Predicting the present with Bayesian structural time series

Steven L. Scott* and Hal R. Varian

Google, Inc.,
1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA
E-mail: stevescott@google.com
E-mail: hal@google.com
*Corresponding author

Abstract: This article describes a system for short term forecasting based on an ensemble prediction that averages over different combinations of predictors. The system combines a structural time series model for the target series with a regression component capturing the contributions of contemporaneous search query data. A spike-and-slab prior on the regression coefficients induces sparsity, dramatically reducing the size of the regression problem. Our system averages over potential contributions from a very large set of models and gives easily digested reports of which coefficients are likely to be important. We illustrate with applications to initial claims for unemployment benefits and to retail sales. Although our exposition focuses on using search engine data to forecast economic time series, the underlying statistical methods can be applied to more general short term forecasting with large numbers of contemporaneous predictors.

Keywords: Bayesian model averaging; Bayesian structural time series models; Markov chain Monte Carlo; economic time series; machine learning; predicting the present; spike and slab priors; state space models

Reference to this paper should be made as follows: Scott, S.L. and Varian, H.R. (2014) 'Predicting the present with Bayesian structural time series', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 5, Nos. 1/2, pp.4–23.

Biographical notes: Steven L. Scott is a Senior Economic Analyst at Google where he has worked since 2008, specialising in Bayesian computation. Prior to Google, he was a Director of Statistical Analysis at Capital One, and an Assistant Professor of Statistics at the University of Southern California's Marshall School of Business. He received his PhD and AM from the Harvard Statistics Department, and BS in Mathematics and Economics from Texas Christian University.

Hal R. Varian is the Chief Economist at Google. He started in May 2002 as a consultant and has been involved in many aspects of the company, including auction design, econometric, finance, corporate strategy and public policy. He is also an Emeritus Professor at the University of California, Berkeley in three departments: business, economics, and information management. He received his BS from MIT in 1969 and his MA and PhD from UC Berkeley in

1973. He has published numerous papers in economic theory, econometrics, industrial organisation, public finance, and the economics of information technology.

1 Introduction

For reasons of cost, effort, or convention, many economic time series are reported infrequently (e.g., on a monthly or quarterly basis) despite being theoretically observable on finer time scales. Economic time series are also frequently revised after they are first reported, as new information becomes available. Modern econometric methods make it is possible to maintain a current estimate of the value of an economic time series before it is officially reported or revised. This problem is sometimes called 'nowcasting' because the goal is to forecast a current value instead of a future value (Banbura et al., 2011). An effective nowcasting model will consider both the past behaviour of the series being modelled, as well as the values of more easily observed contemporaneous signals. This article focuses on economic time series, but the nowcasting problem is present in other fields as well. Google Flu Trends (Google.org, http://www.google.org/flutrends/about/how.html) is an early example. See Shaman and Karspeck (2012) and Dukic et al. (2012) for others.

Choi and Varian (2009, 2012) demonstrated that Google search data could be used as an effective external signal for a nowcasting model, but their methods required carefully selecting the set of predictors to be used. This article describes a more robust and automatic system for selecting the predictors in a nowcasting model. Our system uses a structural time series model (Harvey, 1989) to capture the trend, seasonal, and similar components of the target series. A regression component in the structural model incorporates contributions from contemporaneous explanatory factors. Because the number of potential predictors in the regression model is large (often larger than the number of observations available to fit the model), we induce sparsity by placing a spike-and-slab prior distribution on the regression coefficients. This leads to a posterior distribution with positive mass at zero for sets of regression coefficients, so that simulated values from the posterior distribution have many zeros. We use a Markov chain Monte Carlo (MCMC) sampling algorithm to simulate from the posterior distribution, which smooths our predictions over a large number of potential models using Bayesian model averaging (Hoeting et al., 1999; Madigan and Raftery, 1994). As a by-product of the Bayesian analysis, our method also gives compelling reports indicating the marginal posterior inclusion probability for each predictor, and a graphical breakdown of how the model has apportioned time series variation between different components of state.

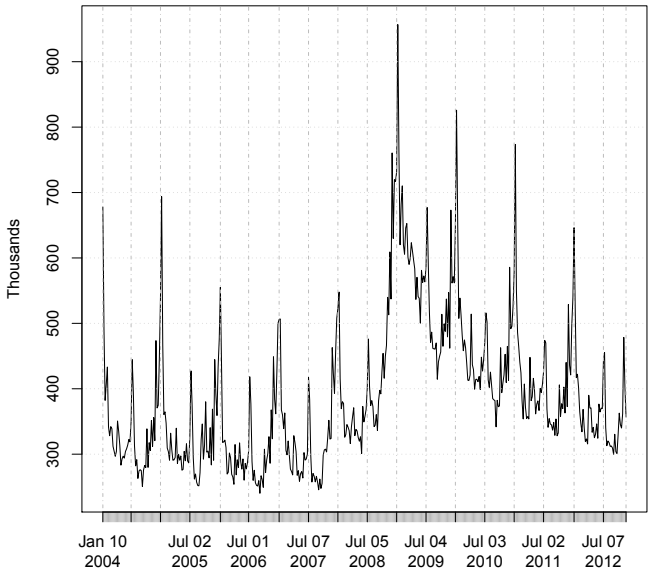
Our methods differ from forecasting techniques that handle large numbers of predictors by constructing latent factors. We do not directly model the distribution of our regressors, as would be the case in a dynamic factor model (e.g., Forni and Reichlin, 1998; Forni et al., 2000). An alternate approach is to construct static factors through methods like principal components (e.g., Stock and Watson, 2002a, 2002b). One could argue that we implicitly use static factors by considering the query verticals produced by Google Trends (see Section 2) instead of raw queries. This reduces many billions of

raw queries to several hundred query verticals, but we do not attempt to further reduce the dimension by constructing factors out of verticals.

Figure 1 shows an example of the type of data we might want to model in a nowcast. It plots weekly initial claims for unemployment in the USA. The initial claims data has a pronounced seasonal pattern, and clear bulge during the 2008–2010 financial crisis. The hope is that by using Google search queries as a proxy for public interest in unemployment related matters we can gain a clearer view of the current state of the series than we can by relying on the past behaviour of the series alone.

The remainder of this article is structured as follows. Section 2 describes Google search data. Section 3 describes the structural time series models used to capture the trend and seasonal components of the prediction, as well as the regression component of the model, where Bayesian model averaging is used to account for the uncertainty in which predictors are to be included. Section 4 describes the MCMC algorithm used to produce simulations from the Bayesian posterior distribution of both the structural time series and the regression components of the model. Section 5 illustrates how the method works on two time series chosen from the FRED database. Section 6 offers some concluding remarks.

