WHAT IS AN MBSTS?

AJ BARRY, ERIC ARAGUNDI, LUKE ARCENEAUX

ABSTRACT. For our project, we implement a Multivariate Bayesian Structural Time Series model to perform prediction of the stock prices of Apple and Microsoft, using google trends data as regressors. The use of a Bayesian approach has several advantages over a traditional model. Firstly, it provides us with a clear-cut method for feature selection of the predictor variables by using spike-slab regression. Additionally, the Bayesian approach yields more interpretable confidence intervals and a clear method for continuous updating of the model.

1. The Model

The Multivariate Bayesian Time Series model represents quite a departure from the models we have discussed in class. Firstly, it is a multivariate model. Secondly, it includes the Kalman filter, which would be a sufficient extension of the work we have done in class on its own. Thirdly, it incorporates a Bayesian methodology to our time series forecasting and feature selection. As such, understanding the model and its various components will require substantial exposition; thus, we will endeavour to make this as succinct as possible.

1.1. **Structural Time Series.** We will begin by providing a basic exposition of univariate structural time series. The structural time series model, also known as a state space model, assumes that the next observation in our time series is dependent on our underlying state space, rather than the previous observation. As an example, we consider the following model. Let y_t denote our observed time series, and let μ_t denote the underlying state of the model at time t. We can decompose a simple state space model (local level model) as follows:

- The observed data equation:

$$y_t = \mu_t + \epsilon_t \qquad\qquad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$

- The state space equation:

$$\mu_{t+1} = \mu_t + u_t \qquad\qquad u_t \sim N(0, \sigma^2 u_t)$$

Date: 2023-4-22.

As we can see, in this case our state space is simply white noise. However, our observed data contains an additional independent noise component. A more general form of this model can be expressed in the following definition 1.1.

Definition 1.1. [6][2.1] The General form of a structural time series model is:

- Observation equation:
$$y_t = Z_t^T \alpha_t + \epsilon_t \qquad \epsilon_t \sim N(0, H_t)$$
- Transition equation:
$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t \qquad \eta_t \sim N(0, Q_t)$$

For our purposes, our structural time series model will be defined in 1.2.

Definition 1.2. [6][2.1] Let m be the dimension of our time series y_t. The state space components $\mu_t, \tau_t, \omega_t, \xi_t$ of our model are defined as follows:

$$\mu_{t+1} = \mu_t + \delta_t + u_t \quad u_t \overset{iid}{\sim} N(0, \Sigma_\mu) \tag{1.1}$$

for δ being subject to the following updating rule:

$$\delta_{t+1} = \tilde{D} + \rho(\delta_t - \tilde{D}) + v_t \quad v_t \overset{iid}{\sim} N(0, \Sigma_\delta) \tag{1.2}$$

Here, ρ_t is an $m \times m$ diagonal matrix with $0 \leq \rho_{i,i} \leq 1$. We now define our seasonal component τ_t.

$$\tau_{t+1}(i) = \sum_{s=0}^{S_i-2} \tau_{t-s}^{(i)} + w_t^{(i)} \qquad w_t = [w_t^{(1)}, \dots, w_t^{(m)}]^T \overset{iid}{\sim} N(0, \Sigma_\tau) \tag{1.3}$$

For S_i denoting the number of seasons for $y_t^{(i)}$. Our cyclical component ω_t evolves according to the following:

$$\begin{aligned} \omega_{t+1} &= \varrho \cos(\Lambda) + \varrho \sin(\Lambda) \omega_t^* + \kappa_t & \kappa_t &\overset{iid}{\sim} N(0, \Sigma_\omega) \\ \omega_{t+1}^* &= -\varrho \cos(\Lambda) + \varrho \sin(\Lambda) \omega_t^* + \kappa_t^* & \kappa_t^* &\overset{iid}{\sim} N(0, \Sigma_\omega) \end{aligned} \tag{1.4}$$

For ϱ, $\sin(\Lambda)$, $\cos(\Lambda)$ being $m \times m$ diagonal matrices with entries $0 \leq \varrho_{i,i} \leq 1$, $\sin(\lambda_{i,i})$ and $\cos(\lambda_{i,i})$ for $\lambda_{i,i} = 2\pi/q_i$ and $0 < q_i < 2$. Our final component is our regression component with static coefficients

$$\xi_t^{(i)} = \beta_i^T x_t^{(i)}. \tag{1.5}$$

Combining all of our state representations we can write

$$y_t = \mu_t + \tau_t + \omega_t + \xi_t + \epsilon_t \quad \epsilon_t \overset{iid}{\sim} N(0, \Sigma_\epsilon) \tag{1.6}$$

More succinctly, we can combine all n iterations of our time series into the following matrix representation:

$$Y = M + T + W + X\beta + E \tag{1.7}$$

For $Y = [y_1, \ldots, y_n]$, and M, T, W, E defined analogously for $\mu_t, \tau_t, \omega_t, \epsilon_t$ respectively. X is a block diagonal matrix such that $X_{i,i}$ is an $n \times k_i$ of candidate predictors at each iteration for $y_t^{(i)}$ with $\sum_{i=1}^m k_i = K$ and β_i is a k_i dimensional vector.

