ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

BACHELOR THESIS: INTERNATIONAL BACHELOR ECONOMETRICS AND OPERATIONS RESEARCH

Model Selection Vector Autoregressive Processes via Partial Correlation adjusted Adaptive Lasso

Stein Kooijman (472610)

Supervisor: Schnucker, A.M.

Second assessor: Klooster, J.

Date final version: 5 July 2020

Abstract

When estimating vector autoregressive (VAR) models, it is common to find the dimensionality of the estimation to be high. Having accurate variable selection methods is one way to tackle this problem. Hsu et al. (2008) show the relative performance of a number of variable selection methods in this context and this paper seeks to extend this by considering the Adaptive Lasso and a novel augmented Adaptive Lasso by using the partial correlation function. The Adaptive methods have higher predictive performance by inducing low bias on the large parameters but only in presence of low dimensionality and high relative parameter size differences. They trade the low bias in the large parameters for an increase in bias in the small parameters, which can result in the more frequent wrong exclusion of variables. Furthermore, the novel Adaptive Lasso is a competitive method to the original Adaptive Lasso in predictive performance and substructure selection, by reducing the penalisation on variables with high partial correlation.

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction **2**

2 Methodology **5**

2.1 VAR model 6

2.2 Penalised Likelihood Lasso 6

2.3 Adaptive Methods 7

2.3.1 Adaptive Lasso 7

2.3.2 Adaptive Lasso-n 7

2.4 Order Selection with AIC and BIC 9

2.5 Backward Elimination (BE) and Forward Selection (FS) 9

2.6 Hybrid methods 10

2.7 Performance Measures 10

2.8 Cross-validation hyperparameters 12

3 Data **12**

3.1 Application 13

3.2 Simulation 13

4 Results **15**

4.1 Model 1 15

4.2 Model 2 16

4.3 Application 17

4.3.1 Comparison with results Hsu et al. (2008) 18

5 Conclusion **20**

5.1 Further Research 21

6 Appendix **24**

6.1 Generated Parameters 24

6.2 Python Code Details 25

6.3 R Code Details 25

1 Introduction

In machine learning, high predictive accuracy, discovering the relevant variables and model forms are among the most important (Zou, 2006). Variable selection is therefore of great importance, not only to understand which variables play a role in the determination of the dependent variable but also to improve predictive accuracy. When the true model is sparse with respect to the variables considered, accurate variable selection methods are crucial.

Vector auto regressive (VAR) models have a wide range of applications but are one of the main starting points for any macro-economic model. The number of explanatory variables considered in a VAR model is equal to k^2p+k, where k is the amount of time series considered and p is the chosen lag order of the VAR model. Due to the quadratic relationship of k, the dimensionality rises quickly. Because of this, it is common that after selecting the order there is a need for variable selection due to the sparsity with respect to the variables considered. As a result of this problem, Hsu et al. (2008) have applied multiple variable selection methods on VAR models to be able to discover with greater precision the relevant variables and increase predictive accuracy. These include the conventional methods such as the Forward Selection (FS), Backward Elimnation (BE) but also include the more advanced such as using regularization terms in Lasso. Zou (2006) has expanded on the idea of the Lasso by introducing the popular Adaptive Lasso. This paper expands on the research of Hsu et al. (2008), by considering the Adaptive Lasso and a novel augmentation of it with the goal to reduce potential problems of the Adaptive Lasso.

The aim of this paper is to augment the Adaptive Lasso, as introduced by Zou (2006), using partial correlation. The motivation for augmenting the Adaptive Lasso is that it focuses on reducing the bias of larger parameters relative to the smaller parameters. This approach is not only seen here but also for example in the Smoothly Clipped Absolute Deviation (SCAD) method by Fan and Li (2001). While these can result in higher predictive performance, asymptotic properties and higher sparsity, this paper shows the negative implications for the bias of the smaller parameters and the resulting substructure selection. The augmentation this paper proposes, maintains these good qualities of the Adaptive Lasso and focuses to reduce the bias in the smaller parameters. This results in more accurate substructure selection and most prominently help the Adaptive Lasso not wrongly eliminate variables from the model. The augmentation reduces the weights for the variables with a high partial correlation with the dependent variable. This allows small variables to have significantly reduced penalisation and therefore attain less bias. This augmentation will be referred to as the Adaptive Lasso-n.

As a non-expert in the field of variable selection methods an initial approach would be to try all different combinations of variables that are being considered and picking the best one according to some criterion. Such an elementary method would provide the most accurate results however due to computational limitations, this is not always practical. For example, the number of possible combinations in VAR models is equal to 2^{k^2p}. This calls for more computational efficient methods that do not require such an exhaustive search of all combinations but can systemically add or eliminate certain variables from the model. From here, one finds the most conventional methods, namely the Backward Elimination (BE) and Forward Selection (FS). The BE starts from a full model and as the name suggests eliminates certain variables iteratively according to an information criterion. The FS on the other hand starts from an empty model and selects certain variables to be included in the method. The information criterion used for selecting or eliminating variables from the model can have a large impact on the final outcome of the model. The most common are the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).

The AIC is known for selecting variables to maximimize the predictive performance of the model and is therefore more lenient on selecting or retaining (respectively for FS and BE) variables in the model. The BIC however is more strict on which variables to select or retain in the model. Due to this, it is most commonly used for descriptive regressions and therefore useful for finding the true data generating process (DGP), These features result in the AIC selecting higher orders than the BIC. Besides the information criterion, the order in which variables are tested to be selected or eliminated also plays a large role in the final outcome. Sauerbrei et al. (2019) state that in the case a weak predictor is selected first by for example the FS, the coefficient might be significantly biased from 0. In addition to this, the random element of which variable is selected contributes to an increase in the mean squared error (MSE) of the predicted values (Sauerbrei et al., 2019). Therefore, methods following simultaneously estimate the coefficients and select the variables, which makes them more reliable. These methods are accompanied by criteria that are needed for an accurate comparison of their performance. First are the asymptotic oracle properties of a variable selection method, whose importance is argued by Fan and Li (2001) and Fan et al. (2004). The oracle properties state that the method, as the sample size approaches infinity, estimates the parameters without bias and chooses the correct substructure. Furthermore, Fan and Li (2001) go on to highlight three additional important features. The first one being sparsity of selection, meaning the method should be able to either select strong or screen weak predictors (or both). Secondly unbiasedness of the estimates, which are included in the model. The last condition is continuous penalisation, not having continuous penalisation causes instability in the estimates Fan and Li (2001).

Tibshirani (1996) proposed the Lasso method, which by employing shrinkage in the form of an L1 regularization method, shrinks some variables to exactly 0. Other shrinkage regularization methods such as the Ridge (L2), proposed by Hoerl and Kennard (1970), do not have this property. The Lasso imposes sparsity on the variables and has continuous penalisation. On the other hand, the Lasso, in the presence of multicolinearity or grouped variables is not able to include all of the variables, unlike the Ridge (Sauerbrei et al., 2019). Zou (2006) shows that only under certain conditions (orthogonal design or p=2 and proper choice of λ) the Lasso has the oracle properties. Furthermore, it also estimates the parameters with bias. The final disadvantage of the Lasso is that it cannot select more variables than data points noted by Sauerbrei et al. (2019). Due to these issues, new solutions were proposed based around the original Lasso concept of Tibshirani (1996).

