

Improve Forecasting using Bayesian Structural Time Series with Google Trends data

Edoardo Marcelli

10/09/2021

Abstract

Bayesian Structural Time Series models are an innovative tool for time series forecasting. They build on State Space models and thanks to a Spike and Slab regression component they are able to deal with high-dimensional data. For this reason, we choose them to predict US Unemployment rate using a large set of predictors composed by Google Trends data. Forecasting performances are compared to the ones obtained by ARIMAX models that will be used as benchmark. We find that overall including Google data improves predictions relative to competing models that does not employ search data queries. Furthermore, BSTS slightly overperform ARIMAX and they better quantify forecast uncertainty.

1 Introduction

In decision making, the more information we got the better it is. However, there may be several reasons why the information we need is not available on time. It takes time to gather, revise and report data, therefore some important statistics are released with time lag. In business this happens with some kind of Key Performance Indicator (KPI) which measure business performance after the fact, such as sales, customer satisfaction, or revenue churn. Whereas, in finance and economics, this is the case of official statistics such as GDP, inflation and unemployment rate. For this reason, in the last decade the concept of “nowcasting” has been introduced in economics. This term, originated in meteorology, refers to the problem of predicting the present, the very near future, and the very recent past state of an economic or business indicator. Many models have been developed so far to anticipate the values of indicators. These models usually rely on large amounts of data at different frequencies and with different publication lags. Data sources can be for instance business leading indicators, financial time series or social media contents.

In this work, we decided to focus on prediction economic time series using Google Trends data. The use of Google Trends as an effective tool for nowcasting have been demonstrated by Choi and Varian (2009, 2012). From that moment several authors have applied this intuition to predict time series in marketing (Bangwayo-Skeete 2015, Boone et al 2017), finance (Preis 2013, Hamid 2015), economics (Naccarato 2018, Woloszko 2021) and several other fields. Very briefly, Google Trends is a search trends feature provided by Google that shows how frequently a given search term is entered into Google’s research engine, relative to the site’s total search volume over a given period of time. In other words, data are transformed such that 100 identify the highest frequency of the searched term, while 0 the lowest. Google Trends are particularly attractive because they provide large amounts of data at zero costs, however this also means that we have to cope with the problem of having a very high dimensional set of predictors. To

this purpose, Scott and Varian (2013) have developed Bayesian Structural Time Series (BSTS) models with Spike and Slabs priors for variable selection. BSTS models are a part of the wider class of State Space models. Thanks to their flexibility this models are able to capture the trend, seasonal and similar components of the target and accommodate for static and dynamic regression components. Therefore, they provide a valid choice for modeling time series when the number of potential predictors is large.

Bayesian Structural Time Series models have been used in several scientific areas, from the medical field to finance and marketing. We focused on a macroeconomic application and we will try to predict, or better nowcast, the US unemployment rate in the United States of America. The big challenge of this work is to show how Google Trends can, under some circumstances, significantly improve forecasts. Then we will compare the results obtained using BSTS with those of ARIMA models, which represents the most classical approach to time series analysis. We want to understand which are the strength and the weakness of the two approaches and whether and when one is preferable to the other.

Operationally, in Section 2 we provide the main idea behind Classical Time Series Decomposition which represents the building blocks of new modern time series analysis approaches such as BSTS. Then, in Section 3, we will briefly review ARIMA models, a pillar in time series analysis and for this reason it will be used as the benchmark to assess the goodness of BSTS models. Section 4 will be dedicated to State Space Models, here we will provide the preliminaries for the discussion of Bayesian Structural Time Series Models that will follows in Section 5. Finally, Sections 6, 7 and 8 will be dedicated to the case study of predicting the US Unemployment rate shock caused by the pandemic crisis.

2 Classical Time Series Decomposition

Time series decomposition represents a starting point for time series analysis. It is based on the simple idea that any observed time series contains systematic components (trend, seasonality, cycle) and a random component (or noise) such that if we were able to identify and isolate each component we would be able to predict the future development of the series. More formally we have

$$y_t = f(T_t, S_t, C_t, E_t) \quad (1)$$

where

- T_t , the trend, is the general direction in which the time series is moving.
- S_t , the seasonal component, refers to short term fluctuations that occur regularly (usually monthly or quarterly).
- C_t , the cycle, is a long term fluctuation that occurs regularly. Note that usually the identification of a cycle requires a long time series that can go from two to twenty or thirty years. For this reason it is quite difficult to detect it in shorter analysis and it might be confused with the trend, therefore it is common to consider C_t and T_t together as the trend-cycle component.
- E_t , the residual, is the unexplained component of variation. We assume that after having identified the main components of the series what remains is characterized by the absence of any type of predictable structure and can be considered noise.

The way these components interact with each other twofold. In general, if the behaviors of the components are independent from each other, for example the magnitude of the seasonal component does not vary with the level of the time series, then an «Additive»model, $y_t = T_t + S_t + C_t + E_t$, would be more appropriate. Otherwise, when the components show some dependence, we would prefer a «Multiplicative»model, $y_t = T_t \cdot S_t \cdot C_t \cdot E_t$. Nevertheless, note that we can easily switch from a representation to the other one by simply use a logarithmic transformation. For this reason, without loss of generality we will deal only with «Additive»models in this work.

At this stage, in the Classical Decomposition approach, a common way to proceed is through a graphical analysis to visualize the features of the data including patterns, unusual observations and eventual structural breaks. Once we have identified the components we step into the estimation phase. The trend can be estimated by fitting a smooth function of the time t , i.e. $T_t = f(t) + \epsilon_t$. For instance we can assume a linear trend, $T_t = \beta_0 + \beta_1 t + \epsilon_t$, but also a quadratic trend, $T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$, or a exponential trend, $T_t = \beta_0 e^{\beta_1 t}$. In the first and easiest case we use the OLS to estimate the coefficients and obtain $\hat{T}_t = \hat{\beta}_0 + \hat{\beta}_1 t$. In alternative, a moving average can be as effective to estimate the trend, $\hat{T}_t = \frac{1}{m} \sum_{i=1}^m y_t$. Once the trend has been estimated, we detrend the series and obtain $Y_t^\Delta = Y_t - \hat{T}_t$. Then, in order to estimate the seasonal component for each “time”, we may simply average the detrended values in each seasonal period and obtain \hat{S}_t . Finally we obtain residually the error component, $E_t = Y_t - \hat{T}_t - \hat{S}_t$ and we check that it does not present any structure. If the latter condition holds, it means that we have successfully identified the main components. Therefore we can forecast the decomposed time series by forecasting each component separately, which is an easy task since we assume that the evolution of the latter is stable over time.

Despite its simplicity and its poor forecasting performances due to the strong assumptions behind this approach, Classical Decomposition is still a landmark for many forecasting techniques. In particular, the idea of time series as a aggregate of components is still used nowadays by many modern time series models including Bayesian Structural Time Series models.

3 ARIMA models

ARIMA models represents the most standard approach to time series analysis and forecasting. Tons of literature have been written on this class of models and in this section we will provide just a brief review. The basic idea of ARIMA models is that the future is rather similar to the past, assumption which implies the series to be weakly stationary. A time series is said to be weakly stationaly when it has stable mean and variance over time, and the correlation of the variable with each of its own lags is also stable over time. In other words, we want a series such that:

$$\begin{aligned} E(y_t) &= E(y_{t+h}) \forall t, h \\ V(y_t) &= V(y_{t+h}) \forall t, h \\ Cov(y_t, y_{t-m}) &= Cov(y_{t+h}, y_{t-m+h}) \forall t, m, h \end{aligned}$$

A weakly stationary process can always be represented as

$$y_t = \sum_{i=0}^{\infty} c_i \epsilon_{t-i} = \sum_{i=0}^{\infty} c_i L^i \epsilon_t = c(L) \epsilon_t \quad (2)$$

where L is the lag operator such that $L^i \epsilon_t = \epsilon_{t-i}$ and the error process is a white noise process with mean zero and constant variance, i.e. $\epsilon_t \sim WN(0, \sigma^2)$. However, because of the infinite number of parameters c_i to be estimated, we approximate $c(L)$ with a ratio of two finite polynomials

$$c(L) = \frac{\psi(L)}{\phi(L)}$$

where $\psi(L) = 1 - \psi_1 L - \dots - \psi_q L^q$ and $\phi(L) = 1 - \phi_1 L - \dots - \phi_p L^p$. Note that $\phi(L)$ is invertible thanks to stationarity. Therefore we can rewrite the previous process as

$$\phi(L) y_t = \psi(L) \epsilon_t \quad (3)$$

which is the compact ARMA representation for

$$y_t = \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t - \psi_1 \epsilon_{t-1} - \dots - \psi_q \epsilon_{t-q}$$

the latter is moving average autoregressive process of order p and q , i.e. $ARMA(p, q)$.

Applying ARMA model to non-stationary series is possible to integration. An integrated process y_t is a non stationary process such that $(1 - L)^d y_t$ is stationary, where d is the order of integration. A process of order d is called $I(d)$. If we consider an integrated process $I(d)$ in the context of ARMA models we would end up with an $ARIMA(p, d, q)$ process

$$\phi(L) \Delta^d y_t = \psi(L) \epsilon_t$$

where $\Delta^d \equiv (1 - L)^d$. Therefore, the only difference with the $ARMA(p, q)$ model is that y_t has been replaced by an integrated process.