Now that we have defined our model, we turn to the interpretation of the various model parameters. Our parameter μ_t is simply a generalization of the local linear trend model and as such μ_t can be thought of as the slope of the model at time t. Our parameter D represents the long term slope of the model, and our parameters $\rho_{i,i}$ represents the rate at which the local trend is updated. This parameter ρ thus allows us to tune the model so that it balances short term and long term information, in much the same way as an EWMA model would. If $\rho_{i,i} = 1$, then our corresponding slope becomes a random walk.

It is also important to denote the difference between the cyclical and seasonal components of the model. The cyclical component is a commonly used component to refer to the so called "Business Cycle", which is recurrent, but not precisely periodic, deviations around the long term path of the series. We may note that when our $\lambda_{i,i}$ components are equal to zero or π, the model devolves to an AR(1), and as such the ϱ terms should be strictly less than one for stationary purposes.

1.2. **Spike and Slab Regression.** We now discuss how feature selection is performed in MBSTS. Given an arbitrarily chosen set of predictors, it is natural to begin with the presumption that the majority of predictors are equal to zero. In fact, in frequentist statistics, this is explicitly our null hypothesis for the regression coefficients. In the Bayesian paradigm, we often represent this phenomenon by using Spike and Slab coefficients. In the matrix representation 1.7 of our model, we write our product $X\beta$ as the block diagonal matrix and vector

$$X = \begin{bmatrix} X_1 & 0 & \ldots & 0 \\ 0 & X_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & X_m \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} \tag{1.8}$$

for $\beta_i = [\beta_{i,1}, \ldots, \beta_{i,k_i}]^T$ and X_i being a $n \times k_i$ matrix of k_i candidate predictors for $y_t^{(i)}$ at time $t \in 1, \ldots, n$. It is important to note that defining our equations in matrix form as such permits us to use a different set of predictors for each component of our multivariate time series. We can therefore define $\gamma_{i,j}$ as an indicator variable taking 1 if $\beta_{i,j} = 1$. Thus, our spike prior is

$$\gamma \sim \prod_{i=1}^{m} \prod_{j=1}^{k_i} \pi_{i,j}^{\gamma_{i,j}} (1 - \pi_{i,j})^{1-\gamma_{i,j}} \tag{1.9}$$

where $\pi_{i,j}$ is the prior inclusion probability of the jth predictor for the ith target series. For our purposes, our prior will be to assume that all the $\pi_{i,j}$ are equal.

To specify our slab prior, we will assume that β, Σ_ϵ are prior independent (conditional on γ).

$$\begin{aligned} p(\beta, \Sigma_\epsilon, \gamma) &= p(\beta|\gamma)p(\Sigma_\epsilon|\gamma)p(\gamma) \\ \beta|\gamma &\sim N_K(b_\gamma, \kappa(X_\gamma^T X_\gamma)^{-1}) \\ \Sigma_\epsilon|\gamma &\sim IW(v_0, V_0). \end{aligned} \tag{1.10}$$

where κ is the number of observations worth of weight on the prior mean vector b_γ. To encode our prior so that the β's are equal to zero, we can simply set $b = 0$. Our other priors on Σ_ν, $\nu \in \{\mu, \tau, \omega, \delta\}$ can similarly be expressed as

$$\Sigma_\nu \sim IW(w_0, W_0). \tag{1.11}$$

1.3. **The Model Training Algorithm.** To fit the posterior distributions of our data, we will conduct MCMC simulation. We will use the `mbsts` package in R to fit our model. The package uses a composite algorithm to train the model, which cycles through the following steps:

Algorithm 1.3 (MBSTS Model Training)**.** *[6]*

(1) *Draw latent state* $\tilde{\alpha} = (\mu, \delta, \tau, \omega)$ *from* $p(\alpha|Y, \theta, \gamma, \Sigma_\epsilon, \beta)$ *using the forwards filtering, backwards sampling algorithm [1]:*
   (a) *Let* $w = [\epsilon_1, \eta_1, \ldots, \epsilon_n, \eta_n]$. *Draw a random vector* $\tilde{w}$ *from* $p(w)$ *and use it to generate* α^*, Y^* *by the recursive relationship defined in 1.1, with the respective* ϵ_i, η_i *replaced by the values from* $\tilde{w}$.
   (b) *Compute* $\hat{\alpha} = \mathbb{E}(\alpha|y)$ *and* $\hat{\alpha^*} = \mathbb{E}(\alpha^*|Y^*)$ *by means of the Kalman filter.*
   (c) *Take* $\tilde{\alpha} = \hat{\alpha} - \hat{\alpha}^* + \alpha^*$.

