universität
wien

MAGISTERARBEIT

Titel der Magisterarbeit

On the Predictive Performance of Bayesian Structural Time Series-Models: An Empirical Comparison of Nowcasting and Forecasting Applications

Verfasst von

Mag.rer.soc.oec. Matthias Schmidl, Bakk.rer.soc.oec.

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften
(Mag.rer.soc.oec.)

Wien, 2015

Studienkennzahl lt. Studienblatt:	A 066 951
Studienrichtung lt. Studienblatt:	Magisterstudium Statistik
Betreuerin / Betreuer	ao. Univ.-Prof. Dipl.-Ing. Dr. Erhard Reschenhofer

Abstract

This thesis analyses the predictive accuracy of Bayesian Structural Time Series (BSTS) models in empirical nowcasting and forecasting applications. In the first part *Google Trends*-data are used to nowcast the index of German retail sales via mail order houses or via Internet. Therefore, out-of-sample predictions of BSTS-models are compared to those of a stepwise regression model. In the second part BSTS-models are applied to the M3-Competition data to compare the forecast accuracy to other major time series methods. The results indicate that the BSTS-models perform exceptionally well in both applications – especially with respect to the M3-Competition BSTS rank among the top-performing methods.

Abstract

Diese Magisterarbeit untersucht die Vorhersagegenauigkeit von Bayesian Structural Time Series (BSTS) Modellen in empirischen Nowcasting- und Forecasting-Anwendungen. Im ersten Teil werden *Google Trends*-Daten herangezogen um Nowcasts des deutschen Umsatzindex für den Versand- und Internet-Einzelhandel zu erstellen. Im Zuge dessen werden Out-of-Sample Prognosen von BSTS-Modellen mit jenen eines schrittweisen Regressionsmodells verglichen. Um die Forecast-Genauigkeit mit anderen bedeutenden Zeitreihenmodellen zu vergleichen, werden im zweiten Teil BSTS-Modelle auf die Daten der M3-Competition angewandt. Die Ergebnisse zeigen, dass BSTS-Modelle eine ausgesprochen gute Performance aufweisen – insbesondere im Bezug auf die M3-Competition rangieren die BSTS-Modelle unter jenen Methoden mit der höchsten Genauigkeit.

Keywords: Bayesian Structural Time Series, Nowcasting, Forecasting

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

In 2013, Steven L. Scott and Hal Varian presented a forecasting framework called *Bayesian Structural Time Series* (abbreviated *BSTS*). This framework combines three statistical methods: A basic structural time series model to capture trend, seasonality or other components of the target series; spike-and-slab variable selection for the static regression component and Bayesian model averaging to calculate the final forecasts.

The method has proven to be especially useful for the estimation of short-term forecasts. In this connection the authors apply the method on *Google Trends*-data to predict the level of initial claims for unemployment benefits. It should be noted that in this application both the target series and the predictor variables have the same time index. Such problems that rather try to '*predict the present'* (Hal Varian) than the future are also called "*nowcasts*".

The purpose of this thesis is to evaluate BSTS-predictions in a nowcasting application as well as in forecasting exercises, with both parts analysing different aspects of the BSTS-method. In the nowcasting exercise shopping related *Google Trends*-series are used to predict the level of retail sales via mail order houses or via Internet for Germany. On the one hand the variable selections of the spike-and-slab algorithm are compared to the model selection of a stepwise regression model and on the other hand the out-of-sample predictions of these models along with a baseline model are benchmarked to each other. In the forecasting exercise the performance of the BSTS-method is evaluated in the context of the M3-Competition, an empirical study that compares the predictive performance of major time series methods.

The thesis is structured as follows: It starts with a detailed description of the BSTS-method with focus on the state space components and the spike-and-slab variable selection. Section 3 is devoted to the application of the BSTS-method in a nowcasting exercise. It describes the data sources as well as the methodology that is used for the comparisons and for the analysis of the results. The accuracy of BSTS-forecasts is evaluated in Section 4, where first the M3-Competition – respectively the M3-Dataset – is characterized and issues on the evaluation of the forecasts are discussed. For the estimation of the predictions a method is applied that automatically selects one out of ten BSTS-models for each time series of the M3-competition to calculate the final forecasts. Then the out-of-sample forecasts of the

BSTS-models are compared to the results of the M3-Competition. In the final section the results of the nowcasting and the forecasting exercises are discussed and main conclusions of these comparisons are drawn.

2 Bayesian Structural Time Series

This section describes the structural time series models used in the recently published R-package ‘BSTS’ (Scott, 2014) and is organized as follows. It starts with a brief introduction to state space analysis, in which the basic formulation of state space models is explained and the primary tools for the estimation are presented. Section 2.2 specifies the state components that are applied in this thesis and Section 2.3 deals with the spike-and-slab algorithm that is used for variable selection. Finally, Section 2.4 explains the Markov chain Monte Carlo (MCMC) algorithm that is used for posterior inference.

2.1 State Space Models

The methodological foundations of BSTS are state space models, which are used to treat a wide range of problems in time series analysis. The subsequent explanation of the BSTS-model is based on articles by Scott & Varian (2013, 2014) and Brodersen et al. (2014). A comprehensive analysis of state space models – especially with respect to computational aspects – can be found in Durbin & Koopman (2001).

The basic concept of state space analysis considers a series of observations $\boldsymbol{y} = y_{1:n}$ and assumes that the development over time is determined by a set of latent state vectors $\boldsymbol{\alpha} = \alpha_{1:n}$. A state space model specifies the relation between the α_t’s and the y_t’s with the objective to infer relevant properties of the α_t’s from knowledge of the observations. State space models are defined by a pair of equations

$$y_t = Z_t^T \alpha_t + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d.}\, N(0, H_t) \tag{1}$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t \qquad \eta_t \sim \text{i.i.d.}\, N(0, Q_t) \tag{2}$$

where equation (1) is known as the *observation equation* or *measurement equation* and equation (2) is called the *state equation* or *transition equation*. The observation equation links the observed data y_t to the state vector α_t, whereas the transition equation determines the development of the state variables. In this formulation y_t denotes a real-valued observation at time t, Z_t is a d-dimensional vector (*observation vector* or *output vector*), T_t is a $d \times d$-dimensional matrix (*transition matrix*) and R_t is a rectangular matrix (*control matrix*) of dimension $d \times q$. The observation error term ϵ_t is normally distributed with mean zero and variance H_t (a positive scalar). Q_t denotes a $d \times q$ state variance matrix. The authors point out that the error term $R_t\eta_t$ of equation (2) allows the modeler to work with a full-rank variance matrix Q_t, so that it is possible to incorporate state components of less than full rank (for example seasonality components).

The attractive feature of the state space model approach is its flexibility and modularity – in the sense that various state components can be assembled to capture important properties of the data. This can be done by concatenating the individual observation vectors to form Z_t and the state vectors to form α_t. The other model matrices are then arranged as elements in a block-diagonal matrix. These transformations can be made under the assumption of independent state components, which means that the errors of different state-components are independent so that Q_t is a block-diagonal matrix.

Models in the form of equation (1) and (2) are said to be in *state space form*. This framework allows the representation of a large class of models in state space form: Beginning with classical linear regressions all ARIMA and VARIMA models can be implemented. Moreover models with time-varying parameters (for example trend or seasonality) can be operationalized or heteroskedastic data can be modelled.

The methodological foundations for the estimation of models in state space form are the *Kalman filter and smoother.* The Kalman filter computes a statistically optimal estimate of the state vector α_{t+1}, conditional on the available information upon time t, in other words the distribution of $p(\alpha_{t+1} \,|y_{1:t})$. The Kalman smoother incorporates the full set of information in the sample $y_{1:n}$ to compute $p(\alpha_t \,|y_{1:n})$. Both algorithms are recursive processes that jointly follow a forward-backward pattern: While the Kalman filter moves forwards through the list of observations, the smoother proceeds backwards through time. For details on the Kalman filter see Kalman (1960). Harvey (1989) gives a comprehensive review of Kalman filtering in the context of time series methodology and Durbin & Koopman (2001) also discuss computational aspects of state space model estimation.

Another method that is used in state space model analysis is Bayesian data augmentation. This technique is used to simulate the state from its posterior distribution given the data – that is $p(\boldsymbol{\alpha}|\boldsymbol{y})$. The necessity of employing such methods arises from the fact that α_t and α_{t-1} are serially correlated. In this respect the BSTS-package employs an algorithm developed by Durbin & Koopman (2002), which generates random noise that has the same covariance as $p(\boldsymbol{\alpha}|\boldsymbol{y})$. The simulation of $p(\boldsymbol{\alpha}|\boldsymbol{y})$ is completed by adding the expected value $E(\boldsymbol{\alpha}|\boldsymbol{y})$ to this random noise. In order to calculate the required expectations, Durbin & Koopman use the Kalman filter and a 'fast state mean smoother'.

2.2 State Components

The following sections list the state components that are implemented in the BSTS-package and will be used for the empirical analysis of this thesis. The descriptions are based on the accompanying manual of the package.

2.2.1 Local Level

The local level model is a simple form of a model that assumes that the trend follows a random walk.

$$\alpha_{t+1} = \alpha_t + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d. } N\,(0,\sigma^2) \tag{3}$$

2.2.2 Local Linear Trend

The local linear trend component is defined by the following pair of equations:

$$\mu_{t+1} = \mu_t + \delta_t + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d. } N(0,\sigma_\mu^2) \tag{4}$$

$$\delta_{t+1} = \delta_t + \eta_\mathrm{t} \qquad \eta_t \sim \text{i.i.d. } N(0,\sigma_\delta^2) \tag{5}$$

Both the mean (equation 4) and the slope (equation 5) of the trend follow a random walk, where μ_t is the value of the trend at time t and δ_t is the slope at time t.

2.2.3 Robust Local Linear Trend

A robust specification of the local linear trend model assumes that mean and slope move according to random walks, but here the errors are t-distributed with ν_μ and ν_σ degrees of freedom.

$$\mu_{t+1} = \mu_t + \delta_t + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d.}\, \mathcal{T}_{\nu_\mu}(0,\sigma_\mu^2) \tag{6}$$

$$\delta_{t+1} = \delta_t + \eta_t \qquad \eta_t \sim \text{i.i.d.}\, \mathcal{T}_{\nu_\delta}(0,\sigma_\delta^2) \tag{7}$$

2.2.4 Generalized Local Linear Trend

This generalization of the local linear trend model assumes that the level follows a random walk, while the slope of the time trend moves according to an $AR(1)$ process centered around a long-term slope D (with $|D| > 0$). The model is given by

$$\mu_{t+1} = \mu_t + \delta_t + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d.}\, N(I, \sigma_\mu^2) \tag{8}$$

$$\delta_{t+1} = D + \phi(\delta_t - D) + \eta_\text{t} \qquad \eta_t \sim \text{i.i.d.}\, N(I, \sigma_\delta^2) \tag{9}$$

where ϕ determines the rate at which the local trend is updated. With $|\phi| < 1$ the slope exhibits short term stationarity around the long run slope D.

