Automatic Model Selection: An Evaluation

Fiona Sloof

Kellogg College, University of Oxford

Word count: 29,346

Submitted in partial fulfilment of the requirements for the degree of the Master of Philosophy in Economics

Trinity Term 2016

Abstract

Three different automatic model selection algorithms are described. Various criteria for the evaluation of model selection algorithms are outlined including the gauge, potency and conditional mean squared errors. Monte Carlo simulations are used to evaluate the algorithms across three different states of nature. The algorithms are compared based on their ability to select relevant variables, exclude irrelevant variables and the accuracy of their parameter estimates. They are also compared to the benchmarks or 'best case scenarios' providing a measure of the costs of search and inference. A separate application of automatic model selection for nowcasting is then considered. The algorithms are used to nowcast and forecast the rate of influenza-like illness in the United States using Google search queries.

Acknowledgements

I would like to thank my dedicated and endlessly enthusiastic supervisors Jennifer Castle and David Hendry. Their brilliance and passion for econometrics and automatic model selection is inspiring, and the insight, guidance and support they have provided every step of the way is immeasurable. It has been been an absolute privilege working with them.

I would like to extend my thanks to my wonderful parents and Kevin for their never ending support, love and encouragement through the entire process. And to Whiskers, the best study buddy in the world.

Declaration

I declare that this thesis is entirely my own work unless otherwise stated.

Contents

Chapter 1

Introduction

At the moment, big data is a big deal. Technology developed in the past few decades means that more information than ever before is being collected. Management, storage and processing capabilities have advanced rapidly in recent years. There is a huge amount of information and potential in this data.

In many fields, one of the biggest buzzwords is machine learning. Broadly, machine learning consists of techniques, algorithms, etc., which allow us to 'learn' from the data. Machine learning discovers patterns and makes predictions, with minimal conditions. Given that the world has more data than most people know how to handle, computers end up doing most of the work and discovery, making understanding data and utilizing it to its fullest potential much easier.

Economic modelling stands to benefit immensely from big data and the ideas in machine learning. More data can provide a clearer picture of the world, and should contribute to a clearer understanding of its dynamics and complexities. However, while data has become increasingly available, the implementation of the theory regarding how to actually use all this data is still catching up. The field of economics is still figuring out how to exploit big data.

The reason why economics hasn't enthusiastically 'jumped on the machine learning train' lies in the fact that econometrics is fundamentally concerned with inference and machine learning is primarily related to finding patterns and making predictions. This means that the objective in machine learning algorithms is for example to minimize the prediction error, not to necessarily understand how the predictions are being formed. For economists however, there are three main uses for economic modelling: forecasting, policy and modelling. In the modelling and policy worlds economists are concerned with what drives a particular variable. Standard econometrics allows economists to test models, draw conclusions, and provide information on the underlying process driving a particular variable. This is valuable because it then allows a researcher, government, business, etc., to understand the impact of changing a 'driving force' on a given variable.

For economic modellers and policy makers an important question arises. How can we use big data to our advantage without foregoing inference? This is where automatic model selection becomes a very useful tool.

As its name would suggest, automatic model selection is using algorithms to automatically select models. Very generally, model selection algorithms work to find the 'best' model (according to a particular set of criteria, objective function, etc.) of a particular dependent variable, given a set of potential explanatory variables. This is incredibly useful; the computer is performing tests and doing the work that a researcher might normally do herself.

It is important to point out that there are differences between model selection and forecasting. The goal of model selection is to select a model which is as close to the data generating process (DGP) as possible. The goal of forecasting is to produce models which have minimal prediction error. Because the world is non-stationary, full of correlations and structural breaks these two goals do not necessarily coincide. Standard machine learning techniques can generally only be applied to forecasting and prediction problems, which highlights the need for techniques focussed on the model selection problem to be developed and analyzed.

The goal of this thesis is to study, evaluate and compare the performance of model selection algorithms under different states of nature. The state of nature refers to the correlation structure of the variables being considered in the model. Three different algorithms are tested: Autometrics, the Lasso and Bayesian Structural Time Series. This is the first time a comparison between these three algorithms has been undertaken. While studies do exist which feature pairwise comparison, these are rare and do not consider the various states of nature considered here. Additionally, the studies that do exist do not endeavor to compare the algorithms using the metrics used here and tend to focus on the predictive power (i.e. accuracy of forecasts) of model selection algorithms. In this thesis Monte Carlo simulations are used so the DGP is known, which allows the algorithms to be compared using metrics which are more relevant to the model selection problem.

The thesis is structured as follows: Chapter 2 describes the algorithms included in this study, Chapter 3 tests the algorithms in the context of Monte Carlo simulations, Chapter 4 is an empirical application of automatic model selection building off the Google Flu Trends application and Chapter 5 concludes.

Chapter 2

The Algorithms

This chapter outlines Autometrics, the Lasso and Bayesian Structural Time Series, the three different automatic model selection algorithms which are tested and compared in this thesis. The difficulties and dangers associated with the 'black box' treatment of econometric software by econometric practitioners is also examined.

2.1 Autometrics

Autometrics is an application of general-to-specific (GETS) model selection. GETS model selection begins with the formulation of an initial model, which includes all variables which the modeller believes may be regressors in the final model. The model space is the set of all possible models; in theory, for k variables there are 2^k possible models to evaluate and compare. GETS algorithms search through the model space by systematically eliminating variables, and estimating the resulting models. Various tests are conducted as the algorithm progresses through the model space, which ensure that not too much information is being lost relative to the initial model, and that the estimated models are statistically well-behaved. Often there will be multiple models that the algorithm deems to be sufficiently informative and well-behaved. Therefore a tie-breaking criteria is often required to select the final model. An overview of the GETS procedure can be found in the General-to-specific modelling section of Hendry and Doornik (2014).

Following work done by Hoover and Perez (1999), and Hendry and Krolzig (1999 and 2005), Autometrics is a third implementation of a GETS modelling procedure. While all GETS algorithms have the same goal, the implementation varies. The following four features broadly describe the main components of GETS model selection, and are the four main ways in which various GETS algorithms may differ in their approach:

1. Formulation of a General Unrestricted Model (GUM).

2. Path search through the model space.

3. Tests of resulting models: both statistical validity tests and encompassing tests.

4. Tie-breaker for resulting models.

With these in mind, Autometrics consists of four main components, or broadly, 'steps'. First, the GUM is formulated and becomes the initial model. It is up to the researcher to determine which variables are included in the GUM. The GUM is estimated and must pass a series of diagnostic tests and also provide sufficient information about the variable being modelled. Included in the battery of tests are normality, residual correlation, residual ARCH, heteroskedasticity and in-sample Chow tests. The modeller has the choice to set the p-value for these tests, and if the initial model fails, this can be adjusted. A detailed treatment of the Autometrics algorithm can be found in Doornik (2009).

Consider the following simple example, taken from Doornik (2009). A modeller thinks variables A, B, C and D may be relevant for modelling variable Y, but is unsure if he should include all four (obviously this a very simple example and in practice a modeller would be unlikely to use Autometrics to determine the model). In this case, the GUM would be the model composed of variables A, B, C and D. This model is required to pass tests which ensure it is statistically well-behaved, before the algorithm will proceed. The significance level for these tests is set by the modeller and is denoted p_d.

The second component in the Autometrics algorithm is a tree search through the model space. The ideas of tree search can be more easily understood with reference to Figure 2.1. The initial model, the GUM, is estimated. Then t-statistics for all the potential regressors are calculated, and from these values, the variables are ranked from least significant to most significant. Continuing with the same example, the GUM is the model which includes variables A, B, C and D, and is depicted in the diagram as node ABCD. Suppose that variable A is the least significant, followed by B, C, then D. The tree search begins by eliminating the least significant variable and estimating the resulting model. In the example, this corresponds to moving to node BCD. Model BCD is backtested with respect to the GUM via a simple F-test, where BCD is the restricted model and ABCD is the unrestricted model. If backtesting does not reject model BCD, the next least significant variable, which is B in the example, is eliminated and the resulting model is estimated and backtested with respect to the GUM. This continues until either a model fails with backtesting or until there are no more variables to eliminate. Once either of these occurs, the search continues by backtracking through the tree until a subbranch that has not been explored fully is reached. In the example, once model D is estimated and tested, the algorithm backtracks to model CD, then continues by eliminating D and estimating model C. The algorithm then backtracks to model BCD (since all subbranches from model CD have been explored) and continues by eliminating variable C, since after B, this is the least significant variable. The algorithm continues in this way, and if no model ever failed with respect to backtesting, all 2^k models would be estimated. However, as mentioned, estimating and testing all 2^k models becomes computationally infeasible very quickly. There are several techniques Autometrics utilizes to improve on its computational efficiency: namely

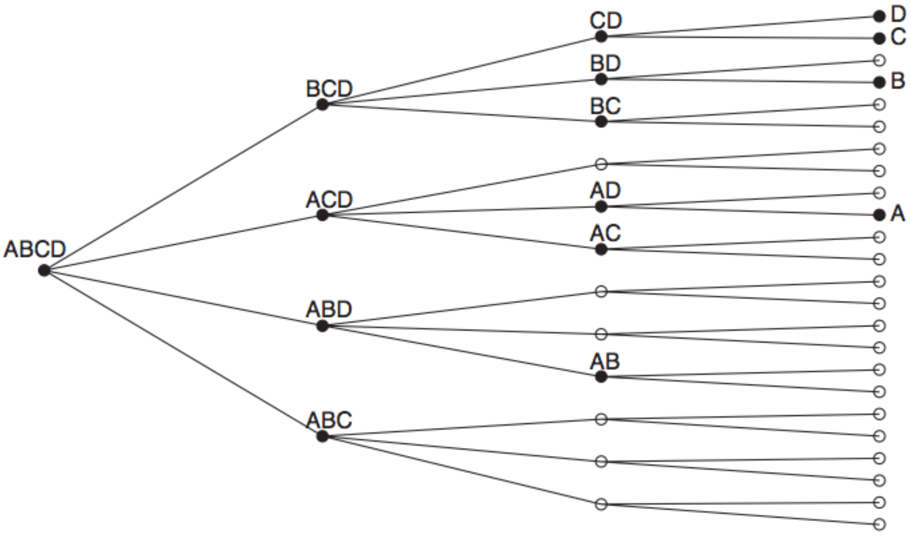


Figure 2.1: An example of tree search
Source: Doornik (2009)

pruning, bunching, chopping and model contrasts:

- Pruning: If a model fails after the deletion of one variable (either as a result of backtesting, or because of diagnostic tests) then all models emanating from this branch can be ignored. In the example, if deletion of variable A causes model BCD to fail, then all the models in subbranch BCD can be ignored. This process is called pruning. No variable with significance level less than p_a, which is the main Autometrics p-value and is set by the user, can be deleted from the model.
- Bunching: Variables are grouped together in bunches and simultaneously deleted from the model. If the resulting model passes both backtesting and all diagnostic tests, then all of the subbranches leading away from these variables can be deleted. If variables are bunched together, deleted and the resulting model fails the tests, then a smaller bunch is deleted and the resulting model is tested. This continues until a model passes the tests or until the number of variables in a 'bunch' equals 1. In the example, beginning at node BCD, if BC were sufficiently insignificant to be bunched together, the resulting model would include just D. If this model passes, then any model including B and C can be ignored. The amount of bunching is determined by p_b, and by default $p_b = max\{\frac{1}{2}p_a^{1/2}, p_a^{3/4}\}$.
- Chopping: If a variable is 'highly insignificant', all models which include it can be ignored. In the example, if A is sufficiently insignificant, then any models which include it are not estimated. Chopping is governed by p-value p_c, and by default, $p_c = p_b$.
- Model Contrasts: A terminal model is one that cannot be reduced any further. Once a terminal model has been found, it is possible to determine which variables must be deleted

in order to end up with a different model. Any models which include these variables can be ignored.

To summarize, the second component of the algorithm is a tree search through the model space, which involves systematically deleting variables and estimating the resulting models. Pruning, bunching, chopping and model contrasts are used in order to ignore certain branches of the tree, which greatly improves computational efficiency.

The third component in the algorithm is testing. As the algorithm progresses through its tree-search, tests are conducted. There are two types of tests: backtesting with respect to the GUM and diagnostic tests which evaluate if the model is statistically well-behaved. Backtesting with respect to the GUM, often just referred to as backtesting, occurs each time a variable (or a bunch of variables) is deleted and exists to ensure that the deletion of a variable is not causing significant information to be lost. Backtesting is done with an F-test, is governed by p_d and by default $p_d = p_a$. If backtesting fails it means that the eliminated variable contains non-negligible information about the variable being modelled. Models extending from a subbranch which fails with backtesting therefore are not considered.

Diagnostic testing is conducted differently, as it is computationally expensive and not efficient. Autometrics avoids this by only performing diagnostic tests once a terminal model has been found. If the terminal model fails diagnostic tests, then the algorithm backtracks along the branch that led to this model, testing each model until one is found which passes the diagnostic tests. This then becomes the terminal model. This means that insignificant variables can be retained.

The fourth component of Autometrics is using an iterative procedure to select the final model. Once the tree search is done, there are likely to be multiple terminal models. The union of all terminal models is found and each of the terminal models is backtested with respect to this union. Any model which fails with respect to backtesting is deleted, and the union of the remaining models becomes the GUM. The same tree search and testing procedure as outlined above is then performed, with this new GUM as the starting point. This iterative procedure continues until the union of terminal models in one iteration is the same as that in the previous iteration (convergence). Then if there are still multiple terminal models, the minimum Schwarz Criterion is used as a tie-breaker.

There are several additional optional features of the Autometrics algorithm, including a presearch feature, impulse indicator saturation to test for outliers and step indicator saturation to find structural breaks. The presearch includes lag reduction and variable reduction. Autometrics is designed so that while presearch can be beneficial, it is not necessary. It works particularly well if the variables are orthogonal or if there are only a few highly significant variables. These features are not used in this thesis but details on impulse indicator saturation and step indicator saturation can be found in Hendry et. al (2008), Johansen and Nielsen (2009) and Castle et. al (2015).

2.2 The Lasso

The Lasso (least absolute shrinkage and selection operator) is a technique for estimating models which involves variable selection by setting some coefficient parameters equal to zero, effectively eliminating them from the model (Tibshirani, 1996). It is based on the same ideas as ridge regression (Hoerl and Kennard, 1970), however in ridge regression the objective function is such that coefficients are shrunk towards zero, while never actually equalling zero.

The Lasso estimator is defined by:

$$\hat{\beta}^{Lasso} = \underset{\beta}{\operatorname{argmin}} \|y - \sum_{j=1}^{N} x_j \beta_j\|^2 + \lambda \sum_{j=1}^{N} |\beta_j|$$

The tuning parameter λ governs how much of a penalty there is for non-zero coefficients and can be chosen according to a variety of criteria, depending on the objective. As λ increases, the Lasso shrinks the coefficients towards zero. If λ is zero, the Lasso estimator becomes the OLS estimator. An important question is how to choose λ. Generally, packages that implement the Lasso will produce parameter estimates for a range of values of λ. Among techniques widely applied to select the most appropriate tuning parameter are cross-validation, and information criteria (BIC). There are different types of cross-validation, including K-fold, generalized cross-validation, and leave-one-out. Studies examining different methods for selecting the correct tuning parameter include Tibrishirani (1996), Efron et al (2004), Zou et al (2007), Chen and Chen (2008), Zhang et al (2010) among may others. There is no 'right' way to choose the tuning parameter, but the default of many statistical packages is K-fold cross validation which works by dividing the data set into K sets, computing the cross-validation error for a range of values of λ and selecting the λ which produces the smallest cross-validation error. Details on this procedure can be found in James et al (2013).

2.3 Bayesian Structural Time Series (BSTS)

BSTS is a system developed by Hal Varian and Steven Scott, with the purpose of nowcasting and short-term forecasting when there are many potential regressors (Scott and Varian, 2014). There are two main components in the system: a structural time series component and a regression component. In the regression component, variable selection techniques are used, allowing BSTS to be used on data sets with many potential regressors.

2.3.1 Structural Time Series Component

The purpose of the first step in the BSTS algorithm is to capture the time series components (trends and seasonal patterns) in the data. To understand how this step of BSTS works, it is useful to use state space representation. Using the notation of Durbin and Koopman (2001), a structural time series model is given by the following two equations:

$$
\begin{aligned}
y_t &= Z_t'\alpha_t + \epsilon_t && \epsilon_t \sim N(0, H_t) \\
\alpha_{t+1} &= T_t\alpha_t + R_t\eta_t && \eta_t \sim N(0, Q_t)
\end{aligned}
$$

where y_t denotes an observation, and α_t is a vector of latent state variables. The first equation is the observation equation and specifies how the observed variable y_t is related to the latent state. The second equation is called the transition equation because it specifies how the latent state evolves. The latent state can include components which model trends and seasonality which are captured in Z_t, T_t and R_t. This state space specification encompasses a large range of models. As an example, the following is the 'basic structural model':

$$
\begin{aligned}
y_t &= \mu_t + \tau_t + \epsilon_t && \epsilon_t \sim N(0, \sigma_\epsilon^2) \\
\mu_t &= \mu_{t-1} + \delta_{t-1} + u_t && u_t \sim N(0, \sigma_u^2) \\
\delta_t &= \delta_{t-1} + v_t && v_t \sim N(0, \sigma_v^2) \\
\tau_t &= -\sum_{s=1}^{S-1} \tau_{t-s} + w_t && \tau_t \sim N(0, \sigma_w^2)
\end{aligned}
$$

Here, μ_t denotes the trend, δ_t is the slope of the trend at t and τ_t denotes the seasonal component. The state α_t is made up of trend and seasonal components so $\alpha_t = (\mu_t, \delta_t, \tau_t)$. Using the terminology of Durbin and Koopman again the first equation is the observation equation and the subsequent three together can be expressed as the transition equation as above. BSTS adds a regression component to the basic structural model, so the observation equation becomes:

$$
y_t = \mu_t + \tau_t + \beta'\mathbf{x}_t + \epsilon_t \qquad \epsilon_t \sim N(0, \sigma_\epsilon^2)
$$

where $\mathbf{x}_t$ is a vector of regressors the researcher believes may be relevant for modelling y_t. This 'basic structural model + regression component' is the model that BSTS estimates. This distinction between the time series and structural 'parts' of y_t is fundamental to the estimation method by which BSTS produces its estimates. BSTS uses Kalman filtering, smoothing and Bayesian data augmentation to determine the time series component of y_t, subtracts this calculated time series component from y_t and then uses spike-and-slab regression on the 'remaining' part of y_t. The Kalman filtering, Kalman smoothing and Bayesian data augmentation techniques are now briefly described. For a much more detailed explanation on these techniques, readers should consult Durbin and Koopman (2001).

Kalman filtering is used to predict the distribution of a latent process. For example, consider the simple local level model described by the following equations:

$$\begin{aligned} y_t &= \alpha_t + \epsilon_t & \qquad \epsilon_t &\sim N(0, \sigma^2_\epsilon) \\ \alpha_{t+1} &= \alpha_t + \eta_t & \qquad \eta_t &\sim N(0, \sigma^2_\eta) \end{aligned}$$

Kalman filtering works by first finding the distribution of α_t, conditional on the information available at $t-1$, which is contained in the series of observations $y_{1:t-1} = y_1, ..., y_{t-1}$. This distribution is denoted $p(\alpha_t | y_{t-1})$. Once a new data point, y_t, becomes available Kalman filtering updates the distribution of α_t, producing $p(\alpha_t | y_t)$. In determining $p(\alpha_t | y_t)$, the algorithm uses information from the previously found distribution, the new data point, and the variance of each process. Estimates for the α_ts are found by taking expectations from the calculated distributions. The 'filter' is the term that determines how much of the new estimate should reflect the new information available via y_t. If for example there is a lot of noise in the y_ts then the filter will place less weight on the new observation, and more on the previous estimate of α_t. The outputs of the Kalman filter are the estimated α_ts, the prediction errors denoted by v_t (calculated from $v_t = y_t - \alpha_t$), the prediction error variance, and the variance of the estimated α_ts.

Kalman smoothing works to smooth out the estimates of the α_ts after the Kalman filter has been applied. Kalman smoothing finds new distributions of $\alpha_1, \alpha_2, ..., \alpha_n$ using the entire sample Y_n. This is in contrast to filtering which estimates α_t using only information available up until time t. Kalman smoothing relies on the idea of backward recursion, to produce the updated distributions, which are denoted $p(\alpha_t | y_{1:n})$ to reflect the fact that they are calculated using all n observations of y_t.

Estimation of the regression component of BSTS requires values of α_t. Bayesian data augmentation makes it possible to simulate α_t from the distributions derived from Kalman filtering and smoothing. It is not possible to draw α_ts directly from the derived $p(\alpha_t | y_{1:n})$ because of the correlation that exists between α_t and α_{t+1}. The authors of BSTS use an algorithm developed by Durbin and Koopman which takes this correlation into account and produces simulated values for the α_ts.

Using these three techniques together, the algorithm determines 'how much' of the dependent variable can be explained by structural time series components. This component is then subtracted from the dependent variable, and the regression techniques are run on what 'remains' in y_t. As a side note, simply subtracting the times series component from y_t is theoretically flawed, and results in biased and inconsistent estimates of the regression parameters. This is shown in a simple proof later on.

