

FORECASTING MACROECONOMIC VARIABLES USING SEVERAL PREDICTORS



Lærke Hæstrup Camilla Lund Ypkendanz Group G2-104B

AALBORG UNIVERSITY – FACULTY OF ENGINEERING AND SCIENCE STUDY BOARD OF MATHEMATICS, PHYSICS AND NANOTECHNOLOGY MASTER'S THESIS - CAND.SCIENT.OECON. IN FINANCIAL ENGINEERING

Title page: Forecasting Macroeconomic Variables Using Several Predictors

Semester: 4th semester, master's degree

Project Period: Spring 2016

Educational Institution: Aalborg University

Study program: Cand.scient.oecon. in Financial Engineering

Thesis Group: Lærke Hæstrup Camilla Lund Ypkendanz

Supervisor: Matt Dziubinski

Master's Thesis: 92 pages Appendix: 2

Completed the 10.06.16



AALBORG UNIVERSITET STUDENTERRAPPORT

PREFACE

This thesis is written by group G2-104b, Lærke Hæstrup and Camilla Lund Ypkendanz, during the 10th semester in Mathematics-Economics, specializing in Financial Engineering, at the faculty of Engineering and Science, Aalborg University. The thesis was computed in accordance with the curriculum of September 2012, from the Study Board of Mathematics, Physics and Nanotechnology.

The thesis is written in the period from the 1st of February to the 10th of June 2016.

Aalborg, June 10, 2016

Lorh He

Lærke Hæstrup

Jath June

Camilla Lund Ypkendanz

Cover photo: http://www.fm.dk/arbejdsomraader/oekonomisk-politik

RESUMÉ

Forecasting af makroøkonomiske variabler er af stor interesse, da de beskriver den forventede økonomiske udvikling, f.eks. handler aktører på det finansielle marked på baggrund af forventningerne til de fremtidige aktiekurser. Hvis aktøren kan forudsige de makroøkonomiske variable relativt præcist, har han en klar fordel i forhold til hans konkurrenter, da han er i stand til at handle når det er attraktivt.

Den empiriske analyse i dette speciale omhandler modellering med mange prædiktorer samt forcasting af tre makroøkonomiske variabler: GDP, arbejdsløshed og inflation. Vi benytter fire metoder til at konstruere vores modeler, som videre benyttes til at forecaste de tre variable. De fire benyttede metoder er: Dynamiske factor model med principal komponent analyse, shrinkage metoder, VAR modeler og Bayesian model average.

For at kunne bestemme nøjagtigheden af de forskellige forecasts, estimeret på baggrund af de makroøkonomiske modeller, benytter vi Diebold-Mariano testen. Diebold-Mariano tester de makroøkonomiske forecasts mod et benchmark forecast, estimere på baggrund af en AR(p) model.

Det mest nøjagtige forecast af GDP stammer fra factor-augment VAR modellen, som også forecaster inflations raten bedst. Lasso-VAR modellen giver det bedste forecast for arbejdsløsheds raten, hvilket betyder at de mest præcise modellerfor de tre variable, alle tilhører en VAR model. Diebold-Mariano testen benyttes på de mest præcise VAR modeller, og det viser sig at de komplekse modeller ikke producerer et mere præcist forecast end den simple AR benchmark.

Ud over hvilke modeller der producere det bedste forecast, kigger vi også på hvilke prædiktorer de forskellige modeller vælger at medtage. Ingen af prædiktorene er universelle og inkluderet i alle modeller, men inden for den samme makroøkonomiske variable er der prædiktorer der er medtaget i alle modeller.

READING GUIDE

This report is a master's thesis in Mathematics-Economics, at Aalborg University, and we therefore assume that the reader is familiar with technical terms and methods used in mathematics and economics.

The Structure of the Thesis

The thesis is divided into 9 chapters with corresponding sections and subsections. These are numbered such that chapters are assigned a number, the sections are further assigned a number, and subsection further a number. A summary of the thesis' content is viewed here:

Chapter 1 Introduction.

Chapter 2 Macroeconomic Variables.

Chapter 3 Model Assessment and Selection.

Chapter 4 Dynamic Factor Models and Principal Component Regression.

Chapter 5 Shrinkage Methods.

Chapter 6 Vector Autoregressive Models.

Chapter 7 Bayesian Model Averaging.

Chapter 8 Empirical Analysis.

Chapter 9 Conclusion.

An illustration of the thesis' content are shown in Figure 0.1.





Notation

In this thesis matrices are denoted by a bold capital letter, X, and vectors are denoted by a bold small letter, y. When matrices and vectors are transposed, it is denoted by an apostrophe, x'.

References

Sources are, through the thesis, referenced by using IAT_EX 's internal source reference system $BibT_EX$. The source references refer to the alphabetical bibliography in the back of the thesis. When referring to a sours, we use the Harvard method, in which the printed and electronic source is referenced in the form [author's last name, year]. Does the source contain two or more main authors it is referenced in the form [name one. al, year].

Figures and tables are give a specific number and explanatory text. The first number corresponding to a figure or a table refer to its corresponding chapter, such that the first figure/table in Chapter x is identified as figure/table x.1, next as figure/table x.2 etc.

We wish you a Happy reading!

CONTENTS

Contents

1	Intr	roduction	1
2	Ma	croeconomic Variables	3
	2.1	Gross Domestic Product	3
	2.2	Unemployment	5
	2.3	Inflation	6
3	Mo	del Assessment and Selection	9
	3.1	Assessing Model Accuracy	9
		3.1.1 The Bias-Variance Trade-Off	11
	3.2	Cross-Validation	12
		3.2.1 The Validation Set Approach \ldots	12
		3.2.2 k-Fold Cross-Validation	12
	3.3	Model Selection	13
		3.3.1 The Akaike Information Criterion	14
		3.3.2 The Bayesian Information Criterion	14
4	Dyr	namic Factor Models and Principal Component Analysis	15
	4.1	Estimation Methods	16
		4.1.1 Estimation of Factors by Principal Component Analysis	17
	4.2	Selecting the Number of Factors	19
5	Shr	inkage Methods	21
	5.1	Ridge Regression	22
		5.1.1 Comparison with Principal Component Regression	26
	5.2	The Least Absolute Shrinkage and Selection Operator (lasso) $\ \ldots \ \ldots \ \ldots \ \ldots$	27
		5.2.1 The Coordinate Descent Algorithm and the Lasso	29
	5.3	Grouped Lasso	32
	5.4	Elastic Net	33

6	tor Autoregressive Models	35							
	6.1	Factor-Augmented VAR Model	36						
	6.2	Lasso-VAR Model	36						
7	Bay	vesian Model Averaging	39						
	7.1	Choosing the Priors in Bayesian Model Averaging	40						
	7.2	Posterior Computation	41						
8	$\mathbf{Em}_{\mathbf{j}}$	pirical Analysis	43						
	8.1	Dynamic Factor Models	45						
	8.2	Shrinkage Methods	49						
		8.2.1 Gross Domestic Product	49						
		8.2.2 Unemployment Rate	56						
		8.2.3 Inflation Rate	63						
		8.2.4 Conclusion	70						
	8.3	VAR Models	70						
		8.3.1 Gross Domestic Product	71						
		8.3.2 Unemployment Rate	75						
		8.3.3 Inflation Rate	79						
		8.3.4 Conclusion	82						
	8.4	Bayesian Model Average	83						
	8.5	Empirical Conclusion	88						
		8.5.1 Comparison of the Included Predictors	90						
9	Con	nclusion	91						
\mathbf{Li}	st of	Figures	93						
\mathbf{Li}	List of Tables 97								
Bi	ibliog	graphy	103						
Α	Dat	a	107						
в	R-C	Codes	111						
	B.1	Dynamic Factor Models	111						
	B.2	Shrinkage Methods	117						
	B.3	VAR models	143						

B.4	Bayesian Model Average	 	 	 •		•		•	•	•	 •		•	156
B.5	Empirical Conclusion	 	 					•	•		 •			162

CHAPTER 1

INTRODUCTION

Historically, macroeconomic methods used for forecasting have been focused on a small number of predictors. Both univariate autoregressions and vector autoregressions have been widely use in macroeconomic forecasting, and today they serves as standard benchmarks used to evaluate economic forecasts. However, in resent years the pool of available data has grown extensively, and today there are often hundreds of explanatory variables available, many of which are highly collinear. Much of the available data has a limited history, which sometimes causes the number of predictors to exceeds the number of available observations.

This massive growth in available data has led to the development of multiple data shrinkage techniques, where the most popular approach, for economic forecasting, is the dynamic factor models.

The dynamic factor model approach was first popularized by [Stock and Watson, 2002] who describes the use of factor models, where the factors are estimated by principal component analysis, when forecasting with many predictors. The main idea behind the factor models is to split each variable up, and divide them into a few common factors, which accounts for much of the total variation in the variables. When the factors have been estimated, they are used as the predictors in regression models for forecasting. Other important shrinkage techniques, such as ridge regression, [Hoerl and Kennard, 1970], lasso, [Tibshirani, 1996], grouped lasso, [Bakin, 1999], elastic net, [Zou and Hastie, 2005], and Bayesian model average, [Zellner, 1986], are available.

The aim of this thesis is to conduct an empirical analysis, on a wide range of econometric methods, designed to handle many predictors. We determine the most accurate model and test if this model forecast outperforms the forecast mate by the benchmark, AR(p) model. To compare the forecast accuracy of the benchmark model and the conducted models, we apply the Diebold-Mariano test [Diebold and Mariano., 1995].

We focus our empirical work around three U.S. target macroeconomic variables: Gross domestic product, unemployment, and inflation. These variables are corr focus points in the Federal Reserve's Dual Mandate, in which the goals is maximum employment, stable prices and moderate long-term interest rates.

CHAPTER 2

MACROECONOMIC VARIABLES

This Chapter is based on [Froyen, 2013] and [Blanchard et al., 2010].

Macroeconomics is the study of how the economy behaves as a whole. It is used by consumers, firms, and governments to analyze, which future decisions to make in order to deal with possibly adverse future economic events.

The main goals in every economy are high output growth, low unemployment, and low inflation. Hence, the success af an economy's overall performance is assessed by studying how high rates of output and consumption growth can be achieved. For conducting of such an assessment, three macroeconomic variables are particularly important, that is, the gross domestic product, unemployment, and inflation.

2.1 Gross Domestic Product

Gross domestic product (GDP) is one of the primary indicators used to gauge the health of a country's economy. It is a measure of aggregated output in the national income accounts, and it is counted by the value of all currently produced financial goods and services over a given period, typically a year.

Notice, that only final goods and services enter the definition of GDP. Intermediate goods and services do not enter because they contribute to the value of the final goods, thus adding intermediate goods directly in GDP results in double counting. However, two types of goods used in the production process, which is usually associated with intermediate goods, are included in GDP. First, currently produced capital goods¹ are included in GDP, as only a depreciation² of these goods contributes to the value of the final goods in each period. Hence, not including capital goods in GDP is equivalent to assuming full depreciation in the present period. Second, inventory investments³ are included, as inventory stock of final goods are currently produced output.

¹Capital goods are capital resources such as factories and machinery used to produce other goods.

²Depreciation is the portion of capital stock that wears out each year.

 $^{^{3}}$ Inventory investments are the net change in inventories of final goods awaiting sale, or of materials used in the production process.

Nominal GDP is evaluated at market prices, which makes it sensitive to changes in the average price level. This sensitivity is caused by the re-evaluation of the market price whenever a shock in either supply or demand leads to a new equilibrium. To overcome this problem, we compute real GDP which, in a traditional setting, is constructed as the sum of all produced final goods and services, multiplied by the constant prices from a base year. Since there is more than one final good, real GDP is defined as a weighted average of the output of all final goods, where the weights are the relative prices.

There are two problems concerning real GDP. The first arises if the base year is changed: A changed base year causes the weights and thus the GDP history to changes. The second and most important problem with real GDP is relative price changes, and substitutions among products contained in GDP. To avoid these problems, we can use a chain-weighted measure of real GDP, where the constant base year is replaced by the average prices in the present- and previous years. This means, that the base moves forward each year, eliminating the problem caused by relative price-induced substitutions.

GDP comprises four main components: Consumption, investment, government purchase, and net exports. Consumption, which is the largest GDP component, is the household sector's demand for current used output, and consists of consumer durable goods, nondurable consumption goods, and consumer services. The investment component covers the part of GDP which is purchased by the business sector, plus residential construction. That is business fixed investments, residential construction investments, and inventory investments. The government purchases of goods and services component covers the share of the current output bought by the government sector. The government sector covers the federal government as well as state and local governments. The net export component covers the total export minus import, that is, the currently produced goods and services sold to foreign buyers. To gain bigger insight into GDP and what its many components covers, see figure 2.1.



Figure 2.1: The macroeconomic variable GDP, comprises four main components, that is consumption, investment, government purchase, and net exports. Consumption and investment can be further divided into three subcomponents.

2.2 Unemployment

This section is further based on [BLS, 2016a].

Unemployment is an important variable in macroeconomics, because it tells something about whether the economy uses its resources, in the form of manpower, optimally. Furthermore, when the number of unemployed people rises, investors and consumers become more reticent in their investment and spending patterns, because they fear an economic recession. On the other hand, when the number of unemployed people declines, investors and consumers have more confidence in the economy, which is reflected in their investment and spending patterns.

Formally, unemployment is defined as the number of people who does not have a job, but are looking for one. Hence, to be categorized as unemployed a person must meet two criteria, first he or she must be unemployed, and second he or she must be job seeking. To compare the unemployment numbers through different years, we construct the unemployment rate, which is the number of unemployed persons expressed as a percentage of the labor force.

In the U.S., as in many European countries, a federal agency, here the Bureau of Labour Statistics, conducts a monthly survey in order to determine the unemployment rate. The Current Population Survey is conducted on 60,000 households and provides a comprehensive body of data on the labor force, employment, unemployment, persons not in the labor force, hours of work, earnings, as well as other demographic and labor force characteristics, all of which are used in the computation of the unemployment rate.

2.3 Inflation

This section is further based on [BLS, 2016b].

The last macroeconomic variable which we consider in this thesis is inflation. Inflation is the term we use to describe a rise in the general price level, and it is measured as the rate at which the price level increases. Inflation is an important macroeconomic variable because of its huge impact on an economy. Economies with high inflation rates normally experience higher uncertainty, as the variation in the relative prices makes it harder for firms to make decisions about their future costs. Furthermore, high inflation rates corresponds to high interest rates, thus it is necessary to compensate for the decline in purchasing power of future interest and principal repayments, resulting in higher costs and an overall drag on the economy.

To compute the inflation rate, we use the percentage rate of change of a price index over a given period, where the price index measures the aggregated price level relative to a base year. One of the challenges concerning the computation of the inflation rate is which price index to chose. In the U.S. the inflation rate is determined by the Bureau of Labor Statistics, which uses various indexes to measure different aspects of inflation:

• The Consumer Price Index is the weighted average of prices for a representative basket of consumer goods and services, released on a monthly basis, and containing two separate indexes in order to represent different groups or populations of consumers. The two indexes are the Consumer Price Index for All Urban Consumers, which is the most frequently used index, and the Consumer Price Index for Urban Wage Earners and Clerical Workers, which is often used for wage escalation agreements.

The Consumer Price Indexes are often used to escalate or adjust payments for rents, wages, alimony, child support, and other obligations that may be affected by changes in the cost of living.

• The Producer Price Index is a weighted index, that measures the average change over time in the selling prices received by domestic producers goods and services. It shows the trends within the wholesale markets, manufacturing industries, and commodity markets. Hence, all of the physical goods-producing industries that make up the U.S. economy are included. Opposite to the Consumer Price Index, which measures the price changes for the consumer's perspective, the Producer Price Index measures the price changes from the producer's perspective.

- Import and Export Prices Indexes measures the average price changes of non-military goods and services, imported to or exported from the U.S.. This index helps to measure the inflation rate in globally traded products.
- Employment Cost Trends measures the changes in labor costs over time, and the average costs per working hour. Unlike the previous three indexes, these trends are only published quarterly.