2.2.5 AR(p) State Component

The basic autoregressive model (AR) of order p can be written as follows

$$\alpha_t = \phi_1 \alpha_{t-1} + \cdots + \phi_p \alpha_{t-p} + \epsilon_{t-1} \qquad \epsilon_t \sim \text{i.i.d.}\, N(0, \sigma^2) \tag{10}$$

The part of the state transition matrix T_t representing the autoregressive model has ϕ in its first row and ones along its first subdiagonal, zeros elsewhere. The state variance matrix Q_t has σ^2 in its first element and zeros elsewhere, the observation matrix Z_t has 1 in its first element and zeros elsewhere.

2.2.6 Seasonal State Component

The seasonal state component provides a method to account for seasonal variations.

$$\gamma_\text{t+1} = -\sum_{i=1}^{S-1} \gamma_{t-i} + \epsilon_t \qquad \epsilon_t \sim \text{i.i.d.}\, N(0, \sigma^2) \tag{11}$$

The seasonality is modelled as a regression on dummy variables so that the expected value of the coefficients is zero over a horizon of S seasons. For a cycle of S seasons the state vector γ_t has dimension $S - 1$ and the seasonal component part of the transition matrix has dimension $S - 1 \times S - 1$ with value −1 along the top row, 1 in the subdiagonal and zero elsewhere.

2.2.7 Static Regression

A linear regression component with K constant coefficients β and covariates $\mathbf{x}_t$ can be implemented in a state space model by appending $\beta^T\mathbf{x}_t$ to the observation vector Z_t and adding a constant 1 to each α_t.[1] The regressors $\mathbf{x}_t$ are assumed to be contemporaneous – potential effects of previous values can be incorporated by shifting the corresponding covariates in time.

2.3 Spike-and-slab Regression

In models containing a static regression component the BSTS-package uses spike-and-slab regression for variable selection. Especially in the case of a large set of potential predictors[2] the usefulness of classical variable selection methods, such as significance testing or information criteria, is increasingly limited (among others the problem of 'fat regressions' arises). In this regard, the spike-and-slab method drastically reduces the size of the regression problem by placing a spike-and-slab prior on the regression coefficients. (For details consult George & McCulloch 1993, 1997; Madigan & Raftery 1994; The following description is based on Scott & Varian 2013).

Let γ denote a K-dimensional vector that indicates if a particular covariate is included in the regression, precisely $\gamma_k = 1$ if $\beta_k \neq 0$, and $\gamma_k = 0$ otherwise. Further β_γ denotes the subset of β where $\beta_k \neq 0$. A spike-and-slab prior is then factorized as follows

$$p(\beta, \gamma, \sigma_\epsilon^2) = p(\beta_\gamma | \gamma, \sigma_\epsilon^2) p(\sigma_\epsilon^2 | \gamma) p(\gamma) \tag{12}$$

[1] The BSTS-package also contains a dynamic regression state component to fit a linear model using time-varying coefficients. Since a spike-and-slab variable selection algorithm is efficiently implemented for static regression, the dynamic regression component is not used and not presented in the present thesis. Readers interested in the dynamic regression component can consult Scott (2014) or Brodersen et al. (2014).

[2] The number of covariates can even exceed the number of observations.

The 'spike' component is the marginal distribution $p(\gamma)$ that places the point mass at zero. In principle $p(\gamma)$ is a positive probability, but it is convenient to use the product of independent Bernoulli distributions

$$\gamma \sim \prod_{k=1}^{K} \pi_k^{\gamma_k} (1-\pi_k)^{1-\gamma_k} \tag{13}$$

Since the prior inclusion probabilities[3] π_k are usually not available, the π_k's are all set to the ratio of expected model size[4] p to the number of regressors K that is $\pi_k = p/K$.[5]

Now focus on the 'slab' part of the prior: let b be a vector of prior expectations about the value of the regression coefficients β. With respect to the symmetric matrix Ω^{-1}, the rows and columns of Ω^{-1} corresponding to $\gamma_k = 1$ are denoted as Ω_γ^{-1}. Then a conjugate normal-inverse Gamma distribution is used for the conditional 'slab' priors $p(1/\sigma_\epsilon^2|\gamma)$ and $p(\beta_\gamma|\sigma_\epsilon,\gamma)$:

$$\beta_\gamma|\sigma_\epsilon^2,\gamma \sim N\left(b_\gamma, \sigma_\epsilon^2\left(\Omega_\gamma^{-1}\right)^{-1}\right) \qquad \frac{1}{\sigma_\epsilon^2}|\gamma \sim \Gamma(\frac{\nu}{2},\frac{ss}{2}) \tag{14}$$

The authors argue that in practice it is conventional to set the vector of prior means $b = 0$ and to assume Ω^{-1} proportional to $\mathbf{X}^T\mathbf{X}$, where $\mathbf{X}$ is the matrix of predictors so that $\boldsymbol{x}_t$ is row t. The authors also note that the ratio $\mathbf{X}^T\mathbf{X}/\sigma_\epsilon^2$ is the total Fisher information of the covariates. Zellner's g-prior (Zellner, 1986; Chipman et al., 2001; Liang et al., 2008) sets $\Omega^{-1} = \frac{\kappa}{n}\mathbf{X}^T\mathbf{X}$, which can be interpreted as the average information available from κ observations. Since $\mathbf{X}^T\mathbf{X}$ possibly has not full rank, Zellner's g-prior is adapted by averaging $\mathbf{X}^T\mathbf{X}$ with its diagonal, obtaining

$$\Omega^{-1} = \frac{\kappa}{n}(w\mathbf{X}^T\mathbf{X} + (1-w)diag(\mathbf{X}^T\mathbf{X})), \tag{15}$$

[3] π_k denotes the probability of regressor k being included in the model.

[4] The expected number of nonzero predictors.

[5] The supplied default value in the BSTS-package is $\pi_k = 0.5$.

with default values $w = 1/2$ and $\kappa = 1$. The remaining parameters that have to be chosen are ss, which can be considered as a prior sum of squares and ν, the prior sample size. These prior parameters can be elicited by asking the modeler for the expected R^2 and ν determining the weight given to that guess. Let s_y^2 denote the marginal standard deviation of the response, then $ss = \nu(1 - R^2)s_y^2$. The default values for the prior parameters supplied by the software are $R^2 = 0.5$, $\nu = 0.01$.

2.3.1 Posteriors

For the calculation of the conditional posteriors of β and σ_ϵ^2 given γ, the observations y_t are considered without the contribution of the time series state components. Therefore $y_t^* = y_t - Z_t^T\alpha_t + \beta^T\boldsymbol{x}_t \sim N(\beta^T\boldsymbol{x}_t, \sigma_\epsilon^2)$ and $\boldsymbol{y}^* = y_{1:n}^*$ so that the calculation is reduced to a pure spike-and-slab regression problem. Now the conditional posterior $p(\gamma, \beta, \sigma_\epsilon^2)$ can be factored into $p(\gamma|\boldsymbol{y}^*)p(\,\sigma_\epsilon^{-2}|\gamma, \boldsymbol{y}^*)p(\beta\ |\gamma, \sigma_\epsilon, \boldsymbol{y}^*)$ and because of conjugacy β and σ_ϵ^{-2} can be integrated out. Then the marginal posterior of γ and the conditional posteriors of β_γ and σ_ϵ^{-2} are[6]

$$\gamma|\boldsymbol{y}^* \sim C(\boldsymbol{y}^*)\frac{|\Omega_\gamma^{-1}|^{\frac{1}{2}}}{|V_\gamma^{-1}|^{\frac{1}{2}}}\frac{p(\gamma)}{SS_\gamma^{\frac{N}{2}-1}} \tag{16}$$

$$\beta_\gamma|\sigma_\epsilon\ , \gamma, \boldsymbol{y}^* \sim N\left(\tilde{\beta}_\gamma\, , \sigma_\epsilon^2\left(V_\gamma^{-1}\right)^{-1}\right) \tag{17}$$

$$\frac{1}{\sigma_\epsilon^2}|\gamma\, , \boldsymbol{y}^* \sim \Gamma(\frac{N}{2}\, , \frac{SS_\gamma}{2}) \tag{18}$$

where the data sufficient statistics are

$$V_\gamma^{-1} = (\boldsymbol{X}^T\boldsymbol{X})_\gamma + \Omega_\gamma^{-1} \qquad \tilde{\beta}_\gamma = \left(V_\gamma^{-1}\right)^{-1}\left(\boldsymbol{X}_\gamma^T\boldsymbol{y}^* + \Omega_\gamma^{-1}b_\gamma\right)$$

$$N = \nu + n \qquad SS_\gamma = ss + \boldsymbol{y}^{*T}\boldsymbol{y}^* + b_\gamma^T\Omega_\gamma^{-1}b_\gamma - \tilde{\beta}_\gamma^T V_\gamma^{-1}\tilde{\beta}_\gamma$$

[6] Details can be found in Gelman et al. (2002).

and $C(\boldsymbol{y}^{*})$ is an unknown normalizing constant that is not required to be computed explicitly by the MCMC algorithm. In order to simulate β and σ_{ϵ}^{2} from the posterior distribution $p(\beta, \sigma_{\epsilon}^{2} | \boldsymbol{y}, \boldsymbol{\alpha}, \theta)$ the stochastic search variable selection (SSVS) algorithm by George & McCulloch (1997) is applied. A sequence of Gibbs sampling steps is thereby used to draw from (16), followed by a draw from (17) and (18). Although improvements to the SSVS algorithms have been proposed, the authors note that the SSVS algorithm obtains satisfactory results.[7]

2.4 The Markov Chain Monte Carlo algorithm

2.4.1 Estimating the Model Parameters

Having specified the priors, a Markov Chain Monte Carlo (MCMC) algorithm is applied to simulate the model parameters and estimate forecasts based on these samples. Let θ denote the subset of model parameters that are not associated with the static regression parameters β and σ_{ϵ}^{2}. Then the posterior of the model parameters $\phi = (\beta, \sigma_{\epsilon}^{2}, \theta)$ and state component $\boldsymbol{\alpha}$ given the data $\boldsymbol{y}$ is

$$p(\phi, \boldsymbol{\alpha} | \boldsymbol{y}) \propto p(\phi) p(\alpha_0) \prod_{t=1}^{n} p(y_t | \alpha_t, \phi) p(\alpha_t | \alpha_{t-1}, \phi) \qquad (19)$$

A MCMC-algorithm that repeatedly cycles through the following three steps (as described in Scott & Varian, 2014) is employed to simulate the posterior distribution (19):

1. Simulate the state component $\boldsymbol{\alpha}$ from $p(\boldsymbol{\alpha} | \phi, \boldsymbol{y})$ using the Kalman filtering and smoothing method by Drubin & Koopman (2002)
2. Simulate $\theta \sim p(\theta | \boldsymbol{y}, \boldsymbol{\alpha}, \beta, \sigma_{\epsilon}^{2})$
3. Simulate β and σ_{ϵ}^{2} from a Markov chain with stationary distribution $p(\beta, \sigma_{\epsilon}^{2} | \boldsymbol{y}, \boldsymbol{\alpha}, \theta)$.