2.3.2 Regression Component

The regression component of the algorithm uses Bayesian parameter estimation techniques. Bayesian parameter estimation is based on Bayes' formula. If θ are the parameters that need to be estimated, and the available data is $\mathbf{d}$, then Bayes' formula states:

$$\Pr(\theta|\mathbf{d}) = \frac{\Pr(\mathbf{d}|\theta)\Pr(\theta)}{\sum_{\theta'}^{\Theta}\Pr(\mathbf{d}|\theta')\Pr(\theta')}$$

By setting priors which reflect beliefs about the distribution of the parameters, Bayes' formula can be used to update these beliefs and obtain a posterior distribution using the actual realized data. To use these techniques priors for the parameters of interest need to be set. Often these are set somewhat ambiguously. Once the priors have been set and posteriors have been calculated, parameter estimates can be inferred for example by finding the mean of the posterior, or performing simulations. When the prior and posterior distributions come from the same family of distributions they are called conjugates. This is often a desirable feature for priors.

BSTS uses a spike-and-slab prior on the regression coefficients, which induces sparsity in the model (Ishwaran, 1970). For the purposes of describing the algorithm, consider the following model:

$$y_t = \sum_{j=1}^{N}\beta_j x_{j,t} + \epsilon_t$$

where the parameters to be estimated are the β_js. Let $\gamma_j = 1$ if $\beta_j \neq 0$ and let $\gamma_j = 0$ otherwise (so γ is an indicator variable). The spike-and-slab prior takes the following form:

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

Priors must be set for $p(\gamma)$, $p(\sigma_\epsilon^2|\gamma)$ and $p(\beta_\gamma|\gamma, \sigma_\epsilon^2)$, where the γ index is used to indicate that it refers only to the distribution of parameters of the variables where $\gamma_k = 1$. In the BSTS package, default priors are built in, but can be changed.

The $p(\gamma)$ prior gives the prior belief that a particular combination of variables is selected as the set of regressors in the model. Because the objective of spike-and-slab is to induce sparsity in the model, the prior is set as an independent Bernoulli prior, where π_n is the probability that a particular variable is included in the model, or $\Pr(\gamma_n = 1) = \pi_n$, and therefore:

$$\gamma \sim \prod_{n=1}^{N}\pi_n^{\gamma_n}(1-\pi_n)^{1-\gamma_n}$$

For simplicity, it is often assumed that $\pi_n = \pi$ for all n. When this is the case, π_n represents the proportion of all possible predictors which are expected to be selected and in the final model. The default in BSTS is set to $\pi_n = \pi = 0.5$. If the researcher has reason to believe that a particular variable almost certainly should be included in the model, they can set $\pi_k = \Pr(\gamma_n = 1)$ close to one to reflect this. The distribution of γ is the 'spike' because it sets the probability that $\beta_n = 0$ at some positive value (resulting in a sparse model).

BSTS sets the conditional variance prior as the conjugate:

$$\frac{1}{\sigma_\epsilon^2}|\gamma \sim Ga(\frac{v}{2}, \frac{ss}{2})$$

where $Ga(r, s)$ refers to the Gamma distribution, with mean r/s and variance r/s^2. The Gamma distribution is often used as a prior for the variance as the parameters r and s are always positive. Here, the researcher has the choice to set the parameters v, interpreted as the prior sample size, and ss, interpreted as the prior sum of squares, as they wish. The most straightforward method of setting ss and v is for the user to provide the expected R^2 and the expected sample size v. Then, BSTS uses a trick to calculate the sum of squares ss. Let rss denote the residual sum of squares. Since:

$$1 - R^2 = \frac{rss}{ss}$$

and

$$s_y^2 = \frac{ss}{v}$$

it follows that:

$$\frac{ss}{v} = (1 - R^2)s_y^2$$

That is, the marginal standard deviation of y, s_y, is used to find the prior sum of squares, which is then used in the prior for the standard deviation. Using s_y^2 in the prior for σ_ϵ^2 violates Bayesian rules whereby priors cannot be data determined. The defaults in BSTS are $v = 0.01$ and $R^2 = 0.5$.

The prior for the conditional distribution of β is also a conjugate and is set in BSTS as:

$$\beta_\gamma | \sigma_\epsilon^2, \gamma \sim N(b_\gamma, \sigma_\epsilon^2(\Omega_\gamma^{-1})^{-1})$$

The researcher has a choice to set the prior means of the β_ks, as well as the the information matrix Ω_γ. The default in BSTS sets $b_\gamma = 0$ and sets the information matrix as $\Omega^{-1} = \kappa X'X$, where κ is a weighting term. If the researcher uses these default settings in BSTS, the problem of priors being data driven exists again. The default calculates the second moment terms in the information matrix using the full sample. Once the priors are set by the researcher, the conditional posterior distributions of β and σ_ϵ^2 can be calculated from conjugacy formulas.

Define $y_t^* = y_t - Z_t^{*'}\alpha_t$, where $Z_t^{*'}\alpha_t$ is the time series component. That is, y_t^* is the dependent variable with the time series component subtracted. Setting $\mathbf{y*} = y_{1:n}^*$,

$$\beta_\gamma | \sigma_\epsilon, \gamma, \mathbf{y*} \sim N(\tilde{\beta_\gamma}, \sigma_\epsilon^2(V_\gamma^{-1})^{-1})$$
$$\frac{1}{\sigma_\epsilon^2} | \gamma, \mathbf{y*} \sim Ga(\frac{N}{2}, \frac{SS_\gamma}{2})$$

The sufficient statistics to calculate these distributions are:

$$\begin{aligned}
V_\gamma^{-1} &= (\mathbf{X}'\mathbf{X})_\gamma + \Omega_\gamma^{-1} \\
N &= v + n \\
\tilde{\beta_\gamma} &= (V_\gamma^{-1})^{-1}(\mathbf{X}_\gamma^T \mathbf{y*} + \Omega_\gamma^{-1} b_\gamma) \\
SS_\gamma &= ss + \mathbf{y*}'\mathbf{y*} + b_\gamma^T \Omega_\gamma^{-1} b_\gamma - \tilde{\beta_\gamma}_\gamma^T V_\gamma^{-1} \tilde{\beta_\gamma}_\gamma
\end{aligned}$$

The posterior distribution of γ is different because it does not come from a conjugate prior. Let C($\mathbf{y*}$) be a normalizing constant. It can be shown that the posterior distribution of γ is (Scott and Varian, 2014):

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

To summarize, at this point the prior distributions have been described, the parameters which can be set by the researcher (or which are alternatively set to the BSTS defaults) have been outlined, and the calculated posterior distributions have been provided. The next step is understanding how these posteriors are employed in the algorithm itself to estimate the parameters.

2.3.3 Markov Chain Monte Carlo Simulations in Bayesian Parameter Estimation

The objective of Markov Chain Monte Carlo simulations (MCMC) is to estimate the parameters of interest (Robert and Casella, 2004). In BSTS, the parameters can be classified as either relevant to the time series component or the regression component. These two sets of parameters are both estimated via simulation, but the method of simulation differs. Broadly, these two 'methods' correspond to two main steps:

1. Kalman filter, Kalman smoothing and Bayesian data augmentation are used to simulate α, the time series component, as well as any other parameters in the state space model. In the basic structural model above, this would include the variance parameters σ_u^2, σ_v^2, σ_τ^2. Durbin and Koopman's simulation smoother is used for this step (Durbin and Kooman, 2001).

2. Simulate β and σ_ϵ^2 from the posterior conditional probabilities. This involves several steps because the posterior probabilities are conditioned on γ. In order to draw values of β and σ_ϵ^2 draws of γ are required. Values of γ are drawn via stochastic search variable selection (SSVS) (George and McCulloch, 1993). SVSS is a variable selection technique which relies

on Gibbs Sampling. As outlined above, the posterior distribution of γ is based on a number of statistics which are provided by the researcher, as well as the prior for γ, $p(\gamma)$. If there are n possible regressors, then there are 2^n possible subsets of regressors. It is computationally infeasible to calculate the posteriors for each of these subsets. Gibbs sampling is therefore used, generating a sequence of 'model subsets' by selecting m different draws of γ:

$$\gamma_1, ..., \gamma_m$$

where each γ_k is one of the m sample draws, corresponding to a different selection of the possible regressors being selected.

Gibbs sampling operates so that only subsets which are highly probable will be sampled and included in the sequence. As it continues and generates more draws, the sampler has more information, and it begins to narrow in on the regressors which are most likely to be included in the model, selecting them more often. The sequence of draws therefore gives a good indicator of the subset that should be included in the model. Tabulating across the sequence gives a useful measure of the probability that a particular variable should be included in the model. In many cases the sequence of draws converges rapidly in distribution to $\gamma \sim f(\gamma|Y)$. The nature of the Gibbs sampler means that it converges quickly, making it far more efficient to compute than the actual posterior. The draw of γ from Gibbs sampling is used to find the conditional σ_ϵ^2 and β distributions. From these, draws are obtained for σ_ϵ^2 and β from their respective posterior distributions.

To summarize, in each repetition, draws of γ from SSVS and Gibbs sampling are obtained. These are used to compute the posterior distribution of σ_ϵ^2, which is used to compute the posterior distribution of β. These distributions can then be used to obtain parameter estimates for β.

Because the algorithm computes distributions and not point estimates for the parameters, the algorithm does not explicitly select variables; it calculates the probability that a particular variable has coefficient $\beta \neq 0$. To use BSTS as a selection algorithm then, the user must determine what the threshold for considering a variable selected is.

2.3.4 A Note on the Bias and Inconsistency of BSTS Estimates

After calculating the time series component, BSTS subtracts this from the dependent variable and proceeds with spike-and-slab regression to estimate the parameters on the regression components. The process employed by BSTS to 'eliminate' the time series component from the dependent variable is theoretically flawed and likely results in biased coefficient estimates. The complexity of the process through which the time series component is determined, and the use of spike-and-slab regression make this difficult to prove directly, but the impact of performing this sort of 'trick' can easily be seen in the case of a multivariate regression using OLS. Consider the following process, which using the terms from BSTS, consists of both a time series component

Y_Γ and a regression component $X\beta$:

$$Y = Y\_\Gamma + X\beta + \epsilon \tag{2.1}$$

where Y is a $T \times 1$ matrix, $Y\_$ is a $T \times L$ matrix containing lags of Y, seasonal indicators, and variables relevant to the time series dynamics of Y, and X is a $T \times K$ matrix containing regression components. BSTS estimates the time series component using Kalman filtering, smoothing and Bayesian data augmentation described above, finding estimates for the latent state variables in α in the following equation:

$$Y = Z'\alpha + u \tag{2.2}$$

where Z is the $S \times T$ observation matrix containing all state components (i.e. local linear trends, or in this case, an autoregressive component), and α is a $S \times 1$ matrix containing the latent components. To find the regression component, BSTS estimates the following spike-and-slab regression, with $Y^* = Y - Z'\alpha$:

$$Y^* = X\beta + \epsilon \tag{2.3}$$

Since both Y_Γ and $Z'\alpha$ correspond to the time series dynamics of Y, assume that:

$$Y\_\Gamma \approx Z'\alpha \tag{2.4}$$

So that estimating (2.3) is equivalent to estimating:

$$Y - Y\_\Gamma = X\beta + \epsilon \tag{2.5}$$

Estimating (2.3) by OLS results in the following estimate for β:

$$\widehat{\beta} = (X'X)^{-1}X'Y^* \tag{2.6}$$

However the true value of β is found from estimating (2.1). By the Frisch-Waugh theorem, estimating (2.1) is equivalent to estimating:

$$M_{Y\_}Y = M_{Y\_}X\beta + M_{Y\_}\epsilon \tag{2.7}$$

where $M_{Y\_}$ is the annihilator matrix:

$$M_{Y\_} = I - Y\_(Y\_'Y\_)^{-1}Y\_' \tag{2.8}$$

The correct estimates for β from (2.1) are therefore given by:

$$\widetilde{\beta} = ((M_{Y\_}X)'(M_{Y\_}X))^{-1}(M_{Y\_}X)'M_{Y\_}Y \tag{2.9}$$

Unless $Y\_$ is orthogonal to X or $\alpha = 0$, $\widehat{\beta} \neq \widetilde{\beta}$. Therefore, to obtain correct parameter estimates for β in the regression which does not include $Y\_$ as a regressor it is necessary to apply transformations to Y and X to get Y^{**} and X^{**}:

$$Y^{**} = M_{Y\_}Y \tag{2.10}$$

$$X^{**} = M_{Y\_}X \tag{2.11}$$

So, Y^{**} and X^{**} should be used in the spike-and-slab regression. By design however, the algorithm uses untransformed X, and $Y^* = Y - \alpha Z$. Since

$$Y^* = Y - \alpha Z \neq M_{Y\_}Y = Y^{**} \tag{2.12}$$

and

$$X \neq M_{Y\_}X = X^{**} \tag{2.13}$$

it follows that unless $Y\_$ and X are orthogonal, so that none of the time series dynamics present in Y are present in X, or if $\alpha = 0$, so there are no time series dynamics in Y, the coefficient estimates from the spike-and-slab regression will be incorrect.

2.3.5 Summary table for Autometrics, the Lasso and BSTS

To use any of the three algorithms, user inputs are required. Table 2.1 provides a summary of the user inputs and how they influence the algorithm results.

Algorithm	Setting	Description	Default
Autometrics	α	- Must be set by user: main Autometrics p-value - α governs the level of pruning, so $p_a = \alpha$. - Close to the fraction of irrelevant variables retained by the procedure	No - set by user
	p_b	- Governs the amount of bunching - Not (usually) set directly by user - Too high results in excessive backtracking - Too low results in bunching effectively turned off	$p_b = max\{\frac{1}{2}p_a^{1/2}, p_a^{3/4}\}$
	p_c	- Not (usually) set directly - Governs what is considered ‘highly’ significant, and therefore chopping	$p_c = p_b$
	p_d	- Not (usually) set directly by user - Built in features to relax this if necessary	$p_d = p_a$
Lasso	tuning parameter λ	- λ determines the sparsity of the selected model - Higher λ will result in a sparse model - Lower λ will result in a less sparse model - If $\lambda = 0$ the Lasso is equivalent to OLS - Various techniques for choosing the ‘best’ λ, usually some form of cross validation or information criteria is used	No - either set by user or determined via a separate algorithm
BSTS	Prior for γ	- Bernoulli prior: $\gamma \sim \prod_{n=1}^{N} \pi_n^{\gamma_n}(1-\pi_n)^{1-\gamma_n}$ - User could choose to use an entirely different prior if so inclined - If $\pi_n = \pi \; \forall n$ then π is the expected model size (or fraction of non-zero predictors); therefore π affects sparsity of selected model - If user has reason to believe that a certain predictor is particularly relevant, can set π_n high for that regressor	Bernoulli prior with $\pi_n = \pi = 0.5$
	Prior for $\frac{1}{\sigma_\epsilon^2}\|\gamma$	- Gamma prior: $\frac{1}{\sigma_\epsilon^2}\|\gamma \sim Ga(\frac{v}{2}, \frac{ss}{2})$ - ss interpreted as prior sum of squares - v interpreted as prior sample size - Can also be set by asking user for expected R^2 and number of observations worth of weight v - User could choose to use an entirely different prior if so inclined	Gamma with parameters determined by $R^2 = 0.5$, $v = 0.01$
	Prior for $\beta_\gamma\|\sigma_\epsilon^2$	- Normal prior: $\beta_\gamma\|\sigma_\epsilon^2, \gamma \sim N(b_\gamma, \sigma_\epsilon^2(\Omega_\gamma^{-1})^{-1})$ - Very common to assume vector of prior means $b = 0$, however if user has reason to believe a certain predictor is particularly relevant, can set b_γ higher, thereby affecting sparsity - Information matrix Ω_γ^{-1} default violates Bayesian rules as is data determined	Normal prior with $b = 0$, $\Omega_\gamma^{-1} = \kappa\mathbf{X}'\mathbf{X}/n$, κ observations worth of weight on prior b

Table 2.1: Algorithm settings

2.4 A Note on Econometric Software as a 'Black Box'

The previous sections outlined and explained Autometrics, the Lasso and BSTS in a fair amount of detail. While each effectively accomplish the same task of selecting a model from a general unrestricted model, each does so in a very different manner, relying on different statistical, computational and econometric techniques. Ideas from each of these three disciplines are used to varying degrees, and combine to create algorithms which may be difficult for an economist with little training in computer programming, or a computer programmer with little training in economics to understand. On the surface this seems to imply that usage of the algorithms would be restricted to the subset of economists/statisticians/computer scientists who have an understanding of all three subjects, and indeed the ideal user would have an understanding of these three disciplines. In practice however this is generally not the case. Since its inception, there has been a separation between those who derive econometric theory, those who translate this theory into econometric software thereby making it accessible to econometric practitioners, and those who actually use the econometric software to conduct their research. This means that there is little requirement for the practitioners to understand the processes going on 'behind the scenes' when they estimate a model, or when they consider the results of statistical tests displayed in the output. Most practitioners simply take displayed results as correct, under the assumption that the program they are working with has correctly implemented the theory behind the estimators, tests, etc. which they are interested in.

As discussed at length in the text 'The Practice of Econometric Theory' (Renfro, 2009) the development of econometrics and computer software over the past sixty years has led to a divergence between those who develop, implement and apply econometric theory. There are many reasons for this disparity. Perhaps the biggest is that there is little incentive for an econometric practitioner to spend the time learning and developing their own econometric software due to the time investment it would require and the availability of software which performs most of what a practitioner would like it to. The cost of learning a language, writing code and maintaining documentation, is huge and, particularly when there are a range of existing options which perform the same task, seems like an inefficient use of time for practitioners.

A valid question is if there are serious consequences for this divergence. On one hand, perhaps there is little need for a practitioner to understand precisely how a software package does its calculations. If a practitioner understands the theory behind some estimator or statistical test, and can use a well-regarded software package to calculate it, on the surface there seems to be little need for her to implement it herself. There are deeper issues however, which point to more systemic problems in the practice of econometrics.

One consequence is the fact that the work of practitioners is guided by what software is available, instead of what theoretically might make the most sense. It is generally not econometric theory which guides research, but the software that is readily available. As Renfro states:

> 'A danger inherent in these circumstances and frictions may be more a tendency for

> the econometrics theorist to become isolated from the economic practitioner. Obviously, to the degree that practitioners accept the software that is available rather than writing it themselves, having carefully studied the theory, the effect is to impose an intermediary barrier between these two groups [...] Already, both textbooks and software manuals are perceptibly becoming focused on the description of the techniques actually available in the existing econometric software packages - rather than always on the issues of greater present interest to theoreticians.'

A second consequence is the treatment of econometric software as a series 'black boxes'. As stated by Renfo:

> 'It is all too easy for the econometrics software developer to become a technological fan-dancer, offering enticing glimpses amidst smoke and mirrors, yet hiding the essential truth from the spectator. The problem is how to avoid this opacity. It can be avoided if developers make the effort to provide clear descriptions of the characteristics of the displays they provide and if those who teach communicate to their students precisely what the statistics mean.'

The 'black box' problem is compounded as the algorithms become more complicated. Consider the case of the three algorithms described earlier. A researcher simply wishing to employ GETS modelling needs to know very little about the algorithms to actually employ them. A user wishing to generate a sparse model or forecasts can fairly easily use Autometrics, the Lasso and BSTS, doing almost no background research if they have basic skills in OxMetrics and R. The ease with which these algorithms can be applied can be viewed as positive, due to the time it inevitably saves for users. On the other hand it means that, especially when the algorithms are very complicated, there is almost no incentive to understand what the algorithms are actually doing.

Added complexity often means it is nearly impossible to disentangle the theory behind the algorithm, making these algorithms 'super opaque' black boxes in the sense that both the underlying theory and software may not be well understood by the user. Consider BSTS which is the most complicated of the three algorithms described earlier. It is difficult to know, and is at least not well explained in the documentation, what the effect of changing many of the inputs actually is. This makes it very difficult for a user to know whether they are employing the algorithm in the correct manner, and how to interpret results. Implementing Autometrics and the Lasso correctly relies on knowledge of computer software to a lesser extent than the BSTS, and the properties of the results are more easily mapped to the inputs.

Two additional consequences of the 'black box' phenomenon are the disregard and even acceptance of numerical inaccuracy within software packages, and the misinterpretation of output. Most users do not consider numerical accuracy when deciding between different statistical packages. Even econometrics textbooks give the impression that any of the multiple econometrics packages available are acceptable to solving the problem, and as McCullough (1999) states with the ‘implicit and unwarranted assumptions being that the computer’s solution is accurate and that one software package is as good as any other.’ It would likely come as a surprise to many users that, as analyzed extensively by Renfro, different software packages performing the same task, using identical data produced different results. Even if they were more aware of the inaccuracies, most do not have the computer programming background actually do anything about it, and as the field generally seems to accept econometric software as it is, there is no real incentive to do so anyways. Moreover, most users think the onus is on the developers to produce software which produces accurate results. But since numerical accuracy is not a selling feature of econometric software, it is not surprising that the developers have little interest in ensuring it, instead focusing on new features or speed of computation, both of which are important to users. As McCullough points out with multiple examples, numerical inaccuracy can have a big impact on results and findings, and is therefore an issue that needs to be addressed.