Although various price indexes can be used to determine the inflation rate, the most commonly used index is the consumer price index.

CHAPTER 3

MODEL ASSESSMENT AND SELECTION

This chapter is based on [Hastie et al., 2008] and [James et al., 2013]

The purpose of this chapter is to present some key tools used to select the optimal model to predict our data set. To find the optimal model, we consider two things:

- Model selection –Estimate the performance of different models in order to choose the best model.
- Model assessment -Having chosen a final model, estimate its prediction error on new data.

First, we introduce a measure used to evaluate the performance of different models.

3.1 Assessing Model Accuracy

To see how well a model performs, we measure the accuracy of the predictions relative to the observed data. This measure is essential in order to select the right model for a given data set, since, in general, no model is dominating for all data sets.

Consider the case of a quantitative response, and assume that y is a target variable, x is a vector of inputs, $\hat{f}(x)$ is a model estimated from a training set \mathcal{T} , and $L(y, \hat{f}(x))$ is the loss function for measuring the errors between y and $\hat{f}(x)$. Typical the loss function is expressed as one of the following error measures,

$$L(y, \hat{f}(\boldsymbol{x})) = \begin{cases} (y - \hat{f}(\boldsymbol{x}))^2, & \text{squared error} \\ |y - \hat{f}(\boldsymbol{x})|, & \text{absolute error.} \end{cases}$$

The prediction error is the predicted error over an independent validation set,

$$Err_{\mathcal{T}} = E[L(y, \widehat{f}(\boldsymbol{x}))|\mathcal{T}],$$

where both \boldsymbol{x} and \boldsymbol{y} are drawn randomly from their joint distribution, the training set, \mathcal{T} , is fixed, and the prediction error is the error for the specific training set.

Most methods in statistic estimates the expected prediction error, because it is a more fundamental characteristic of a learning algorithm, such as cross-validation, since it averages over whether we predict correct or not with a particular training set. The expected prediction error is given by,

$$Err = E[L(y, \hat{f}(\boldsymbol{x}))]$$
$$= E[Err_{\mathcal{T}}].$$

Computing the expected prediction error, Err, is difficult without a validation set, which in most cases is not available. Therefore, we consider to use the training error, which is the average loss over the training set, as an alternative to the prediction error, the training error is given as,

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \widehat{f}(x_i)), \qquad (3.1)$$

where N is the sample size. When a model becomes more complex, it uses more training data and is therefore able to adapt to a more complicated underlying data generating process. Hence, an increase in variance and a decrease in bias occur. The training error is not a good estimate for the prediction error, because the training error decreases as the model complexity increases. If the model complexity is high enough, the training error will reach zero. A zero training error will typically perform poorly, because it overfits the training data, as illustrated in Figure 3.1.



Figure 3.1: The behaviour of prediction and training errors as the model complexity is varied. The solid blue curve show the expected prediction error, Err, and the solid red curve shows the expected training error, $E[\overline{err}]$. The light blue curves is the training errors \overline{err} , and the light red curves is the conditional prediction errors $Err_{\mathcal{T}}$, for 100 training sets of size 50 each, as the model complexity is increased. Source: [Hastie et al., 2008].

3.1.1 The Bias-Variance Trade-Off

In Figure 3.1 the prediction error curve has a U-shape, which is a result of two competing properties of statistical learning methods. To explain these two properties further, we start by showing that the expected prediction error, for a input point, $(x) = x_i$, can always be decomposed into the sum of three fundamental quantities:

- The variance of $\widehat{f}(x_i)$.
- The squared bias of $\widehat{f}(x_i)$.
- The variance of the error terms ε .

Assume, $y = f((x)) + \varepsilon$, where $E(\varepsilon) = 0$ and $Var(\varepsilon) = \sigma_{\varepsilon}^{2}$.

The expected prediction error is defined using the squared error loss,

$$Err(x_i) = E[(y - \hat{f}(x_i))^2 | (x) = x_i]$$

$$= \sigma_{\varepsilon}^2 + [E[\hat{f}(x_i)] - f(x_i)]^2 + E[\hat{f}(x_i) - E[\hat{f}(x_i)]]^2$$

$$= \sigma_{\varepsilon}^2 + \text{Bias}^2(\hat{f}(x_i)) + \text{Var}(\hat{f}(x_i))$$

$$= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance.}$$
(3.2)

The irreducible error is the variance of the target around the true mean $f(x_i)$, and will always exist, unless $\sigma_{\varepsilon}^2 = 0$. The squared bias is the amount by which the average of our estimate differs from the true mean. The variance is the expected squared deviation of $\hat{f}(x_i)$.

To minimize the expected prediction error, Equation (3.2) shows that we need to select a method that simultaneously achieves low variance and low bias. Notice, that both the variance and the squared bias are non-negative quantities, which means that the expected prediction error can not be less than the irreducible error.

A general rule is, that when the complexity of a method increase, the variance increase and the bias decrease, as illustrated in Figure 3.1. Whether or not the prediction error decreases or increases is determined by the rate at which the variance and the bias change. E.g., as we increase the complexity of the method, the bias tends to initially decrease faster than the variance increases, causing the expected prediction error to decline. On the other hand, if the increase in complexity only has little effect on the bias, and at the same time significantly increase the variance, the prediction error increases.

The relationship mentioned in Figure 3.1 and in Equation (3.2) is what is called the bias-variance trade-off. It is called the bias-variance trade-off because it is relatively easy to either obtain a method with low bias and high variance, e.g., by drawing a curve that passes through every single

training observation, or a method with low variance and high bias, e.g., by fitting a horizontal line to the data. Therefore, the problem is to find a method where both variance and bias are as low as possible.

3.2 Cross-Validation

In the previous section we discuss how to estimate the prediction error if a validation set is available. Unfortunately, this is usually not the case, thus we introduce cross-validation, which estimate the prediction error from the training set by holding out a subset. The subset is then used as the validation set.

3.2.1 The Validation Set Approach

The validation set approach is a very simple strategy for estimating the prediction error on a set of observations. The validation set approach splits the observation randomly into two comparable subsets, a training set and a validation set. Various regression models are fitted on the training set, and their performance is evaluated on the validation set. The validation set error is usually obtained by the MSE, and results in an estimate of the prediction error.

The validation set approach is simple and easy to implement, but it has two potential concerns:

- The prediction error can be highly variable, due to the random choice of observations included in the training set and the validation set.
- The validation set approach divides the data set into two subset, which means that there are fewer observation to train the method on. Hence, the validation set may tend to overestimate the prediction error for the model, fitted on the entire data set, because statistical methods perform worse on small training sets.

In order to address these two concerns, a refinement of the validation set approach is introduced.

3.2.2 k-Fold Cross-Validation

The k-fold cross-validation, k-fold CV, divides the observation into k random groups, or folds, of approximately the same size. Here the first fold serves as the validation set, and the remaining k-1 folds as the training set. When the method is fitted on the k-1 folds, the Err₁ is computed on the observations in the held-out fold. This procedure is repeated k times, where, each time, a new group of observations is treated as the validation set. This results in k estimations of the prediction error, $\text{Err}_1, \text{Err}_2, \ldots, \text{Err}_k$, thus the k-fold CV estimate is computed by taking the average of the prediction errors,

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \text{Err}_i.$$
 (3.3)

When we use k-fold CV, we typically choose k = 5 or k = 10, which has shown to yield prediction error rate estimates, that does not contain high bias or high variance. The k-fold CV approach, with k = 5, is illustrated in Figure 3.2

1	2	3	5							
Train	Train	Validation	Train	Train						

Figure 3.2: The k-fold CV approach with k=5. In this case we fit the model to the first, second, fourth, and fifth group of the date, and calculate the prediction error of the fitted model by predicting on the third part of the data. Source: [Hastie et al., 2008]

The k-fold CV has a low variance, because Equation (3.3) is averaging the output of k fitted models which has a low correlation, due to the relative small overlap between the training sets in each model. The bias could be a problem, due to the way the learning method chooses the size of the training set. If we do not have enough training observations, the k-fold CV approach will have high bias. Hence, there is a bias-variance trade-off associated with the choice of k in k-fold CV.

In the last two sections, we have discussed the importance of low variance and low bias, but how can we check whether our learning algorithm suffers from a bias, or a variance problem, or both? To check whether our algorithm suffers from a bias or a variance problem, learning curves can be applied.

3.3 Model Selection

This section is further based on [Lutkepohl, 2005] and [Hayashi, 2000].

Model selection is the task of selecting the model which fits the the data best, from a group of models. A good model selection method balance goodness of fit with the complexity of the model.

The most commonly used model selection measures are

- The Akaike Information Criterion.
- The Bayesian Information Criterion.

3.3.1 The Akaike Information Criterion

The Akaike Information Criterion (AIC) deals with the trade-off between goodness of fit and the complexity of the model, but does not give an estimate of how well the model fits the data. The AIC value of a model is given by

$$AIC = -2 \cdot \text{loglik} + 2 \cdot d,$$

where loglik is the maximization of the log-likelihood function for the model, and d is the number of estimated parameters in the model.

When we use AIC as a model selecting measure, we choose the model with the lowest AIC value. This is because the AIC rewards the goodness of fit through the likelihood function, and at the same time penalize overfitting by including a penalty, as an increasing function of the number of estimated parameters. Hence, the lowest AIC measure of a model corresponds to the best trade-off between the goodness of fit and the minimum number of parameters.

3.3.2 The Bayesian Information Criterion

The Bayesian information criterion (BIC) is, like the AIC, a measure that gives an estimate of the model performance on a new data set. The BIC value of a model is given by

$$BIC = -2 \cdot \log lik + \log(N) \cdot d,$$

where loglik is the maximization of the log-likelihood function for the model, d is the number of estimated parameters in the model, and N is the size of the training set.

The only difference between the AIC and the BIC is the penalty which is imposed as the number of predictors are increased. The BIC penalty factor is $\log(N)$, whereas the AIC penalty term is 2. Hence, the BIC penalty factor grows logarithmic as the number of observations increases, whereas the AIC penalty factor is constant. This means that the BIC prefers more parsimonious models compared to the AIC, as it penalizes the model complexity harder.

Chapter 4

DYNAMIC FACTOR MODELS AND PRINCIPAL COMPONENT ANALYSIS

This section is based on [Stock and Watson, 2010], [Arrow and Intriligator, 2006], and [Fornaro, 2011].

This chapter discuss the use of dynamic factors models, first introduced by [Geweke, 1977], and principal component analysis, which were extended to dynamic principal components analysis by [Brillinger, 1964], as methods for forecasting with many predictors.

The reason that dynamic factor models are widely used in forecasting with many predictors is, that they make it possible to investigate not easily measured variables by assemble multiple predictors into a few underlying factors, which account for much of the total variation in the data.

The main idea in dynamic factor models is to split each of the predictors into a common component⁴, expressed in terms of a few common factors and their lags, and an idiosyncratic disturbance,

$$\boldsymbol{x}_{it} = \boldsymbol{\lambda}_i(L)' \boldsymbol{f}_t + \boldsymbol{u}_{it} \tag{4.1}$$

where \mathbf{f}_t is a $q \times 1$ vector of the unobserved common factors, $\lambda_i(L)$ is a $q \times 1$ vector lag polynomial, called the dynamic factor loading for \mathbf{x}_{it} , and \mathbf{u}_{it} is the idiosyncratic disturbance. $\lambda_i(L)\mathbf{f}_t$ is called the common component of the *i*'th series, and \mathbf{u}_{it} is assumed to be an independent stationary process, which means that it is uncorrelated with both leads and lags of the common factors and with the other idiosyncratic components. Hence,

$$E[\boldsymbol{f}_t \boldsymbol{u}_{is}] = 0 \quad \forall i, s.$$

The unobserved common factors, \boldsymbol{f}_t , follows a time series process, which is commonly taken to be a vector autoregression (VAR),

$$\boldsymbol{f}_t = \boldsymbol{\Gamma}(L) \boldsymbol{f}_{t-1} \boldsymbol{\eta}_t \tag{4.2}$$

where $\Gamma(L)$ is an $q \times q$ lag polynomial matrix, and η_t is a $q \times 1$ vector consistent of idiosyncratic components.

 $^{^{4}}$ Because we split the predictors into common components, the common components are sensitive to the unit of measurements in the predictors. Hence, all the predictors are standardized before they are used to construct the common components.

Forecasting with dynamic factor models is done through a two-step process,

- 1. The factors are estimated by the predictors.
- 2. The relationship between the variable to be forecast and the factors is estimated by linear regression.

The single forecasting equation for y_t can be derived from Equation (4.1) and (4.2) by augment x_t in the expression for y_t , such that

$$\boldsymbol{y}_t = \boldsymbol{\lambda}_y(L) \boldsymbol{f}_t + \boldsymbol{u}_{yt},$$

where u_{yt} is distributed independently of f_t and u_{it} .

$$E \left[\boldsymbol{y}_{t+1} | \boldsymbol{y}_{t}, \boldsymbol{x}_{t}, \boldsymbol{f}_{t}, \boldsymbol{y}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{f}_{t-1}, \dots, \dots \right] = E \left[\boldsymbol{\lambda}_{y}(L) \boldsymbol{f}_{t+1} + \boldsymbol{u}_{yt+1} | \boldsymbol{y}_{t}, \boldsymbol{x}_{t}, \boldsymbol{f}_{t}, \boldsymbol{y}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{f}_{t-1}, \dots, \dots \right]$$

$$= E \left[\boldsymbol{\lambda}_{y}(L) \boldsymbol{f}_{t+1} | \boldsymbol{y}_{t}, \boldsymbol{x}_{t}, \boldsymbol{f}_{t}, \boldsymbol{y}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{f}_{t-1}, \dots \right] +$$

$$E \left[\boldsymbol{u}_{yt+1} | \boldsymbol{y}_{t}, \boldsymbol{x}_{t}, \boldsymbol{f}_{t}, \boldsymbol{y}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{f}_{t-1}, \dots \right]$$

$$= E \left[\boldsymbol{\lambda}_{y}(L) \boldsymbol{f}_{t+1} | \boldsymbol{f}_{t}, \boldsymbol{f}_{t-1}, \dots \right] +$$

$$E \left[\boldsymbol{u}_{yt+1} | \boldsymbol{u}_{yt}, \boldsymbol{u}_{yt-1} \dots \right]$$

$$(4.3)$$

$$= \boldsymbol{\beta}(L)' \boldsymbol{f}_t + \boldsymbol{\gamma}(L) \boldsymbol{z}_t + \boldsymbol{\varepsilon}_{t+1}, \qquad (4.4)$$

where $\boldsymbol{z}_t = \boldsymbol{y}_t$, such that the regression can include the predictors and not just the lagged variable \boldsymbol{y}_t , $E\left[\boldsymbol{\varepsilon}_{t+1}|\boldsymbol{y}_t, \boldsymbol{x}_t, \boldsymbol{f}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{f}_{t-1}, \ldots\right] = 0$, Equation (4.3) follows from Equation (4.2), and Equation (4.4) follows from Equation (4.1). Hence, if \boldsymbol{f}_t are known for all t, then $\boldsymbol{\beta}(L)'$ and $\boldsymbol{\gamma}(L)$ can be estimated using ordinary least squares and a h-period forecast can be computed.

Before forecasting, two issues are addressed. The first issue is to estimate the common factors, and the second issue is to determine the number of factors. In order to estimate the common factors to include in the model, we introduce some estimation methods.

4.1 Estimation Methods

According to [Stock and Watson, 2010] there exist three generations of time-domain estimation methods of dynamic factor models:

- First generation -maximum likelihood via Kalman filter.
- Second generation –nonparametric averaging methods.

- Cross-sectional averaging.
- Principal component estimation.
- Generalized principal component estimation.
- Dynamic principal component.
- Third generation –hybrid principal components and space state methods.
 - State space with static factors.
 - State space with dynamic factors.

