University of Oxford MPhil Thesis: Working Draft

Fiona Sloof

April 13, 2016

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 modeling 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. Machine learning is primarily related to finding patterns and making predictions. This means that the objective in machine learning algorithms is to, for example, minimize the predication error, not to necessarily understand how the predictions are being formed. For economists, however, there are three main uses for economic modeling: forecasting, policy and modeling. In the modeling 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" of a particular

variable.

Then, for economic modelers 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 themselves.

This goal of this thesis is to study, evaluate and compare the performance of model selection algorithms under different states of nature. 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. Epprecht et al (2013)

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, Chapter 5 is another empirical economics application, and Chapter 6 concludes.

Chapter 2

The Algorithms

This chapter outlines the automatic model selection algorithms which are tested and compared in this study. The evaluation techniques used to compare and evaluate the different algorithms are also discussed.

2.1 Autometrics

Autometrics is an application of general to specific (GETS) model selection. Following work done by Hoover and Perez, and Hendry and Krolzig, Autometrics is a third implementation of the GETS model selection procedure. GETS model selection begins with the formulation of an initial model, which includes all variables which the modeler 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. The algorithm searches 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.

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 algorithms 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.

Figure 2.1: An example of tree search

4. Tiebreaker for resulting models.

With these components in mind, Autometrics consists of five main components:

1. **The General Unrestricted Model (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 modeled. Included in the battery of tests are normality, residual correlation, residual ARCH, and in-sample Chow tests. The modeler has the choice to set the p-value for these tests, and if initial model fails, this can be adjusted. Alternatively, there is the option in Autometrics to ignore these diagnostic tests for the initial model.

   As a very simple example, consider the following. A modeler thinks that variables A, B, C and D might be relevant for modeling variable Y, but is unsure if he should include all four. (Obviously this a very simple example and in practice a modeler would be unlikely to use Autometrics to determine what the final model should be.) 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 statically well-behaved, before the algorithm will proceed. The significance level for these tests is set by the modeler and is denoted p_d.

2. **Autometrics uses a tree search to search through the model space.**

   The ideas of tree search can be more easily understood with reference to Figure 2.1 (taken from Autometrics, Doornik 2009).

   The initial model, the GUM, is estimated. T-statistics for all the potential regressors are found, and from these values, the variables are ranked according from least significant to most significant. In our example, the GUM is the model which includes variables A, B, C and D, and is depicted in the diagram as node ABCD. In our example, suppose that variable A is the least significant, followed by B, C, then D. The tree search begins by eliminating the most insignificant variable and estimating the resulting model. In this case, this corresponds to moving to node BCD. Model BCD is backtested with respect to the GUM (explained in more detail in the next section). If backtesting passes, the next most insignificant variable - in our case B - 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 when there are no more insignificant variables to remove. 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 our example, once model D is

estimated and tested, the algorithm backtracks to model CD, then continues by estimating model C. The algorithm would then backtrack to model BCD (since all subbranches from model CD have now been explored) and continue by eliminating variable C, since after B, this is the most insignificant 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 infeasible very quickly. Thus, there are several techniques Autometrics utilizes to improve on its computational efficiency: namely 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 our 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.

Bunching
This occurs when are 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 equals 1. In our 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.

Chopping
If a variable is highly insignificant, then we can ignore all models which include it. In our example, if A is sufficiently insignificant, then we can ignore estimating any model which includes it.

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. We can bunch these variables together, delete them, and continue along the resulting path.

To summarize, Autometrics uses tree search to search the the model space, systematically deleting variables and estimating the resulting models. It uses pruning, bunching, chopping and model contrasts in order to ignore certain branches of the tree, which greatly improves its computational efficiency.

3. **Backtesting with respect to the GUM and diagnostic tests**

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 via a simple F test, and the significance level is set by the modeler. If backtesting fails it means that the eliminated variable contains non-negligible information about the variable being modeled. Models extending from a subbranch which fails with backtesting therefore are not considered.

Diagnostic testing is conducted differently, as it is computationally expensive. Conducting diagnostic testing on each estimated model is 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.

4. **Iteration until convergence**

   Once the tree search is done, there are likely to be multiple terminal models. To select the final model, Autometrics uses an iterative procedure. 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 the GUM. The same tree search and testing procedure as outlined above is then performed, with the 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.

Other features of the Autometrics algorithm

Presearch

There is an optional presearch which can be conducted before the initial GUM is set. The presearch includes lag reduction and variable reduction. The algorithm 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.

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. It is based on the same ideas of ridge regression, however in ridge regression the objective function is such that coefficients are shrunk towards zero, while never actually equaling 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, 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, then, is how to choose λ. Generally packages that implement the lasso will output 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 a number of studies evaluating these techniques with varying results. In this study, cross-validation is used.

2.3 Bayesian Structural Time Series (BSTS)

BSTS is a system developed by Steven Scott (a statistician at Google), with the purpose of now-casting and short-term forecasting when there are many potential regressors. 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 may possible 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. BSTS uses state space representation to accomplish this. Using the notation of Durbin and Koopman, a structural time series model is given by the following two equations:

$$\begin{aligned} y_t &= Z_t^T \alpha_t + \epsilon_t \qquad & \epsilon_t &\sim N(0, H_t) \\ \alpha_{t+1} &= T_t \alpha + R_t \eta_t \qquad & \eta_t &\sim N(0, Q_t) \end{aligned}$$

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 is related to the latent state (which is unobserved). 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. Z_t, T_t and R_t contain both known and

unknown values. This state space specification encompasses a large range of models. As an example, the following is the "basic structural model":

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

Here, μ_t denotes the trend, and τ_t denotes the seasonal component. Then, using the notation of Durbin and Koopman:

$$\begin{aligned} \alpha_t &= (\mu_t, \delta_t, \tau_t) \\ \eta_t &= (u_t, v_t, w_t) \end{aligned}$$

That is, the state consists of trend and seasonal components. BSTS adds a regression component to the basic structural model, so the first equation above becomes:

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

where $\mathbf{x}_t$ is a vector of variables the researcher believes should possibly be included in the model. This "basic structural model + regression component" form the model from which BSTS is based. BSTS uses Kalman filtering, smoothing and Bayesian data augmentation to determine the time series component of y_t, subtracts this from y_t and then uses spike-and-slab regression on the "remaining" part of y_t. A brief overview of the Kalman filtering, Kalman smoothing and Bayesian data augmentation techniques is below. For more detailed explanation, readers should consult Durbin and Koopman (2001).

Kalman Filtering

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

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

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 both the previously found distribution, the new data point, and the variance of each process. Estimates for the α_t's 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_t's 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_t's, the prediction errors denoted by v_t (calculated from $v_t = y_t - \alpha_t$), the prediction error variance denoted by F_t, and the variance of the estimated α_t's denoted by P_t.

Kalman Smoothing

After the Kalman filter has been applied, Kalman smoothing works to smooth out the estimates of the α_t's. Kalman smoothing finds new distributions of $\alpha_1, \alpha_2, ..., \alpha_n$ using the entire sample Y_n. This is in contrast to Kalman 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 $p(\alpha_t|y_{1:n})$.

For a much more detailed explanation of Kalman filtering and smoothing, see Durbin and Koopman (2001).

Bayesian Data Augmentation

BSTS requires simulated values of the state. Bayesian data augmentation makes it possible to simulate from the distributions derived from Kalman filtering and smoothing. It is not possible to draw α_t's 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_t's.

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 the y_t. As detailed in the criticism section, simply subtracting the times series component is theoretically flawed, and results in biased estimates of he regression parameters.

2.3.2 Regression component

The regression component of the algorithm uses Bayesian parameter estimation techniques. As a quick review of these ideas and terminology important in Bayesian parameter estimation:

- The techniques are fundamentally based on Baye's formula. If θ are the parameters that need to be estimated from some data d, then Baye's formula states:

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

  The idea is that by setting priors which reflect beliefs about the distribution of the parameters, Baye's formula can be used to update these beliefs and obtain a posterior distribution using the actual realized data.

- Therefore, to use these techniques, we need to set priors for the parameters of interest. Often, these are set somewhat ambiguously (and this is where Bayesians face criticism and scrutiny)

- Once we have set the priors and obtained the posteriors, we can infer parameter estimates - i.e. 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 very desirable feature of priors.

BSTS uses a spike and slab prior on the regression coefficients, which induces sparsity in the model. 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_j's. 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:

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

It is up to the researcher to set the priors for $p(\gamma)$, $p(\sigma_\epsilon^2|\gamma)$ and $p(\beta_\gamma|\gamma, \sigma_\epsilon^2)$. Note that the γ index is used to indicate that it refers only to the parameters associated with the variables where $\gamma_k = 1$. In the BSTS package, default priors are built in, and can be changed if the researcher wishes to do so. The three priors that must be set are described here:

1. The probability $p(\gamma)$ is the chance that a particular combination of variables are included 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_k is the probability that a particular variable is included in the model, or $p(\gamma_n = 1) = \pi_k$, 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 $\pi_k = \pi$ for all k. When this is the case, π_n is the proportion of all possible predicators that are expected in the final model. The default in BSTS is $\pi_k = \pi = 0.5$. This setting, however, allows flexibility if the researcher has reason to believe that a particular variable almost certainly should be included in the model. In this case, the researcher would set the $P(\gamma_k = 1)$ close to 1. The distribution of γ is the "spike" because it sets the probability that $\beta_k = 0$ at some positive value (resulting in a sparse model). The default in BSTS is $\pi_k = \pi = 0.5$.