Numerical inaccuracy is likely to be even more prevalent in a software package like R. The manner in which R is developed and maintained is different from that of the more traditional econometric packages as it is free and relies on individual packages which can be developed and shared by anyone via the internet. The following quote is taken from R’s official website:

> ‘Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.’

This approach where the testing requirements are minimal leaves further room for computational error. Although in time users find errors, and packages are updated, if the packages are sufficiently complex it is less likely that numerical errors are found and reported simply because most users are likely unsure as to what the results should actually be. The disregard, or even indifference, towards numerical inaccuracy in the field of economics is troublesome and is a symptom of the much greater problem in the field, which is that of economist’s blind faith in the econometric software packages available to them.

A separate but related issue is the interpretation of results. Errors of this nature exist for even the most experienced economist; consider David Hendry’s debate with Milton Friedman where Friedman failed to realize that their software had made a heteroskedasticity correction using a normalization, meaning all their t-values were inflated, which greatly impacted their analysis (Ericsson et al., 2016). Mistakes of these nature are not uncommon even when software and interpretation of results is well documented, and are an even bigger problem where the algorithms are complex and the documentation is vague, imprecise or incomplete.

Given the tendency of economists to treat econometric software as a ‘black box’, model selection algorithms should be as accurate, robust and straightforward as possible with documentation which clearly explains the inputs, the theory behind the algorithm, and what the results actually mean. What the criteria for accurate, robust and straightforward software should be is debatable, but a reasonable question that a user should ask is ‘what is the proper way to use this algorithm’. If the answer is relatively simple and the outcome of changing the user inputs is clear, this should be viewed favourably. If the answer is long and not straightforward, it seems naive to expect users to implement it properly.

As machine learning ideas become more mainstream in the field of economics, there are likely to be increasing numbers of algorithms which make forecasts, select models, etc. in a seemingly automatic way. While there is potential for these algorithms to simplify the job of an econometrician, it is crucial that algorithms are documented and taught in such a way that allows the average econometrician to use them appropriately. This will be necessary until the field begins to shift towards a more data-driven style of research, and when an education in economics also includes a more in depth understanding of how economic software is developed.

Chapter 3

Evaluating Model Selection Algorithms

3.1 The Monte Carlo Technique

Monte Carlo (MC) simulations are a useful tool which provide a framework for evaluating and comparing model selection algorithms. In the context of testing and evaluating model selection algorithms, Monte Carlo simulations involve four main steps as described below.

3.1.1 Formulating the DGP

First, the data generating process (DGP) must be formulated. There are a number of features to consider in the DGP design which will have an impact on the success of the model selection algorithms. Of particular importance in this study is the distribution and non-centrality of the regressors. As will be explained in more detail later, the non-centrality of a particular regressor is the signal-to-noise ratio, and thus is a function of both the coefficient β_k and the variance of a particular regressor. A simple example of a DGP is:

$$y_t = \beta_0 + \delta y_{t-1} + \sum_{k=1}^{n} \beta_k x_{k,t} + \epsilon_t, \qquad \epsilon_t \sim \mathsf{IN}(0,1)$$

$$\mathbf{x}_t \sim \mathsf{IN}_\mathsf{n}(0,\mathsf{I})$$

for $t = 1, ..., T$, where $y_0 = 0$. In practice $H + T$ draws of $\mathbf{x_t}$ are made and the first H are the lead in to the sample for estimation.

3.1.2 Running Simulations

The second step is to perform M simulations for the specified DGP, generating M 'replications' of the dependent variable, $\mathbf{y}_t = (y_1, ..., y_T)$. There are two options for drawing the regressors (excluding the lagged dependent variable y_{t-1}). Either new values of the regressors can be drawn for each of the M simulations (stochastic regressors), or a single draw of the regressors can be used (fixed regressors). In each of the M replications, a new draw of $\epsilon_{\mathbf{t}} = \epsilon_1, ..., \epsilon_T$ is taken.

3.1.3 Formulating the GUM

The third step is to formulate the GUM with the DGP nested within it. The GUM can include lags, nonlinearities, impulse indicators, etc. An example of a GUM associated with the DGP described above is:

$$y_t = \beta_0 + \delta y_{t-1} + \sum_{k=1}^{N} \beta_k x_{k,t} + u_t \qquad u_t \sim \mathsf{IN}[0,1]$$

$$\mathbf{x}_t \sim \mathsf{IN_N}(0, \mathsf{I})$$

where $N \geq n$ and $t = 1, ..., T$. To be explicit, the difference between the DGP and the GUM is that the DGP includes $n+1$ regressors while the GUM includes $N+1$ regressors. That is, $x_{1,t}, ..., x_{n,t}$ in the GUM are exactly the $x_{1,t}, ..., x_{n,t}$ in the DGP. If fixed regressors are used for $x_{1,t}, ..., x_{N,t}$, the GUM does not change over the M simulations. If stochastic regressors are used, each of the M simulations would have different draws for each of the N regressors.

3.1.4 Running the Algorithm

At this point, there are M sets of generated data. The final step is to run the model selection algorithm on each of these sets. The algorithm commences from the GUM, searches through the variables and returns the selected model. For each of the M generated y_ts the algorithm selects the 'best' model, eliminating some of the regressors, and providing parameter estimates for those remaining. Thus, the algorithm will produce M final model estimates: one for each of the M simulations.

3.2 Evaluating Automatic Model Selection

Evaluating automatic model selection algorithms empirically is made difficult by the fact that the DGP is never known. This is why evaluating automatic model selection using Monte Carlo simulations is so useful. Because the DGP is precisely known, it is possible to test the accuracy with which an algorithm selects a model that is close to it. In this section, several metrics which provide a convenient means for evaluating the efficacy of model selection algorithms in Monte

Carlo simulations are introduced and explained.

3.2.1 Gauge and Potency

The gauge and potency provide measurements of the accuracy with which a model selection algorithm excludes irrelevant variables and selects relevant variables (Castle et al., 2011). The gauge is the proportion of the time an algorithm selects variables which are not in the DGP and the potency is the proportion of the time the algorithm selects variables which are in it. The formulas for the gauge and potency are based on another metric called the retention rate. The retention rate is the rate at which a particular regressor in the GUM is selected by the algorithm to be included in the final model. The potency is then the average retention rate across the variables which were part of the DGP. The gauge is the average retention rate across the variables which were not part of the DGP. Say there are M simulations. Variables $x_1, ..., x_n$ are in the DGP, making these the relevant variables. Variables $x_{n+1}, ..., x_N$ are not in the DGP, making these the irrelevant variables. Let $\beta_1, ..., \beta_N$ be the true parameter values, so that $\beta_{n+1}, ..., \beta_N = 0$. Let $\widetilde{\beta}_{k,i}$ be the estimated coefficient on variable x_k in simulation i for each of the N variables following model selection. If a particular variable is not selected in a simulation, $\widetilde{\beta}_{k,i} = 0$. Let $1(\cdot)$ be an indicator variable with $1(\cdot) = 1$ if x_k is selected (so $\widetilde{\beta}_k \neq 0$) and $1(\cdot) = 0$ if x_k is not selected (so $\widetilde{\beta}_k = 0$). The retention rate, potency and gauge are defined as:

$$\begin{aligned}
&\text{retention rate} : \widetilde{p}_k = \frac{1}{M} \sum_{i=1}^{M} 1(\beta_{k,i} \neq 0) \\
&\text{potency} = \frac{1}{n} \sum_{k=1}^{n} \widetilde{p}_k \\
&\text{gauge} = \frac{1}{N-n} \sum_{k=n+1}^{N} \widetilde{p}_k
\end{aligned}$$

Note that usually the intercept is not included in the potency and gauge calculations as it is almost always selected and is often forced. Similarly depending on the nature the DGP the lagged dependent variable is often excluded as well.

3.2.2 Mean Squared Errors

The mean squared errors (MSEs) provide a measure for the accuracy of the parameter estimates. Here, a distinction is made between conditional and unconditional MSEs. If there are M simulations, the conditional MSE for variable x_k is calculated as:

$$\text{CMSE}_k = \frac{\sum_{i=1}^{M} [(\widetilde{\beta}_{k,i} - \beta_k)^2 \cdot 1(\widetilde{\beta}_{k,i} \neq 0)]}{\sum_{i=1}^{M} 1(\widetilde{\beta}_{k,i} \neq 0)}, \; (\beta^2_{k,i} \text{ when } \sum_{i=1}^{M} (1(\widetilde{\beta}_{k,i} \neq 0) = 0)$$

where $\widetilde{\beta}_{k,i}$ is the estimated parameter if x_k is selected. The unconditional MSE is defined as:

$$\text{UMSE}_k = \frac{1}{M}\sum_{i=1}^{M}(\widetilde{\beta}_{k,i} - \beta_k)^2,\ \forall\ k$$

If x_k is not selected, then $\widetilde{\beta}_{k,i} = 0$. In this thesis the square root of the conditional mean squared errors denoted RCMSE is reported. Whereas the CMSE_k averages across simulations in which a particular x_k is selected, UMSE_k takes the average across every simulation. The UMSE is a measure which many authors report, however in this study it is not reported because the CMSE, when interpreted alongside the gauge and potency, is much more informative. For example, say that a retention rate of a particular irrelevant variable x_k is 0.01; that is, it is selected in 10 of 1000 simulations. Then, irrespective of what the parameter estimates in those 10 cases were, the UMSE is going to close to be 0. On the other hand, the CMSE provides a measure of how close (or far) the estimates are to the true parameter ($\beta_k = 0$) in those 10 cases and can actually be quite large. Similarly, consider a relevant variable x_j which has a potency of 0.6, and is not selected in 400 of 1000 simulations. The UMSE for x_j, of course, will include the estimates from those 400 simulations where x_j is not selected, and $\widetilde{\beta}_{j,i} = 0$. For a relevant variable which has a relatively low potency then, the UMSE can end up being quite large, and says little about the accuracy of the parameter estimates themselves. The CMSE on the other hand provides a measure of how effective an algorithm is at actually estimating parameters, which taken alongside the gauge and potency, is much more useful. Additionally, in empirical research only the conditional selected model is known meaning that the properties of selected models are more useful when evaluating selection algorithms. While UMSEs are not reported in this study, they are available upon request.

3.2.3 A Note on Statistical Size

At first glance, the 'size' of each algorithm might seem like an informative metric to include. In statistics, size refers to the probability of a Type 1 error, namely that the null hypothesis is rejected when it is true. In the context of model selection algorithms, size is then the probability that the DGP is not selected, which in simulations corresponds to the proportion of the time that either a relevant variable is excluded or a irrelevant variable is included in the selected model. Under the null that $\beta_k = 0$ the probability of excluding an irrelevant variable is $(1 - \alpha)$. Then, if there are $N - n$ irrelevant variables, the probability that they are all excluded is $(1 - \alpha)^{N-n}$. The probability of retaining a single irrelevant variable is then $1 - (1 - \alpha)^{N-n}$ which could be very high, especially if there are many irrelevant variables. It is for this reason that the size of each procedure is not reported. It seems sensible to accept that while model selection may not always select the precisely correct DGP, there is immense value in excluding 'garbage', a fact which is largely overlooked when statistical size is used. Note that statistical size is very dependent on the number of irrelevant variables $N - n$. However since the GUM includes many variables to ensure that nothing important is missed, it will usually be the case that $N - n$ is sufficiently large to result in a high statistical size.

3.3 The Benchmark

Comparing the gauge, potency and RCMSE across different algorithms tells an interesting story in itself. Another interesting and related story however, is how these results compare to the 'benchmark' or 'best case scenario'. In the context of model selection, the 'best case scenario' means doing model selection under the conditions which offer the 'best chance' for an algorithm to successful identify the DGP. Identifying the benchmark allows the results of model selection to be qualified. What the optimal conditions are depends on the DGP specification, and specifically whether or not the regressors are orthogonal or not. These two cases are considered in turn.

3.3.1 Orthogonal Regressors

Consider first the case of orthogonal regressors, and specifically consider the following DGP:

$$y_t = \beta_0 + \delta y_{t-1} + \sum_{k=1}^{n} \beta_k x_{k,t} + \epsilon_t, \qquad \epsilon_t \sim \mathsf{IN}(0,1)$$
$$\mathbf{x}_t \sim \mathsf{IN}_\mathsf{k}(0, I)$$

for $t = 1, ..., T$. Now suppose the researcher (magically) knew precisely which variables were in the DGP and estimated the following model:

$$y_t = \beta_0 + \delta y_{t-1} + \sum_{k=1}^{n} \beta_k x_{k,t} + u_t$$

This is the 'best case scenario' for finding the DGP; every relevant variable is included in the model being estimated with no irrelevant variables that could be mistakenly selected. In real life, however, it would be impossible to know that modelling was in fact commencing from the DGP and a good econometrician naturally would be interested in conducting inference to determine which variables, from a statistical perspective, were significant. This would generally be done via t-statistics, with high p-values leading the researcher to conclude that those variables were insignificant. The results from conducting this process which is referred to as selection from the DGP are therefore the benchmark when the regressors are orthogonal.

This idea is captured in a model selection technique called the 1-cut approach. Consider the following GUM, with $N << T$:

$$y_t = \beta_0 + \sum_{k=1}^{N} \beta_k x_{k,t} + u_t$$

Because $N << T$, the GUM can be estimated directly, resulting in coefficient estimates and standard errors for $\beta_1, ..., \beta_N$. Then t-statistics are computed and ordered as follows:

$$t^2_{(1)} \geq t^2_{(2)} \geq ... \geq t^2_{(N)}$$

Variables with $t^2_k \geq c^2_\alpha$ are retained. Thus only a single decision is required to select variables; there is no 'data mining' or repeated testing.

The 1-cut approach embodies the approach that a researcher may and usually will undertake in practice (although it is not explicitly referred to as model selection). Therefore, the potency and RCMSE results from the 1-cut approach on the DGP are reported alongside the potency, retention rate and RCMSE results from the three algorithms. This provides some context for the results from the model selection algorithms. For example, if a particular set of simulations shows that a model selection algorithm has a potency of 0.4 and selects relevant variables 40% of the time, critics may be inclined to dismiss the algorithm as ineffective and inaccurate. This is not a valid criticism, however, if when estimating from the DGP itself, relevant variables are deemed insignificant by their t-tests (or equivalently, not selected by 1-cut) a similar proportion of the time. To re-iterate, if in the 'best case' scenario a relevant variable is deemed irrelevant or insignificant, it should not come as surprise or indeed be considered a 'drawback' if this variable is not selected by a model selection algorithm, amongst many other variables. Model selection algorithms should not lose credibility if they are unsuccessful at finding 'significant variables' which are not even 'significant' when the model is estimated from the DGP. It should also be noted that even when the regressors are orthogonal, there will be some amount of correlation between the regressors for any given sample, meaning that algorithms have the potential to outperform the 1-cut approach, which is explained in more detail below.

3.3.2 Correlated Regressors

When the regressors are correlated, the 1-cut approach is not valid. The best hope a researcher would have at finding variables which matter in this case would be to use a model selection algorithm which is equipped to deal with correlation (which Autometrics, the Lasso, and BSTS all claim to be) on the DGP itself. Because each of the algorithms employ different approaches to model selection, and the question over which one is the 'best' is subjective, the benchmark for a particular algorithm are the results from using that algorithm to select from the DGP. To make this more formal, consider the following DGP:

$$y_t = \beta_0 + \delta y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$

$$\epsilon_t \sim \text{IN}[0,1]$$

$$\mathbf{x}_t \sim \text{IN}_\text{N}[0,\Omega]$$

with $\omega_{kk} = 1$ and $\omega_{jk} = 0.9$ for $j \neq k$ for $t = 1, ..., T$. Now suppose the researcher knew which variables were relevant and used the following GUM:

$$y_t = \beta_0 + \delta y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + u_t$$

The benchmark for an algorithm searching through a much larger GUM would be the potency and RCMSE results from commencing from the above GUM, which is the true DGP. When regressors are correlated, it is precisely these conditions which allow the algorithm the best chance at selecting the correct model.

3.4 Interpreting the Results

The results from the Monte Carlo simulations that are presented in the next section provide insight into three questions: (1) how algorithms perform relative to one another given a particular correlation structure, (2) how algorithms perform relative to their respective benchmarks and (3) more broadly how the results of empirical model selection using Autometrics, the Lasso and BSTS can be interpreted. The next section details how the gauge, potency, retention rates, RCMSE and benchmark results provide insight into these questions.

3.4.1 Algorithm vs. Algorithm

The gauge, potency and RCMSE from selection from the GUM make it easy to compare the algorithm's performance within a particular correlation structure. High potencies, low gauges and low RCMSEs are all desirable in the MC simulations.

3.4.2 Algorithm vs. Benchmark

Evaluating the algorithm's results against their benchmark by comparing the potencies, retention rates and RCMSEs in each case is straightforward. If the results from model selection are similar to the benchmark then the algorithm is performing as well as could optimistically be expected. A productive framework for thinking about this is considering the costs of inference and the costs of search. Costs of inference are inevitable even for a modeller who commenced from the DGP but (as is always the case in economics) did not know if the specification was correct and thus conducted inference. Similarly, there are obviously costs of inference when using automatic model selection on a GUM of any size. These costs can be measured with the RCMSE values. Costs of search are the additional costs, and are the result of commencing from a more general model. The difference between the RCMSE when commencing from the GUM and the RCMSE when commencing from the DGP (either using 1-cut or using model selection directly on the DGP) measures the costs of search for a particular model selection algorithm. Thus, the benchmark DGP results reported in the results tables are useful because they provide context for the cost of inference when performing model selection (via the RCMSEs), and also provide a measure of the cost of model selection (via the RCMSE differences).

3.4.3 Simulation Results vs. Empirical Model Selection

MC simulations provide lots of interesting results but they are just that: simulations. A valid question is what MC results say about model selection empirically. The difficulty with MC simulations is that their results are generally limited to the specified DGP. That is, the potency and gauge measures for a particular set of simulations would only be directly relevant in a real world situation where the DGP has the exact same specification as in the MC simulation. As has been discussed, this is an impossible problem. A method is therefore required, which allows the results of MC simulations to be 'mapped' to empirical model selection. This is mostly easily done with the use of the non-centrality parameter.

Intuitively, regressors which are 'highly significant' should be 'picked up' more frequently than those which are less significant. A measure which provides a useful interpretation of the 'significance' of a variable x_k is the non-centrality of x_k, denoted ψ_k. The non-centrality is the 'signal-to-noise' ratio, where the signal is the value of the parameter β_k and the noise is σ_{β_k}. The non-centrality is therefore just the expectation of the t-statistic for variable x_k:

$$t_{\widehat{\beta}_k} = \frac{\widehat{\beta}_k}{\widehat{\sigma}_{\widehat{\beta}_k}} \simeq \frac{\widehat{\beta}_k}{\sigma_{\widehat{\beta}_k}} \sim \mathsf{N}\left[\frac{\beta_k}{\sigma_{\widehat{\beta}_k}}, 1\right] = \mathsf{N}\left[\psi_k, 1\right]$$

Therefore, the formula for ψ_k is given by:

$$\psi_k = \frac{\beta_k}{\sigma_{\widehat{\beta}_k}}$$

The above formula makes explicit that relationship between ψ_k, the coefficient β_k and the variance $\sigma^2_{\widehat{\beta}_k}$. The non-centrality is useful in understanding the relationship between the specification of the DGP and the potency or retention rates of particular variables. Consider, as discussed earlier, estimating a model directly from the DGP, and obtaining the t-statistics for each of the relevant variables. Because the DGP is known, it is possible to derive the distribution of the t-statistics and as the above formulas suggest, the non-centrality of a variable characterizes this distribution. Knowing the t-statistic distribution means it is possible to determine the probability that a variable is deemed significant, or more formally, the probability that the null $H_0 : \beta_k = 0$ is rejected for a given significance level α. This probability is called the theoretical retention probability, and is calculated from:

$$\mathsf{P}_\alpha = \Pr(\ |t_k| \ \geq \ c_\alpha \mid \psi_k\ ) = \Phi(c_\alpha - \psi_k)$$

where $\Phi(x)$ denotes the integral of the normal density. It is possible to calculate the theoretical retention probability for every non-centrality and significance level. This is important because as it turns out certain algorithms have retention rates which are in line with the calculated theoretical retention probabilities.

3.5 Experimental design

The experimental design consists of three separate sets of experiments. In the first, the regressors are all orthogonal, in the second there is correlation between the relevant regressors, and in the third there is correlation between all regressors. In all three sets of experiments the algorithm settings remain constant; these settings are outlined in the next section. Furthermore, the DGPs and more specifically the values of the non-centralities ψ_k are chosen in such a way that the results are comparable across the three separate sets of experiments. In the following sections, the DGPs and GUMs are described and the results are presented for each of the three sets of experiments. The final section provides a summary of results.

3.5.1 Algorithm Settings

As discussed in Chapter 2, there are a number of decisions that must be made when applying the algorithms. The following table outlines the settings for the three algorithms compared, as well as the relevant software that was used for the simulations.