The Adaptive Lasso introduces additional weights using pre-estimated coefficients with all variables (Ordinary Least Squares (OLS) or Ridge) to the Lasso loss function. Properly chosen weights introduce additional information to the Lasso and help it create additional sparsity while incurring less bias. In the original proposal of the method, the weights were equal to the inverted pre-estimated parameters. The increase in sparsity is due to the fact that the weights help to incur more loss in the variables with smaller coefficients. Furthermore, the larger estimates receive less penalisation and therefore are less biased. Lastly, Zou (2006) proved that the Adaptive Lasso enjoys the oracle properties. On the other hand as its penalisation is still in the form of L1, it still struggles in selecting group variables in presence of multicolinearity. This poor performance of the (Adaptive) Lasso in the presence of multicolinearity gave rise to the Elastic Net (ENET) method by Zou and Hastie (2005) and the Adaptive ENET by Zou and Zhang (2009). The (Adaptive) ENET is a weighted average of the L1 (Lasso) and L2 (Ridge) penalisation methods, in this way it enjoys the variable selection of the L1 penalisation and the grouping effect of the L2 (Ogutu et al., 2012). However, it loses sparsity and induces additional bias in comparison to the (Adaptive) Lasso.

More advanced methods such as the SCAD or the Adaptive Lasso are centered around relatively penalising absolute larger parameters less than smaller parameters. The additional penalisation on the smaller parameters results into additional bias and incorrect substructure selection. In response to this, Qian and Yang (2013) propose to augment the weights with the standard error of the parameters (while maintaining the asymptotic properties of the Adaptive Lasso). The weights now resemble the statistical significance of the parameters in contrast to the absolute size. This paper is inspired by the approach to augment the Adaptive Lasso and find a metric, which is not dependent on only the size of the absolute parameter size.

Besides the popular penalty based approaches, Bühlmann et al. (2010) have developed an approach which is based around partial correlation. By introducing a simplified PC-Algorithm they first screen variables using standard correlation and afterwards pick variables by looking at the partial correlations the variables have with the dependent variable. The paper shows that this method is consistent for variable selection.

Partial correlation shows the amount of information a certain variable adds, in addition to the variables already in the model. The idea of the Adaptive Lasso-n is therefore a combination of the partial correlation approach by Bühlmann et al. (2010) and the idea of augmenting the weights of the Adaptive Lasso by Qian and Yang (2013). By combining these approaches, the information of partial correlation is added and less emphasis is put on the absolute parameter size when penalising, as partial correlation is not directly related to it. The Adaptive Lasso-n lowers the original weights of the Adaptive Lasso when the partial correlation is high and maintains the weights when low.

This paper shows that the Adaptive Lasso and Adaptive Lasso-n improve on the Lasso only in the presence of low dimensionality and high parameter size differences. Furthermore, the Adaptive Lasso-n lowers the bias of the small parameters relative to the Adaptive Lasso consistently throughout the different data sets. However, it loses some of the sparsity properties and on occassion incurs additional bias on the large parameters. In predictive accuracy and substructure selection it is a competitive method to the Adaptive Lasso when estimating VAR models.

The details of how the methods are applied are stated in Section 2. These methods are applied on simulated data and historic data. First, the details of the simulated and historical data are discussed in Section 3. Section 4 discusses the results of these methods when applied on this data. The paper finishes with concluding remarks on the research and potential further research that can be done on this topic in Section 5.

2 Methodology

This section discusses the variable selection methods considered in this paper. First the most basic penalisation method is discussed, which is the Lasso proposed by Tibshirani (1996) in Section 2.2. Section 2.3 discusses the details of the Adaptive Lasso (Zou, 2006) and the Adaptive Lasso-n (together referred to as the Adaptive Methods). After this, two information criteria (AIC and BIC) are discussed for selecting the order of the lags in Section 2.4. Following this, in Section 2.5, are the details of how the conventional BE and FS methods are implemented. Moreover, due to the promising results in the paper of Hsu et al. (2008), hybrid methods of the latter are discussed in Section 2.6. Lastly the details about the performance measures and the cross-validation process for finding the optimal hyperparameter are in Sections 2.7 and 2.8 respectively. The notation and details of the methodology for the VAR model, AIC, BIC, BE, FS and Lasso, are based on the paper from Hsu et al. (2008). As the focus of this paper is towards the Lasso and Adaptive methods, only these will be tested in the simulation.

2.1 VAR model

This section introduces notation necessary for the description of the VAR model. The following vector $y_t = (y_{1,t}, y_{2,t}, ..., y_{k,t}) : t = 1,2,...n$ represents a k-dimensional time series, where y_t is a k-dimensional vector. A VAR model with order p (VAR(p)), is described as follows:

$$y_t = \psi + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \mu_t \tag{1}$$

Where ψ , is a k-dimensional vector giving the intercept, $\phi_i, \forall i = 1,2,...p$ is kxk dimensional coefficient matrix and the k-dimensional vector $\mu_t, \forall t = 1,2,...n$ is a white noise process with covariance matrix Σ_u. Through the following vectorisations the above VAR(p) model can be rewritten to $Y = X\beta + M$, where:

1. $Y^* = (y_{p+1}, y_{p+2}, ..., y_n), \quad Y = vec(Y^*), \quad M^* = (u_{p+1}, u_{p+2}, ..., u_n), \quad M = vec(M^*)$
2. $X_t = (1, y'_t, y'_{t-1}, ..., y'_{t-p+1})', \quad X^* = (X_p, X_{p+1}, ...X_{n-1}), \quad X = (X^*)' \otimes I_k,$
3. $B = (\psi, \phi_1, \phi_2, ..., \phi_p), \quad \beta = vec(B)$

I_k denotes the kxk dimensional identity matrix. The least squares cost function to be minimized in this case becomes the following: $(Y - X\beta)'(Y - X\beta)$.

2.2 Penalised Likelihood Lasso

Penalised likelihood regressions are structured according to equation (2), here it is assumed that the least squares is the standard (without any penalisation) loss function. Moreover, $c_\lambda(|\beta|)$ is the penalisation term that can take on many forms. The Lasso can also be seen as a regular OLS regressing, however, with the constraint: $\sum |\beta_j| < s$. You can estimate B similarly by estimating the regression in equation 2, with $c_\lambda(|\beta|)$ set as in equation (3).

$$(Y - X\beta)'(Y - X\beta) + c_\lambda(|\beta|) \tag{2}$$

Hsu et al. (2008) propose to apply the Lasso penalisation method on the VAR model described in equation (1). This results in the $c_\lambda(\beta)$ taking the following form, where λ is a hyperparameter chosen through cross-validation (discussed in Section 2.7).

$$c_\lambda(|\beta|) = \lambda \sum |\beta_j| \tag{3}$$

Hsu et al. (2008) propose two ways of implementing this method, one is applied on each individual series and the other on the entire system, these refer to the Lasso-s and Lasso-f respectively (Only Lasso-f used for simulations). Zou (2006) build on this principle to develop the Adaptive Lasso. He shows that, by adding weights to the penalisation function, it is possible to maintain continuity, achieve additional sparsity, unbiasedness and obtain the oracle properties. The execution of the methods in Section 2.2 and 2.3 are done in Python 3.7, further details on packages used for optimisation and code are in Section 6.2.

2.3 Adaptive Methods

2.3.1 Adaptive Lasso

The Adaptive Lasso as proposed by Zou (2006) can be seen in equation (4), where β is a k-dimensional vector and $\gamma > 0$. Here w_i can take many forms although the original idea was to do a 2-step approach to this penalisation function, where first the full model is estimated using an OLS regression. These estimates are then used to calculate the weights and with these weights do the second model estimation as in equation (4).

$$c_\lambda(|\beta|) = \lambda \sum_{i=1}^{k} w_i |\beta_i|, \quad w_i = \frac{1}{|\hat{\beta_{i,OLS}}|^\gamma} \tag{4}$$

Note that this method is applied only to the entire system (and not to the individual models), similar to Lasso-f.