Finally, ARIMA models can be enriched with exogenous variables, this models are called ARIMAX models. This explicative variables can provide additional information and hence improve predictions. Notice that y_t and x_t have both to be stationary, hence it may be necessary differencing both dependent and independent variables before fitting an ARIMA model. An ARIMAX model can be written as

$$y_t = \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t + \beta_1 x_1 + \dots + \beta_n x_n - \psi_1 \epsilon_{t-1} - \dots - \psi_q \epsilon_{t-q}$$

3.1 Model Specification

There are three main approaches to identify (p, d, q) . The first one is based on the identification of the peaks in the Autocorrelation (AC) and Partial Autocorrelation (PAC) functions. The first one is defined as

$$AC(h) = \frac{Cov(y_t, y_{t-h})}{\sqrt{V(y_t)} \sqrt{V(y_{t-h})}}$$

while the PAC is

$$AC(h) = \frac{Cov(y_t, y_{t-h} | y_{t-1}, \dots, y_{t-h+1})}{\sqrt{V(y_t | y_{t-1}, \dots, y_{t-h+1})} \sqrt{V(y_{t-h} | y_{t-1}, \dots, y_{t-h+1})}}$$

They can be easily computed. If they both decay when h increases, this provides evidences that the series is stationary and therefore we set $d = 0$. If the AC decay slowly and $PAC(1)$ is almost 1, then there is evidence for a unit root and we will differentiate the series. If necessary we further differentiate it. Note that in seasonal time series, taking seasonal difference can be enough to obtain a stationary time series.

Moreover, if the PAC presents some, say p , peaks and it is close to zero everywhere else, while the AC declines exponentially, this signal a pure $AR(p)$ process. While, if instead the AC has q peaks whereas the PAC decays exponentially, we probably have a pure $MA(q)$ process. However in the case of ARMA model, when we expect to have both processes, we usually guess some possible models from the observed peaks in the AC and PAC, but then we should also use further techniques.

A useful quantitative diagnostic tool is the Ljung-Box Q test which tests whether a number of (residual) autocorrelations are equal to zero. We indeed expect that the estimated errors of our fitted model are temporally uncorrelated. The test statistic is

$$Q^* = T(T+2) \sum_{k=1}^h \frac{\hat{\rho}^2}{T-k}$$

where $\hat{\rho}^2$ is the sample autocorrelation at lag k while h is the number of lags being tested. Under the null hypothesis the test statistic is $\chi^2_{(h)}$ distributed. Note that when we use the Ljung-Box Q test on the residuals of a fitted $ARMA(p, d, q)$, the actual degrees of freedom should be set to $h - p - q$, to reflect the fact that parameters have been estimated.

Finally, we use information criterion. These criterion combine model goodness of fit with a penalty function linked to the numerosity of model parameters such that parsimonious models are preferred to larger one when the fit is similar. The idea is to chose the model with the lower information criteria (AIC, BIC) since all the models can be written in the form

$$C = -2L(\xi) + c(k, T)$$

where k is the number of estimated parameters and T the sample size. Since $c(k, T)$ is increasing in k , being equal the likelihood, the more parsimonious model is choose.

3.2 Estimation and Forecasting

The parameters of ARIMA models are estimated by Maximum Likelihood. Given a parameter vector ξ and assuming a Gaussian process, the log likelihood to be maximized is

$$l(\xi) = -\frac{T}{2} \log(2\pi) - \frac{1}{2} [\log|\Sigma| + (y - \mu)' \Sigma^{-1} (y - \mu)]$$

This likelihood is usually maximized with numerical optimization methods. It is possible to show that Maximum Likelihood Estimators (MLE) of Gaussian ARMA processes are consistent, asymptotically normal and asymptotically efficient. Furthermore, under mild conditions, this

properties hold also when the true distribution is not normal.

Once we have estimated the parameters, we can finally proceed to forecasting. Given an information set at a time t , i.e. \mathcal{J}_t , the optimal forecast in the mean squared forecast error sense is

$$\hat{y}_{t+h} = E(y_{t+h} | \mathcal{J}_t)$$

Here we present the more basic idea. The idea behind optimal forecast for an $ARIMA(p, d, q)$ is the following. Without loss of generalization, we will consider an already integrated $ARMA(p, q)$

$$y_t = \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t - \psi_1 \epsilon_{t-1} - \dots - \psi_q \epsilon_{t-q}$$

from

$$y_{t+1} = \phi_1 y_t + \dots + \phi_p y_{t-p+1} + \epsilon_{t+1} - \psi_1 \epsilon_t - \dots - \psi_q \epsilon_{t-q+1}$$

it follows

$$\hat{y}_{t+1} = E(y_{t+1} | \mathcal{J}_t) = \phi_1 y_t + \dots + \phi_p y_{t-p+1} - \psi_1 \epsilon_t - \dots - \psi_q \epsilon_{t-q+1}$$

and similarly for $t+2, \dots, t+h$

$$\hat{y}_{t+h} = E(y_{t+h} | \mathcal{J}_t) = \phi_1 \hat{y}_{t+h-1} + \dots + \phi_p \hat{y}_{t-p+h} - \psi_h \epsilon_t + \dots + \psi_q \epsilon_{t-q+h}$$

4 State Space Models

State space models are a class of stochastic models that describe the probabilistic dependence between the latent state variable and the observed measurement. They originated in the early sixties in the field of control engineering and then they found applications in many other areas such as time series analysis. They are characterized by a much greater degree of flexibility with respect to their frequentist counterpart, i.e. ARIMA models. State space models can indeed be used to model both univariate and multivariate time series in presence of non-stationarity, structural breaks and irregular patterns.

State space models rely on the main idea that the observed time series $(Y_t)_{t \geq 0}$ is generated by a latent state process $(\theta_t)_{t \geq 0}$ which evolves as a Markov Chain and if we could measure it then the observations would be independent. More formally, we define two discrete time processes $\{Y_t\} := (Y_t)_{t \geq 0}$ and $\{\theta_t\} := (\theta_t)_{t \geq 0}$ taking values respectively in spaces \mathcal{Y} and Θ that can be multi-dimensional Euclidean spaces, discrete spaces or also less standard. Such series have to satisfy the following two conditions:

- (A.1) (θ_t) is a Markov Chain
- (A.2) Conditional on (θ_t) , the Y_t 's are independent and Y_t depends on (θ_t)

The consequence of such assumptions is that any state space model is completely specified by initial distribution p_0 , the transition distribution $p_t(\theta_t | \theta_{t-1})$ and the emission distribution $f_t(y_t | \theta_t)$. In other words, for any $t > 0$,

$$p(\theta_{0:t}, y_{1:t}) = p_0(\theta_0) \prod_{j=1}^t f_j(y_j | \theta_j) p_j(\theta_j | \theta_{j-1}) \quad (4)$$

This conditional independence structure can be graphically represented by a Direct Acyclic Graph (DAG)

$$\begin{array}{ccccccccccc}
\theta_0 & \longrightarrow & \theta_1 & \longrightarrow & \theta_2 & \longrightarrow & \cdots & \longrightarrow & \theta_{t-1} & \longrightarrow & \theta_t & \longrightarrow & \theta_{t+1} & \longrightarrow & \cdots \\
& & \downarrow & & \downarrow & & & & \downarrow & & \downarrow & & \downarrow & & \\
& & Y_1 & & Y_2 & & & & Y_{t-1} & & Y_t & & Y_{t+1} & &
\end{array}$$

Notice that from Equation (1) one can derive, by conditioning or marginalization, any other distribution of interest.

Similarly, a State Space model can be seen as a model that relates the variables θ_t and Y_t to sequences of independent “shocks” or “noise terms”:

$$Y_t = f_t(\theta_t, \epsilon_t) \quad (5)$$

$$\theta_t = g_t(\theta_{t-1}, \eta_t) \quad (6)$$

with v_t and w_t independent across time, $(\epsilon)_t \perp\!\!\!\perp (\eta)_t \perp\!\!\!\perp \theta_0$ and $\theta_0 \sim p_0$. Equation (1) is called the observation equation, because it links the observed data with the unobserved latent state, while Equation (2) is called the state (or transition) equation because it defines how the latent state evolves over time.

Within this huge class of models we focus on a subset of linear Gaussian models also known as Dynamic Linear Model (DLM). A Dynamic Linear model is indeed specified by a Normal prior distribution for the p -dimensional state vector at time $t = 0$

$$\theta_0 \sim N_p(m_0, C_0)$$

together with a pair of equations for each time $t \geq 1$

$$Y_t = Z_t \theta_t + \epsilon_t, \quad \epsilon_t \sim N_m(0, H_t) \quad (7)$$

$$\theta_t = T_t \theta_{t-1} + R_t \eta_t, \quad \eta_t \sim N_m(0, Q_t) \quad (8)$$

where Z_t, T_t and R_t are known matrices, usually containing values 0 or 1, but sometimes they might also contain unknown parameters. $(\epsilon_t)_{t \geq 1}$ and $(\eta_t)_{t \geq 1}$ are two independent sequences of independent Gaussian random vectors with mean zero and variance matrices $(H_t)_{t \geq 1}$ and $(Q_t)_{t \geq 1}$. The latter matrices, usually unknown, can be estimated through maximum likelihood or bayesian inference as we will see in the discussion of Bayesian Structural Time Series.

4.1 Filtering and Forecasting

The linear Gaussian nature of this kind of model allows to solve a very important issue in state space model analysis, i.e. the problem of filtering. Very briefly filtering means to estimate the true value of some system from an incomplete, potentially noisy set of observations on that system. According to the notation just introduced, we have knowledge of the latent process $(\theta_t)_{t \geq 0}$, also known as signal in the engineering literature, only indirectly through the observed process $(Y_t)_{t \geq 0}$, which depends on the states but also on a sequence of independent “shocks” or “noise terms”. For this reason we need a filter, i.e. a device that suppresses unwanted components or features from the signal. Considering a time series with streaming data, the goal is to estimate the state value at time t given all the available information at that time. Being θ_t a random

vector, a full description of it is provided by the conditional density $p(\theta_t|y_{1:t})$. Thanks to the Markovian structure of the state dynamics and the assumptions on the conditional independence for the observables described above (see (A.1) and (A.2)), the filtered and predictive densities can be computed recursively starting from $\theta_0 \sim p_0$. Even though there are several available filtering strategies, the overall idea can be summarized in few steps.

