Annals of Mathematics and Artificial Intelligence (2020) 88:1061–1082
https://doi.org/10.1007/s10472-020-09710-6

Multivariate time series analysis from a Bayesian machine learning perspective

Jinwen Qiu[1] · **S. Rao Jammalamadaka**[1] · **Ning Ning**[2]

Published online: 4 September 2020

Abstract
In this paper, we perform multivariate time series analysis from a Bayesian machine learning perspective through the proposed multivariate Bayesian time series (MBTS) model. The multivariate structure and the Bayesian framework allow the model to take advantage of the association structure among target series, select important features, and train the data-driven model at the same time. Extensive analyses on both simulated data and empirical data indicate that the MBTS model is able to, cover the true values of regression coefficients in 90% credible intervals, select the most important predictors, and boost the prediction accuracy with higher correlation in absolute value of the target series, and consistently yield superior performance over the univariate Bayesian structural time series (BSTS) model, the autoregressive integrated moving average with regression (ARIMAX) model, and the multivariate ARIMAX (MARIMAX) model, in one-step-ahead forecast and ten-steps-ahead forecast.

Keywords Multivariate analysis · Bayesian inference · Structural time series · Feature selection · Prediction

Mathematics Subject Classification (2010) 62H86 · 62M10 · 62F15 · 62F07

1 Introduction

A time series consists of a series of data points on the same variable(s) collected over time, and occurs frequently in statistics (see, for example, [3]), signal processing (see, for

✉ Ning Ning
patning@umich.edu

Jinwen Qiu
jqiu@pstat.ucsb.edu

S. Rao Jammalamadaka
rao@pstat.ucsb.edu

[1] Department of Statistics and Applied Probability, University of California, Santa Barbara, CA, USA

[2] Department of Statistics, University of Michigan, Ann Arbor, MI, USA

example, [7]), pattern recognition (see, for example, [25]), econometrics (see, for example, [32]), mathematical finance (see, for example, [27]), control engineering (see, for example, [23]), to name a few. Time series analyses focus on extracting meaningful statistics and other characteristics of the data, with the primary goal of forecasting future values given previously observed values, which is extremely hard especially for multivariate target time series with a great number of contemporary explanatory variables. Nowadays, machine learning algorithms have become all pervasive and accomplished tasks that until recently only experts could perform. The world is gradually being reshaped by machines possessing "intelligence" and making our lives easier. Machine learning is encompassing every significant aspect of our lives and becoming an integral part of it.

Applying machine learning techniques on time series forecast is very hard in general, mainly because common machine learning techniques assume sample independence, but the time series data do not qualify. While there is success in applying deep learning techniques on time series forecast in recent years (see, for example, [37]), theoretical support for deep learning is still in its infancy (see [12] for a selective overview). On one hand, deep learning models suffer from over-parametrization and nonconvexity. Specifically, the number of parameters in deep learning models is often much larger than the sample size (see Table 1 of [12]) giving them the potential to overfit the training data; even with the help of GPUs, in the worst-case deep learning models is still NP-hard (see [1]) due to the highly nonconvex loss function to minimize. On the other hand, for most successful deep learning algorithms, there is no clear answer on how deep the network should be designed for general classes of problems. For example, in [37], the authors used 1 layer of 128 neurons, while there are no rules on why is the 128 neurons on each layer and how many layers should be used (or suggested to attempt to use).

1.1 Multivariate Bayesian time series (MBTS) model

In this paper, we perform multivariate time series analysis from a Bayesian machine learning perspective through the proposed multivariate Bayesian time series (MBTS) model. The MBTS model as a structural time series model belongs to state space models, which refer to a class of probabilistic graphical models that describe the probabilistic dependence between the latent state variable and the observed measurement. Graphical models as a tool for dealing with the problems of uncertainty and complexity, play an increasingly important role in the design and analysis of machine learning algorithms. They combine probability theory and graph theory, under the idea that is a complex system is built by combining simpler parts, upon which probability theory ensures that the system as a whole is consistent and provides ways to interface models to data. Interested readers are referred to a classical book, [14], for mathematical and algorithmic properties of graph theory on practical problems.

In the MBTS model, multiple time series are decomposed to trend, seasonal, cyclical, and regression components. In the regression part, after removing the effects of other components, we consider the usual multivariate regression setup in that these multiple target series are affected by the same set of predictors but with different coefficients for each target series. We conduct feature selection among those candidate predictors under the sparsity assumption that only a few predictors are crucial among a large number of predictors. The multivariate structure and the Bayesian framework allow the model to take advantage of the association structure among target series, and enable us to do feature selection and model training at the same time. Although a great deal of multiple time series are obviously correlated, most of the existing works about time series analysis focus on univariate target

series analysis. By using the multivariate structure, our model takes into account the correlations among multiple target series and forecasts them as a whole instead of predicting them individually.

Technically, the MBTS model contains three main features: Kalman filter (see [11, 16, 30]), "spike and slab" variable selection (see [13], and [26]), and Bayesian model averaging (see [20]). Specifically, we handle feature selection through the Bayes selection technique via Markov chain Monte Carlo (MCMC) methods, among a set of contemporary predictors which enables online learning. A different set of predictors can be selected in each MCMC iteration, and important predictors will be selected according to their overall frequency of numbers being selected over the total numbers of MCMC iterations. The posterior inclusion probability of each predictor can serve as an indicator that shows its importance to the target series. Through Bayesian model averaging, we do not commit to any particular set of covariates or to point estimates of their coefficients, which helps avoid arbitrary selections, prevents overfitting, and avoids the problem of collinearity.

1.2 Related existing results

In recent years, feature selection is a popular machine learning technique that has wide applications in many areas. [29] provided an efficient feature selection methodology via $\ell_{2,0}$-norm Constrained Sparse Regression. [24] generalized uncorrelated regression with adaptive graphs for unsupervised feature selection. [41] performed feature selection under regularized orthogonal least square regression with optimal scaling. [39] established the link between local regression and global information-embedded dimension reduction. [40] investigated generalized uncorrelated ridge regression with nonnegative labels for unsupervised feature selection. [21] proposed a framework for unsupervised feature selection through joint embedding learning and sparse regression.

Significant progresses of advanced artificial intelligence techniques on multi-dimensional regression analysis of time-series data ([6, 36]), include, but are not limited to, the following: [34, 35] introduced and explored a univariate Bayesian structural time series (BSTS) model, a new Bayesian machine learning technique to explore a given time series along with possible covariates; [8] adopted multi-scale convolutional neural networks for time series classification; [9] achieved texture classification and retrieval using shearlets and linear regression; [4] explored efficient ant colony optimization for image feature selection; [5] automatically decomposed features for single view co-training.

1.3 Superior performance of the MBTS model

Extensive analyses on both simulated data and empirical data verified that the MBTS model gives superior performance over the univariate BSTS model, the autoregressive integrated moving average with regression (ARIMAX) model, and the multivariate ARIMAX (MARIMAX) model. Specifically, through numerical analysis on simulated data, we examine model accuracy in three aspects including parameter estimation, feature selection, and forecastability, by creating several datasets with different correlations among two target series. The MBTS model is used to select predictors for each target series during each MCMC iteration, to investigate its performance on generated datasets and explore its strengths over other models. We demonstrated that the 90% credible interval contains the true value of a specific regression coefficient, the MBTS model can select the most important predictors and ignore those that do not significantly contribute to the target series, and the higher the correlation in absolute value the better the prediction accuracy.

In our empirical study we used the MBTS model to analyze data of three most commonly followed equity indices: Dow Jones Industrial Average (DJIA), Nasdaq Composite Index (Nasdaq), and Standard & Poor's 500 (S&P 500), each of which is considered as one of the best representations of the U.S. stock market and a bellwether for the U.S. economy. The feature selection is conducted upon a pool of 23 economic leading indicators and 27 domestic Google trends, all of which are able to provide unique economic insights. The daily data sample (from 03/05/2007 to 02/13/2018) obtained from Google Finance, Yahoo! Finance and Federal Reserve Economic Data (FRED) in the U.S. region, is split for cross-validation into a training set and a validation set (tuning hyperparameters), and then further examined in a test set. Empirical results demonstrate that MBTS outperforms several benchmark models, in several ways including one-step-ahead forecast and ten-steps-ahead forecast.

1.4 Structure of the paper

The rest of the paper proceeds as follows: In Section 2, we explain the functions of different time series components of the MBTS model, rigorously derive the mathematical formulas underlying the whole framework, and further provide the exact algorithms for model training and forecasting; In Section 3, simulated data are used to analyze the estimation accuracy and forecast performance, and establish the positive relationship with target time series' correlations in absolute value; In Section 4, an empirical study on a portfolio of three most commonly followed equity indices demonstrates that MBTS outperforms the other benchmark models; In Section 5, we conclude and remark on further extensions and applications.

2 The MBTS model

In this section, based on the Structural Time Series (STS) model and "spike and slab" regression, a new model-based approach is introduced to do feature selection and forecast among multiple target time series in the multivariate context.