This paper considers the effect of inaccurate estimation of the weights in finite samples, this gets amplified in the case of high dimensionality. While the Ridge penalisation Hoerl and Kennard (1970) could be used in this case, this method itself would also incur additional bias in the estimates and thus in the weights. Therefore, a solution would be to combine the the weights with information from another source thereby dampening any bias or high variance there might be in the estimates. This is for example done by augmenting the weights with partial correlation. Secondly, as discussed earlier the additional penalisation incurs additional bias in these parameters and could result in incorrect substructure selection. In the next subsection the Adaptive Lasso-n method is discussed. This method, by augmenting the weights of the Adaptive Lasso, aims to decrease the severity of the problems mentioned above.

2.3.2 Adaptive Lasso-n

This novel method is referred to in this paper as the Adaptive Lasso-n. The method applies partial correlation (and cross partial correlation) functions to the Adaptive Lasso's weights to augment them. The intuition behind the method comes from combining the penalisation and the PC-algorithm approaches. Partial correlation shows how much information a certain variable adds to the model given the parts already explained by previous variables. Thereby, using it as a criterion for penalisation the Adaptive Lasso-n can penalise variables that do not add additional information. This results in a parsimonious model and Bühlmann et al. (2010) proved that by solely using partial correlation you can have a consistent variable selection method. Furthermore, the information the partial correlation provides is not directly related to the absolute parameter size and therefore will not incur additional bias solely on the smaller parameters, which is the source of the potential problems of the Adaptive Lasso. Moreover, Qian and Yang (2013) showed that it is possible to include information from another source into the Adaptive Lasso by augmenting the weights. In the paper they used the standard error of the parameters. The Adaptive Lasso-n originated by combining this new source of information from a fundamentally different approach by Bühlmann et al. (2010) and augmenting the Adaptive Lasso with this information, in a similar way to Qian and Yang (2013).

The Adaptive Lasso-n uses the same two-step approach the Adaptive Lasso. In the first step the parameters (using OLS) and the partial correlation are estimated. The partial correlation is estimated per individual model and individual series added to those models. Consider the model for the j^{th} series, where all series are introduced as explanatory variables. Consider adding the i^{th} series as explanatory variables.

1. For a given lag order p, find the partial correlation function (cross-correlation in case of $j \neq i$) of the i^{th} series with the j^{th} series. Find the (cross) partial correlation of the p lags from the i^{th} series with respect to the j^{th} series.

2. Repeat this for all series $i = 1, 2, ..., k$ and all models $j = 1, 2, ...k$

Note that by using this procedure the partial correlation is estimated for each explanatory series's individually. For example the partial correlation function of the $i+1^{th}$ series with the dependent variable is unaffected by the i^{th} series and all previous. This is done such that the order in which the series as explanatory variables are added, does not affect any results, in contrast to for example FS. Furthermore, within each series added the lag structure is followed for the order of variables in the partial correlation. Due to the time-series nature this is a logical choice but could be changed for future research. As a result, there is an estimated partial correlation for each parameter, the resulting estimated vectorised partial correlation vector is denoted as $\hat{p}$. Using the partial correlation, the weights are calculated as follows: $w_i = \frac{1}{|\hat{\beta_{i,OLS}}|^{\gamma}}(1 - |\hat{p}_i|), \gamma > 0$, where $\hat{p}_i$ is the estimated partial correlation of the i^{th} variable in vectorized form with the dependent variable. This paper uses $\gamma = 1$, due to this being a standard approach. For further research, the γ can be treated as an additional hyperparameter subject to optimisation.

A benefit to calculating partial correlation per individual series added is that its accuracy is only dependent on the amount of data points and not dimensionality of the VAR model. Due to this the method hopes to improve the performance in finite samples and high dimensionality. Partial correlation in time series is often used by researchers when investigating the lag stucture. The weights as calculated above always remain positive, furthermore the weights for the variables with high partial correlation are lowered and are kept the same for variables with low partial correlation (relative to original Adaptive Lasso weights). As a final result, as the method uses partial correlation, it can distinguish between variables that are redundant with respect to previously added variables and which ones provide unique additional information.

2.4 Order Selection with AIC and BIC

The information criteria can be used to select the orders of the lags to include. This can be done with many different information criteria, this paper discusses using the AIC and BIC. Respectively for AIC and BIC, the order is chosen as seen below. The notation is taken from the paper of Hsu et al. (2008). The symbols $\hat{p_{aic}}$ and $\hat{p_{bic}}$ represent the order chosen using AIC and BIC respectively.

$$\hat{p_{aic}} = arg\ min_p\{n\ ln|\tilde{\Sigma_u}| + 2pk^2\}$$

$$\hat{p_{bic}} = arg\ min_p\{n\ ln|\tilde{\Sigma_u}| + ln(n)pk^2\}$$

Here $\tilde{\Sigma_u}$ represents the maximum likely hood estimate of the covariance matrix Σ_u. Furthermore n represents the sample size and p and k the order and amount of time-series considered respectively.

2.5 Backward Elimination (BE) and Forward Selection (FS)

This section introduces two conventional benchmark models. These two methods can be differently implemented based on ones choice of information criterion and order of selection for variables to be included or eliminated. The methods in this paper use the AIC. Additionally B_j denotes the $kp+1$ dimensional parameter vector associated with the j^{th} series. Furthermore, $B_{j,i}$ denotes the i^{th} parameter in this vector. Consider a set $\zeta(B_j)$, which includes all parameters associated with the j^{th} series. Lastly, $\forall m < kp+1$, let $\zeta(B_j/\{B_{j,i_1}, B_{j,i_2}, ..., B_{j,i_m}\})$ be a set, which includes all parameters except $B_{j,i_1}, B_{j,i_2}, ..., B_{j,i_m}$. The BE method is as follows:

1. Let $i = kp+1$ and the set $\upsilon = \zeta(B_j)$
2. If $IC(\upsilon/\{B_{j,i}\}) \leq IC(\upsilon)$, Update υ to $\upsilon/\{B_{j,i}\})$
3. Update i to $i-1$
4. Repeat 2-3 until $i = 0$

Note that IC is a function for any information criteria but as eluded earlier in this paper uses the AIC. The BE method starts out with a full model and systematically checks every variable to be eliminated. Secondly, the FS method is discussed. In contrast to the BE, the FS starts with an empty model and systematically goes through all variables to check whether to select them. Using the same notation as with the BE, the process is summarised in the steps below for selecting the variables of the j^{th} series:

1. $\zeta(B_j)$=$\emptyset$, i=1
2. Keep variables in $\zeta(B_j)$ and add the i^{th} variable ($B_{j,i}$) as an additional explanatory variable to $\zeta(B_j)$.
3. If $IC(\zeta(B_j)) \geq IC(\zeta(B_j)/\{B_{j,i}\})$, remove $B_{j,i}$ from $\zeta(B_j)$
4. Update i to i=i+1
5. Repeat steps 2-4 until i=kp+1

Specifically in this FS process, Hsu et al. (2008) suggested to begin with an empty model and add univariate AR series as explanatory variables of each endogenous variable considered in the VAR model. However, in order to keep the same structure for the FS as the BE, the process above is followed.