Algorithm	**Software/Relevant package**	**User input**	**Comment**
Autometrics	OxMetrics version 7.00	$\alpha = 0.01$	All other defaults used, as described in Table 2.1
Lasso	R version 3.2.3, glmnet package	λ chosen by 10-fold cross validation	K-fold cross validation built into glmnet
BSTS	R version 3.2.3, bsts package	$R^2 = 0.5$, and $v = 0.01$	All default priors used

Table 3.1: Algorithm settings

An additional setting for BSTS was the decision to consider variables with posterior inclusion probability greater than 0.15 as selected. While this seems low, it was rare to observe posterior inclusion probabilities greater than 0.2.

3.5.2 Orthogonal Regressors

The first set of experiments considers cases where the regressors, with the exception of the lagged dependent variable, are mutually orthogonal. There are three different DGPs considered, which vary according to the coefficients $\beta_1, ..., \beta_5$. The values of these coefficients affect the non-centralities ψ, which is a more interpretable characterization. The β_0 and δ coefficients are the same across all experiments, with $\beta_0 = 1$ and $\delta = 0.5$. Each of these DGPs is nested in two separate GUMs: one with the total number of regressors N, excluding the lagged dependent variable (LDV), as $N = 80$ and the second with $N = 120$. In all experiments, $T = 100$. Thus, there are six separate experiments performed for the case of orthogonal regressors. In all experiments, fixed regressors were used. Note that in practice, in any given sample there will be degree of correlation between the simulated $x_1, ..., x_N$, and that when $N >> T$ orthogonality is in fact impossible. Nevertheless, variables can be generated so that correlation between them is small. Let $\mathbf{x}_t' = (x_{1,t}, ..., x_{N,t})$. The DGPs take the following form:

$$y_t = \beta_0 + \delta y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$

$$\epsilon_t \sim \mathsf{IN}[0, 1]$$

$$\mathbf{x}_t = 0.5\mathbf{x}_{t-1} + \mathbf{v}_t, \mathbf{v}_t \sim \mathsf{IN}_N[0, \Omega]$$

with the persistence parameter $\rho = 0.5$, $\omega_{kk} = 1 - \rho^2 = 0.75$ and $\omega_{jk} = 0$ for $j \neq k$. While this may seem like an arbitrary choice for the variance-covariance matrix Ω, it has been chosen because the resulting non-centralities are integers and in line with previous studies. Note that the results that follow break down as $\rho \to 1$, but since $\rho = 0.5$ here it is not a issue. Table 3.2 describes the three DGPs considered in this set of experiments, and gives the theoretical retention probabilities $\mathsf{P}_{0.01}$. Because the regressors are orthogonal in this set of experiments, the benchmark is model selection from the DGP via the 1-cut approach.

		x_1	x_2	x_3	x_4	x_5
DGP 1	β_k	0.6	0.6	0.6	0.6	0.6
	ψ	6	6	6	6	6
	$\mathsf{P}_{0.01}$	0.999	0.999	0.999	0.999	0.999
DGP 2	β_k	0.2	0.2	0.2	0.2	0.2
	ψ	2	2	2	2	2
	$\mathsf{P}_{0.01}$	0.266	0.266	0.266	0.266	0.266
DGP 3	β_k	0.2	0.3	0.4	0.5	0.6
	ψ	2	3	4	5	6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999

Table 3.2: DGP specification for experiments with orthogonal x_ks

Each DGP is nested in two separate GUMs, one with $N = 80$, and one with $N = 120$. These two GUMs take the following form:

$$y_t = \beta_0 + \delta y_{t-1} + \sum_{k=1}^{N} \beta_k x_{k,t} + u_t$$

Tables 3.3 and 3.4 report the gauge and potency for the DGP 1 and DGP 2 respectively, while Table 3.5 and Table 3.6 report their RCMSE results. The lowest gauge, highest potency and lowest RCMSE are in bold. Figures 3.1 and 3.2 graph the RCMSE results. Tables 3.7-3.8 report the results for DGP 3 and Figures 3.3 and 3.2 graph these results. There are varying non-centralities in DGP 3 which is why Table 3.8 reports the retention rates and the theoretical retention probabilities for each individual relevant regressor.

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.994	0.013	0.991	0.013
Lasso	**1.000**	0.216	**0.997**	0.174
BSTS	0.715	**0.001**	0.757	**0.000**
DGP	0.999	N/A	0.999	N/A

Table 3.3: Gauge and potency for DGP 1

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.320	0.029	0.263	0.029
Lasso	**0.526**	0.146	**0.428**	0.125
BSTS	0.022	**0.001**	0.023	**0.001**
DGP	0.278	N/A	0.278	N/A

Table 3.4: Gauge and potency for DGP 2

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β_k	0.5	0.6	0.6	0.6	0.6	0.6
$\boldsymbol{N = 80}$	**Autometrics**	**0.057**	**0.112**	**0.127**	**0.099**	**0.105**	**0.111**
	Lasso	0.085	0.201	0.196	0.183	0.173	0.201
	BSTS	0.082	0.274	0.259	0.232	0.235	0.238
	DGP	0.055	0.097	0.102	0.110	0.105	0.108
$\boldsymbol{N = 120}$	**Autometrics**	**0.068**	**0.106**	**0.113**	**0.129**	**0.121**	**0.107**
	Lasso	0.124	0.165	0.204	0.285	0.215	0.239
	BSTS	0.105	0.186	0.262	0.273	0.241	0.211
	DGP	0.055	0.097	0.102	0.110	0.105	0.108

Table 3.5: RCMSE for DGP 1

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β	0.5	0.2	0.2	0.2	0.2	0.2
$\boldsymbol{N = 80}$	**Autometrics**	**0.120**	0.164	0.221	0.124	0.142	0.151
	Lasso	0.218	**0.129**	0.137	0.128	**0.127**	**0.133**
	BSTS	0.153	0.216	**0.125**	**0.119**	0.141	0.221
	DGP	0.091	0.119	0.126	0.161	0.146	0.157
$\boldsymbol{N = 120}$	**Autometrics**	**0.139**	0.140	0.162	0.210	0.180	0.146
	Lasso	0.258	**0.123**	**0.130**	0.139	**0.131**	0.132
	BSTS	0.191	0.182	0.180	**0.084**	0.159	**0.131**
	DGP	0.091	0.119	0.126	0.161	0.146	0.157

Table 3.6: RCMSE for DGP 2

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.712	0.018	0.708	0.021
Lasso	**0.877**	0.179	**0.835**	0.146
BSTS	0.361	**0.001**	0.334	**0.001**
DGP	0.766	N/A	0.766	N/A

Table 3.7: Gauge and potency for DGP 3

	$\boldsymbol{\psi}$	2	3	4	5	6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999
$\boldsymbol{N = 80}$	**Autometrics**	0.216	0.414	0.948	0.989	0.995
	Lasso	**0.552**	**0.869**	**0.974**	**0.995**	**0.996**
	BSTS	0.003	0.136	0.325	0.562	0.781
	DGP	0.331	0.672	0.847	0.982	1.000
$\boldsymbol{N = 120}$	**Autometrics**	0.367	0.573	0.655	0.953	0.992
	Lasso	**0.669**	**0.779**	**0.740**	**0.989**	**0.997**
	BSTS	0.059	0.036	0.092	0.602	0.880
	DGP	0.331	0.672	0.847	0.982	1.000

Table 3.8: Retention rates for DGP 3

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β_k	0.5	0.2	0.3	0.4	0.5	0.6
$\boldsymbol{N = 80}$	**Autometrics**	**0.077**	0.153	**0.143**	**0.097**	**0.112**	**0.117**
	Lasso	0.119	0.130	0.171	0.200	0.196	0.223
	BSTS	0.114	**0.115**	0.181	0.192	0.226	0.238
	DGP	0.070	0.120	0.090	0.097	0.101	0.110
$\boldsymbol{N = 120}$	**Autometrics**	**0.103**	0.143	**0.111**	**0.133**	**0.127**	**0.112**
	Lasso	0.180	**0.119**	0.181	0.252	0.218	0.258
	BSTS	0.131	0.172	0.177	0.216	0.239	0.227
	DGP	0.070	0.120	0.090	0.097	0.101	0.110

Table 3.9: RCMSE for DGP 3

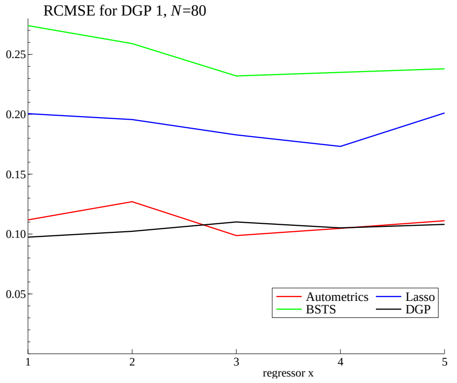


Figure 3.1: RCMSE for DGP 1

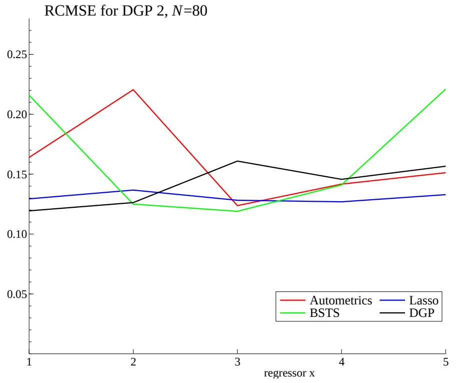


Figure 3.2: RCMSE for DGP 2

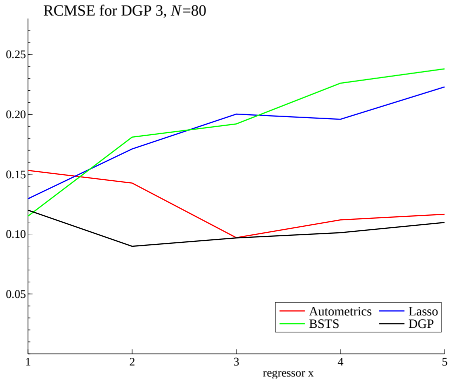


Figure 3.3: RCMSE for DGP 3

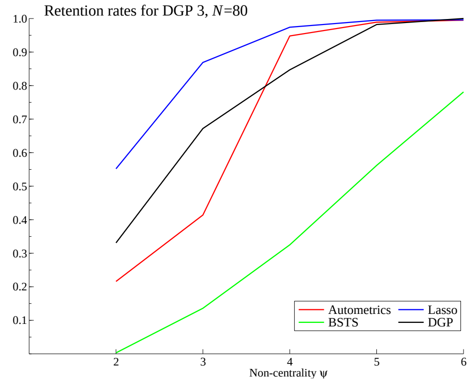


Figure 3.4: Retention rates for DGP 3

The above results show the Lasso consistently achieves the highest potency and has the highest retention rates across all levels of ψ but also has by far the highest gauge. Autometrics selects relevant regressors at a level which is generally similar to the theoretical retention probability (with a couple of exceptions) and does nearly as well as the Lasso when $\psi > 4$. In comparison to Lasso and Autometrics, BSTS does not fare well when selecting relevant variables, even for variables with high values of ψ. Retention rates and potency measures do not change drastically between $N < T$ and $N > T$ when $\psi > 4$, but results for lower values of ψ vary quite significantly.

BSTS consistently achieves the lowest gauge but its low potency and retention rates indicate that it tends to select very sparse models in general. Note that when conducting the experiments, various priors were tested but this did not seem to have an impact on the results for BSTS. The gauge for Lasso is generally very high and in the range of 0.12 to 0.22, which along with the high levels of potency indicate that it selects models which include a lot of variables. In all of the six experiments, Autometrics has a gauge close to the significance level $\alpha = 0.01$, however in DGP 2 with $\psi = 2$, the gauge is slightly higher at 0.029 probably because it misses some relevant variables with such low non-centralities. It could also possibly correspond to a 'bad draw' due to the fact that fixed regressors are used.

The RCMSE results are most easily interpreted by the graphs, which show that the results vary depending on values of ψ. As can be seen in Figures 3.1 and 3.3, Autometrics performs well for high values of ψ. Figure 3.3 shows that Lasso and BSTS increasingly struggle as ψ rises. Results are mixed in Figure 3.2, where there is a low non-centrality of $\psi = 2$. Note that the spike in RCMSE for x_2 in Autometrics could be due to the fact that fixed regressors are used.

Both the Lasso and Autometrics have results similar to, or better than, the benchmark when examining the potency and retention rates. As mentioned, across the board, the Lasso has high potencies and high gauges, meaning that overall there is less 'selection' going on. It is a different story when examining the RCMSE results however; Autometrics is much closer to the benchmark (except in the case where $\psi = 2$ where results are varied). This means the costs of inference are high in Lasso and BSTS.

The five biggest takeaways from this set of experiments are:

1. Autometrics consistently has a gauge close to the significance level and retention rates/potencies close to the theoretical retention probabilities.

2. The Lasso scores high for successfully retaining relevant variables, but taken next to the gauges, this is less impressive.

3. BSTS selects models which are very sparse, which is seen clearly through its extremely low measures for gauge, and relatively low levels of potency.

4. The costs of inference vary with ψ, but generally are quite low with Autometrics, but high for both the Lasso and BSTS.

5. The huge differences in gauge between the Lasso and BSTS make comparisons using just

these two measures impossible. It is infeasible to derive a 'gauge-corrected' potency that would make comparisons meaningful.

3.5.3 Correlation Between Relevant Regressors

The second set of experiments considers several cases where there is correlation between the relevant regressors. There are two different DGPs considered, which again vary according to their coefficients $\beta_1, ..., \beta_5$ and therefore their respective non-centralities. In both DGPs $\beta_0 = 1$ and $\delta = 0.5$. $\beta_1, ..., \beta_5$ have been chosen so that the non-centralities in this set of experiments are in line with those in the first set of experiments, making the results comparable. In all experiments fixed regressors are used. As in the first set of experiments, the two DGPs are nested in two different GUMs with $N = 80$ and $N = 120$. In all experiments, $T = 100$. Thus, there are four separate experiments performed and reported on. Let $\mathbf{x}_t' = (x_{1,t}, ..., x_{N,t})$. The DGPs take the following form:

$$y_t = \beta_0 + \delta y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$

$$\epsilon_t \sim \mathsf{IN}[0, 1]$$

$$\mathbf{x}_t \sim \mathsf{IN}_N[0, \Omega]$$

with $\omega_{kk} = 1$, $\omega_{jk} = 0.9$ for $j \neq k$, $j, k \leq 5$ and $\omega_{jk} = 0$ elsewhere. Due to the correlation between regressors, it is difficult to derive a formula which analytically gives the relationship between ψ_k, $\sigma_{\widehat{\beta}_k}$, and t_{β_k}. However, ψ_k can also be expressed as $\mathsf{E}[t_{\widehat{\beta}_k}]$. Thus the β_ks in this set of experiment are set such that the average t-statistic for variable x_k across simulations is equal to the desired ψ. More explicitly:

$$\psi_k = \mathsf{E}[t_{\widehat{\beta}_k}] = \frac{1}{1000} \sum_{i=1}^{1000} t_{\widehat{\beta}_k, i}$$

where $t_{\widehat{\beta}_k, i}$ is the calculated t-statistic for $\widehat{\beta}_k$ in simulation i. Table 3.10 describes the two DGPs considered.

		x_1	x_2	x_3	x_4	x_5
DGP 4	β_k	0.58	0.84	1.25	1.55	1.75
	ψ_k	2	3	4	5	6
DGP 5	β_k	-0.58	0.84	-1.25	1.55	-1.75
	ψ_k	-2	3	-4	5	-6

Table 3.10: DGP specification for experiments with correlation between relevant regressors

When the regressors are not orthogonal, as explained, simply using 1-cut selection on the DGP is not a feasible baseline. The 'best case scenario' then is doing selection on the DGP using the algorithms themselves. These results capture how effective the algorithm is in the case where the GUM includes exactly the correct regressors. The difference between the RCMSE from selection on the DGP and selection from the GUM is therefore a measure of the costs of inference.

Tables 3.11 and 3.12 report the gauge, potency and retention rates for DGP 4, and include results when selecting from the DGP for each algorithm. Tables 3.13 and 3.14 do the same for DGP 5. Tables 3.15 and 3.16 report RCMSE results. The lowest gauge, highest potency, and lowest RCMSE are in bold. Figures 3.5-3.8 graph the results. Figures 3.9 and 3.10 graph the differences between RCMSE for each algorithm and their base line, and thus give an indication of the costs of inference for each of the algorithms.

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.777	0.014	0.769	0.013
from DGP	0.797		0.797	
Lasso	**0.993**	0.108	**0.993**	0.077
from DGP	0.994		0.994	
BSTS	0.690	**0.000**	0.677	**0.000**
from DGP	0.784		0.784	

Table 3.11: Gauge and potency for DGP 4

	$\boldsymbol{\psi}$	2	3	4	5	6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999
$\boldsymbol{N = 80}$	**Autometrics**	0.333	0.568	0.988	0.998	0.998
	Lasso	**0.977**	**0.990**	**1.000**	**1.000**	**1.000**
	BSTS	0.179	0.340	0.949	0.993	0.991
$\boldsymbol{N = 120}$	**Autometrics**	0.316	0.611	0.923	0.996	0.999
	Lasso	**0.968**	**0.998**	**1.000**	**1.000**	**1.000**
	BSTS	0.155	0.369	0.874	0.988	0.999
From DGP	**Autometrics**	0.330	0.723	0.934	0.996	1.000
	Lasso	0.979	0.993	1.000	1.000	1.000
	BSTS	0.344	0.589	0.990	0.999	0.998

Table 3.12: Retention rates for DGP 4, including from DGP

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	**0.716**	0.014	**0.719**	0.014
from DGP	0.775		0.775	
Lasso	0.429	0.088	0.374	0.057
from DGP	0.986		0.986	
BSTS	0.531	**0.000**	0.439	**0.000**
from DGP	0.675		0.675	

Table 3.13: Gauge and potency for DGP 5

	$\boldsymbol{\psi}$	-2	3	-4	5	-6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999
$\boldsymbol{N = 80}$	**Autometrics**	**0.255**	**0.355**	0.981	**0.996**	0.994
	Lasso	0.065	0.013	**0.984**	0.095	**0.988**
	BSTS	0.022	0.056	0.798	0.827	0.951
$\boldsymbol{N = 120}$	**Autometrics**	**0.290**	**0.482**	**0.841**	**0.984**	**1.000**
	Lasso	0.254	0.007	0.590	0.019	0.998
	BSTS	0.054	0.061	0.353	0.727	0.999
From DGP	**Autometrics**	0.336	0.631	0.925	0.981	1.000
	Lasso	0.967	0.963	1.000	1.000	1.000
	BSTS	0.150	0.267	0.974	0.989	0.995

Table 3.14: Retention Rates for DGP 5, including from DGP

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β_k	0.5	0.58	0.84	1.25	1.55	1.75
$\boldsymbol{N = 80}$	**Autometrics**	**0.017**	0.438	0.423	0.277	0.377	0.377
	Lasso	0.038	**0.279**	**0.339**	**0.259**	**0.292**	**0.307**
	BSTS	0.024	0.441	0.572	0.448	0.624	0.541
$\boldsymbol{N = 120}$	**Autometrics**	**0.016**	0.432	0.332	0.431	0.405	**0.342**
	Lasso	0.038	**0.272**	**0.314**	**0.333**	**0.313**	0.354
	BSTS	0.020	0.530	0.555	0.606	0.626	0.499
From DGP	**Autometrics**	0.016	0.442	0.317	0.366	0.364	0.326
	Lasso	0.021	0.266	0.329	0.255	0.286	0.297
	BSTS	0.026	0.395	0.504	0.375	0.484	0.432

Table 3.15: RCMSE for DGP 4, including baseline

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β_k	0.5	-0.58	0.84	-1.25	1.55	-1.75
$\boldsymbol{N = 80}$	**Autometrics**	0.334	**0.281**	**0.316**	**0.287**	**0.325**	**0.258**
	Lasso	0.120	0.463	0.656	0.778	1.074	1.225
	BSTS	**0.078**	0.288	0.527	0.446	0.572	0.571
$\boldsymbol{N = 120}$	**Autometrics**	0.273	**0.292**	**0.303**	**0.296**	**0.341**	**0.257**
	Lasso	0.226	0.424	0.554	1.030	1.029	0.955
	BSTS	**0.097**	0.405	0.510	0.603	0.722	0.496
From DGP	**Autometrics**	0.051	0.342	0.239	0.330	0.334	0.330
	Lasso	0.052	0.286	0.362	0.266	0.307	0.326
	BSTS	0.071	0.342	0.482	0.384	0.411	0.408