[7] In the meantime there is another method implemented in the BSTS-package (version 0.6.1) called orthogonal data augmentation (ODA) by Ghosh & Clyde (2011).

The algorithm produces a sequence of draws $(\phi, \boldsymbol{\alpha})_0$, $(\phi, \boldsymbol{\alpha})_1, \ldots$ from a Markov chain with stationary distribution $p(\phi, \boldsymbol{\alpha}|\boldsymbol{y})$, the posterior distribution stated above. For details on step 2 and 3 see Section 2.3.1. As noted in Scott, Varian (2013), the key point is that conditional on $\boldsymbol{\alpha}$, the regression components are independent from the other state components. Thus the conditional probability $p(\theta, \beta, \sigma_\epsilon^2|\boldsymbol{\alpha}, \boldsymbol{y})$ can be factorized into $p(\theta, \beta, \sigma_\epsilon^2|\boldsymbol{\alpha}, \boldsymbol{y}) = p(\theta|\boldsymbol{\alpha}, \boldsymbol{y})p(\beta, \sigma_\epsilon^2|\boldsymbol{\alpha}, \boldsymbol{y})$ and consequently the sampling process decomposes into several simulations from different conditional posterior distributions.

2.4.2 Predictions

A typical method to produce forecasts in Bayesian data analysis is to use posterior simulations to generate draws from the posterior predictive distribution $p(\tilde{y}|\boldsymbol{y})$, where $\tilde{y}$ denotes the time series to be forecasted

$$p(\tilde{y}|\boldsymbol{y}) = \int p(\tilde{y}|\phi)p(\phi|\boldsymbol{y})\, d\phi. \tag{20}$$

Simulations from the posterior predictive distribution are generated by iterating equation (1) and (2) forward (starting from α_n) using the sequence of parameter draws $(\phi, \boldsymbol{\alpha})_0$, $(\phi, \boldsymbol{\alpha})_1, \ldots$ from the Section 2.4.1. Thus the resulting forecasts are based on different parameter simulations, particularly with regard to the regression coefficients β. This property is typical for Bayesian model averaging and accounts automatically for model uncertainty.

Now the sampled values from the posterior predictive distribution $p(\tilde{y}|\boldsymbol{y})$ can be used to produce summary statistics such as arithmetic mean[8] or median. For a comprehensive assessment of the model estimates, it is suggested to use graphical representations like dynamic distribution plots, significance plots, or histograms.

[8] The arithmetic mean is a Monte Carlo estimate of $E(\tilde{y}|\boldsymbol{y})$.

3 Nowcasting Online Retail Trade with Google Trends

A prominent application of BSTS-models is the calculation of economic nowcasts. Originally, the term *‘nowcast’* - a contraction of the terms *‘now’* and *‘forecast’* - describes methods that are used for short-term weather forecasts in meteorology. In economics this term has a slightly different meaning: It describes the prediction of an unknown variable that is indexed at time t (denoted as y_t) with usage of additional predictors that are already known at time t (denoted as x_t). Strictly speaking, this is not a forecast anymore, but a nowcast. In some circumstances even past data needs to be predicted; in other words some data x_t can be used to estimate the – at that time unknown – value of y_{t-1}.

This is due to the fact that economic data are often released with a substantial lag – and even then the data may be revised. Meanwhile predictor variables of the same time period may be already available. A well-known example for such an application is conducted by Choi & Varian (2012), who use *Google*-search engine statistics to nowcast the level of economic indicators – among others the level of initial claims for unemployment. In their recent article, Scott & Varian (2013) describe the BSTS-method and illustrate it with an application of *Google Trends* data to nowcast initial claims for unemployment.

Inspired by these articles, the BSTS-method is now applied on shopping-related *Google Trends* data to nowcast the index of retail sales turnover via mail order houses or via Internet in Germany. In this exercise, out-of-sample predictions of a baseline BSTS-model and nowcasts of two models augmented with the *Google Trends* variables are estimated. Next to the evaluation of the accuracy of those predictions the spike-and-slab variable selections of the BSTS-method are contrasted to the selections of a stepwise regression algorithm that minimizes AICc. The stepwise regression algorithm is also applied in Schmidl (2014), where nowcasts of Austrian unemployment and tourism figures are estimated. In contrast to these applications, the data basis is far larger in the current thesis. While the *Google Trends*-series are manually preselected in Schmidl (2014), resulting in sets of up to 20 GI-variables, the current thesis uses data of 63 *Google Trends*-categories and does not analyse specific search terms.

The remaining section is structured as follows: First, the dataset used for the empirical comparison of the nowcast performances, consisting of consumption data from Eurostat and search query data from *Google Trends*, is described in detail. It is succeeded by a precise explanation of the methodology that is used to estimate the models and to assess their predictions. Finally, the results of these nowcasts are analysed and discussed.

3.1 Data Description

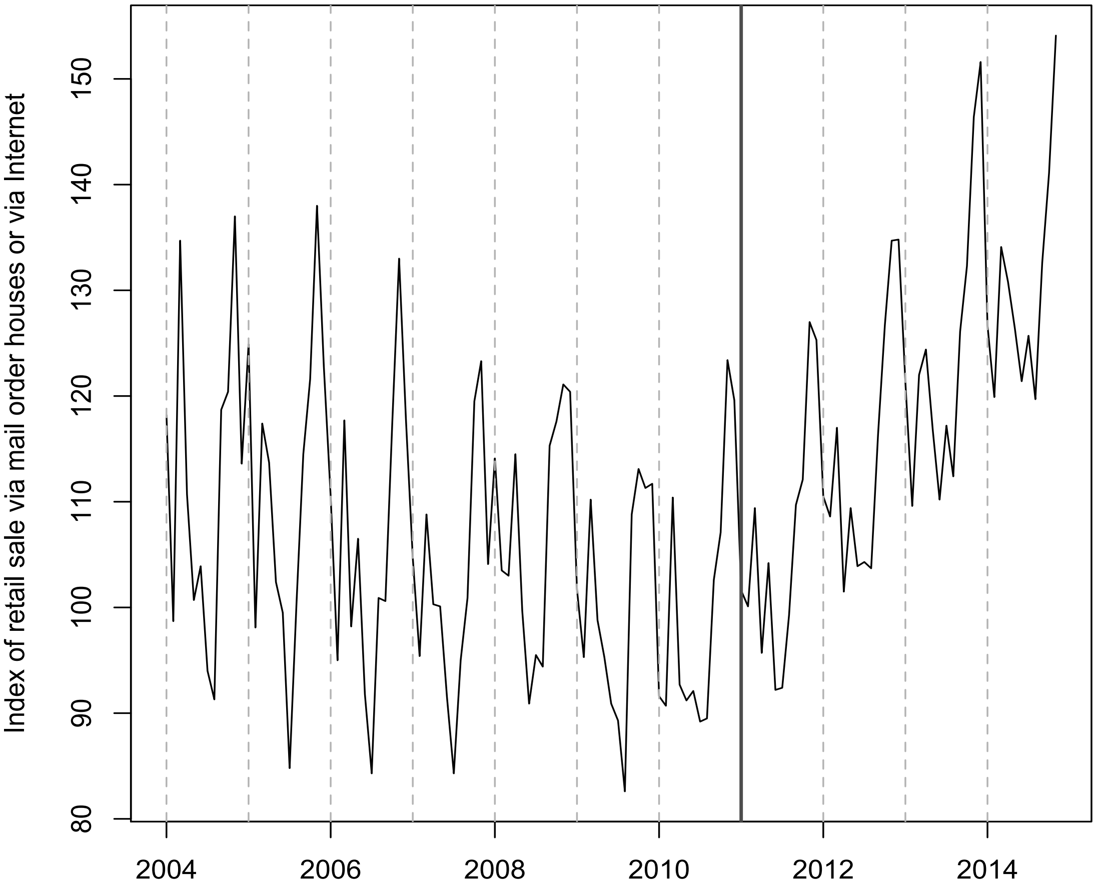
3.1.1 Eurostat Data

The main source for the retail sales data in this thesis is the database of the statistical office of the European Union (Eurostat). Their short-term business statistics database (STS) provides a number of monthly statistics on wholesale and retail trade, where various indices on *turnover and volume of sales in wholesale and retail trade* are available. These indicators are indexed to the base year 2010, for which the average value is equal to 100. The time series are available with and without seasonally adjustments, but since *Google* does not seasonally adjust their *Google Trends* series, the gross data on *turnover and volume of sales in wholesale and retail trade* is taken as a basis for further calculations. Moreover Eurostat publishes the STS indicators according to the NACE Rev. 2 classifications of economic activities. In this context the NACE Rev. 2 class on '*Retail sale via mail order houses or via Internet'* (Code G4791)[9] is selected. For details on the methodology and the legal rules regarding the definitions of indicators, reference periods, etc. visit the Eurostat web page[10].

[9] Abbreviated as 'index of retail sales'.

[10] http://ec.europa.eu/eurostat/web/short-term-business-statistics/methodology
http://ec.europa.eu/eurostat/web/short-term-business-statistics/legislation

Figure 1: Turnover-Index of retail sale via mail order houses or via Internet in Germany (2010=100)



Source: Eurostat (2015), own illustration.

3.1.2 Google Trends

Google search engine statistics are available via *Google Trends*[11], a web service that provides a query index, which indicates the frequency of a user-defined query in relation to the total amount of *Google*-search queries. *Google Trends* are computed on a sample of the total *Google* search queries and hence the results vary slightly from day to day.

[11] http://www.google.com/trends/

The analysis can be limited to specific time periods, regions and search categories. Users can select a time period within January 1, 2004 and the present day. Regions can be filtered from a state level down to a territorial, respectively federal level. *Google* classifies search queries into certain categories, for example the query ‘*red wine’* is classified to the categories [*Food&Drink*], [*Alcoholic Beverages*] and [*Wine*]. These classifications allow the analysis of the amount of queries related to a certain search category in comparison to the total amount of search queries. Besides that the development of the popularity of certain queries in a category can be analysed. In this application the entire categories including all subcategories related to [*Shopping*] are used as predictor variables. Overall these are 103 categories with data available for 63 time series.

The query index is calculated in two steps: First, a query share (21) is computed by calculating the ratio of the total query volume of the user-defined search term or category to the total number of search queries in that region during the selected time period. Secondly, the query share is scaled (22) from 0 to 100 by division with the highest observed value of the query share.

$$query\ share = \frac{search\ volume\ of\ the\ user-defined\ term\ or\ category}{total\ search\ volume} \tag{21}$$

$$scaled\ value = \frac{query\ share}{highest\ value\ of\ query\ share} \cdot 100 \tag{22}$$

The resulting query index is by default on a weekly basis and in case of low traffic for the search term the query index is set to zero.[12] *Google Trends* data are available almost in real time and can be downloaded by users having a *Google* account. For details on the development process of *Google Trends* and its features consult Schmidl (2014) or the *Google Trends*-web page.