2.6 Hybrid methods

Due to the excellent performance of hybrid methods in the paper by Hsu et al. (2008), hybrid methods are also investigated. The hybrid methods considered are as follows:

1. **BE + FS**, first implement the FS method for each series then apply the BE on each series
2. **AIC+BE**, select order of VAR model using AIC (VAR($p_{\hat{AIC}}$)) then apply BE to each series.
3. **AIC+Lasso-f**, apply the Lasso-f to the VAR($p_{\hat{AIC}}$) model.
4. **AIC+Lasso-s**, apply the Lasso-s to the VAR($p_{\hat{AIC}}$) model's individual series.
5. **AIC+Adaptive Lasso**, apply the Adaptive Lasso to the VAR($p_{\hat{AIC}}$) model.
6. **AIC+Adaptive Lasso-n**, apply the Adaptive Lasso-n to the VAR($p_{\hat{AIC}}$) model.

2.7 Performance Measures

In order to evaluate the performance of the chosen methods, the following performance measures are chosen. As there is more information in the simulated data section, it is possible to include more performance measures. For the actual data the following performance measures are tracked: the maximum order selected, the proportion of parameters equal to 0 and the prediction mean square error (PMSE) for both the individual models and a normalised version for the models combined. The different PMSE will be calculated as follows:

Individual PMSE for model j, where H is the amount of data points to be forecast. Furthermore, to normalise the PMSE the covariance matrix estimated with the $VAR(p_{\hat{aic}})$ model is used, this is denoted as $\hat{\Sigma_{aic}}$. Moreover, y_i denotes the i^{th} observation and $y_{\hat{Z+i}}$ the dynamic forecasted value of the $(Z+i)^{th}$ observation given all data before and including the Z^{th} observation. Z denotes the last observation in the training set. The training and test division of the actual data set is specified in Section 3.1.

$$PMSE_j = \sum_{i=1}^{H} (y_{j,Z+i} - y_{j,\hat{Z+i}})^2, \forall j = 1,2,3$$

$$PMSE_{All} = \sum_{i=1}^{H} (y_{Z+i} - y_{\hat{Z+i}})' \hat{\Sigma_{aic}}^{-1} (y_{Z+i} - y_{\hat{Z+i}})$$

Furthermore, for the simulated data the following performance measures are tracked, the PMSE for the combined models alongside: proportion correct of substructure (PROP), proportion of parameters equal to 0 (Propzero), amount of wrongly excluded variables (ZeroW) and lastly, the bias in the large and small parameters (BiasL and BiasS respectively). Below the formulas can be found for the PMSE and the bias for small and large parameters. For the PMSE with the simulated data the first n_1 data points (90%) are used for training and the last n_2 (10%) for testing. The static forecasted series with the estimates parameters using the first n_1 data points is denoted as $\hat{y}_i$. Furthermore, y_i^* is static forecast value given the DGP. The Σ of the DGP is used to standardize the models in contrast to an estimated Σ, in order to get the most accurate normalisation possible. This will cause the $PMSE_{ALL}$ to most accurately represent the error of the models combined. Furthermore, it also gives us the property below.

$$PMSE_{All} = 1 + \frac{1}{n_2} \sum_{i=n_1}^{n_1+n_2} (y_i^* - \hat{y}_i)' \Sigma^{-1} (y_i^* - \hat{y}_i)$$

The reason why 1+the sum is considered described above, is the fact that the forecasted series using the estimated parameters are compared to the forecasted series knowing the DGP, not the observed values. The above equation is formed by splitting the error in two parts, the error of the DGP forecast with the observed and the error of the DGP forecast with the forecast using the estimated parameters. The expected value for the normalised squared difference between the DGP forecast and the observed values is 1 (because normalisation is done with the DGP Σ). Therefore this is taken in consideration above when calculating this.

Let ψ and φ be two sets containing the parameters in the DGP which are considered small and large respectively (See Appendix (Section 6.1) for selected parameters). These two sets combined do not span the entire parameter set. They were chosen by hand, looking at the parameters when ordered from small to large the parameters are chosen for the ψ set by including all the small parameters until the next one constituted a relatively larger difference. The same process was executed for choosing the variables for the φ set. Furthermore B and $\hat{B}$ represent the DGP parameters and estimated parameter vectors respectively.

$$BiasS = \sum_{i \in \psi} (B_i - \hat{B}_i)^2$$

$$BiasL = \sum_{i \in \varphi} (B_i - \hat{B}_i)^2$$

These values are used to compare the bias relatively between methods.

2.8 Cross-validation hyperparameters

The optimal hyperparameters for the penalisation methods (Lasso and Adaptive Lasso (-n)) are unknown. As suggested and implemented by Hsu et al. (2008), cross-validation is used in this paper to optimise this parameter. The cross validation consisted of a search over 20 different hyperparameters. Data points 1 till T_1 from Figure 1 is used to estimate the model for every hyperparameter considered, in this paper this constitutes as 72% of the data (76% for the historical data). Each model estimated does a static forecast on T_1 till T_2, which constitutes for 18% of the data (19% for the historical data). The optimal hyperparameter is chosen such that it minimises the $PMSE_{ALL}$ over this time period. Lastly, the model is re-estimated with the optimal hyperparameter using the data from 1 till T_2. The remaining data is then used for performance/forecast evaluation.

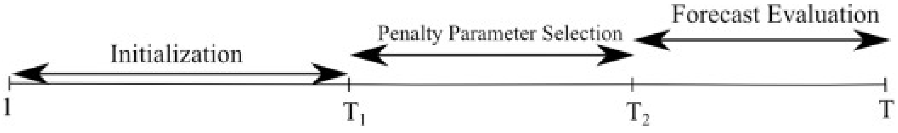

Figure 1: An illustration by Nicholson et al. (2017) of process for selecting the optimal hyperparameter

3 Data

In this section the process of data collection and simulation is discussed.

3.1 Application

Based on the paper of Hsu et al. (2008), the methods described in Section 2 are applied to model the relationship between the following three variables: unemployment rate (UER), annual percentage change of M1 and nominal GDP from the U.S (aggregation method: end of period). This is a good exercise to test the performance of the methods and gives a chance to compare the results to those from the paper of Hsu et al. (2008). Furthermore the data used is quarterly data from 1st of January 1960 till 1st of January 2006, amounting to a total sample size of 185. From this, the first 176 data points are used to train the model and the last 9 to test relative performance of the methods. Moreover in order to remove non-stationarity the first difference of all 3 variables is taken. Table 1 shows that the mean of the first differenced variables is around 0. Furthermore, the varying sizes of the standard deviations and minimum and maximum values illustrate that the variables differ in their variance.

Table 1: Descriptive statistics U.S. Macro-Economic Data

	Obs.	Mean	St.Dev.	Min. value	Max. value
GDP	184	0.0015	1.2250	-5.0850	3.4380
M1	184	0.0045	1.6614	-6.7543	7.4986
UER	184	-0.0038	0.3469	-0.9000	1.4000

UER measured in percentage;

Nominal GDP and M1 measured in annual percentage difference

The seasonally adjusted data is collected using the database from the Federal Reserve Bank of St. Louis (2020). Following the execution of these methods by Hsu et al. (2008), the maximum lag order considered for the methods (henceforth referred to as p_{max}) will be equal to 10.

Furthermore, Figure 2 and Figure 3 display the differenced and the non-differenced macro-economic data.