(2) *Draw time series state component parameters* $\theta = (\Sigma_\mu, \Sigma_\delta, \Sigma_\tau, \Sigma_\omega)$ *given* $\tilde{\alpha}$.

(3) *Simulate* γ *from* $p(\gamma|Y^*, \Sigma_\epsilon)$ *one by one using Stochastic Search Variable Selection [3][4.5.1].*

(4) *Draw* β *from* $p(\beta|\Sigma_\epsilon, \gamma, Y^*)$.

(5) *Draw* Σ_ϵ *from* $p(\Sigma_\epsilon|\beta, \gamma, Y^*)$.

2. Model computation

Before we delve into the results of our model fit, we will first summarize our data. We obtain google trends data using the package `gtrendsR`, and the stock data using the `quantmod` package. The google trends are the weekly trend score (as computed by Google) and the stock data is the weekly closing price. Both sets of data run from April 29, 2018 to April 23, 2023. The summary statistics for the stock data can be seen below:

AAPL.Close	MSFT.Close
Min. : 37.88	Min. : 94.17
1st Qu.: 54.80	1st Qu.:136.69
Median :117.70	Median :214.32
Mean :104.56	Mean :204.02
3rd Qu.:146.61	3rd Qu.:260.25
Max. :178.95	Max. :340.26

Likewise we examine the summary statistics for the google trends data:

Health.Care	Credit	Finance	Oil	Interest.Rate
Min. :3.000	Min. : 69.00	Min. :15.00	Min. :47.00	Min. : 29.00
1st Qu.:4.000	1st Qu.: 79.00	1st Qu.:20.00	1st Qu.:51.00	1st Qu.: 41.00
Median :5.000	Median : 82.00	Median :23.00	Median :53.00	Median : 45.00
Mean :4.736	Mean : 83.12	Mean :22.81	Mean :53.72	Mean : 50.02
3rd Qu.:5.000	3rd Qu.: 88.00	3rd Qu.:25.00	3rd Qu.:55.00	3rd Qu.: 57.00
Max. :6.000	Max. :100.00	Max. :34.00	Max. :92.00	Max. :100.00

Home.Improvement	Sustainability	Artificial.Intelligence
Min. : 6.00	Min. :1.000	Min. :1.000
1st Qu.: 8.00	1st Qu.:1.000	1st Qu.:1.000
Median :10.00	Median :1.000	Median :1.000
Mean :13.41	Mean :1.192	Mean :1.034
3rd Qu.:17.00	3rd Qu.:1.000	3rd Qu.:1.000
Max. :34.00	Max. :2.000	Max. :2.000

Similarly, we may plot the evolution of the stocks and trends data over time.