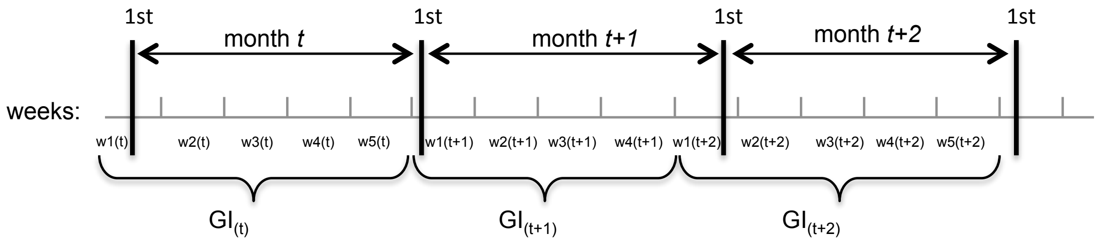
[12] For shorter time spans the query index is available on a daily basis. The Google Trends data used in the present analysis covers the period from January 1, 2004 to January 17, 2015. Hence the query indices are solely on a weekly basis.

3.1.3 Google Index

The above described predictor variables (*Google Trends*) have a higher frequency than the depended variables (*Index on retail sale via mail order houses or via Internet*). According to the methodology used in Schmidl (2014) the mixed frequency structure of the data is not modelled. Therefore the weekly *Google Trends* time series are transformed to monthly *Google Index* (GI) variables.

The split criteria for the transformations are in each case the weeks including the first day of the month. The GI for month *t* is then calculated as the average value of the split week and its successive weeks up to the week before the next split. The described time structure is depicted in Figure 2. To give an example for the GI calculation: October 1, 2014 is on Wednesday of the 40th calendar week; November 1, 2014 is on Saturday of the 44th calendar week. Thus the GI for October 2014 is the average value of the *Google Trends* values related to the weeks 40, 41, 42 and 43. The GI can be calculated on basis of all values available up to the current date. Therefore the current month's GI can be calculated even if the month's last week has not been passed.

Figure 2: Google Index – time structure



Source: Own illustration.

Now focus on the release calendar of Eurostat. Retail trade figures are usually released with a lag of one month, which means that in the example the October figures for 2014 are released on December 3, 2014. Using the GI (calculated according to the above described time structure) the current month's retail sales index can be estimated on basis of real data of the same period up to eight weeks before the official release. For comparability to the stepwise regression model the GI-variables are used in the BSTS-calculations as well.

3.2 Methodology

In this framework GI-data are used to directly estimate the index of retail sales turnover. The available data are separated into a training set and a test set, where the training set contains approximately two thirds of the observations. The training set consists of the data within the period of January 2004 and June 2011 (90 months) and the test set ranges from July 2011 to November 2014 (41 months). The models are estimated on basis of the training set and are further used to generate predictions for the test period.

This exercise compares nowcasts of two GI-augmented models to predictions of a baseline BSTS-model that is estimated entirely on basis of (past) turnover index-values. The baseline model (23) combines a local level component with a seasonal component to capture the seasonality of the (gross) retail figures.

$$\begin{aligned} y_t &= \mu_t + \tau_t + \epsilon_t & \epsilon_t &\sim N(0, \sigma_\epsilon^2) \\ \mu_t &= \mu_{t-1} + u_t & u_t &\sim N(0, \sigma_u^2) \\ \tau_t &= -\sum_{i=1}^{11} \tau_{t-i} + v_t & v_t &\sim N(0, \sigma_v^2) \end{aligned} \tag{23}$$

The spike-and-slab regression model (24) captures the seasonal pattern by using the GI-series (denoted as $\mathbf{x}$). Thus, the model consists of a local level component and a regression component.

$$\begin{aligned} y_t &= \mu_t + \beta^T \mathbf{x}_t + \epsilon_t & \epsilon_t &\sim N(0, \sigma_\epsilon^2) \\ \mu_t &= \mu_{t-1} + u_t & u_t &\sim N(0, \sigma_u^2) \end{aligned} \tag{24}$$

Besides the BSTS-models, a standard regression model is estimated as a benchmark model. Therefore, an adapted version of Venables & Ripleys (2002) stepwise model selection algorithm, which is implemented via the function *stepAIC* of the R-package *MASS*, is used. Beginning with an intercept only-model this function repeatedly adds (or removes) variables to the model until a selected information criterion is minimized. The information criterion can be adjusted with respect to the number of degrees of freedom k. The default value $k = 2$ gives the AIC, $k = \log(n)$ is referred to as BIC/SBC. However, Burnham & Anderson (2002) argue that AIC has theoretical advantages over BIC and they recommend the usage of the AIC with correction for finite samples sizes, which is

$$AICc = AIC \ + \ \frac{2k(k \ + \ 1)}{n - k - 1} = -2\log L \ + \ 2k \ + \ \frac{2k(k \ + \ 1)}{n - k - 1} \tag{25}$$

Thus the *stepAIC*-function is adapted with respect to the information criterion, so that the stepwise algorithm minimizes the AICc-criterion. The final model[13] resulting from the stepwise model selection algorithm can be formulated as:

$$y_t \ = \alpha \ + \beta^T \mathbf{x}_t + \epsilon_t \qquad\qquad \epsilon_t \ \sim \ N\,(0, \sigma^2) \tag{26}$$

Predictions are estimated for the entire test period (41 predictions) on basis of these three models. While point estimates can be directly estimated using the function *predict.lm*[14] for the stepwise-model*,* predictions for BSTS-models are sampled from the posterior predictive distribution. In this application a total number of 1000 MCMC draws are generated from which 100 draws are discarded as burn-in. Finally the point estimates are calculated as the median values of these MCMC draws.

3.3 Results

For the evaluation of the results, first the model fits of both types of methods are analysed and then the out-of-sample nowcasts are compared. The BSTS model fits are summarized by graphical methods such as dynamic density plots, histograms and significance plots, where in particular the regression component of the spike-and-slab model is closely examined. Then the model specifications of the (final) stepwise model are analysed, with specific regard to the GI-selections. The main assumptions of the stepwise model are tested by applying the Breusch-Godfrey test for serial correlation, the Breusch-Pagan test for homoscedasticity and Shapiro-Wilk test for normality of the residuals. Finally the out-of-sample predictions of all models are compared by their mean

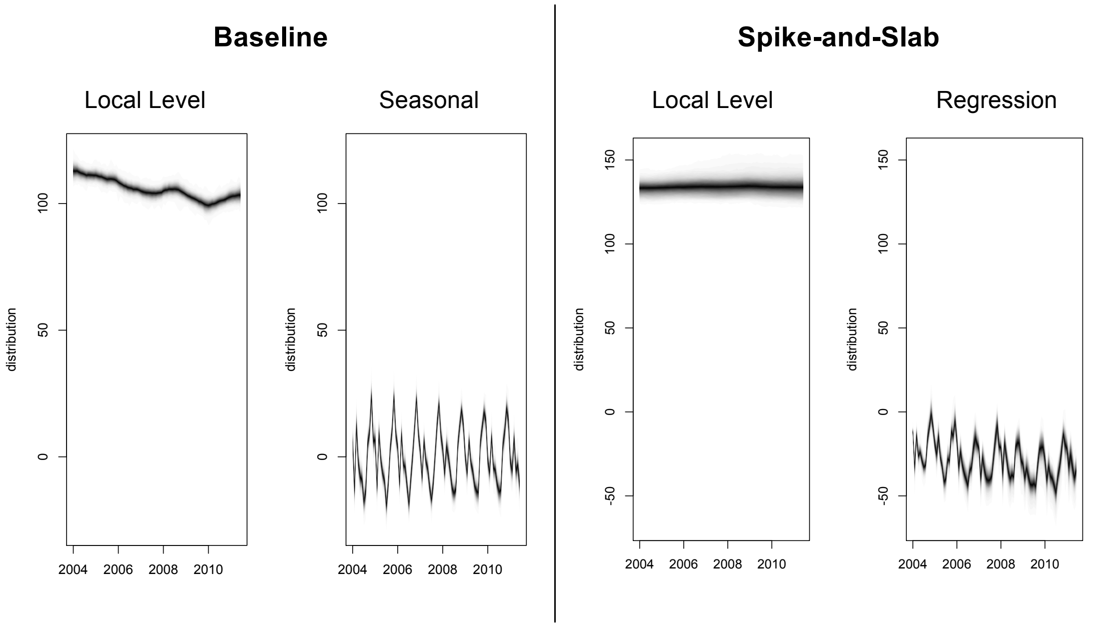
[13] Also denoted as stepwise model.

[14] Available via the R-Package *stats* (R Core Team 2014).

absolute prediction errors (MAE) and their mean absolute scaled errors (MASE)[15]. Besides these statistics, plots of the forecasts and the cumulative prediction errors are presented.

The differences between the baseline (seasonal) model and the spike-and-slab regression model are visible in Figure 3, where the contribution of each state component is plotted. The distribution of the local level component of the baseline model is not static – at the beginning of the timeline it is distributed around the value 110 and declines gradually. The seasonal component shows the expected regular pattern around zero. On the other hand the values of the local level component of the spike-and-slab regression model are more or less constant; the mean varies between the values 131 and 133 along the timeline. The reason for this difference is that the regression component explains the seasonality and the variation of the trend as well.

Figure 3: Contributions of the state components



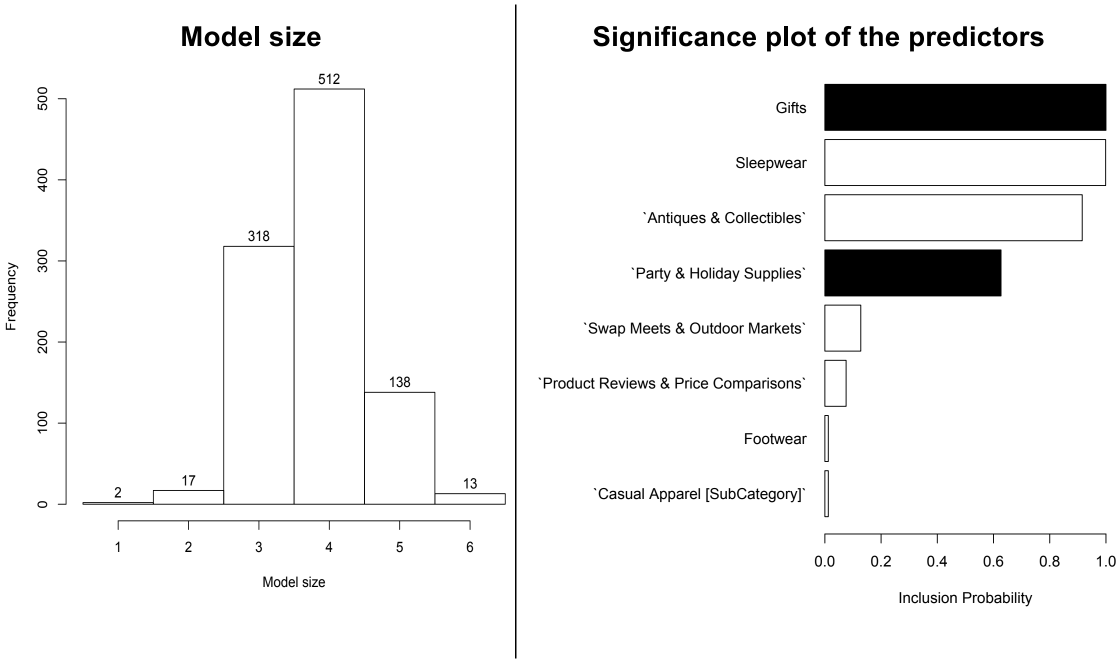
Source: Own illustration.