2. BSTS sets the conditional variance prior as the conjugate:

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

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. Then, using the following formula, the values of ss and v can be determined:

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

Here, s_y^2 is the marginal variance of the dependent variable. There is a slight problem with using s_y^2 in the prior: Bayesian rules mean that priors should not rely on the data. The authors of BSTS contend that this is not a problem, however. The defaults in BSTS are $v = 0.01$ and $R^2 = 0.5$.

3. 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_k's, as well as the the information matrix Ω_γ. The default in BSTS sets $b_\gamma = 0$ and sets the information matrix as $\Omega^{-1} = \kappa X^T X$, where κ is a weighting term.

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

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

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

The sufficient statistics to calculate these distributions are:

$$V_\gamma^{-1} = (\mathbf{X}^T\mathbf{X})_\gamma + \Omega_\gamma^{-1}$$
$$N = v + n$$
$$\tilde{\beta_\gamma} = (V_\gamma^{-1})^{-1}(\mathbf{X}_\gamma^T\mathbf{y^*} + \Omega_\gamma^{-1}b_\gamma)$$
$$SS_\gamma = ss + \mathbf{y^{*T}}\mathbf{y^*} + b_\gamma^T\Omega_\gamma^{-1}b_\gamma - \tilde{\beta_\gamma}_\gamma^T V_\gamma^{-1}\tilde{\beta}_{\gamma_\gamma}$$

The posterior distribution of γ is different because it does not come from a conjugate prior. But it can be shown that the posterior distribution of γ is:

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

C($\mathbf{y^*}$) is a normalizing constant.

To summarize, at this point the prior distributions have been described, the parameters required to be set by the researcher have been outlined, and calculated posterior distributions have been provided. The next question, then, is how these posteriors are actually used in simulations.

2.3.3 MCMC in Bayesian Parameter Estimation

The goal is to run simulations to determine the parameters in the model; and there are many of them. There are a number which are part of the time series component - and thus the number and nature of these depends on what exactly is included in the state component. The other parameters pertain the regression component. The BSTS algorithm can be broken down into 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.

2. Simulate β and σ_ϵ^2 from the posterior conditional probabilities. This involves several steps because the posterior probabilities are conditioned on γ. Thus, in order to draw values of β and σ_ϵ^2 we need to first simulate draws of γ. To do this BSTS uses an algorithm called stochastic search variable selection (SSVS). SVSS is a variable selection technique based on Gibbs Sampling. It involves the following:

    - From the priors, we obtain the posterior distribution of γ. We could in theory, then calculate the posterior probability of every single one of the 2^N possible subsets. However computationally this is not usually feasible. Instead, Gibbs sampling generates a sequence of "model subsets" by selecting m different draws of γ:

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

    - Because γ_k is an indicator variable, each one of the m sample draws corresponds to a different selection of the possible regressors being selected.

    - Gibbs sampling works in such a way that only subsets with a high probability will be sampled and included in the sequence. This means that the sequence gives us a good indicator of what subset should be included in the model (See Gibbs and McCulloch, 1993). In effect, each "sampling" selects a different model, and the parameters of this model are used in the next "sampling". Thus, by tabulating across the sequence, we can determine which subsets are the most likely.

    - In many cases this sequence 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. With this sequence we can easily determine which variables should be included or not in our model.

Once we have a draw for γ from Gibbs sampling, we use γ in the conditional σ_ϵ^2 and β distributions. We can then obtain draws for σ_ϵ^2 and β from their respective posterior distributions.

To summarize, in each repetition, we obtain draws of γ from SSVS and Gibbs sampling, which are used to compute the posterior distribution of σ_ϵ^2, which is used to compute the posterior distribution of β. We draw from these distributions to obtain parameter estimates of β. The inclusion probability is then the proportion of these MCMC simulations which have $\beta_n \neq 0$.

2.3.4 Critique of BSTS

Inherent Bias

Explanation of Frisch-Waugh theorem + how BSTS could be altered to fix this. Also why it is computationally likely not feasible to implement.

The researcher has a lot of decisions to make

Another issue which exists in Bayesian techniques is how exactly to set the priors. Compared to Autometrics and even lasso, BSTS is complicated and requires many subjective decisions to be made by the user. The default values seem to be set according to some arbitrary decision on the part of the developers. Even with these defaults in place, there remains a fair amount of flexibility for the researcher, which is problematic when dealing with automatic selection. In summary, the following can be chosen by the researcher:

- Prior for β
- Prior for σ_ϵ^2
- Prior for γ - if using spike and slab then the probability of inclusion for each particular possible regressors, or for increased simplicity, what the expected/desired model size should be. This somewhat defeats (?) the objective of model selection to begin with!
- Value of ss
- Value of v

  Individuals using BSTS may have limited knowledge of how the algorithm works, and even with an understanding of the techniques involved, it may not be clear how to set these distributions and values.

2.4 AutomaticStatistician

2.5 Probabilistic programming algorithm

2.6 Neural nets (?)

Chapter 3

Evaluating Model Selection Algorithms

3.1 The Monte Carlo technique

Monte Carlo 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.

3.1.1 Formulate 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:

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

for $t = 1, ..., T$.

3.1.2 Run Simulations

The second step is to perform M simulations for the specified DGP, generating M 'replications' of the dependent variable, $y_t = (y_1, ..., y_T)$. There are two options for drawing the regressors. 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 ϵ_t is taken.

3.1.3 Formulate 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:

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

$$x_{k,t} \sim IN(0, 1)$$

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 regressors while the GUM includes N regressors. That is, $x_{t,1}, ..., x_{t,n}$ in the GUM are exactly the $x_{t,1}, ..., x_{t,n}$ in the DGP. If fixed regressors are used, 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 Run 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 the sets. The algorithm commences from the GUM, searches through the variables and returns the selected model. For each of the M generated y_t's 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

It is impossible to know how often you find what you you're looking for if you don't actually know what you should be looking for (better analogy here!). Analogously, evaluating automatic model selection algorithms empirically is made difficult by the fact

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 on the accuracy with which a model selection algorithm excludes irrelevant variables and selects relevant variables. 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 β_k be the true parameter values, so that $\beta_{n+1}, ..., \beta_N = 0$. Let $\tilde{\beta}_{k,i}$ 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, $\tilde{\beta}_{k,i} = 0$. Let $1(\cdot)$ be an indicator variable with $1(\cdot) = 1$ if x_k is selected (so $\tilde{\beta}_k \neq 0$) and $1(\cdot) = 0$ if x_k is not selected (so $\tilde{\beta}_k = 0$). The retention rate, potency and gauge are defined as:

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

3.2.2 Mean Squared Errors

The means squared errors (MSEs) defined provide a measurement 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:

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

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

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

If x_k is not selected, then $\tilde{\beta}_{k,i} = 0$. The results in this study report the square root of the conditional mean squared errors, denoted RCMSE. 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. The reason is that the CMSE, when interpreted alongside the gauge and potency, is much more informative than the UMSE. For example, say that a retention rate of 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 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 $\tilde{\beta}_{j,i} = 0$. 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.

3.2.3 A Note on Statistical Size

At first glance, the 'size' of each algorithm might seems like a 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 would correspond to the proportion of the time that either a relevant variable is excluded, or a irrelevant variable is included in the 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 easily could be very high. 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.

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:

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

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

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

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 modeling was, in fact, commencing from the LDGP, 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 on the LDGP 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 $N < T$:

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

Because $N < T$, the GUM can be estimated directly, resulting in coefficient estimates and standard errors for $\beta_1, ..., \beta_N$. 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 in practice it is not explicitly referred to as model selection). Therefore, the potency and RCMSE results from the 1-cut approach on the LDGP 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 a 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 LDGP 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 LDGP.

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, Lasso, and BSTS all claim to be) on the LDGP 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 LDGP. To make this more formal, consider the following DGP:

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

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

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

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

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

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 LDGP. 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 MCMC simulations that are presented in the next section provide insight into three questions: how algorithms perform relative to one another given a particular correlation structure, how algorithms perform relative to their respective benchmarks, and 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 algorith's performance within a particular correlation structure. High potencies, low gauges and low RCMSEs are all desirable in the MCMC simulations.