2.1 Structural time series

As a special case of the state-space representation, the STS model is formulated in terms of unobserved components, which have direct interpretations. One of the obvious advantages that the STS model has, over the widely used autoregressive integrated moving average (ARIMA) representations, can be easily seen from the following facts: data can be easily and structurally modeled with slowly changing trends and further superimposed short-term movements; differencing is a standard stationary and/or de-trend technique in other time series models, however the differenced observations usually become non-invertible, together with long lags, many parameters, and unfavorable results in unit root or cointegration test (see [18]); analyses based on autoregressive integrated moving average (ARIMA) models can also be misleading, if such models are built up primarily on grounds of parsimony (see [17]). On the contrary, structural time series models are able to provide a very useful framework within which to present stylized facts on time series.

A typical structural model decomposes time series into four components: trend (level and slope), seasonal effects, cyclical, and an irregular component (or the error term). Here, to allow for the effects of covariates on target series, a regression component is

also added. Thus, a multiple response series, representing m target time series $\tilde{y}_t = [y_t^{(1)}, \ldots, y_t^{(i)}, \ldots, y_t^{(m)}]^T$, may be expressed in the following structural form:

$$\tilde{y}_t = \tilde{\mu}_t + \tilde{\tau}_t + \tilde{\omega}_t + \tilde{\xi}_t + \overset{\epsilon}{\sim}_t, \qquad \tilde{\epsilon}_t \overset{iid}{\sim} N_m(0, \Sigma_\epsilon), \tag{2.1}$$

where $\tilde{\mu}_t$, $\tilde{\tau}_t$, $\tilde{\omega}_t$, $\tilde{\xi}_t$ and $\tilde{\epsilon}_t$ are all m-dimension vectors, representing linear trend components, seasonal components, cyclical components, regression components, and observation error terms respectively, for $t = 1, 2 \ldots, n$ observations. The variance of the error term Σ_ϵ is a $m \times m$-dimensional variance-covariance matrix, positive definite and constant over time for simplicity. In the model, all state components are assembled independently (their variance-covariance matrices are diagonal), with each component yielding an additive contribution to $\tilde{y}_t$.

2.2 Components of the time series

The first component $\tilde{\mu}_t$ in (2.1), is a general local linear trend (GLLT). A trend is the long-term growth of the time series, and it can be further decomposed into two components: level and slope. Level represents the actual mean value of the trend and slope represents the tendency to grow or decline from the trend. Whether to include the slope depends on the features shown in the series under investigation and any prior knowledge. More specifically, a slope component is often used when we have data with at least locally continuous growth or decay such as GDP, population, oil reserves, etc. The GLLT is described as:

$$\tilde{\mu}_{t+1} = \tilde{\mu}_t + \tilde{\delta}_t + \tilde{u}_t, \qquad \tilde{u}_t \overset{iid}{\sim} N_m(0, \Sigma_\mu), \tag{2.2}$$

$$\tilde{\delta}_{t+1} = \tilde{D} + \tilde{\rho}(\tilde{\delta}_t - \tilde{D}) + \tilde{v}_t, \qquad \tilde{v}_t \overset{iid}{\sim} N_m(0, \Sigma_\delta). \tag{2.3}$$

Here, $\tilde{\delta}_t$ is a m-dimensional vector and models the expected increase or decrease in $\tilde{\mu}_t$, which can be thought of as the local slope at time t. $\tilde{u}_t$ stands for the error term in (2.2) and $\tilde{v}_t$ stands for error term in (2.3). In the dynamics of $\tilde{\delta}$, the m-dimensional vector $\tilde{D}$ models the long-term slope, and the parameter $\tilde{\rho}$ is a $m \times m$-dimensional diagonal matrix, whose diagonal entry $\rho_{ii} \in [0, 1]$ represents the learning rate. That is, the slope captures and balances short-term information and long-term information through the value of ρ_{ii}, for example the new information dominates the slope when $\rho_{ii} = 1$.

The second component $\tilde{\tau}_t$ in (2.1), is the one describing seasonality, a characteristic of a time series in which the data experiences regular and predictable changes that recur every period. Equivalently, any predictable change or pattern in a time series that recurs or repeats over equal length periods can be said to be seasonal, such as weather fluctuations or the Christmas effect. A seasonal component may be necessary when we have quarterly, monthly or daily time series. One frequently used model for the seasonal component (see, for example, Example 2.16 in [10] and the terminology "seasonally adjust" on page 8 therein, for reference), is of the form:

$$\tau_{t+1}^{(i)} = -\sum_{k=0}^{S_i-2} \tau_{t-k}^{(i)} + w_t^{(i)}, \quad i = 1, \cdots, m,$$

$$\tilde{\tau}_t = [\tau_t^{(1)}, \cdots, \tau_t^{(m)}]^T, \quad \tilde{w}_t = [w_t^{(1)}, \cdots, w_t^{(m)}]^T \overset{iid}{\sim} N_m(0, \Sigma_\tau), \tag{2.4}$$

where S_i represents the number of seasons for $y^{(i)}$ and the m-dimensional vector $\tilde{\tau}_t$ denotes their joint seasonal contribution to the observed response $\tilde{y}_t$. Here, $\tilde{w}_t = [w_t^{(1)}, \cdots, w_t^{(m)}]^T$

stands for the error term in the seasonal component, and $w_t^{(i)}$ stands for the error term of each time series i for $i = 1, \cdots, m$. There are S_i seasonal factors for each response series $y^{(i)}$, and the expected sum of the total seasonal effects is zero. Another obvious strength of the structural model is that, it can be used to model multiple time series data displaying several different seasonal components with their own periods.

The third component $\tilde{\omega}_t$ in (2.1), is the one accounting for the cyclical effect in the series, which refers to regular or periodic fluctuations around the trend, revealing a succession of phases of expansion and contraction. In contrast to seasonality that is always of fixed and known periods, a cyclic pattern exists when data exhibits ups and downs that are not of fixed periods. Therefore, a model with the cyclical component is capable of reproducing commonly acknowledged essential features, such as the presence of strong autocorrelation, the existence of recurrence and alteration of phases, the dampening of fluctuations, to name a few. A cyclical component can capture the short-term movement of serially correlated stationary series (see, [19]). Specifically, the cycle component is formulated as a fully-coupled dynamical system, in that $\tilde{\omega}$ is defined with $\tilde{\omega}^\star$ and $\tilde{\omega}^\star$ is defined with $\tilde{\omega}$:

$$\begin{aligned} \tilde{\omega}_{t+1} &= \tilde{\varrho}\widehat{\cos(\lambda)}\tilde{\omega}_t + \tilde{\varrho}\widehat{\sin(\lambda)}\tilde{\omega}_t^\star + \tilde{\kappa}_t, \qquad \tilde{\kappa}_t \overset{iid}{\sim} N_m(0, \Sigma_\omega), \\ \tilde{\omega}_{t+1}^\star &= -\tilde{\varrho}\widehat{\sin(\lambda)}\tilde{\omega}_t + \tilde{\varrho}\widehat{\cos(\lambda)}\tilde{\omega}_t^\star + \tilde{\kappa}_t^\star, \qquad \tilde{\kappa}_t^\star \overset{iid}{\sim} N_m(0, \Sigma_\omega), \end{aligned} \tag{2.5}$$

where $\tilde{\varrho}$, $\widehat{\sin(\lambda)}$, and $\widehat{\cos(\lambda)}$ are $m \times m$ diagonal matrices with diagonal entries ϱ_{ii}, $\sin(\lambda_{ii})$, and $\cos(\lambda_{ii})$ respectively. $\tilde{\kappa}$ stands for the error term in the dynamic of $\tilde{\omega}$ and $\tilde{\kappa}$ stands for error term in the dynamic of $\tilde{\omega}^\star$. Here, $\lambda_{ii} = 2\pi/q_i$ is the frequency with q_i being the period, and ϱ_{ii} is the corresponding damping factor for target series $y^{(i)}$, such that $0 < \lambda_{ii} < \pi$ and $0 < \varrho_{ii} < 1$. The value of this damping factor determines the decaying speed of the amplitude of the cycle, in that a larger value generating slower decay and vise versa. The boundary values are not in the consideration for the following reasons: The two boundary values of 0 and π of λ_{ii} will degenerate the model to the AR(1) process; The boundary value 0 of ϱ_{ii} will fully regenerate the model to white noise and 1 will cancel restrictions on the cyclical movement which results in extending the amplitude of the cycle.

The fourth component $\tilde{\xi}_t$ in (2.1), is the regression component with static coefficients formulated as

$$\tilde{\xi}_t = B^T \tilde{x}_t, \tag{2.6}$$

where $B = [\beta_1, \ldots, \beta_i, \ldots, \beta_m]$ denotes a $k \times m$-dimensional matrix with $\beta_i = [\beta_{1i}, \ldots, \beta_{ki}]^T$ representing static regression coefficients, and $\tilde{x}_t = [x_{t1}, \ldots, x_{tk}]^T$ is a pool of regressors. Although all predictors are supposed to be contemporaneous, a known lag can be easily included by shifting the corresponding covariate in time. In this study, we focus on variable selection and a high degree of sparsity is expected, in the sense that coefficients for the vast majority of massive predictors will be zero. In the Bayesian paradigm, a natural way to represent sparsity is through the "spike and slab" technique, which will be covered next.