3.2 Simulation

The simulation study consists of 4 different models from which 50 data samples of sample size 150, 300 and 600 are generated. Different models are investigated to test the performance of the different methods under varying conditions. Furthermore, multiple sample sizes are considered to see the effect of increasing sample size on the accuracy of the different methods. Due to some methods possessing the asymptotic oracle properties, it is interesting to see whether this can manifest itself in these increasing sample sizes. The four models consist of two templates, where for both templates two different parameter generating ranges are

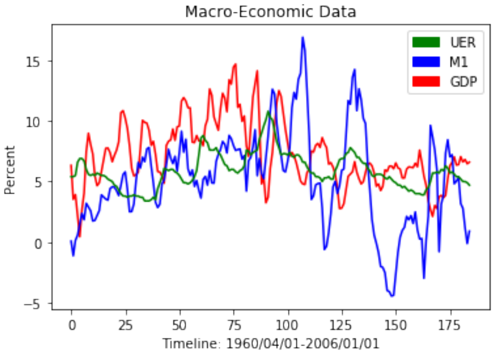

Figure 2: Data from Federal Reserve Bank of St. Louis (2020)

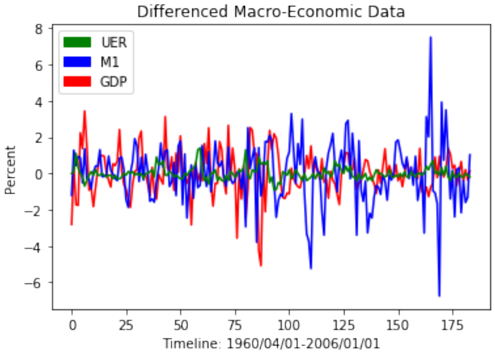

Figure 3: Data from Federal Reserve Bank of St. Louis (2020)

used. The two templates can be seen below and are referred to as Model 1 and 2. The instances of the two models that are generated can be seen in the Appendix (Section 6.1). B represents the lag operator

Model 1:

$$(I - A_1B - A_2B^3 - A_3B^5)y_t = u_t, u_t \sim N(0, \Sigma_{u1})$$

Model 2:

$$(I - A_4B - A_5B^2 - A_6B^3)y_t = u_t, u_t \sim N(0, \Sigma_{u2})$$

Where $A_1 = \begin{bmatrix} a_{11} & 0 \\ a_{12} & a_{13} \end{bmatrix}$, $A_2 = \begin{bmatrix} a_{21} & 0 \\ 0 & a_{22} \end{bmatrix}$, $A_3 = \begin{bmatrix} a_{31} & a_{32} \\ a_{33} & 0 \end{bmatrix}$, $\Sigma_{u1} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ and

$$A_4 = \begin{bmatrix} a_{41} & 0 & 0 & 0 \\ a_{42} & 0 & 0 & a_{43} \\ a_{44} & 0 & a_{45} & 0 \\ 0 & a_{46} & 0 & a_{47} \end{bmatrix}, A_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a_{51} & 0 & 0 \\ 0 & a_{52} & a_{53} & 0 \\ 0 & a_{54} & 0 & 0 \end{bmatrix}, A_6 = \begin{bmatrix} a_{61} & 0 & 0 & 0 \\ 0 & a_{62} & 0 & 0 \\ 0 & 0 & a_{63} & 0 \\ 0 & 0 & 0 & a_{63} \end{bmatrix},$$

$$\Sigma_{u2} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix}$$

The first model considers two time series, because of its high lag order the model does become high dimensional. Furthermore, the model contains seasonality by not including the third and fifth lag in the DGP. Because of this seasonality, the model is sparse with respect to the variables considered. The p_{max} considered is 5 for this model. The total amount of parameters in the first 5 lags, including the constants, sum to 22. The number of non-zero parameters to be estimated from those are 8. This causes the model to have a sparsity of 64%. The second model, on the other hand, has higher dimensionality and lower order, while not having seasonality. The sparsity of this model is due to the sparsity in the parameter matrices. The p_{max} considered is 3 for this model. The total amount of variables in the first three lags, including the constants, sum up to 52. The number of variables to be estimated from those are 15. This causes the model to have a sparsity fraction of 71%. The main differences between the two models are therefore the origins of their sparsity and their dimensionality.

The entries in the parameter matrices were randomly generated twice for each model. Once they are generated according to $U \sim (0, 0.8)$ and once according to $U \sim (0, 0.4)$. In order to guarantee the stability of the models, parameters are generated according to these distributions until parameter matrices are generated which correspond to a stable process. The two differently generated parameters illustrate how the methods handle different absolute size of parameters. Furthermore, the covariance matrix was chosen to allow for correlation between the error terms. Data was simulated using the mlVAR package in Rstudio (Details Section 6.3).

4 Results

In this section, the results of the methods on both data sets are discussed. First are the results from the simulation, where Model 1 and 2 are discussed (in that order), and second are the results from the historical data set.

4.1 Model 1

Firstly, Model 1 with the parameter set generated with the larger parameters (set 1 henceforth) is discussed, results reported in Table 2. According to the PMSE, the methods rank from best to worst as Adaptive Lasso-n, Adaptive Lasso and Lasso respectively, where the Adaptive methods are closest in their performance. The order is identical in the bias of the large parameters and explains for a large portion the ranking of the PMSE. These parameters have the largest effect on the forecasting performance and thus being able to estimate these with low bias results into lower PMSE. The Adaptive methods however, both trade low bias in the large parameters for higher bias in the small parameters and thereby eliminating more often the wrong variables (relative to Lasso). Therefore, there seems to exist a trade-off between low bias in the large parameters resulting into higher predictive performance and low bias in the small parameters resulting in more correct substructure selection. While this trade-off occurs in both Adaptive methods, the Adaptive Lasso-n has lower bias in the smaller parameters relative to the Adaptive Lasso and therefore, is also less likely to wrongly eliminate a variable. However, the bias in the smaller parameters still remains larger than with the Lasso.

Furthermore, the Adaptive methods attain the highest sparsity, while the Adaptive Lasso-n trades some of this sparsity for the benefits above. There is no clear conclusion from the PROP of the methods, they are fluctuating for the different sample sizes and have no conclusive ranking. Lastly, the performance of both the Adaptive methods increase the most with sample size for the PMSE. This follows from their oracle properties. This is also expected to happen for the PROP, but it does not. Therefore, it could be that the oracle properties for choosing the correct substructure occurs with higher sample sizes.

Moving on to the parameter set generated with the smaller parameters (set 2 henceforth), the results are different (See Table 2). Although the relative performance of the Adaptive Lasso-n and Adaptive Lasso has remained generally the same. The large difference is that the Lasso outperforms both Adaptive methods for sample sizes 150 and 300 in all performance measures. The oracle properties of the Adaptive methods take effect when the sample size increases to 600 and therefore outperform the Lasso here. This difference can be explained by the inaccurate estimate of the weights when the parameters are closer to 0. The OLS estimates due to their high variance in the high dimensional setting struggle to represent the relative sizes of the parameters. This results into inaccurate weights, certainly in lower sample sizes, and negatively impact the performance of both Adaptive methods. The Adaptive Lasso-n, which also uses partial correlation partly rectifies this problem. While performing better than the Adaptive Lasso because of it, it is not enough to surpass the Lasso. Lastly, when looking at correct substructure selection, the Lasso outperforms both Adaptive methods for all sample sizes, between the Adaptive methods they are generally equal.