3.4.2 Algorithm vs. Benchmark

Evaluating the algorithm results against their benchmark by comparing the potencies, retention rates and RCMSE's in each case is straightforward. If the results from model selection are similar to the benchmark then the algorithm is performing as well could realistically be hoped for. A productive framework for thinking about this is considering costs of inference and costs of search. Costs of inference are inevitable even for a modeler who commenced from the LDGP 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 LDGP (either using 1-cut or using model selection directly on the LDGP) measures the costs of search for a particular model selection algorithm. Thus, the benchmark LDGP 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

MCMC simulations are wonderful and provide lots of interesting results. But they are just that: simulations. A valid question is what MCMC results say about model selection empirically. The difficulty with MCMC simulations is that their results are generally specific 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 that 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:

$$t_{\hat{\beta}_k} = \frac{\hat{\beta}_k}{\hat{\sigma}_{\hat{\beta}_k}} \simeq \frac{\hat{\beta}_k}{\sigma_{\hat{\beta}_k}} \sim N\left[\frac{\beta_k}{\sigma_{\hat{\beta}_k}}, 1\right] = N\big[\psi, 1\big]$$

Therefore, the formula for ψ_k is given by:

$$\psi_k = \frac{\beta_k}{\sigma_{\beta_k}}$$

The above formula makes explicit that relationship between ψ_k, the coefficient β_k and the variance $\sigma^2_{\beta_k}$. 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 LDGP, 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:

$$p_\alpha = P(\ |t_k|\ \geq\ c_\alpha \mid \psi_k\ ) = \Phi(c_\alpha - \psi_k)$$

It is possible to calculate the theoretical retention probability for every possible non-centrality and significance level. To be more explicit, the theoretical retention probability for a given non-centrality does not depend on the correlation structure of the regressors in the DGP or in the GUM, the number of variables in the GUM, or whether or not $N < T$ or $N > T$: no assumptions about the DGP are necessary to calculate the theoretical retention probability. 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 separates sets of experiments. In the following sections, the DGPs and GUMs are laid out 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, there are a number of decisions that must be taken when applying the algorithms. The following table outlines the settings for the three algorithms compared.[Still working on this table]

Algorithm	Settings	Comment
Autometrics		
Lasso		
BSTS		

Table 3.1: Algorithm settings

3.5.2 Orthogonal regressors

The first set of experiments considers cases where the regressors are orthogonal. There are three different DGP's considered, which vary according to the coefficients $\gamma, \beta_1, ..., \beta_5$.

The values of these coefficients affect the non-centralities ψ, which is a more interpretable characterization. Each of these DGPs is nested in two separate GUMs: one with the total number of regressors N 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. Let $\mathbf{x}_t' = (x_{1,t}, ..., x_{N,t})$. The DGPs take the following form:

$$y_t = \beta_0 + \gamma y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$
$$\epsilon_t \sim \text{IN}[0,1]$$
$$\mathbf{x}_t = 0.5\mathbf{x}_{t-1} + \mathbf{v}_t, \mathbf{v}_t \sim \text{IN}_\text{N}[0,\Omega]$$

with $\rho = 0.5$ and $\omega_{kk} = 1 - \rho^2 = 0.75$ and $\omega_{jk} = 0$. While this 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 breakdown as $p \to 1$, but since $p = 0.5$ 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 $p_{0.01}$. Because the regressors are perfectly orthogonal in this set of experiments, the benchmark is model selection from the LDGP via the 1-cut approach.

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
DGP 1	β	0.5	0.6	0.6	0.6	0.6	0.6
	ψ		6	6	6	6	6
	$p_{0.01}$		0.999	0.999	0.999	0.999	0.999
DGP 2	β	0.5	0.2	0.2	0.2	0.2	0.2
	ψ		2	2	2	2	2
	$p_{0.01}$		0.266	0.266	0.266	0.266	0.266
DGP 3	β	0.5	0.2	0.3	0.4	0.5	0.6
	ψ		2	3	4	5	6
	$p_{0.01}$		0.266	0.645	0.914	0.990	0.999

Table 3.2: DGP specification for experiments with orthogonal regressors

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

$$y_t = \beta_0 + \gamma y_{t-1} + \sum_{k=1}^{N} \beta_n x_{n,t} + u_t$$

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. Figures 3.1 and 3.2 graph the RCMSE results. Tables 3.7-3.9 report the results for DGP3 and Figures 3.3 and 3.4 graph the 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
LDGP	0.999	N/A	0.999	N/A

Table 3.3: Gauge and potency for DGP1

	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
LDGP	0.278	N/A	0.278	N/A

Table 3.4: Gauge and potency for DGP2

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	β	0.5	0.6	0.6	0.6	0.6	0.6
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
	LDGP	0.055	0.097	0.102	0.110	0.105	0.108
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
	LDGP	0.055	0.097	0.102	0.110	0.105	0.108

Table 3.5: CMSE for DGP1

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	β	0.5	0.6	0.6	0.6	0.6	0.6
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
	LDGP	0.091	0.119	0.126	0.161	0.146	0.157
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
	LDGP	0.091	0.119	0.126	0.161	0.146	0.157

Table 3.6: CMSE for DGP2

	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
LDGP	0.766	N/A	0.766	N/A

Table 3.7: Gauge and potency for DGP3

	ψ	2	3	4	5	6
	$p_{0.01}$	0.266	0.645	0.914	0.990	0.999
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
	LDGP	0.331	0.672	0.847	0.982	1.000
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
	LDGP	0.331	0.672	0.847	0.982	1.000

Table 3.8: Retention rates for DGP3

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	β	0.5	0.2	0.3	0.4	0.5	0.6
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
	LDGP	0.070	0.120	0.090	0.097	0.101	0.110
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
	LDGP	0.069705	0.120	0.090	0.097	0.101	0.110

Table 3.9: CMSE for DGP3

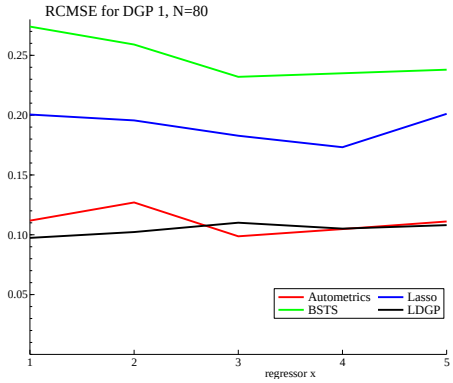

Figure 3.1: RCMSE for DGP 1

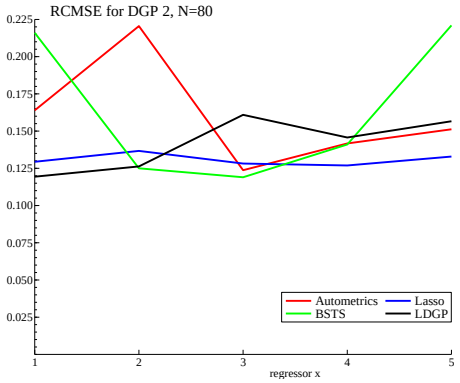

Figure 3.2: RCMSE for DGP 2

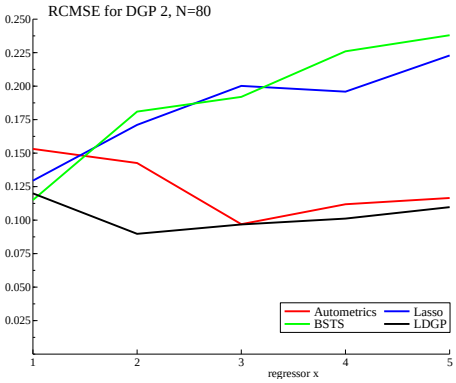

Figure 3.3: RCMSE for DGP 3

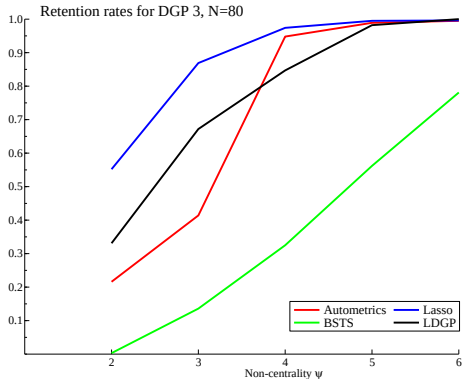

Figure 3.4: Retention rates for DGP 3

The above results show the the Lasso consistently achieves the highest potency, and has the highest retention rates across all levels of ψ. 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 fair 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. The gauge for Lasso is generally very high and in the range of $0.12 - 0.22$, 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 of close to the the significance level $\alpha = 0.01$, however in DGP2 with $\psi = 2$, the gauge is slightly higher at 0.029.

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 struggled as ψ rises. Results are mixed in 3.2, where there is a low non-centrality of $\psi = 2$.

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 low 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 all over the place). This means the costs of inference are high in Lasso and BSTS.

The four biggest takeaways from this set of experiments are:

1. Autometrics consistently has a gauge close to the significance level and a retention rates/potencies close to the theoretical retention probabilities.
2. The Lasso scores high for successfully retaining irrelevant 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 Lasso and BSTS.

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 $\gamma, \beta_1, ..., \beta_5$ and therefore their respective non-centralities. Note that $\beta_1, ..., \beta_5$ have been chosen so that the non-centralities in this set of experiments are inline with those in the first set of experiments, making the results more comparable. As in the first set of experiments, these 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:

$$y_t = \beta_0 + \gamma y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$
$$\epsilon_t \sim \text{IN}[0,1]$$
$$\mathbf{x}_t \sim \text{IN}_\text{N}[0,\Omega]$$

with $\omega_{kk} = 1$, $\omega_{jk} = 0.9$ for $j, k \leq 5$ and $\omega_{jk} = 0$ elsewhere. Table 3.10 describes the two DGPs considered, from hereon referred to as DGP 4 and 5.