2.3 Spike and slab regression

To facilitate our analysis and obtain a neat form for the natural conjugate prior for the regression coefficients, we rewrite the model in a matrix form as

$$Y = M + T + W + XB + E, \tag{2.7}$$

where $Y = [\tilde{y}_1, \ldots, \tilde{y}_n]^T$, $M = [\tilde{\mu}_1, \ldots, \tilde{\mu}_n]^T$, $T = [\tilde{\tau}_1, \ldots, \tilde{\tau}_n]^T$, $W = [\tilde{\omega}_1, \ldots, \tilde{\omega}_n]^T$ and $E = [\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_n]^T$ are $n \times m$-dimensional matrices, and $X = [\tilde{x}_1, \ldots, \tilde{x}_n]^T$ is the $n \times k$-dimensional matrix of n observations on k common predictors.

2.3.1 Prior distribution and elicitation

We consider the case that the set of selected predictors at each iteration is the same for different target series, and then set the inclusion indicator $\gamma = [\gamma_1, \cdots, \gamma_j, \cdots, \gamma_k]$ for these k common predictors, that is $\gamma_j = 1$ then $\beta_{ji} \neq 0$ and $\gamma_j = 0$ then $\beta_{ji} = 0$, for all $i = 1, \ldots, m$ target series. Let B_γ denote the subset of rows of B where $\beta_{ji} \neq 0$ for all i and X_γ represents the subset of columns of X where $\gamma_j \neq 0$. A spike and slab prior for the joint distribution of regression coefficient matrix B, variance-covariance matrix for error terms Σ_ϵ and inclusion indicator γ can be factorized into several conditional and marginal distributions as

$$p(B, \Sigma_\epsilon, \gamma) = p(B|\Sigma_\epsilon, \gamma)p(\Sigma_\epsilon|\gamma)p(\gamma), \tag{2.8}$$

where $p(\cdot)$ stands for the probability density function and $p(\gamma)$ is the prior distribution in Bayesian inference terminology.

The prior distribution $p(\gamma)$ is the so-called "spike", since it sets positive probability mass to zero. For simplicity, a spike prior may be written as an independent product of Bernoulli probabilities

$$\gamma \sim \prod_{j=1}^{k} \pi_j^{\gamma_j}(1 - \pi_j)^{1-\gamma_j}, \tag{2.9}$$

where π_j is the prior inclusion probability of the j-th predictor. π_j can be easily elicited by asking the researcher for an "expected model size", so that if one expects q nonzero predictors, then $\pi_j = q/k$ for all $i = 1, \ldots, m$ target series, where k is the total number of common predictors. π_i could be set as 0 or 1 for some specific values of i under some circumstances, which decides whether to include certain variables. Note that, another prior probability could be assigned to γ if necessary.

Given knowledge of which coefficients are nonzero, or simply say, given γ, a prior for the values of the nonzero coefficients gives the so-called "slab", which can be expressed through the conjugate multinormal inverse Wishart distribution of $\beta_\gamma = \text{vec}(B_\gamma)$ and Σ_ϵ:

$$\beta_\gamma|\Sigma_\epsilon, \gamma \sim N_{m \cdot k}(b_\gamma, \Sigma_\epsilon \otimes \Omega_\gamma^{-1}), \qquad \Sigma_\epsilon|\gamma \sim IW(v_0, V_0), \tag{2.10}$$

where $\otimes$ is the Kronecker product. In line with [34] and the references therein, the full-model prior information matrix Ω_γ^{-1} can be set in the Zellner's g prior way as $\Omega_\gamma^{-1} = \kappa X_\gamma^T X_\gamma / n$, where κ is the number of observations worth of weight on the prior mean b_γ. In the case that the positive definite property for $X_\gamma^T X_\gamma$ is violated, an alternative version can be used as $\Omega_\gamma^{-1} = \frac{\kappa}{n}(\omega X_\gamma^T X_\gamma + (1 - \omega)\text{diag}(X_\gamma^T X_\gamma))$. In the inverse Wishart distribution of $\Sigma_\epsilon|\gamma$, the number of degrees of freedom is denoted as v_0, and V_0 is a $m \times m$-dimensional scale matrix, both of whose values can be set based on researchers' desired R^2. More specifically, v_0 can be specified based on the number of observations worth of weight, and $V_0 = (v_0 - m - 1) \times (1 - R^2) \times \Sigma_y$ with Σ_y being the variance-covariance matrix for multiple target series Y.

The prior distribution of other variance-covariance matrices can be expressed as:

$$\Sigma_u \sim IW(w_u, W_u), \quad \text{for } u \in \{\mu, \delta, \tau, \omega\}. \tag{2.11}$$

By assuming that these matrices are diagonal, the prior distribution in multivariate case degenerates to the corresponding univariate case, i.e., diagonals of these matrices follow inverse gamma distributions.

2.3.2 Posterior inference

Now, let us focus on the regression part and subtract out the time series components (trend, seasonality, and cycle) from Y. That is, we consider $Y^\star = [\tilde{y}_1^\star, \ldots, \tilde{y}_t^\star, \ldots, \tilde{y}_n^\star]^T$ where $\tilde{y}_t^\star = \tilde{y}_t - \tilde{\mu}_t - \tilde{\tau}_t - \tilde{\omega}_t$. By the law of total probability, we have

$$p(Y^\star, B, \Sigma_\epsilon, \gamma) = p(Y^\star|B, \Sigma_\epsilon, \gamma) \times p(B|\Sigma_\epsilon, \gamma) \times p(\Sigma_\epsilon|\gamma) \times p(\gamma), \tag{2.12}$$

$$p(Y^\star|B, \Sigma_\epsilon, \gamma) \propto |\Sigma_\epsilon|^{-n/2} \exp\left(\mathrm{Tr}\left(-\frac{1}{2}(Y^\star - X_\gamma B_\gamma)^T (Y^\star - X_\gamma B_\gamma)\Sigma_\epsilon^{-1}\right)\right), \tag{2.13}$$

$$p(B|\Sigma_\epsilon, \gamma) \propto |\Omega_\gamma|^{m/2} |\Sigma_\epsilon|^{-k/2} \exp\left(\mathrm{Tr}\left(-\frac{1}{2}(B_\gamma - \bar{B}_\gamma)^T \Omega_\gamma (B_\gamma - \bar{B}_\gamma)\Sigma_\epsilon^{-1}\right)\right), \tag{2.14}$$

$$p(\Sigma_\epsilon|\gamma) \propto |\Sigma_\epsilon|^{-(v_0+m+1)/2} \exp\left(\mathrm{Tr}\left(-\frac{1}{2}V_0\Sigma_\epsilon^{-1}\right)\right), \tag{2.15}$$

where $\mathrm{Tr}(\cdot)$ stands for the trace of a matrix. Let us combine terms from the natural conjugate prior to produce a posterior, which is a product of an inverse Wishart and a "matrix" normal kernel. That is, we combine the two terms in $\exp(\mathrm{Tr}(\cdot))$ involving B:

$$\begin{aligned} & (B_\gamma - \bar{B}_\gamma)^T \Omega_\gamma (B_\gamma - \bar{B}_\gamma) + (Y^\star - X_\gamma B_\gamma)^T (Y^\star - X_\gamma B_\gamma) \\ = \; & (Z - W B_\gamma)^T (Z - W B_\gamma) \\ = \; & (Z - W\tilde{B}_\gamma)^T (Z - W\tilde{B}_\gamma) + (B_\gamma - \tilde{B}_\gamma)^T W^T W (B_\gamma - \tilde{B}_\gamma). \end{aligned}$$

Here, W is a matrix built up by placing the matrix X on the top of the matrix U, for U being the Cholesky decomposition of Ω_γ, as follows:

$$W = \begin{pmatrix} X \\ U \end{pmatrix}, \qquad \Omega_\gamma = U^T U.$$

Z is a matrix built up by placing the matrix $Y^\star$ on the left of the matrix $U\bar{B}_\gamma$, as follows:

$$Z = \begin{pmatrix} Y^\star, & U\bar{B}_\gamma \end{pmatrix}, \qquad \tilde{B}_\gamma = (X_\gamma^T X_\gamma + \Omega_\gamma)^{-1}(X_\gamma^T Y^\star + \Omega_\gamma \bar{B}_\gamma).$$