Table 3.16: RCMSE for DGP 5, including from DGP

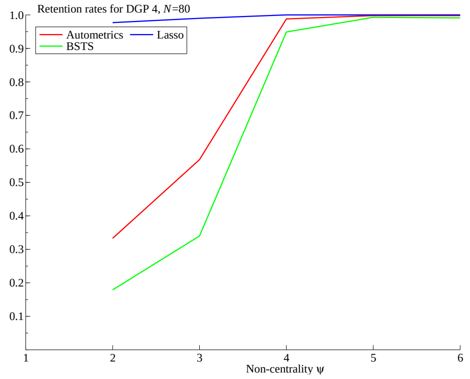


Figure 3.5: Retention rates DGP 4
$N = 80$

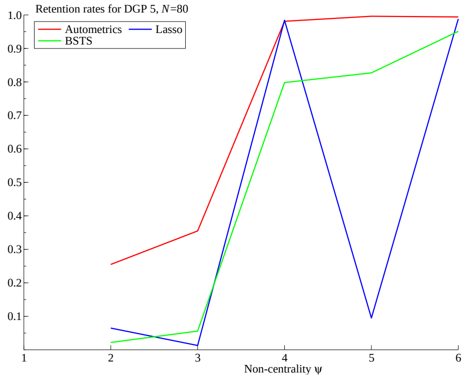


Figure 3.6: Retention Rates DGP 5
$N = 80$

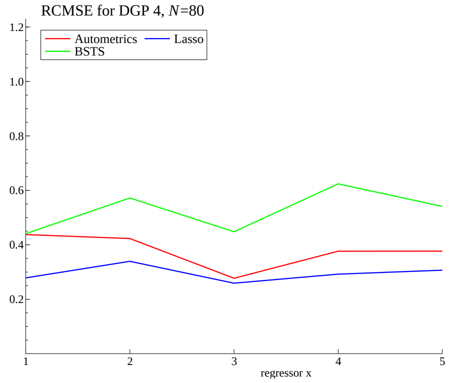


Figure 3.7: RCMSE for DGP 4
$N = 80$

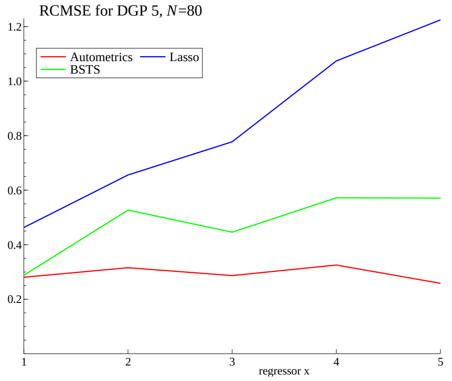


Figure 3.8: RCMSE for DGP 5
$N = 80$

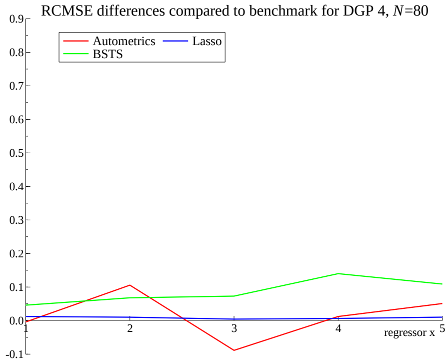


Figure 3.9: RCMSE Differences for DGP 4
$N = 80$

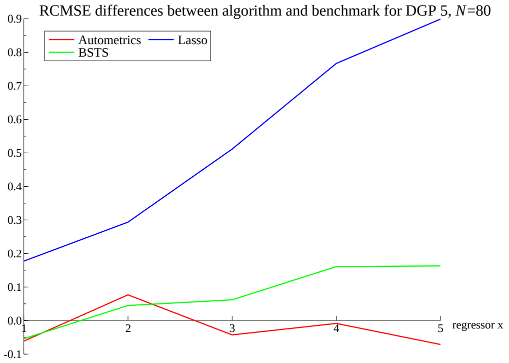


Figure 3.10: RCMSE Differences for DGP 5
$N = 80$

When considering DGP 4 results, when $\psi_k > 0, \forall k$, the Lasso has both the highest measures of potency and gauge. BSTS has the lowest gauge and Autometric's gauge is close to the significance level α. Going from $N < T$ to $N > T$ in most cases does not have a big impact on the results. Both Autometrics and the Lasso have retention rates similar to their benchmark, while the benchmark for BSTS gives quite different results. The graph for the RCMSE differences is interesting, and shows there are almost no costs of search for the Lasso, and for Autometrics there are in fact negative costs of search, meaning that sometimes parameter estimates are even more accurate than in the theoretical best case scenario. This is likely due to adventitiously selecting irrelevant variables which partly proxy a relevant variable which has not been selected, which results in a smaller error variance.

When looking at DGP 5, it is clear that the algorithms struggle to varying degrees with alternating signs. This is particularly true for the Lasso, with retention rates on regressors with $\psi > 0$ being very low, irrespective of their magnitude, when the largest ψ in the experiment is negative. The search costs of the Lasso also increase with $|\psi|$. The results for the gauge are similar to previous experiments; the gauge for BSTS is low in every experiment, the gauge for Autometrics is slightly higher than the significance level α, and the gauge for the Lasso varies across the experiments. The costs of search for both Autometrics and BSTS, as measured by the RCMSE differences, are quite low.

Comparing the results from DGP 4 to DGP 5 is striking. While intuitively and theoretically the DGPs have nearly identical properties, the actual results vary drastically. Thinking about non-centrality as the signal-to-noise ratio, whether a variable has a positive or negative non-centrality should not have an impact on how often it is selected by an algorithm; it is the magnitude of the the signal-to-noise ratio, $|\psi|$, which should influence an algorithm's ability to select that variable. Indeed, in the case of orthogonal variables alternating signs did not result in potency and retention rate measures which were notably different from cases where the signs were all the same.

It is immediately clear however that something peculiar is going on with DGP 5 when the signs are alternating. DGP 4 has retention rates as expected are increasing with ψ. In DGP 5 while both Autometrics and BSTS have higher retention rates as $|\psi|$ rises, the Lasso struggles to select either of the regressors with $\psi > 0$. It should also be noted that while the retention rates do increase with $|\psi|$ for Autometrics and BSTS, the increase is not at all 'uniform' and both Autometrics and BSTS are more successful at selecting variables with $\psi < 0$. The RCMSE results illustrate the parameter estimates also vary across the DGPs. In DGP 4, the Lasso has the lowest RCMSE for all variables, while it has the highest RCMSEs for all variables in DGP 5. Furthermore, the RCMSEs increase with ψ for the Lasso with alternating signs.

The reason why the Lasso has such difficulty selecting the regressors with negative signs is due to the nature of the search its algorithm uses to identify non-zero coefficients. The algorithm identifies the regressor which is 'most significant' and shrinks the coefficients of regressors which are highly correlated with the identified regressor to zero. Certain implementations of the Lasso have built in features to account for this when the coefficients of the correlated regressors have the same sign (i.e. see Hastie and Zou (2005)) but no algorithm exists to account for regressors which have coefficients with opposite signs. In the case of DGP 5, the Lasso (correctly) identifies x_5 with $\psi_5 = -6$ as the most significant, and since x_5 is highly correlated with $x_1, ..., x_4$, it shrinks their coefficients to zero. Because the Lasso allows for variables which are significant 'in the same direction' (i.e. in this case negatively significant), to be effectively 're-added', variables x_1 and x_3 which have respective non-centralities of -2 and -4 are selected, but there is no mechanism for x_2 and x_4 with their positive coefficients to be 'reconsidered' for selection. If it was the case that the most significant variable had a positive coefficient, then any variables with negative coefficients would be shrunk to zero and not selected. That is, negative coefficients and positive correlations are isomorphic to positive coefficients and negative correlations, so this result would also hold if it parameter signs were switched.

The biggest takeaways from this second set of experiments are:

1. Selection when signs alternate produces remarkably different results, especially for the Lasso.

2. When $\psi_k > 0 \; \forall \; k$ the Lasso selects a lot of both relevant and irrelevant variables, BSTS does not select many of either, and Autometrics is somewhere in the middle.

3. In terms of their RCMSEs, the behaviour of Autometrics commencing from the GUM is relatively similar to starting from the DGP. The retention rates for relevant variables in both these cases are close to the theory-derived retention probabilities.

4. Remarkably, across all experiments, there is little difference in the performance of Autometrics between $N < T$ and $N > T$ ($N = 80$ and $N = 120$).

5. As found in the previous set of experiments, search costs for Autometrics can sometimes be negative.

6. The RCMSEs for Autometrics are relatively constant across all ψ and for the Lasso and BSTS they are not.

3.5.4 Correlation Between all Regressors

The third set of experiments considers several cases where there is correlation between all regressors. There are two different DGPs considered, which again vary according to the coefficients $\beta_1, ..., \beta_5$ and therefore their respective non-centralities. Fixed regressors are used in all experiments. These two DGPs are nested, as previously, in two different GUMs with $N = 80$ and $N = 120$. In all experiments, $T = 100$. There are four separate experiments performed and reported on. Let $\mathbf{x}'_t = (x_{1,t}, ..., x_{N,t})$. The DGPs take the following form:

$$y_t = \beta_0 + \delta y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$

$$\epsilon_t \sim \mathsf{IN}[0,1]$$

$$\mathbf{x}_t \sim \mathsf{IN}_N[0,\Omega]$$

with $\omega_{kk} = 1$, $\omega_{jk} = 0.8$ for $k \neq j$. The biggest difference between this set of experiments and previous set is that here Ω is a full $N \times N$ matrix. As with the second set of experiments, determining the relationship between ψ_k, β_k and $\sigma_{\widehat{\beta}_k}$ is difficult to do analytically because of the correlation structure. Thus $\beta_1, ..., \beta_5$ in this set of experiment are set such that the average t-statistic for variable x_k across simulations aligns with the non-centralities in the previous experiments. More explicitly:

$$\psi_k = \mathsf{E}[t_{\widehat{\beta}_k}] = \frac{1}{1000}\sum_{i=1}^{1000} t_{\widehat{\beta}_k,i}$$

where $t_{\widehat{\beta}_k,i}$ is the calculated t-statistic for $\widehat{\beta}_k$ in simulation i. Table 3.17 describes the two DGPs considered, with the β_ks that correspond to the desired ψ_ks.

The following table describes the two DGPs considered, from hereon referred to as DGP 6 and 7:

		x_1	x_2	x_3	x_4	x_5
DGP 6	β_k	0.4	0.6	0.9	1.1	1.25
	ψ_k	2	3	4	5	6
DGP 7	β_k	-0.4	0.6	-0.9	1.1	-1.25
	ψ_k	-2	3	-4	5	-6

Table 3.17: DGP specification for experiments with correlation between all regressors

The following tables report the potency, gauge, retention rates and RCMSE for DGP 6 and DGP 7. The lowest gauge, highest potency and lowest RCMSE are in bold. Again in this set of experiments the significance level for Autometrics was set to $\alpha = 0.01$. The results are also graphed.

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.736	0.016	0.718	0.015
from DGP	0.798		0.798	
Lasso	**0.909**	0.182	**0.894**	0.131
from DGP	0.993		0.993	
BSTS	0.653	**0.004**	0.631	**0.005**
from DGP	0.764		0.764	

Table 3.18: Gauge and potency for DGP 6

	$\boldsymbol{\psi}$	2	3	4	5	6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999
$\boldsymbol{N = 80}$	**Autometrics**	0.262	0.439	0.985	0.996	0.996
	Lasso	**0.766**	**0.778**	**1.000**	**1.000**	**1.000**
	BSTS	0.116	0.251	0.923	0.984	0.990
$\boldsymbol{N = 120}$	**Autometrics**	0.212	0.524	0.890	0.966	0.999
	Lasso	**0.560**	**0.924**	**0.991**	**0.998**	**0.999**
	BSTS	0.093	0.285	0.810	0.971	0.997
From DGP	**Autometrics**	0.327	0.729	0.938	0.995	1.000
	Lasso	0.969	0.994	1.000	1.000	1.000
	BSTS	0.281	0.554	0.990	0.998	0.999

Table 3.19: Retention Rates for DGP 6

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	δ/β_k	0.5	0.4	0.6	0.9	1.1	1.25
$\boldsymbol{N = 80}$	**Autometrics**	**0.025**	0.318	**0.313**	**0.190**	**0.241**	**0.250**
	Lasso	0.041	**0.230**	0.347	0.257	0.349	0.315
	BSTS	0.036	0.318	0.404	0.386	0.386	0.348
$\boldsymbol{N = 120}$	**Autometrics**	**0.023**	0.329	**0.227**	**0.263**	**0.254**	**0.232**
	Lasso	0.033	**0.254**	0.313	0.376	0.462	0.389
	BSTS	0.026	0.306	0.370	0.430	0.428	0.346
From DGP	**Autometrics**	0.024	0.307	0.219	0.259	0.255	0.230
	Lasso	0.029	0.192	0.234	0.182	0.204	0.211
	BSTS	0.036	0.275	0.361	0.258	0.337	0.314

Table 3.20: RCMSE for DGP 6

	$N = 80$		$N = 120$	
	Potency	Gauge	Potency	Gauge
Autometrics	0.718	0.014	0.707	0.014
from DGP	0.778		0.778	
Lasso	**0.787**	0.154	**0.823**	0.131
from DGP	0.986		0.986	
BSTS	0.512	**0.001**	0.401	**0.001**
from DGP	0.657		-0.657	

Table 3.21: Gauge and potency for DGP 7

	ψ	-2	3	-4	5	-6
	$\mathsf{P}_{0.01}$	0.266	0.645	0.914	0.990	0.999
N=80	**Autometrics**	0.230	0.380	0.988	0.993	0.998
	Lasso	**0.507**	**0.466**	**0.999**	**0.963**	**1.000**
	BSTS	0.011	0.041	0.812	0.791	0.907
N=120	**Autometrics**	0.253	0.510	0.803	0.972	0.999
	Lasso	**0.603**	**0.683**	**0.904**	**0.926**	**1.000**
	BSTS	0.037	0.051	0.303	0.641	0.971
From DGP	**Autometrics**	0.330	0.649	0.931	0.980	1.000
	Lasso	0.964	0.967	1.000	1.000	1.000
	BSTS	0.103	0.242	0.965	0.983	0.993

Table 3.22: Retention Rates for DGP 7

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
		0.5	-0.4	0.6	-0.9	1.1	-1.25
$\boldsymbol{N = 80}$	**Autometrics**	**0.059**	0.286	**0.267**	**0.187**	**0.224**	**0.246**
	Lasso	0.078	0.257	0.415	0.330	0.514	0.532
	BSTS	0.111	**0.188**	0.358	0.327	0.405	0.438
$\boldsymbol{N = 120}$	**Autometrics**	**0.070**	0.328	**0.230**	**0.221**	**0.253**	**0.232**
	Lasso	0.121	**0.257**	0.370	0.526	0.564	0.421
	BSTS	0.147	0.272	0.362	0.458	0.515	0.359
From DGP	**Autometrics**	0.059	0.246	0.168	0.237	0.236	0.232
	Lasso	0.059	0.205	0.256	0.190	0.216	0.229
	BSTS	0.084	0.236	0.344	0.281	0.294	0.298

Table 3.23: RCMSE for DGP 7

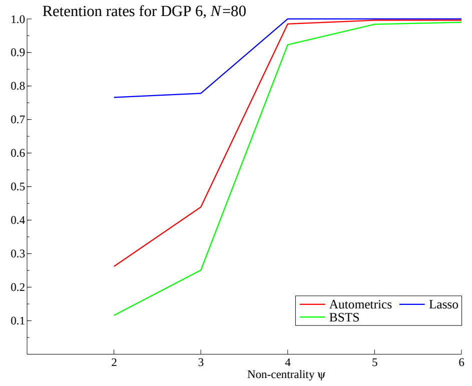


Figure 3.11: Retention rates DGP 6
$N = 80$

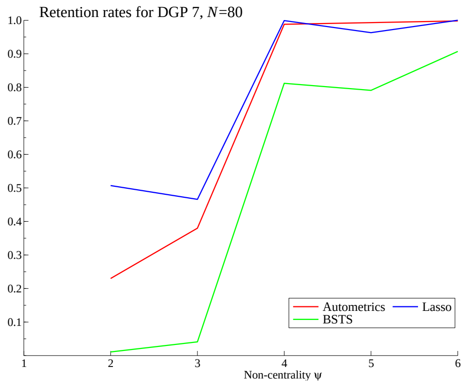


Figure 3.12: Retention Rates DGP 7
$N = 80$

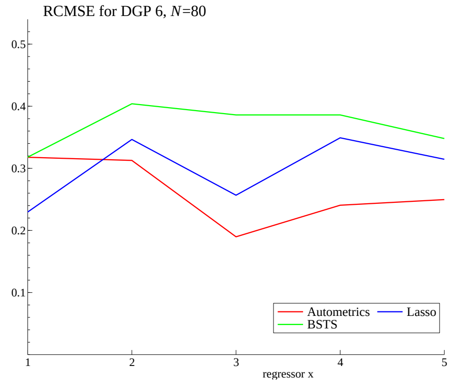


Figure 3.13: RCMSE DGP 6
$N = 80$

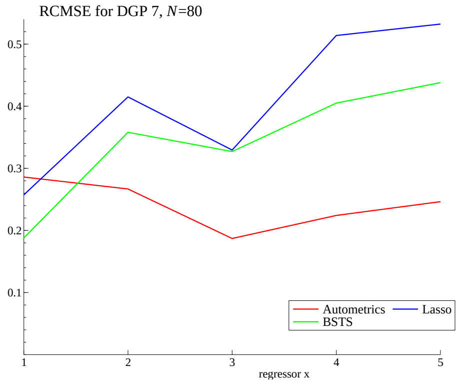


Figure 3.14: RCMSE DGP 7
$N = 80$

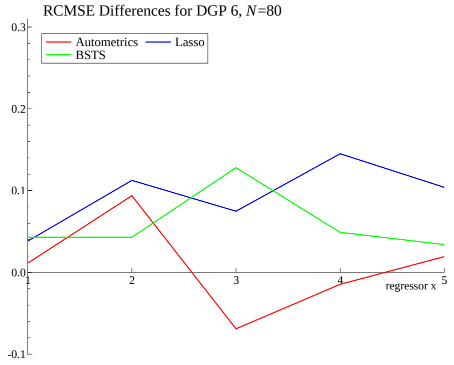


Figure 3.15: RCMSE differences DGP 6
$N = 80$

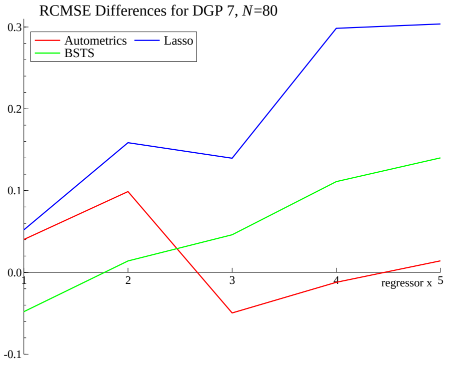


Figure 3.16: RCMSE differences DGP 7
$N = 80$

First considering the results from DGP 6, the Lasso appears to be effective at identifying relevant variables but also tends to select many irrelevant variables overall, as evidenced by the high gauge. When $\psi > 3$, the retention rates for all three algorithms are very high. The gauge for Autometrics is slightly higher than the significance level α, and almost 0 for BSTS. The RCMSE are generally highest for BSTS and lowest for Autometrics. Search costs are lowest for Autometrics.

With the DGP 7 results, the retention rates are similar to those of DGP 6 but different in a subtle and important way. Both the Lasso and BSTS struggle to pick up variables with $\psi > 0$. Similarly, the Lasso struggles to estimate the parameters on variables with $\psi > 0$. These difficulties are not as pronounced as in the previous set of experiments, but clear nonetheless. While the Lasso and BSTS struggle increasingly to estimate the parameter estimates as $|\psi|$ increases, Autometrics has RCMSEs which are low and consistent for all variables.

Comparing DGP 6 and DGP 7, several important differences arise. While in theory the DGPs are very similar with almost identical properties, the results vary. The Lasso and BSTS again appear to struggle with alternating signs both in terms of how effectively they pick up relevant variables and how accurate the parameter estimates are. The measures of gauge are consistent across all algorithms in both DGP 6 and DGP 7; again BSTS with the lowest gauge near 0, the Lasso with the highest in the range of 0.13-0.18, and Autometrics in the middle with gauges slightly above the significance level.

An additional observation is that the Lasso struggled less when dealing with alternating signs and all variables correlated in this set of experiments, than it did when only the relevant regressors were correlated in the previous set of experiments. This is likely due to the fact that in this set of experiments the correlation between all variables was $\omega_{jk} = 0.8$ and in the previous set of experiments the correlation between relevant variables was $\omega_{jk} = 0.9$.

The takeaways from this set of experiments are:

1. Alternating signs matter a lot for the Lasso, a reasonable amount for BSTS and a little for Autometrics. Note that negative coefficients and positive correlations are isomorphic to positive coefficients and negative correlations, so this result would also hold if it parameter signs were switched.

2. The Lasso is excellent at selecting relevant variables when all the signs are the same, but not very good at excluding irrelevant variables.

3. The gauge for Autometrics is consistently slightly above the significance level for all experiments.

4. BSTS generally selects very sparse models, which do not include many relevant or irrelevant variables.

.