Proposition 4.1 *For a general state space model defined by (A.1) and (A.2), the following statements hold.*

- (i) *The one-step-ahead predictive density for the states can be computed from the filtered density $p(\theta_{t-1}|y_{1:t-1})$ according to*

$$p(\theta_{t-1}|y_{1:t-1}) = \int p(\theta_t|\theta_{t-1})p(\theta_{t-1}|y_{1:t-1}) d\theta_{t-1}$$

- (ii) *The one-step-ahead predictive density for the observations can be computed from the predictive density for the states as*

$$p(y_t|y_{1:t-1}) = \int f(y_t|\theta_t)p(\theta_t|y_{1:t-1}) d\theta_t$$

- (iii) *The filtering density can be computed from the above densities as*

$$p(\theta_t|y_{1:t}) = \frac{f(y_t|\theta_t)p(\theta_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$

In the linear Gaussian case, the filtering and predictive distribution are available in closed form. Indeed, since the random vector $(\theta_0, \theta_1, \dots, \theta_t, Y_1, \dots, Y_t)$ has indeed a multivariate Gaussian distribution for any $t \geq 1$, it follows that the marginal and conditional distributions are Gaussian as well and therefore it is sufficient to compute their means and variances to have a full description of them.

5 Bayesian Structural Time Series

Developed by Steven L. Scott and Hal Varian, respectively Senior Economist Analyst and Chief Economist at Google Inc., Bayesian Structural Time Series were born with the aim of improving short term forecasting by using an ensemble method that averages over different combinations of predictors.

The baseline model is the dynamic linear model of Section 4 enriched by structural components and regression components. Therefore the state vector is $\theta_t = (\mu_t, \delta_t, \tau_t)'$, where μ_t is the current level of the trend, δ_t is the time-varying slope of the trend and τ_t is the seasonal component. The latter can be thought of as a set of S dummy variables with dynamic coefficients constrained to have zero expectation over a full cycle of S seasons. The dynamic of the system

is the following

$$y_t = \mu_t + \tau_t + \beta^T x_t + \epsilon_t \quad (9)$$

$$\mu_t = \mu_{t-1} + \delta_{t-1} + u_t \quad (10)$$

$$\delta_t = \delta_{t-1} + v_t \quad (11)$$

$$\tau_t = - \sum_{s=1}^{S-1} \tau_{t-s} + w_t \quad (12)$$

Where $x_t = (x_{1,t}, \dots, x_{n,t})$ is the vector of a contemporaneous set of search queries or trends verticals, including any desired lags or other transformations.

The unknown parameters are the variance-covariance matrix of $\eta_t = (u_t, v_t, w_t)'$, where $\eta_t \sim N_3(0, Q_t)$ with $Q_t = \text{diag}(\sigma_u^2, \sigma_v^2, \sigma_w^2)$, the variance-covariance matrix of $\epsilon_t \sim N(0, H_t)$ where H_t is a positive scalar we will call for convenience σ_ϵ^2 , and the regression coefficient β . We will assign a prior to any of these parameters and update it as new observations come up.

Forecasts are provided by simulating from the posterior predictive distribution $p(\tilde{y}|y_{1:t})$ where $\tilde{y} = (y_{t+1}, \dots, y_{t+h})$, for instance for the one-step ahead forecast our quantity of interest will be $p(y_{t+1}|y_{1:t})$. Notice we can rewriting it as

$$p(y_{t+1}|y_{1:t}) = \int_{\Psi} p(y_{t+1}|\phi) p(\phi|y_{1:t}) d\phi \quad (13)$$

where $\phi = (\sigma_\epsilon^2, \sigma_u^2, \sigma_v^2, \sigma_w^2, \beta, \theta_{1:t})$. Simulation from this posterior distribution is easy once we have draws of model parameters and state from their posterior distribution. Markov Chain Monte Carlo methods are used in order to obtain such draws. An illustration of these methods will take place in Section 5.2, in the following section instead we will discuss how to handle time series regression with a huge amount of predictors.

5.1 Spike and Slab regression

As we have seen in the previous section, BSTS models allow for regression components. In the following discussion we will focus on static regression components, however, the model can also accommodate for a (small) set of dynamic regressors. As the Scott and Varian (2013) notice, there are several possible ways to insert a static regression component in a state space model. Among them, the authors simply append a constant 1 to each θ_t and $\beta^T x_t$ to Z_t in the observation equation. In this way the dimension of the state vector just increases by one, regardless the number of predictors. This allows great computational savings: the computational complexity of the Kalman filter is indeed linear in the length of the data and quadratic in the dimension of the state vector. This is also the reasons why the authors recommend not to use a large set of dynamic regressors: the coefficient of the latter would indeed enter into the state vector, increasing computational efforts.

Since the number of predictors we consider is potentially huge and, above all, much larger than the number of observations ($p \gg T$, where p is the number of predictors $(x_{1,t}, \dots, x_{p,t})$ and T the temporal observations (y_1, \dots, y_T)), then estimation problems arise. For this reason the authors introduced a famous Bayesian variable selection technique known as “spike and slab regression”, that introduce sparsity among the coefficients of the regression. It is indeed safe to

assume that many predictors do not play any role in the regression, and therefore they have to be shrunk to zero.

We introduce an indicator γ_t assuming two values

$$\begin{aligned}\gamma_t &= 1 \quad \text{if } \beta_k \neq 0 \\ \gamma_t &= 0 \quad \text{if } \beta_k = 0\end{aligned}$$

where β_k is the coefficient of the k -th predictor. Let β_γ be the subset of β where $\beta_k \neq 0$. A spike-and-slab can be written

$$p(\beta, \gamma, \sigma_\epsilon^2) = p(\beta_\gamma | \gamma, \sigma_\epsilon^2) p(\sigma_\epsilon^2 | \gamma) p(\gamma)$$

The prior distribution associated with the indicator is a Bernoulli

$$\gamma \sim \prod_{k=1}^K \pi_k^{\gamma_k} (1 - \pi_k)^{1-\gamma_k}$$

That can be simplified by assuming that the probability of the indicator to be 0 or 1 is the same for every predictor and hence $\pi_k = \pi$. Moreover one simple strategy to set the value of π is considering $\pi = k/p$, wondering which is the “expected model size”, i.e. how many coefficients k we expect to be non null over all the coefficient p . Concerning the other two parameters, i.e. β and σ_ϵ^2 , we model them using a convenient conditional Normal-Gamma prior distribution

$$\begin{aligned}\beta_\gamma | \gamma, \sigma_\epsilon^2 &\sim N(b_\gamma, \sigma_\epsilon^2 (\Omega_\gamma^{-1})^{-1}) \\ \frac{1}{\sigma_\epsilon^2} | \gamma &\sim G\left(\frac{df}{2}, \frac{ss}{2}\right)\end{aligned}$$

where b_γ is a vector of prior guesses for non null predictors, Ω is symmetric matrix and Ω_γ denote the rows and columns of Ω for which $\gamma_k = 1$. On the other hand, df and ss are the prior sample size and the prior sum of squares. Their values might be set by asking the researcher for a guess on the expected R2 of the regression and the number of observations worth of weight. Finally, for effective shrinkage, a possible prior distribution for $\beta_\gamma | \gamma, \sigma_\epsilon^2$ is the “g-prior” proposed by Zellner (1983). The latter is characterized by a null vector b_γ and $\Omega^{-1} = kX^T X/n$.

The package built in R by the authors (Package “bsts”, Scott (2021)) has embedded the prior and slab regression with a default g-prior and default hyperparameters $R^2 = 0.5$, $df = 0.01$ and $\pi_k = 0.5$. For empirical applications, we are going to use this default parameters in this work.

Posterior distribution can be easily computed thanks to the conjugate prior results given by the specification of the model. Starting from a Normal-Gamma prior, the update process is known and it ends up in a Normal-Gamma posterior with parameters

$$\begin{aligned}\beta_\gamma | \gamma, \sigma_\epsilon^2, y^* &\sim N(\tilde{\beta}_\gamma, \sigma_\epsilon^2 (V_\gamma^{-1})^{-1}) \\ \frac{1}{\sigma_\epsilon^2} | \gamma, y^* &\sim G\left(\frac{N}{2}, \frac{SS_\gamma}{2}\right)\end{aligned}$$

where

$$\begin{aligned} V_\gamma^{-1} &= (X^T X)_\gamma + \Omega_\gamma^{-1} \\ N &= df + n \\ \tilde{\beta}_\gamma &= (V_\gamma^{-1})^{-1} (X_\gamma^T y^* + \Omega_\gamma^{-1} b_\gamma) \\ SS_\gamma &= ss + y^{*T} y^* + b_\gamma^T \Omega_\gamma^{-1} b_\gamma - \tilde{\beta}_\gamma^T V_\gamma^{-1} \tilde{\beta}_\gamma \end{aligned}$$

where $y_t^* = y_t - Z_t^* \theta_t$ and Z_t^* being the observation matrix of equation 7 with the regression component $\beta^T x_t$ set to zero. Note that $y^* = (y_1^*, \dots, y_t^*)$.

Finally, thanks to the aforementioned conjugacy priors results, it is possible to marginalize out β_γ and σ_ϵ^2 from the full posterior distribution $p(\beta_\gamma, \sigma_\epsilon^2, \gamma)$ and obtain the marginal posterior distribution of the indicator

$$\gamma|y^* \sim C(y^*) \frac{|\Omega_\gamma^{-1}|^{\frac{1}{2}} p(\gamma)}{|V_\gamma^{-1}|^{\frac{1}{2}} SS_\gamma^{\frac{N}{2}-1}}$$

where $C(y^*)$ is a normalizing constant that depends on y^* but not on γ . This equation can be used to draw from the posterior distribution of every γ_k given γ_{-k} , i.e. all the elements of γ except γ_k . Moreover, the MCMC algorithms used to fit the model is such that it does not require $C(y^*)$ to be computed explicitly.