Then the posterior density can be written as:

$$\begin{aligned} p(B, \Sigma_\epsilon, \gamma|Y^\star) \propto \; & p(\gamma) \times |\Sigma_\epsilon|^{-(v_0+n+m+1)/2} \exp\left(\mathrm{Tr}\left(-\tfrac{1}{2}(V_0 + S_\gamma)\Sigma_\epsilon^{-1}\right)\right) \\ & \times |\Omega_\gamma|^{m/2} |\Sigma_\epsilon|^{-k/2} \exp\left(\mathrm{Tr}\left(-\tfrac{1}{2}(B_\gamma - \tilde{B}_\gamma)^T W^T W (B_\gamma - \tilde{B}_\gamma)\Sigma_\epsilon^{-1}\right)\right), \end{aligned} \tag{2.16}$$

with

$$\begin{aligned} S_\gamma &= (Z - W\tilde{B}_\gamma)^T (Z - W\tilde{B}_\gamma) \\ &= (Y^\star - X_\gamma \tilde{B}_\gamma)^T (Y^\star - X_\gamma \tilde{B}_\gamma) + (\tilde{B}_\gamma - \bar{B}_\gamma)^T \Omega_\gamma (\tilde{B}_\gamma - \bar{B}_\gamma) \end{aligned}$$

and

$$W^T W = X_\gamma^T X_\gamma + \Omega_\gamma.$$

Note that, the term involving B_γ is a density expressed as a function of an arbitrary $k \times m$-dimensional matrix, which can be converted from a function of B_γ to a function of $\beta_\gamma = \text{vec}(B_\gamma)$ using standard results on vectorization:

$$\begin{aligned} & \text{Tr}((B_\gamma - \tilde{B}_\gamma)^T W^T W (B_\gamma - \tilde{B}_\gamma)\Sigma_\epsilon^{-1}) \\ = \ & \text{vec}(B_\gamma - \tilde{B}_\gamma)^T \ \text{vec}(W^T W (B_\gamma - \tilde{B}_\gamma)\Sigma_\epsilon^{-1}) \\ = \ & \text{vec}(B_\gamma - \tilde{B}_\gamma)^T (\Sigma_\epsilon^{-1} \otimes W^T W)\text{vec}(B_\gamma - \tilde{B}_\gamma) \\ = \ & (\beta_\gamma - \tilde{\beta}_\gamma)^T (\Sigma_\epsilon^{-1} \otimes W^T W)(\beta_\gamma - \tilde{\beta}_\gamma), \end{aligned}$$

where $\tilde{\beta}_\gamma = \text{vec}(\tilde{B}_\gamma)$. Thus, the posteriors are in the form of conjugacy:

$$\Sigma_\epsilon | Y^\star, \gamma \sim IW(v_0 + n, V_0 + S_\gamma), \tag{2.17}$$

$$\beta_\gamma | Y^\star, \Sigma_\epsilon, \gamma \sim N_{m \cdot k}(\tilde{\beta}_\gamma, \Sigma_\epsilon \otimes (X_\gamma^T X_\gamma + \Omega_\gamma)^{-1}). \tag{2.18}$$

Because of the conjugacy properties, one can analytically marginalize over β_γ and Σ_ϵ to obtain the posterior of γ

$$p(\gamma | Y^\star) = C(Y^\star) \frac{|\Omega_\gamma|^{\frac{m}{2}}}{|X_\gamma^T X_\gamma + \Omega_\gamma|^{\frac{m}{2}}} \frac{p(\gamma)}{|V_0 + S_\gamma|^{\frac{v_0+n}{2}}}, \tag{2.19}$$

where $C(Y^\star)$ is a normalizing constant that depends on $Y^\star$ but not on γ. The (2.19) places positive probabilities on coefficients being zero, leading to the sparsity incorporated in the model.

The error terms for time series components follow a multivariate normal distribution with mean equal to zero, and variance follows an inverse Wishart distribution based on (2.11). Thus, the posterior of variances of the error terms for time series components can be easily derived based on the conjugate property of multivariate normal and inverse Wishart distribution. That is, given the draw of time series components, the posterior distribution will remain inverse Wishart distributed with parameters depending on residuals of these components, according to

$$\Sigma_u | u \sim IW(w_u + n, W_u + AA^T), \quad \text{for } u \in \{\mu, \delta, \tau, \omega\}, \tag{2.20}$$

where $A = [\tilde{A}_1, \ldots, \tilde{A}_n]$ is a $m \times n$-dimensional matrix, representing a collection of residues for each time series component.

2.4 Algorithms

In this section, we introduce two algorithms for model training using the Markov chain Monte Carlo (MCMC) technique and prediction using the Bayesian model averaging technique respectively, incorporated perfectly to select important features in a fully data-driven way and avoid overfitting together. During model training, the "spike and slab" regression selects important regression predictors, by removing redundant variables and keeping important variables during each iteration. The relative importance of each predictor depends on its corresponding empirical inclusion probability, which can be obtained by the proportion of draws with the inclusion indicator $\gamma = 1$. As is typical in Bayesian data analysis, forecasts are based on the posterior predictive distribution. Given draws of model parameters and latent states from their posterior distributions, we can draw samples from the posterior predictive distribution indirectly.

2.4.1 Model training

Denote the set of state component parameters as $\theta = (\Sigma_\mu, \Sigma_\delta, \Sigma_\tau, \Sigma_\omega)$ and denote γ_{-i} as the vector whose elements are those of γ other than γ_i for $i \in \{1, \cdots, k\}$. Based on Algorithm 1 of model training, we draw samples from posterior distributions for corresponding parameters at each step. Step 1 is the data augmentation step using the Kalman filter technique. The draw of θ in step 2 depends on which state components are presented in the model. The SSVS algorithm used in step 3 is a Gibbs sampling algorithm, where each element of γ is drawn from its conditional posterior distribution, proportional to (2.19). Although a closed form of the distribution is not possible to be derived, it is not necessary actually in that by calculating the probability of the only two possible values for each γ_i, we can get rid of the unknown constant value $C(Y^\star)$ not involving γ in (2.19). Then we need to loop over all elements of γ in random order during each MCMC iteration. Let $\tilde{\psi} = (\alpha, \theta, \gamma, \Sigma_\epsilon, \beta)$ represent all parameters and latent states. Based on Algorithm 1, sequential draws $\tilde{\psi}^{(1)}, \tilde{\psi}^{(2)}, \tilde{\psi}^{(3)} \ldots$ are simulated by going through the five steps in Algorithm 1 repetitively, which forms the empirical stationary posterior distribution $p(\tilde{\psi}|Y)$.

Algorithm 1 Model Training.

1: Draw the latent state $\alpha = (\tilde{\mu}, \tilde{\delta}, \tilde{\tau}, \tilde{\omega})$ from given model parameters and Y, namely $p(\alpha|Y, \theta, \gamma, \Sigma_\epsilon, \beta)$, using the posterior simulation algorithm from [11].
2: Draw state component parameters θ given α, namely simulating $\theta \sim p(\theta|Y, \alpha)$ based on (2.20).
3: Loop over i in random order, draw each $\gamma_i \Big| \{\gamma_{-i}, Y, \alpha, \Sigma_\epsilon\}$, namely simulating $\gamma \sim p(\gamma|Y^\star)$ one by one, using the stochastic search variable selection (SSVS) algorithm from [13], based on (2.19).
4: Draw Σ_ϵ given γ, α and Y, namely simulating $\Sigma_\epsilon \sim p(\Sigma_\epsilon|\gamma, Y^\star)$ based on (2.17).
5: Draw β given $\Sigma_\epsilon, \gamma, \alpha$ and Y, namely simulating $\beta \sim p(\beta|\Sigma_\epsilon, \gamma, Y^\star)$ based on (2.18).

2.4.2 Forecasting

Under the situation that the posterior inclusion probability density function has no closed-form expression, we turn to the empirical posterior inclusion probability computed by the number of the proportion of the number of one predictor being selected to the total number of MCMC iterations. That is, if one predictor was selected 100 times in 200 MCMC iterations, the empirical posterior inclusion probability is $100/200 = 0.5$. The chance that each predictor will be selected relies on (2.19). With each MCMC draw from model training, a forecast value is generated according to steps $1 - 3$ of Algorithm 2. The samples drawn in this way have the same distribution as those simulated directly from posterior predictive distribution. Through Bayesian model averaging, we commit neither to any particular set of covariates nor to point estimates of their coefficients, which helps avoid arbitrary selections and prevents overfitting. By using the multivariate model, we also take into account the correlations among multiple target series, when sampling for predicted values of several target series, that is forecasting multiple target series as a whole instead of predicting them individually. In particular, the point prediction values could be formed by taking the average

of drawn samples, and prediction intervals could be formed by computing corresponding quantiles of drawn samples, as in step 4 of Algorithm 2.

Algorithm 2 Model Forecast.

1: Draw the next latent time series states $\alpha_{t+1} = (\tilde{\mu}_{t+1}, \tilde{\delta}_{t+1}, \tilde{\tau}_{t+1}, \tilde{\omega}_{t+1})$ given current latent time series states $\alpha_t = (\tilde{\mu}_t, \tilde{\delta}_t, \tilde{\tau}_t, \tilde{\omega}_t)$ and component parameters $\theta = (\Sigma_\mu, \Sigma_\delta, \Sigma_\tau, \Sigma_\omega)$, based on (2.2), (2.3), (2.4) and (2.5).
2: Based on indicator variable γ, compute the regression component given the information about predictors at time $t+1$ by (2.6).
3: Draw a random error in multivariate normal distribution with variance equal to Σ_ϵ and sum them up using (2.1).
4: Sum up all the predictions and divide by the total number of MCMC iterations to generate the point prediction; establish the prediction intervals according to the corresponding quantile of predictors.