Figure 1 Weekly (non-seasonally adjusted) initial claims for US unemployment benefits



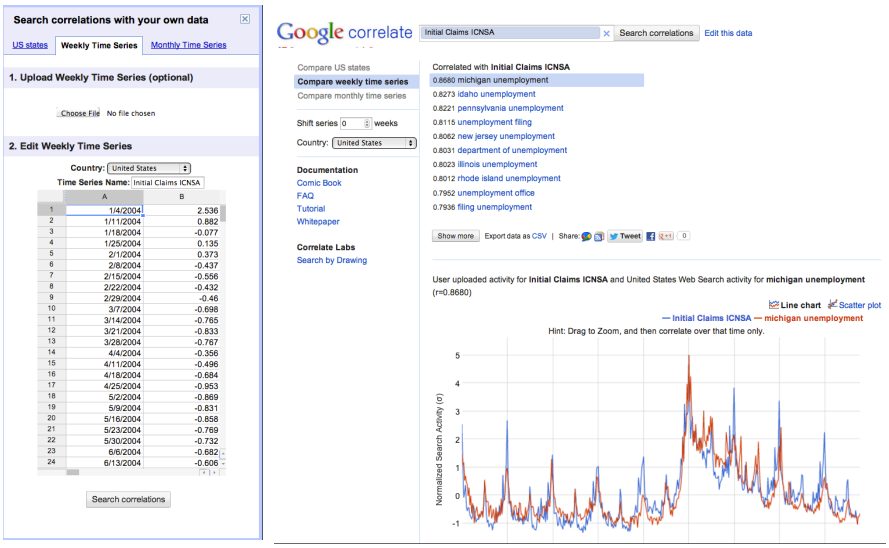
2 Google search data

Google search query data is interesting because it offers a quasi-real time view into the topics of greatest interest to the public. Google offers two public tools for obtaining search query data, Google Trends (http://google.com/trendsgoogle.com/trends) and Google Correlate (http://www.google.com/trends/correlate google.com/trends/correlate). The input into Google Trends is a specific search query (e.g., 'file for unemployment'). The output is a time series of the relative popularity of that search query over time. The time series is normalised by the overall search volume, so that it represents the

fraction of overall search traffic for that query, and it is scaled so that the maximum of the time series is 100. The series can be restricted to a particular time range (with the earliest date being January 2004) or geographic area, such as a country, or a state within the USA. The context of the search query can also be restricted to lie within a set of pre-defined 'verticals', such as *entertainment*, *science*, *automotive*, etc. The verticals have a hierarchical structure that can represent different levels of granularity. For example the *automotive* vertical has a *hybrid and alternative vehicles* subvertical. Across all levels there are about 600 search verticals, which are normalised and scaled in the same manner as individual search queries.

Google Correlate takes as input a time series, which can be the name of individual search query, or a time series uploaded by the user. It then finds the set of individual search queries that are most highly correlated with the input. Google Correlate will provide up to 100 correlates. These can be restricted by geography, but not by vertical. A screen shot of the tool is shown in Figure 2.

Figure 2 The interface to Google correlate, showing the top ten correlates to the data in Figure 1, and a plot of the initial claims series against the query for 'Michigan unemployment' (see online version for colours)



The combination of Google Trends and Google Correlate means that as many as 700 predictors are available for any time series one might wish to nowcast. The 600 verticals from Google Trends will be the same regardless of the target series being modelled. The 100 individual queries from Google Correlate will be specific to the time series being modelled.

3 The model

Our nowcasting model has two components. A time series component captures the general trend and seasonal patterns in the data. A regression component captures the

impact of the Google search query data. Because both components are additive, we can easily estimate the joint model using Bayesian methods. Section 3.1 explains the time series component of our model. Section 3.2 discusses the regression component.

3.1 Structural time series

Let y_t denote observation t in a real-valued time series. A structural time series model can be described by a pair of equations relating y_t to a vector of latent state variables α_t.

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

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

Equation (1) is called the *observation* equation, because it links the observed data y_t with the unobserved latent state α_t. Equation (2) is called the *transition* equation because it defines how the latent state evolves over time. The model matrices Z_t, T_t, and R_t typically contain a mix of known values (often 0 and 1), and unknown parameters. The transition matrix T_t is square, but R_t can be rectangular if a portion of the state transition is deterministic. Having R_t in equation (2) allows the modeler to work with a full rank variance matrix Q_t because any linear dependencies in the state vector can be moved from Q_t to R_t. In our application H_t is a positive scalar. A model that can be described by equations (1) and (2) is said to be in *state space form*. A very large class of models can be expressed in state space form, including all ARIMA and VARMA models.

State space models are attractive in part because they are modular. Independent state components can be combined by concatenating their observation vectors Z_t, and arranging the other model matrices as elements in a block diagonal matrix. This provides the modeler with the considerable flexibility to choose components for modelling trend, seasonality, regression effects, and potentially other state components that may be necessary. For example, one useful model can be obtained by adding a regression component to the popular 'basic structural model'. This model can be written

$$\begin{aligned} y_t &= \mu_t + \tau_t + \beta^T \mathbf{x}_t + \epsilon_t \\ \mu_t &= \mu_{t-1} + \delta_{t-1} + u_t \\ \delta_t &= \delta_{t-1} + v_t, \\ \tau_t &= -\textstyle\sum_{s=1}^{S-1} \tau_{t-s} + w_t \end{aligned} \tag{3}$$

where $\eta_t = (u_t, v_t, w_t)$ contains independent components of Gaussian random noise. Though the model matrices in a structural time series model are permitted to depend on t, in this case Q_t is a constant diagonal matrix with diagonal elements σ_u^2, σ_v^2, and σ_w^2, and H_t is a constant σ_ϵ^2. This model contains trend, seasonal, and regression components. The current level of the trend is μ_t, the current 'slope' of the trend is δ_t. The seasonal component τ_t 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. We will provide more details on the regression component in Section 3.2, but for the remainder of this Section one may consider β to be known. The parameters in equation (3) are the variances $\sigma_\epsilon^2, \sigma_u^2, \sigma_v^2, \sigma_w^2$ and the regression coefficients β. In our context, the vector $\mathbf{x}_t$ is a contemporaneous set of search queries or trends verticals,

including any desired lags or other transformations. Of course $\mathbf{x}_t$ can be extended to include other factors as well.