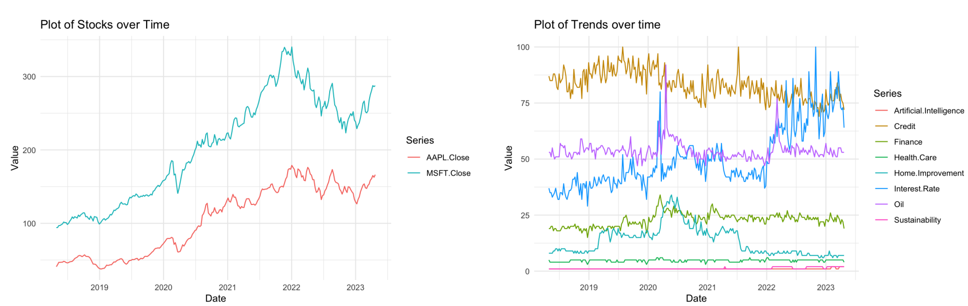
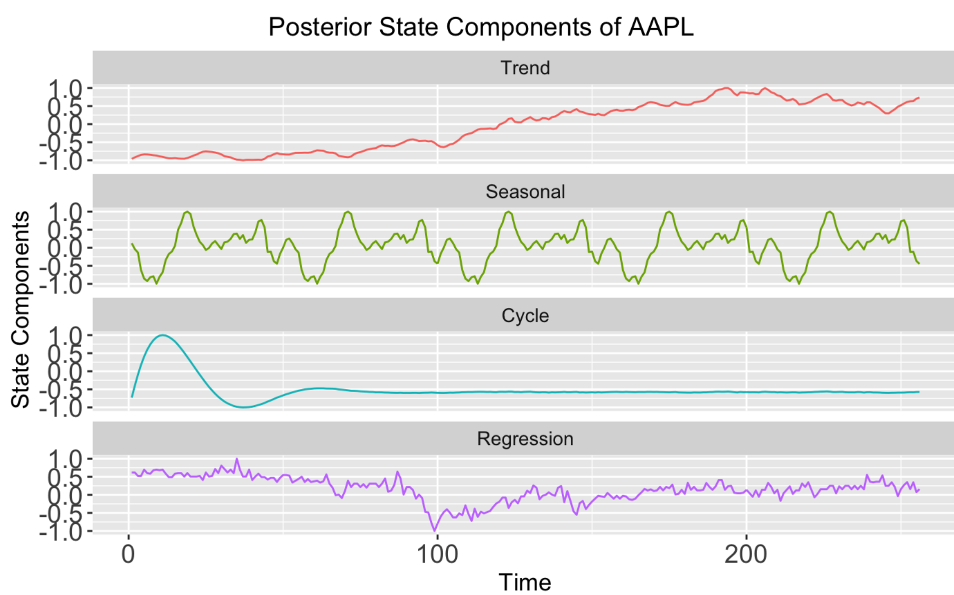
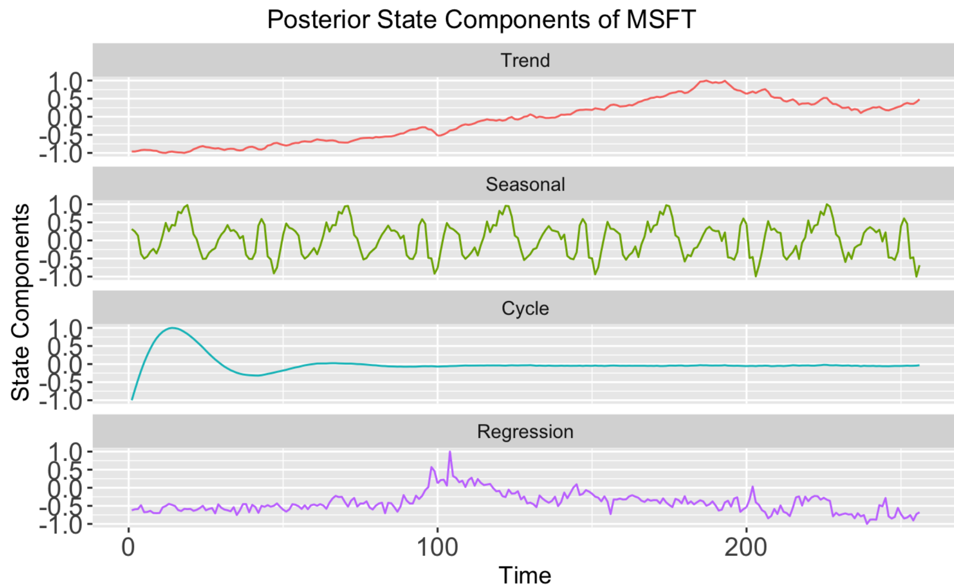


FIGURE 1. Stocks and trends over time

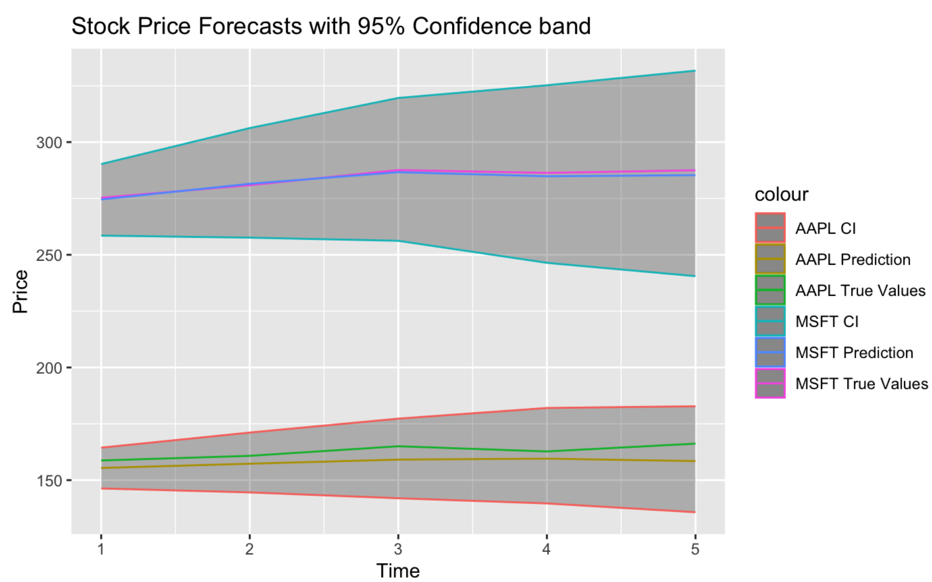
Clearly, we can see that the movement of each stock is highly correlated with the other.