5.2 Markov Chain Monte Carlo

In order to draw from the predictive distribution $p(\tilde{y}|y_{1:t})$ we saw that we have to draw from the poster parameter distribution $p(\psi|y_{1:t})$. Therefore the authors implemented a Markov Chain Monte Carlo algorithm based on Gibbs sampling and that works as described below. Let $\psi = (\sigma_u^2, \sigma_v^2, \sigma_w^2)$ then

- Simulate $\theta_{0:t}$ from $p(\theta_{0:t}|\beta, \sigma_\epsilon^2, \psi, y_{1:t})$ using the method proposed by Durbin and Koopman (2002)
- Simulate ψ from $p(\psi|\beta, \sigma_\epsilon^2, \theta_{0:t}, y_{1:t})$
- Simulate β and σ_ϵ^2 from a Markov Chain with stationary distribution $p(\beta, \sigma_\epsilon^2|\psi, \theta_{0:t}, y_{1:t})$ using Stochastic Search Variable Selection (SSVS) ¹.

SSVS is a Gibbs sampling algorithm that generate a sequence $(\beta^{(i)}, \sigma_\epsilon^{2(i)}, \gamma^{(i)})$ for $i = 0, \dots, N$ from the joint posterior distribution in the following way

- Draw the coefficient vector $\beta^{(i)}$ from $p(\beta|y_{1:t}, \theta_{0:t}, \sigma_\epsilon^{2(i-1)}, \gamma^{(i-1)})$
- Draw $\sigma_\epsilon^{2(i)}$ from $p(\sigma_\epsilon^2|y_{1:t}, \theta_{0:t}, \beta^{(i)}, \gamma^{(i-1)})$
- Draw the vector indicator $\gamma^{(i)}$ componentwise by sampling consecutively from $p(\gamma_k|y_{1:t}, \theta_{0:t}, \beta^{(i)}, \sigma_\epsilon^{2(i)}, \gamma_{-k}^{(i)})$

In this way we obtain $(\phi^{(1)}, \dots, \phi^{(i)}, \dots, \phi^{(N)})$ as random draws from the posterior parameter distribution of interest $p(\phi|y_{1:t})$. Given $\phi^{(i)}$ then it is possible to draw from the predictive distribution of interest $p(\tilde{y}|y_{1:t})$ by sampling $p(\tilde{y}^{(i)}|\phi^{(i)})$ which is known and easy to sample.

An interesting feature of the model is that each draw $\phi^{(i)}$ includes a different draw $\delta^{(i)}$ and therefore different element of the regression coefficient vector β will be shrunk to 0. In other words, each MCMC iteration is associated with a different model. Therefore the posterior predictive distribution can be thought as a weighted average of model-specific predictive distributions

¹George and McCulloch (1997)

where the weights are given by the posterior distribution of the indicator γ given $y_{1:t}$.² This mechanism is known as Bayesian Model Averaging.

6 Predicting US Unemployment rate with BSTS

As mentioned in the introduction, the idea of using Google Trends in time series regression analysis have been heavily discussed in the literature. Here we want to provide a motivating example of when and how Google Trends are worth to be used as predictors.

The analysis is based on FRED data³ which provides monthly data on unemployment rate. The source of the data is the U.S. Bureau of Labor Statistics which publish the data with a conspicuous lag, usually one week after the end of the month. The whole time series is shown in Figure 1.

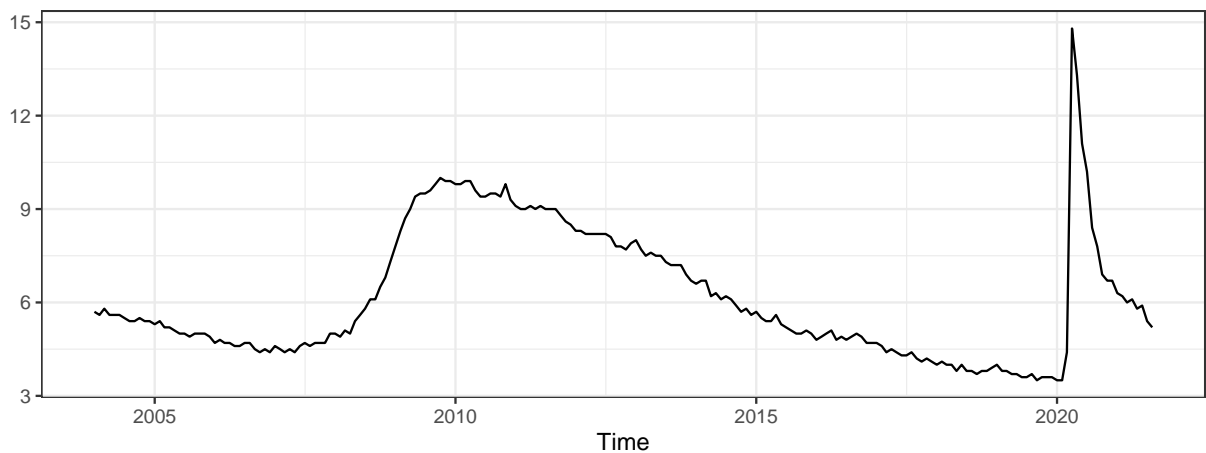


Figure 1: US Unemployment Rate, Monthly Data from 2004 to 2021. (Source: U.S. Bureau of Labor Statistics)

The time series shows many features. There is a clear trend-cycle and possible seasonality that are worth to be investigated through a time series decomposition. As we have already mentioned in the theoretic discussion of Classical Time Series Analysis, time series decomposition is a very valid tool to start an empirical analysis. Even though not essential in the context of BSTS, it may be very useful to adapt the model to the case study, adding or avoiding to add components to our baseline model. As already discussed, the insertion of state components have computational costs, hence this preliminary analysis might allow us to save time and CPU.

In Figure 2 a graphical representation of the decomposed series is provided. Clearly the residuals are not totally random since the peaks are localized in proximity of seasonal movements. We are going to take into account this aspect when we will fit an ARIMA model. At the moment, it is enough for us to know that the decomposition analysis strengthen our prior belief by showing a clear time-varying trend and seasonality. Therefore we would consider a BSTS model with full specification. We will come back to the analysis of the residual in ARIMA models

²This is not probably the most rigourose way to explain it, however it is still quite effective

³URL: <https://fred.stlouisfed.org/series/UNRATE>

when we will need the series to be stationary.

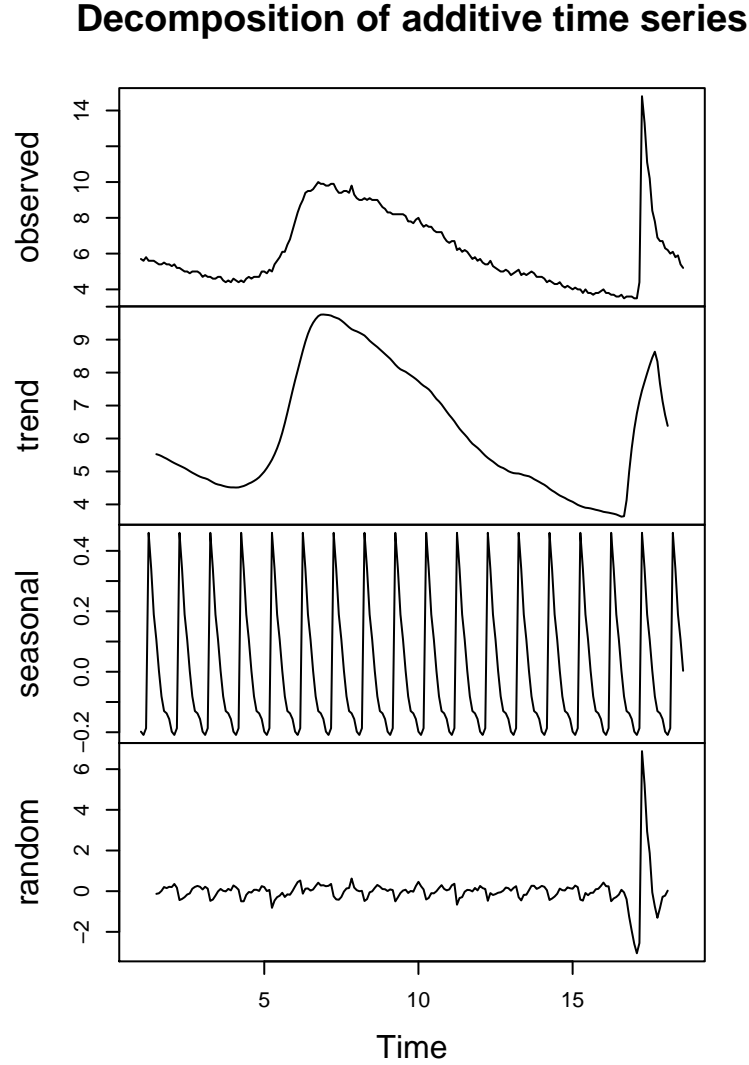


Figure 2: Decomposed Time Series of US Unemployment Rate, Monthly Data from 2004 to 2021. (Source: U.S. Bureau of Labor Statistics)

We decide to fit this time series with two models: a full specified BSTS and a full specified BSTS with a static regression component accounting for Google Trends. For this purposes a large dataset of predictor has been built thanks to the “gtrendsR” package of Philippe Massicotte and Dirk Eddelbuettel (2021). This package allows to download time series from Google Trend referring to the keywords of interest directly in R. This keywords have been selected manually after a preliminary analysis. The starting points have been: “unemployment”, “unemployment rate” and “unemployment benefits” and then we selected all the correlated queries and trends that we think could be related with our analysis ⁴. Obviously we expect that some of this queries

⁴Full list in Appendix

will not be drivers for unemployment rate prediction and thus they will be automatically shrunk to zero.