The first generation which us maximum likelihood to estimate the factors has been used, inter alia, by [Stock and Watson, 1989] and [Diebold et al., 2006]. It has provided great results in frameworks where the number of predictors have been modest. However, this estimation method has a disadvantage when the number of predictors is large. When multiple predictors are included, the computation time rises, due to the demand of maximizing the log-likelihood function, and for this reason other methods have been used when high-dimensional data is involved. The most widely used estimation method of the common factors, when dealing with a large number of predictors, is principal component analysis, which in this context is introduced by [Stock and Watson, 2002]. Since we focus on forecasting with several predictors, we use the principal component analysis as our estimation method.

4.1.1 Estimation of Factors by Principal Component Analysis

Define y_t as the time series to be forecast, and x_t as the N-dimensional multiple time series of predictors. The model with r common latent factors, F_t , is defined as

$$\boldsymbol{x}_t = \boldsymbol{\Lambda} \boldsymbol{F}_t + \boldsymbol{u}_t \tag{4.5}$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{\beta}' \boldsymbol{F}_t + \boldsymbol{\gamma}' \boldsymbol{z}_t + \boldsymbol{\varepsilon}_{t+1}. \tag{4.6}$$

Equation (4.5) and (4.6) are derived from Equation (4.1) and (4.4) by assuming that the lag polynomials $\lambda_i(L)$, $\beta(L)$, and $\gamma(L)$ have finite order p.

The variables in Equation (4.5) are $\mathbf{F}_t = [f_t f_{t-1} \dots f_{t-p+1}]'$, Λ which is a matrix consisting of zeros and the coefficients of $\lambda_i(L)$, and $\boldsymbol{\beta}$ which is a vector consisting of the parameters composed of the elements of $\boldsymbol{\beta}(L)$. The term $\boldsymbol{\beta}' \mathbf{F}_t$ in Equation (4.6) can be replaced by the distribution lag of \mathbf{F}_t if the numbers of lags in $\boldsymbol{\beta}$ exceed the number of lags in Λ .

Equation (4.5) and (4.6) transform the dynamic factor model into a static factor model, which includes r static factors, F_t , consisting of the current and lagged values of the q dynamic factors,

 f_t , where $r \leq pq$. If one or more lagged dynamic factors are redundant, r is strictly less than pq.

Assume that F_t and u_t are uncorrelated at all leads and lags, thus the covariance matrix of x_t is defined as

$$\Sigma_{xx} = \Lambda \Sigma_{FF} \Lambda' + \Sigma_{uu}.$$

where Σ_{FF} and Σ_{uu} are the variance matrices of F_t and u_t , respectively

To estimate the common factors F_t and the loading matrix Λ we solve the nonlinear least-squares problem,

$$\min_{\boldsymbol{F}_{1},\dots,\boldsymbol{F}_{T},\boldsymbol{\Lambda}} V_{t}\left(\boldsymbol{\Lambda},\boldsymbol{F}\right) = \left(T\right)^{-1} \sum_{t=1}^{T} \left(\boldsymbol{x}_{t} - \boldsymbol{\Lambda}\boldsymbol{F}_{t}\right)' \left(\boldsymbol{x}_{t} - \boldsymbol{\Lambda}\boldsymbol{F}_{t}\right),$$
(4.7)

subject to $\Lambda' \Lambda = I_r$, where T is the number of observations. First, we minimize Equation (4.7) with respect to F_t , given Λ , to obtain

$$\widehat{\boldsymbol{F}}_{t}\left(\boldsymbol{\Lambda}\left(\boldsymbol{\Lambda}^{\prime}\boldsymbol{\Lambda}\right)^{-1}\right) = \left(\boldsymbol{\Lambda}^{\prime}\boldsymbol{\Lambda}\right)^{-1}\boldsymbol{\Lambda}^{\prime}\boldsymbol{x}_{t}.$$
(4.8)

By substituting Equation (4.8) into Equation (4.7), the concentrated objective function becomes,

$$\min_{\mathbf{\Lambda}} T^{-1} \sum_{t=1}^{T} \mathbf{x}_{t}^{\prime} \left[\mathbf{I} - \mathbf{\Lambda} \left(\mathbf{\Lambda}^{\prime} \mathbf{\Lambda} \right)^{-1} \mathbf{\Lambda} \right] \mathbf{x}_{t}.$$
(4.9)

Minimizing Equation (4.9) is equivalent to maximizing

$$\max_{\boldsymbol{\Lambda}} \operatorname{tr}\left\{\left(\left(\boldsymbol{\Lambda}'\boldsymbol{\Lambda}\right)^{-1/2}\right)' \boldsymbol{\Lambda}' \left(T^{-1} \sum_{t=1}^{T} \boldsymbol{x}_{t} \boldsymbol{x}_{t}'\right) \boldsymbol{\Lambda} \left(\boldsymbol{\Lambda}'\boldsymbol{\Lambda}\right)^{-1/2}\right\} = \max_{\boldsymbol{\Lambda}} \operatorname{tr}\left\{\left(\left(\boldsymbol{\Lambda}'\boldsymbol{\Lambda}\right)^{-1/2}\right)' \boldsymbol{\Lambda}' \widehat{\boldsymbol{\Sigma}}_{xx} \boldsymbol{\Lambda} \left(\boldsymbol{\Lambda}'\boldsymbol{\Lambda}\right)^{-1/2}\right\},$$
(4.10)

where $\widehat{\Sigma}_{xx} = \left(T^{-1}\sum_{t=1}^{T} X_t X_t'\right)$ and tr(·) represents matrix trace. Note, that a trace of a squared matrix is defined as the sum of its diagonal elements, or the sum of all its eigenvalues.

Equation (4.10) is then equivalent to $\max_{\Lambda} \Lambda' \widehat{\Sigma}_{xx} \Lambda$ subject to $\Lambda' \Lambda = I_r$.

To estimate \mathbf{F}_t we set the loading estimate $\widehat{\mathbf{\Lambda}}$ equal to the first r eigenvectors of $\widehat{\mathbf{\Sigma}}_{xx}$, corresponding to its r largest eigenvalues. Hence, the principal components estimator of \mathbf{F}_t is given by

$$\widehat{F}_t = \widehat{\Lambda}' x_t,$$

due to $\widehat{\Lambda}' \widehat{\Lambda} = I_r$. \widehat{F}_t is then a vector consisting of the first r principal components of x_t . Hence, the factor estimates are the principal components of the predictors defined as

$$\widehat{F}_{it} = \widehat{\varphi}'_j x_t, \qquad i =, \dots, r,$$

where $\hat{\varphi}'_j$ denotes the normalized eigenvector of $\hat{\Sigma}_x$ corresponding to the *i*'th largest eigenvalues. The principal components are computed, such that the first component accounts for as much data variability as possible, thus has the largest variance, the second component is then a linear combination of the variables that is uncorrelated with the first principal component, and has the largest variance subject to this constraint. Subsequently, each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components. Hence, the new set of time series are uncorrelated and the first principal components in the series retains the majority of the variation present in the original time series.

The *h*-period ahead forecast can be determined by Equation (4.6). Once the common factors have been estimated by the principal components the forecast is computed by regressing y_{t+h} against \mathbf{F}_t and \mathbf{z}_t , and then forecasting on y_{t+h} .

4.2 Selecting the Number of Factors

Selecting the correct number of factors, to include in the model, plays an important role in the validity of the factor models. The literature proposes numerous was of estimating the correct number of factors. However, when applying dynamic factor models to many predictors, not to many options are available. One option is to use a model selection method to identify the factors that belong in the forecasting equation, (4.6), [Stock and Watson, 2002] shows that this can be achieved by an information criterion. A second approach is to estimate the number of factors entering the full dynamic factor model, [Bai and Ng, 2002] shows that this can be achieved by a suitable information criteria which they provide.

When we apply the dynamic factor model with principal components, we choose to use cross-validation, see Section 3.2, in order to determine the optimal number of factors to include in our regression. Cross-validation is widely used to determine the optimal number of components in dynamic factor models with principal component analysis, see e.g. [Josse and Husson, 2012], and offers an easy implementation in R.
CHAPTER 5

SHRINKAGE METHODS

This chapter is based on [Hastie et al., 2008] and [James et al., 2013].

An alternative to the dynamic factor model approach is to use shrinkage estimators in our regression. Unlike dynamic factor models, the shrinkage estimators does not reduce the dimensionality of the date by extracting common factors. Instead they focus on estimating a regression model over a constrained parameter space.

The main idea of the shrinkage methods is to use them for linear regression in order to modify the least squares estimates by imposing a penalty on their size. In statistics, a linear regression models the relationship between the dependent variable and the predictors. The linear regression model has the form

$$y_i = \beta_0 + \sum_{j=1}^p x_{ij} \beta_j,$$
 (5.1)

and can be solved by ordinary least squares, which finds the coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$ that minimize the residual sum of squares,

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2.$$
 (5.2)

The solution to the ordinary least squares is found by differentiating Equation (5.2) with regard to β , and leads to a closed-form expression for the estimated value of the unknown parameter

$$\widehat{\boldsymbol{\beta}}^{OLS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}.$$
(5.3)

The ordinary least squares estimate performs rather well when the number of observations, n, is much larger than the number of predictors, p. However, since we focus on forecasting with several predictors, we have p > n. This causes a problem for the ordinary least squares estimation because, when p > n some of the predictors might be colinear, which means that X does not have full rank. If X is not of full rank, then X'X is singular and there is no longer a unique least squares estimate. Furthermore, when the number of predictors are large, the least squares estimates will often have low bias but high variance. Therefore, we introduce some methods which shrinks the coefficient estimates towards zero or setting some equal to zero. These methods introduce some bias but reduce the variance of the predicted values, and thus may improve the overall prediction accuracy measured in terms of the mean-squared error.

There exist several shrinkage methods, the ones that we implement are the ridge regression, lasso, grouped lasso, and elastic net methods.

5.1 Ridge Regression

Ridge regression is a technique used to analyze multiple regression data that suffer from multicollinearity, and was first introduced by [Hoerl and Kennard, 1970]. It is a technique very similar to least squares except that its coefficients are estimated by minimizing a slightly different quantity than Equation (5.2).

The ridge regression coefficients that estimate $\widehat{\boldsymbol{\beta}}^R$ are the values that minimize

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$
(5.4)

where $\lambda \geq 0$ is the tuning parameter, which controls the amount of shrinkage. Note, that the first expression in Equation (5.4) is equal to Equation (5.2), which means that ridge regression, as least squares, seeks coefficients that fits the data well by making the residual sum of squares small. Hence, the only difference between ridge regression and least squares is the shrinkage penalty, the second term in Equation (5.4), which shrinks the estimates of β_j towards zero.

The ridge regression coefficient estimates are not scale invariant, thus $\boldsymbol{X}_{j} \boldsymbol{\hat{\beta}}_{j,\lambda}^{R}$ depends not only on the λ value, but also on the scaling of the *j*'th predictor. Therefore, if the predictors are not in the same unit of measurement, we standardize them before solving Equation (5.4), using the following formula

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(x_{ij} - \overline{x}_{j}\right)^{2}}}$$

where the denominator is the estimated standard deviation of the jth predictor.

We now turn to the solution of Equation (5.4), and the fact that it can be divided into two

separate parts. We rewrite Equation (5.4) by adding $\overline{x}_j - \overline{x}_j$, a zero value, such that

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \left(x_{ij} + \overline{x}_j - \overline{x}_j \right) \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$
$$= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \overline{x}_j \beta_j - \sum_{j=1}^{p} \left(x_{ij} - \overline{x}_j \right) \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{5.5}$$

and define the centered values of β as

$$\beta_0^{\text{centered}} = \beta_0 - \sum_{j=1}^p \overline{x}_j \beta_j \tag{5.6}$$

$$\beta_j^{\text{centered}} = \beta_j, \quad j = 1, 2, \dots, P.$$
(5.7)

If Equation (5.6) and (5.7) is inserted into Equation (5.5) we obtain

$$\sum_{i=1}^{n} \left(y_i - \beta_0^{\text{centered}} - \sum_{j=1}^{p} \left(x_{ij} - \overline{x}_j \right) \beta_j^{\text{centered}} \right)^2 + \lambda \sum_{j=1}^{p} \left(\beta_j^{\text{centered}} \right)^2.$$
(5.8)

So fare, we have reparametrized and centered the x_i 's into having zero mean, such that all the points are at the origin, thus from Equation (5.4) to (5.8) only the intercept is modified. The value of the intercept, $\beta_0^{\text{centered}}$, is computed by setting the derivative in Equation (5.8), with respect to $\beta_0^{\text{centered}}$, equal to zero, such that

$$0 = \sum_{i=1}^{n} 2 \left(y_i - \beta_0^{\text{centered}} - \sum_{j=1}^{p} \left(x_{ij} - \overline{x}_j \right) \beta_j^{\text{centered}} \right)$$
$$\beta_0^{\text{centered}} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \left(x_{ij} - \overline{x}_j \right) \beta_j^{\text{centered}} \right)$$
$$= \frac{1}{n} \sum_{i=1}^{n} y_i.$$

Hence, the value of β_0 can be computed as a separate part without any interference from the other coefficients. The remaining coefficients are estimated by a ridge regression without the intercept, using the centered x_{ij} 's. Henceforth, Equation (5.4) is in matrix form, and we assume that the centering has been done, such that the input matrix has p, and not 1 + p columns,

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\boldsymbol{\beta}$$

= $\boldsymbol{y}'\boldsymbol{y} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{y} + \lambda\boldsymbol{\beta}'\boldsymbol{\beta}.$ (5.9)

We obtain the ridge regression solution by differentiating Equation (5.9) with respect to β ,

$$2X'X\beta - 2X'y + 2\lambda\beta,$$

and setting the derivative equal to zero, such that

$$2\mathbf{X}'\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}'\mathbf{y} + 2\lambda\boldsymbol{\beta} = 0$$

$$2(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = 2\mathbf{X}'\mathbf{y}$$

$$\widehat{\boldsymbol{\beta}}^{R} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}.$$

Notice, that in contrast with the least squares regression, the ridge regression solution adds a positive constant, λ , to $\mathbf{X}'\mathbf{X}$'s diagonal, which makes the problem nonsingular even if $\mathbf{X}'\mathbf{X}$ is not of full rank. Hence, unlike in the least squares approach, in the ridge regression $(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})$ is always invertible, and a unique solution is thereby guaranteed.

To gain insight into the nature of ridge regression and the role of the tuning parameter λ , we perform a singular value decomposition (SVD) of the centered input matrix \boldsymbol{X} . The SVD of \boldsymbol{X} has the form

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}',\tag{5.10}$$

where U is a $N \times p$ matrix, which columns spans the column space of X, V is a $p \times p$ matrix, which columns spans the row space of X, and D is a $p \times p$ matrix with diagonal entries equal to the singular values of X, $d_1 \ge d_2 \ge \ldots \ge d_p \ge 0$. From the representation (5.10) and (5.3), we can derive a new expression for X'X as,

$$\begin{aligned} X'X &= VDU'UDV' \\ &= VD^2V'. \end{aligned} \tag{5.11}$$

Using Equation (5.11) we can compute the least squares fitted values as

$$\widehat{\boldsymbol{y}} = \boldsymbol{X} \widehat{\boldsymbol{\beta}}^{OLS}$$

$$= \boldsymbol{X} (\boldsymbol{X}' \boldsymbol{X})^{-1} \boldsymbol{X}' \boldsymbol{y}$$

$$= \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}' (\boldsymbol{V} \boldsymbol{D}^2 \boldsymbol{V}')^{-1} \boldsymbol{V} \boldsymbol{D} \boldsymbol{U}' \boldsymbol{y}$$

$$= \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}' \boldsymbol{V}^{-T} \boldsymbol{D}^{-2} \boldsymbol{V}^{-1} \boldsymbol{V} \boldsymbol{D} \boldsymbol{U}' \boldsymbol{y}$$

$$= \boldsymbol{U} \boldsymbol{U}' \boldsymbol{y}$$

$$= \sum_{j=1}^{P} \boldsymbol{u}_{j} (\boldsymbol{u}_{j} \boldsymbol{y}). \qquad (5.12)$$

Note, that the Equation (5.12) is based on the fact that U'y are the coordinates of y with respect to the orthonormal basis spanned by the columns of U.