The primary tools for working with state space models are the Kalman filter (Kalman, 1960; Harvey 1989), the Kalman smoother, and Bayesian data augmentation. Denote $y_{1:t} = y_1, \ldots, y_t$. The Kalman filter recursively computes the predictive distribution $p(\alpha_{t+1}|y_{1:t})$ by combining $p(\alpha_t|y_{1:t-1})$ with y_t using a standard set of formulas that is logically equivalent to linear regression. The Kalman smoother updates the output of the Kalman filter to produce $p(\alpha_t|y_{1:n})$, where n is the length of the time series, at each value of t. Because all components of the model are Gaussian, both $p(\alpha_{t+1}|y_{1:t})$ and $p(\alpha_t|y_{1:n})$ are multivariate normal distributions parameterised by their mean $\boldsymbol{\mu}_t$ and variance $\mathbf{P}_t$. The Kalman filtering accumulates information about the time series as it moves forward through the list of $(\boldsymbol{\mu}_t, \mathbf{P}_t)$ elements. The Kalman smoother moves backward through time, distributing information about later observations to successively earlier $(\boldsymbol{\mu}_t, \mathbf{P}_t)$ pairs. The forward-backward pattern is idiomatic for general message passing algorithms (Cowell et al., 1999), of which the Kalman filter and smoother are special cases. Readers interested in the computational details of the Kalman filter and smoother can consult Durbin and Koopman (2001) or numerous other sources.

Filtering and smoothing are the traditional computational operations associated with state space models. A third, related task often arises in Bayesian computation, in which it is often desirable to simulate the state from its posterior distribution given the data. Let $\mathbf{y} = y_{1:n}$ and $\boldsymbol{\alpha} = \alpha_{1:n}$ denote the full sets of observed and latent data. Bayesian data augmentation methods produce simulations from $p(\boldsymbol{\alpha}|\mathbf{y})$. One cannot simply draw each α_t from $p(\alpha_t|\mathbf{y})$, because the serial correlation between α_t and α_{t+1} must be respected. Instead, there are stochastic versions of the Kalman smoother that can be used to sample directly from $p(\boldsymbol{\alpha}|\mathbf{y})$. Important early algorithms from Carter and Kohn (1994) and Frühwirth-Schnatter (1995) were improved by de Jong and Shepard (1995) and Durbin and Koopman (2002). Durbin and Koopman developed a way to simulate random noise with the same covariance as $p(\boldsymbol{\alpha}|\mathbf{y})$, to which they add the appropriate mean, which can be computed using a fast 'state mean smoother'. Rue (2001), McCausland et al. (2011) and Chan and Jeliazkov (2009) suggest that even further efficiency gains can be had by avoiding the Kalman filter altogether and treating the whole sampling algorithm as a sparse multivariate normal calculation to be handled by sparse matrix routines. We use the Durbin and Koopman method in this paper.

3.2 Spike and slab regression

Equation (3) contains a regression component that allows a set of external factors to contribute to the prediction. There are several ways of organising the model matrices to add a regression component to a state space model. A convenient method is to append a constant 1 to each α_t, and append $\beta^T\mathbf{x}_t$ to Z_t in the observation equation. The advantage of this specification is that it only increases the dimension of the state vector by one, regardless of the number of predictors. The computational complexity of the Kalman filter is linear in the length of the data and quadratic in the size of the state, so it is important to avoid artificially inflating the state size. If there is a small set of regression coefficients that are believed to change over time, those coefficients can be added to model as additional state variables. Otherwise, the rest of this Section focuses on regression coefficients that are constant through time.

3.2.1 Prior specification and elicitation

Because there are so many potential Google search queries, and economic time series are typically short, the set of predictors can be much larger than the set of observations. However we expect a high degree of sparsity, in the sense that the coefficients for the vast majority of predictors will be zero. The natural way to represent sparsity in the Bayesian paradigm is through a spike and slab prior on the regression coefficients. Let $\gamma_k = 1$ if $\beta_k \neq 0$, and $\gamma_k = 0$ if $\beta_k = 0$. Let β_γ denote the subset of elements of β where $\beta_k \neq 0$. A spike-and-slab prior may be written

$$p(\beta, \gamma, \sigma_\epsilon^2) = p(\beta_\gamma|\gamma, \sigma_\epsilon^2)p(\sigma_\epsilon^2|\gamma)p(\gamma). \tag{4}$$

The marginal distribution $p(\gamma)$ is the 'spike', so named because it places positive probability mass at zero. In principle, $p(\gamma)$ can be specified to control for best practices like the hierarchical principle (lower order terms must be present if higher order interactions are included). In practice it is convenient to simply use an independent Bernoulli prior

$$\gamma \sim \prod_{k=1}^{K} \pi_k^{\gamma_k}(1-\pi_k)^{1-\gamma_k}. \tag{5}$$

Equation (5) is often further simplified by assuming all the π_k are the same value π. This is a practical consideration, because setting a different value for each π_k can be a burden, but it can be justified on the basis of prior exchangeability. A natural way to elicit π is to ask the analyst for an 'expected model size', so that if one expects p non-zero predictors then $\pi = p/K$, where K is the dimension of $\mathbf{x}_t$. In certain instances it can also be useful to set $\pi_k = 0$ or 1, for specific values of k, to force certain variables to be excluded or included. Another strategy that one could pursue (but we have not) is to subjectively segment predictors into groups based on how likely they would be to enter the model. One could then assign all the elements of each group the same subjectively determined prior inclusion probability.

For a symmetric matrix Ω^{-1}, let Ω_γ^{-1} denote the rows and columns of Ω^{-1} corresponding to $\gamma_k = 1$. Then the conditional priors $p(1/\sigma_\epsilon^2|\gamma)$ and $p(\beta_\gamma|\sigma_\epsilon, \gamma)$ can be expressed as the conditionally conjugate pair

$$\beta_\gamma|\sigma_\epsilon^2, \gamma \sim \mathcal{N}\left(b_\gamma, \sigma_\epsilon^2\left(\Omega_\gamma^{-1}\right)^{-1}\right) \qquad \frac{1}{\sigma_\epsilon^2}|\gamma \sim Ga\left(\frac{\nu}{2}, \frac{ss}{2}\right), \tag{6}$$

where $Ga(r, s)$ denotes the gamma distribution with mean r/s and variance r/s^2. Equation (6) is the 'slab' because one can choose the prior parameters to make it only very weakly informative (close to flat), conditional on γ. As with equation (5) there are reasonable default values that can be used to simplify equation (6). It is very common to assume the vector of prior means b is zero. However it is easy to specify an informative prior if one believes that certain predictors will be particularly helpful. The values ss and ν (interpretable as a prior sum of squares, and prior sample size) can be set by asking the user for an expected R^2 from the regression, and a number of observations worth of weight (ν) to be given to their guess. Then $ss/\nu = (1 - R^2)s_y^2$, where s_y^2 is the marginal standard deviation of the response. Scaling by s_y^2 is a minor violation of

the Bayesian paradigm because it means our prior is data-determined. However, it is a useful device, and we have identified no practical negative consequences from using it.