3 Application to simulated data

In this section, we examine model accuracy in three aspects including parameter estimation, feature selection, and forecastability, by creating several datasets with different correlations among two target series. The data are generated by a model containing a general linear trend component, seasonal component and a static regression component with eight predictors, while two of these components do not affect multiple target series, i.e. with zero coefficient. MBTS model will be used to select the same set of predictors for each target series during each MCMC iteration, to investigate its performance on generated datasets and explore its strengths over other models, specifically to answer questions such as, how likely the 90% credible interval will contain the true value of a specific regression coefficient, and how possible the model will select the most important predictors or ignore some that do not significantly contribute to target series.

3.1 Model setup and data generation

We consider a general linear trend component having a global slope 0.02, and consider a seasonal component with a period of 4, initialized at $\tilde{\mu}_0 = [3, -2]^T$, $\tilde{\delta}_0 = [0.01, -0.01]^T$, $\tilde{\tau}_{1,-2} = -4$, $\tilde{\tau}_{2,-2} = 4$, $\tilde{\tau}_{1,-1} = -3$, $\tilde{\tau}_{1,-1} = 3$, $\tilde{\tau}_{1,0} = -2$, and $\tilde{\tau}_{2,0} = 2$. The model applied to the generated data is described as follows:

$$\tilde{y}_t = \tilde{\mu}_t + \tilde{\tau}_t + B^T \tilde{x}_t + \tilde{\epsilon}_t, \qquad \text{for } t = 1, 2, \cdots, 3000,$$

$$\tilde{\mu}_{t+1} = \begin{bmatrix} \mu_{1,t+1} \\ \mu_{2,t+1} \end{bmatrix} = \begin{bmatrix} \mu_{1,t} \\ \mu_{2,t} \end{bmatrix} + \begin{bmatrix} \delta_{1,t} \\ \delta_{2,t} \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix},$$

$$\tilde{\tau}_{t+1} = \begin{bmatrix} \tau_{1,t+1} \\ \tau_{2,t+1} \end{bmatrix} = \begin{bmatrix} -\sum_{k=0}^{2} \tau_{1,t-k} \\ -\sum_{k=0}^{2} \tau_{2,t-k} \end{bmatrix} + \begin{bmatrix} w_{1,t} \\ w_{2,t} \end{bmatrix},$$

$$B = \begin{bmatrix} 2 & -1 & -0.5 & 0 & 1.5 & -2 & 0 & 3.5 \\ -1.5 & 4 & 2.5 & 0 & -1 & -3 & 0 & 0.5 \end{bmatrix}^T \quad \text{and} \quad \tilde{x}_t = [x_{t_1}, x_{t_2}, \cdots, x_{t_8}]^T,$$

where the distributions are given as

$$\begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} \overset{iid}{\sim} N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1 \end{bmatrix} \right), \quad \begin{bmatrix} \delta_{1,t} \\ \delta_{2,t} \end{bmatrix} \overset{iid}{\sim} N_2\left(\begin{bmatrix} 0.8\delta_{1,t-1} + 0.2 * 0.02 \\ 0.5\delta_{2,t-1} - 0.5 * 0.02 \end{bmatrix}, \begin{bmatrix} 0.08^2 & 0 \\ 0 & 0.16^2 \end{bmatrix} \right),$$

$$x_{t_1} \overset{iid}{\sim} N(5, 5^2), \quad x_{t_2} \overset{iid}{\sim} \text{Poi}(10), \quad x_{t_3} \overset{iid}{\sim} \text{Bin}(1, 0.5), \quad x_{t_4} \overset{iid}{\sim} N(2, 5^2),$$

$$x_{t_5} \overset{iid}{\sim} N(-5, 5^2), \quad x_{t_6} \overset{iid}{\sim} \text{Poi}(15), \quad x_{t_7} \overset{iid}{\sim} \text{Poi}(20), \quad x_{t_8} \overset{iid}{\sim} N(0, 10),$$

$$\tilde{\epsilon}_t \overset{iid}{\sim} N_2(0, \Sigma_\epsilon) \quad \text{with} \quad \Sigma_\epsilon = \begin{bmatrix} 1.1 & 0.7 \\ 0.7 & 0.9 \end{bmatrix}, \quad \begin{bmatrix} w_{1,t} \\ w_{2,t} \end{bmatrix} \overset{iid}{\sim} N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.01^2 & 0 \\ 0 & 0.01^2 \end{bmatrix} \right).$$

We also create variables $(x^\star_{t_2}, x^\star_{t_5}, x^\star_{t_8})$, whose values are obtained by rearranging a partial portion of data values for $(x_{t_2}, x_{t_5}, x_{t_8})$. We are going to use $(x^\star_2, x^\star_5, x^\star_8)$ instead of (x_2, x_5, x_8) in model training, therefore regression coefficients for (x_2, x_5, x_8) used for target series generation are expected to not able to reflect the true linear relationship between y^i and $(x^\star_2, x^\star_5, x^\star_8)$.

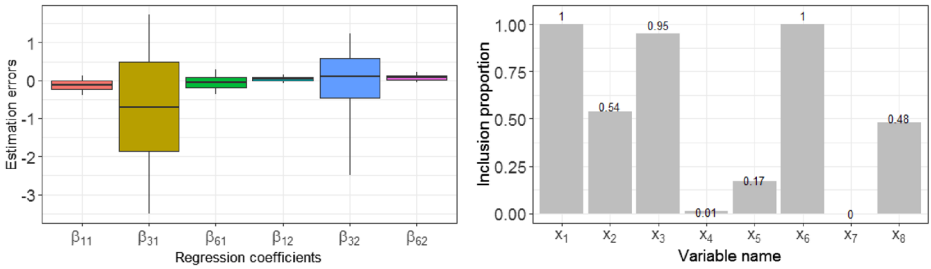
3.2 Estimation accuracy

After model training, we draw 6000 sample values from MCMC iterations to build the sample posterior distribution for each parameter, and discard the first 1000 samples as burn-in. Firstly, we use the absolute values of differences between true values and MCMC results as the metric to measure estimation errors of regression coefficients. To visualize the distribution of estimation errors for regression coefficients, we construct a box plot in Fig. 1a, whose top and bottom represent the upper bound and lower bound of the 90% credible interval respectively and only shows regression coefficients of important predictors. Secondly, we create a bar plot in Fig. 1b to specially check the feature selection performance, another important characteristic of our model, based on inclusion probabilities for all candidate predictors, which are computed by the empirical posterior distribution of the indicator variable γ.

From Fig. 1a, we can see that zero falls within the two endpoints of all boxplots, implying that all 90% credible intervals contain true values of the regression coefficients, verifying accurate estimation. Furthermore, boxes for regression coefficients of category variable x_3 are much wider than those for numerical variables x_1, x_6 generated from the normal distribution and the Poisson distribution. The estimated regression coefficients of x_1, x_6 are also closer to true values than that of x_3. Figure 1b clearly shows that x_1, x_3 and x_6 are important predictors selected almost every time during 5000 MCMC iterations with inclusion probabilities 1, 0.95 and 1 respectively. The sample inclusion probabilities are almost zero for x_4 and x_7, which indicate their corresponding coefficients are zero and these two should not be included as predictors. The posterior inclusion probabilities of the remaining variables x_2, x_5 and x_8 vary depending on the relationship between partially reshuffled values and target series, and in general they are all below 60%. In sum, our approach is capable of precisely estimating regression coefficients and the same as identifying the most important variables as well.

3.3 Model comparison

From the last section, we see that when a predictor is significant in the target series, the estimation accuracy is extremely high, no matter the signs of one predictor are the same or opposite. Therefore, one can expect better prediction results when the target series have



(a) Estimation accuracy (b) Feature selection

Fig. 1 **a** displays posterior distribution of estimation errors for regression coefficients and **b** shows proportion of variables to be selected during MCMC iterations

a high correlation in absolute value. In this section, we verify this conjecture in ten one-step-ahead predictions, using a growing window approach, which simply adds one new observation in the test set and obtains a new model with fresher data to constantly forecast a new value in the test set. We use the same model to generate multiple datasets with various correlations ($\rho = 0, 0.2, -0.5, 0.8$) between error terms of two target series, in order to check the effects of correlation on model forecast performance.

We evaluate the prediction performance in terms of cumulative one-step-ahead absolute forecast errors computed by formula $\sum_{t=1}^{t=10} \sum_{i=1}^{2} |y_t^{(i)} - \hat{y}_t^{(i)}|$, and compare with two benchmark models: BSTS and seasonal ARIMAX (SARIMAX). Here, since BSTS (resp. SARIMAX) is a univariate model, we applied it to one time series and then to another time series, after which we added their errors up in the error computation formula. Figure 2 provides a clear picture that a strong correlation between the two target series contributes to better forecast performances of the MBTS model. Figure 2a and b indicate that there is no significant difference among three models in terms of forecast errors, but Fig. 2c and d demonstrate that the higher the correlation in absolute value the better the prediction accuracy. Therefore, the MBTS model is a great choice to be applied in the multivariate analysis of several target series with strong correlations.