[15] For details see Section 4.2.

Figure 4 shows some properties of the spike-and-slab model's regression component. The majority of the regression models contain three predictors; in total approximately 97% of the models contain two to five predictors. The significance plot shows which GI-series are most likely included in the model and indicates the signs of the corresponding coefficients. In fact, eight out of 63 variables have an inclusion probability higher than 0.1 and only four variables have an inclusion probability higher than one half. The GI variables for the *Google Trends* categories *[Gifts]* and *[Sleepwear]* are most frequently included: *[Gifts]* in 976 and *[Sleepwear]* in 975 models. Other often included variables are *[Antiques & Collectibles]* (824 times) and *[Party & Holiday Supplies]* (636 times).

In comparison to the stepwise model fit, which is summarized in Table 1, the variable selections of the spike-and-slab model are quite similar: Four of the variables with inclusion probability larger than 0.1 of the spike-and-slab model are also included in the stepwise model. Beyond that the coefficients of these variables have the same sign as the coefficients in the spike-and-slab model. It is not surprising that the stepwise model includes more variables than the spike-and-slab model. This is due to the fact that AICc penalizes additional coefficients not as strong as other information criteria (like BIC) and on the other hand the BSTS-method averages over a large number of relatively sparse models with different parameterizations.

Figure 4: Model size and significance of the predictors of the spike-and-slab model



Source: Own illustration.

Explanatory note: The colour of the bars indicates the probability of the coefficients being positive. Positive coefficients are white, negative coefficients are black. Only predictors with posterior inclusion probabilities of at least 0.1 are shown.

Table 1: Stepwise model fit

```
Residuals:
     Min       1Q   Median       3Q      Max
-12.6283  -2.7898  -0.0979   3.1884  14.8227

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    118.79179    2.55159  46.556  < 2e-16 ***
`Book Retailers`                 0.59312    0.09156   6.478 7.31e-09 ***
Outerwear                        0.13324    0.04858   2.743 0.007536 **
Gifts                           -0.14684    0.02482  -5.916 8.06e-08 ***
`Formal Wear`                    0.26720    0.06623   4.034 0.000126 ***
`Swap Meets & Outdoor Markets`   0.13324    0.02561   5.203 1.51e-06 ***
`Clothing Accessories`          -0.51716    0.17048  -3.034 0.003269 **
Sleepwear                        0.36306    0.08964   4.050 0.000119 ***
Classifieds                     -0.07955    0.03592  -2.214 0.029688 *
`Party & Holiday Supplies`      -0.13893    0.05852  -2.374 0.020027 *
Headwear                         0.14025    0.09149   1.533 0.129289
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.853 on 79 degrees of freedom
Multiple R-squared:  0.8122,	Adjusted R-squared:  0.7885
F-statistic: 34.17 on 10 and 79 DF,  p-value: < 2.2e-16
```

Source: Own calculations.

Table 2 states the result of the above-mentioned hypothesis tests. None of these tests reject their corresponding null hypothesis at a 5%-significance level.

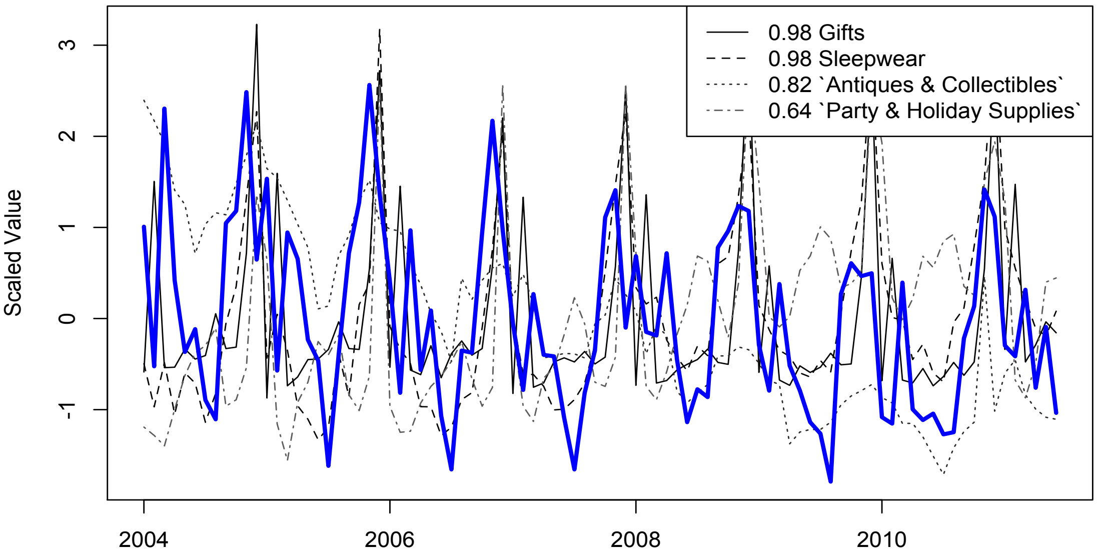
Table 2: Statistical hypothesis tests of the stepwise-fit

	Test statistic	degrees of freedom	p-value
Breusch-Godfrey test (Order=1)	2.550	1	0.110
Breusch-Godfrey test (Order=2)	3.211	2	0.201
Breusch-Pagan test	14.008	10	0.173
Shapiro-Wilk test	0.983	(-)	0.299

Source: Own calculations.

The predictors with the highest inclusion probability are shown in Figure 5. Obviously, the GI-series strongly resemble the seasonal pattern of the retail sales index. Noteworthy are the mid-seasonal peaks of the GI-variable *[Party & Holiday Supplies].* These counter movements (compared to the target series) are possibly the reason for the negative signs of the coefficients for this variable.

Figure 5: Top predictor variables vs. the level of retail sales



Source: Own illustration.

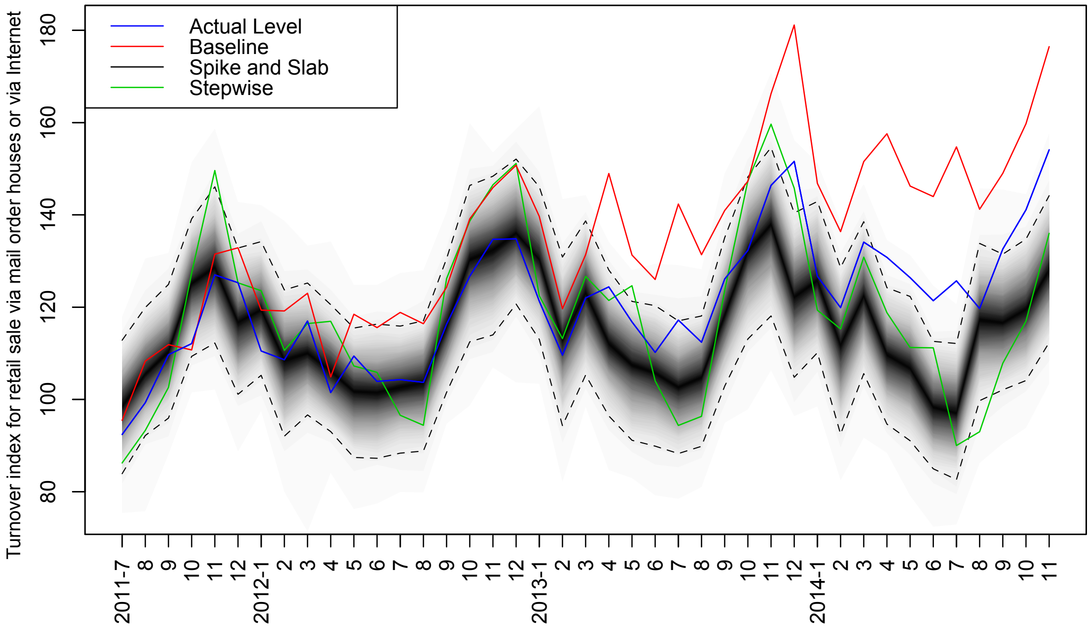
Explanatory note: The level of retail sale via mail order houses or via internet is plotted in blue. The plot shows predictors with inclusion probabilities of at least 0.4.

Figure 6 shows the out-of-sample predictions of all three models. The plot illustrates the point estimates of the baseline and the stepwise model as well as the actual values of the retail sales index. It also shows the posterior predictive distribution of the spike-and-slab model, which gives an intuitive impression of the predictive accuracy. The point estimates are the median values of the draws, located in the very dense area of the distribution.

It is obvious that the baseline model performs worst in comparison to the other two models. The predictions of the baseline model are too high almost across the entire test period and overshoot quite substantially in the second half of the test set.

With respect to the GI-models, the picture is not so clear. It can be observed that the predictions of the spike-and-slab model fluctuate not as strongly as the predictions of the stepwise model, which is due to Bayesian model averaging. As a result of these larger variations the stepwise model performs worse compared to the spike-and-slab model in two thirds of the predictions, precisely in 27 of 41 instances.

Figure 6: Out-of sample predictions



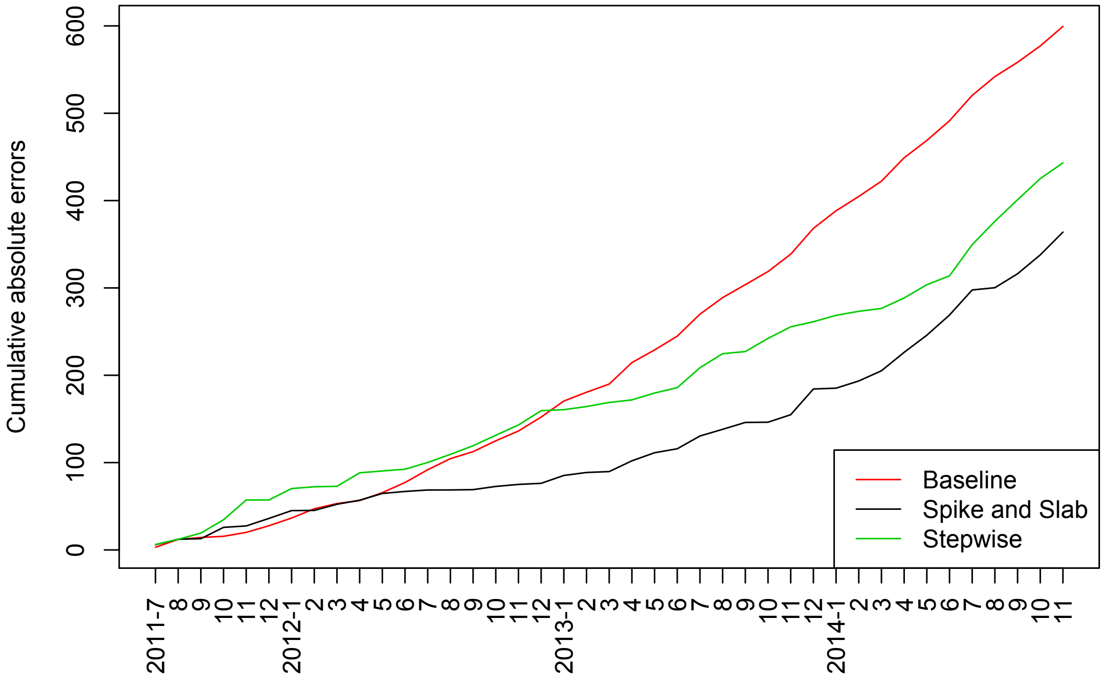
Source: Own illustration.