Table 2: Simulation Results Model 1

Model 1		Param $\sim U(0,0.8)$						Param $\sim U(0,0.4)$					
Sample Size	Method	PMSE	PROP	PropZero	ZeroW	BiasS	BiasL	PMSE	PROP	PropZero	ZeroW	BiasS	BiasL
150	Lasso	1.1716	0.5973	0.3182	0.9200*	0.0256*	0.0284	1.1129*	0.5900*	0.3464	1.3200*	0.0080*	0.0234*
	Adapt. Lasso	1.1535*	0.6045*	0.3755*	1.4800	0.0320	0.0193*	1.1473	0.5627	0.3645*	1.3400	0.0115	0.0270
	Adapt. Lasso-n	1.1574	0.5964	0.3564	1.3600	0.0322	0.0199	1.1424	0.5709	0.3545	1.3300	0.0108	0.0264
300	Lasso	1.0708	0.6236	0.3527	1.0200*	0.0097*	0.0166	1.0709*	0.6812*	0.4336	1.3000	0.0045*	0.0142*
	Adapt. Lasso	1.0693	0.6264	0.3972*	1.4800	0.0145	0.0122	1.0969	0.6364	0.4839*	1.2800	0.0055	0.0181
	Adapt. Lasso-n	1.0645*	0.6400*	0.3891	1.2400	0.0136	0.0118*	1.0941	0.6236	0.4752	1.2000*	0.0053	0.0168
600	Lasso	1.0381	0.6163	0.3327	0.8800*	0.0060*	0.0069	1.0312	0.6509*	0.3864*	0.7400	0.0028*	0.0039
	Adapt. Lasso	1.0378	0.6318*	0.4118*	1.0200	0.0081	0.0066	1.0307*	0.6409	0.3745	0.6800	0.0033	0.0030*
	Adapt. Lasso-n	1.0354*	0.6036	0.4016	1.0200	0.0076	0.0060*	1.0309	0.6318	0.3514	0.6600*	0.0031	0.0033

* shows best performance of performance measure for respective data set. PMSE is normalized with DGP Covariance Matrix, PROP and PropZero is fraction correct and zero respectively.

4.2 Model 2

Similar to the result analysis of Model 1, first the parameter set generated with the larger parameters (set 3 henceforth) is analyzed (See Table 3). The results are similar to set 2. However now also when the sample size is 600, the Lasso outperforms the Adaptive methods. The possible reason for this is again inaccurate weight estimation. As stated in Section 3, Model 2 has constituted a severe increase in the dimensionality of the model. OLS estimation under high dimensionality (certainly in combination with low sample size) suffers from high variance in its estimates. The resulting weight estimation is therefore inaccurate and thus causes the Adaptive methods to become inaccurate. The Adaptive Lasso-n succeeds also in set 3 to lower the bias of the smaller parameters relative to the Adaptive Lasso and thus achieve better substructure selection. The effects of higher dimensions on the performance of the Adaptive methods is certainly apparent in the low sample sizes. Here the Adaptive methods struggle to attain their sparsity. The effects of high dimensionality on the performance of the three methods are clearly reflected in the heat maps shown below. They show the parameter matrices estimated (with constants) for all three methods for set 1 and set 3 for sample size 150 (as this sample size is closest to the application data as well). The difference in sparsity for the Adaptive methods for the set 1 and set 3, certainly relative to the base case of the Lasso.

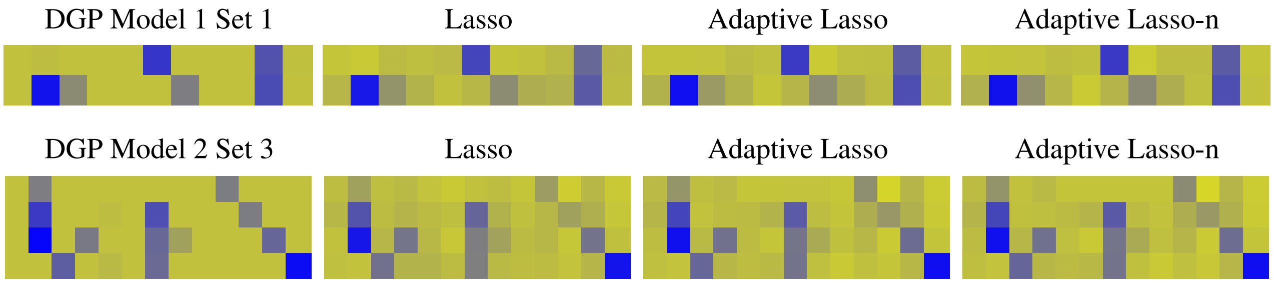

Figure 4: Heat Maps Model 1 (set 1) and Model 2 (set 3) Sample Size: 150

Moving on to the parameter set generated with the smaller parameters (set 4 henceforth) (See Table 3), there is a combination of the effects discussed earlier. Not only is set 4 in a high dimensional setting, but the parameter absolute differences are also smaller. The combined effect has a severe negative impact on the accuracy of the weights in the Adaptive methods and therefore they also perform poorly relative to the Lasso. The Adaptive Lasso-n outperforms the Adaptive Lasso in all sample sizes in most performance measures. Furthermore, the sparsity of the Adaptive methods is remarkably low, however this is explained above using inaccurate weight estimation as an argument. This is reflected in the low PROP and is due to worse substructure selection. The low number of wrongly excluded variables by the Adaptive methods is also due to their low sparsity in this data set. Lastly, the bias in the large estimates in the Adaptive methods remains lower than that of the Lasso but do not result in lower PMSE.

4.3 Application

This section discusses the results of the methods from Hsu et al. (2008) and the added (hybrid) Adaptive Methods when applied on the macro-economic data set, seen in Table 4 and 5. The AIC+Lasso-f and AIC+Lasso-s methods perform best, with AIC+BE following afterwards. The Adaptive methods are among

Table 3: Simulation Results Model 2

Model 2		Param $\sim U(0,0.8)$						Param $\sim U(0,0.4)$					
Sample Size	Method	PMSE	PROP	PropZero	ZeroW	BiasS	BiasL	PMSE	PROP	PropZero	ZeroW	BiasS	BiasL
150	Lasso	1.3595*	0.6031*	0.3946*	1.3800	0.0292*	0.0365	1.2931*	0.5931*	0.4062*	2.6400	0.0115*	0.0439
	Adapt. Lasso	1.4241	0.4730	0.2442	1.5400	0.0307	0.0237	1.3520	0.4731	0.2600	1.9600	0.0219	0.0412
	Adapt. Lasso-n	1.4219	0.4731	0.2308	1.2000*	0.0336	0.0235*	1.3479	0.4617	0.2512	1.9500*	0.0214	0.0393*
300	Lasso	1.1900*	0.6269*	0.3823*	1.1400	0.0167*	0.0168*	1.1387*	0.6327*	0.4269*	2.1600	0.0082*	0.0210
	Adapt. Lasso	1.1941	0.5150	0.3511	1.2200	0.0188	0.0195	1.1637	0.4800	0.2435	1.3600	0.0150	0.0162
	Adapt. Lasso-n	1.1904	0.5335	0.3631	1.1000*	0.0168	0.0169	1.1626	0.4827	0.2458	1.3400*	0.0147	0.0158*
600	Lasso	1.0816*	0.6431*	0.3846*	0.8000	0.0071*	0.0062*	1.0713*	0.6500*	0.4423*	2.0600	0.0064*	0.0096
	Adapt. Lasso	1.0928	0.5265	0.3634	0.9000	0.0079	0.0092	1.0759	0.4862	0.2438	1.2000	0.0088	0.0068
	Adapt. Lasso-n	1.0902	0.5347	0.3611	0.6000*	0.0076	0.0070	1.0756	0.4912	0.2458	1.1200*	0.0085	0.0066*

* shows best performance of performance measure for respective data set. PMSE is normalized with DGP Covariance Matrix, PROP and PropZero is fraction correct and zero respectively.