Using the same procedure as before, the ridge regression fitted values are computed as

$$\begin{aligned} \mathbf{X}\widehat{\boldsymbol{\beta}}^{R} &= \mathbf{X} \left(\mathbf{X}\mathbf{X}' + \lambda \mathbf{I} \right)^{-1} \mathbf{X}' \mathbf{y} \\ &= \mathbf{U} \mathbf{D} \mathbf{V}' \left(\mathbf{V} \mathbf{D}^{2} \mathbf{V}' + \lambda \mathbf{V} \mathbf{V}' \right)^{-1} \mathbf{V} \mathbf{D} \mathbf{U}' \mathbf{y} \\ &= \mathbf{U} \mathbf{D} \mathbf{V}' \left(\mathbf{V} \mathbf{V}' \left(\mathbf{D}^{2} + \lambda \mathbf{I} \right) \right)^{-1} \mathbf{V} \mathbf{D} \mathbf{U}' \mathbf{y} \\ &= \mathbf{U} \mathbf{D} \mathbf{V}' \mathbf{V}^{-T} \mathbf{V}^{-1} \left(\mathbf{D}^{2} + \lambda \mathbf{I} \right)^{-1} \mathbf{V} \mathbf{D} \mathbf{U}' \mathbf{y} \\ &= \mathbf{U} \mathbf{D} \left(\mathbf{D}^{2} + \lambda \mathbf{I} \right)^{-1} \mathbf{D} \mathbf{U}' \mathbf{y} \\ &= \sum_{j=1}^{P} \mathbf{u}_{j} \frac{d_{j}^{2}}{d_{j}^{2} + \lambda} \mathbf{u}_{j}' \mathbf{y}, \end{aligned}$$
(5.13)

where $\boldsymbol{D} \left(\boldsymbol{D}^2 + \lambda \boldsymbol{I} \right)^{-1}$ is a diagonal matrix with elements given by $\frac{d_j^2}{d_j^2 + \lambda}$. If we compare the least squares regression, Equation (5.12), with the ridge regression, Equation (5.13), it is evident that the two expressions are quite similar, except for the fact that the ridge regression scales the inner product $\boldsymbol{U}'\boldsymbol{y}$ by the factor $\frac{d_j^2}{d_i^2 + \lambda}$.

Equation (5.13) demonstrates the huge impact that the value of λ has on the coefficients, when $\lambda = 0$ there is no shrinkage of the inner product, thus the ridge regression becomes equivalent to the least squares. Conversely, as $\lambda \to \infty$ the impact of the shrinkage penalty becomes larger, which sends the ridge regression coefficients towards zero. Hence, choosing the right value of λ is an important task in ridge regression, a task which is normally solved by cross-validation, see Section 3.2, in which we need a grid of λ values to compute the cross-validation error on. When the grid of cross-validation errors are computed, we select the λ value with the smallest error, and re-fit the model using all of the available observations and the selected value of λ .

[Weatherwax and Epstein, 2013] computes the grid of λ values by the effective degrees of freedom. The effective degrees of freedom for the ridge regression is a monotone decreasing function given as

$$df(\lambda) = \operatorname{tr} \left[\boldsymbol{X} (\boldsymbol{X}' \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}' \right]$$

=
$$\operatorname{tr} \left[\boldsymbol{U} \boldsymbol{D} \left(\boldsymbol{D}^2 + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{D} \boldsymbol{U}' \right].$$
(5.14)

Recall, that the trace of a square matrix is defined as the sum of its diagonal elements, or the sum of all its eigenvalues. Hence, the singular value decomposition, for which the matrix D is defined as a $p \times p$ matrix whose diagonal entries equal the singular values of X, provides eigenvalues equal to the elements given by $\frac{d_j}{d_j\lambda}$, which means that Equation (5.14) can be expressed as

$$df(\lambda) = \sum_{j=1}^{p} \frac{d_j}{d_j \lambda}.$$
(5.15)

Usually, when we fit a linear regression with p variables, the degrees of freedom is equal to the number of free parameters, p, and although all p coefficients in a ridge regression fit will be non-zero, the idea is, that the parameters are fit in a restricted fashion controlled by λ . To find the grid of λ values, we use Newton's root finding method⁵ to solve

$$\sum_{j=1}^{p} \frac{d_j}{d_j \lambda} = k$$

for λ , where k = 1, 2, ..., p and represents all possible degrees of freedom. We start by defining a function $f(\lambda)$ as

$$f(\lambda) = \sum_{j=1}^{p} \frac{d_j}{d_j \lambda} - k, \qquad (5.16)$$

and want to chose the value λ such that Equation (5.16) is equal to zero. Hence, we use Newton's algorithm, with a starting value of λ_0 , such that

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda)}{f'(\lambda)},$$

where

$$f'(\lambda) = -\sum_{j=1}^{p} \frac{d_j}{(d_j \lambda)^2}.$$

As we are looking for p values of λ , we start by solving the problems for k = p, p-1, p-2, ..., 1. Note, that when k = p the value of λ that solves $df(\lambda) = p$ is $\lambda = 0$. Since Newton's algorithm is iterative, we use the estimated λ in the previous run of the algorithm as the initial guess for the current run of the algorithm.

5.1.1 Comparison with Principal Component Regression

The ridge regression has great similarities with the principal component regression, since both methods does not result in models that relies upon a small set of the original features, but simply scales the already existing ones. The ridge regression shrinks the coefficients of the principal components depending on the size of the corresponding eigenvalue.

To illustrate this, recall, the SVD of X, Equation (5.10), and the eigenvalue decomposition of X'X, Equation (5.11). Furthermore, notice that the columns of V are the eigenvectors v_j , which are called the principal components directions of X, discussed in Section 4.1.1. Moreover,

⁵Newton's root finding method is an iterative method which looks for solution of the equation f(x) = 0 as fixed points of $g(x) = x - \frac{f(x)}{f'(x)}$, thus it choose x such that x = g(x). The algorithm for Newton's method is as follows: 1) Pick a starting value x_0 , 2) For each estimated x_n , calculate a new estimate $\lambda_{n+1} = \lambda_n - \frac{f(\lambda)}{f'(\lambda)}$. 3) repeat step 2 until the estimates are close enough to a root, or until the method fails. Source: [Adams and Essex, 2010].

since Equation (5.10) is equal to XV = UD, the principal components can be rewritten, such that

$$oldsymbol{z}_m = oldsymbol{X} v_m$$

= $oldsymbol{u}_m d_m$

where u_m is the normalized principal component, and since $\mathbf{X}'\mathbf{X}$ is the sample covariance matrix up to factor N, we know that the variance of the principal components are

$$Var(\boldsymbol{z}_m) = Var(\boldsymbol{X}v_m)$$
$$= \frac{d_j^2}{N}.$$

Combining the above knowledge with the discussion in Section 4.1.1, notice, that $\boldsymbol{z}_m = \boldsymbol{u}_m d_m$ is the *m*'th principal component of \boldsymbol{X} , which have maximum variance $\frac{d_j^2}{N}$ and is uncorrelated with $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{m-1}$.

By Equation (5.13), this shows that the ridge regression coefficients are computed by shrinking all normalized principal components, where the low variance principal components are shrunk more than high variance principal components.

In contrast the principal component regression uses the principal component analysis to compute r < p new high-variance coefficients, and then discards the p-r smallest eigenvalue components, where r is the number of components to include in the regression, and p is the number of predictors. Hence, ridge regression can be seen as a continuous version of principal component regression.

5.2 The Least Absolute Shrinkage and Selection Operator (lasso)

This section is further based on [Hastie et al., 2015] and [Wright, 2015].

Like ridge regression, the lasso, first introduced by [Tibshirani, 1996], is a shrinkage method. However, unlike ridge regression, which includes all of the p predictors, the lasso excludes some of the predictors by setting them equal to zero; this property makes the lasso a feature selecting method.

The lasso coefficients that estimates $\widehat{\boldsymbol{\beta}}_{\lambda}^{L}$ are the values that minimize the quantity

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$
 (5.17)

If we compare Equation (5.17) with (5.2) and (5.4) we see, that the first expression is equal to the least squares cost function, and that the only difference between the ridge regression and the lasso is the shrinkage penalty, where the lasso penalty $|\beta_j|$ is an ℓ_1 norm and the ridge regression penalty β_j^2 is an ℓ_2 norm. Hence, as with ride regression, the lasso shrinks the coefficient estimates towards zero, but when the tuning parameter λ is sufficiently large, the ℓ_1 penalty forces some of the coefficient estimates to be exactly zero.

To understand why the ℓ_1 penalty, unlike the ℓ_2 , causes some of the coefficients to be exactly zero, look at Figure 5.1 and note, that Equation (5.17) and (5.4) can be expressed as

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \le t$$
(5.18)

and

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \le t$$
 (5.19)

respectively. We can interpret Equation (5.18) and (5.19) such that when we perform lasso or ridge regression, we seek the set of coefficient estimates that yields the smallest residual sum of squares, subject to the constraint t which determines how large $\sum_{j=1}^{p} |\beta_j|$ or $\sum_{j=1}^{p} \beta_j^2$ can be.



Figure 5.1: Illustration of the estimation of the lasso (left) and the ridge regression (right). Here, the blue diamond and circle represents the constraint regions for the lasso and ridge regression respectively, the ellipses centered around $\hat{\beta}$ represent regions with constant residual sum of squares, and $\hat{\beta}$ represent the least squares solution. Source:[James et al., 2013]

In Figure 5.1 the least squares solution is located at $\hat{\beta}$, the blue diamond and circle represents the constraints in Equation (5.18) and (5.19) respectively, and the ellipses centered around $\hat{\beta}$ represents regions with constant residual sum of squares. If we consider the case where p = 2, then according to Equation (5.18) and (5.19) the coefficient estimates are given at the intersection between the constant residual sum of squares and the constraint region. Hence, the ridge regression constraint region $\beta_1^2 + \beta_2^2 \leq t^2$ spans a circle, which means that the intercept between the constrained region and the residual sum of squares normally does not occur at an axis. On the other hand, the lasso constraint region $|\beta_1| + |\beta_2| \leq t$ spans a diamond, which means that the intersection normally occurs at an axis. In the example illustrated in Figure 5.1, the lasso will only include β_2 in the model since the constraint region intersects with the ellipse at $\beta_1 = 0$, whereas the ridge regression includes both β_1 and β_2 because the constant residual sum of squares intersects with the constraint region just to the right of the y-axis.

The constraint region for the lasso is a polyhedron when p = 3 and a polytope when p > 3, for the ridge regression it is a sphere when p = 3 and a hypersphere when p > 3. Hence, as $p \to \infty$ the number of coefficients which the lasso ascribe zero value increases due to the many corners, flat edges, and faces of the polyhedron and polytope.

Due to these different penalties, which shrinkage method to choose depends on the scenario. The ridge regression will normally perform better than the lasso, in terms of prediction error, in cases where all the predictors are related to the response. This is a consequence of how the lasso implicitly assumes that some of the coefficients are equal to zero, and thus exclude some related predictors for the estimate. Opposite, the lasso will tend to perform better in cases where not all of the predictors are directly related to the response. This is a consequence of how the ridge regression does not exclude any coefficient, and thus include some irrelevant predictors for the estimate. Hence, non of the two methods will dominate the other.

Following the same procedure as for the ridge regression, we re-parametrize the intercept β_0 in Equation (5.17) by standardizing the predictors, and thereby find the separate solution for the intercept $\beta_0 = \frac{1}{n} \sum_{i=1}^{n} y_i$. Because of this standardization, we can fit a model without the intercept, however, in contrast with ridge regression the solution to the lasso is non-linear because of the ℓ_1 penalty. The ℓ_1 penalty makes the lasso problem a quadratic programming problem, with a convex constrain. Although there are many quadratic programming methods for solving the lasso, we follow [Hastie et al., 2015], and use the coordinate descent algorithm in our implementation.

5.2.1 The Coordinate Descent Algorithm and the Lasso

Coordinate descent algorithms, see [Wright, 2015] for an extensive review, solves optimization problems by successively performing approximate minimization along coordinate directions. Each iteration in the algorithm is obtained by fixing nearly all of the predictors at their current values, and then update the coefficient β_j by approximately minimizing the objective function with respect to the remaining coefficients.

We assume that both y_i and the x_{ij} 's have been standardized such that $\frac{1}{n} \sum_i y_i = 0$, $\frac{1}{n} \sum_i x_{ij} = 0$, and $\frac{1}{n} \sum_i x_{ij}^2 = 1$, furthermore, for convenience, we rewrite the criterion, Equation (5.17), in the Lagrangian form as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\left\{\frac{1}{2n}||\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}||_2^2+\lambda||\boldsymbol{\beta}||_1\right\}.$$
(5.20)

Single Predictor: Soft Thresholding

We start by considering the single predictor setting, in which the problem to be solved is

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_i x_{ji} \right)^2 + \lambda |\boldsymbol{\beta}| \right\}.$$
(5.21)

The gradient with respect to β for Equation (5.21) can only be computed in the cases where $\beta \neq 0$, since the absolute value function at zero does not have a derivative. The gradient with respect to β for $\lambda \neq 0$ is

$$\frac{1}{2n} \left(2 \boldsymbol{X}' \boldsymbol{X} \boldsymbol{\beta} - 2 \boldsymbol{X}' \boldsymbol{y} \right) + \lambda,$$

and if we set this equation equal to zero and isolate β we have that

$$egin{aligned} \widehat{oldsymbol{eta}}^L = \left\{ egin{aligned} rac{1}{n} \langle oldsymbol{x}, oldsymbol{y}
angle - \lambda & ext{if } rac{1}{n} \langle oldsymbol{x}, oldsymbol{y}
angle > \lambda \ 0 & ext{if } rac{1}{n} \langle oldsymbol{x}, oldsymbol{y}
angle & \leq \lambda \ rac{1}{n} \langle oldsymbol{x}, oldsymbol{y}
angle + \lambda & ext{if } rac{1}{n} \langle oldsymbol{x}, oldsymbol{y}
angle & < -\lambda \ \end{aligned}
ight.$$

Note, that we can write this in a more simple form, by using the soft-thresholding operator

$$S = \operatorname{sign}(x)(|x| - \lambda)_+, \tag{5.22}$$

which translates its argument x towards zero by the amount λ , and sets it to zero if $|x| \leq \lambda^6$. Hence, by Equation (5.22) we can express $\hat{\beta}$ as

$$\widehat{\boldsymbol{\beta}}^{L} = \mathcal{S}_{\lambda}\left(rac{1}{n} \langle \boldsymbol{x}, \boldsymbol{y}
angle
ight).$$

We can now derive a simple coordinate scheme for solving the full lasso problem (5.20), by using the cyclic coordinate descent.

 $^{^{6}(\}cdot)_{+}$ denotes the positive part of $(\cdot) \in \mathbb{R}$, equal to (\cdot) if $(\cdot) > 0$ and 0 otherwise.

Multiple Predictors: Cyclic Coordinate Descent

If we write the objective in Equation (5.20) as

$$\frac{1}{2n}\sum_{i=1}^{n}\left(y_{i}-\sum_{k\neq j}x_{ik}\beta_{k}-x_{ij}\beta_{j}\right)^{2}+\lambda\sum_{k\neq j}|\beta_{k}|+\lambda|\beta_{j}|,$$

the solution for each β_j is achieved by removing the outcome from all but the j'th predictor using the partial residual $r_i^{(j)} = y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k$. In terms of the partial residual, the j'th predictor is updated as

$$\widehat{\beta}_{j}^{L} = \mathcal{S}_{\lambda} \left(\frac{1}{n} \langle \boldsymbol{x}_{j}, \boldsymbol{r}^{(j)} \rangle \right).$$
(5.23)

This procedure is repeated in a cyclical manner, by applying Equation (5.23) and thereby updating the coordinates of $\hat{\beta}_i^L$ along the way.