When the regressors are not orthogonal, as explained, simply using 1-cut selection on the LDGP is not a feasible baseline. The 'best case scenario' in this case is then, is doing selection on the LDGP 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 LDGP and selection from the GUM is therefore a measure of costs of inference.

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
DGP 4	β	0.5	0.58	0.84	1.25	1.55	1.75
	ψ		2	3	4	5	6
DGP 5	β	0.5	-0.58	0.84	-1.25	1.55	-1.75
	ψ		-2	3	-4	5	-6

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

Tables 3.11 and 3.12 report the gauge, potency and retention rates for DGP4, and include results when selecting from the LDGP for each algorithm. Tables 3.13 and 3.14 do the same for DGP5. Tables 3.15 and 3.16 report RCMSE results. 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 LDGP	0.797		0.797	
Lasso	0.993	0.108	0.993	0.077
from LDGP	0.994		0.994	
BSTS	0.690	0.000	0.677	0.000
from LDGP	0.784		0.784	

Table 3.11: Gauge and potency for DGP4

	ψ	2	3	4	5	6
	$p_{0.01}$	0.266	0.645	0.914	0.990	0.999
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
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 LDGP	**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 DGP4, including from LDGP

	N=80		N=120	
	potency	**gauge**	**potency**	**gauge**
Autometrics	0.716	0.014	0.719	0.014
from LDGP	0.775		0.775	
Lasso	0.429	0.088	0.374	0.057
from LDGP	0.986		0.986	
BSTS	0.531	0.000	0.439	0.000
from LDGP	0.675		0.675	

Table 3.13: Gauge and potency for DGP5

	ψ	-2	3	-4	5	-6
	$p_{0.01}$	0.266	0.645	0.914	0.990	0.999
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
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 LDGP	**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 DGP5, including from LDGP

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	β	0.5	0.58	0.84	1.25	1.55	1.75
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
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 LDGP	**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: CMSE for DGP4, including baseline

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
	β	0.5	-0.58	0.84	-1.25	1.55	-1.75
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
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 LDGP	**Autometrics**	0.0514	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: CMSE for DGP5, including from LDGP

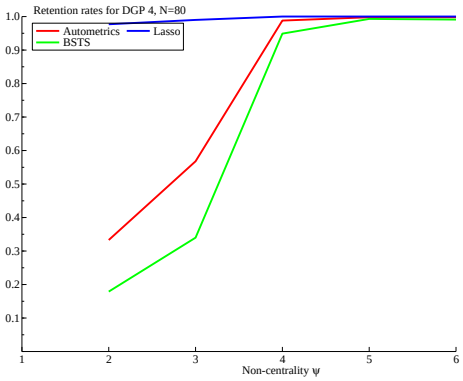

Figure 3.5: Retention rates DGP 4 N=80

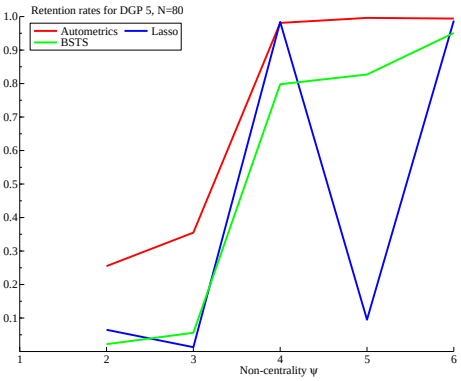

Figure 3.6: Retention Rates DGP 5 N=80

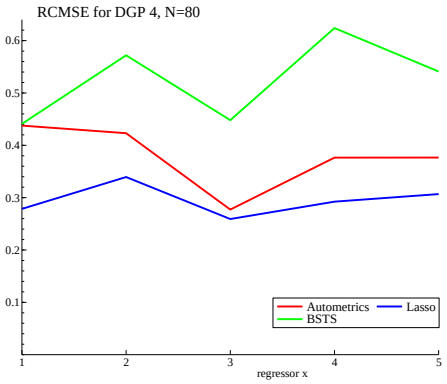

Figure 3.7: CMSE for DGP 4 N=80

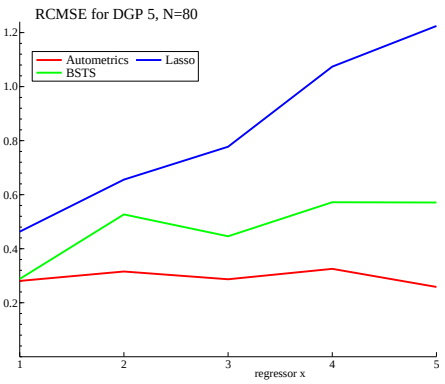

Figure 3.8: CMSE for DGP 5 N=80

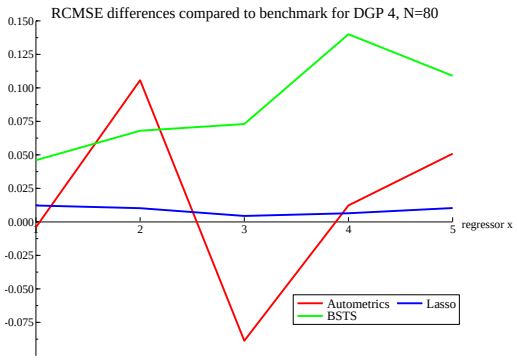

Figure 3.9: CMSE Differences for DGP 4 N=80

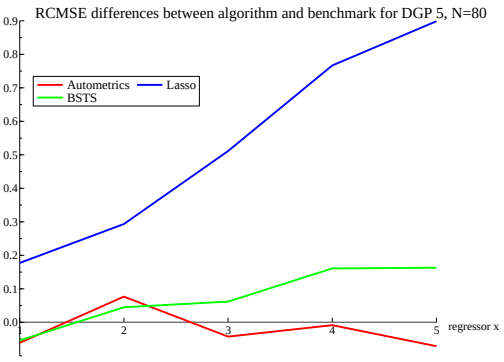

Figure 3.10: CMSE Differences for DGP 5 N=80

First considering DGP4 results independently, 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 (is there a meaningful way to interpret?), and shows there are almost no costs of search for 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).

Next looking at DGP5, it is clear that the algorithms struggle to varying degrees with alternating signs. This is particularly true for Lasso, with retention rates on regressors with $\psi > 0$ being very low, irrespective of their magnitude. The search costs of Lasso also increase with ψ. The story for the gauge is 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 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 DGP4 to DGP5 gets interesting. 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 impact how often it is selected by an algorithm; it is the magnitude of the the signal-to-noise ratio, $|\psi_k|$, 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 strange is going on in DGP5 with alternative signs. DGP4 has retention rates which are as expected, increasing with ψ. In DGP5 while both Autometrics and BSTS have higher retention rates as $|\psi_k|$ rises, Lasso struggles to select either of the regressors with $\psi_k > 0$. 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 DGP4, the Lasso has the lowest RCMSE for all variables, while it has the highest RCMSEs for all variables in DGP5. Furthermore, the RCMSEs increase with ψ for the Lasso with alternating signs.

The biggest takeaways from this second set of experiments are:

1. Selection when signs alternate produces remarkably different results (especially for 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.

4.

3.5.4 Correlation between all regressors

The third set of experiments considers several cases where there is correlation between all of the relevant regressors. There are two different DGPs considered, which again vary according to the coefficients $\gamma, \beta_1, ..., \beta_5$ and therefore their respective non-centralities. 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:

$$y_t = \beta_0 + \gamma y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \beta_5 x_{5,t} + \epsilon_t$$
$$\epsilon_t \sim \text{IN}[0,1]$$
$$\mathbf{x}_t \sim \text{IN}_\text{N}[0,\Omega]$$

with $\omega_{kk} = 1$, $\omega_{jk} = 0.8$. The following table describes the two DGPs considered, from hereon referred to as DGP 6 and 7:

		y_{t-1}	x_1	x_2	x_3	x_4	x_5
DGP 6	β	0.5	0.4	0.6	0.9	1.1	1.25
	ψ		2	3	4	5	6
DGP 7	β	0.5	-0.4	0.6	-0.9	1.1	-1.25
	ψ		-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 results are also graphed.