the worst performing methods of all, including the most basic such as AIC, which selects only the order. However, the poor results of the Adaptive methods when applied on this data set concurs with the results of the simulation. The simulation results show an increase in dimensionality with low sample size causes poor performance in the Adaptive methods (from Model 1 to 2). The application has the same conditions as Model 2 with sample size 150, referring to the similar sample size and high dimensionality. The poor performance because of the high dimensionality in the result section is explained by inaccurate weight estimation. The poor performance of the AIC method is an indicator for the inaccurate weight estimation of the Adaptive methods. This method is an OLS regression with lag order 8, its inverted parameters are equal to the weights for AIC+Adaptive Lasso and used in calculating those for the AIC+Adaptive Lasso-n. The AIC method performs the worst of all 12 methods. Therefore, when the weights are constructed from these parameters, they themselves also do not give much information. In this case, they even sabotage the effectiveness of these Adaptive Lasso methods. This is also reflected in the worse performance of the non-hybrid Adaptive methods, which have a higher p_{max} (10) and therefore higher dimensionality. This amplifies the effects of high dimensionality and therefore causes the performance of these methods to deteriorate.

Furthermore, the low sparsity of the Adaptive methods are also reflected in the simulation results. When looking at the results of set 3 and 4 with sample size 150 (very similar to macro-economic data set, which has 185), the low sample size combined with high dimensionality caused problems with the sparsity of the methods. This gets resolved by a higher sample size or when looking at set 1, with lower dimensionality. The inaccurate weight estimation for high dimensionality partly exists due to limitations of this paper's implementation of the Adaptive methods. These are discussed further in the Conclusion (Section 5).

4.3.1 Comparison with results Hsu et al. (2008)

As noted throughout the paper, the methods proposed in Table 4 and the macro-economic data on which the methods are applied form a replication of the paper from Hsu et al. (2008). However the results from this

Table 4: Results Macro-Economic Data, methods from Hsu et al. (2008)

Perform. Crit.	AIC	BIC	AIC+BE	BE+FS	Lasso-s	Lasso-f	AIC+Lasso-s	AIC+Lasso-f
Proportion zero	-	-	0.5067	0.6267	0.3441	0.6021	0.3880	0.6167
Lag Order	8	4	8	9	10	10	8	8
$PMSE_1$	0.5181	0.2300	0.2208	0.4965	0.2695	0.2679	0.2246	0.1771
Rank	11	4	2	9	6	5	3	1
$PMSE_2$	0.0447	0.0380	0.02647	0.0431	0.0573	0.0247	0.0285	0.0246
Rank	11	8	3	10	12	2	5	1
$PMSE_3$	3.1514	2.1107	2.1290	1.6648	1.5278	2.4656	1.1562	1.8921
Rank	10	5	6	3	2	9	1	4
$PMSE_{all}$	3.7095	2.4639	2.1700	2.8938	2.6374	2.5172	1.6892	1.9794
Rank	12	4	3	9	6	5	1	2

$PMSE_{all}$ is normalized with $VAR(\hat{p_{aic}})$ Sigma, Proportion of zero is fraction of parameters equal to 0.

$PMSE_1$, $PMSE_2$ and $PMSE_3$ correspond respectively to GDP, Unemployment rate and M1. Rank combined with Table 5.

Table 5: Results Macro-Economic Data, (Hybrid) Adaptive methods

Perform. Crit.	Adaptive Lasso	Adaptive Lasso-n	AIC+Adaptive Lasso	AIC+Adaptive Lasso-n
Proportion zero	0.1075	0.0967	0.1200	0.0933
Lag Order	10	10	8	8
$PMSE_1$	0.3657	0.4752	0.5148	0.5389
Rank	7	8	10	12
$PMSE_2$	0.0394	0.0359	0.0281	0.0306
Rank	9	7	4	6
$PMSE_3$	3.5819	3.4135	2.1571	2.1318
Rank	12	11	8	7
$PMSE_{all}$	3.6799	3.5757	2.7839	2.7755
Rank	11	10	8	7

$PMSE_{all}$ is normalized with $VAR(\hat{p_{aic}})$ Sigma, Proportion of zero is fraction of parameters equal to 0.

$PMSE_1$, $PMSE_2$ and $PMSE_3$ correspond respectively to GDP, Unemployment rate and M1. Rank combined with Table 4.

paper differs from theirs. This can be because of multiple reasons, which are discussed in this subsection. The first possible reason is the use of seasonally adjusted data provided to us by the Federal Reserve Bank of St. Louis. It is not entirely clear from the paper by Hsu et al. (2008), whether they used this as well but it could play a minor role. Furthermore, for the methods not involving the Lasso, this paper uses standard OLS regression to find the parameters. Whether this is to compare models for elimination or selection in BE or FS or choosing the lag order in AIC and BIC. Hsu et al. (2008) most likely used weighted OLS. This could affect the different number of elimination and selection by the BE and FS and the accuracy of estimation of the other methods. Furthermore, specifically for the FS, Hsu et al. (2008) uses a version where they add the parameters as univariate series. On the other hand, in this paper the parameters are added lag wise, starting per model with the first lags and considering the final lags last. This is further explained in Section 2.4. This affects in which order the different variables are considered and as noted before affect the final result severely. Moreover, to optimise the cost function of the Lasso, this paper employs a gradient descent method. Hsu et al. (2008), on the other hand, use the more efficient algorithm: least-angle regression (LARS) to optimise. As the gradient descent method this paper employs is inferior to that of the LARS, the parameters estimated might not be the optimal ones (or atleast different to the ones found with LARS).

Lastly, the cross-validation process for choosing the lambda for the penalisation methods (Lasso, Adaptive Lasso, Adaptive Lasso-n) were not fully explained by Hsu et al. (2008). This paper follows the methodology described in section 2.8 for finding the optimal hyperparameter. To what extent this is similar to that from Hsu et al. (2008) is not known. However, it is likely they are different, which plays a role in the final outcome of the penalisation.

5 Conclusion

In the results of the simulation, the oracle properties for both the Adaptive methods cause accurate predictive performance (low PMSE) in high sample size conditions. However, the substructure selection accuracy did not increase consistently with sample size, which is expected for the methods, certainly the Adaptive methods, which have the oracle properties. Furthermore, in presence of low dimensionality and parameters that are generated with potentially large parameters (set 1), the Adaptive methods with OLS estimates outperform the Lasso in substructure selection (PROP), sparsity (PropZero) and predictive accuracy (PMSE). Both the Adaptive methods did have higher bias in the smaller parameters relative to the Lasso, which resulted in more often wrongfully excluding variables (ZeroW). However, this effectiveness was reduced severely when either the parameters were generated to be lower (set 1 to set 2), the dimensionality increased (set 1 to set 3) or when both these effects appear (set 1 to set 4). The results of the methods replicated from Hsu et al. (2008) and applied on the same macro-economic data set, mostly concur with that of the paper. Furthermore, the poor results of the added Adaptive methods in this data set concur with the simulations. Namely the model's high dimension and relatively low sample size causes a similar negative effect on the performance of the Adaptive methods as from set 1 to set 3 of the simulation.