For a time series like the one of Figure 1, adding a static regression component would probably be unnecessary. Except for the spikes due to the 2021 pandemic, the series can be simply fitted by a BSTS with a time-varying trend and a seasonal component. Static coefficients predictors would just add avoidable noise. This is shown in Figure 3 where we see that overall the cumulative absolute errors of the BSTS with Google trends (GTrends) are larger with respect to the BSTS without static regression component (No GTrends). However, in the same Figure we can also notice that in 2020 the basic BSTS it is not able to predict the huge spike due to the pandemic while the GTrends model accumulates error at a steadier rate.

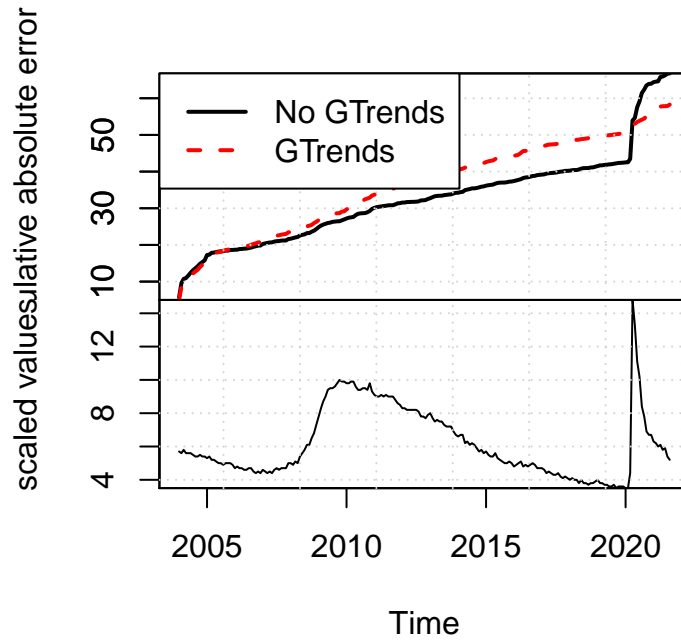


Figure 3: Model Comparison. US Unemployment Rate, Monthly Data from 2004 to 2021. (Source: U.S. Bureau of Labor Statistics)

For this shocking events, Google Trends can indeed make the difference in nowcasting and here we explain why. The Employment Situation for April 2020 has been released on 8 May 2020 by the US Bureau of Labor Statistics. In the meanwhile Google searches for unemployment benefits and programs had increased enormously at the end of March 2020. For instance, the following image has been taken by Google Trends ⁵ and shows the relative number of searches for “unemployment benefits” in US over the last 5 years. The spike during the pandemic, between the period from March 2020 and April 2020, is evident and it anticipates the one we see in the FRED series.

⁵URL: <https://trends.google.it/trends/>



Therefore, we repeated the analysis focusing on the last 5 years. As we expected, the model with Google Trends performs much better during the pandemic (see Figure 4).

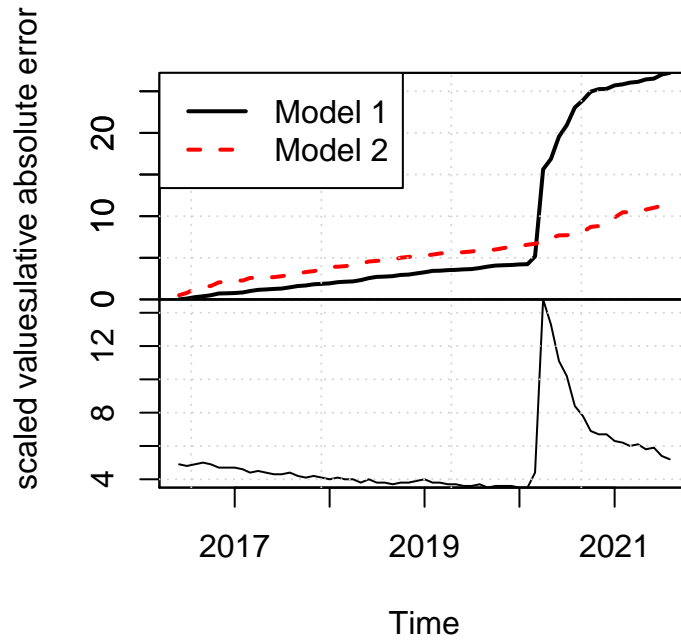


Figure 4: Model Comparison. US Unemployment Rate, Monthly Data from 2017 to 2021. (Source: U.S. Bureau of Labor Statistics)

7 Predicting US Unemployment rate with ARIMA

We would extend the previous analysis using ARIMA and ARIMAX models. The series is clearly not stationary and even though this does not affect BSTS models, stationarity of the series is of fundamental importance in the field of ARIMA models. We made several attempts to keep the series stationary, trying multiple differentiation of the series, however the best results have been achieved through a simple first difference of the data. We report below the resulting differentiated series (Figure 5)

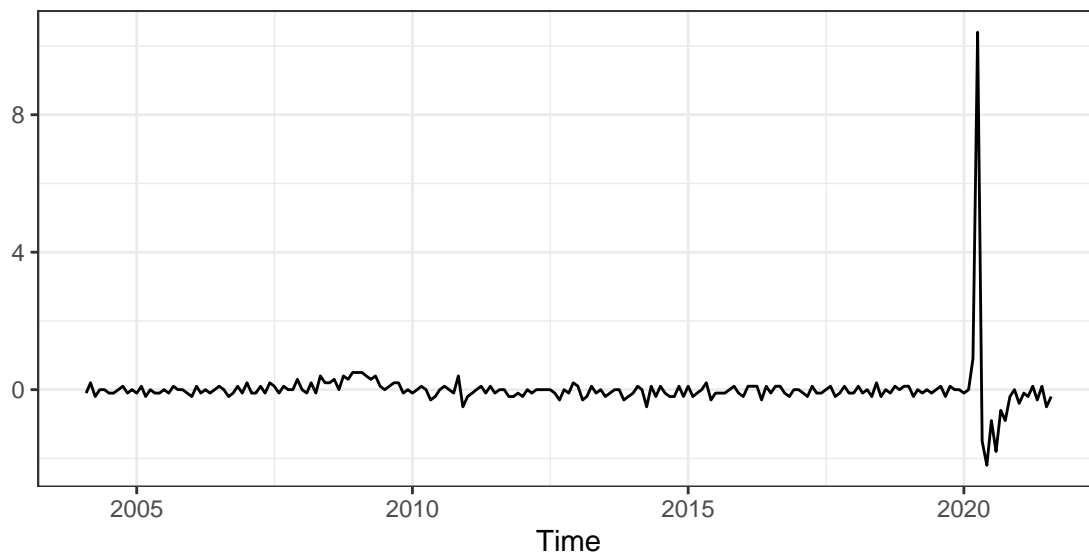


Figure 5: Transformed Time Series of US Unemployment Rate, Monthly Data from 2004 to 2021.
(Source: U.S. Bureau of Labor Statistics)

We assessed stationarity through graphical and quantitative tools. In Figure 6 and 7 we plot respectively the Autocorrelation and Partial Autocorrelation functions of the transformed time series. Both ACF and PACF looks like that of a white noise process. Moreover, there are no autocorrelations lying outside the 95% limits, and the Ljung-Box Q^* statistic has a p-value of 0.72 (for $h = 10$), which suggests that the observations are not correlated.

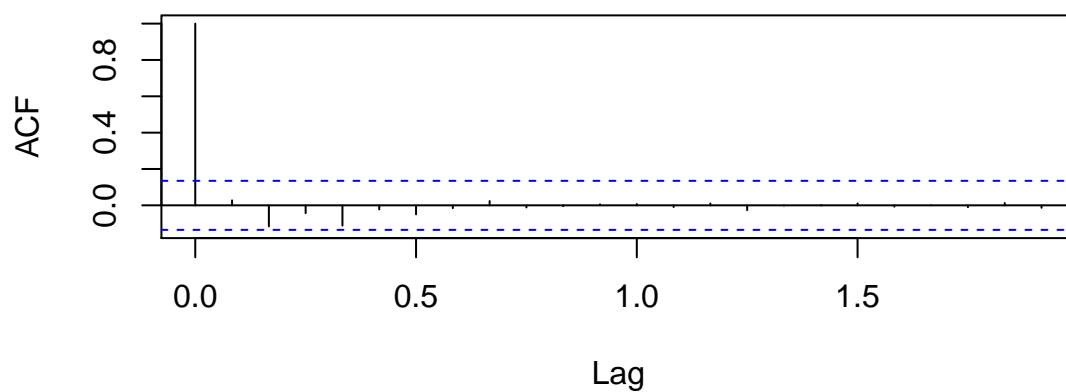


Figure 6: Autocorrelation function

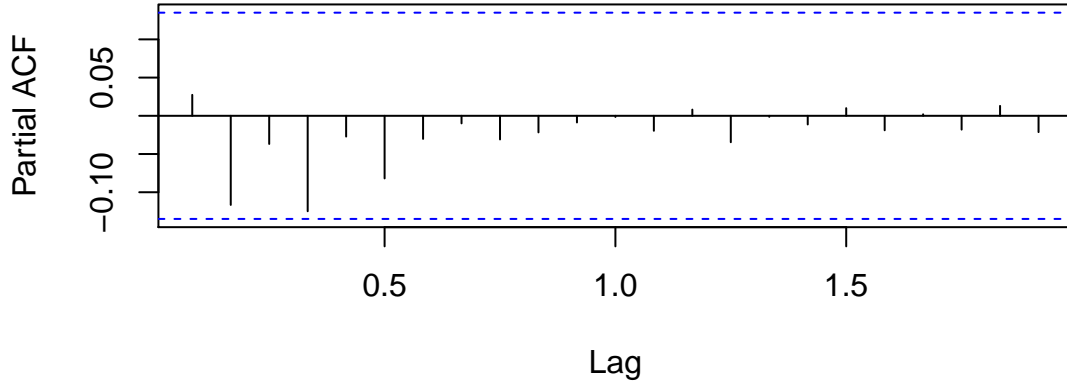


Figure 7: Partial autocorrelation function

Absent peaks in both ACF and PACF, this suggest a basic $ARIMA(0, 1, 0)$ for the original process. In practice a comparison in terms of AIC , AIC_C and BIC is useful to identify the best model, therefore we are going to use the `auto.arima` function of the package (Rob Hyndman et al., 2021) that performs model comparison automatically and the best model envisaged is exactly the one we guessed. In Figure 8 we provide some insights on the residuals of the model. First, the residuals' mean is zero or close to zero and no significant correlation in the residuals is detected. Moreover, the time plot of the residuals shows that the variance of the residuals is stable across the historical data, except for the period around April 2020 which we consider an outlier. Overall, the residual variance can be treated as constant. We also observe this on the histogram of the residuals. Furthermore, the histogram suggests that, ignoring the outlier, the residuals are almost normal.