	N=80		**N=120**	
	potency	**gauge**	**potency**	**gauge**
Autometrics	0.736	0.016	0.718	0.015
from LDGP	0.798		0.798	
Lasso	0.909	0.182	0.894	0.131
from LDGP	0.993		0.993	
BSTS	0.653	0.004	0.631	0.005
from LDGP	0.764		0.764	

Table 3.18: Gauge and potency for DGP6

	ψ	2	3	4	5	6
	$p_{0.01}$	0.266	0.645	0.914	0.990	0.999
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
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 LDGP	**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 DGP6

		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
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
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 LDGP	**Autometrics**	0.024022	0.30655	0.21898	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: CMSE for DGP6

	N=80		**N=120**	
	potency	**gauge**	**potency**	**gauge**
Autometrics	0.718	0.014	0.707	0.014
from LDGP	0.778	N/A	0.778	N/A
Lasso	0.787	0.154	0.823	0.131
from LDGP	0.986	N/A	0.986	N/A
BSTS	0.512	0.001	0.401	0.001
from LDGP	0.6572	N/A	-0.6572	N/A

Table 3.21: Gauge and potency for DGP7

		ψ	-2	3	-4	5	-6
		$p_{0.01}$	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 LDGP	**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 DGP7

		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
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
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 LDGP	**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: CMSE for DGP7

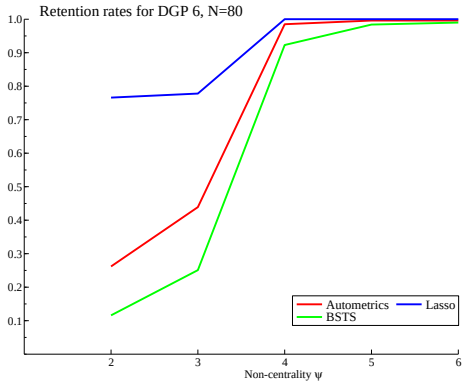

Figure 3.11: Retention rates DGP 6 N=80

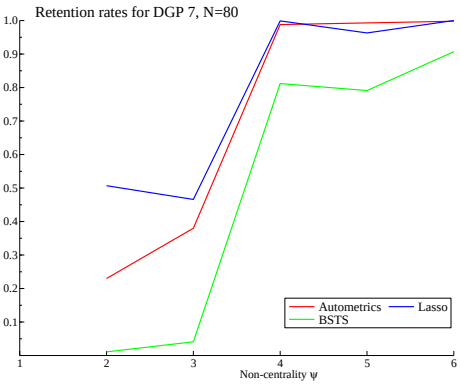

Figure 3.12: Retention Rates DGP 7 N=80

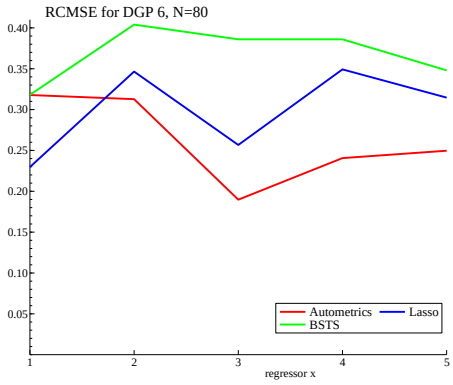

Figure 3.13: CMSE DGP 6 N=80

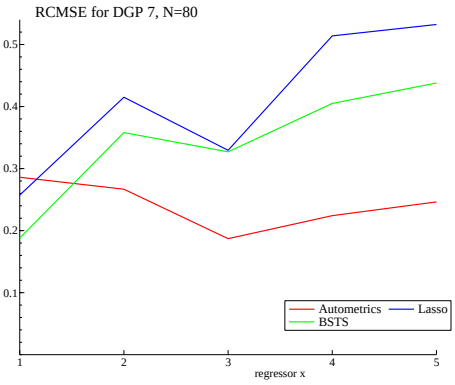

Figure 3.14: CMSE DGP 7 N=80

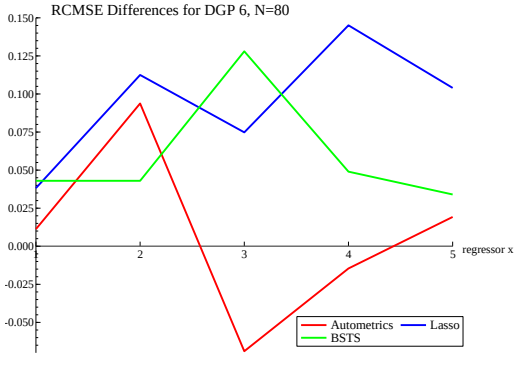

Figure 3.15: CMSE DGP 6 N=80

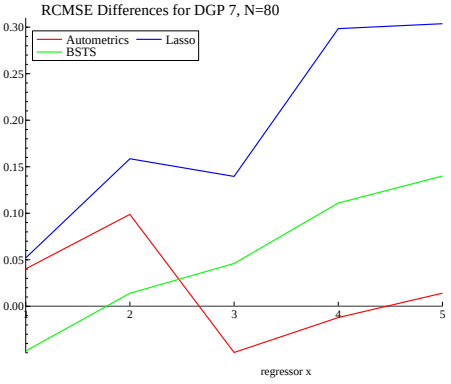

Figure 3.16: CMSE DGP 7 N=80

First looking at the results from DGP6, the Lasso is quite effective at identifying relevant variables. It 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.

Turning to the DGP7 results, the retention rates are similar to those of DGP 6 but different in a subtle but important way. Both Lasso and BSTS struggle to pick up variables with $\psi_k > 0$. Similarly, Lasso struggles to estimate the parameters on variables with $\psi_k > 0$. These difficulties are not as pronounced as in the previous set of experiments, but clear nonetheless. While 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 DGP6 and DGP7, several important differences arise. While in theory the DGPs are very similar with almost identical properties, the results vary. Lasso and BSTS again seem to to struggled 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 DGP6 and DGP7; again BSTS with the lowest gauge near 0, Lasso with the highest in the range of 0.13-0.18, and Autometrics in the middle with gauges slightly above the significance level.

The takeaways from this set of experiments are:

1. Alternating signs matters a lot of Lasso, a reasonable amount for BSTS and a little for Autometrics
2. 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 very 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 Lasso on a set of orthogonal variables, she would know that the model selected by Autometrics included α percent of the irrelevant variables, and the model selected by Lasso included 15-20 percent of the irrelevant variables. Similarly, she would know there was a 99 percent change that a variable with a signal-to-noise ratio of 4 was included in the selected model. This statements rely on orthogonality, however. 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 'settings' or correlation structures produce different results. Since a researcher can never know the true DGP and and therefore can not qualify results in this manner, ideally model selection results should not depend on what the correlation structure is.

A desirable property of model selection would be for a variable with a given 'signficance', 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 a 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 DGP3, DGP4, and DGP6 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	0.99	0.778	0.136	0.34	0.251
4	0.948	0.988	0.985	0.974	1	1	0.325	0.949	0.923
5	0.989	0.998	0.996	0.995	1	1	0.562	0.993	0.984
6	0.995	0.998	0.996	0.996	1	1	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: 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	0.24	0.255	0.23	0.608	0.065	0.507	0.007	0.022	0.011
3	0.476	0.355	0.38	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	0.095	0.963	0.943	0.827	0.791
-6	0.996	0.994	0.998	0.998	0.988	1	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: 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, Lasso picks up variable 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 Lasso experiments is also remarkably varied, ranging between 0.09 and 0.18. These discrepancies and their implications are huge. 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 MCMC 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 are decidedly not even in the upper left quadrant. In fact, Lasso has points all over the graph. While BSTS consistently achieves a low gauge, it fairs poorly when it comes to potency. Autometrics points, on the other hand, are grouped together closely. While it may not always have as high a potency as Lasso, or as low a gauge as BSTS, it does reasonably well in both and perhaps most importantly it is consistent in the sense that the correlation strucuture of the regressors has no bearing on the results.

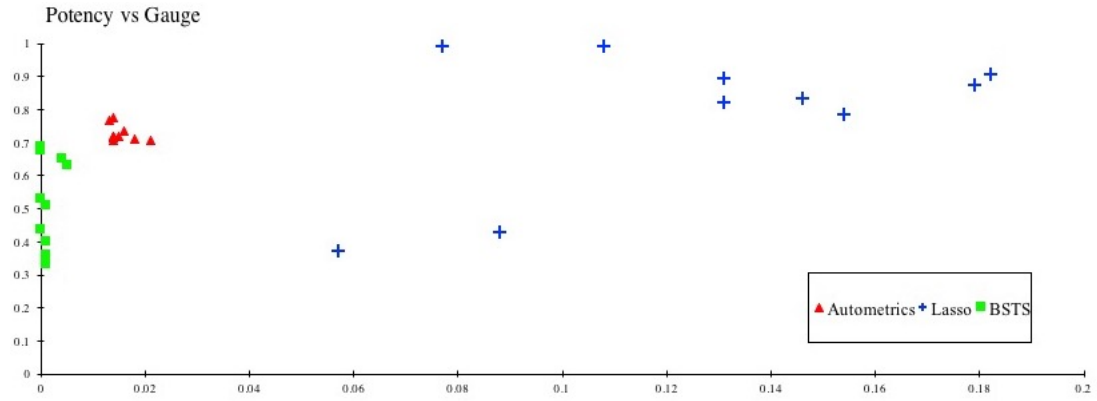

Figure 3.17: Scatter plot of gauge vs. potency across all experiments

What do the MCMC mean for 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 MCMC simulations are so important and useful, and in particular why analyzing what happens when correlation structures change is vital. Because a modeler 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 modeler 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 as discussed earlier, but studies show that this a cost-less way of improving parameter accuracy)
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 immensely.

Lasso

1. There can be anywhere in the range of 8-20% of the total number of irrelevant variables included in the selected model.
2. If all important variables in the model happen to be relevant to the dependent variables in the same 'direction' (i.e. all have matter in either a positive or negative way for the dependent variable) , then 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 percent.
4. Parameter estimates become increasingly inaccurate the more 'significant' a variable is.