4 Application to real data

Here we are concerned with financial time series, and empirical time series of stock returns clearly contain uncertainties. With the help of some known contemporaneous series, our approach sheds light on machine learning based multivariate financial time series analysis. More specifically, it allows one to improve the understanding and prediction of multivariate financial time series (for instance, a portfolio of stock returns), which are extremely crucial to Wall Street practitioners for investment and/or risk management purposes. In the following, we train the MBTS model, do one-step and ten-steps ahead forecast respectively, and then compare its performance with three benchmark models: ARIMA, ARIMAX, and BSTS.

4.1 Data and predictors description

In this section, we analyze the data of three major stock market indices (Dow Jones Industrial Average, Nasdaq Composite Index, and S&P 500), using 23 economic leading

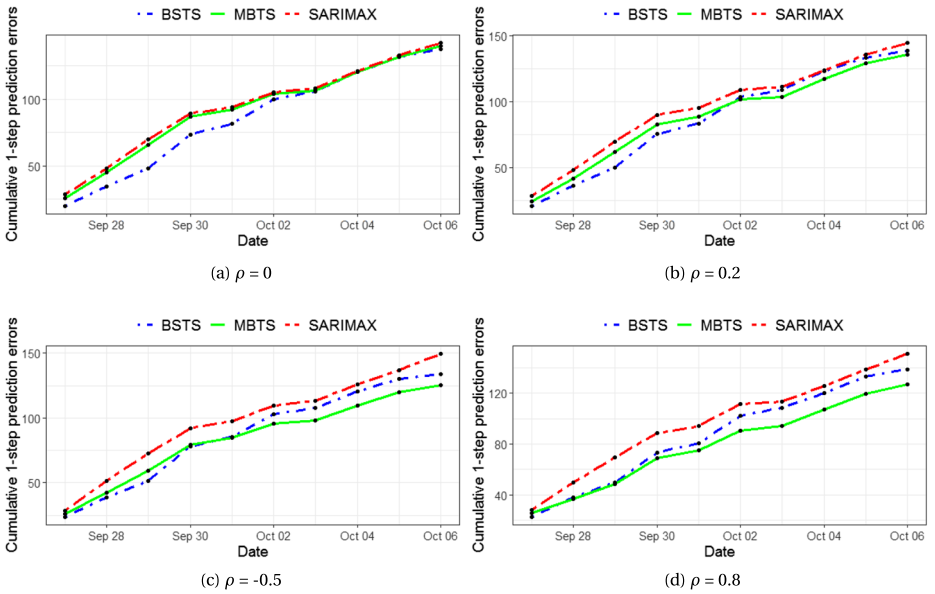


Fig. 2 The effect of correlation on cumulative forecast errors (CFEs). The black curve marked with ● represents the CFEs generated by the BSTS model. The blue curve marked with ▲ represents the CFEs generated by our MBTS model. The red curve marked with ■ represents the CFEs generated by the SARIMAX model

indicators and 27 domestic Google trends as predictors. Computed from the prices of representative stocks, these stock market indices are widely acknowledged measurements of specific sections of the stock market that is directly affected by economic conditions. The daily data samples from 03/05/2007 to 02/13/2018 are obtained from Google Finance, Yahoo! Finance, and Federal Reserve Economic Data (FRED) in the U.S. region. The dataset is split into training set (from 03/05/2007 to 01/16/2018), validation set (from 01/17/2018 to 01/30/2018), and test set (from 01/31/2018 to 02/13/2018). We note that the portion of data to allocate to these sets may vary among different empirical examples, while enough data samples should be used to train MBTS to guarantee its performance.

4.1.1 Target time series

It is crucial to keep an eye on the fluctuations of financial markets in order to beat the wisdom in Wall Street. In this study, we dig into the monthly returns y_t for three major stock market indices (Dow Jones Industrial Average, Nasdaq Composite Index, and S&P 500), whose values are mainly influenced by general economic activities. The daily close quotes are used to compute arithmetic monthly returns defined as

$$y_t = \frac{C_{t+30}}{C_t} - 1,$$

where C_t is the close quote for day t. Visualization of the daily values time series and corresponding monthly return y_t for Dow Jones Industrial Average, can be seen in Fig. 3.

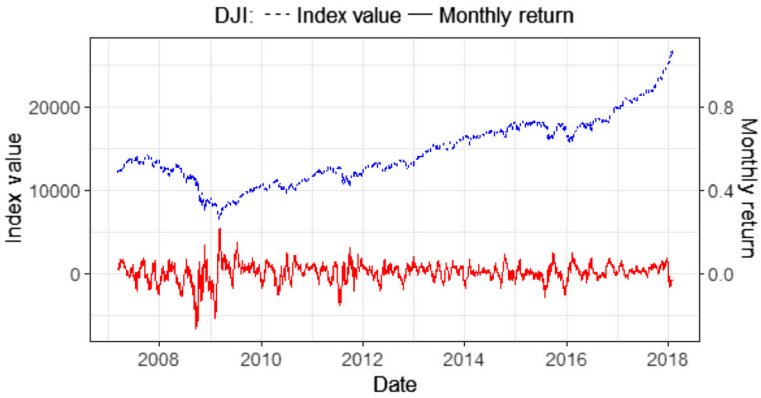


Fig. 3 Index values and monthly return of Dow Jones Industrial Average

4.1.2 Predictors

Well known, markets may incorrectly price a financial product in the short run but will eventually correct that mistake. Therefore, profits can be achieved by purchasing the undervalued product and then waiting for the market to recognize its "mistake" and bounce back to the fundamental value. Since macroeconomy has a significant effect on the financial market, economic analysis plays an important role in giving accurate stock return predictions, especially in forecasting values of stock market indices. Therefore, we collect 27 Google domestic trends and 23 economic leading indicators as predictors representing general economic conditions or trends in the U.S. market.

Google domestic trends, developed and maintained by Google since 2004 through collecting the daily volume of searches for related queries to a specific segment, allows users to look at various sectors of the U.S. economy based on how they are performing in Google's search index. [31] showed the existence of correlations between Google domestic trends and the equity market, which motivates us to use this trend data as a representation of the public interest in various macroeconomic factors. In this study, we include 27 domestic trends which are listed in Table 1 with their abbreviations. Economic leading indicators are measurable economic factors that change before the economy starts to follow a particular pattern or trend, and they are often used to predict changes in the economy. We select 23 important and popular economic leading indicators which are listed in Table 2 with their abbreviations.

4.2 Training results

Visually checking three target series, no obvious upward or downward trend is found, but a non-constant variance pattern with a stronger fluctuation at the beginning of the business cycle is detected. Therefore, we decide to include a local trend component without slope δ using (2.2) and a cycle component using (2.5). Through spectral analysis, the cycle period is determined to be 82 transaction days (about 4 months), and parameters for the cyclic component ϱ_{ii} for $i = 1, 2, 3$ are selected to be 0.98 based on prediction accuracy in the validation set. Then we ran the MCMC algorithm with 6000 iterations, and discarded the first 1000 as burn-in. It is worth noting that all predictors do not show obvious trends and most of them are stationary in the sense that their unit-root null hypotheses have p-values

Table 1 Google domestic trends

Trend	Abbr.	Trend	Abbr.
Advertising & marketing	advert	Air travel	airtvl
Auto buyers	auto	Auto financing	autoby
Automotive	autofi	Business & industrial	bizind
Bankruptcy	bnkrpt	Commercial Lending	comlnd
Computers & electronics	comput	Construction	constr
Credit cards	crcard	Durable goods	durble
Education	educat	Finance & investing	invest
Financial planning	finpln	Furniture	furntr
Insurance	insur	Jobs	jobs
Luxury goods	luxury	Mobile & wireless	mobile
Mortgage	mtge	Real estate	rlest
Rental	rental	Shopping	shop
Small business	smallbiz	Travel	travel
Unemployment	unempl		

less than 0.05 in the augmented Dickey-Fuller test (see [33]). The prior inclusion probability is set to 0.1 in that we expect model size to be 5 (= 50 predictors $\times 0.1$). We suppose that three stock market indices are affected by the same set of predictors during each MCMC iteration.

4.2.1 Feature selection

The "spike and slab" regression component in our model allows feature selection to be done during model training. It can reduce the model to user desired size by removing redundant variables and keeping important ones during each MCMC iteration, which in general prevent overfitting and solve the problem of collinearity. Besides, the posterior inclusion

Table 2 Economic leading indicators

Trend	Abbr.	Trend	Abbr.
Business confidence index	BCI	Consumer confidence index	CCI
Consumer expectation index	CEI	Average weekly working hours	AWH
Average hourly earnings	AHE	Unemployment rate	UR
Weekly jobless claims for unemployment insurance	JCU	Purchasing managers' index	PMI
Manufacturer's new orders for consumer goods	OCG	Interest rate spread	IRS
Manufacturers' new orders for all manufacturing industries	OMI	Bond yield spreads	BYS
Manufacturer's new orders for non-defense capital goods	OAG	Industrial production index	IPI
Building permits for new private housing units	BPH	Real retail sales	RRS
Level of new business startups	NBS	Money supply	M2S
Money zero maturity	MZM	Currency strength	CST
Diffusion index (New York)	DIN	Diffusion index (Texas)	DIT
Diffusion index (Philadelphia)	DIP		

probability can be used to indicate the importance of its corresponding predictor in explaining the fluctuation of the target series. A threshold can be set to select the most important variables for further analysis. Due to the lack of a closed-form expression of the posterior inclusion probability density function, we turn to empirical posterior inclusion probability computed.The chance that each predictor will be selected relies on (2.19).