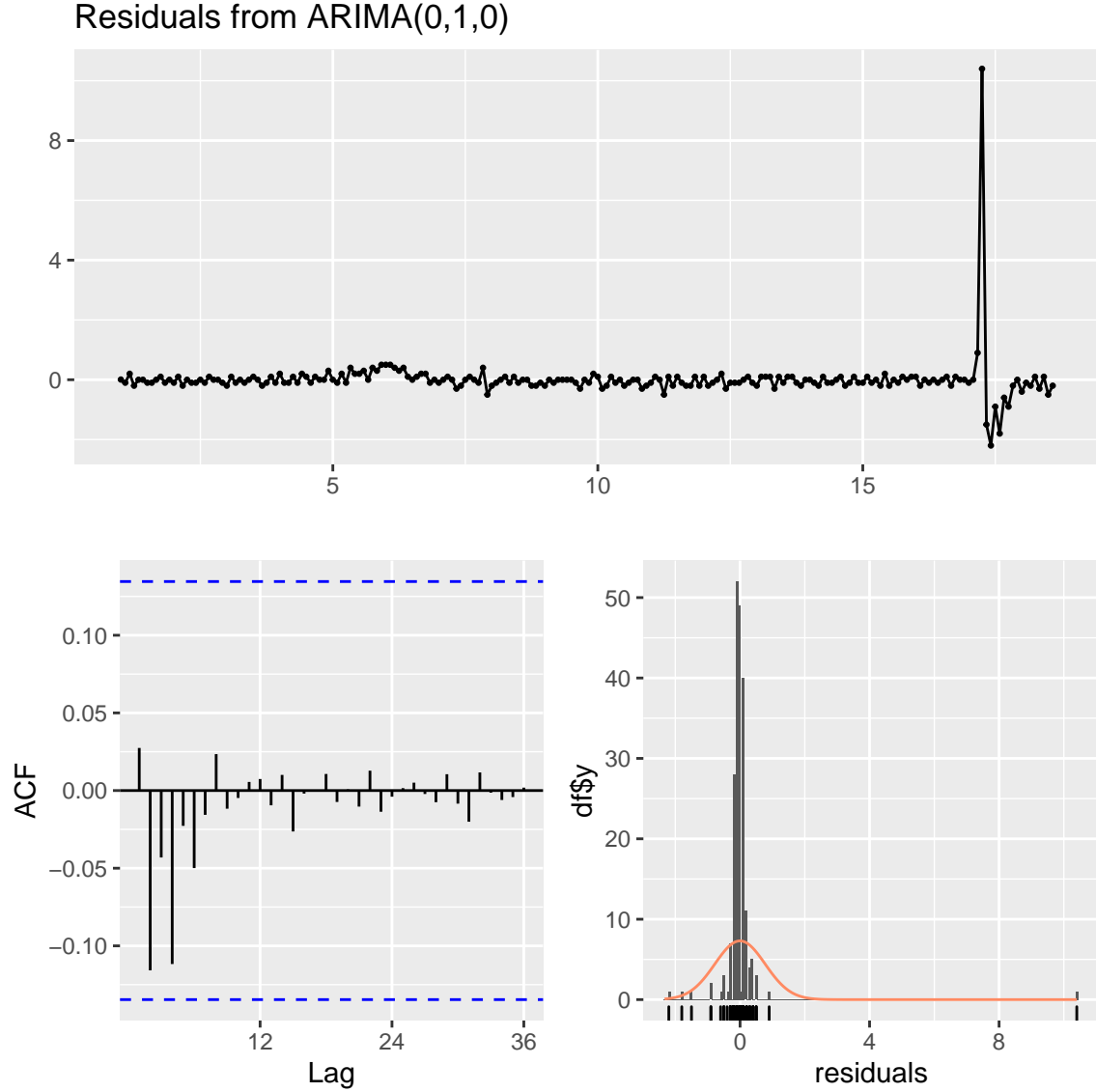


Figure 8: ARIMA: Diagnostic Checks

Then, repeating the analysis on BSTS models, we decide to add some predictors to the models. In an ARIMAX model, it is necessary to difference also the predictors as estimation of a model with non-stationary errors is not consistent and can lead to “spurious regression”. Hence, we repeated the same steps described for the main series. Since ARIMA models are not endowed of automatic variable selection methods, we decide to reduce the dimensionality by keeping only those predictors that have shown to be statistically significant at a significance level of 5% in a preliminary analysis when we fit the model with all the predictors. We selected six predictors out of twenty-one. The diagnostic checks on the residuals of the resulting models is in Figure 9. The structure of the residuals is such that they are uncorrelated (Ljung-Box Q^* statistic has a p-value of 0.53) and almost normally distributed.

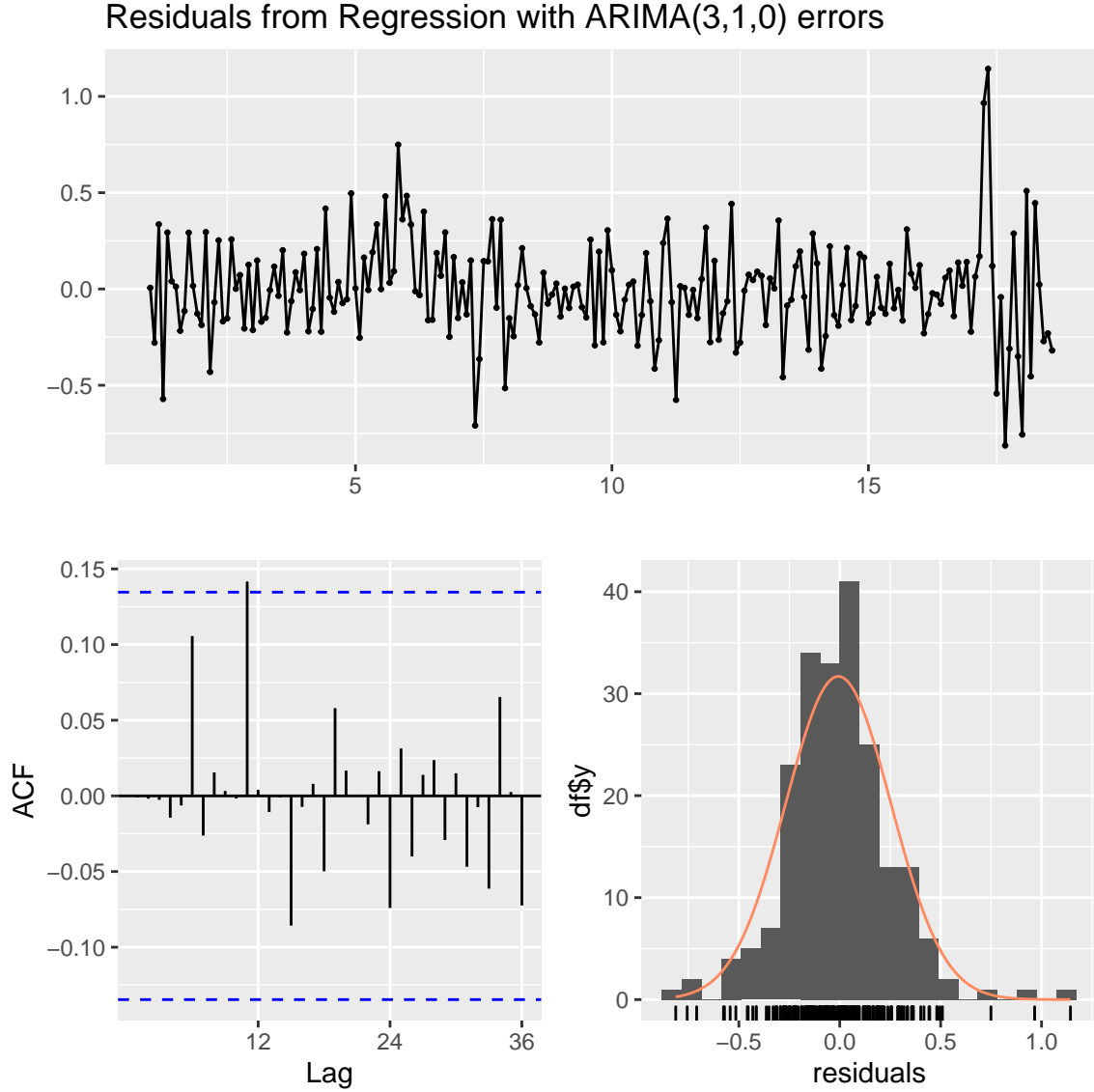


Figure 9: ARIMAX: Diagnostic Checks

Notice how, in the residuals time series, the extent of the outlier is now lower. This probably means that predictors do a good job in predicting the spike due to the pandemic crises. We confirm our intuition in the next Section.