We first divide our sample into a training set and a test set of length 5. After some experimentation with the model parameters, we run our MBSTS model using prior parameters $\rho = [0.5, 0.5], S = [52, 52], \varrho = [0.95, 0.95]$, $\lambda = \left[\frac{\pi}{26}, \frac{\pi}{26}\right]$. Additionally, we set prior $\pi_{i,i} = \frac{1}{8}$ and run our model for 950 iterations (1000 total with a 50 cycle burn in rate). We can view the posterior decomposition of our state components in the following figures:

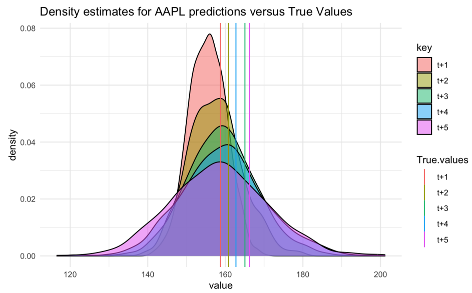


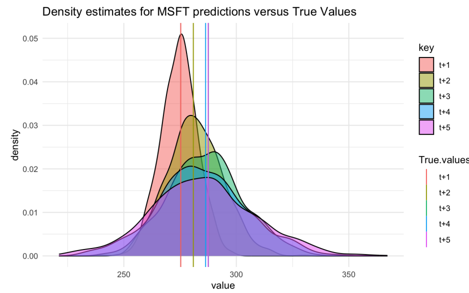


We now evaluate the model's forecasting ability on the training set and contrast it to a simple $ARIMA$ model. We plot the predicted values versus the true values, along with our 95% CI.



Additionally, we can examine how the true values line up against our density estimates for each series.





For a simple 5 step ahead forecast, we find that our MBSTS model has forecast MSE of 25.69 for AAPL and a paltry 1.74 for MSFT. This compares exceptionally favorably with the models fit by `auto.arima` from the `forecast` package. These models have respective MSE's for AAPL and MSFT of $66.53, 166.95$ respectively.

3. Conclusions

We have seen that our simple MBSTS model outperforms ARIMA forecasting by a substantial margin, allows us to decompose the model into its constituent components (which aids interpretability), and provides us with easily interpretable confidence intervals. However, any exposition of the MBSTS model would not be replete without discussing the major drawback of the model: its computational complexity. Although we had initially collected substantially more data, we found it computationally infeasible to run the model using the `mbsts` package with substantially more stocks in our model. As such, we were unable to fully leverage the model's ability to exploit the clear cross correlations across the multivariate time series'. However, in developing the requisite technologies, we have been able to examine many domains of time series analysis which we were not able to explore for our course, including state space models and the Bayesian paradigm.

References

[1] J. Durbin and S. J. Koopman. "A Simple and Efficient Simulation Smoother for State Space Time Series Analysis". In: *Biometrika* 89.3 (2002), pp. 603–615.

[2] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford University Press, May 2012. DOI: 10.1093/acprof:oso/9780199641178.001.0001.

[3] Edward I. George and Robert E. McCulloch. "APPROACHES FOR BAYESIAN VARIABLE SELECTION". In: *Statistica Sinica* 7.2 (1997), pp. 339–373.

[4] James D. Hamilton. *Time series analysis*. Princeton, N.J: Princeton University Press, 1994.

[6] S. Rao Jammalamadaka, Jinwen Qiu, and Ning Ning. *Multivariate Bayesian Structural Time Series Model*. arXiv:1801.03222 [stat]. Sept. 2018.

[7] Jinwen Qiu and Ning Ning. *mbsts: Multivariate Bayesian Structural Time Series*. R package version 3.0. 2023.

[8] Guangchuang Yu. *gtrendsR: A Package for Analyzing Google Trends Data in R*. R package version 1.5.1. 2018.