In this study, we set the user desired model size to 6 out of a pool of 50 candidate predictors. From Fig. 4, we can see that the median model size is 5 in the sparse models generated by the sampling algorithm in each iteration, which matches our desired model size in the prior setting. That is to say, there are only five variables that contribute to the variation of monthly return for three indices significantly, with each predictor having different relationship with the target series. More specifically, increasing in OMI (manufacturers' new orders for all manufacturing industries) and OAG (manufacturer's new orders for non-defense capital goods) sends a positive signal for higher monthly returns of DJIA and S&P 500, whereas negative for that of Nasdaq. This might be due to the fact that these two predictors reveal new orders for manufacturing industries, but the composition of the Nasdaq is heavily weighted towards information technology companies. In addition, PMI (purchasing managers' index) has negative effects on three stock market indices. PMI provides information about current business conditions, and apparently high price values of indices at present are usually harder to achieve high future returns. BYS (bond yield spreads) pushes up monthly returns

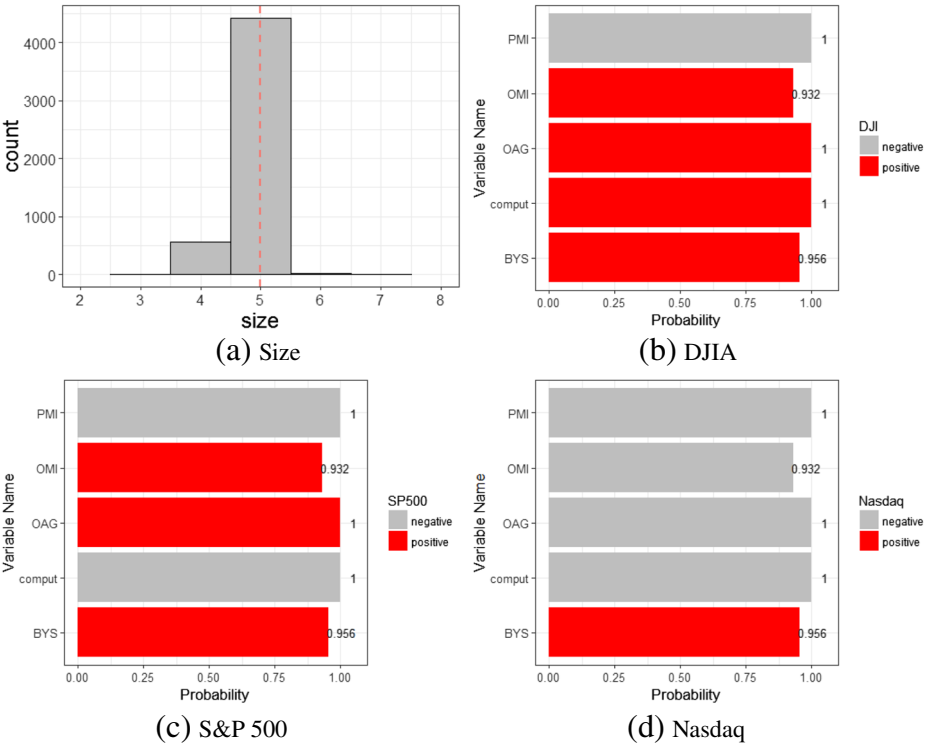
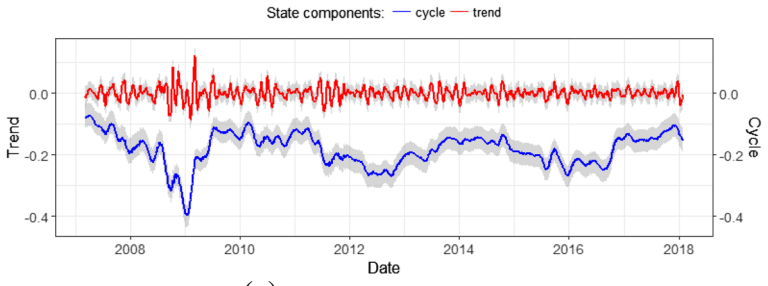


Fig. 4 **a** shows histogram of model size for 5000 MCMC iterations. **b**, **c**, and **d** display empirical posterior inclusion probabilities of important predictors for DJIA, S&P 500, and Nasdaq respectively, with colors indicating signs of estimated values

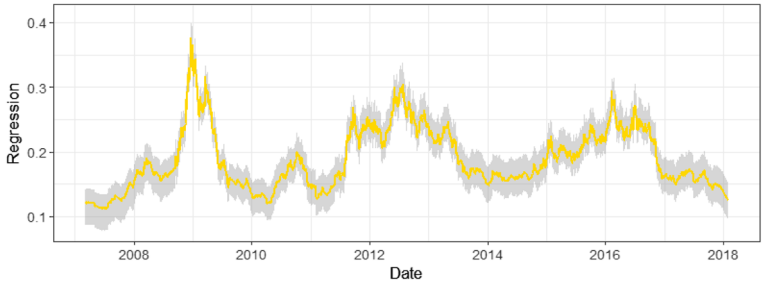
of these three indices. A higher BYS value indicates higher returns in the financial market, and positive relationships with these three indices returns can be easily understood. Out of 27 Google domestic trends, we only find "comput" (computers & electronics) is significant, and it has negative relationships with S&P 500 and Nasdaq, whereas positive with DJIA.

4.2.2 Contribution of components

In this section, we use DJIA as an example to further explore the relationship between the target series and its corresponding state components. The fitted target series is decomposed into three components: trend (local level in this example) component, cycle component and regression component, whose posterior distributions can be seen in Fig. 5 with shaded area indicating the 90% confidence bands based on MCMC draws. Compared with the other two components, the regression component has a wider confidence band due to larger dispersion of MCMC draws for regression coefficients B. The local level displays the long-run movements of target series with a nadir around 2009, while cyclical component captures several large variations from external shocks especially around 2009. As the effects of shocks diminish, the cyclical component shows relatively smoother fluctuation. As a whole, including these two components enable our approach to explain both short-term and long-term movements of the target series. In general, the regression component fluctuates at basically the same time as the cyclical component, however in the opposite direction and with more variation. Unlike the trend component, it shows several obvious peaks accounting for some variations not captured otherwise. In sum, decomposition of response series into three components provides us enough information on how each component contributes to explaining the variations in target series.



(a) Time series components



(b) Regression component

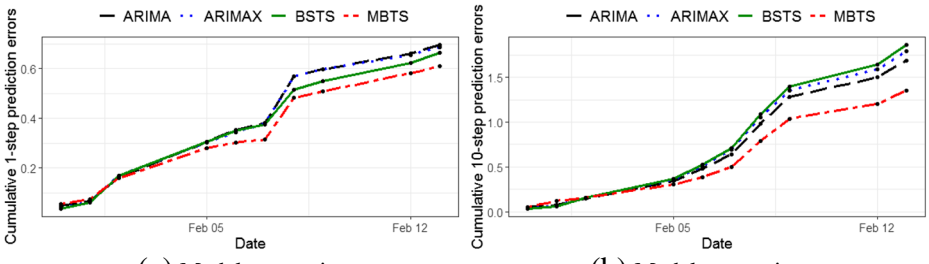
Fig. 5 Dynamic posterior distribution of three state components (DJIA)

4.3 Target series forecast

Time series forecasting is a model-based approach to predict future values based on previously observed values, and is extremely challenging when it comes to multivariate response series. Our approach allows users to model multiple target series with a great number of contemporaneous predictors and make a prediction of future values. In addition, Bayesian model averaging with spike-and-slab regression takes sparsity and model uncertainty into account, thus improves prediction accuracy. Based on cumulative one-step and ten-steps ahead prediction errors, we investigate forecast performances of our model and compare with the univariate BSTS model and two traditional benchmark time series models (ARIMA and ARIMAX). The prediction error at each step PE_t is defined by summing up each target series' absolute value of difference between true value $y_t^{(i)}$ and predicted value $\hat{y}_t^{(i)}$, i.e.

$$PE_t = \sum_{i=1}^{m} \left| y_t^{(i)} - \hat{y}_t^{(i)} \right|.$$

Figure 6a illustrates the cumulative one-step-ahead forecast errors using four different models (ARIMA, ARIMAX, BSTS, and MBTS) which all have short running times (2.19 seconds, 7.09 minutes, 4.29 minutes, and 4.35 minutes, respectively, in this empirical example). Clearly, the BSTS model and our MBTS model beat the traditional time series models, verifying the crucial role of the Bayesian machine learning setting in capturing variations of target series that can not be explained by other time series models. Precisely, on one hand, through feature selection, we can keep important predictors while drop redundant predictors from a pool of candidate predictors; on the other hand, Bayesian moving average provides great flexibility in that we commit neither to any particular set of predictors which avoids model uncertainty issues, nor to point estimates of regression coefficients which bypasses overfitting problems. Moreover, benefiting from the multivariate setting in fully incorporating the strong correlations among error terms of target series, our MBTS model shows much better forecast performances than the univariate BSTS model trained by each response series individually. Figure 6b illustrates the cumulative ten-steps-ahead forecast errors of four trained models, among which our MBTS model is still the best in terms of cumulative ten-steps-ahead forecast errors, which further confirms its obvious strength in time series forecasting. In contrast to the cases of one-step-ahead forecasting, all ten-steps-ahead predicted values over time show very similar patterns for these three indices and are much smoother without significant variation.