Depending on the λ value the lasso includes different predictors, because it determines the size of the shrinkage penalty. Hence, to achieve the best possible estimate, the optimal λ value is found by cross-validation, see Section 3.2. To use cross-validation we must computed the lasso solution not only with one constant λ value, but with a range of possible λ values. To do this, we begin by computing the lasso solution with $\lambda_{\max} = \max_j |\frac{1}{n} \langle \boldsymbol{x}_j, \boldsymbol{y} \rangle|$, which is just large enough to make the all-zero vector the only possible solution. We then decrease λ by a small amount, and run coordinate descent until convergence. This procedure is repeated using the previous solution as a "warm start" for the new λ value.

Cycle coordinate descent is an especially fast algorithm for the lasso because the iterative search, which the algorithm normally conducts along each coordinate, is not conducted in the lasso case since the coordinate minimizers, Equation (5.23), are explicitly available. Furthermore, the algorithm uses the sparsity of the lasso where, for large enough λ values, many of the coefficients are zero and will thus not be moved from zero.

The lasso method does both parameter shrinkage and variable selection due to its shrinkage penalty which zero out the coefficients of collinear predictors. This can be a drawback when working with many predictors, because many predictors can be divided into a smaller number of themed groups, and the predictors within a specific group are typically correlated. Hence, the lasso tends to only select a few predictors within a group, which means that it may discard some meaningful predictors because of correlation within the group. As we are trying to forecast GDP, the unemployment rate and the inflation rate, we may be interested in including a whole group of coefficients in our model, e.g. to include all the predictor in the price, wages, and inflation group in the inflation rate model, therefore, we now introduce the grouped lasso.

5.3 Grouped Lasso

This section is further based on [Yuan and Lin, 2007] and [Hastie et al., 2010].

The grouped lasso, introduced by [Bakin, 1999], is an extension to the lasso method, which allows predefined groups of predictors to be shrunken or selected out of the model together. Hence, all the predictors in a group are either included or not included in the model.

Assume that the *p* predictors are divided into *H* groups, where p_h denotes the number of predictors in group *h*. Let X_h be the predictors corresponding to the *h*'th group, and β_h the corresponding coefficient vector. Then the grouped lasso, which is a natural generalization of the standard lasso objective, solves the convex optimization problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}} \left\{ || \boldsymbol{y} - \sum_{h=1}^{H} \boldsymbol{X}_h \boldsymbol{\beta}_h ||_2^2 + \lambda \sum_{h=1}^{H} \sqrt{p_h} || \boldsymbol{\beta}_h ||_2 \right\},$$
(5.24)

where $\sqrt{p_h}$ accounts for the varying group sizes, such that large groups are not penalized harder because of their size, and $|| \cdot ||_2$ denote the Euclidean norm.

The penalty in Equation (5.24) reduces to an Euclidean norm of a vector which only has zero value if all of its components are zero. Hence, it cannot select only some of the predictors from a group, similar to ridge regression which also has an Euclidean penalty, and who also includes all or non of the predictors. However, it still acts like the lasso, since it encourages sparsity at a group level, because the penalty is the sum over the different subspace norms and the constraint thus have some non-differentiable points. Thus the penalty function for the grouped lasso can be seen as the intermediate between the ℓ_1 and ℓ_2 penalty, illustrated in Figure 5.2.



Figure 5.2: Illustration of the penalty functions for the ridge regression, the lasso, and the grouped lasso. (a) illustrates the ℓ_1 penalty use in the lasso method, (c) illustrates the ℓ_2 penalty used in the ridge regression method, and finally (b) illustrates the penalty function for the grouped lasso, which is an intermediate between the ℓ_1 and ℓ_2 penalty. Source:[Yuan and Lin, 2007]

The grouped lasso solution can be computed using the same algorithm, cyclic coordinate de-

scent, as for the lasso, see Section 5.2.1.

The grouped lasso introduces sparsity at a group level, which solves the problem of excluding significant predictors in a group because of collinearity within the group. However, if a group containing a large number of predictors is included, it may be rewarding to introduce sparsity within the group, such that not all of the predictors in the group are included. A method which encounter this problem is the elastic net method.

5.4 Elastic Net

This section is based on [Friedman et al., 2010].

The elastic net method, introduced by [Zou and Hastie, 2005], blends the ℓ_1 norm, from the lasso, with the ℓ_2 norm, from the ridge regression (or group lasso). Hence, the elastic net method yields solutions that are sparse at both the group and individual feature level.

The elastic net solves the following problem

$$\min_{\beta \in \mathbb{R}} \left\{ \frac{1}{2n} \left(\sum_{i=1}^n y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \left[\frac{1}{2} (1-\alpha) \beta_j^2 + \alpha |\beta_j| \right] \right\},\$$

where α provides a compromise between the ℓ_1 norm, $\alpha = 1$ and ℓ_2 norm, $\alpha = 0$, see Figure 5.3. It thus have the effect of averaging predictors that are highly correlated and then entering the averaged predictor into the model.



Figure 5.3: Illustrations of the penalty function for the ridge regression (dotted), lasso (dashed) and elastic net penalty (solid), for a single group with two predictors. Source:[Hastie et al., 2010].

As for the lasso and grouped lasso, the solution to the elastic net can be found using the cyclic coordinate descent algorithm, see Section 5.2.1.

Because the elastic net method yields solutions that are sparse at both the group and individual feature level, it includes less predictors than the ridge regression. However, it includes more predictors than the lasso because it yield sparsity at the group level, and less predictors than the group lasso because it yields sparsity at an individual level within the group.

CHAPTER 6

VECTOR AUTOREGRESSIVE MODELS

So far, we have dealt with methods that are designed to handle a large number of predictors. These methods are usually combined with a linear regressions in which the relationship between the dependent variable and the estimated factors, dynamic factor models, or the constrained predictors, shrinkage methods, are modeled. In this chapter we use these method in the context of the vector autoregressive (VAR) models, in order to compute models that not only regress the dependent variable at the predictors, but also includes the variable's own lagged values.

VAR models, first introduced as a method for estimating economic relationships by [Sims, 1980], is widely used for structural analysis and simultaneous forecasting of a number of temporally observed variable. In a VAR model each variable has an equation explaining its evolution based on its own lags and the lags of the other model variables. Hence, if we measure three different time series, $y_{t,1}$, $y_{t,2}$, $y_{t,3}$, and want to explain their evolution over one period, we have the following VAR model

 $\begin{aligned} y_{t,1} &= v + \phi_{11}y_{t-1,1} + \phi_{12}y_{t-1,2} + \phi_{13}y_{t-1,3} + \varepsilon_{t,1} \\ y_{t,2} &= v + \phi_{21}y_{t-1,1} + \phi_{22}y_{t-1,2} + \phi_{23}y_{t-1,3} + \varepsilon_{t,2} \\ y_{t,3} &= v + \phi_{31}y_{t-1,1} + \phi_{32}y_{t-1,2} + \phi_{33}y_{t-1,3} + \varepsilon_{t,3}. \end{aligned}$

The conventional VAR models have quadratically growing parameter spaces, e.g. in our case we have 80 time series of 31 observations, thus we want to compute a VAR(30) model for the 80th time series, which requires estimating 900 parameters. However, such large number of stationary observations are not available in practice which means that the conventional VAR model suffers from a dimensionality problem. To overcome this problem, [Litterman, 1979] introduced the Bayesian VAR approach, in which the problem with dimensionality is solved by shrinking the variables by imposing priors. Another method that overcomes the dimensionality problem is the factor-augmented VAR model, proposed by [Bernanke et al., 2005], where the variable that we would like to forecast together with estimated factors of the predictors, see Section 4.1.1, are arguments in a conventional VAR model. More recent research reduces the parameter space of a VAR model by incorporating the Lasso, see Section 5.2. This method is called the Lasso-VAR model and was first introduced by [Hsu et al., 2008].

Since we have already discussed the use of factor estimation and shrinkage methods in forecasting with many predictors, we incorporate factor VAR and lasso-VAR in our empirical work.

6.1 Factor-Augmented VAR Model

This section is based on [Bernanke et al., 2005].

The factor-augmented VAR (FAVAR) context allows us to include much of the information stored in the 80 macroeconomic variables in the model, by including the dependent variable and the estimated factors, which describes much of the variance in the remaining 79 predictors. Let y_t be a $k \times 1$ vector of observable variables, and F_t be a $r \times 1$ vector of unobservable factors. Assume that the joint dynamics of (F_t, Y_t) are given by

$$\begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{y}_t \end{bmatrix} = \Phi(L) \begin{bmatrix} \boldsymbol{F}_{t-1} \\ \boldsymbol{y}_{t-1} \end{bmatrix} + \boldsymbol{v}_t, \qquad (6.1)$$

where $\Phi(L)$ is a lag polynomial of finite order p, and v_t is the error term with zero mean. Equation (6.1) is the FAVAR model, in which the factors can be estimated by PCA as described in Section 4.1.1. Note, that the dynamic factor models described in Chapter 4 and the FAVAR models are estimates in the same way. The dynamic factor models are preferred when the corr purpose is forecasting, whereas the FAVAR models are preferred when conducting a structural analysis of the variables. The dynamic factor model is used when forecasting because it only produces the model for the desired forecast variable. The FAVAR model is used to conducting a structural analysis, because it produces a grid of models, one for each incorporated variable.

6.2 Lasso-VAR Model

This section is based on Nicholson et al. [2014].

The lasso-VAR model applies the shrinkage penalty described in Section 5.2 to the conventional VAR model, which in vector notation is given by

$$Y = v1' + BZ + U$$

where $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T)$ is a $k \times T$ response matrix, $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_p)$ is a $k \times kp$ coefficient matrix, $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_T)$ is a $kp \times T$ covariate matrix, in which $\mathbf{z}_t = (\mathbf{y}'_{t-1}, \mathbf{y}'_{t-2}, \dots, \mathbf{y}'_{t-p})'$ is a $kp \times 1$ vector, $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_T)$ is a $k \times T$ error matrix, $\mathbf{v} = (v_1, v, \dots, v_k)$ is a $k \times 1$

intercept vector, and **1** is a $T \times 1$ vector of ones. Notice, that p is the maximum lag length, and k is the number of variables included.

The idea behind the introduction of lasso in the context of VAR models is, to apply the ℓ_1 penalty to the convex least squares objective function

$$\frac{1}{2}||\boldsymbol{Y} - \boldsymbol{v}\boldsymbol{1}' - \boldsymbol{B}\boldsymbol{Z}||_F^2 + \lambda||\boldsymbol{B}||_1, \qquad (6.2)$$

in which $|| \cdot ||_F$ denotes the Frobenius norm⁷, $|| \cdot ||_1$ denotes the ℓ_1 norm, and λ is the tuning parameter.

As discussed in Section 5.2, the lasso introduce sparsity in the coefficients because it sets some of them to zero. Hence, in the lasso-VAR model this is equivalent to sparsity in the coefficient matrix \boldsymbol{B} , illustrated in Figure 6.1.



Figure 6.1: The sparsity pattern for a lasso-VAR model with p = 5 and k = 3. The blue squares are the coefficients that are included in the models, and the white are the coefficients that are set to zero. Source:[Nicholson et al., 2014]

All of the conditions and assumptions, such as the standardization of the variables, that apply in Section 5.2 does also apply here. Furthermore, Equation (6.2) are solved by using the coordinate decent algorithm, described in Section 5.2.1. However, in contrast to the discussion about λ in Section 5.2, the optimal tuning parameter is determined by the BIC, see Section 3.3.2, and not cross-validation. When applying the lasso-VAR model, the conventional VAR model becomes linear regression models, in which the current values of the variables are treated as the dependent variables and the lagged values are treated as the predictors. Hence, the only difference between the lasso-VAR model and the model described in Section 5.2 is, that the lasso-VAR model uses the lagged values of the dependent variable in the regression.

The lasso-VAR model performs both model selection, discarding some coefficients by setting them equal to zero, and parameter estimation, shrinking the remaining coefficients in the model, simultaneously. Hence, it has an advantage in comparison to both the Bayesian VAR and the FAVAR.

⁷The Frobenius norm is defined as $\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2$.

CHAPTER 7

BAYESIAN MODEL AVERAGING

This chapter is based on [Arrow and Intriligator, 2006], [Fernandez et al., 2001], and [Koop and Potter, 2004]

A fourth alternative to handle many predictors in macroeconomic forecasting is the Bayesian model averaging, which was first introduced by [Leamer, 1978]. In contrast to the models described so far, Bayesian model averaging account for model uncertainty in linear regression models, by averaging over all possible combinations of the predictors. Hence, the main idea of the Bayesian model averaging is to forecast combining, thus by applying Bayesian model averaging, we compute the overall forecast as a weighted average of multiple individual model forecasts.

We choose to work with the Bayesian model averaging in the context of linear regressions. In this setting, consider K different models, M_1, \ldots, M_K , for a dependent variable with n observations, then a model M_j , $j = 1, \ldots, K$ consists of a choice of $0 \le k_j \le K$ potential predictors. This leads to the following j'th linear regression

$$\boldsymbol{y} = \boldsymbol{X}_j \boldsymbol{\beta}_j + \sigma \boldsymbol{\varepsilon}, \tag{7.1}$$

where \boldsymbol{y} is the $n \times 1$ vector of the dependent variable, \boldsymbol{X}_j is the $n \times k_j$ submatrix of relevant predictors, $\boldsymbol{\beta}_j$ is the vector of regression coefficients, σ is a scale parameter, and $\boldsymbol{\varepsilon}$ is a $n \times 1$ vector error which follows a Normal distribution, with zero mean and identity covariance matrix. Notice, that the data has been centered, such that the model can be formulated without the intercept.

The Bayesian model averaging follows the theory of Bayesian econometrics, in which the parameters in Equation (7.1) belongs to a specific prior distribution which is given by a density function

$$P(\boldsymbol{\beta}_j, \sigma | M_j), \tag{7.2}$$

and a Dirac distribution at zero, which contains the irrelevant coefficients in β_j , $\beta_j \in \mathbb{R}^{k-k_j}$,

$$P_{\boldsymbol{\beta} | \boldsymbol{\beta}_j, M_j, \sigma} = \text{Dirac at } (0, \dots, 0).$$

The Bayesian model averaging deals with model uncertainty, where each model candidate is treated as random and has a prior probability of being the correct dependent variable generating model. When dealing with model uncertainty, we put a prior distribution over the model space $\mathcal{M} = \{M_j : j = 1, \dots, K\},\$

$$P(M_j) \quad j = 1, \dots, K$$

with $P_j > 0$ and $\sum_{j=1}^{K} P_j = 1$. If we denote the posterior probability, that M_j is the correct model, as $P(M_j | \boldsymbol{y}, \boldsymbol{X}_j)$, the law of total expectation⁸ implies that

$$E(y_{T+1}|\boldsymbol{y},\boldsymbol{X}_j) = \sum_{j=1}^{K} E(y_{T+1}|\boldsymbol{y},\boldsymbol{X}_j,M_j) P(M_j|\boldsymbol{y},\boldsymbol{X}_j).$$
(7.3)

When the number of interesting models M becomes large, Bayes' rule stated that the posterior is proportional to the prior times the likelihood, thus

$$P(M_j|\boldsymbol{y}, \boldsymbol{X}_j) \propto P(\boldsymbol{y}, \boldsymbol{X}_j|M_j)P(M_j)$$

and the posterior probability of model j is given as

$$P(M_j | \boldsymbol{y}, \boldsymbol{X}_j) = \frac{P(\boldsymbol{y} | \boldsymbol{X}_j, M_j) P(M_j)}{\sum\limits_{h=1}^{K} P(\boldsymbol{y} | \boldsymbol{X}_j, M_h) P(M_h)}.$$

Here $P(\boldsymbol{y}|\boldsymbol{X}_j, M_j)$ is the marginal likelihood under model M_j given as

$$P(\boldsymbol{y}|\boldsymbol{X}_j, M_j) = \int P(\boldsymbol{y}|\boldsymbol{\beta}_j, M_j, \sigma) P(\boldsymbol{\beta}_j, \sigma|M_j) d\boldsymbol{\beta}_j d\sigma,$$

where $P(\boldsymbol{y}|\boldsymbol{\beta}_{i}, M_{j}, \sigma)$ and $P(\boldsymbol{\beta}_{i}, \sigma|M_{j})$ is defined through Equation (7.1) and (7.2), respectively.