The highest dimensional parameter in equation (6) is the full-model prior information matrix Ω^{-1}. Let $\mathbf{X}$ denote the design matrix, obtained in the standard way by stacking the predictors so that $\mathbf{x}_t$ is row t. The likelihood for an ordinary regression model has information matrix $\mathbf{X}^T\mathbf{X}/\sigma_\epsilon^2$, so taking $\Omega^{-1} = \kappa\mathbf{X}^T\mathbf{X}/n$ would place κ observations worth of weight on the prior mean b. This is known as Zellner's $g-$prior (Zellner, 1986; Chipman et al., 2001; Liang et al., 2008). Some care must be exercised in practice to guard against perfect collinearity in the columns of $\mathbf{X}$. Full rank can be guaranteed by setting $\Omega^{-1} = \kappa(w\mathbf{X}^T\mathbf{X} + (1-w)\text{diag}(\mathbf{X}^T\mathbf{X}))/n$, where $\text{diag}(\mathbf{X}^T\mathbf{X})$ is the diagonal matrix with diagonal elements matching those of $\mathbf{X}^T\mathbf{X}$. By default we set $w = 1/2$ and $\kappa = 1$. To recap, the spike and slab prior developed here offers ample opportunity to express prior opinions through the prior parameters π_k, b, Ω^{-1}, ss, and ν. For the analyst who prefers simplicity at the cost of some reasonable assumptions, useful prior information can be reduced to an expected model size, an expected R^2, and a sample size ν determining the weight given to the guess at R^2. For the analyst who wishes avoid thinking about priors altogether our software supplies default values $R^2 = .5$, $\nu = .01$ and $\pi_k = .5$.

3.2.2 The conditional posterior of β and σ_ϵ^2 given γ

Let $y_t^* = y_t - Z_t^{*T}\alpha_t$, where Z_t^* is the observation matrix from equation (1) with $\beta^T\mathbf{x}_t$ set to zero. Let $\mathbf{y}^* = y_{1:n}^*$, so that $\mathbf{y}^*$ is $\mathbf{y}$ with the time series component subtracted out. Conditional on γ the joint posterior distribution for β and σ_ϵ^2 is available from standard conjugacy formulas (e.g., Gelman et al., 2002)

$$\beta_\gamma|\sigma_\epsilon,\gamma,\mathbf{y}^* \sim \mathcal{N}\left(\tilde{\beta}_\gamma, \sigma_\epsilon^2(V_\gamma^{-1})^{-1}\right) \qquad 1/\sigma_\epsilon^2|\gamma,\mathbf{y}^* \sim Ga\left(\frac{N}{2}, \frac{SS_\gamma}{2}\right) \tag{7}$$

where the sufficient statistics can be written

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

The expression for SS_γ can be written in several ways. The preceding form is computationally convenient. Other equivalent representations emphasise its interpretation as a sum of squared residuals that is obviously positive.

3.2.3 The marginal posterior of γ

Because of conjugacy, one can analytically marginalise over β_γ and $1/\sigma_\epsilon^2$ to obtain

$$\gamma|\mathbf{y}^* \sim C(\mathbf{y}^*)\frac{|\Omega_\gamma^{-1}|^{\frac{1}{2}}}{|V_\gamma^{-1}|^{\frac{1}{2}}}\frac{p(\gamma)}{SS_\gamma^{\frac{N}{2}-1}}, \tag{8}$$

where $C(\mathbf{y}^*)$ is a normalising constant that depends on $\mathbf{y}^*$ but not on γ. The MCMC algorithms used to fit the model do not require $C(\mathbf{y}^*)$ to be computed explicitly.

Equation (8) is inexpensive to evaluate, because the only matrix that needs to be inverted is V_γ^{-1}, which is of low dimension if the models being investigated are sparse. Also notice that, unlike L_1-based methods for sparse modelling (e.g., the lasso), equation (8) places positive probability on coefficients being zero (as opposed to probability density). Thus the sparsity in this model is a feature of the full posterior distribution, and not simply the value at the mode.

4 Markov chain Monte Carlo

4.1 Parameter learning

Let θ denote the set of model parameters other than β and σ_ϵ^2. The posterior distribution of the model described in Section 3 can be simulated by a straightforward MCMC algorithm with the following steps.

1 simulate the latent state $\boldsymbol{\alpha}$ from $p(\boldsymbol{\alpha}|\mathbf{y}, \theta, \beta, \sigma_\epsilon^2)$ using the simulation smoother from Durbin and Koopman (2002)

2 simulate $\theta \sim p(\theta|\mathbf{y}, \boldsymbol{\alpha}, \beta, \sigma_\epsilon^2)$

3 simulate β and σ_ϵ^2 from a Markov chain with stationary distribution $p(\beta, \sigma_\epsilon^2|\mathbf{y}, \boldsymbol{\alpha}, \theta)$.

Let $\phi = (\theta, \beta, \sigma_\epsilon^2, \boldsymbol{\alpha})$. Repeatedly cycling through the three steps given above yields a sequence of draws $\phi^{(1)}, \phi^{(2)}, \ldots$ from a Markov chain with stationary distribution $p(\phi|\mathbf{y})$, the posterior distribution of ϕ given $\mathbf{y}$. The draw of θ in step 2 depends on which state components are present in the model, but it is often trivial. For example, the model in equation (3) has three variance parameters, all of which have conditionally independent inverse gamma full conditional distributions (assuming independent inverse gamma priors).

The draw in step 3 can be done using the stochastic search variable selection (SSVS) algorithm from George and McCulloch (1997). SSVS is a Gibbs sampling algorithm in which each element of γ is drawn from its full conditional distribution, proportional to equation (8). The full conditional distributions are straightforward because each γ_k has only two possible values. After one sweep through all the variables (which we do in random order), β_γ and σ_ϵ^2 are drawn from their closed form full conditional distribution given in equation (7). Improvements to SSVS have been proposed by several authors, notably Ghosh and Clyde (2011). However, the basic SSVS algorithm has performed adequately for our needs.

4.2 Forecasting

As is typical in Bayesian data analysis, forecasts from our model are based on the posterior predictive distribution. It is trivial to simulate from the posterior predictive distribution given draws of model parameters and state from their posterior distribution. Let $\tilde{y}$ denote the set of values to be forecast. The posterior predictive distribution of $\tilde{y}$ is

$$p(\tilde{y}|\mathbf{y}) = \int p(\tilde{y}|\phi)p(\phi|\mathbf{y})\ d\phi. \tag{9}$$

If $\phi^{(1)}, \phi^{(2)}, \ldots$ are a set of random draws from $p(\phi|\mathbf{y})$, then one samples from $p(\tilde{y}|\mathbf{y})$ by sampling from $p(\tilde{y}^{(g)}|\phi^{(g)})$, which is done by simply iterating equations (1) and (2) forward from $\alpha_n^{(g)}$, with parameters $\theta^{(g)}$, $\beta^{(g)}$, and $\sigma_\epsilon^{2(g)}$. Because different elements of β will be zero in different Monte Carlo draws, the draws from the posterior predictive distribution automatically account for sparsity and model uncertainty, and thus benefit from Bayesian model averaging. This method of forecasting produces a sample of draws from the posterior predictive distribution $p(\tilde{y}|\mathbf{y})$. The draws can be summarised, for example by their mean (which is a Monte Carlo estimate of $E(\tilde{y}|\mathbf{y})$). Multivariate summaries, like sets of interesting quantiles, are also appropriate. Our preference is to avoid summarising the draws, instead reporting them using graphical methods such as histograms, kernel density estimates, or the dynamic density plots used in Section 5.