BSTS

1. There are almost no irrelevant variables (around 0.1 percent) included in the selected model.

Chapter 4

An Application: Using Automatic Model Selection for Nowcasting

In this section, 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. This section is included to show an interesting application of the algorithms, 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. Every second, 40 000 Google searches are processed. 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 many 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 how unemployment benefits are. Google is also one of the first places they would turn to being searching for a new job. Thus, Google search queries potentially tell a 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.

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.

4.1.1 Google Flu Trends

One of the first ways in which the information contained in Google search query data was harnessed and put to use 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. This data is the 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 huge; 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, rather than finding out in the middle of it. Google developed the Google Flu Trends 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. 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 algorithms for their 'model selecting ability' using the accuracy with with they are capable of nowcasting in these sections should be done cautiously, if at all.

4.2 The Data

Nowcasting using automatic model seleciton commences in a similar way to the 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

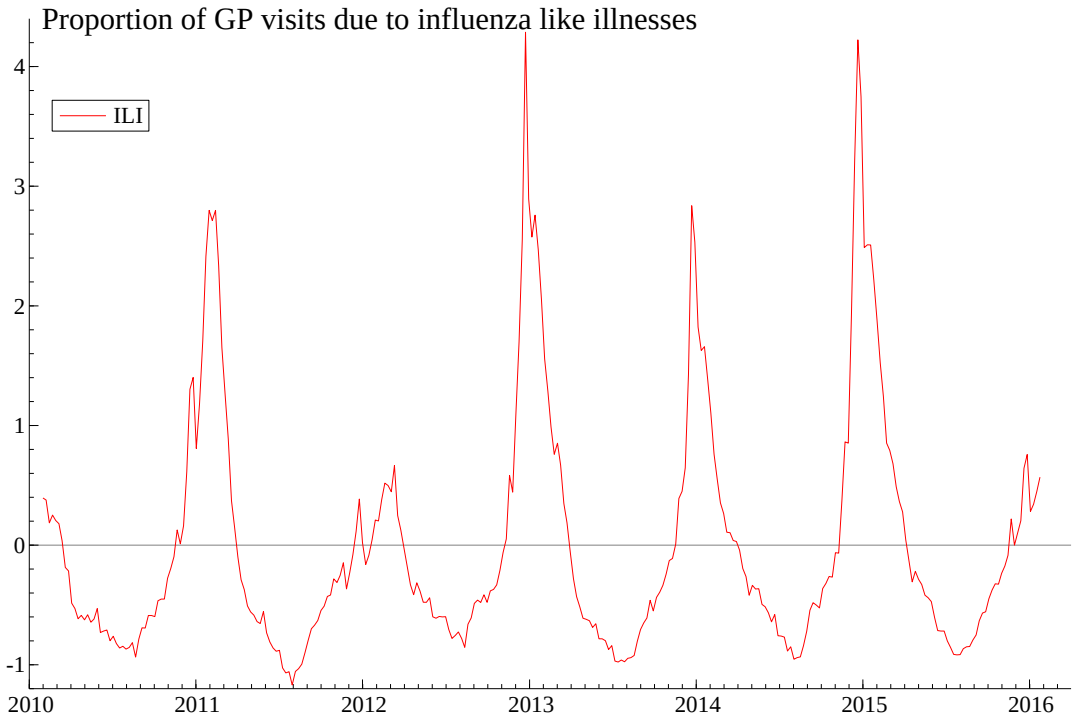

Figure 4.1: Proportion of GP visits due to influenza like illness

what data is available at time t which may be relevant for nowcasting ILI. Here, the past ILI values, along with the relevant Google Correlate data is explained.

4.2.1 Influenza-like-illness data

As mentioned, the CDC releases weekly data on the rate of 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 (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 until Week $t-2$ is available. Figure 4.1 plots the data from Week 5, 2010 to Week 4 2016.

4.2.2 Google correlate data

The search terms which are most correlated with ILI, called the correlates, as well as their first lags, are included in the GUM. To find these correlates, the weekly ILI time series was entered into the Google Correlate application. The sample entered into Google Correlate was different depending on the model, which will be explained in more detail below. Conveniently, like ILI, Google Correlate measures weeks from Sunday-Saturday. Google Correlate produced a list of the 100 search queries which were most correlated with ILI for the given sample, and provided the de-meaned 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 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.

4.3 Nowcasting with Autometrics, Lasso and BSTS

Nowcasting here is evaluated both using in-sample and out-of-sample results. Implementation of in-sample and out-of-sample models is relatively straightforward, as will be shown, 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 very easily. At a very basic level, nowcasts are made by drawing from the derived posterior distribution, and taking the mean across these MCMC simulations. BSTS's reliance on Kalman filtering and smoothing, however, means that it is not (at this point) capable of producing "out-of-sample" nowcasts using this technique. Therefore, only in-sample nowcasting results are reported for BSTS. (Note: Scott Steve and I are in correspondence over this - hoping to have more clarification).

4.3.1 In-sample Nowcasting

The in-sample nowcasting for Autometrics and Lasso used the entire sample period to estimate a model. The fitted values from this model are the nowcasts. The sample period was from week 5, 2010 to week 4, 2016 (The week beginning January 31 2010 to the week beginning on January 24 2016).

First, the GUM was identified. The GUM included any variables potentially relevant for modeling ILI. A quick glance at the graph of ILI in Figure 4.1 indicates it is likely following an auto regressive 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 weeks. Other potential regressors included the 100 correlates, and their first lags. Let $x_{1,t}, x_{2,t}, ..., x_{100,t}$ denote the search volumes of the top 100 correlates. These, their lags, and the lagged ILI_t formed the GUM:

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

Next, the automatic model selection algorithms were applied to the above GUM. Autometrics and Lasso simply return a selected model and the fitted values. BSTS works in a slightly different way (it as a Bayesian algorithm after all), and has a built in predictor which simulates draws across the calculated posterior distributions for each t. Fitted values were found by taking the mean across these draws. Figure 4.2 graphs the fitted values against the de-meaned ILI values for each of the algorithms. The mean squared errors in Table 4.1 provide a clearer picture of how the various algorithms compare.

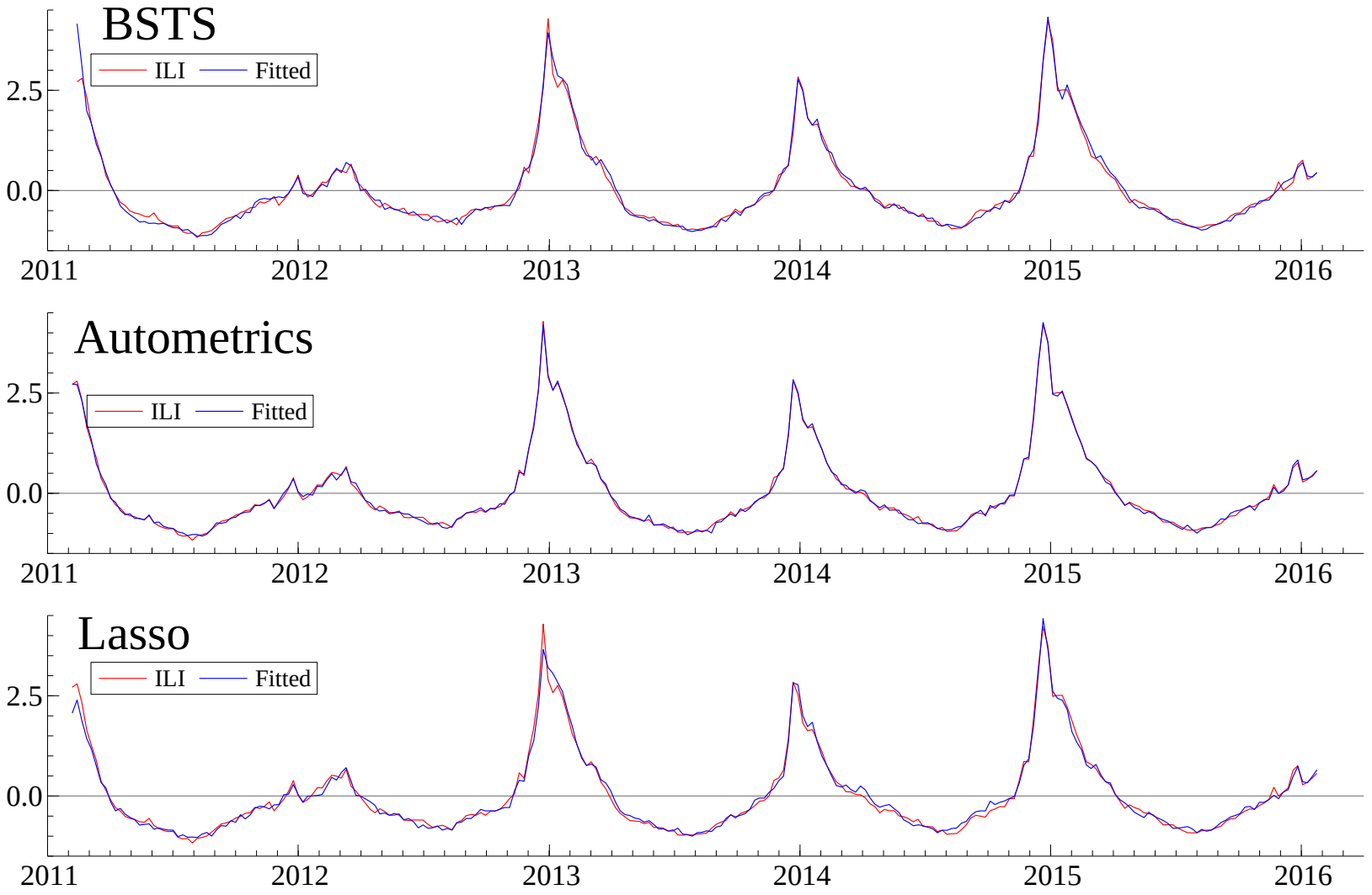

Figure 4.2: In-sample nowcasts

	MSE
Autometrics	0.003
Lasso	0.016
BSTS	0.012

Table 4.1: MSEs from in-sample nowcasting

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. More detailed output from running each of the models is provided in the Appendix (necessary?).