7.1 Choosing the Priors in Bayesian Model Averaging

Two important issues arise when implementing Bayesian model averaging. First, the two set of priors, the prior distribution of the parameters given the model and the prior probability of the mode, for the parameters must be found. Because we are handling a large number of predictors, we follow the work of [Fernandez et al., 2001] which propose a benchmark set of conjugate priors for Bayesian model averaging in the linear model context with, a large number of predictors. The prior for the scale parameter σ , is the noninformative prior $P(\sigma) \propto \sigma^{-1}$, and the prior for the covariance vector β_j is the g-prior, originally introduced by [Zellner, 1986],

$$\boldsymbol{\beta}_{j} | \sigma \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^{2} \left(g_{j} \boldsymbol{X}_{j}^{\prime} \boldsymbol{X}_{j} \right)^{-1}
ight).$$

 $[\]overline{{}^{8}$ if M_1, M_2, \ldots, M_K is a partition of the whole outcome space, i.e. these events are mutually exclusive and exhaustive, then $E(X) = \sum_{j=1}^{K} E(X \mid M_j) P(M_j)$. Soruce: [Wikipedia, 2016]

The g-prior is an widely used benchmark prior for the regression coefficients of a multiple regression. It only requires us to compute one hyperparameter, g_j , which we do following [Fernandez et al., 2001] who recommends,

$$g = \frac{1}{\min(T, k^2)}$$

The prior model probability, also needs to be specified. Hence, we follow [Koop and Potter, 2004] which uses multinomial distribution priors, where the probability is determined by the prior probability that an individual variable enters the model.

7.2 Posterior Computation

The second issue in the implementation of Bayesian model averaging, is the huge number of models for which $P(M_j|\boldsymbol{y}, \boldsymbol{X}_j)$ and $E(y_{T+1}|\boldsymbol{y}, \boldsymbol{X}_j)$ is evaluated. In our application we define models based on the inclusion/exclusion of each predictor. Hence, we have $j = 2^K$ models which makes it computationally impossible to evaluate $P(M_j|\boldsymbol{y}, \boldsymbol{X}_j)$ and $E(y_{T+1}|\boldsymbol{y}, \boldsymbol{X}_j)$ for each model. To overcome this computation problem numerous algorithms, which forces Bayesian model averaging not to evaluate every model, has been developed. These algorithms builds on the fact that a large fraction of the model most likely have posterior model probabilities close to zero. As a consequence of these posterior model probabilities, it is possible to obtain a close approximation of $E(y_{T+1}|\boldsymbol{y}, \boldsymbol{X}_j)$ without evaluating the entire set of model candidates. In our implementation, we use the Markov chain Monte Carlo model composition algorithm.

CHAPTER 8

EMPIRICAL ANALYSIS

In this chapter we estimate models for the GDP, the unemployment rate, and the inflation rate, shown in Figure 8.1, from available macroeconomic data. Furthermore, we use the constructed models to forecast the three macroeconomic variables 2.5 years ahead.



Figure 8.1: The three macroeconomic variables, constructed on quarterly measured U.S. time series starting 2001-12-31 to 2015-12-31.

The data used in this analysis is retracted from Quandl.com, and contains 80 quarterly measured macroeconomic U.S. time series starting 2001-12-31 to 2015-12-31.

The data represents a wide range of macroeconomic time series which we roughly divide into the following 10 categories:

- Industrial production growth: Contains 12 series, 13 when GDP is included.
- Real personal income and consumption growth: Contains 5 series.

- Employment and unemployment growth: Contains 16 series, 17 when the unemployment rate is included.
- Housing starts growth: Contains 5 series.
- Manufacturing and retail sale: Contains 2 series.
- Real inventories and inventory-sales ratio: Contains 7 series.
- Price, wages, and inflation: Contains 11 series, 12 when the inflation rate is included.
- Money and credit quantity aggregates: Contains 6 series.
- Interest rates: Contains 7 series.
- Exchange rates and stock prices: Contains 6 series.

These categories and the 80 time series are chosen based on the theory regarding the three macroeconomic variables, described in Chapter 2, and the work in [Economics, 2016] and [Watson, 2000]. The raw data are transformed to eliminate trends and obvious nonstationarities. First, many of the time series are seasonally adjusted, and second, real variables are transformed to growth rates, prices are transformed to changes in growth rates, and interest rates are transformed to spreads. A detailed enumeration of the data and its modifications is available in Appendix A.

When we construct the models we use 79 predictors and 1 dependent variable, and in order to both construct and validate the models as well as forecast, we split the full dataset into three separate subsets:

- Training set, containing 31 quarters randomly drawn from the first 46 quarters. This subset is used to train the models.
- Validation set, containing the 15 quarters from the first 46 quarter which was not included in the training set. This subset is used to validate the model performance in order to conduct a model selection.
- Forecasting set, containing the last 10 quarters of the full dataset. This subset is used to evaluate the performance of the forecasts.

The fragmentation of the training and validation set follows the validation set approach, see Section 3.2.1, thus these subsets are randomly split.

8.1 Dynamic Factor Models

In this section we present the results for the dynamic factor model with principal component analysis, described in Chapter 4. All the models are estimated on the training set and their performance are evaluated on the validation set. Furthermore, the forecasts are evaluated on the forecast set.

The goodness-of-fit and prediction error for the models describing the three macroeconomic variables are listed in Table 8.1.

	No. Factors	\mathbf{R}^2	RMSE	MSPE
GDP	14	0.945215	0.002042	4.909782e-05
Unemployment Rate	12	0.959829	0.000342	1.006426e-06
Inflation Rate	21	1.00	0.000369	2.333727e-05

Table 8.1: The goodness-of-fit and validation of the models. R^2 is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The R^2 and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green numbers represents the best measures.

Notice, that in the three dynamic factor models with principal components the factors explains close to 100 % of the variation in the three macroeconomic variables.

The mean squared prediction errors indicates that the dynamic factor model captures most of the trends in the three variables. The relationship between the predictions and the true values of the variables are illustrated in Figure 8.2, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.2: A comparison between the predicted- and real values for the three macroeconomic variables. The predicted values are based on the dynamic factor model with principal components.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
GDP	0.2468
Unemployment Rate	0.1382
Inflation Rate	0.3136

Table 8.2: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.2 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.3 the histograms of the residuals are plotted together with their normal distribution.



(c) Inflation Rate.

Figure 8.3: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values show evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.3. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.4, which, apart from a few outliers, shows a random pattern dispersed around the horizontal axis. This means that the residuals are unbiased and homoscedastic, showing no clear linear relationship.



(c) Inflation Rate.

Figure 8.4: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

So far, we have shown that the computed dynamic factor models with principal component analysis describes the three macroeconomic variables reasonable, and that they approximately fulfills the model requirements. This could indicate that these models are reasonable to use for forecasting, thus they are used in the computation of a 2.5 year forecast, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.3.

	RMSFR	MAFE
GDP	0.006867	0.005692
Unemployment Rate	0.002061	0.001928
Inflation Rate	0.004587	0.003259

Table 8.3: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted GDP, unemployment rate, and inflation rate. The green numbers represents the lowest error.

The analysis of these forecast errors will be conducted in Section 8.5. However, Figure 8.5, which illustrates the forecast relative to the true value of the variables, gives an indication of the forecast accuracy.



Figure 8.5: A comparison between the forecasted and the real values for the three macroeconomic variables. The forecasted values are based on the dynamic factor model with principal components.

8.2 Shrinkage Methods

In this section we present the results for the constructed linear regressions based on the shrinkage methods, described in Chapter 5. The section is further divided into subsections, where each subsection presents the work for one of the three macroeconomic variables described in Chapter 2. All the models are estimated on the training set and their performance are evaluated on the validation set. Furthermore, the forecasts are evaluated on the forecast set.

8.2.1 Gross Domestic Product

In this section we estimate linear regressions for predicting the GDP, where the GDP is the dependent variable and the remaining 79 macroeconomic time series in the training set are predictors.

The shrinkage methods are all applied for a range of different λ values, and we use 5-fold crossvalidation, see Section 3.2, to estimate the optimal λ value and thereby estimate the best possible model. The cross-validation measures the mean squared error of the model for each λ value.



Hence, the model with the smallest mean squared error is used. The cross validation path and its corresponding λ values together with the number of predictors are shown in Figure 8.6.

(c) Elastic net.

Figure 8.6: The cross validation path, which shows the relationship between the mean squared error and the λ values suggested to the model. The y-axis represents the mean squared errors of the model, and the x-axis represents the log(λ) values tested in the model. Furthermore, the top-line shows the number of predictors included in the model.

The optimal λ value for the four methods, given in Table 8.4, shows that the grouped lasso has the largest tuning parameter and thereby the largest shrinkage penalty. Notice, that the ridge regression model has an optimal lambda value equal to zero, which means that the ridge regression model, in this case, is reduced to the ordinary least squares model.

	Optimal Lambda	Number of Predictors
Ridge Regression	0	79
Lasso	7.989409e-05	9
Grouped Lasso	0.000524	5 groups, 26 individual
Elastic Net	0.000405	16

Table 8.4: The optimal value for the tuning parameter λ , and the number of predictors included in the optimal model.

The number of predictors included in the models differ a lot. Obviously, the ridge regression

includes all 79 predictors as it only shrinks its coefficient towards zero, but never reaches it. Furthermore, in this particular case, it does not impose any shrinkage at all. The predictors included by the lasso, grouped lasso and elastic net are shown in Table 8.5.

Lasso

Industrial Production: Final Product Personal Consumption Expenditures: Nondurable Goods Personal Consumption Expenditures: Services Housing Starts in Midwest Census Region M2 Money Stock Monetary Base; Total 6-Month Treasury Bill 1-Year Treasury Constant Maturity Rate S&P 500 Real Sales Growth by Quarter

Grouped Lasso

Real Personal Income and consumption Growth Manufacturing and retail sale Money and credit quantity aggregates Interest Rates Exchange rates and stock prices

Elastic Net

Industrial Production: Final Product All Employees: Total nonfarm All Employees: Government: State Government Personal Consumption Expenditures: Durable Goods Personal Consumption Expenditures: Nondurable Goods Personal Consumption Expenditures: Services Housing Starts: Total Housing Starts in Northeast Census Region Housing Starts in Midwest Census Region M2 Money Stock Monetary Base; Total 6-Month Treasury Bill 1-Year Treasury Constant Maturity Rate 5-Year Treasury Constant Maturity Rate TED Spread S&P 500 Real Sales Growth by Quarter

Table 8.5: The predictors included by the lasso, grouped lasso and elastic net.

Notice, that the lasso, grouped lasso and elastic net methods all includes predictors from the same groups. If we compare the predictors included by the methods, with the components theoretically connected with GDP, see Figure 2.1, it is worth mentioning, that the methods includes some of the theoretical components. However, around half of the predictors included does not have any direct theoretical meaning to the GDP. This could indicate that the models, based on

these three methods, lag information about the components in the GDP.

The goodness-of-fit and prediction error for the models estimated by the four methods are listed in Table 8.6.

	R^2	RMSE	MSPE
Ridge Regression	1.0	1.041577e-17	8.990637e-05
Lasso	0.6414284	0.002681	4.415568e-05
Grouped Lasso	0.652029	0.002563	3.99307 e-05
Elastic Net	0.7668502	0.001988	4.981591e-05

Table 8.6: The goodness-of-fit and validation of the optimal models. R² is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The R² and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green and red numbers represents the best and worst measures, respectively.

The \mathbb{R}^2 states that, for the model build on ridge regression, the predictors explains 100 % of the variation in GDP, whereas for the lasso, only 64.14 % is explained. The model estimated by ridge regression has the lowest root mean squared error, but the highest prediction error. Hence, this indicates that the model overfits the training data, which corresponds to the theoretical knowledge about the ordinary least squares model. The ordinary least squares model tends to have high variance and low bias, when the number of predictors exceed the number of observations.

The relationship between the predictions and the true values of GDP are illustrated in Figure 8.7, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.7: A comparison between the predicted- and real values for the GDP. The predicted values are based on the models estimated by the shrinkage methods.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
Ridge Regression	0.192
Lasso	0.4083
Grouped Lasso	0.8156
Elastic Net	0.2057

Table 8.7: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.7 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.8 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.8: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.8. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.4. The ridge regression residuals shows a random pattern dispersed around the horizontal axis, which means that the residuals are unbiased and homoscedastic, showing no clear linear relationship. The three other residual plots shows a linear relationship. The grouped lasso residuals seams to be biased, since the mean value is clearly not zero, and homoscedastic, since all the predictors have approximately the same finite variance. The lasso and the elastic net residuals seams to be both biased and heteroscedastic.



Figure 8.9: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

Based on the above observations and measurements, we are now able to do model selection. The selected model is based on the grouped lasso, because it exhibits the lowest prediction error. Normally, we only forecast on the selected model, because of its ability to describe the trend in the variable, up until today. Because we have removed the last 2.5 year from the data set, we are able to compute the forecast error. Hence, it could be interesting to compare the selected model forecast with the forecast for the remaining models, to see if the selected model actually produces the most accurate forecast.

The 2.5 year forecast of GDP, starting 2013-06-30 to 2015-12-31, are listed in Table 8.8.

	RMSFE	MAFE
Ride Regression	0.006347	0.005336
Lasso	0.005633	0.005015
Grouped Lasso	0.005831	0.005121
Elastic Net	0.005941	0.005173

Table 8.8: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted GDP, using models constructed by the ridge regression, lasso, grouped lasso and elastic net methods. The green and red numbers represents the highest and lowest errors, respectively.

Notice, that the selected model does not produce the most accurate forecast, that belongs to the model estimated by the lasso. However, the deviation between the two forecast errors are only 5 %. The forecasts are illustrated in Figure 8.10.



Figure 8.10: A comparison between the forecasted and the real values for the GDP. The forecasted values are based on the models estimated by the shrinkage methods.

8.2.2 Unemployment Rate

In this section we estimate linear regressions for predicting the unemployment rate, where the unemployment rate is the dependent variable and the remaining 79 macroeconomic time series
in the training set are predictors.

The optimal lambda value for the four methods, given in Table 8.9, shows that the elastic net method has the largest tuning parameter and thereby the largest shrinkage penalty. Notice, that the ridge regression model has an optimal lambda value equal to zero, which means that the ridge regression model, in this case, is reduced to the ordinary least squares model.

	Optimal Lambda	Number of Predictors
Ridge Regression	0	79
Lasso	5.822121e-05	12
Grouped Lasso	3.538197e-05	9 groups, 66 individual
Elastic Net	6.360407 e-05	16

Table 8.9: The optimal value for the tuning parameter λ , and the number of predictors included in the optimal model.

The cross validation path and its corresponding λ values together with the number of predictors are shown in Figure 8.11.



(c) Elastic net.

Figure 8.11: The cross validation path, which shows the relationship between the mean squared error and the λ values suggested to the model. The y-axis represents the mean squared errors of the model, and the x-axis represents the $\log(\lambda)$ values tested in the model. Furthermore, the top-line shows the number of predictors included in the model.

The predictors included in the lasso, grouped lasso and elastic net are shown in Table 8.10.