8 Models Comparison

In this section we present a unified table with forecast accuracy measures to compare BSTS models and ARIMA models for US Unemployment rate prediction. The metrics we use are based on the forecast error, i.e. $e_i = y_i - \hat{y}_i$. Since the Mean Absolute Percentage Error (MAPE) cannot be used because of some data in the transformed series of Figure 5 are zeros, we use as

benchmarks the Weighted Mean Absolute Percentage Error or (WMAPE):

$$WMAPE = \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{\sum_{i=1}^n |y_t|}$$

which compares the errors with the average instead of the actual values, and the Mean Absolute Scaled Error (MASE):

$$MASE = \frac{1}{n} \frac{\sum_{t=1}^n |y_t - \hat{y}_t|}{\frac{1}{n-1} \sum_{t=2}^n |y_t - y_{t-1}|}$$

that compares the model forecasts with naive forecasts. A $MASE < 1$ means that our model provides point forecast which are more accurate than the lagged values of the same series. Moreover, we will use a Root Mean Square Error (RMSE) which penalizes for larger errors and a Mean Error (ME) to understand the direction of the forecast bias.

The series has been divided in two subsamples: a training sample (about two third of the whole sample) and a testing sample (the remaining part). The results in the test set are shown in the following table. Both ARIMA and BSTS performs well and, overall, results do not differ much among them. As we can see, the best performing model is the BSTS with Google Trends. In particular, it is the only one having a $MASE < 1$ in the reduced sample. By and large, we are satisfied of the performances, the $WMPAE$ is indeed almost always below 20% and in some cases below 10%. The most relevant feature reported by the data is the evident improvement of the performances when considering Google Trends as predictors in every case. Finally, we observe a slight upward bias in forecasts when considering predictors and a downward bias otherwise.

Table 1: Predictive Performances Comparison

Model	Sample	Google.Trends	WMAPE	MASE	RMSE	ME
BSTS	Full Sample		0.116	1.58	1.758	-0.127
		Yes	0.075	1.024	1.294	0.13
	Reduced Sample		0.247	1.62	3.094	-0.355
		Yes	0.148	0.974	2.247	0.477
ARIMA	Full Sample		0.103	1.399	1.926	-0.115
		Yes	0.075	1.023	1.18	0.092
	Reduced Sample		0.241	1.582	4.748	-0.391
		Yes	0.163	1.074	2.661	0.574

However, point forecasts does not show the full picture. Credible or confidence intervals, depending on the bayesian or frequentist approach, are of crucial importance for statisticians. In the Figure 10 forecast plots for the critical pandemic period are provided. BSTS with Google Trends is clearly the model that better predict the shock, whereas the ARIMA without any

predictor clearly miss it, even though it provides good forecasts thereafter. However, the bigger difference between BSTS and ARIMA relies in quantifying the uncertainty around the point forecasts. Thanks to Markov Chain Monte Carlo simulations, BSTS models are able to provide a full description of the uncertainty providing at any time $t + 1$ the one-step-ahead posterior distribution from which we can infer meaningful information on the future development of the series.

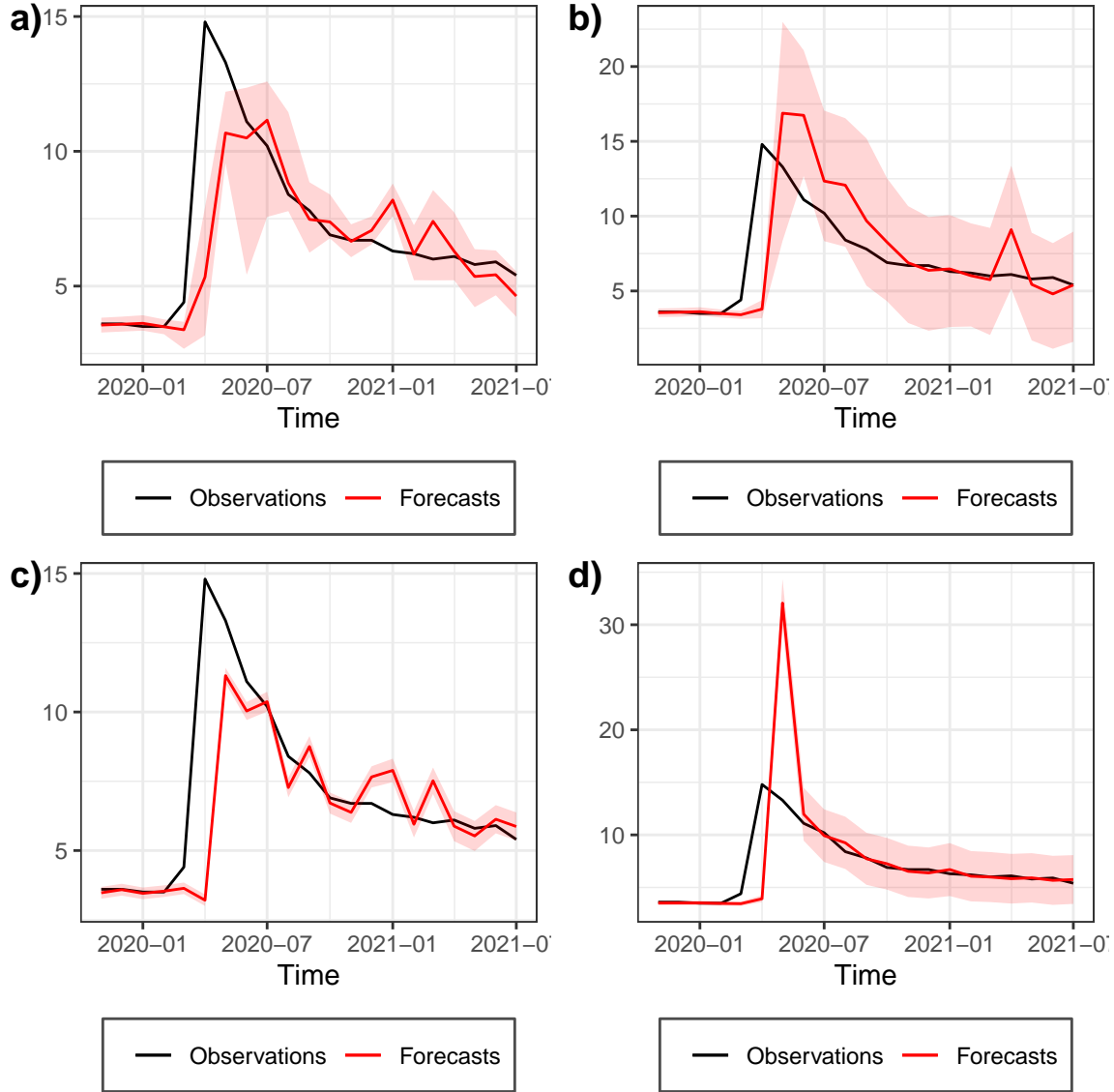


Figure 10: Point forecast and 95 percent credible/confidence interval. a) BSTS with Google Trends. b) BSTS basic. c) ARIMA with Google Trends. d) ARIMA basic.

9 Conclusion

The web provides tons of data everyday. In the last decade, researchers in academy and companies have tried to use this data to understand and, in some cases, anticipate social, economical

and business facts. On the heels of the recent literature, we tried to predict shocks in macroeconomic time series using such new methodologies.

Bayesian Structural Time Series models is indeed a promising approach in time series analysis. Thanks to its state space model structure and spike and slab regression it is able to account for several underlying unobservable components of the series and to deal with large amounts of predictors. In Section 8 (and in Appendix 3) we compared BSTS and ARIMA models. Even though BSTS models provide just slightly better performances in terms of WMAPE, MASE, RMSE, MAE, there are several reasons to prefer the BSTS to ARIMA. First, no need of preliminary checks. Even though checks are useful in model selection also in BSTS, the latter does not require the series to be stationary, the assessment of ACF and PACF and hypothesis tests. Therefore we can consider BSTS more intuitive and automatic. Second, Bayesian variable selection. BSTS are able to deal with a large set of predictors, generating sparsity through a fully automated procedure, i.e. spike and slab regression. Predictors are instead manually selected in ARIMA models. Third, forecast uncertainty quantification. BSTS provides a full description of the one-step-ahead (or h-step-ahead) predictive distribution through MCMC simulations, whereas ARIMA models just estimate the point forecast and provide the standard error.

Nevertheless, overall, what we observe is that models enriched with Google Trends data systematically reported better predictive performances with respect to baseline models. Moreover, these models were able to better predict the peak in April 2020 due to the pandemic crises. This fact is extremely interesting from a policy making perspective since it improves the timing and the effectiveness of containment measures.

Finally, in Appendix 3 of this work, we provide some insights of the results found by repeating the same analysis for the Italian unemployment rate.

10 References

- Choi, H. and Varian, H. (2009). Predicting the present with Google Trends. Tech. rep., Google.
- Choi, H. and Varian, H. (2012). Predicting the present with Google Trends. *Economic Record* 88.
- Bangwayo-Skeete, Prosper F., and Ryan W. Skeete. 2015. Can Google data improve the forecasting performance of tourist arrivals? Mixed-data sampling approach. *Tourism Management* 46: 454–64
- Bañbura, Marta, Modugno, Michele (2012). “Maximum Likelihood Estimation of Factor Models on Datasets with Arbitrary Pattern of Missing Data”. *Journal of Applied Econometrics*. 29 (1): 133–160. doi:10.1002/jae.2306
- Boone, Tonya, Ram Ganeshan, Robert L. Hicks, and Nada R. Sanders. 2017. Can Google Trends Improve your Sales Forecast? *Production and Operations Management* 27: 1770–74.
- Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.
- Durbin, J., & Koopman, S. J. (2002). A Simple and Efficient Simulation Smoother for State Space Time Series Analysis. *Biometrika*, 89(3), 603–615. <http://www.jstor.org/stable/4140605>
- George, E. I., & McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica sinica*, 339–373.