One can forecast an arbitrary number of periods ahead. The examples presented below focus on one-step-ahead forecasts because they are the most relevant to the nowcasting problem.

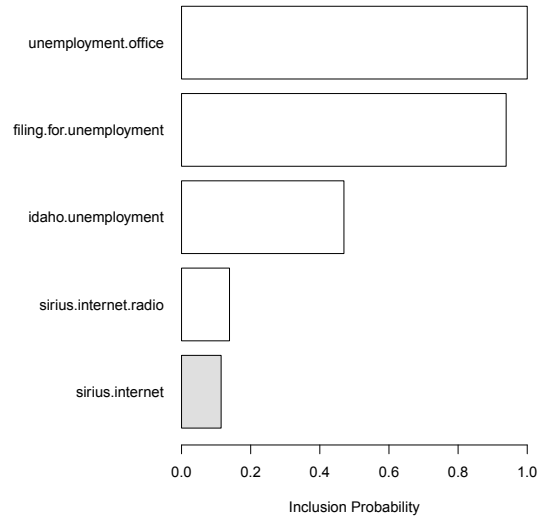
5 Examples

5.1 Weekly initial claims for unemployment

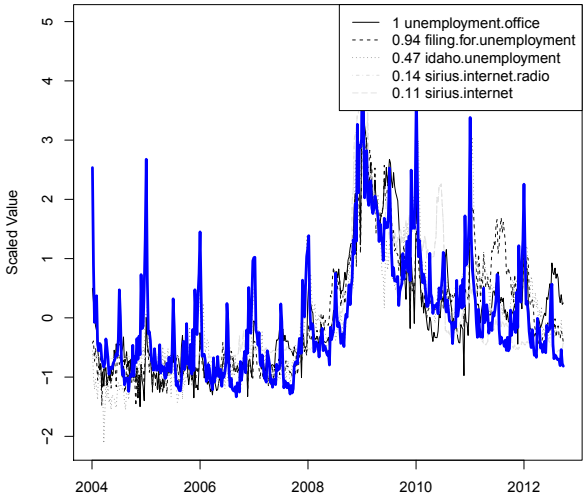
We used Google Correlate to produce the 100 search terms with values most correlated to the initial claims data from Figure 1. We then ran the MCMC algorithm for 5,000 iterations, and discarded the first 1,000 as burn in. The computation took about 105 seconds on a 3.4 GHz Linux workstation with ample memory. The software was coded in C++, with an R interface. We used the default priors in our software, except we set the 'expected model size' to 5, which corresponds to setting the prior inclusion probability to $5/100 = .05$. The time series component of our model matched equation (3), in that it included both a local linear trend component and an annual seasonal component with $S = 52$ weeks. Initial claims for unemployment benefits for the previous week are released on Thursdays, while Google Trends data is available with a two-day lag. This allows us to estimate the previous week's values four to five days before the data release. Initial claims data are subsequently revised, and we use only the final numbers in this analysis.

Figure 3 shows the predictors to which the MCMC algorithm assigned posterior inclusion probability greater than .1. The posterior inclusion probability is a margin of equation (8), which cannot be computed directly because the sum over model space is intractable. However, marginal inclusion probabilities can be trivially estimated from the Monte Carlo sample by computing the proportion of Monte Carlo draws with $\beta_k \neq 0$. The three predictors with the highest posterior inclusion probabilities were economically meaningful, and the difference in inclusion probabilities between the other predictors is substantial. Of the 100 top predictors from Google Correlate, 14 were queries for unemployment for a specific state. This model run settled on Idaho, but the other states are all highly correlated with one another (correlations range between .67 and .92). Repeated runs with different lengths and seeds confirmed that the model does prefer the information from Idaho to that from other states, all of which are selected for inclusion much less often (the second most predictive state was California, which only appeared in about 2% of the models). The sampling algorithm did generate sparse models, with 67% of models having at most three predictors, and 98% having at most 5.

Figure 3 (a) The posterior inclusion probabilities for all the variables with inclusion probability greater than 0.1 (b) A time series plot comparing the standardised initial claims series with the standardised search term series for the terms with inclusion probabilities of at least 0.1 (see online version for colours)



(a)

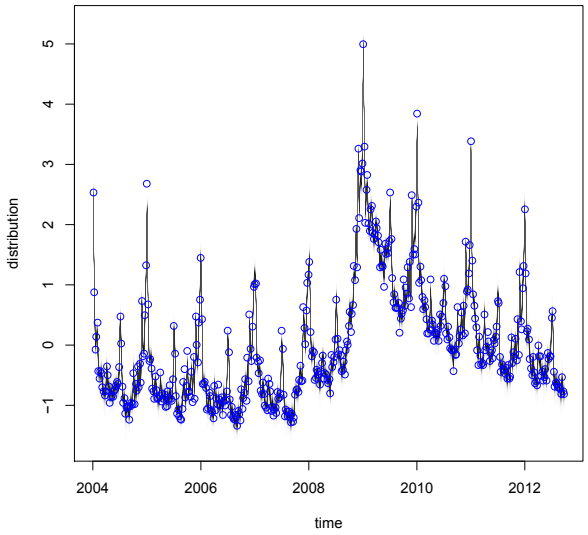


(b)

Notes: In Figure 3(a), bars are shaded on a continuous [0, 1] scale in proportion to the probability of a positive coefficient, so that negative coefficients are black, positive coefficients are white, and grey indicates indeterminate sign.
In Figure 3(b), the shading of the lines is weighted by the marginal inclusion probability.

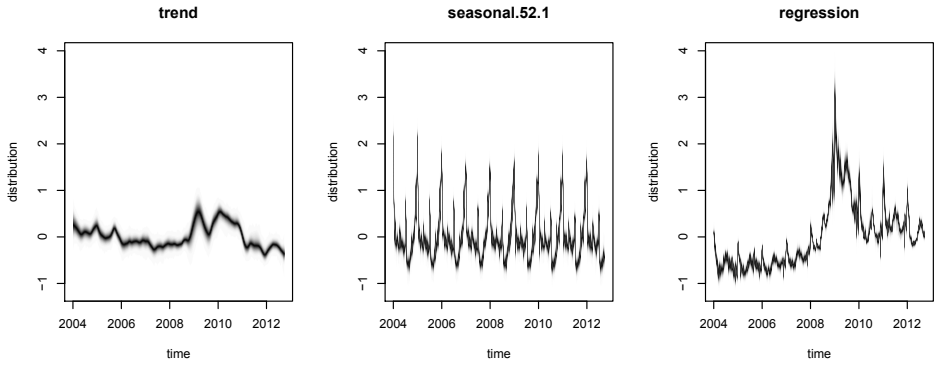
Figures 4 and 5 show the posterior distribution of the latent state at each time point. Figure 4 plots the combined state ($Z_t^T \alpha_t$), representing the smoothed value of the series in the absence of observation noise. Figure 5 shows the contribution of each component. Figures 4 and 5 represent the posterior distribution using a dynamic distribution plot. The pointwise posterior median is coloured black, and each 1% quantile away from the median is shaded slightly lighter, until the 99th and 1st percentiles are shaded white. Figure 5 is a visual representation of how much variation in initial clams is being explained by the trend, seasonal and regression components. The trend component shows a 'double dip' in initial claims. Both the seasonal and the regression components exhibit substantially more variation than the trend.