Explanatory note: The shaded area represents the posterior predictive distribution of the spike-and-slab model. The dotted lines represent the 0.025 and 0.975 quantiles.

Figure 7 shows the cumulative sum of the absolute prediction errors. In case of a divergent growth of the graphs it can be concluded that there is a significant difference related to the forecast accuracy. If we compare the graphs of the stepwise and spike-and-slab model to the graph of the baseline model, this is clearly the case. At the end of the test period there is a substantial gap between these curves.

The spike-and-slab model and the stepwise model do not show such a clear divergence. Although the cumulative prediction errors of the spike-and-slab model are lower across the whole period, it is uncertain if these lines are divergent. In fact, the gap between the graphs in summer 2014 is smaller than the gap the year before.

Figure 7: Cumulative absolute prediction errors



Source: Own illustration.

A comparison of the error measures supports these findings: The baseline model has both the highest MAE (14.619) and the highest MASE (1.341). It is followed by the stepwise model, which yields a MAE of 10.810 and a MASE of 0.991. In these terms, the spike-and-slab model has the highest predictive accuracy (8.876, respectively 0.814).

4 A Benchmark of the Forecast Accuracy

In this section, the focus lies on the comparison between the forecast accuracy of state space models estimated by the BSTS-method and the forecast accuracy of other widely used methods. The basis for the comparison provides the M3-Competition (Makridakis & Hibon, 2000), an empirical study that compares the predictive performance of a large number of time series methods. Moreover the M3-Competition dataset is described and the results and conclusions of the M3-Competition are reflected – particularly with respect to different evaluation criteria. After this, the methodological framework, which is used to estimate the BSTS-models in the M3-Competition, is explained. The results of the BSTS-forecasts in context of the M3-Competition are discussed at the end of this section.

4.1 The M3-Competition

The M3-Competition (Makridakis & Hibon, 2000) is the successor of the M- and M2-Competition[16], a series of empirical studies that compare the forecast accuracy of various forecasting methods. The M3-Competition dataset contains overall 3,003 time series, including six types of time series data (micro, macro, industry, finance, demographic and other) and four frequencies (yearly, quarterly, monthly and other). The time series were selected on a quota basis and have each a fixed set of observations to forecast a certain number of periods. Table 3 shows the distribution of the 3,003 time series according to the data types and the time interval. It also states the minimum number of observations for each frequency and the corresponding number of forecasts to be made.

[16] Also known as the Makridakis Competitions or M Competitions.

Table 3: The M3-Competition dataset

Frequency (Min. number of observations / Number of periods to forecast)	**Types of M3-Competition data**						
	Micro	Industry	Macro	Finance	Demo-graphic	Other	**Total**
Yearly (14/6)	146	102	83	58	245	11	**645**
Quarterly (16/8)	204	83	336	76	57		**756**
Monthly (48/18)	474	334	312	145	111	52	**1,428**
Other (60/8)	4			29		141	**174**
Total	**828**	**519**	**731**	**308**	**413**	**204**	**3,003**

In total, 24 time series methods are included in the M3-Competion. The applied methods can be classified into six categories: naïve/simple-methods, explicit trend models, decomposition methods, ARIMA/ARARMA models, expert systems and neural networks. A brief description of the 24 methods can also be found in Makridakis & Hibon (2000). Analysing the results, the authors find four major conclusions: (1) *"Statistically sophisticated or complex methods do not necessarily produce more accurate forecasts than simpler ones*. (2) *The rankings of the performance of the various methods vary according to the accuracy measure being used.* (3) *The accuracy of the combination of various methods outperforms, on average, the specific methods being combined and does well in comparison with other methods.* (4) *The accuracy of the various methods depends upon the length of the forecasting horizon".*

4.2 Evaluation

In the M3-Competition, five different accuracy measures are utilized to analyse the performance of the 24 forecasting methods[17]: symmetric mean absolute percentage error (sMAPE), average ranking, a percentage better measure, median symmetric absolute percentage error and median relative absolute error. However, Hyndman & Koehler (2006) discuss and compare accuracy measures used for univariate time series forecasts – in particular the accuracy measures of the M3-Competition. They conclude that all these accuracy measures have some disadvantages and might give infinite or undefined values in certain situations. Instead they propose the mean absolute scaled error (MASE) as a standard measure for forecast accuracy comparisons. In order to preserve comparability both sMAPE, which served as the main error metric in the M3-Competiton, and the MASE are reported in the present analysis. These accuracy measures are defined as:

Symmetric Mean Absolute Percentage Error (sMAPE)

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n} 200 * \frac{|F_t - A_t|}{F_t + A_t} \tag{27}$$

Mean Absolute Scaled Error (MASE)

$$MASE = \frac{1}{n}\sum_{t=1}^{n} \frac{|F_t - A_t|}{\frac{1}{m-1}\sum_{i=2}^{m} |Y_i - Y_{i-1}|} \tag{28}$$

where F_t denotes the forecast values, A_t the actual values of a sequence of n forecasts and $Y_i - Y_{i-1}$ are the one-step in-sample forecast errors of the naïve method.

[17] Makridakis & Hibon (2000) compare 24 methods; Hyndman & Koehler (2006) add the HKSG-method in their calculations.

4.3 Methodology

The BSTS-package provides flexible tools to produce forecasts – in fact there is a wide range of options to configure a model and evaluate its goodness of fit. Since it is beyond the scope of this thesis to evaluate various model configurations for each and every time series of the M3-Competition by hand, an automatic procedure is applied. Per time series a set of BSTS-models (see Section 4.4) are fitted and in each case the (in-sample) one-step-ahead prediction errors are calculated. That model, which yields the minimal cumulative absolute mean errors, is selected to produce the final forecasts. Therefore, draws from the posterior predictive distribution are generated for the number of periods specified in Table 3 and summarized by their median.[18] These median values are subsequently used as the forecasts (point estimates) of the BSTS-model. In order to compare the forecast accuracy the error measures defined in Section 4.2 are calculated based on these values and averaged across all forecast horizons and across all series, with partial results for yearly, quarterly, monthly and other data. The R-source code for these calculations can be found in Appendix 7.1.

4.4 State Specifications

The following set of state space models is used in the subsequent analysis. In case of monthly and quarterly data additional models that include a seasonal component with seasonality $S = 12$ respectively $S = 4$ are also taken into account (model 7 to 10). For all models the default prior parameters supplied by the BSTS-package are used (details can be found in the accompanying manual of the package). Table 4 lists the set of models used for the forecast competition. Details on each single component can be found in Section 2.2.

[18] For each model 1000 MCMC draws are generated; 100 draws are discarded as burn-in.

Table 4: BSTS-models used for M3-Competition

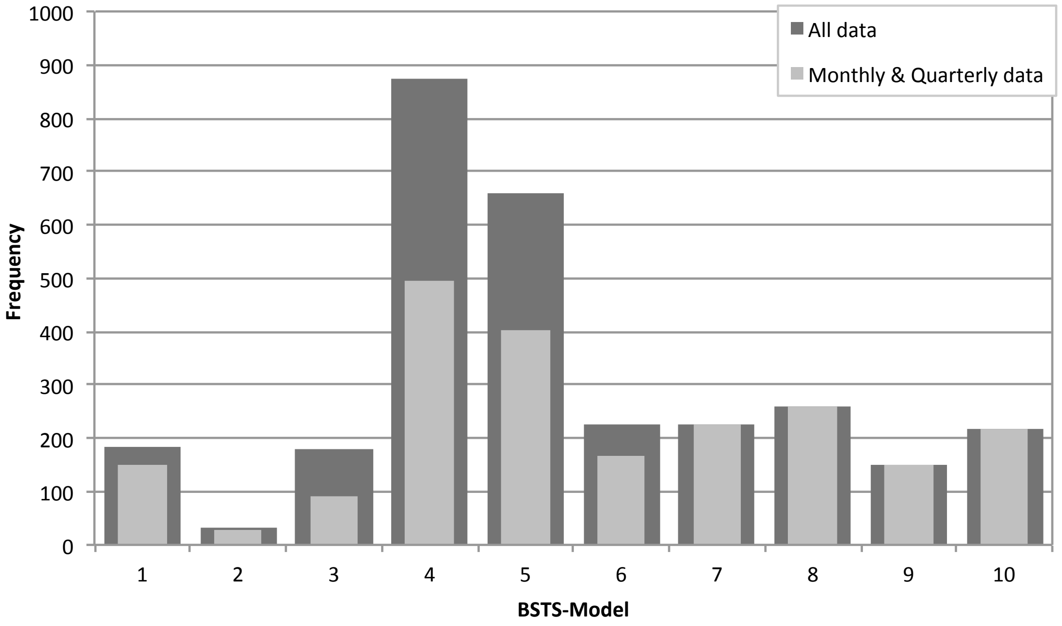
1	**Local level**	$y_t = \mu_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_t = \mu_{t-1} + u_t$	$u_t \sim N(0, \sigma_u^2)$
2	**Local linear trend**	$y_t = \mu_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim N(0, \sigma_\mu^2)$
		$\delta_{t+1} = \delta_t + \eta_t$	$\eta_t \sim N(0, \sigma_\delta^2)$
3	**Robust local linear trend**	$y_t = \mu_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim \mathcal{T}_{\nu_\mu}(0, \sigma_\mu^2)$
		$\delta_{t+1} = \delta_t + \eta_t$	$\eta_t \sim \mathcal{T}_{\nu_\delta}(0, \sigma_\delta^2)$
4	**Generalized local linear trend**	$y_t = \mu_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim N(I, \sigma_\mu^2)$
		$\delta_{t+1} = D + \phi(\delta_t - D) + \eta_t$	$\eta_t \sim N(I, \sigma_\delta^2)$
5	**Generalized local linear trend + AR(1)**	$y_t = \mu_t + \alpha_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim N(I, \sigma_\mu^2)$
		$\delta_{t+1} = D + \phi(\delta_t - D) + \eta_t$	$\eta_t \sim N(I, \sigma_\delta^2)$
		$\alpha_t = \phi_1 \alpha_{t-1} + w_{t-1}$	$w_t \sim N(0, \sigma^2)$
6	**Local level + AR(1)**	$y_t = \mu_t + \alpha_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_t = \mu_{t-1} + u_t$	$u_t \sim N(0, \sigma_u^2)$
		$\alpha_t = \phi_1 \alpha_{t-1} + w_{t-1}$	$w_t \sim N(0, \sigma^2)$
7	**Local level + Seasonal**	$y_t = \mu_t + \tau_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_t = \mu_{t-1} + u_t$	$u_t \sim N(0, \sigma_u^2)$
		$\tau_t = -\sum_{i=1}^{S-1} \tau_{t-i} + v_t$	$v_t \sim N(0, \sigma_v^2)$
8	**Generalized local linear trend + Seasonal**	$y_t = \mu_t + \tau_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim N(I, \sigma_\mu^2)$
		$\delta_{t+1} = D + \phi(\delta_t - D) + \eta_t$	$\eta_t \sim N(I, \sigma_\delta^2)$
		$\tau_t = -\sum_{i=1}^{S-1} \tau_{t-i} + v_t$	$v_t \sim N(0, \sigma_v^2)$
9	**AR(1) + Seasonal**	$y_t = \alpha_t + \tau_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\alpha_t = \phi_1 \alpha_{t-1} + w_{t-1}$	$w_t \sim N(0, \sigma^2)$
		$\tau_t = -\sum_{i=1}^{S-1} \tau_{t-i} + v_t$	$v_t \sim N(0, \sigma_v^2)$
10	**Generalized local linear trend + Seasonal + AR(1)**	$y_t = \mu_t + \tau_t + \alpha_t + \epsilon_t$	$\epsilon_t \sim N(0, \sigma_\epsilon^2)$
		$\mu_{t+1} = \mu_t + \delta_t + u_t$	$u_t \sim N(I, \sigma_\mu^2)$
		$\delta_{t+1} = D + \phi(\delta_t - D) + \eta_t$	$\eta_t \sim N(I, \sigma_\delta^2)$
		$\tau_t = -\sum_{i=1}^{S-1} \tau_{t-i} + v_t$	$v_t \sim N(0, \sigma_v^2)$
		$\alpha_t = \phi_1 \alpha_{t-1} + w_{t-1}$	$w_t \sim N(0, \sigma^2)$