G. Petris, Sonia Petrone, Patrizia Campagnoli (2009), *Dynamic Linear Models with R*, Springer.

Ghysels, E. and M. Marcellino (2018): *Applied economic forecasting using time series methods*, Oxford University Press.

Hamid, A., & Heiden, M. (2015). Forecasting volatility with empirical similarity and Google Trends. *Journal of Economic Behavior & Organization*, 117, 62-81.

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O’Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmien F (2021). *forecast: Forecasting functions for time series and linear models*. R package version 8.15, <https://pkg.robjhyndman.com/forecast/>

Hyndman RJ, Koehler AB (2006). “Another Look at Measures of Forecast Accuracy.” *International Journal of Forecasting*, 22, 679–688.

Mulero, R., García-Hiernaux (2021), A. Forecasting Spanish unemployment with Google Trends and dimension reduction techniques. *SERIEs* 12, 329–349.

Naccarato, A., Falorsi, S., Loriga, S., & Pierini, A. (2018). Combining official and Google Trends data to forecast the Italian youth unemployment rate. *Technological Forecasting and Social Change*, 130, 114-122.

Philippe Massicotte and Dirk Eddelbuettel (2021). *gtrendsR: Perform and Display Google Trends Queries*. R package version 1.4.8. <https://CRAN.R-project.org/package=gtrendsR>

Preis, T., Moat, H. S., & Stanley, H. E. (2013). Quantifying trading behavior in financial markets using Google Trends. *Scientific reports*, 3(1), 1-6.

Rockova, V., & McAlinn, K. (2021). Dynamic variable selection with spike-and-slab process priors. *Bayesian Analysis*, 16(1), 233-269.

Scott, Steven L. and Varian, Hal R., Predicting the Present with Bayesian Structural Time Series (June 28, 2013). Available at SSRN: <https://ssrn.com/abstract=2304426> or <http://dx.doi.org/10.2139/ssrn.2304426>

Steven L. Scott (2021). *bsts: Bayesian Structural Time Series*. R package version 0.9.7. <https://CRAN.R-project.org/package=bsts>

Woloszko, N. (2021). Tracking GDP using Google Trends and machine learning: A new OECD model. *Central Banking*, 12, 12.

11 Appendix

11.1 Appendix 1: BSTS with Dynamic Regression Component

BSTS models allows also to insert a dynamic regression component into the model. We did not focused on this aspect in the main part of the work nor in the application because of the complexity of the topic and the incremental computational costs in the implementations. In this appendix we describe very briefly how does the dynamic regression component work and why we expect an improvement from it.

The evolution of the dynamic coefficients is the following

$$\beta_{i,t} = \beta_{i,t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \frac{\sigma_i^2}{\sigma_{x_i}^2})$$

meaning that each coefficient evolves independently as a random walk and its own variance term

is scaled by the variance of the i 'th column of X . The precision is modeled as a gamma

$$\frac{1}{\sigma^2} \sim G(a, b)$$

and finally we place a Gamma prior also on the hyperparameters $\sqrt{a/b}$ and a .

In the following plot (Figure 11) we report results of model comparison between a BSTS with dynamic regression component and the standard BSTS. As we expect, the first model performs better in terms of cumulative absolute error with respect to the latter. This improvement comes at a high computational cost, therefore it is not recommended to use it for recurrent forecasting at least we believe that treating some coefficients as dynamics will lead to significant improvements. Modern technique have been developed to overcome this computational difficulty, we want to mention Rockova (2021).

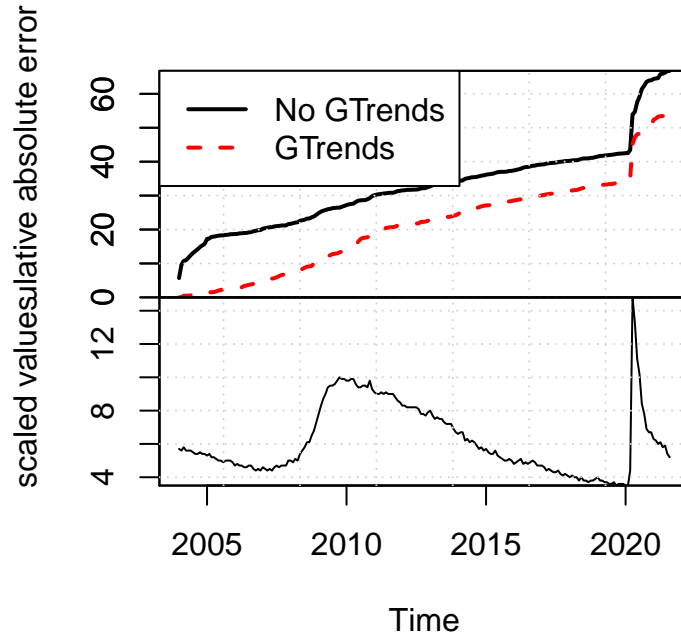


Figure 11: Model Comparison. US Unemployment Rate, Monthly Data from 2004 to 2021. (Source: U.S. Bureau of Labor Statistics)

11.2 Appendix 2: Google Trend Queries

Here we provide a list of key words we looked for in Google Trends engine

Table 2: Key Words List

Unemployment	Unemployment insurance	Monster.com
Careerbuilder	Google careers	unemployment ohio
Unemployment benefits	Unemployment agency	unemployment login
Unemployment rate	Indeed.com	unemployment ny
Dice	Unemployment office	unemployment claim
Unemployment compensation	Job search engine	florida unemployment
Glassdoor	LinkedIn	unemployment minnesota

11.3 Appendix 3: Predict Italian Unemployment with BSTS

We repeat the analysis in the Italian context, using ISTAT data which provides quarterly data on employment rate over an horizon of 16 years (from the first quarter of 2004 to the forth quarter of 2020). The time series is shown in Figure 12. The series looks quite different with respect to the one observed for US. Because of the Italian job markets' rigidities, shocking events affects the economy with a greater lag and persistence. This may explain the absence of peaks linked to the 2008 financial crisis and 2020 pandemic crisis that we instead observe in US. From a forecasting point of view, this implies a more stable and predictable series.

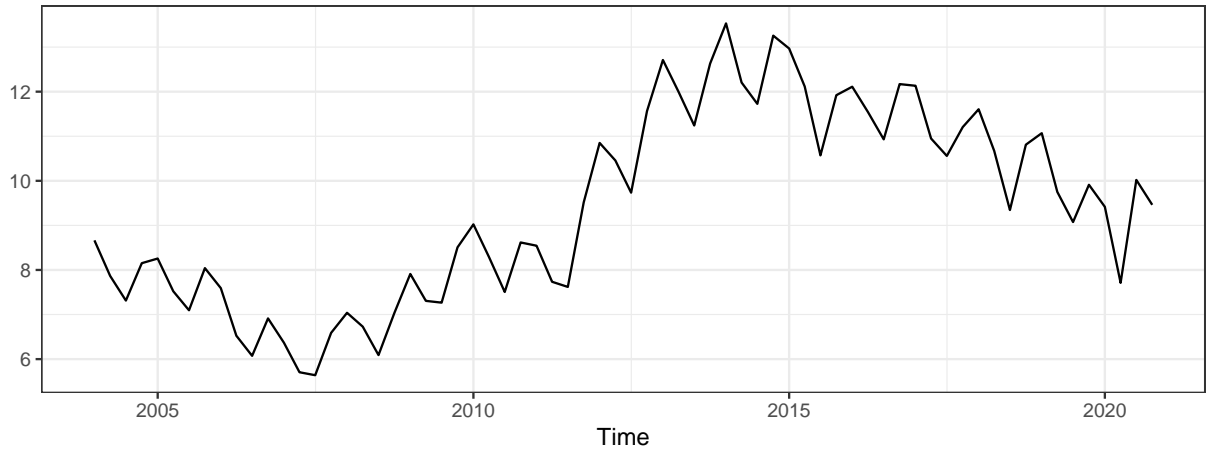


Figure 12: Italian Employment Rate, Quarterly Data from 2004 to 2020. (Source: ISTAT)

After a preliminary analysis of time series components (trend-cycle and seasonality), stationary, order of integration, ACF and PACF and diagnostic checks, we selected the following models: full specified BSTS, full specified BSTS with Google Trends predictors, $ARIMA(1, 1, 0)(0, 1, 1)_4$, $ARIMAX(0, 1, 0)(1, 0, 1)_4$. In Table 3 the results in terms of one-step-ahead forecasts are shown. In accordance with the previous considerations, point forecasts are much more accurate than in the US case. $WMAPE$ is always below 10 and $MSE < 1$. BSTS models outperforms. This is explained by the fact that the latent process succeed in capturing the components of the series. Finally, models with Google Trends produce slightly better performances.

Table 3: Predictive Performances Comparison

Model	Google.Trends	WMAPE	MASE	RMSE	ME
BSTS		0.044	0.512	0.763	0.022
	Yes	0.044	0.509	0.755	-0.051
ARIMA		0.084	0.977	1.098	-0.103
	Yes	0.076	0.884	0.973	-0.002

We envisaged a couple of reasons why the gains due to Google Trends data are lower in Italy with respect to US. First, job market rigidities. Looking for a job and being employed are not necessarily correlated if job opportunities are scarce. Second, Italians old style job search. Especially for low skill labor, word-of-mouth and references are still the rule. Thus Google searches do not show the full picture. This aspects are worth to be investigated in future analysis.

In the following table we report the key words we looked for in Google Trends engine

Table 4: Key Words List

disoccupazione	corsi per disoccupati
inps disoccupazione	disoccupato
la disoccupazione	lavoro subito.it
inps	ebay annunci lavoro
domanda disoccupazione	infojobs
indennità disoccupazione	subito.it
requisiti disoccupazione	almalaurea
disoccupazione ordinaria	monster
inps disoccupazione ordinaria	kijiji lavoro
inps indennità disoccupazione	domanda disoccupazione ordinaria