3.6 Results Summary

Evaluating the algorithms up until this point has focused on comparing the results from model selection conditional on a known correlation structure between the regressors. Therefore, when thinking about what the results in the previous section say about empirical model selection, the correlation structure of the variables under consideration must be taken into account. For example, if a researcher used Autometrics or the Lasso on a set of orthogonal variables, she would know that the model selected by Autometrics included α% of the irrelevant variables, and the model selected by the Lasso included 15-20% of the irrelevant variables. Additionally she would know there was a 99% chance that a variable with a signal-to-noise ratio of 4 was included in the selected model. These statements rely on orthogonality however, and given that the world is dynamic, full of relationships and extremely interdependent, orthogonal results on their own are of limited use. Because a researcher never knows the true DGP, she cannot possibly know what the true correlation structure is. Thus, to truly be able to interpret and say something meaningful about the effectiveness of model selection empirically, it is vital to understand how different 'states of nature' influence the results. Since a researcher can never know the true DGP and therefore cannot qualify results in this manner, ideally model selection results should not depend on the correlation structure.

A desirable property of model selection would be for a variable with a given 'significance', which can be measured by its non-centrality, to be selected at the same level regardless of the properties of the other variables in the GUM. It is for this reason that in each of the three sets of experiments, $\beta_1, ..., \beta_5$ were chosen so that non-centralities are the same and retention rates and RCMSEs are comparable across experiments. Looking at how results vary across the three different correlation structures is therefore a straightforward way to analyze how orthogonality or otherwise influences results across values of ψ. Table 3.24 shows a summary of the results for DGP 3, DGP 4, and DGP 6 for $N = 80$. Similarly, Table 3.25 shows the results for DGPs with alternating signs. Note that results from alternating signs when the regressors are orthogonal are largely the same as when the signs are all the same, and were not reported earlier.

	Autometrics			**Lasso**			**BSTS**		
ψ	Orthogonal	Some Corr	All Corr	Orthogonal	Some Corr	All Corr	Orthogonal	Some Corr	All Corr
2	0.216	0.333	0.262	0.552	0.977	0.766	0.003	0.179	0.116
3	0.414	0.568	0.439	0.869	0.990	0.778	0.136	0.340	0.251
4	0.948	0.988	0.985	0.974	1.000	1.000	0.325	0.949	0.923
5	0.989	0.998	0.996	0.995	1.000	1.000	0.562	0.993	0.984
6	0.995	0.998	0.996	0.996	1.000	1.000	0.781	0.991	0.99
Potency	0.712	0.777	0.736	0.877	0.993	0.909	0.361	0.690	0.653
Gauge	0.018	0.014	0.016	0.179	0.108	0.182	0.001	0.000	0.004

Table 3.24: Retention rate, potency and gauge summary results for all positive ψ

	Autometrics			**Lasso**			**BSTS**		
ψ	Orthogonal	Some Corr	All Corr	Orthogonal	Some Corr	All Corr	Orthogonal	Some Corr	All Corr
-2	0.240	0.255	0.230	0.608	0.065	0.507	0.007	0.022	0.011
3	0.476	0.355	0.380	0.74	0.013	0.466	0.043	0.056	0.041
-4	0.928	0.981	0.988	0.981	0.984	0.999	0.349	0.798	0.812
5	0.993	0.996	0.993	1.000	0.095	0.963	0.943	0.827	0.791
-6	0.996	0.994	0.998	0.998	0.988	1.000	0.792	0.951	0.907
Potency	0.727	0.716	0.718	0.865	0.429	0.787	0.427	0.531	0.512
Gauge	0.018	0.014	0.014	0.176	0.088	0.154	0.000	0.000	0.001

Table 3.25: Retention rate, potency and gauge summary results for alternating ψ

These two tables contain many interesting insights. The results for Autometrics do not vary significantly across the different correlation structures, both when $\psi_k > 0, \forall k$ and when the signs of ψ alternate. As $|\psi|$ increases, correlation matters less and results converge to 1. The gauge is also consistent across correlation structures, fluctuating between 0.014 and 0.018. The Lasso, on the other hand, exhibits a very different story. While when $\psi_k > 0, \forall k$, and $\psi > 3$ the retention rates are consistent across the correlation structures, this is not the case when the signs are alternating. In fact, in the case of alternating signs, the Lasso picks up variable x_k with $\psi_k = 3$, 74% of the time when the regressors are orthogonal, compared to 1% of the time when there is correlation between the relevant regressors, and 46% of the time when there is correlation between all regressors. The gauge in the Lasso experiments is also remarkably varied, ranging between 0.09 and 0.18. These discrepancies and their implications are substantial. The same is true but to a lesser extent for BSTS where there is notable variation in retention rates across the three correlation structures, especially when $|\psi| < 4$. The gauge for BSTS, on the other hand, has almost no variation.

Figure 3.17 provides an interesting visual representation of the results in the previous two tables. The potency and gauge are plotted against each other for both $N = 80$ and $N = 120$, meaning that a total of twelve experiments are graphed. The experiments graphed have properties such that one would expect MC results to be similar. The ideal algorithm would have all of its points bunched in the upper left corner of the graph. As can be easily seen, the Lasso and BSTS points are decidedly not in the upper left quadrant. In fact, the Lasso has points all over the graph. While BSTS consistently achieves low gauges, it fares poorly when it comes to potency. Autometrics points are grouped together closely. While it may not always have as high a potency as the Lasso, or as low a gauge as BSTS, it does reasonably well in both and perhaps more importantly is consistent in the sense that the correlation structure of the regressors has no bearing on the results.

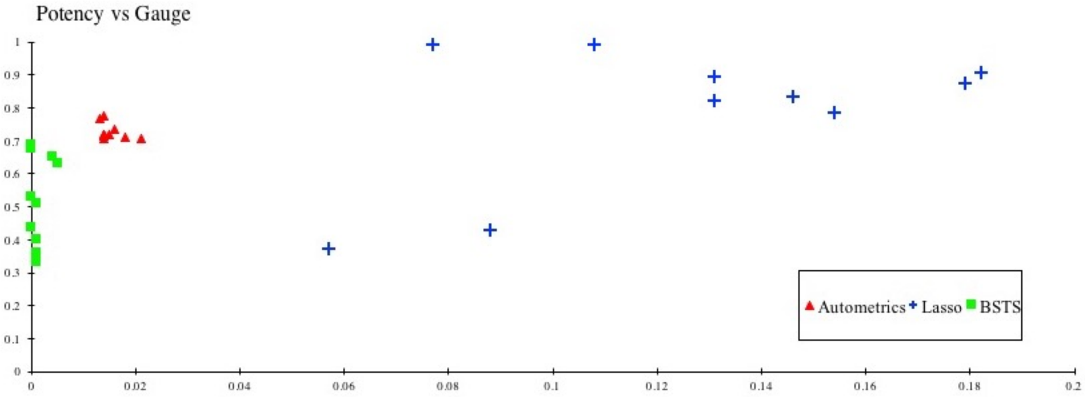


Figure 3.17: Scatter plot of potency vs. gauge across all experiments

3.6.1 What do the Monte Carlo simulation results say about empirical model selection?

As should be clear by now, model selection is difficult to evaluate empirically; it is impossible to judge how 'good' a selected model is because it is impossible to know what to be judging against. This is why the results from MC simulations are so important and useful, and in particular why analyzing what happens when correlation structures change is vital. Because a modeller can never know what the properties of the relevant variables are, to draw valuable insight from selected models it is key that model selection not be reliant on what those properties are. With this in mind, the results in this study imply that a modeller performing model selection using the three algorithms can assume the following:

Autometrics

1. There are slightly higher than $100(\alpha)\%$ of the total number of irrelevant variables included in the selected model.
2. The probability that a variable with 'significance', in signal-to-noise terms of ψ has been selected can be approximated by the theoretical retention probability formula.
3. The costs of inference, or equivalently the parameter estimate accuracy, does not vary according to how 'significant' a variable is. (Bias correction has not been employed here, but studies show that this a costless way of improving parameter accuracy. See Hendry and Krolzig (2005))
4. The costs of search for using Autometrics are minimal and sometimes even negative. That is, a researcher will 'lose' almost nothing employing Autometrics, but stands to gain considerably.

Lasso

1. There can be anywhere in the range of 8-20% of the total number of irrelevant variables included in the selected model. When $N = 120$, this implies keeping up to 24 irrelevant variables, compared to 6 relevant variables.
2. If all the important variables in the model happen to be relevant to the dependent variables in the same 'direction' (i.e. all matter in either a positive or negative way for the dependent variable), then the Lasso will identify and select them with a high probability. In particular, a variable with a signal-to-noise ratio of $\psi > 3$ has over a 97% change of being selected.
3. The probability of selecting a relevant variable with a lower signal-to-noise ratio varies according to how correlated it is with the other variables in the model. For example the probability of selecting a variable with a signal-to-noise ratio of $\psi = 2$ can be anywhere in the range of 55-98%.
4. Parameter estimates become increasingly inaccurate the more 'significant' a variable is.

BSTS

1. There are almost no irrelevant variables (around 0.1%) included in the selected model.
2. This sparsity has an impact on the inability to keep relevant variables.
3. BSTS is affected by the correlation structure of the regressors, and the signs of the coefficients.

Chapter 4

An Application: Using Automatic Model Selection for Nowcasting

In this chapter, automatic model selection is applied to the problem of nowcasting. It is important to note that while nowcasting may be a fruitful application of automatic model selection, nowcasting (or forecasting) is a fundamentally different problem with a different objective than model selection. While in a stationary world with no breaks the best in-sample model produces the best forecast, this is not the case in a non-stationary world. When dealing with non-stationarity, poor in-sample models can outperform in forecasting, and poor forecasting models can outperform in-sample models. Therefore this section is included to show an interesting application of Autometrics, the Lasso and BSTS and the results should be interpreted separately from the previous section.

4.1 Google Search Query Data

The amount of information Google collects through search queries is difficult to comprehend. According Amit Singhal, a Senior Vice President at Google, as of August 2012 Google was processing 100 billion searches every month.[1] That means there are over 3.5 billion Google searches each day and over 1 trillion Google searches per year. People turn to Google to get information on just about every aspect of their lives, so it makes sense that Google search queries contain large amounts of information about the state of the world. It is easy to think of cases where Google search queries may contain information about macroeconomic indicators. For example, if an individual loses their job, one of the first places they are likely to go is Google to determine what their unemployment benefits are. Google is also one of the first places they would turn to begin searching for a new job. Thus, Google search queries potentially tell a

[1] According to tech blogger Danny Sullivan (2012), this was revealed by Amit Singhal at a press event in San Francisco on August 8th 2012, and is a widely quoted figure.

story about the current state of unemployment in the economy. Similarly, individuals looking to buy new cars are likely to turn to Google to research their options before actually going to a dealership. Google search queries related to new cars could therefore provide insight into current consumer sentiment. Examples of studies which have endeavored to use Google search information for nowcasting and forecasting include D'Amuri and Marcucci (2012), Askitas and Zimmermann (2009) and Webb (2009).

Google has recently developed several online tools, namely Google Trends and Google Correlate, which allow users to analyze what people have been Googling. Google Trends allows users to enter any search term and see the frequency with which it has been searched over time. Google Correlate allows users to enter a search query and see which other search queries are most correlated with it. Google Correlate has an additional feature which allows the user to enter their own time series and provides the user with the search queries which are most correlated with their submitted time series (Mohebbi et al., 2011).[2]

4.1.1 Google Flu Trends

One of the first ways in which the information contained in Google search query data was harnessed was with the Google Flu Trends application. The Center for Disease Control (CDC) in America publishes statistics on the proportion of doctor or hospital visits due to influenza-like illness symptoms (ILI) every week.[3] This data is a leading indicator for the prevalence of the flu in the United States. The CDC, however, releases this data at a two-week lag. By economic standards two weeks is not long; most economic statistics are released with at least this much of a lag. For flu epidemics, however, two weeks can be a long time and there can be significant costs to this delay. Action to combat the flu in the form of vaccination, research and public awareness begins later than is ideal. There is therefore obvious value in knowing about a flu outbreak or epidemic as it is beginning. Google developed the Google Flu Trends (GFT) tool with the objective of using Google search queries to accomplish exactly this. GFT produced forecasts - or more precisely nowcasts - using Google search queries to estimate the current measure of ILI. Initially, these nowcasts turned out to be surprisingly accurate.

Google stopped publishing their ILI estimates in 2014, largely because Google Flu Trends performed poorly during the 2013 flu season. While Google never fully disclosed the algorithm behind Google Flu Trends directly, the general approach was outlined in a paper in Nature (Ginsberg et al., 2009). In this section, Google Flu Trends is revisited, and nowcasts are made using the three algorithms already analyzed. Again, it should be stressed that while in this section automatic model selection algorithms are used to produce nowcasts, model selection itself is a fundamentally different problem, with a different objective, than forecasting or nowcasting. Thus, while nowcasting is an interesting and informative exercise, comparing model selection

[2] Google Trends is available at https://www.google.co.uk/trends and Google Correlate is available at https://www.google.com/trends/correlate.

[3] Data is available to download via the FluView Interactive application available at http://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

algorithms for their ‘model selecting ability’ using the accuracy with which they are capable of nowcasting in these sections should be done cautiously, if at all.

4.2 The Data

Nowcasting ILI using automatic model selection commences in a similar way to model selection in general. The first step is to formulate the GUM, which includes all variables which may be predictors of ILI at time t. An important question then is to consider what data is available at time t which may be relevant for nowcasting ILI. Here, the GUM includes both past ILI values, and relevant Google Correlates, which are explained in this section.

4.2.1 Influenza-like-Illness Data

As mentioned, the CDC releases weekly data on the proportion of GP visits due to influenza-like illness with a two week lag. Weeks are measured from Sunday-Saturday and are numbered either 1-52 or 1-53, depending on how many weeks are in a particular year. Week 1 in a given year is the first week which is entirely in the new year; so Week 1 2016 went from Sunday January 3-Saturday January 9th. New data is published on Fridays and gives the ILI from two weeks earlier. For example, the incidence of ILI for Week 8 2016 which went from Sunday February 21-Saturday February 27 was released on Friday in Week 10 2016 (March 4). This means that if a nowcast for ILI for Week t were to be made on or after Friday of Week t, ILI data up to Week $t-2$ is available. Therefore, the GUM can include up until the second lag of ILI.

4.2.2 Google Correlate Data

The Google search terms which are most correlated with ILI, called the correlates, as well as their first lags, are also relevant for nowcasting. To find these correlates over a particular period, the weekly ILI time series corresponding to the desired time period was entered into the Google Correlate application. The time period differed with the two different nowcasting approaches taken, which will be explained below. Conveniently, like ILI, Google Correlate measures weeks from Sunday-Saturday. Google Correlate produces a list of the 100 search queries which are most correlated with ILI for the given sample, and also provides the de-meaned and standardized search volume for those 100 search queries as weekly time series. The search volume for a particular query is the proportion of total search queries throughout a given week which were for that query. As a practical matter, Google Correlate data is available to the public with a lag of several weeks. Acknowledging, however, that Google search query data is collected in real time, and in theory would be available for nowcasts, the nowcasting done here assumes that real time data is available. This approach is line with what was done with the original Google Flu Trends tool as well, making the results here comparable.

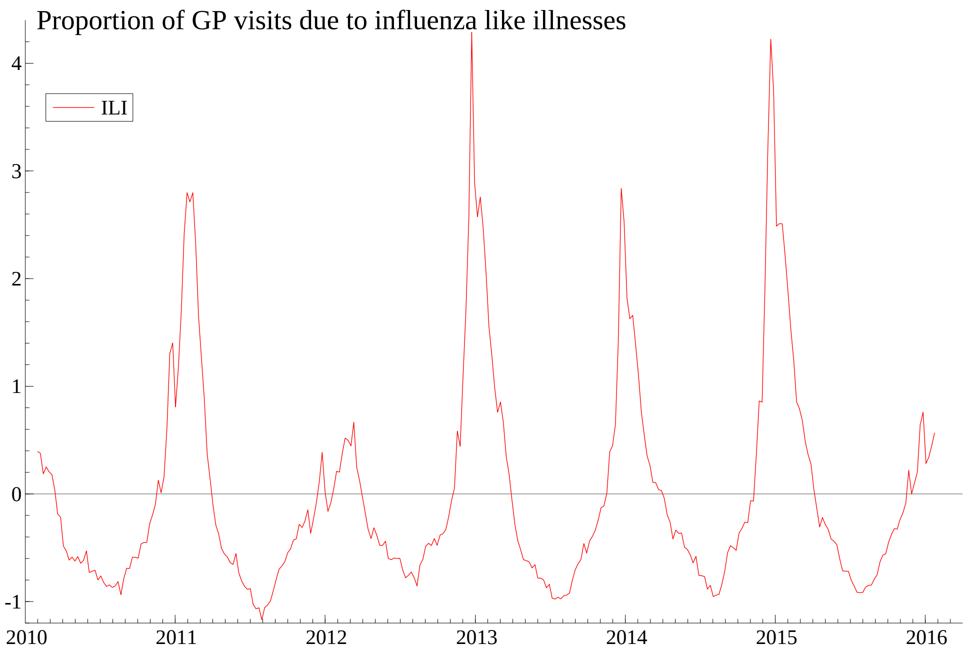


Figure 4.1: Proportion of GP visits due to influenza-like illness

4.3 Nowcasting with Autometrics, Lasso and BSTS

Nowcasting here is evaluated using both in-sample and out-of-sample results. In-sample nowcasting refers to a estimating a single model using data from the entire sample period. Fitted values for the sample period are the nowcasts. Out-of-sample nowcasting refers to selecting a holdout period, estimating a model which does not use the data from the holdout period, and using the estimated model to find fitted values for the holdout period. Finding in-sample and out-of-sample nowcasts is relatively straightforward using Autometrics and Lasso. As likely became apparent when the algorithms were described, BSTS is a much more complicated algorithm. Its main purpose, however, is nowcasting and there are features built into the BSTS R package which allow this to be done almost automatically. The reliance of BSTS on Kalman filtering and smoothing, however, means that at this point it is not capable of producing true out-of-sample nowcasts. Therefore, only in-sample nowcasts are results for BSTS.

4.3.1 In-sample Nowcasts using Autometrics and Lasso

The sample period used to find in-sample nowcasts went from Week 5 2010 to Week 4 2016 (the week beginning January 31 2010 to the week beginning on January 24 2016). The ILI time series was downloaded from the CDC website for this time period, and the observations are denoted $ILI_1, ..., ILI_{313}$ since there are 313 observations in the sample period. The GUM included any

	Search query	**Correlation with ILI**
x_1	how to get over the flu	0.942
x_2	get over the flu	0.942
x_3	how to get rid of the flu	0.932
x_4	get rid of the flu	0.927
x_5	flu fever	0.925
x_6	flu duration	0.924
x_7	type a influenza	0.924
x_8	getting over the flu	0.919
x_9	tamiflu and pregnancy	0.918
x_{10}	influenza type a	0.917

Table 4.1: Top 10 correlates and their correlation with ILI

variables potentially relevant for modeling ILI. A quick glance at the graph of ILI in Figure 4.1 indicates it is likely following an autoregressive process. ILI peaks in the winter around Christmas and is at its lowest in the summer, indicating that seasonality is important. Therefore, lags 2-53 of ILI were included in the GUM. Admittedly, this is not perfect as the number of weeks in a year varies and, for example, Christmas which likely is relevant to ILI can fall in a different week year-to-year. The other set of regressors in the GUM were the top 100 correlates, and their first lags. These were found by entering the ILI time series for the sample period into the Google Correlate application. Let $x_{1,t}, x_{2,t}, ..., x_{100,t}$ denote the search volumes of the top 100 correlates at time t. Table 4.1 shows the 10 search queries most correlated with ILI for the sample period. To give an example of the correlation structure of the Google correlates, the correlation matrix Ω of the top ten correlates is:

$$\Omega = \begin{pmatrix} 1.00 & 0.99 & 0.98 & 0.98 & 0.96 & 0.96 & 0.94 & 0.96 & 0.93 & 0.94 \\ 0.99 & 1.00 & 0.98 & 0.98 & 0.97 & 0.97 & 0.95 & 0.97 & 0.94 & 0.95 \\ 0.98 & 0.98 & 1.00 & 1.00 & 0.96 & 0.96 & 0.94 & 0.95 & 0.93 & 0.94 \\ 0.98 & 0.98 & 1.00 & 1.00 & 0.96 & 0.96 & 0.94 & 0.96 & 0.93 & 0.94 \\ 0.96 & 0.97 & 0.96 & 0.96 & 1.00 & 0.97 & 0.92 & 0.96 & 0.94 & 0.93 \\ 0.96 & 0.97 & 0.96 & 0.96 & 0.97 & 1.00 & 0.93 & 0.96 & 0.95 & 0.93 \\ 0.94 & 0.95 & 0.94 & 0.94 & 0.92 & 0.93 & 1.00 & 0.92 & 0.96 & 0.99 \\ 0.96 & 0.97 & 0.95 & 0.96 & 0.96 & 0.96 & 0.92 & 1.00 & 0.93 & 0.93 \\ 0.93 & 0.94 & 0.93 & 0.93 & 0.94 & 0.95 & 0.96 & 0.93 & 1.00 & 0.95 \\ 0.94 & 0.95 & 0.94 & 0.94 & 0.93 & 0.93 & 0.99 & 0.93 & 0.95 & 1.00 \end{pmatrix}$$

The top 100 correlates, their first lags, and the lagged ILI_t, formed the GUM:

$$ILI_t = \alpha + \sum_{i=2}^{53} \gamma_i ILI_{t-i} + \sum_{i=0}^{1} \sum_{j=1}^{100} \beta_{j,i} x_{j,t-i} + u_t$$

Autometrics and the Lasso were applied to the above GUM. A significance level α must be selected when using Autometrics, and it is not immediately clear which level should be chosen in this case. This problem is discussed in more detail below, but $\alpha = 0.01$ was used. Autometrics returned a model which had 60 parameters selected. Using 10-fold cross validation to select the tuning parameter λ, Lasso selected a model with 29 parameters. The nowcasts produced by Autometrics and the Lasso are the fitted values, and are plotted against ILI in Figure 4.4. Table 4.2 displays the mean squared errors, as calculated by:

$$MSE = \frac{1}{52} \sum_{t=262}^{313} (ILI_t - \widehat{ILI_t})^2$$

where $\widehat{ILI_t}$ is the fitted value for ILI at time t. Since nowcasts were made for the week of February 1 2015 to the week of January 24 2016, there are 52 nowcasts in total, denoted $\widehat{ILI}_{262}, ..., \widehat{ILI}_{313}$ to reflect that they correspond to the last 52 observations in the sample.