4.5 Results

Before the accuracy of the BSTS-predictions is analysed in comparison to the results of the M3-Competition, we examine the structure of the BSTS-models that are used for the predictions. Figure 8 illustrates the frequency of how often the BSTS-models (see Table 4) yield the minimum absolute (in sample) one-step-ahead prediction errors. Consequently Figure 8 shows how many times the models are selected to produce the final forecasts. Overall, the most frequently selected models are model 4 and 5, which are used to estimate the forecasts of approximately every second time series of the M3-Competition (1,530 of 3,003). In consideration of the subsample of the monthly and quarterly time series[19], where seasonal models are estimated as well, the majority of the forecasts are still estimated by model 4 and 5 (41%); another 39% of the forecasts are estimated by the models that include a seasonal component (model 7 to 10). The figure also shows that more complex models with several components do not necessarily produce the best (in-sample) forecasts. In fact, the most complex model in this exercise, a generalized local linear trend model with a seasonal and an AR(1)-component, is selected only 219 times, which account for 10% of the monthly and quarterly time series.

[19] In total 2,184 time series.

Figure 8: Histogram of the BSTS-models with minimum absolute (in sample) one-step-ahead prediction errors



Source: Own illustration.

In order to compare the BSTS-prediction errors with the results of Makridakis & Hibon (2000) as well as with the results of Hyndman & Koehler (2006), both sMAPE and MASE of the BSTS-predictions are calculated for each type of data (yearly, quarterly, monthly, others and all). Table 5 shows the average sMAPE of the BSTS-method in context of the original M3 analysis. Overall, the BSTS-method has an average sMAPE of 13.40 and ranks the third among 25 benchmark methods. Comparing the sub-aggregates, the BSTS-method also has the third lowest sMAPE on quarterly data, whereas it is less accurate with respect to yearly, monthly and other time series.

Table 5: Symmetric mean absolute percentage error (sMAPE) for the M3-forecast competition

	Yearly	Quarterly	Monthly	Other	All
THETA	16.90	**8.96**	**13.85**	4.41	**13.01**
ForecastPro	17.14	9.77	13.86	4.60	13.19
BSTS	*16.83*	*9.30*	*15.09*	*4.67*	*13.40*
ForcX	16.48	9.54	14.45	4.64	13.49
COMB.S.H.D	17.07	9.22	14.48	4.56	13.52
DAMPEN	17.18	9.33	14.59	4.61	13.63
RBF	**16.42**	9.57	14.77	5.60	13.75
THETAsm	17.87	10.07	14.66	4.93	13.88
B.J.auto	17.73	10.26	14.81	5.06	14.01
PP.Autocast	17.05	9.36	15.15	4.62	14.01
Auto.ANN	18.13	10.20	14.93	4.80	14.11
SMARTFCS	17.68	10.15	15.03	4.86	14.13
Flors.Pearc2	17.84	10.43	15.17	4.89	14.29
SINGLE	17.82	9.72	15.32	6.29	14.32
AutoBox2	16.52	9.90	15.69	4.41	14.41
HOLT	19.27	10.67	15.36	4.81	14.60
WINTER	19.27	10.61	15.44	4.81	14.65
Flors.Pearc1	17.21	9.95	15.96	5.09	14.70
ARARMA	18.36	10.12	15.84	**4.38**	14.74
AutoBox1	21.59	10.96	15.83	4.93	15.23
AutoBox3	20.36	10.93	16.18	4.71	15.33
NAIVE2	17.88	9.95	16.91	6.30	15.47
ROBUST.Trend	16.78	9.79	18.40	4.58	16.30

Source: Own calculations; Makridakis & Hibon (2000).

Table 6 shows the average MASE of the BSTS-method paired with the results of Hyndman & Koehler (2006). In comparison to the above-mentioned findings with respect to sMAPE, the BSTS-method performs even better – in fact it has the lowest MASE across all methods (2.19). Moreover, the BSTS-method performs best on monthly data, although the accuracy is rather mediocre with respect to sMAPE. Such large changes also occur in the analysis of Hyndman & Koehler, where Robust-Trend performed third best with respect to MASE, while its sMAPE is the highest among all methods in the original M3 analysis. However the authors state that the application of *“MASE does not substantially affect the main conclusions about the* *best-performing methods"* and with the methodological advantages of MASE in mind, this should be true for the BSTS-method.

Table 6: Mean absolute scaled error (MASE) for the M3-forecast competition

	Yearly	Quarterly	Monthly	Other	All
BSTS	*2.81*	*2.07*	***1.99***	*2.05*	***2.19***
Theta	2.81	**1.97**	2.08	1.95	2.20
Theta-sm	2.81	2.00	2.09	1.95	2.22
Robust-Trend	**2.63**	2.15	2.14	1.92	2.23
Comb SHD	2.88	2.05	2.12	2.09	2.26
ForecastX	2.77	2.22	2.20	1.97	2.31
ForecastPro	3.03	2.35	2.04	1.96	2.33
Dampen	3.03	2.10	2.18	2.08	2.34
HKSG	3.06	2.29	2.15	1.86	2.36
RBF	2.72	2.19	2.27	2.70	2.37
BJ-automatic	3.16	2.21	2.21	2.30	2.42
Flores/Pearce1	2.94	2.23	2.31	2.27	2.42
Holt	3.18	2.40	2.15	2.04	2.43
ARARMA	3.48	2.29	2.07	2.05	2.43
SmartFcs	3.00	2.39	2.23	2.08	2.43
PP-Autocast	3.02	2.12	2.44	2.09	2.46
Autobox3	3.18	2.45	2.23	2.01	2.47
Flores/Pearce2	3.02	2.41	2.27	2.34	2.47
Pegels	3.58	2.30	2.13	**1.85**	2.47
Automati ANN	3.06	2.35	2.34	2.13	2.49
Winter	3.18	2.37	2.43	2.04	2.55
SES	3.17	2.27	2.44	3.14	2.59
Naive2	3.17	2.28	2.50	3.13	2.62
Autobox1	3.68	2.61	2.20	2.12	2.62
Autobox2	2.75	2.20	3.39	1.90	2.87

Source: Own calculations; Hyndman & Koehler (2006).

The results are especially remarkable considering the fact that all BSTS-models are estimated and selected automatically with no further adjustments for individual time series being made. Thus there are certainly improvement possibilities to many of the BSTS-models. In particular other model specifications may be more suitable or prior knowledge and beliefs can be used to model the data more accurately. In this sense, the predictions represent a lower bound for the accuracy of BSTS-models, and thus the results are all the more surprising.

5 Conclusion

This thesis describes the recently published R-package Bayesian Structural Time Series (BSTS) and compares the predictive accuracy of this method both in nowcasting and forecasting applications on empirical data. The main results of these comparisons can be summed up as follows.

In the nowcasting exercise, data on shopping-related *Google* search queries are used to estimate the contemporaneous index *on retail sale via mail order houses or via Internet* in Germany. With focus on the spike-and-slab variable selections of the BSTS-method versus the selections of a stepwise regression that minimizes AICc, the weekly *Google Trends*-verticals are transformed into monthly *Google Indices* (GI) to provide a uniform data basis for the calculations. Although the selected set of GI-variables is overall quite similar, the main characteristics of each method become apparent: The spike-and-slab model averages over many small models, whereas the stepwise model tends to be more complex. Hence, the spike-and-slab method paired with model averaging helps to protect against spurious regressors and dampens excessive variations of single models – a fact that also can be seen in the out-of-sample predictions. The stepwise model's predictions fluctuate more strongly than the predictions of the spike-and-slab model and altogether the spike-and-slab model has the highest out-of-sample predictive accuracy in terms of MAE and MASE in this example. However, it is also shown that both methods perform significantly better than a baseline model that is estimated entirely on the basis of the past values of the target data.

Regarding the forecast accuracy the BSTS-method performs surprisingly well in the M3-Competition. With respect to sMAPE, the BSTS-models have the third highest accuracy among 25 benchmark models and in the terms of the more favourable error measure MASE the BSTS-predictions have the highest accuracy. In view of the automated methodology that is used to estimate and select the BSTS-models, the results represent only a lower bound for the predictive accuracy of the BSTS-method. Further improvements of the models are certainly possible.

The presented applications and comparisons demonstrate the usefulness and flexibility of BSTS. The empirical examples have shown that BSTS also have a good nowcasting as well as a good forecasting performance in comparison to the benchmark methods.

6 References

6.1 Literature

Brodersen K. H., Gallusser F., Koehler J., Remy N., Scott S. L. (2014). Inferring causal impact using Bayesian structural time-series models. *Annals of Applied Statistics (in press)*

Burnham, K. P., & Anderson, D. R. (2002). Model selection and multimodel inference: a practical information-theoretic approach (2nd ed.), Springer, ISBN 0-387-95364-7.

Chipman, H., George, E., McCulloch, R., Clyde, M., Foster, D., Stine, R. (2001). The practical implementation of bayesian model selection. *Lecture Notes-Monograph Series* 65–134.

Choi, H., & Varian, H. (2012). Predicting the present with Google trends. *Economic Record*, 88(s1), 2-9.

