}] + a_t\mathbb{E}[(\alpha_t - a_t)^T] = Var(\alpha_t|Y_{t-1})$

$F_t = Var(v_t|Y_{t-1}) = Var(Z_t\alpha_t + A_t o_t + (1 - A_t)\varepsilon_t - Z_t a_t|Y_{t-1})$

$F_t = \mathbb{E}(Z_t(\alpha_t - a_t)(\alpha_t - a_t)^T Z_t^T|Y_{t-1}) + \mathbb{E}[(A_t o_t + (1 - A_t)\varepsilon_t)(A_t o_t + (1 - A_t)\varepsilon_t)^T] = Z_t P_t Z_t^T + p_t^a H_t^o + [1 - p_t^a]H_t^\varepsilon$

$a_{t|t} = a_t + P_t Z^t F_t^{-1} v_t$

$P_{t|t} = Var(\alpha_t|Y_t) = Var(\alpha_t|Y_{t-1}, v_t) = Var(\alpha_t|Y_{t-1}) - Cov(\alpha_t, v_t)[Var(v_t)]^{-1}Cov(\alpha_t, v_t)^T = P_t - P_t Z_t^T F_t^{-1} Z_t P_t$

$a_{t+1} = \mathbb{E}(T_t\alpha_t + R_t C_t\eta_t + R_t(1 - C_t)\xi_t|Y_t) = T_t\mathbb{E}(\alpha_t|Y_t) = T_t a_{t|t} = T_t a_t + K_t v_t$

$K_t = T_t P_t Z^t F_t^{-1}$

$P_{t+1} = Var(T_t\alpha_t + R_t C_t\eta_t + R_t(1 - C_t)\xi_t|Y_t) = T_t Var(\alpha_t|Y_t)T_t^T + R_t[p_t^c Q_t^\eta + (1 - p_t^c)Q_t^\xi]R_t^T = T_t P_{t|t} T_t^T + R_t[p_t^c Q_t^\eta + (1 - p_t^c)Q_t^\xi]R_t^T = T_t P_t(T_t^T - Z_t^T F_t^{-1} Z_t P_t T_t^T) + R_t[p_t^c Q_t^\eta + (1 - p_t^c)Q_t^\xi]R_t^T$

We obatain Kalman Filter Recursion

$v_t = y_t - Z_t a_t, \qquad\qquad F_t = Z_t P_t Z_t^T + p_t^a H_t^o + [1 - p_t^a]H_t^\varepsilon$

$a_{t|t} = a_t + P_t Z^t F_t^{-1} v_t, \qquad a_{t+1} = T_t a_t + K_t v_t$

$P_{t|t} = P_t - P_t Z_t^T F_t^{-1} Z_t P_t, \quad P_{t+1} = T_t P_t(T_t^T - K_t Z_t) + R_t[p_t^c Q_t^\eta + (1 - p_t^c)Q_t^\xi]R_t^T$

References

[1] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung, (2015), *Time series analysis: forecasting and control*

[2] Peter R Winters, (1960) *Forecasting sales by exponentially weighted moving averages*

[3] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder, (2008), *Forecasting with exponential smoothing: the state space approach*

[4] Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, Steven L. Scott, (2015), *Inferring Causal Impact Using Bayesian Structural Time-Series Models*

[5] Karl Aberer Tao Lin, Tian Guo, (2017) *Hybrid neural networks for learning the trend in time series*

[6] Anderson Y. Zhang, Miao Lu, Deguang Kong, Jimmy Yang, (2018), *Bayesian Time Series Forecasting with Change Point and Anomaly Detection*