4.3.2 In-sample Nowcasts using BSTS

While it is fairly easy to understand how Autometrics and the Lasso calculate their nowcasts, BSTS is more complicated. BSTS deals with time series components and regression components separately, so the first step is determining which time series components, and which regression components to include in the system. Based on the graph of ILI in Figure 4.1, a local linear trend, and seasonal component with $S = 53$ are used to model the time series dynamics. The regression component consists of the 100 Google correlates and their first lags, denoted by $\mathbf{x_t}$ and $\mathbf{x_{t-1}}$ respectively. Using the state space representation as earlier, ILI can then be described by the following system, which is summarized by 'time series + regression components':

$$\begin{aligned}
ILI_t &= \mu_t + \tau_t + \beta_1' \mathbf{x_t} + \beta_2' \mathbf{x_{t-1}} + \epsilon_t \\
\mu_t &= \mu_{t-1} + \gamma_{t-1} + u_t \\
\gamma_t &= \gamma_{t-1} + v_t \\
\tau_t &= -\sum_{s=1}^{52} \tau_{t-s} + w_t
\end{aligned}$$

Operationally, BSTS first determines 'how much' of ILI_t is explained by the local linear trend and seasonality components using Kalman filtering, smoothing and Bayesian data augmentation.

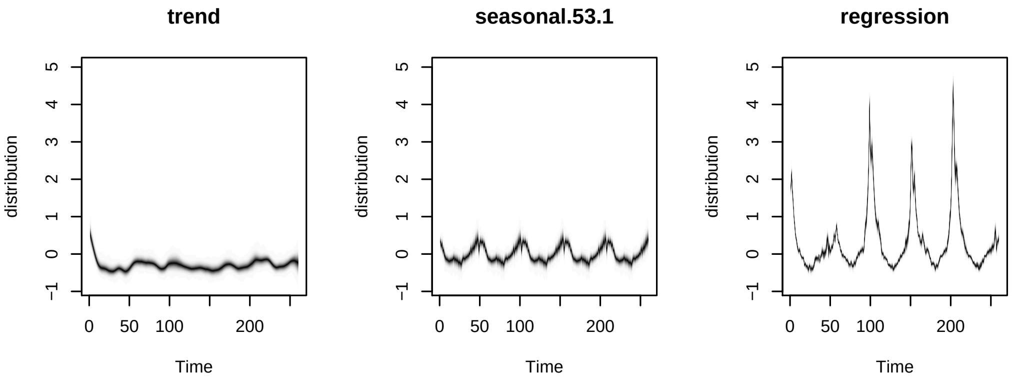


Figure 4.2: Contributions to ILI by state

Thus, the following system is relevant for the time series component of the algorithm:

$$\begin{aligned} ILI_t &= \mu_t + \tau_t + \epsilon_t \\ \mu_t &= \mu_{t-1} + \gamma_{t-1} + u_t \\ \gamma_t &= \gamma_{t-1} + v_t \\ \tau_t &= -\sum_{s=1}^{52} \tau_{t-s} + w_t \end{aligned}$$

which is exactly as above, but with the regression component subtracted out. BSTS subtracts the time series component from ILI to end up with ILI^*. Then a spike-and-slab regression is used with ILI^* as the dependent variable to determine which of the Google correlates (and their first lags) are relevant for explaining what is not explained by time series components. From the results of the spike-and-slab regression the posterior inclusion probabilities are calculated for each of the Google correlates. Figure 4.2 provides a graphical illustration of the contributions of each state. Note that the lines are fuzzy because the draws are graphed. Figure 4.3 shows Google correlates which have posterior inclusion probability greater than 0.15.

As a Bayesian algorithm, BSTS does not, in fact, calculate fitted values; it derives the predictive posterior distribution for the response variable, and subsequently produces a sample of draws from this distribution (Scott and Varian, 2014). Nowcasts can then be found by taking the mean across these draws. Figure 4.4 graphs the BSTS nowcasts and Table 4.2 shows the mean squared errors of these nowcasts.

4.3.3 Comparing In-Sample Nowcasts

Looking at the graphs in Figure 4.4, all three algorithms produce reasonably accurate nowcasts. Of the three, the Lasso seems to struggle the most. This is confirmed by the MSEs in Table 4.2.

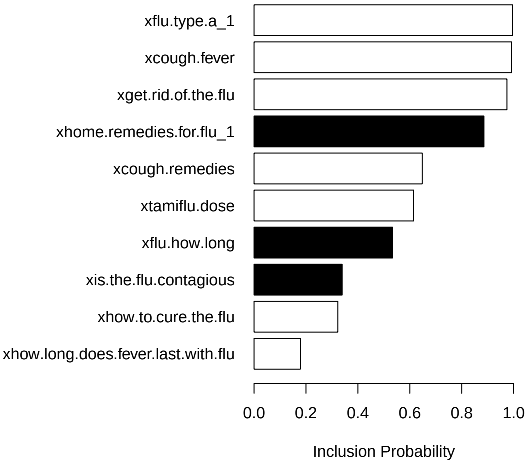


Figure 4.3: Posterior inclusion probabilities for $p > 0.15$. White bars indicate the coefficient sign is positive and black bars indicate the coefficient sign is negative.

	MSE	**Variables Selected**
Autometrics	0.003	60
Lasso	0.008	26
BSTS	0.004	10

Table 4.2: MSEs from in-sample nowcasting

The number of regressors selected by each of the algorithms, or in the case of BSTS, the number of variables with inclusion probability greater than 0.15, is also shown in Table 4.3. Table 4.2 provides a full list of the variables selected by each algorithm.

Autometrics has the smallest MSE and selects the largest number of variables with 60 of the 252 possible variables selected. BSTS identifies only 10 variables which have posterior inclusion probabilities greater than 0.15. Comparatively, the Lasso's nowcasts are decidedly less accurate. The fact that the Lasso selected half as many variables as Autometrics seems to point to a situation where there is high degree of correlation between the regressors, and where some of these variables are relevant in opposite ways. The table and graphs show that while all three algorithms are reasonably effective and nowcast with a high degree of accuracy, Autometrics does particularly well.

4.3.4 Out-of-sample Nowcasting with Autometrics and the Lasso

In-sample nowcasting or forecasting where the GUM is the starting point obviously has the potential to be very accurate due to the fact that the training set and the validation set are

BSTS	**Lasso**	**Autometrics**
flu type a_1	anas barbariae	anas barbariae
cough fever	breaking a fever	breaking a fever
get rid of the flu	cold fever	child with flu
home remedies for flue	cough fever	cold fever
cough rememdies	cough remedies	cold or flu
tamiflu dose	cure a cough	cough fever_1
flu how long	dangerous fever	cure a cough
is the flu contagious	flu medicines	cure a cough_1
how to cure the flu	flu medicines_1	dangerous fever
how long does fever last with flu	ILI_14	flu a
	ILI_15	flu a_1
	ILI_16	flu aches
	ILI_17	flu aches_1
	ILI_2	flu and pregnancy
	ILI_28	flu how long_1
	ILI_29	flu recovery
	ILI_3	flu remedies_1
	ILI_50	flu treatments
	ILI_51	flu type a
	influenza treatment_1	flu type a_1
	tamiflu and breastfeeding	generic tamiflu_1
	tamiflu drug_1	get rid of the flu
	tamiflu pediatric dosing	home remedies for flu
	tamiflu side effects	how long contagious
	treat a fever	how long does fever last with flu
	type a flu	how long does it take to get over the flu
		how long is the flu contagious
		how long is the flu_1
		how to cure the flu
		how to cure the flu_1
		how to get rid of flu_1
		how to treat the flu_1
		ILI_2
		ILI_3
		ILI_30
		ILI_4
		ILI_5
		ILI_6
		ILI_9
		influenza a_1
		influenza type a
		is the flu contagious
		is the flu contagious_1
		medicine for the flu_1
		over the counter flu
		remedies for flu_1
		symptoms of pneumonia
		tamiflu and alcohol_1
		tamiflu and pregnancy
		tamiflu and pregnancy_1
		tamiflu drug_1
		tamiflu side effects
		treat a fever
		treat a fever_1
		treat the flu
		treat the flu_1
		treating the flu_1
		type a flu_1

Table 4.3: Selected regressors

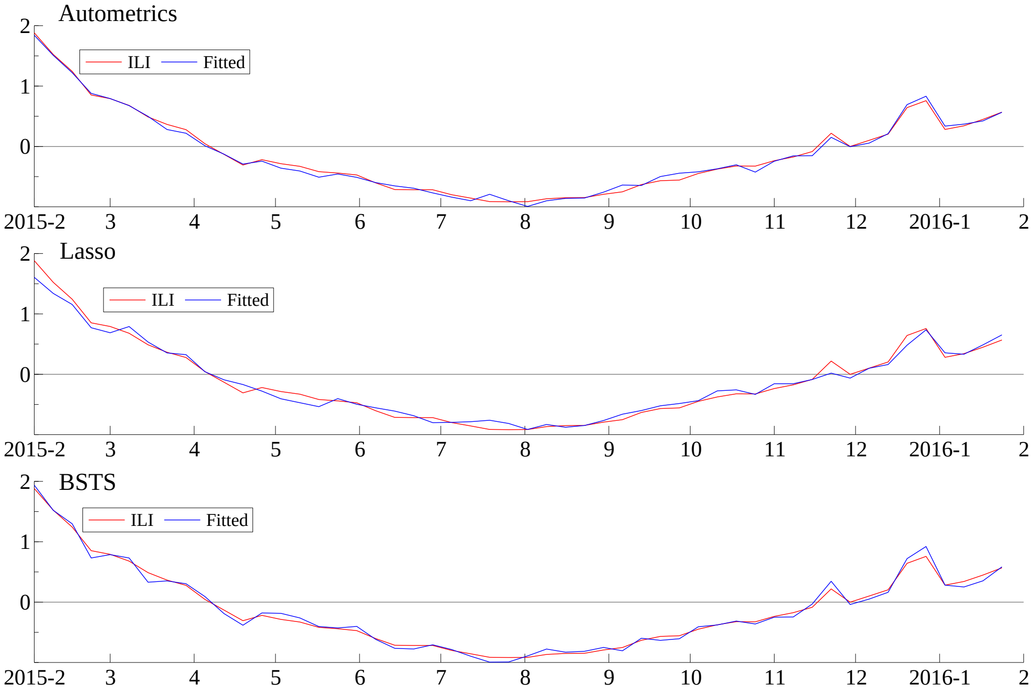


Figure 4.4: In-sample nowcasts

the same. The example of an extreme case of using OLS when $T = N$, and the fitted values exactly correspond to the true values comes to mind. It is important to consider how nowcasts perform out-of sample, as this reflects the situation in which nowcasts would be made in reality. A nowcast for time t is based on a model estimated using only the information available at t is a better reflection of how nowcasting would be useful in practice.

To calculate out-of-sample nowcasts using automatic model selection there are several decisions that need to be made. First it must be decided whether the algorithm should reselect a model from the GUM for each successive nowcast. If a new model is estimated for each successive nowcast, another question is whether a rolling or recursive window is more appropriate for determining the sample period. An issue specific to using Google Correlates is whether a new set of correlates should calculated from the Google Flu Trends application for each new nowcast.

The following explanation is easier to understand with reference to Table 4.7. The chosen sample period went from Week 5 2010 to Week 4 2016 (the week beginning January 31 2010 to the week beginning on January 24 2016) from which there are 313 observations. The 'modified holdout' period for which nowcasts were made went from Week 5 2015 and Week 4 2016 (the week beginning February 01, 2015 to the week beginning January 24, 2016). These nowcasts are denoted $\widehat{ILI}_{262}, ..., \widehat{ILI}_{313}$. The top 100 Google correlates were found by entering the ILI time series for the sample period excluding the holdout period $(ILI_1, ..., ILI_{261})$ into the Google

Correlate application. The top 100 Google correlates are denoted $x_{1,t}, ..., x_{100,t}$. Variables were selected once and selection was made using only the data in the subsample which excluded the holdout period. So, variables were selected from the following GUM:

$$ILI_t = \alpha + \sum_{i=2}^{53} \gamma_i ILI_{t-i} + \sum_{i=0}^{1} \sum_{j=1}^{100} \beta_{j,i} x_{j,t} + u_t$$

for $t = 1, ..., 261$. The model estimated from the GUM was used to produce the first nowcast, for the week of Feb 1, 2015, denoted $\widehat{ILI}_{262}$. The reason why the holdout period is 'modified' is because for each successive nowcast, an additional observation was added to the sample and a new model was estimated by OLS. The variables included in this model were the variables initially selected by the GUM. Let $z_1, ..., z_L$ denote the L regressors selected initially from the GUM. Nowcast $\widehat{ILI}_{262+k}$ was found by estimating the following model, for k=1,..,51:

$$ILI_t = \alpha + \sum_{i=1}^{L} \xi_i z_{i,t} + v_t, \qquad t = 1, ..., 262 + k$$

Therefore, new estimates for the ξ_is were made to find the new model used to produce each nowcast. The fitted value $\widehat{ILI}_{262+k}$ from the k^{th} model estimated is the nowcast. This process was used for both Autometrics and the Lasso. For Autometrics, the significance level α was set to 0.01. In the case of Lasso, selecting variables once and using OLS with the selected variables as the regressors for the subsequent nowcasts is equivalent to setting the tuning parameter $\lambda = 0$. The nowcasts and ILI produced by Autometrics and Lasso are graphed in Figure 4.5. The nowcast MSEs, and the number of variables selected by each algorithm is in Table 4.4. Autometrics and Lasso perform similarly, with Lasso achieving a slightly smaller MSE than Autometrics.

	MSE	**Variables Selected**
Autometrics	0.016	33
Lasso	0.014	30
Average	0.005	

Table 4.4: MSEs from out-of-sample nowcasting

The in-sample nowcast graphs indicate that the Autometrics nowcasts tend to be above the true ILI, and that the Lasso's tend to be below the true ILI. A graph of the residuals in Figure 4.6 makes this a little clearer. This suggests that perhaps the average of the two nowcasts would produce an even more accurate nowcast. Let $\widehat{ILI}_{ave,t}$ denote this average, $\widehat{ILI}_{auto,t}$ denote the Autometrics nowcast and $\widehat{ILI}_{lasso,t}$ denote the Lasso nowcast as calculcated above. The new average nowcast for time t is:

$$\widehat{ILI}_{ave,t} = \frac{1}{2}(\widehat{ILI}_{auto,t} + \widehat{ILI}_{lasso,t})$$

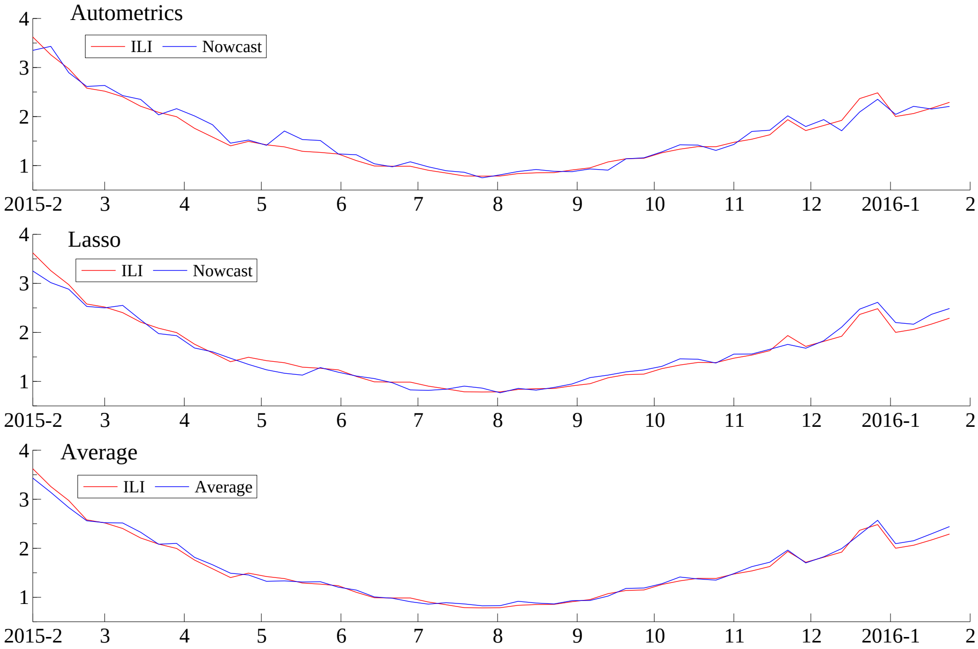


Figure 4.5: Out-of-sample Nowcasts

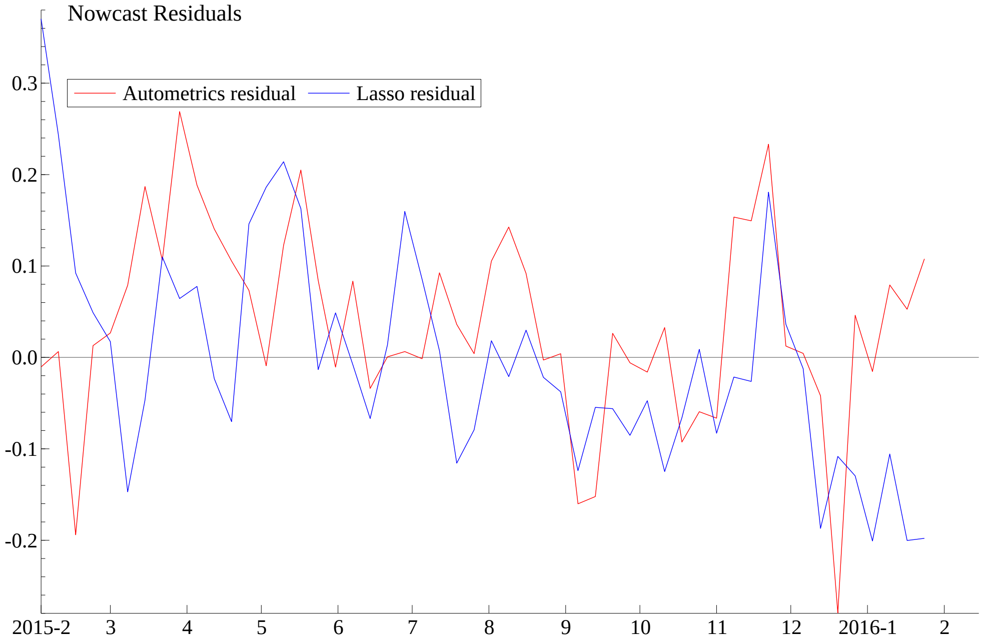
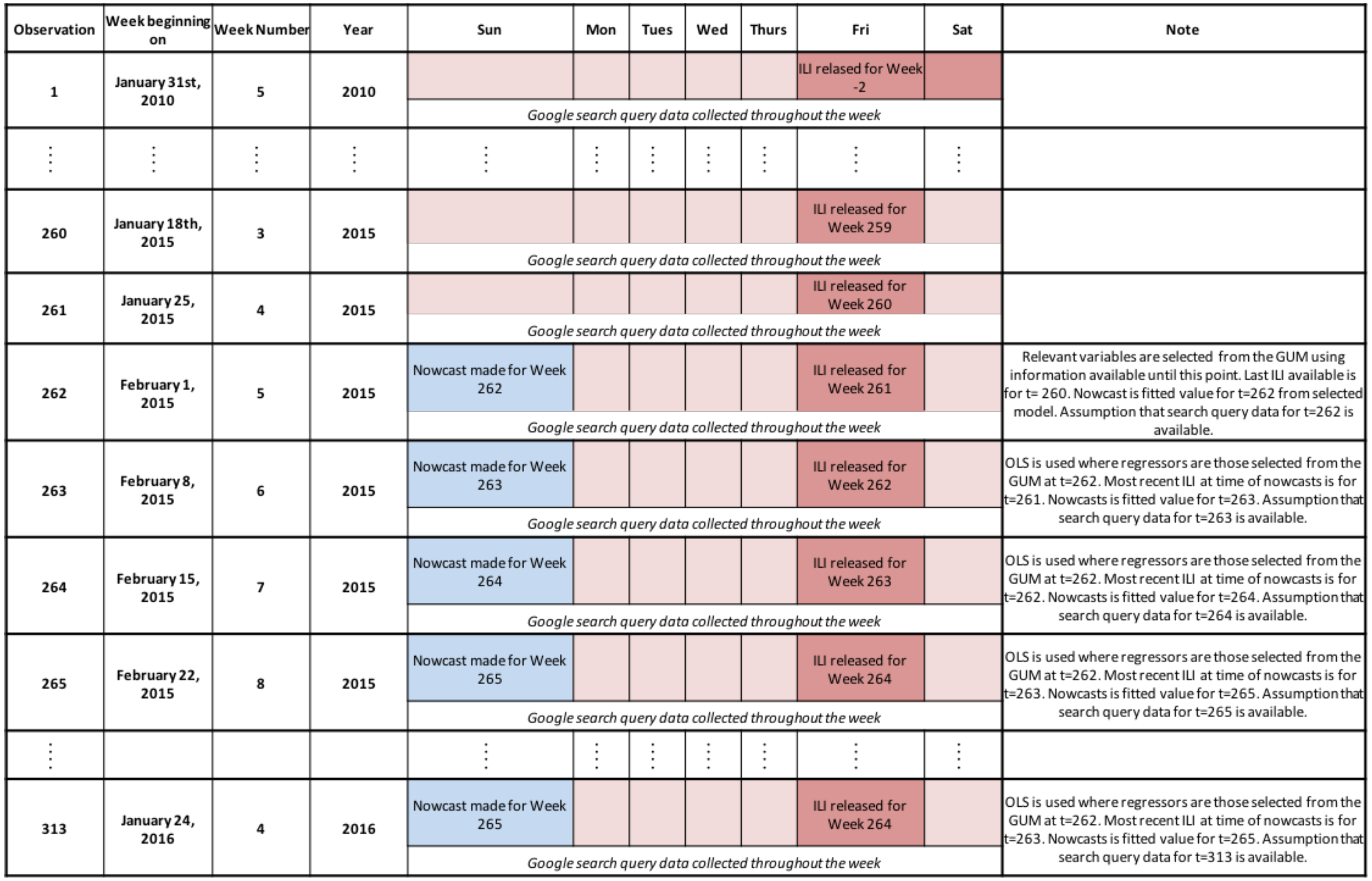