4.3.2 Out-of-sample nowcasting

In-sample nowcasting or forecasting where the GUM is the starting point obviously has the potential to be very accurate; the example of the extreme case of using OLS when $T = N$, and the fitted values exactly correspond to the true values comes to mind. Thus it is important to consider how nowcasts perform out-of sample. That is, how accurate a nowcast for time t is, if the model relies only on information available up until time t. To take on this sort of approach in automatic model selection, there are several decisions that need to be made. It must be decided whether the algorithm should be applied and a new model selected each time a new observation is available. Another question is whether a rolling or recursive forecast is more appropriate. An issue specific to using Google Correlates is whether a new set of correlates should be found each time an observation is added.

To find out-of-sample nowcasts for ILI, the recursive nowcasting (or equivalently an expanding window) approach was used. The entire 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). The holdout period went from Week 5, 2015 to Week 4, 2016 (the week beginning February 01, 2015 to the week beginning January 24, 2016). Nowcasts were made for each of the 52 weeks in the holdout period, with a new model being selected, and a new fitted value being found for each of the 52 nowcasts. Google correlates were found using the ILI data from Week 5, 2010 to Week 4, 2015 and thus the correlates in the GUM remained the same in each nowcast model. Thus for each t for a which a nowcast produced, the algorithms commenced from the following GUM:

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

The model selected at time t was used to find the fitted value, or nowcast for time t. The fitted values and true ILI are graphed for Autometrics and Lasso in Figure 4.3. The

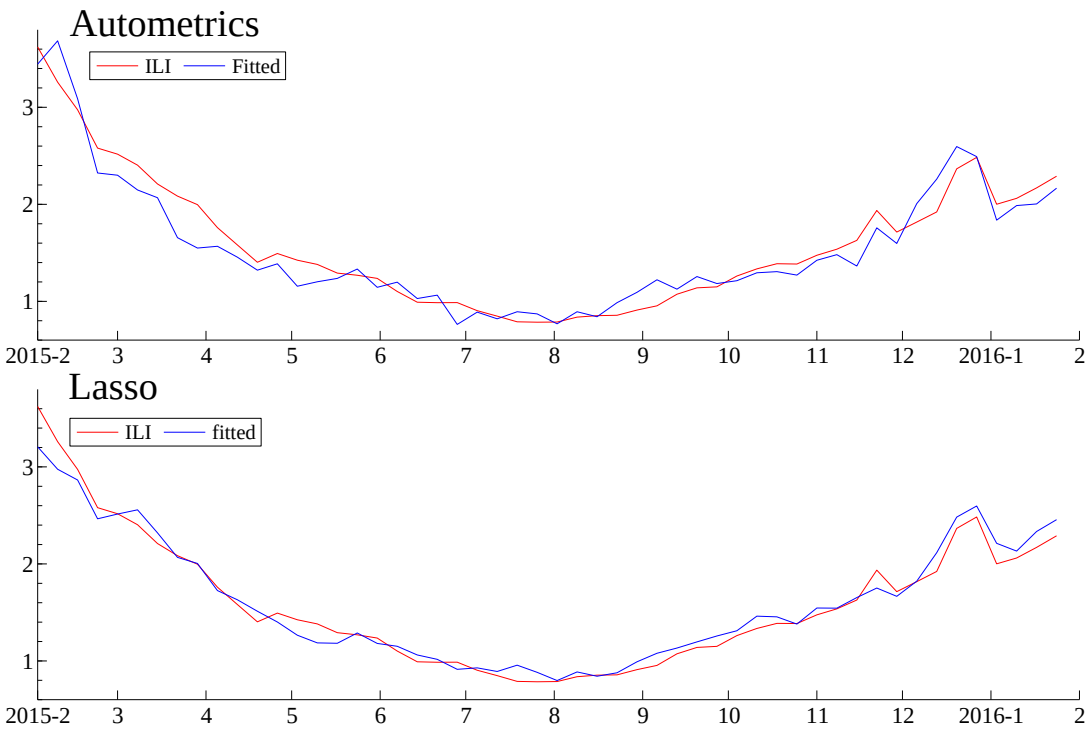

Figure 4.3: Out-of-sample Nowcasts

	MSE
Autometrics	0.032
Lasso	0.014

Table 4.2: MSEs from out-of-sample nowcasting

nowcast MSE are in Table 4.2. Both Autometrics and Lasso perform well, with Lasso achieving a slightly smaller MSE than Autometrics.

4.3.3 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. An interesting exercise is then to see how these estimates compare to the estimates produced using the algorithms studied here.

Figure 4.4 graphs the estimates produced by Google's 2014 model. Note that it is not

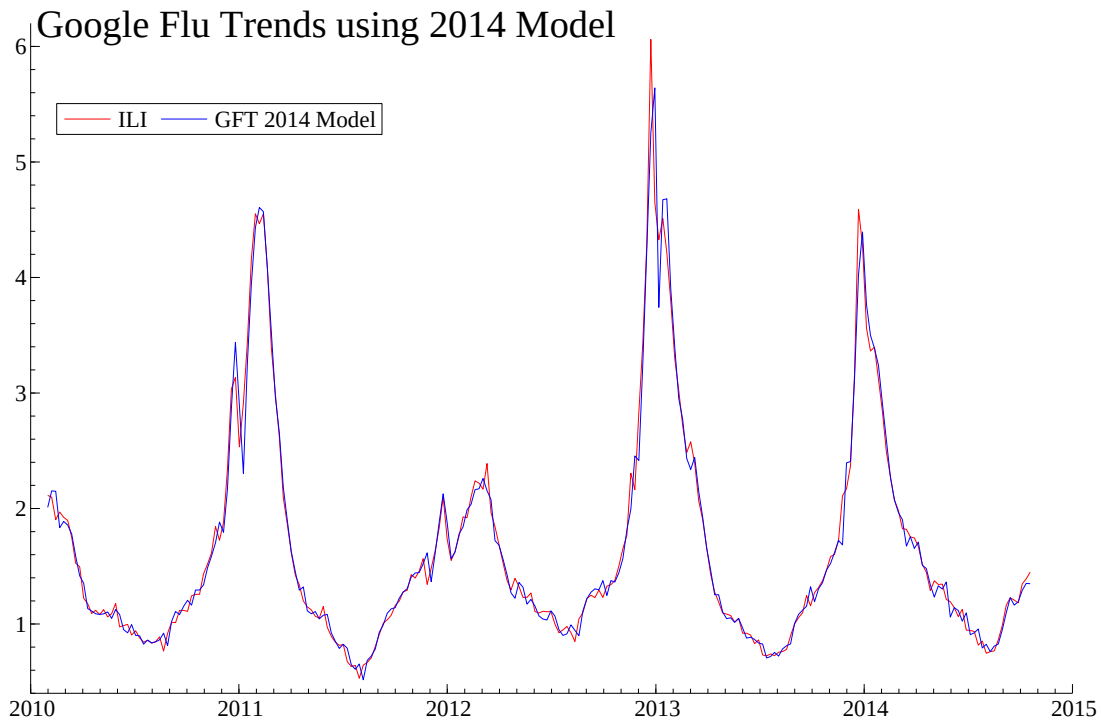

Figure 4.4: Nowcasts for 2010-2015 using Google's 2014 model

clear whether these are in-sample or out-of-sample nowcasts. The nowcast MSE is 0.021, which is much larger then Autometrics in-sample MSE and smaller than both the in-sample and out-of-sample Lasso MSE.

4.4 Nowcasting take-aways

As this chapter shows, nowcasting is an interesting, relevant and useful application of automatic model selection. 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 very useful tool for economist, statistical agencies and policy makers alike.

Chapter 5

Meeting notes

5.1 Meeting 1: March 27th, 2015

What was discussed:

- Big data has become a big topic in computer science and also in economics
- Economics and in particular macroeconomics, could stand to benefit from the increasingly available amount of data
- The difficult and still relatively new aspect of big data is how to analyse it (particularly for economic purposes)
- In particular for economists who are often interested in casual relationships, big data involves many, many (hundreds, thousands) of variables and thus standard econometric techniques do not necessarily apply
- Machine learning is increasingly becoming an important tool in the economist (and specifically econometrician) toolkit
- There are differing ideas on how to analyse big data - i.e. Autometrics vs. other ML techniques !
- There are a number of different machine learning techniques which are beginning to be noticed by economists. Namely, Hal Varian is a big proponent of machine learning (neural nets, pruning trees, etc.. ?Big Data: New Tricks for Econometrics? paper)
- Varian/Google do a lot with search query data and look at how it can be used as a predictor (i.e. for the flu) although there are varying opinions on how valid/accurate these techniques actually are !