Figure 4 Posterior state for the model based on initial claims data (see online version for colours)



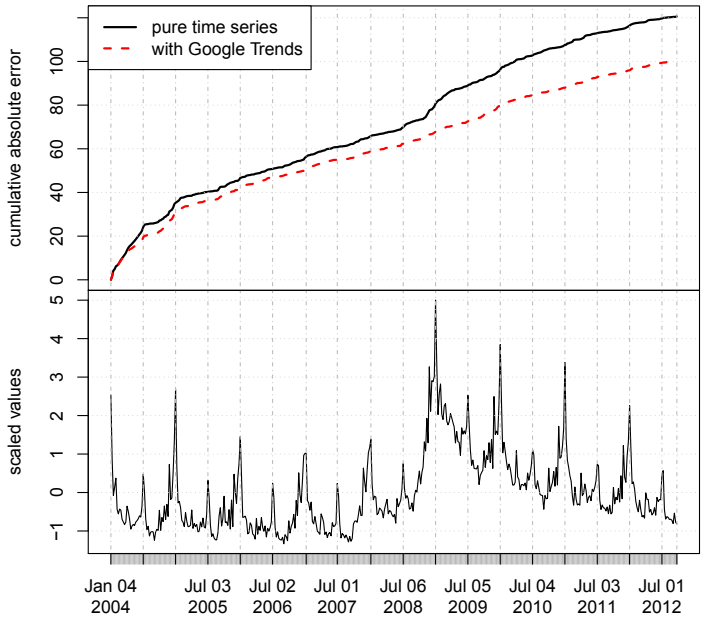
Note: The dots represent the raw observations.

Figure 5 Contributions to state for the initial claims data



To better understand the value of the Google Trends data, we fit a pure time series model equivalent to equation (3), but with no regression component. Figure 6 plots the cumulative sum of the absolute values of the one-step-ahead prediction errors for the two models. The most striking feature of the plot is the jump in cumulative absolute error that the pure time series model experiences during the financial crisis of 2008–2009. The model that uses Google Trends data generally accumulates errors at a slightly lower rate than the pure time series model, but the pure time series model experiences large errors near the start of the recession. The model based on the Google Trends data continues accumulating error at a roughly constant rate. This is noteworthy since it is widely recognised that the most challenging problem in forecasting economic time series is predicting the 'turning points'. The techniques described here may help with the turning-point problem.

Figure 6 The top panel shows cumulative absolute errors for equivalent time series models with and without Google Trends data (see online version for colours)



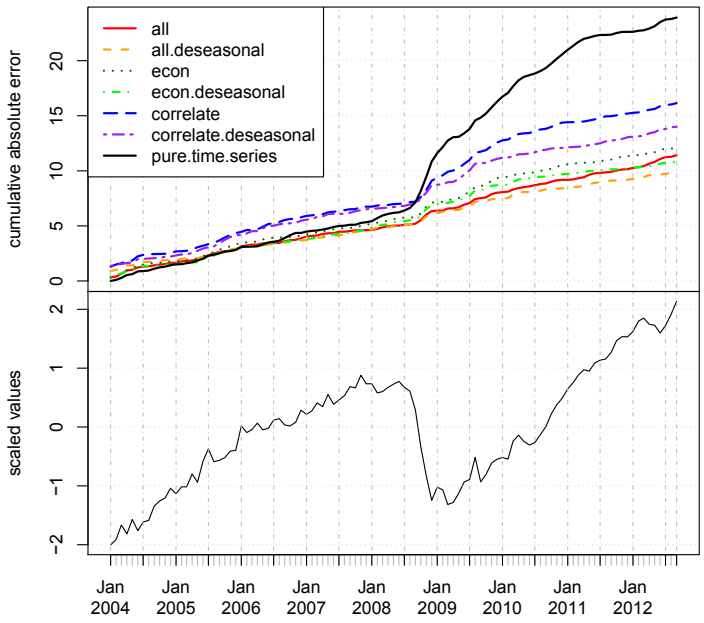
Note: The models are based on the initial claims data in the bottom panel.

5.2 Monthly retail sales

A second example illustrates that our system is not completely automatic, and that it can be helpful to select subsets or transformations of the data based on subject matter expertise. The bottom panel of Figure 7 shows monthly seasonally-adjusted US retails sales, excluding food services. Retail sales data for a particular month are released about two weeks after the month ends, while Google Trends reports data with a two-day lag. Hence we can nowcast retail sales about 12 days ahead of the data release. It would also be possible to fit a separate model conditional on data partway through the month

to give a longer forecasting lead. Like the initial claims data, and many other economic time series, retail sales figures are typically revised after their first release. We use only the final numbers in our analysis.

Figure 7 Cumulative absolute one-step-ahead forecast errors for the suite of models described in Section 5.2, based on the retail sales data in the bottom panel (see online version for colours)

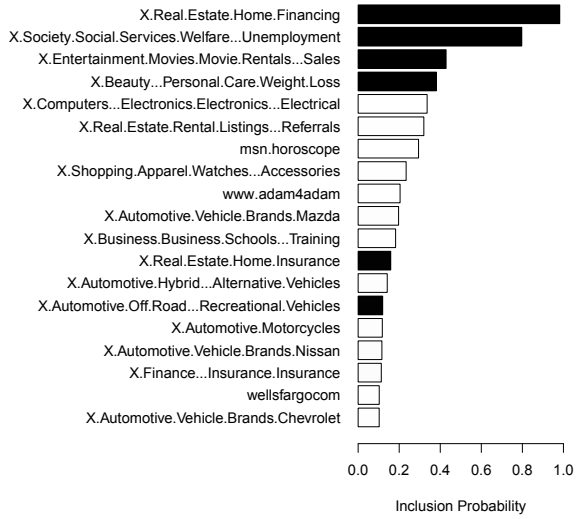


We paired the top 100 results from Google Correlate with the 600 Google Trends verticals. We ran the model on the full data, and with two subsets of predictors. The first subset was the 100 series from Google Correlate, without using any of the Trends data. The second subset paired the Correlate data with a subset of the Trends data that we subjectively determined to be of potential economic relevance. Because this series has been seasonally adjusted, it makes sense to seasonally adjust the predictors, which we did by passing each series through the STL procedure in R (Cleveland et al., 1990), and subtracting off the estimated seasonal component. For each subset we ran the model with and without deseasonalising the predictors, and in all cases we removed the seasonal component from model (3). We also ran a pure time series model (with no seasonal or regression component) as a baseline. As in Section 5.1, we used the default values of all priors, except we set the expected model size to 5.