Figure 4.6: Residuals from Autometrics and Lasso nowcasts



Observation	Week beginning on	Week Number	Year	Sun	Mon	Tues	Wed	Thurs	Fri	Sat	Note
1	January 31st, 2010	5	2010						ILI relased for Week -2		
				Google search query data collected throughout the week							
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
260	January 18th, 2015	3	2015						ILI released for Week 259		
				Google search query data collected throughout the week							
261	January 25, 2015	4	2015						ILI released for Week 260		
				Google search query data collected throughout the week							
262	February 1, 2015	5	2015	Nowcast made for Week 262					ILI released for Week 261		Relevant variables are selected from the GUM using information available until this point. Last ILI available is for t= 260. Nowcast is fitted value for t=262 from selected model. Assumption that search query data for t=262 is available.
				Google search query data collected throughout the week							
263	February 8, 2015	6	2015	Nowcast made for Week 263					ILI released for Week 262		OLS is used where regressors are those selected from the GUM at t=262. Most recent ILI at time of nowcasts is for t=261. Nowcasts is fitted value for t=263. Assumption that search query data for t=263 is available.
				Google search query data collected throughout the week							
264	February 15, 2015	7	2015	Nowcast made for Week 264					ILI released for Week 263		OLS is used where regressors are those selected from the GUM at t=262. Most recent ILI at time of nowcasts is for t=262. Nowcasts is fitted value for t=264. Assumption that search query data for t=264 is available.
				Google search query data collected throughout the week							
265	February 22, 2015	8	2015	Nowcast made for Week 265					ILI released for Week 264		OLS is used where regressors are those selected from the GUM at t=262. Most recent ILI at time of nowcasts is for t=263. Nowcasts is fitted value for t=265. Assumption that search query data for t=265 is available.
				Google search query data collected throughout the week							
⋮				⋮	⋮	⋮	⋮	⋮	⋮	⋮	
313	January 24, 2016	4	2016	Nowcast made for Week 265					ILI released for Week 264		OLS is used where regressors are those selected from the GUM at t=262. Most recent ILI at time of nowcasts is for t=263. Nowcasts is fitted value for t=265. Assumption that search query data for t=313 is available.
				Google search query data collected throughout the week							

Figure 4.7: The timing of in-sample nowcasts and ILI releases

The average nowcast is also graphed against ILI in Figure 4.5. The MSE of the average is 0.005, which is less than half the MSE of both Autometrics and Lasso individually.

4.3.5 Predicting ILI with Autometrics and the Lasso

Given that both Autometrics and the Lasso seem reasonably effective at producing accurate nowcasts, an interesting question is whether actual predictions of ILI would be useful. To test this, forecasts of ILI were made using Autometrics and the Lasso. The forecasting here was done in much the same way as nowcasting was done in the previous section. The sample period from Week 5 2010 to Week 4, 2016 (the week beginning January 31 2010 to the week beginning on January 24 2016) from which there are 313 observations. Forecasts were made for the 52 weeks between Week 5 2015 and Week 4 2016 (the week beginning February 01, 2015 to the week beginning January 24, 2016), and are denoted $\widetilde{ILI}_{262}, ..., \widetilde{ILI}_{313}$. The top 100 Google correlates were found by entering the ILI_{t+1} time series for the sample period into the Google Correlate application. These correlates, then, are the search terms at time t which are most correlated with ILI at $t+1$ and are denoted $x^*_{1,t}, ...x^*_{100,t}$. Again, variables were only selected once, using only the data in the sample period. Thus, the initial model was selected from the GUM:

$$ILI_{t+1} = \alpha + \sum_{i=2}^{53} \gamma_i ILI_{t-i} + \sum_{i=0}^{1} \sum_{j=1}^{100} \beta_{j,i} x^*_{j,t-i} + u_t$$

The significance level was again set to $\alpha = 0.01$ for Autometrics. The first forecast was made for ILI_{262} and was produced using the model selected from GUM. The fitted value $\widetilde{ILI}_{262}$ from this model is the reported forecast. For the 51 subsequent forecasts, OLS was used to re-estimate a model using the variables selected by the GUM, and using the information available at the time the forecast was made. For example, the forecast for ILI_{263} was made using information available at $t = 262$. If $z_1, ..., z_L$ were L the variables selected from the GUM, then to produce $\widetilde{ILI}_{262+k}$, the following model was estimated by OLS:

$$ILI_{t+1} = \alpha + \sum_{i=1}^{L} \xi_i z_{i,t} + v_t, \qquad t = 1, ..., 260 + k$$

The k fitted values for ILI_{t+1} from each of the k models estimated are the reported forecasts. The forecasts are plotted in Figure 4.8, and the MSEs are reported in Table 4.5. The forecasts produced by Autometrics are noticeably less accurate than the nowcasts. The Lasso performs nearly as well at forecasting as it does at nowcasting. The average is better than both, Autometrics and Lasso individually, though only marginally better than the Lasso.

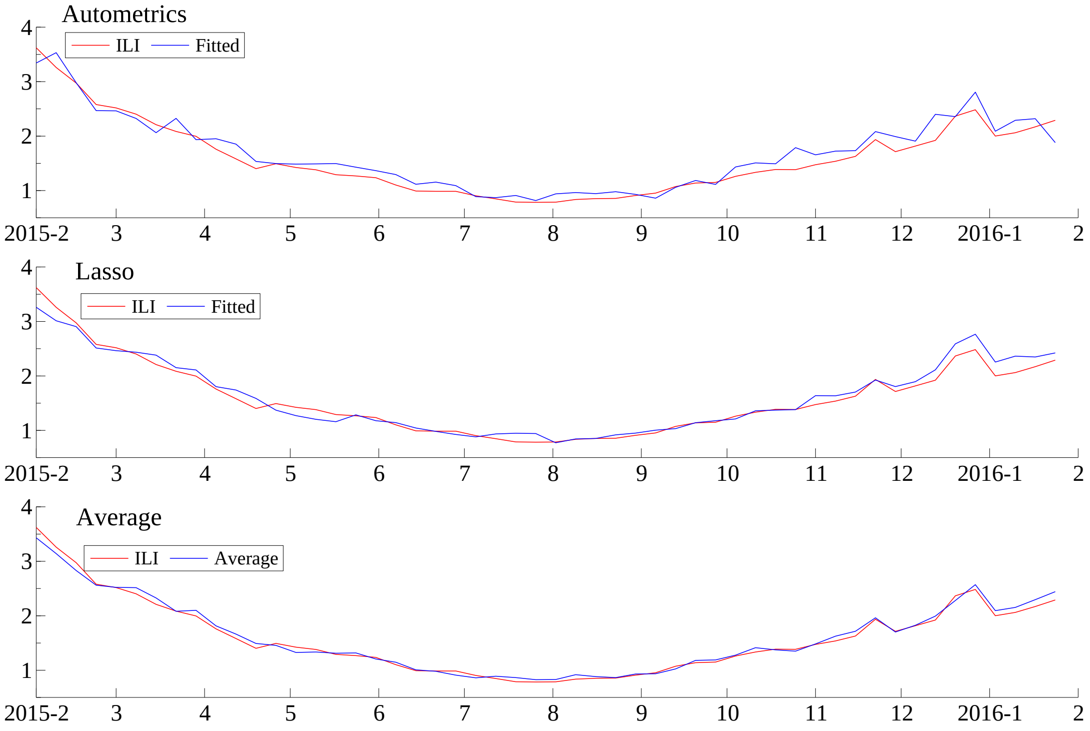


Figure 4.8: Forecasts produced by Autometrics, Lasso, and their average.

	MSE	**Selected Variables**
Autometrics	0.032	41
Lasso	0.018	22
Average	0.016	

Table 4.5: MSEs from forecasting

4.3.6 Original Google Flu Trends Estimates

Since its inception in 2008, the Google Flu Trends model has been updated three times, most recently in 2014. While Google has never released the algorithm it used to compute its ILI estimates, the estimates themselves are available. It has also released the estimates that the revised models produce when applied to historical data.[4]. An interesting exercise is then to see how these estimates compare to the estimates produced using the algorithms studied here.

Figure 4.9 graphs the nowcasts produced by Google's 2014 model, for the period January 24, 2010 until October 19 2015 which is the last estimate available.[5] Note that it is not clear

[4] ILI estimates from all versions of the Google Flu Trend algorithm can be downloaded from `https://www.google.org/flutrends/about/`

[5] The estimates in this graph were downloaded on March 21st from `https://www.google.org/flutrends/about/data/flu/historic/us-historic-v3.txt`

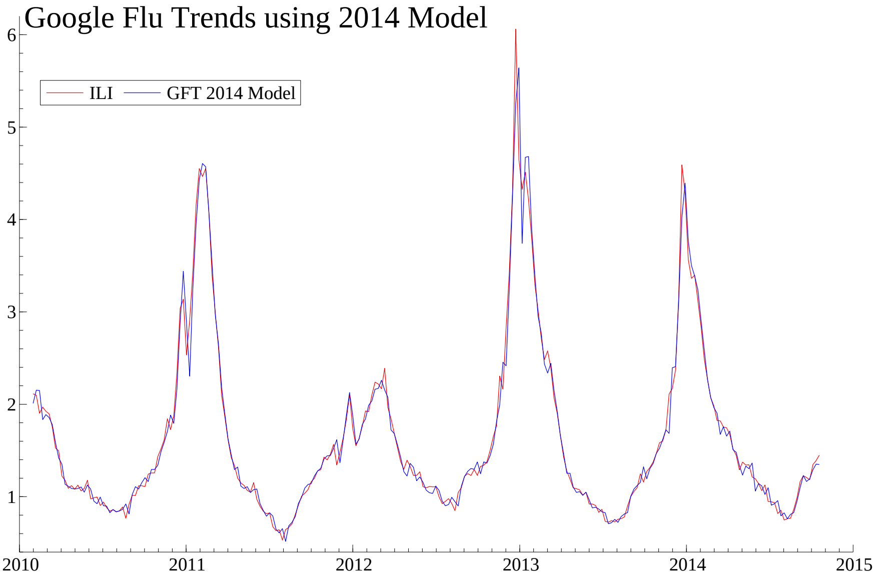


Figure 4.9: Nowcasts for 2010-2015 using Google's 2014 model

whether these are in-sample or out-of-sample nowcasts. The nowcast MSE over this period is 0.021, which is larger than the in-sample MSEs for Autometrics, Lasso and BSTS, and the out-of-sample nowcast MSEs produced by Autometrics and the Lasso.

4.4 Selecting the Significance Level for Autometrics

The accuracy of the nowcasts and forecasts produced by the three different algorithms has been measured by the MSE, which measures how close the predictions and true values are. Clearly, this is a very different way to evaluate the algorithms than the gauge, potency and MSE measures which were used in Chapter 3. When using Autometrics there is strong relationship between the chosen significance level α and the inclusion of irrelevant variables. Approximately $100(\alpha)\%$ of the irrelevant variables present in the GUM are in the model selected by Autometrics. The selection of relevant variables is also influenced by α; a higher α means a lower critical value c_α which leads to more relevant variables selected. Therefore, when selecting α there exists a tradeoff between setting a low α and excluding most of the irrelevant variables and retaining fewer of the relevant variables, or selecting a high α, and retaining most relevant variables, while simultaneously retaining many irrelevant variables. When the goal of using Autometrics is model selection, the consensus is that setting α very low is preferable. However when the goal is not model selection but is instead forecasting, setting a higher α could produce better results. This is due to the fact that in the prediction arena, it is sometimes more costly to exclude relevant variables than it is to retain irrelevant variables. There is a long debate over parsimony versus robustness and it is far from settled, but given that excluding relevant variables is perhaps more costly than including irrelevant variables, it seems sensible to entertain the idea that higher values of α could improve nowcasts and forecasts generated by Autometrics. Out-of-sample nowcasts and forecasts were reproduced with this in mind, using the out-of-sample nowcasting and forecasting methods as previously explained, with various values of α. The MSE results for the recalculated nowcasts are reported in Table 4.6, and results for the recalculated forecasts are reported in Table 4.7. Averages calculated using the updated nowcasts and forecasts are also reported. Both the nowcasts and forecasts produced by Autometrics generally improve with the higher levels of significance.

α	**Autometrics**	**Average**	**Variables Selected**
0.01	0.016	0.005	33
0.05	0.018	0.005	59
0.10	0.012	0.008	62
0.15	0.012	0.005	91
0.20	0.012	0.006	92

Table 4.6: Nowcast MSEs for various levels of α

α	**Autometrics**	**Average**	**Variables Selected**
0.01	0.032	0.016	41
0.05	0.025	0.014	48
0.10	0.015	0.009	73
0.15	0.017	0.012	140
0.20	0.007	0.008	161

Table 4.7: Forecast MSEs for various levels of α

4.5 Nowcasting Takeaways

As this chapter shows, nowcasting is an interesting, relevant and useful application of automatic model selection and depending on the complexity of the algorithm, it can also be employed very easily. As the way that economic data is collected begins to change, and as it starts to become available in real-time, automatic model selection for nowcasting could become a useful tool for economics, statistical agencies and policy makers alike.

Chapter 5

Conclusion

Automatic model selection algorithms have the potential to assist economic researchers a great deal. As the big data revolution starts to play a more integral role in economic research, algorithms like Autometrics, the Lasso and BSTS will become more mainstream. The results of this thesis demonstrate that while the three algorithms have broadly the same goal of selecting a sparse model from a much more general model, the properties of each algorithm vary. Of particular relevance to the results is the correlation structure of the DGP. Because empirically the structure of the DGP is impossible to know it should be viewed unfavourably if an algorithm's properties depend on a particular correlation structure.

This thesis began by first outlining the theory behind the three different algorithms studied. Then, results from simulations conducted across three different correlation structures were presented individually. Within each of these three sets of simulations, different non-centralities for the relevant variable were tested. Also examined was the impact of including more variables than observations in the initial GUM. Algorithms were compared to each other and against a benchmark to examine the costs of search and costs of inference. Finally the results across different correlation structures were analyzed.

Of the three algorithms studied, Autometrics was found to be by the far the most consistent across correlation structures. The Lasso's results vary a great deal depending on the correlation structure and the non-centralities of the relevant variables. This makes it difficult to describe the general properties of a model the Lasso selects. BSTS selects very sparse models overall, with the properties of the selected model dependent on the correlation structure as well. The number of candidate variables did not seem to matter in any of the algorithms; that is in most simulations, each algorithm was just as effective when the number of variables exceeded the number of observations.

A nowcasting application of automatic model selection was then considered, where each of the algorithms was used to nowcast the incidence of the flu in the United States, using information from Google search queries. All three algorithms generated very good in-sample nowcasts. The out-of-sample nowcasts were also very accurate. Autometrics and the Lasso were also used to try to forecast the flu, with success. The results were compared to actual Google Flu Trends estimates, and were shown to be even more accurate. The results in this section suggest that nowcasting and forecasting may be a profitable application of automatic model selection algorithms. However given that the goal of nowcasting is to minimize predictive error, more work needs to be done to understand the best way to use automatic model selection in the nowcasting arena.

The research presented in this thesis has demonstrated that if economists wish is to effectively and appropriately take advantage of the new tools for big data like the ones presented here, it is essential they do their research. The results demonstrate that not all algorithms are created equally, and that choosing one technique over another can have a significant impact on results. That said, the results in this thesis suggest that Autometrics is a promising, consistent and reliable algorithm and has the potential to greatly simplify the work of empirical researchers.

References

Bradley Efron, Trevor Hastie, I. J. and Tibshirani, R. (2004). Least Angle Regression. *The Annals of Statistics*, 32(2):407–499.

Carter, C. and Kohn, R. (2004). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Castle, J. L., Doornik, J. A., and Hendry, D. F. (2011). Evaluating Automatic Model Selection. *Journal of Time Series Econometrics*, 3(1):1–33.

Castle, J. L. and Shephard, N. (2009). *The Methodology and Practice of Econometrics.* Oxford University Press.

Chen, J. and Chen, Z. (2007). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Doornik, J. A. Autometrics: In Castle, and Shepard (2009). pages 88–121.

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

Epprecht, C., Guegan, D., and Veiga, A. Comparing variable selection techniques for linear regression: LASSO and Autometrics.

Ericsson, N. R., Hendry, D. F., and Hood, S. B. (2016). Milton Friedman as a Statistician and Econometrician. In Cord, R. and Hammond, D., editors, *Milton Friedman: Contributions to Economics and Public Policy*, page forthcomimg. Oxford University Press, Oxford.

Gareth James, Daniela Witten, T. H. and Tibshirani, R. (2013). *An Introduction to Statistical Learning.* Springer.

George, E. I. and McCulloch, R. E. (1993). Variable Selection Via Gibbs Sampling. *Journal of the American Statistical Association*, 88(423):881–889.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinkso, M. S., and Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1015.

Harvey, A. C. (1991). *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge University Press.

Hendry, D. F. and Doornik, J. A. (2014). *Empirical Model Discovery and Theory Evaluation: Automatic Selection Methods in Econometrics.* The MIT Press.

Hendry, D. F., Johansen, S., and Santos, C. (2008). Automatic selection of indicators in a fully saturated regression. *Computational Statistics*, 23(2):337–339.

Hendry, D. F. and Krolzig, H.-M. (1999). Improving on 'Data Mining Reconsidered' by K. D. Hoover and S. J. Perez. *Econometrics Journal*, 2(2):202–219.

Hendry, D. F. and Krolzig, H.-M. (2005). The properties of automatic GETS modelling. *Economic Journal*, 115(502):C32–C61.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67.

Hui Zou, T. H. and Tibshirani, R. (1996). On the 'Degrees of Freedom' of the Lasso. *The Annals of Statistics*, 35(5):2173–2192.

Ishwaran, H. (1970). Spike and Slab Variable Selection: Frequentist and Bayesian Strategies. *The Annals of Statistics*, 33(2):730–773.

Johansen, S. and Nielsen, B. An Analysis of the Indicator Saturation Estimator as a Robust Regression Estimator. In Castle, and Shephard (2009). pages 1–36.

Mohebbi, M., Vanderkam, D., Kodysh, J., Choi, H., and Kumar, S. (2011). Google Correlate Whitepaper.

Renfro, C. G. (2009). *The Practice of Econometric Theory.* Springer.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods.* New York:Springer-Verlag.

Scott, S. L. and Varian, H. R. (2014). Predicting the present with Bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(102):4–23.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Webb, G. K. (2009). Internet Search Statistics as a Source of Business Intelligence: Searches on Foreclosure as an Estimate of Actual Home Foreclosures. Issues in Information Systems.

Wu, L. and Brynjolfsson, E. (2013). The Future of Prediction: How Google Searches Foreshadow Housing Prices and Sales. Available at SSRN: http://ssrn.com/abstract=2022293.

Yiyun Zhang, R. L. and Tsai, C.-L. (2010). Regularization Parameter Selections via Generalized Information Criterion. *Journal of the American Statistical Association*, 105(489):312–323.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(102):301–320.