Between the Adaptive methods it is observed that the Adaptive Lasso-n is a competitive method with the Adaptive Lasso. This manifests in the PMSE for both the simulated and the real data. Furthermore, also in choosing the correct substructure they are competitive. The main difference is that the Adaptive Lasso-n is able to reduce the bias in the smaller parameters and thus lowers the amount of wrongly excluded variables. However, it sacrifices some of the sparsity features from the Adaptive Lasso-n for this. Because of this the substructure selection remains equal. Furthermore, it generally keeps the characteristic low bias in the large parameters from the Adaptive Lasso. This causes it to have its competitive PMSE. The relatively lower bias in the smaller parameters is due to the ability of the Adaptive Lasso-n to distinguish between the smaller parameters in the DGP and the zero parameters more accurately, using the information from the partial correlation. This resulted in consistenly reducing the number of wrongful exclusions of variables (ZeroW) by the Adaptive Lasso.

During the research there were limitations, which should be taken in consideration when interpreting the results. The first limitation is the execution of the penalisation methods (Lasso, Adaptive Lasso and Adaptive Lasso-n). For these three methods this paper uses a gradient descent method in order to optimise the objective function. Certainly in a high dimensional setting this causes trouble finding the global optimum. Furthermore, it resulted in a high computation time, which limited the search for the optimal hyperparameters to consider only 20 options. However, as Hsu et al. (2008) suggest, using the LARS algorithm could fix both of these issues, Efron et al. (2004) show that the order of the computational time would be equal to that of the OLS. Another issue with the implementation of the Adaptive methods specifically was the use of OLS for the weight estimation. As suggested by Zou (2006), in presence of multi-colinearity and high dimensions the weights should be estimated with Ridge. Moreover, the simulation used 50 generated data sets per generated parameter set. This number is however quite low and should be increased to further the reliability of the research. Concluding, for the estimation of the simulation models, the p_{max} should be consider higher than that of the DGP, instead of equal. This resembles reality more closely.

5.1 Further Research

For further research, the partial correlation in step 1 of the Adaptive Lasso could be calculated in different ways. Consider the example of the largest parameters being the last lags considered and the smaller ones the first. This would cause the partial correlation to have the same structure as the parameter estimates, namely small parameters score low and vice versa. Therefore, a different approach could calculate partial correlation in order of large to small parameters instead of following the order of the lags. This could be done by using for example the OLS or Ridge estimates (to pick the order). Lastly, there have been other variable selection methods with similar ideas to the ones presented in this paper, such as the the SCAD (Fan and Li, 2001) and the standard error adjusted Adaptive Lasso (Qian and Yang, 2013). It would be interesting to see the relative performance of these methods to the methods of this paper under similar analysis.

References

Bühlmann, P., Kalisch, M., and Maathuis, M. H. (2010). Variable selection in high-dimensional linear models: partially faithful distributions and the pc-simple algorithm. *Biometrika*, 97(2):261–278.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J., Peng, H., et al. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961.

Federal Reserve Bank of St. Louis (2020). Gdp, unrate, m1. data retrieved from FRED, https://fred.stlouisfed.org/.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hsu, N.-J., Hung, H.-L., and Chang, Y.-M. (2008). Subset selection for vector autoregressive processes using lasso. *Computational Statistics & Data Analysis*, 52(7):3645–3657.

Nicholson, W. B., Matteson, D. S., and Bien, J. (2017). Varx-l: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 33(3):627–651.

Ogutu, J. O., Schulz-Streeck, T., and Piepho, H.-P. (2012). Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In *BMC Proceedings*, volume 6, page S10. Springer.

Qian, W. and Yang, Y. (2013). Model selection via standard error adjusted adaptive lasso. *Annals of the Institute of Statistical Mathematics*, 65(2):295–318.

Sauerbrei, W., Perperoglou, A., Schmid, M., Abrahamowicz, M., Becher, H., Binder, H., Dunkler, D., Harrell Jr, F. E., Royston, P., and Heinze, G. (2019). State-of-the-art in selection of variables and functional forms in multivariable analysis–outstanding issues. *ArXiv Preprint ArXiv:1907.00786*.

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

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 67(2):301–320.

Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37(4):1733.

6 Appendix

6.1 Generated Parameters

The red colored elements of the matrices are the ones that are selected to be in the set containing the smaller elements. This corresponds to set denoted by ψ.

The green colored elements of the matrices are the ones that are selected to be in the set containing the larger elements. This corresponds to set denoted by φ.

Set 1 (Model 1: Param $\sim U(0,0.8)$)

$$A_1=\begin{bmatrix} \colorbox{pink}{0.007} & 0 \\ \colorbox{lightgreen}{0.687} & \colorbox{pink}{0.125} \end{bmatrix}, A_2=\begin{bmatrix} \colorbox{lightgreen}{0.485} & 0 \\ 0 & 0.1726 \end{bmatrix}, A_3=\begin{bmatrix} 0.341 & \colorbox{pink}{0.002} \\ 0.375 & 0 \end{bmatrix}$$

Set 2 (Model 1: Param $\sim U(0,0.4)$)

$$A_1=\begin{bmatrix} 0.170 & 0 \\ 0.200 & \colorbox{pink}{0.039} \end{bmatrix}, A_2=\begin{bmatrix} \colorbox{lightgreen}{0.266} & 0 \\ 0 & \colorbox{pink}{0.034} \end{bmatrix}, A_3=\begin{bmatrix} \colorbox{lightgreen}{0.272} & 0.198 \\ 0.125 & 0 \end{bmatrix}$$

Set 3 (Model 2: Param $\sim U(0,0.8)$)

$$A_4=\begin{bmatrix} \colorbox{pink}{0.170} & 0 & 0 & 0 \\ 0.455 & 0 & 0 & \colorbox{pink}{0.006} \\ \colorbox{lightgreen}{0.768} & 0 & 0.185 & 0 \\ 0 & 0.295 & 0 & \colorbox{pink}{0.015} \end{bmatrix}, A_5=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.362 & 0 & 0 \\ 0 & 0.247 & \colorbox{pink}{0.066} & 0 \\ 0 & 0.242 & 0 & 0 \end{bmatrix}, A_6=\begin{bmatrix} 0.174 & 0 & 0 & 0 \\ 0 & 0.175 & 0 & 0 \\ 0 & 0 & 0.255 & 0 \\ 0 & 0 & 0 & \colorbox{lightgreen}{0.730} \end{bmatrix}$$

Set 4 (Model 2: Param $\sim U(0,0.4)$)

$$A_4=\begin{bmatrix} 0.293 & 0 & 0 & 0 \\ 0.335 & 0 & 0 & 0.169 \\ \colorbox{pink}{0.064} & 0 & \colorbox{pink}{0.010} & 0 \\ 0 & \colorbox{lightgreen}{0.363} & 0 & \colorbox{lightgreen}{0.350} \end{bmatrix}, A_5=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.233 & 0 & 0 \\ 0 & 0.120 & 0.214 & 0 \\ 0 & 0.210 & 0 & 0 \end{bmatrix}, A_6=\begin{bmatrix} 0.283 & 0 & 0 & 0 \\ 0 & \colorbox{pink}{0.030} & 0 & 0 \\ 0 & 0 & \colorbox{lightgreen}{0.382} & 0 \\ 0 & 0 & 0 & \colorbox{pink}{0.005} \end{bmatrix}$$

6.2 Python Code Details

The Python Code was written in Jupyter Notebook and for the exact code we refer to the attached ZipFile. The version of Python used was 3.7. Furthermore the optimisation package used was SciPy v.1.5.0. From this specific package we used the L-BFGS-B gradient descent method for the penalisation methods. The Python code was used for all methods and optimisation. The simulated data was made using R, for which the details are in the following subsection.

6.3 R Code Details

The R code was written in RStudio and for the exact code we refer to the attached ZipFile. R was used for simulating the Data. For this purpose we used the mlVAR package, which had a function prewritten for it.