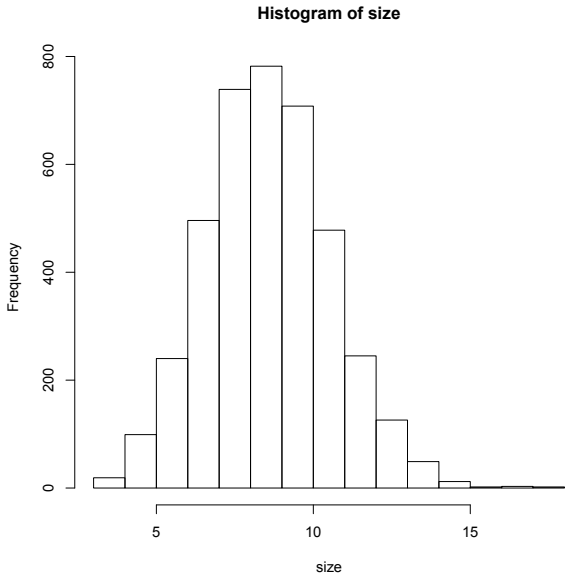
The top panel of Figure 7 plots the cumulative absolute one-step-ahead prediction errors for all seven models. It shows that the pure time series model has trouble adapting to the sudden change points exhibited by the retail sales time series, but the models based on the full data (and the economically relevant subset), accumulate error at roughly the same rate as before the financial crisis. Figure 7 also makes clear that the Google Trends verticals provide additional explanatory power over the variables produced by Google Correlate, but the subset of economically relevant Trends verticals

performs about as well as the entire dataset. In terms of overall fit, all three subsets performed slightly better with deseasonalised predictors.

Figure 8 (a) Posterior inclusion probabilities for the most likely predictors of the retail sales data (b) Posterior distribution of model size



(a)



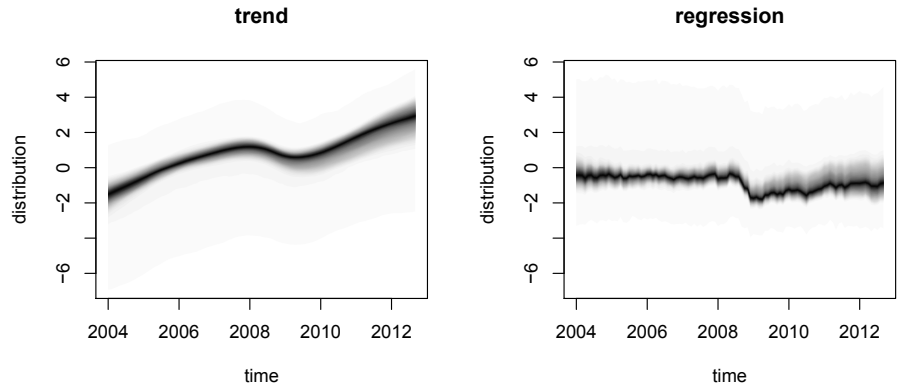
(b)

Notes: In Figure 8(a), the bars are shaded according to the probability the coefficient is positive, white bars correspond to positive coefficients, and black to negative coefficients, predictors starting with X are trends verticals.

The sets of predictors favoured by the algorithm also tended to be more reasonable using deseasonalised data. Figure 8(a) plots the posterior marginal inclusion probabilities for the top variables in the deseasonalised economic subset with posterior inclusion probabilities greater than .1. The bars are shaded according to the probability that the coefficient is positive, so that white bars correspond to positive coefficients, and black bars to negative coefficients. The top two predictors *real estate/home financing* and *social services/welfare/unemployment* are the only ones present in the model with greater than 50% probability, and both have a negative relationship to retail sales. Note that there are 24 sub-verticals relating to specific brands of automobiles, and the probability of including at least one automotive sub-vertical is around 61%, so the collection of automobile brands can be construed as a third frequently-observed factor. The relative infrequency of each individual automobile brand is because the MCMC algorithm is freely switching the brands that it is choosing. The frequent switching is a desirable hedge against choosing the wrong brand, and is a sign that the model averaging algorithm is performing well. Figure 8(b) plots the posterior distribution of the number of predictors in the model. Despite having 184 variables to choose from, the mean and median number of predictors (including the intercept) are both around 9, the 95th percentile is 12, and the largest model in the MCMC sample has 18 predictors.

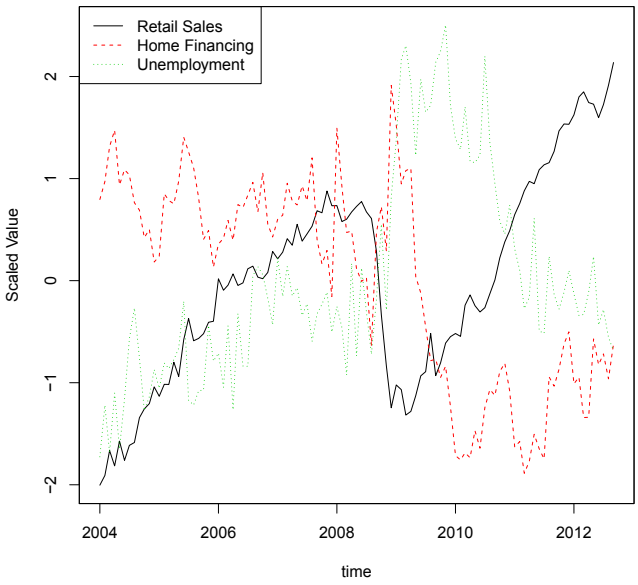
Figure 9 shows the posterior distribution of the trend and regression components of the model. Both components show an adjustment during the recession, but the regression component captures the sudden sharp change in the value of the series, which allows the local linear trend to remain smooth. The regression effect is largely flat prior to the recession, with a sudden change near the time of the financial crisis, followed by a gradual recovery.