Lasso

Industrial Production: Durable Materials Average (Mean) Duration of Unemployment Housing Starts in Northeast Census Region Housing Starts in South Census Region Value of Manufacturers' New Orders for All Manufacturing Industries Value of Manufacturers' New Orders: Consumer Durable Goods Industries With Unfilled Orders Consumer Price Index for All Urban Consumers Consumer Price Index for All Urban Consumers Commercial and Industrial Loans Effective Federal Funds Rate Canada / U.S. Foreign Exchange Rate

Grouped Lasso

Employment and Unemployment Growth Real Personal Income and consumption Growth Housing Starts Growth Manufacturing and retail sale Real inventories and inventory-sales ratios Prices, Wages and Inflation Money and credit quantity aggregates Interest Rates Exchange rates and stock prices

Elastic Net

Industrial Production: Durable Materials Average (Mean) Duration of Unemployment All Employees: Government: Local Government Housing Starts: Total Housing Starts in Northeast Census Region Housing Starts in South Census Region Housing Starts in West Census Region Value of Manufacturers' New Orders for All Manufacturing Industries Value of Manufacturers' New Orders: Consumer Durable Goods Industries With Unfilled Orders Consumer Price Index for All Urban Consumers: Durables Consumer Price Index for All Urban Consumers: Services Commercial and Industrial Loans Effective Federal Funds Rate 3-Month Treasury Bill: Secondary Market Rate 6-Month Treasury Bill: Secondary Market Rate Canada / U.S. Foreign Exchange Rate

Table 8.10: The predictors included by the lasso, grouped lasso and elastic net.

Notice, that the lasso, grouped lasso and elastic net methods all includes a wide range of predictors from all groups, especially the grouped lasso includes all the predictors except the 13 included in the Industrial Production Growth group.

This could be a prior indication that the models, based on these three methods, are able to capture trends included in the unemployment rate.

The goodness-of-fit and prediction error for the models estimated by the four methods are listed in Table 8.11.

	\mathbf{R}^2	RMSE	MSPE
Ridge Regression	1,00	3.722524e-18	8.819696e-07
Lasso	0.852211	0.000248	7.090029e-07
Grouped Lasso	0.870448	0.000168	7.490656e-06
Elastic Net	0.835618	0.000254	6.902786e-07

Table 8.11: The goodness-of-fit and validation of the optimal models. R² is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The R² and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green and red numbers represents the best and worst measures, respectively.

The R^2 states that, for the model build on ridge regression, the predictors explains 100 % of the variation in the unemployment rate, whereas for the elastic net, only 83.6 % is explained. The model estimated by the elastic net method has the highest root mean squared error, but the lowest prediction error. Hence, this indicates that the model underfits the training data, do to high bias and low variance.

The relationship between the predictions and the true values of the unemployment rate are illustrated in Figure 8.12, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.12: A comparison between the predicted- and real values for the unemployment rate. The predicted values are based on the models estimated by the shrinkage methods.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
Ridge Regression	0.5932
Lasso	0.9565
Grouped Lasso	0.2979
Elastic Net	0.9121

Table 8.12: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.12 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.13 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.13: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.13. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.14. The ridge regression, lasso, and elastic net residuals shows a random pattern dispersed around the horizontal axis, which means that the residuals are unbiased and homoscedastic, showing no clear linear relationship. The grouped lasso residuals shows a linear relationship which indicates that they are biased and heteroscedastic.



Figure 8.14: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

Based on the above observations and measurements, we are now able to do model selection. The selected model is based on the elastic net method, because it exhibits the lowest prediction error.

To find out how accurate the four models forecast, and to see if the selected model produces the best forecast, we compute a 2.5 year forecast of the unemployment rate, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.13.

	RMSFE	MAFE
Ride Regression	0.001540	0.001385
Lasso	0.002764	0.002563
Grouped Lasso	0.002628	0.002398
Elastic Net	0.002711	0.002529

Table 8.13: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted unemployment rate, using models constructed by the ridge regression, lasso, grouped lasso and elastic net methods. The green and red numbers represents the highest and lowest errors, respectively.

Notice, that the selected model does not produce the most accurate forecast, that belongs to the

model estimated by the ridge regression. The deviation between the two forecast errors are 43 %, which means that the model estimated on ridge regression produces a much better forecast. The forecasts are illustrated in Figure 8.15.



Figure 8.15: A comparison between the forecasted and the real values for the unemployment rate. The forecasted values are based on the models estimated by the shrinkage methods.

8.2.3 Inflation Rate

In this section we estimate linear regressions for predicting the inflation rate, where the inflation rate is the dependent variable and the remaining 79 macroeconomic time series in the training set are predictors.

The optimal lambda value for the four methods, given in Table 8.14, shows that the elastic net method has the largest tuning parameter and thereby the largest shrinkage penalty. Notice, that the ridge regression model has an optimal lambda value equal to zero, which means that the ridge regression model, in this case, is reduced to the ordinary least squares model.

	Optimal Lambda	Number of Predictors
Ridge Regression	0	79
Lasso	0.000136	7
Grouped Lasso	0.000186	5 groups, 43 individual
Elastic Net	0.000215	14

Table 8.14: The optimal value for the tuning parameter λ , and the number of predictors included in the optimal model.

The cross validation path and its corresponding λ values together with the number of predictors are shown in Figure 8.16.



Figure 8.16: The cross validation path, which shows the relationship between the mean squared error and the λ values suggested to the model. The y-axis represents the mean squared errors of the model, and the x-axis represents the $\log(\lambda)$ values tested in the model. Furthermore, the top-line shows the number of predictors included in the model.

The predictors included in the lasso, grouped lasso and elastic net are shown in Table 8.15.

Lasso

Industrial Production: nondurable Materials Personal Consumption Expenditures: Nondurable Goods Merchant Wholesalers Inventories Producer Price Index: Finished Goods Consumer Price Index for All Urban Consumers: Services Consumer Price Index for All Urban Consumers: All Items Less Food TED Spread

Grouped Lasso

Industrial Production Growth Real Personal Income and consumption Growth Real inventories and inventory-sales ratios Prices, Wages and Inflation Interest Rates

Elastic Net

Industrial Production: Mining Industrial Production: Materials Industrial Production: nondurable Materials Personal Consumption Expenditures: Nondurable Goods Producer Price Index: All Commodities Producer Price Index: Finished Goods Producer Price Index: Finished Consumer Goods Producer Price Index: Finished Consumer Goods Producer Price Index: Intermediate Materials: Supplies & Components Consumer Price Index for All Urban Consumers: Apparel Consumer Price Index for All Urban Consumers: Services Consumer Price Index for All Urban Consumers: Services Consumer Price Index for All Urban Consumers: Services Consumer Price Index for All Urban Consumers: All Items Less Food Commercial and Industrial Loans, All Commercial Banks 10-Year Treasury Constant Maturity Rate

Table 8.15: The predictors included by the lasso, grouped lasso and elastic net.

Notice, that the lasso, grouped lasso and elastic net methods all includes a wide range of predictors from all groups, especially the elastic net method includes many of the predictors in the prices, wages, and inflation group. Hence, the model includes many predictors which corresponds to the theoretically components of inflation, se Section 2.3.

This could be a prior indication that the models, based on these three methods, are able to capture trends included in the inflation rate.

The goodness-of-fit and prediction error for the models estimated by the four methods are listed in Table 8.16.

	\mathbb{R}^2	RMSE	MSPE
Ridge Regression	1.00	8.00964e-18	2.976736e-05
Lasso	1.00	0.000574	2.324016e-05
Grouped Lasso	1.00	0.000858	3.464863e-05
Elastic Net	1.00	0.000642	1.693075e-05

Table 8.16: The goodness-of-fit and validation of the optimal models. R² is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The R² and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green and red numbers represents the best and worst measures, respectively.

The \mathbb{R}^2 states that, in all the models, the predictors explains 100 % of the variation in the unemployment rate. The model estimated by elastic net has the lowest prediction error, thus this model should be selected for further use.

The relationship between the predictions and the true values of the unemployment rate are illustrated in Figure 8.17, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.17: A comparison between the predicted- and real values for the inflation rate. The predicted values are based on the models estimated by the shrinkage methods.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
Ridge Regression	0.1382
Lasso	0.00177
Grouped Lasso	0.4246
Elastic Net	0.0781

Table 8.17: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.17 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis for the ridge regression, grouped lasso, and elastic net residuals. The null-hypothesis are rejected for the lasso, because it has a p-value < 0.05, thus the residual are not normally distributed. In Figure 8.18 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.18: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals for the ridge regression, grouped lasso, and elastic net methods, which is consistent with the histograms in Figure 8.13. The histograms shows that the majority of the residuals are within the Normal distribution curve. Note, that the residuals for the lasso was not normally distributed, which is illustrated in the histogram plot. Here it is shown that the majority of the residuals lies outside the normally distributed curve.

The residuals plotted against the fitted values are illustrated in Figure 8.19. The ridge regression, lasso, and elastic net residuals shows a random pattern dispersed around the horizontal axis, which means that the residuals are unbiased and homoscedastic, showing no clear linear relationship. The grouped lasso residuals shows a linear relationship, which indicates that they are biased and heteroscedastic.



Figure 8.19: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

Based on the above observations and measurements, we are now able to do model selection. The selected model is based on the elastic net method, because it exhibits the lowest prediction error. To find out how accurate the four models forecast, and to see if the selected model produces the best forecast, we compute a 2.5 year forecast of the inflation rate, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.18.

	RMSFE	MAFE
Ride Regression	0.00458	0.003254
Lasso	0.004545	0.003220
Grouped Lasso	0.004582	0.003254
Elastic Net	0.004583	0.003255

Table 8.18: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted inflation rate, using models constructed by the ridge

regression, lasso, grouped lasso and elastic net methods. The green and red numbers represents the highest and lowest errors, respectively.

Notice, that the selected model does not produce the most accurate forecast, that belongs to the model estimated by the lasso. However, the deviation between the two forecast errors are only 0,83 %. The forecasts are illustrated in Figure 8.10.



Figure 8.20: A comparison between the forecasted and the real values for the inflation rate. The forecasted values are based on the models estimated by the shrinkage methods.

8.2.4 Conclusion

In this section, we estimate four regression models for each of the three macroeconomic variables. In view of these models, we have determined which shrinkage method produces the most accurate model with respect to the variables. Furthermore, we have produced a forecast for each model in order to see if the selected model produces the most accurate forecast. The shrinkage methods corresponding to the most accurate model and forecast, for each of the three variable, are listed in Table 8.19.

	Model Selection	Best Forecast
GDP	Grouped Lasso	Lasso
Unemployment Rate	Elastic net	Ridge Regression
Inflation Rate	Elastic Net	Lasso

Table 8.19: The shrinkage methods corresponding to the most accurate model and forecast, for each of the three macroeconomic variable.

Notice, that the shrinkage methods which estimates the best models differs for the three variables, but does not include the ridge regression and the lasso method. A possible explanation could be that the ridge regression and lasso includes to many and to few predictor, respectively. The ridge regression method does not apply any shrinks penalty to the model. Hence, the method includes all the information from the 79 predictors, resulting in a model which overfits the training data. On the other hand, the lasso removes all collinearity between the predictors, and since our predictors are divided into 10 groups, where each group contains numerous predictors dealing with the same subjects, there exist a lot of collinearity inside each group. Hence, due to the extensive collinearity lasso will only select a few individual predictors of significant groups. Table 8.19 also lists the shrinkage methods which estimates the most accurate forecast models. We have noted, that the deviation between the best forecasts and the forecasts for the selected model only differs substantially for the unemployment rate. However, going forward we only include the forecasts for the selected models, because the future is unknown.

8.3 VAR Models

In this section we present the results for the FAVAR and lasso-VAR models, described in Chapter 6. The section is further divided into subsections, where each subsection presents the work for one of the three macroeconomic variables described in Chapter 2. All the models are estimated on the training set and their performance are evaluated on the validation set. Furthermore, the forecasts are evaluated on the forecast set.

8.3.1 Gross Domestic Product

In this section we estimate VAR models for predicting the GDP, where the GDP is the dependent variable and the remaining 79 macroeconomic time series in the training set are predictors.

The goodness-of-fit and prediction error for the models are listed in Table 8.20.

	No. Predictors/Factors	λ value	R^2	RMSE	MSPE
FAVAR	13	-	0.8453	0.003476	1.690936e-05
Lasso-VAR	26	7.398822e-03	0.9928	0.084541	0.665218

Table 8.20: The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green numbers represents the best measures.

The R^2 states that, for the lasso-VAR, the predictors explains 99.28 % of the variation in GDP, whereas for the FAVAR, 84.53 % is explained. The FAVAR models has the lowest root mean squared error, and the lowest prediction error. Hence, the FAVAR model capturers the GDP most accurate.

The relationship between the predictions and the true values of GDP are illustrated in Figure 8.21, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.21: A comparison between the predicted- and real values for the GDP. The predicted values are based on the FAVAR and lasso-VAR models.

The predictors included in the lasso-VAR model are shown in Table 8.21.

Industrial Production: Industrial equipment
Industrial Production: Final Product
Industrial Production: Mining
Industrial Production: Durable manufacturing
Civilian Labor Force
All Employees: Retail Trade
All Employees: Government: State Government
All Employees: Government: Local Government
Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing
Personal Consumption Expenditures: Durable Goods
Personal Consumption Expenditures: Services
Personal consumption expenditures: New autos
Housing Starts in South Census Region
Housing Starts in West Census Region
Retail Trade: Total
Value of Manufacturers' New Orders: Consumer Durable Goods Industries With Unfilled Orders
Producer Price Index: Crude Materials for Further Processing
Consumer Price Index for All Urban Consumers: Apparel
Consumer Price Index for All Urban Consumers: Durables
Reserves Of Depository Institutions, Nonborrowed
Monetary Base; Total
Effective Federal Funds Rate
TED Spread
S&P 500 Real Earnings Growth by Year
Canada / U.S. Foreign Exchange Rate
Switzerland / U.S. Foreign Exchange Rate

Lasso-VAR

Table 8.21: The predictors included in the lasso-VAR model.

Notice, that the lasso-VAR model includes numerous predictors which corresponds to the theoretical components shown in Figure 2.1. Because of the many theoretically explained predictors included in the lasso-VAR includes, we would expect a low prediction error. One explanation for the prediction error could be, that the model includes too many predictors, thus overfits the training set by adding to much variance.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
FAVAR	0.9091
Lasso-VAR	0.533

Table 8.22: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected. Table 8.22 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.22 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.22: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.22. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.23. The FAVAR model residuals shows a random pattern dispersed around the horizontal axis, which means that they are unbiased and homoscedastic, showing no clear linear relationship. The lasso-VAR model residuals shows a linear relationship, which indicates that they are biased, since the mean value is clearly not zero and homoscedastic, since all the predictors have approximately the same finite variance.



Figure 8.23: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

Normally, we only forecast on the selected model, because of its ability to describe the trend in the variable, up until today. Because we have removed the last 2.5 year from the data set, we are able to compute the forecast error. Hence, it could be interesting to compare the selected model forecast with the forecast for the remaining models, to see if the selected model actually produces the most accurate forecast.

The 2.5 year forecast of GDP, starting 2013-06-30 to 2015-12-31, are listed in Table 8.23.

	RMSFE	MAFE
FAVAR	0.005672	0.004513
Lasso-VAR	0.206723	0.206654

Table 8.23: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted GDP, for the FAVAR and lasso-VAR models. The green numbers represents the lowest errors.

Notice, that the selected model does produce the most accurate forecast. The forecasts are illustrated in Figure 8.24.



Figure 8.24: A comparison between the forecasted and the real values for the GDP. The forecasted values are based on the FAVAR and lasso-VAR models.

8.3.2 Unemployment Rate

In this section we estimate VAR models for predicting the unemployment rate, where the unemployment rate is the dependent variable and the remaining 79 macroeconomic time series in the training set are predictors.