(a) Model comparison (b) Model comparison

Fig. 6 Comparison of four models' prediction accuracy

5 Conclusions

In this paper, we have proposed the MBTS model, which is a multivariate generalization of the BSTS model discussed earlier in the literature, to take advantage of the relationship among multiple target series, based on the assumption that each target series is affected by the same subset of predictors chosen from all candidate predictors. The "spike and slab" framework allows the model to do feature selection and model training at the same time. First, we examined the estimation accuracy and forecast performance using simulated data and found a positive relationship with correlations of error terms in the target time series. Then the empirical analysis on a portfolio of equity indices (DJIA, Nasdaq, and S&P 500), with feature selection among a pool of contemporary predictors including 23 leading economic indicators and 27 domestic Google trends, further confirmed that the MBTS model outperforms benchmark models such as pooling the results from univariate BSTS, ARIMA, and ARIMAX, in both one-step-ahead forecast and ten-steps-ahead forecast.

Our proposed methodology of feature selection on the regression component of time series can naturally extend the univariate analysis to multivariate analysis, extend the dimension of predictors by means of feature selection, and extend the corresponding analysis to the time-dependent case which is more realistic in many situations. Other extensions regarding methodology improvement may be seen from the following perspectives, such as adding on spatial analysis to generate a spatio-temporal framework (see [2]), boosting conditional probability estimators (see [15]), generalizing to universal probability-free prediction (see [38]), extending linear regression to nonlinear regression (see [22]), extending state space model (also named hidden Markov model) to hidden semi-Markov model (see [28]).

Acknowledgements We give special thanks to the journal editor and two anonymous reviewers who provided us with many constructive and helpful comments.

Funding The research of Ning Ning was partially supported by NSF grant DMS-1761603.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Arora, S., Barak, B.: Computational complexity: a modern approach. Cambridge University Press, Cambridge (2009)
2. Belussi, A., Migliorini, S.: A spatio-temporal framework for managing archeological data. Ann. Math. Artif. Intell. **80**(3-4), 175–218 (2017)
3. Bretó, C., He, D., Ionides, E.L., King, A.A., et al.: Time series analysis via mechanistic models. Annals Appl. Stat. **3**(1), 319–348 (2009)
4. Chen, B., Chen, L., Chen, Y.: Efficient ant colony optimization for image feature selection. Signal Process. **93**(6), 1566–1576 (2013)
5. Chen, M., Chen, Y., Weinberger, K.Q.: Automatic feature decomposition for single view co-training. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp. 953–960 (2011)
6. Chen, Y., Dong, G., Han, J., Wah, B.W., Wang, J.: Multi-dimensional regression analysis of time-series data streams. In: VLDB'02: Proceedings of the 28th international conference on very large databases, pp. 323–334. Elsevier, New York (2002)
7. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners. IEEE Trans. Signal Process. **50**(3), 736–746 (2002)
8. Cui, Z., Chen, W., Chen, Y.: Multi-scale convolutional neural networks for time series classification. arXiv:1603.06995 (2016)

9. Dong, Y., Tao, D., Li, X., Ma, J., Pu, J.: Texture classification and retrieval using shearlets and linear regression. IEEE Trans. Cybern. **45**(3), 358–369 (2014)
10. Douc, R., Moulines, E., Stoffer, D.: Nonlinear time series: theory, methods and applications with R examples. CRC Press, Boca Raton (2014)
11. Durbin, J., Koopman, S.J.: A simple and efficient simulation smoother for state space time series analysis. Biometrika **89**(3), 603–616 (2002)
12. Fan, J., Ma, C., Zhong, Y.: A selective overview of deep learning. arXiv:1904.05526 (2019)
13. George, E.I., McCulloch, R.E.: Approaches for bayesian variable selection. Stat. Sinica pp. 339–373 (1997)
14. Golumbic, M.C.: Algorithmic graph theory and perfect graphs, vol. 57. Elsevier, New York (2004)
15. Gutfreund, D., Kontorovich, A., Levy, R., Rosen-Zvi, M.: Boosting conditional probability estimators. Ann. Math. Artif. Intell. **79**(1-3), 129–144 (2017)
16. Harvey, A.C.: forecasting structural time series models and the Kalman filter. Cambridge University Press, Cambridge (1990)
17. Harvey, A.C., Jaeger, A.: Detrending, stylized facts and the business cycle. J Appl Economet **8**(3), 231–247 (1993)
18. Harvey, A.C., Koopman, S.J.M., Heij, C., Schumacher, H., Hanzon, B., Praagman, C.: Multivariate structural time series models Series in Financial Economics and Quantitative Analysis (1997)
19. Harvey, A.C., Trimbur, T.M., Van Dijk, H.K.: Trends and cycles in economic time series: a bayesian approach. J. Econ. **140**(2), 618–649 (2007)
20. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: a tutorial. Stat. Sci. pp. 382–401 (1999)
21. Hou, C., Nie, F., Li, X., Yi, D., Wu, Y.: Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Trans. Cybern. **44**(6), 793–804 (2013)
22. Kuleshov, A., Bernstein, A.: Nonlinear multi-output regression on unknown input manifold. Ann. Math. Artif. Intell. **81**(1-2), 209–240 (2017)
23. Li, W., Wang, Z., Ho, D.W.C., Wei, G.: On boundedness of error covariances for kalman consensus filtering problems IEEE Transactions on Automatic Control (2019)
24. Li, X., Zhang, H., Zhang, R., Liu, Y., Nie, F.: Generalized uncorrelated regression with adaptive graph for unsupervised feature selection. IEEE Trans. Neural Netw. Learn. Syst. **30**(5), 1587–1595 (2018)
25. Liao, T.W.: Clustering of time series data—a survey. Patt. Recog. **38**(11), 1857–1874 (2005)
26. Madigan, D., Raftery, A.E.: Model selection and accounting for model uncertainty in graphical models using occam's window. J. Am. Stat. Assoc. **89**(428), 1535–1546 (1994)
27. Mamon, R.S., Elliott, R.J.: Hidden Markov models in finance, vol. 4. Springer, New York (2007)
28. Narimatsu, H., Kasai, H.: State duration and interval modeling in hidden semi-markov model for sequential data analysis. Ann. Math. Artif. Intell. **81**(3-4), 377–403 (2017)
29. Pang, T., Nie, F., Han, J., Li, X.: Efficient feature selection via $\ell_{2,0}$-norm constrained sparse regression. IEEE Trans. Knowl. Data Eng. **31**(5), 880–893 (2019)
30. Petris, G., Petrone, S., Campagnoli, P.: Dynamic linear models. Dynamic Linear Models with R. pp. 31–84 (2009)
31. Preis, T., Moat, H.S., Stanley, H.E.: Quantifying trading behavior in financial markets using google trends Scientific reports 3:srep01684 (2013)
32. Qiu, J., Liu, W., Ning, N.: Evolution of regional innovation with spatial knowledge spillovers: Convergence or divergence? Netw. Spatial Econ. pp. 1–30 (2019)
33. Said, S.E., Dickey, D.A.: Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika **71**(3), 599–607 (1984)
34. Scott, S.L., Varian, H.R.: Predicting the present with bayesian structural time series. Int. J. Math. Model. Numer. Opt. **5**(1-2), 4–23 (2014)
35. Scott, S.L., Varian, H.R.: Bayesian variable selection for nowcasting economic time series. In: Economic analysis of the digital economy, pp. 119–135. University of Chicago Press, Chicago (2015)
36. Su, Y., Gao, X., Li, X., Tao, D.: Multivariate multilinear regression. IEEE Trans. Syst. Man Cybern., Part B (Cybernetics) **42**(6), 1560–1573 (2012)
37. Vincent, L.E., Thome, N.: Shape and time distortion loss for training deep time series forecasting models. In: Advances in neural information processing systems, pp. 4191–4203 (2019)
38. Vovk, V., Pavlovic, D.: Universal probability-free prediction. Ann. Math. Artif. Intell. **81**(1-2), 47–70 (2017)
39. Yao, C., Han, J., Nie, F., Xiao, F., Li, X.: Local regression and global information-embedded dimension reduction. IEEE Trans Neural Netw. Learn. Syst. **29**(10), 4882–4893 (2018)

40. Zhang, H., Zhang, R., Nie, F., Li, X.: A Generalized Uncorrelated Ridge Regression with Nonnegative Labels for Unsupervised Feature Selection. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, pp. 2781–2785 (2018)
41. Zhang, R., Nie, F., Li, X.: Feature selection under regularized orthogonal least square regression with optimal scaling. Neurocomputing **273**, 547–553 (2018)