Durbin, J., & Koopman, S. J. (2001). Time series analysis by state space methods. Oxford, UK: *Oxford University Press*. ISBN 978-0198523543.

Durbin, J. and Koopman, S. J. (2002). A simple and efficient simulation smoother for state space time series analysis. *Biometrika* 89, 603–616.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2002). Bayesian Data Analysis (2nd ed). Chapman & Hall.

George, E. I., & McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association 88*, 881–889.

George, E. I., & McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica 7*, 339–374.

Ghosh, J., & Clyde, M. A. (2011). Rao–blackwellization for bayesian variable selection and model averaging in linear and binary regression: A novel data augmentation approach. *Journal of the American Statistical Association*, *106*(495).

Harvey, A. C. (1989). Forecasting, structural time series models and the Kalman filter. *Cambridge university press*.

Makridakis, S., Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451-476.

Hyndman, R. J., Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679-688.

Kalman, R. E. (1960): A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D), 35–45.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., Berger, J. O. (2008). Mixtures of g-priors for Bayesian variable selection. *Journal of the American Statistical Association* 103, 410–423.

Madigan, D., Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, *89*(428), 1535-1546.

Schmidl, M. (2014). Nowcasting mit Hilfe von Internet-Suchstatistiken. Thesis, Universität Wien.

Scott, S. L., Varian, H. (2013). Bayesian Variable Selection for Nowcasting Economic Time Series. In *Economics of Digitization*. University of Chicago Press.

Scott, S. L., Varian, H. (2014). Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1), 4-23.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In P. K. Goel and A. Zellner, eds., *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, 233–243. North-Holland/Elsevier.

6.2 R-Packages

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Hyndman, R. J., Akram M., Bergmeir, C. (2013). Mcomp: Data from the M-competitions. R package version 2.05. http://CRAN.R-project.org/package=Mcomp

Scott, S. L. (2014). bsts: Bayesian Structural Time Series. R package version 0.6.1. http://CRAN.R-project.org/package=bsts

Venables, W. N., Ripley, B. D. (2002). Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

7 Appendix

7.1 Source Code

```
library(Mcomp)
library(bsts)

M3Forecast_bsts <- list()
list.names <- paste("V",rep(1:18), sep="")
M3Forecast_bsts$BSTS <- vector("list", length(list.names))
names(M3Forecast_bsts$BSTS) <- list.names


best.model <- NULL
prediction.errors <- list()

ptm <- proc.time()
N <- length(M3)
for (s in 1:N)
{
seasonality <- 0
if(M3[[s]]$period == "QUARTERLY") seasonality <- 4
if(M3[[s]]$period == "MONTHLY") seasonality <- 12

ss <- list()
ss[[length(ss)+1]] <- AddLocalLevel(list(), M3[[s]]$x)                                     #1
ss[[length(ss)+1]] <- AddLocalLinearTrend(list(), M3[[s]]$x)                               #2
ss[[length(ss)+1]] <- AddStudentLocalLinearTrend(list(), M3[[s]]$x)                        #3
ss[[length(ss)+1]] <- AddGeneralizedLocalLinearTrend(list(), M3[[s]]$x)                    #4


 tss <- AddGeneralizedLocalLinearTrend(list(), M3[[s]]$x)
 tss <- AddAr(tss, y=M3[[s]]$x, lags = 1)
ss[[length(ss)+1]] <- tss                                                                  #5
 tss <- AddLocalLevel(list(), M3[[s]]$x)
 tss <- AddAr(tss, y=M3[[s]]$x, lags = 1)
ss[[length(ss)+1]] <- tss                                                                  #6

if(seasonality!=0)
{
 tss <- AddLocalLevel(list(), M3[[s]]$x)
 tss <- AddSeasonal(tss, y=M3[[s]]$x, nseasons= seasonality)
ss[[length(ss)+1]] <- tss                                                                  #7
 tss <- AddGeneralizedLocalLinearTrend(list(), M3[[s]]$x)
 tss <- AddSeasonal(tss, y=M3[[s]]$x, nseasons= seasonality)
ss[[length(ss)+1]] <- tss                                                                  #8
 tss <- AddAr(tss, y=M3[[s]]$x, lags = 1)
 tss <- AddSeasonal(tss, y=M3[[s]]$x, nseasons= seasonality)
ss[[length(ss)+1]] <- tss                                                                  #9
 tss <- AddGeneralizedLocalLinearTrend(list(), M3[[s]]$x)
 tss <- AddAr(tss, y=M3[[s]]$x, lags = 1)
 tss <- AddSeasonal(tss, y=M3[[s]]$x, nseasons= seasonality)
ss[[length(ss)+1]] <- tss                                                                  #10
}
burn <- 100
model <- list()
cumulative.errors <- NULL
for (m in 1:length(ss))
{
 print(paste("Iteration for Time Series=", s, " State Spacification:", m))
 model[[m]] <- bsts(M3[[s]]$x, state.specification = ss[[m]], niter = 1000)
 prediction.errors<- bsts.prediction.errors(model[[m]])[-(1:burn),  ]
 cumulative.errors[m] <- sum(abs(colMeans(prediction.errors)))
}
best.model[s] <- which(cumulative.errors ==min(cumulative.errors))
pred <- predict(model[[best.model[s]]], horizon = M3[[s]]$h, burn = 100)$median

for (j in 1:18)
{
 M3Forecast_bsts$BSTS[[j]][s] <- c(pred,rep(NA, 18-length(pred)))[j]
}

}

M3Forecast_bsts$BSTS <- as.data.frame(M3Forecast_bsts$BSTS)




#Compute errors
{
errors <- lapply(M3Forecast_bsts, function(f) {

      res <- NULL

      for(x in 1:length(M3)) {

        curr_f <- unlist(f[x,])
        if(any(!is.na(curr_f)))
        {
           smape <- mean(200*abs(M3[[x]]$xx-
 curr_f[!is.na(curr_f)])/(curr_f[!is.na(curr_f)]+M3[[x]]$xx))
           mase <- mean(abs(M3[[x]]$xx-curr_f[!is.na(curr_f)])/ mean(abs(M3[[x]]$x[-1]-M3[[x]]$x[-
 length(M3[[x]]$x)])))
           curr_res <- c(smape, mase)
            }
        res <- rbind(res, curr_res)

      }

      rownames(res) <- NULL
      colnames(res) <- c("sMAPE", "MASE")
      res
    })
}
```

```
ind_yearly <- which(unlist(lapply(M3, function(x) {x$period == "YEARLY"})))
ind_quarterly <- which(unlist(lapply(M3, function(x) {x$period == "QUARTERLY"})))
ind_monthly <- which(unlist(lapply(M3, function(x) {x$period == "MONTHLY"})))
ind_other <- which(unlist(lapply(M3, function(x) {x$period == "OTHER"})))

yearly_errors <- t(as.data.frame(lapply(errors, function(x) {colMeans(x[ind_yearly,])})))
quarterly_errors <- t(as.data.frame(lapply(errors, function(x) {colMeans(x[ind_quarterly,])})))
monthly_errors <- t(as.data.frame(lapply(errors, function(x) {colMeans(x[ind_monthly,])})))
other_errors <- t(as.data.frame(lapply(errors, function(x) {colMeans(x[ind_other,])})))
all_errors <- t(as.data.frame(lapply(errors, function(x) {colMeans(x[,])})))

plot(as.factor(best.model),xlab="Model", ylab="Frequency")
summary(as.factor(best.model))
summary(as.factor(best.model[c(ind_quarterly, ind_monthly)]))


round(yearly_errors,2)
round(quarterly_errors,2)
round(monthly_errors,2)
round(other_errors,2)
round(all_errors, 2)
```

Curriculum Vitae

PERSONAL INFORMATION

Matthias Schmidl

Born: 2.2.1988 in Vienna

Citizenship: Austria

Contact: matthias.schmidl@gmail.com

EDUCATION

10/2013 – Present

Magister Degree Programme Statistics (Master equiv.)

University of Vienna

03/2011 – 10/2014

Magister Degree Programme Economics (Master equiv.)

University of Vienna

Thesis: *“Nowcasting mit Hilfe von Internet-Suchdaten” (Nowcasting using internet search query data)*, Advisor: Robert Kunst, University of Vienna/Institute for Advanced Studies (IHS)

10/2008 – 03/2011

Baccalaureate Degree Programme Economics (Bachelor equiv.)

University of Vienna

Thesis: “Die österreichische Grundsteuer im internationalen Vergleich: Verteilungswirkungen und Reformoptionen”, Advisor: Margit Schratzenstaller-Altzinger, University of Vienna/Austrian Institute of Economic Research (WIFO)

Thesis: “Das niederländische Gesundheitssystem: Struktur und Reform”, Advisor: Jörg Mahlich, University of Vienna/Austrian Economic Chamber (WKO)

2002 – 2007

Higher Technical Institute for Computing (HTL Wien Donaustadt, Abteilung für Informatik)

Vienna, passed with merit

WORK EXPERIENCE

11/2012 – Present

Research Assistant

Institute for Industrial Research (IWI), Vienna

07/2010 – 09/2010

Reasearch Assistant

Austrian Institute of Economic Research (WIFO); granted a Junior Fellowship

SKILLS

Familiar with statistical software: R, MatLab, SPSS-Statistics, SPSS-Modeler, Stata, EViews

Knowledge of programming languages: Visual Basic, C, C++, C#

(Scientific) PUBLICATIONS

Schmidl, M., & Schratzenstaller, M., (2011): Steuern auf Vermogen und Vermogensertrage: Probleme und Gestaltungsmöglichkeiten fur Österreich, in: Wirtschaft und Gesellschaft, Nr. 3/2011.

Dorfmayr, R., Lengauer, S. D., & Schmidl, M., (2013) Struktur und Entwicklung der Industrie Osterreichs, in: Industriebuch 2013, Industriewissenschaftliches Institut (IWI), ISBN 978-3-901978-14- 2.

Dorfmayr, R., Lengauer, S. D., Lind, T., Luptácik, P., Ramharter, C., Schmidl, M., & Willim, A., (2013) Struktur und Entwicklung der Maschinen & Metallwaren Industrie Niederösterreichs, Industriewissenschaftliches Institut (IWI), Wien.

Brunner, P., & Schmidl, M., (2014) Internationalisierung der Automotiven Unternehmen Österreichs, Industriewissenschaftliches Institut (IWI), Wien.

Brunner, P., Dorfmayr, R., Lengauer, S. D., Schmidl, M., & Schneider, H., (2014) Internationaler Wettbewerb der Wirtschaftsstandorte in der Automotiven Zulieferindustrie Kurzstudie, Industriewissenschaftliches Institut (IWI), Wien.

Brunner, P., Dorfmayr, R., Schmidl, M., & Schneider, H., (2014) Die Industriestandorte Niederösterreich und Oberösterreich im Vergleich, Industriewissenschaftliches Institut (IWI), Wien.

Dorfmayr, R., Luptácik, P., & Schmidl, M., (2014) Volkswirtschaftliche Effekte der PALFINGER Gruppe in Österreich, Industriewissenschaftliches Institut (IWI), Wien.