Figure 9 Components of state for the retail sales data

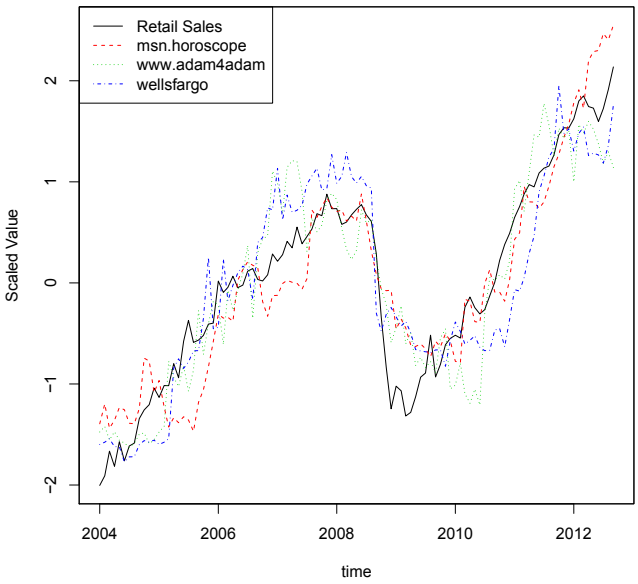


The decomposition in Figure 9 helps explain the set of predictors favoured by the model. Figure 10 compares the predictors with the largest marginal inclusion probabilities from Google Trends and Google Correlate. By construction, the predictors from Google Correlate closely track the target series, but much of the information they provide is redundant once we consider the underlying trend in retail sales. The predictors selected from Google Trends do not match the series as a whole nearly as well, but they do exhibit violent shocks near the same time points as the target series.

Figure 10 Retails sales and the predictors with the highest marginal inclusion probabilities from (a) the economically relevant subset of Google Trends verticals and (b) Google Correlate (see online version for colours)



(a)



(b)

6 Conclusions

We have presented a system useful for nowcasting economic time series based on Google Trends and Google Correlate data. The system combines structural time series models with Bayesian spike-and-slab regression to average over a subset of the available predictors. The model averaging that automatically comes with spike-and-slab priors and MCMC helps hedge against selecting the 'wrong' set of predictors. The spike-and-slab prior helps protect against spurious regressors, but no such protection can be perfect. For example, one of the strongest predictors in the full dataset for retail sales application was the *science/scientific equipment* vertical. The vertical exhibited a spike near the beginning of the financial crisis, attributable to a series of news stories about the large Hadron Collider that went online at that time, and its potential for creating black holes that could destroy the earth. This is a spurious predictor in that it would be unlikely to exhibit similar behaviour in periods of future economic distress. Some amount of subjective judgment is needed to remove such egregious anomalies from the training data, and restrict the candidate variables to those that have at least a plausible economic justification.

One open question that we plan to investigate further is whether and how the predictor variables should be adjusted, either prior to entering the candidate pool, or as part of the MCMC. It is common to 'whiten' predictors by removing a seasonal pattern, and possibly a trend, but which seasonal pattern and which trend (its own, or that of the response), is a matter for further investigation.

Acknowledgements

The authors thank the editor and two anonymous referees for helpful comments on an early draft of this article.

References

Banbura, M., Giannone, D. and Reichlin, L. (2011) 'Nowcasting', in M.P. Clements and D.F. Hendry (Eds.): *Oxford Handbook of Economic Forecasting*, Chap. 7, Oxford University Press, Oxford.

Carter, C.K. and Kohn, R. (1994) 'On Gibbs sampling for state space models', *Biometrika*, Vol. 81, No. 3, pp.541–553.

Chan, J. and Jeliazkov, I. (2009) 'Efficient simulation and integrated likelihood estimation in state space models', *International Journal of Mathematical Modelling and Numerical Optimisation*, Vol. 1, No. 1, pp.101–120.

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

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*, Vol. 88, pp.2–9.

Cleveland, R.B., Cleveland, W.S., McRae, J.E. and Terpenning, I. (1990) 'STL: a seasonal-trend decomposition procedure based on loess', *Journal of Official Statistics*, Vol. 6, pp.3–73.

Cowell, R.G., Dawid, A.P., Lauritzen, S.L. and Spiegelhalter, D.J. (1999) *Probabilistic Networks and Expert Systems*, Springer.

de Jong, P. and Shepard, N. (1995) 'The simulation smoother for time series models', *Biometrika*, Vol. 82, No. 2, pp.339–350.

Dukic, V., Polson, N.G. and Lopes, H. (2012) 'Tracking flu epidemics using Google Flu Trends data and a state-space SEIR model', *Journal of the American Statistical Association*, doi = 10.1080/01621459.2012.713867.

Durbin, J. and Koopman, S.J. (2001) *Time Series Analysis by State Space Methods*, Oxford University Press.

Durbin, J. and Koopman, S.J. (2002) 'A simple and efficient simulation smoother for state space time series analysis', *Biometrika*, Vol. 89, No. 3, pp.603–616.

Forni, M. and Reichlin, L. (1998) 'Let's get real: a factor analytical approach to disaggregated business cycle dynamics', *Review of Economic Studies*, Vol. 65, pp.453–473.

Forni, M.M., Hallin, M., Lippi, M. and Reichlin, L. (2000) 'The generalized dynamicfactor model: identification and estimation', *Review of Economics and Statistics*, Vol. 82, pp.540–554.

Frühwirth-Schnatter, S. (1995) 'Bayesian model discrimination and Bayes factors for linear Gaussian state space models', *Journal of the Royal Statistical Society, Series B: Methodological*, Vol. 57, pp.237–246.

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

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

Ghosh, J. and 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*, Vol. 106, No. 495 pp.1041–1052.

Google.org (2012) [online] http://www.google.org/flutrends/about/how.html.

Harvey, A.C. (1989) *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.

Hoeting, J.A., Madigan, D., Raftery, A.E. and Volinsky, C.T. (1999) 'Bayesian model averaging: a tutorial (disc: P401-417)', *Statistical Science*, Vol. 14, pp.382–401.

Kalman, R. (1960) 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering*, Vol. 82, pp.35–45.

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

Madigan, D. and Raftery, A.E. (1994) 'Model selection and accounting for model uncertainty in graphical models using Occam's window', *Journal of the American Statistical Association*, Vol. 89, No. 428 pp.1535–1546.

McCausland, W.J., Miller, S. and Pelletier, D. (2011) 'Simulation smoothing for state-space models: a computational efficiency analysis', *Computational Statistics and Data Analysis*, Vol. 55, No. 1, pp.199–212.

Rue, H. (2001) 'Fast sampling of Gaussian Markov random fields', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 63, No. 2, pp.325–338.

Shaman, J. and Karspeck, A. (2012) 'Forecasting seasonal outbreaks of influenza', *Proceedings of the National Academy of Science* [online] http://www.pnas.org/content/early/2012/11/21/1208772109.

Stock, J.H. and Watson, M.W. (2002a). 'Forecasting using principal components from a large number of predictors', *Journal of the American Statistical Association*, Vol. 97, pp.1167–1179.

Stock, J.H. and Watson, M.W. (2002b). 'Macroeconomic forecasting using diffusion indexes', *Journal of Business and Economic Statistics*, Vol. 20, pp.147–162.

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*, pp.233–243, North-Holland/Elsevier.