- Castle, Hendry, and Doornik have a different idea as to how to deal with big data, which is embodied in Autometrics
- The Autometrics algorithm is an automatic model selection algorithm, which essentially searches through data and returns the model which best fits the data
- It uses a variety of techniques and allows for different combinations of data to return the best possible model
- Autometrics uses a technique which allows you to have N¿T candidate variables
- Autometrics is different from other machine learning techniques - much debate over which is better
- It would be useful to test Autometrics against other machine learning techniques (particularly those that Hal Varian likes)
- Felix Pretis has a ?big data? data set which Autometrics could analyse and could also be analysed via other machine learning algorithms
- It would be interesting to evaluate, one way or another, which algorithm is ?better?
- Perhaps could compare the different approaches by comparing respective nowcasts

Thesis Idea/General Outline:

- My thesis will examine the Autometrics algorithm in comparison to other ML algorithms
- Will most likely use Felix Pretis?s data set which will require serious manipulation/ cleaning to become useable, but should be sufficient to analyse via the different techniques
- The aim will be to use both (or multiple) techniques to analyse the data set and produce nowcasts/forecasts/predictions (or some other measure which will allow for comparison between the different techniques)
- Possible ML learning techniques include the ones outlined in Varian?s paper
- Would be great to gain access to ML algorithms directly from companies in Silicon Valley (i.e. an old colleague who works as for a data analysis company)
- This colleague who is very knowledgable on ML has agreed to at least be a resource and will help me with the ML side of things (even if he can?t give up hisalgorithms)
- There are also lots of open source algorithms available online which will likely prove very useful

Preliminaries/First Steps:

- Solid understanding of how Autometrics works, specifically how it deals with structural breaks, small effects, how it can be used for nowcasting

- Understanding of other ML techniques and how they differ from Autometrics!
- Coding skills required to clean the data
- Actually cleaning the data so that it is in an appropriate format (i.e. so that we can test out Autometrics and other algorithms)

Timeline:

- I would like to take the summer to do a fair amount of reading to gain a comprehensive understanding of autometrics and other ML algorithms
- I think the summer would also be a good time to start cleaning and getting a feel for the data so that I can begin my analysis in Michaelmas next year!
- Basically, I would like to use the summer to both gain a thorough amount of background knowledge, as well as do most of the ?dirty work? which will be required to undertake this research !

5.2 Meeting 2: October 22nd, 2015

Data Update Generated Data

- Ox code is ready and running and able to be adjusted for multiple scenarios: i.e. structural breaks, different numbers of regressors
- Based on code from Strucutural Break paper
- This data will be main focus of thesis and what I will focus on first

Google Flu Data

- Downloaded from Google Correlate application
- While Google has never explicitly provided the algorithm they used for Google Flu Trends, the Nature paper describes the method used
- From this paper, the data required are search volumes of the search queries most correlated with the Influenza Like Illness statistic published bythe Centre for Disease Control in America
- This data is downloaded and ready to use

Alogirthms update

- Three definite algorithms to be used for sure: Autometrics, Lasso, Bayesian Structural Time Series
- Autometrics: running through OxMetrics
- Lasso: running through R

- Bayesian Structural Time Series: running through R
- Two other possibilities: Bayesian technique from Nature paper and possibly neural networks for time series
- Bayesian techniques from Nature paper by Ghahramani is likely quite easy to include as another method
- Jennie has a contact who is working on neural networks for time series: will provide his information

Current plan

- begin analysis: starting with generated data and Autometrics
- Goal: to have analysis + first draft of write up of generated data completed by end of Christmas break

5.3 Meeting 3: November 5th, 2015

- General discussion of how to consolidate everything I've been doing
- Decided on formulation of several different DGP's to simulate
  - Number of observations constant across simulations at T=100
  - With respect to number of regressors: one case with N=80 (N<T), one case with N=120 (N>T)
  - Number of relevant regressors can remain fixed in each experiment at $n = 5$
  - Value of β coefficients: variations should include:
    - $\beta_1 = ... = \beta_5 = 0.2$
    - $\beta_1 = ... = \beta_5 = 0.6$
    - $\beta_1 = 0.2, \beta_2 = 0.3, \beta_3 = 0.4, \beta_4 = 0.5, \beta_6 = 0.6$
  - Significance level should be no greater than 0.01
  - Should consider cases with both orthogonal and correlated regressors
  - Also should consider structural breaks: but leave this for now
- For next week: consolidate everything that I've done + results

5.4 Meeting 4: November 19th, 2015

- Discussion of results of Autometrics and LASSO - in simulations Autometrics performs far better than LASSO and gauge and potency are approximately what theory would expect them to be
- Discussion over what to set lambda to in LASSO - consensus that it really doesn't matter because it won't drastically change results and most people don't know enough about it anyways to make it worth worrying about
- Apparently LASSO can be run in Ox - can likely find code from online textbook - will look into this
- Up next: get BTST up and running + finish up LASSO on initial experiments
- David mentioned another data set by a guy at Oxford - Steve Bro...?? This could potentially be an excellent economic application of what I'm working on
- Results from Autometrics and LASSO are at the end of this document

5.5 Meeting 4: November 19th, 2015

- Discussion of results of Autometrics and LASSO - in simulations Autometrics performs far better than LASSO and gauge and potency are approximately what theory would expect them to be
- Discussion over what to set lambda to in LASSO - consensus that it really doesn't matter because it won't drastically change results and most people don't know enough about it anyways to make it worth worrying about
- Apparently LASSO can be run in Ox - can likely find code from online textbook - will look into this
- Up next: get BTST up and running + finish up LASSO on initial experiments
- David mentioned another data set by a guy at Oxford - Steve Bro...?? This could potentially be an excellent economic application of what I'm working on
- Results from Autometrics and LASSO are at the end of this document

5.6 Meeting 7: February 4th, 2016

- Discussion of results in latest draft included a number of corrections and suggestions.
- In the tables for retention rates, gauge and potency:

  - Non centralities are being calculated incorrectly: I was using T instead of N and thus getting different results for $N < T$ and $N > T$. We went over the formula and this will be corrected.
  - Should possibly change the β coefficients so that the non-centralities are more easily understood - i.e. 2, 3, 4,...: there are several ways to do this. The ρ used to generated the regressors (in $x_t = \rho x_{t-1} + v_t$) makes things slightly more complicated - but still something to consider. Non-centralities of 2, 2.5, 3... would be more inline with what is published in the existing literature.
  - Theoretical retention probabilities should be included in the tables for each non-centrality .
  - Need to explicit outline how selection from LDGP is being done (i.e. explain one-cut selection).
  - Also need to calculate theoretical retention probabilities with selection from LDGP
  - We discussed that selection from the LDGP when the regressors are non-orthogonal is not a useful benchmark. Instead I will include the results of selection from LDGP using Autometrics, Lasso and BSTS (without changing the settings in BSTS).
- I need to include a discussion on the costs of inference and costs of search across the methods.
- A new plot which shows potency against gauge would be a nice way to present the results: suspect this would result in clusters for each of the three algorithms.
- Early results from non-orthogonal case are very strange:
  - With alternating signs ($\beta_1 = -0.2$, $\beta_2 = 0.3$, $\beta_3 = -0.4$, $\beta_4 = 0.5$, $\beta_5 = -0.6$) and high correlation across the relevant regressors (covariance between regressors is $\rho = 0.99$) both Lasso and Autometrics are doing the opposite of what we'd expect. The retention rate of x_1 (the variable with the smallest non-centrality) is far higher than any of the other variables - in fact, the retention rate in general is decreasing as the non-centrality is increasing.
  - In BSTS, the retention rate of variables with negative coefficients is higher than those with positive coefficients.
  - We were all rather confused by these results - it could be something funny going on in the Autometrics and Lasso programs themselves. I will investigate some more. David - perhaps you have an explanation?
- Will also run some simulations where all N variables are correlated.
- Discussion of MSE tables:

  - Will include a graph based on David's suggestion: Jennie suggested average RMSE for retained irrelevant variables across the different methods
  - The bench mark for MSEs should be the MSEs from the LDGP - will calculate and include this in the table.
  - While using bias correction is perhaps not necessary (if I did it for Autometrics, I could do something similar for Lasso - methods do exists and it wouldn't seem 'fair' to only do it for Autometrics) a discussion of bias correction should at least be included. This includes the direction of bias, what we expect the distribution of the MSEs to be as well as graphs illustrating this.
  - Jennie suggested only including conditional MSE - the unconditional MSE aren't all that useful.
- Should also start thinking about exactly which tables and graphs should be included in final document. An appendix would possibly be useful.
- We also discussed that there are many, many simulations I could do in theory - but perhaps getting onto the empirical section would be useful. This could provide direction for other simulations which might be useful.

Chapter 6

Results