The goodness-of-fit and prediction error for the models are listed in Table 8.24.

	No. Predictors/Factors	λ value	\mathbb{R}^2	RMSE	MSPE
FAVAR	5	-	0.9906	0.000170	8.287679e-06
Lasso-VAR	10	2.881969e-05	0.9940	0.000133	2.09088e-07

Table 8.24: The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green numbers represents the best measures.

The R^2 states that, for the lasso-VAR, the predictors explains 99.40 % of the variation in unemployment rate, whereas for the FAVAR, 99.06 % is explained. The lasso-VAR models has the lowest root mean squared error, and the lowest prediction error. Hence, the lasso-VAR model capturers the unemployment rate most accurate.

The relationship between the predictions and the true values of the unemployment rate are illustrated in Figure 8.25, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.25: A comparison between the predicted- and real values for the unemployment rate. The predicted values are based on the FAVAR and lasso-VAR models.

The predictors included in the lasso-VAR model are shown in Table 8.25.

Lasso-VAR
Civilian Unemployment Rate
All Employees: Retail Trade
Real Disposable Personal Income
Personal Consumption Expenditures: Services
Housing Starts in Northeast Census Region
Housing Starts in South Census Region
Housing Starts in West Census Region
Consumer Price Index for All Urban Consumers: Services
M1 Money Stock
Commercial and Industrial Loans, All Commercial Banks

Table 8.25: The predictors included in the lasso-VAR model.

Notice, that the lasso-VAR model for the unemployment rate includes the lagged unemployment rates. Furthermore, many of the included predictors corresponds to the theory regarding the unemployment rate. Hence, the low prediction error corresponds to the relatively small number of predictors and their theoretical meaning.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
FAVAR	0.2986
Lasso-VAR	0.1609

Table 8.26: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.26 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.26 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.26: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.22. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.27, which shows a random pattern dispersed around the horizontal axis. Hence, the residuals are unbiased and homoscedastic, showing no clear linear relationship.



Figure 8.27: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

To find out how accurate the FAVAR and lasso-VAR models forecast, and to see if the selected model produces the best forecast, we compute a 2.5 year forecast of the unemployment rate, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.27.

	RMSFE	MAFE
FAVAR	0.002483	0.002082
Lasso-VAR	0.000909	0.000810

Table 8.27: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted unemployment rate, for the FAVAR and lasso-VAR models. The green numbers represents the lowest errors.

Notice, that the selected model does produce the most accurate forecast. The forecasts are illustrated in Figure 8.28.



Figure 8.28: A comparison between the forecasted and the real values for the unemployment rate. The forecasted values are based on the FAVAR and lasso-VAR models.

8.3.3 Inflation Rate

In this section we estimate VAR models for predicting the inflation rate, where the inflation rate is the dependent variable and the remaining 79 macroeconomic time series in the training set are predictors.

The goodness-of-fit and prediction error for the models are listed in Table 8.28.

	No. Predictors/Factors	λ value	\mathbf{R}^2	RMSE	MSPE
FAVAR	13	-	0.7245	0.005013	1.673245e-05
Lasso-VAR	26	5.283400e-03	0.9938	0.078833	1.320784

Table 8.28: The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule, which measures the average magnitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green numbers represents the best measures.

The R^2 states that, for the lasso-VAR, the predictors explains 99.38 % of the variation in the inflation rate, whereas for the FAVAR, 72.45 % is explained. The FAVAR models has the lowest root mean squared error, and the lowest prediction error. Hence, the FAVAR model capturers the inflation rate most accurate.

The relationship between the predictions and the true values of inflation rate are illustrated in Figure 8.29, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.29: A comparison between the predicted- and real values for the inflation rate. The predicted values are based on the FAVAR and lasso-VAR models.

The predictors included in the lasso-VAR model are shown in Table 8.29.

Lasso-VAR
Industrial Production: Durable manufacturing
Civilian Labor Force
Civilian Employment
All Employees: Mining and Logging: Mining
All Employees: Retail Trade
All Employees: Other Services
Employees: Government: Local Government
Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing
Personal Consumption Expenditures: Nondurable Goods
Personal consumption expenditures: New autos
Housing Starts in Midwest Census Region
Real Manufacturing and Trade Industries Sales
Retail Trade: Total
Total Business Inventories
Value of Manufacturers' New Orders: Consumer Durable Goods Industries
Producer Price Index: Crude Energy Materials
Consumer Price Index for All Urban Consumers: Services M1 Money Stock
M2 Money Stock
Reserves Of Depository Institutions, Total
Effective Federal Funds Rate
6-Month Treasury Bill: Secondary Market Rate
TED Spread
S&P 500 Real Earnings Growth by Year
Japan / U.S. Foreign Exchange Rate
U.S. / Euro Foreign Exchange Rate

Lasso-VAR

Table 8.29: The predictors included in the lasso-VAR model.

Notice, that the lasso-VAR model only includes a few of the predictors which corresponds to the theoretical components, see Section 2.3. Do to the lag of relevant predictors, the high prediction error was expected.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test, p-value
FAVAR	0.507
Lasso-VAR	0.4141

Table 8.30: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.30 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the

null-hypothesis, that the residuals are normally distributed. In Figure 8.30 the histograms of the residuals are plotted together with their normal distribution.



Figure 8.30: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.30. The histograms shows that the majority of the residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.31. The FAVAR model residuals shows a random pattern dispersed around the horizontal axis, which means that they are unbiased and homoscedastic, showing no clear linear relationship. The lasso-VAR model residuals shows a linear relationship, which indicates that they are biased, since the mean value is clearly not zero and homoscedastic, since all the predictors have approximately the same finite variance.



Figure 8.31: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

To find out how accurate the FAVAR and lasso-VAR models forecast, and to see if the selected model produces the best forecast, we compute a 2.5 year forecast of the inflation rate, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.31.

	RMSFE	MAFE
FAVAR	0.005692	0.004622
Lasso-VAR	0.004439	0.003121

Table 8.31: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted inflation rate, for the FAVAR and lasso-VAR models. The green numbers represents the lowest errors.

Notice, that the selected model does not produce the most accurate forecast. The deviation between the two forecast errors are 22 %, which means that the lasso-VAR produces a much better forecast. The forecasts are illustrated in Figure 8.32.



Figure 8.32: A comparison between the forecasted and the real values for the inflation rate. The forecasted values are based on the FAVAR and lasso-VAR models.

8.3.4 Conclusion

In this section, we estimate the FAVAR and lasso-VAR model for each of the three macroeconomic variables. We have then determined which of the two models that describes the variables most accurate. Furthermore, we have produced a forecast for each model in order to determine whether the selected model produces the most accurate forecast. The most accurate VAR model and forecast, for each of the three variable, are listed in Table 8.32.

	Model Selection	Best Forecast
GDP	FAVAR	FAVAR
Unemployment Rate	Lasso-VAR	Lasso-VAR
Inflation Rate	FAVAR	Lasso-VAR

Table 8.32: The most accurate VAR model and forecast, for each of the three variable.

Table 8.19 lists the shrinkage methods which produces to most accurate forecasts. We have noted, that the selected model also produces the most accurate forecasts, except in the case of the inflation rate. Here the deviation between the best forecasts and the forecasts for the selected model differs substantially. However, going forward we only include the forecasts for the selected models, because the future is unknown.

8.4 Bayesian Model Average

In this section we present the results for the Bayesian model average, described in Chapter 7. All the models are estimated on the training set and their performance are evaluated on the validation set. Furthermore, the forecasts are evaluated on the forecast set.

The goodness-of-fit and prediction error for the models describing the three macroeconomic variables are listed in Table 8.33.

	\mathbf{R}^2	RMSE	MSPE
GDP	0.9997345	8.952136e-05	0.0001735943
Unemployment Rate	0.964166	0.0003218741	6.453361e-07
Inflation Rate	0.9921487	0.0008241812	2.947779e-05

Table 8.33: The goodness-of-fit and validation of the optimal models. R² is a measure of the variance in the dependent variable that is predictable from the predictors. The root mean squared error (RMSE) is a quadratic scoring rule which measures the average magnitude of the error. The R² and the RMSE are computed using the training set. The mean prediction error (MSPE) is the expected value of the squared difference between the fitted values and the true values. The MSPE are computed using the validation set. The green numbers represents the best measures.

Notice, that in the Bayesian model average the factors explains close to 100 % of the variation in the three macroeconomic variables.

The mean squared prediction errors indicates that the Bayesian model average captures most of the trends in the three variables. The relationship between the predictions and the true values of the variables are illustrated in Figure 8.33, where it is seen that the predictions are in fact able to capture most of the spikes and lows of the variables.



Figure 8.33: A comparison between the predicted- and real values for the three macroeconomic variables. The predicted values are based on the Bayesian model average.

The predictors included in the Bayesian model average are listed in Table 8.34.

GDP

Industrial Production: Manufacturing Industrial Production: Business Equipment Industrial Production: Durable Materials Industrial Production: Durable manufacturing Civilian Unemployment Rate Average (Mean) Duration of Unemployment All Employees: Manufacturing All Employees: Nondurable goods Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing Personal Consumption Expenditures: Durable Goods Housing Starts in West Census Region Retail Trade: Total Merchant Wholesalers Inventories Value of Manufacturers' New Orders for Nondurable Goods Industries Producer Price Index: Crude Materials for Further Processing M2 Money Stock Reserves Of Depository Institutions, Total Effective Federal Funds Rate 3-Month Treasury Bill: Secondary Market Rate TED Spread S&P 500 Real Earnings Growth by Year

Unemployment Rate

Average (Mean) Duration of Unemployment Housing Starts in West Census Region Commercial and Industrial Loans, All Commercial Banks

Inflation Rate

Consumer Price Index for All Urban Consumers: All Items Less Food

Table 8.34: The predictors included in Bayesian model average.

Notice, that there are numerous predictors, which corresponds to the theoretical components shown in Figure 2.1, included in the GDP. Because of the many theoretically explained predictors included, we would expect a low prediction error. However, according to Table 8.33 the GDP has a much larger prediction error than the two other variables. This could indicate that the Bayesian model average includes too many predictors in the GDP model, thus overfits the training set by adding to much variance.

The Bayesian model average for the unemployment rate only includes three predictors, however, one of the three are the average duration of unemployment. Hence, we expect a low prediction error, which is backed by Table 8.33. Finally, the Bayesian model average for the inflation rate only includes the consumer price index. This predictor correspond to the theory that inflation rate are computed by the consumer price index, see Section 2.3, thus we expect af low prediction error, which is backed by Table 8.33.

To see if the model requirements are met, we conduct a test of normality on the residuals, as well as a residual plot against the fitted values, to see if they are unbiased and homoscedastic.

	Shapiro-Wilks Test
GDP	0.6909
Unemployment Rate	0.247
Inflation Rate	0.1358

Table 8.35: The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test is that the residuals are normally distributed. If the p-values is lower than 0.05, the null hypothesis is rejected.

Table 8.35 includes the p-values from the Shapiro-Wilks test, from which we cannot reject the null-hypothesis, that the residuals are normally distributed. In Figure 8.34 the histograms of the residuals are plotted together with their normal distribution.



(c) Inflation Rate.

Figure 8.34: Histograms of the residuals plotted together with their normal distribution. The y-axis represents the number of residuals included in each column, and the x-axis represents the value of the residuals in increasing order.

The Shapiro-Wilks test's p-values shows evidence of normally distributed residuals, which is consistent with the histograms in Figure 8.34. The histograms shows that the majority of the

residuals are within the Normal distribution curve.

The residuals plotted against the fitted values are illustrated in Figure 8.35, which, apart from the outliers, shows a random pattern dispersed around the horizontal axis. This means that the residuals are unbiased and homoscedastic, showing no clear linear relationship.



(c) Inflation Rate.

Figure 8.35: The residuals plotted against the fitted values. The y-axis represents the value of the residuals, in increasing order, and the x-axis represents the value of the fitted values, in increasing order.

So far, we have shown that the computed Bayesian model average describes the three macroeconomic variables reasonable, and that they approximately fulfills the model requirements. This could indicate that these models are reasonable to use when forecasting, thus they are used in the computation of a 2.5 year forecast, starting 2013-06-30 to 2015-12-31. The forecast errors are listed in Table 8.36.

	Root Mean Square Error	Mean Absolute Error
GDP	0.007846273	0.006739625
Unemployment Rate	0.0006366026	0.0004500587
Inflation Rate	0.006793207	0.005862169

Table 8.36: The root mean squared forecast error (RMSFE) and the mean absolute forecast error (MAFE) for the forecasted GDP, unemployment rate, and inflation rate. The green numbers represents the lowest error.

The analysis of these forecast errors will be conducted in Section 8.5. However, Figure 8.36, which illustrates the forecast relative to the true value of the variables, gives an indication of the forecast accuracy.



Figure 8.36: A comparison between the forecasted and the real values for the three macroeconomic variables. The forecasted values are based on the Bayesian model average.

8.5 Empirical Conclusion

In this empirical analysis we evaluate different groups of models, and their ability to capture the trends in the three macroeconomic variables. We introduces four groups of models: The dynamic factor models, shrinkage methods, VAR models, and Bayesian model average.

We use model selection to determine the most accurate model within each group. The accuracy

of the model is determined based on the mean squared prediction error, thus the model with the lowest prediction error is selected as the model which is presumed to yield the most accurate forecast. The four selected models and their prediction errors are listed in Table 8.37.

	GDP	Unemployment Rate	Inflation Rate
Dynamic Factor Model	4.909782e-05	1.006426e-06	2.333727e-05
Shrinkage Estimator	3.99307e-05	6.902786e-07	1.693075e-05
VAR Model	1.690936e-05	2.09088e-07	1.673245e-05
Bayesian Model Average	0.0001735943	6.453361e-07	2.947779e-05

Table 8.37: Prediction error for each of the selected models.

Notice, that the VAR models have the lowest prediction error for all three variables, thus in our empirical analysis the VAR models exhibits the trends that comes closest to the true variables. The GDP and the inflation rate are estimated by the FAVAR model, and the unemployment rate is estimated by the lasso-VAR model. The forecasts are only conducted on the selected models, and they are compared to the benchmark model, AR(p), to validate if the selected models forecast outperforms the benchmarks forecast.

The forecast accuracy, for the selected models, is based on the root mean squared forecast error. It can be difficult to conclude anything about the forecasts solely based on the root mean square forecast error. Therefore, we perform a Diebold-Mariano test, which makes it possible to conclude if our more advanced/complex model outperforms the benchmark or not. The null-hypothesis states that both methods have the same forecast accuracy, and a p-value < 0.05 rejects the null-hypothesis.

In order determine if the FAVAR and the lasso-VAR forecast more accurate then the benchmark, the p-values from the Diebold-Mariano test are listed in Table 8.38.

	Diebold-Mariano test, p-value
GDP	0.5
Unemployment rate	0.8
Inflation rate	0.5

Table 8.38: The Diebold-Mariano test for predictive accuracy. The null-hypothesis of the Diebold-Mariano test is that both forecast have the same accuracy. If the p-values is lower than 0.05, the null-hypothesis is rejected.

Notice, that the null-hypothesis are not rejected, which means that none of the models outperform each other. Hence, our attempt to introduce more complex model in order to make a better forecast, of the three macroeconomic variables, did not succeed.

8.5.1 Comparison of the Included Predictors

In this analysis we comment on which predictors the selective models includes. But, is there any predictors which are included in all the model?

The answer is no, however, if we examine the relationship between the predictors corresponding to the three macroeconomic variables, there are some which are included in all the models.

GDP		
S&P 500 Real Sales Growth by Quarter		
M2 Money Stock		
Unemployment rate		
Commercial and Industrial Loans, All Commercial Banks		
Housing Starts in West Census Region		
Inflation rate		

Consumer Price Index for All Urban Consumers: All Items Less Food

Table 8.39: The predictors which all the models has in common relative to the macroeconomic variables.

Notice, that all models describing GDP includes: S&P 500 Real Sales Growth by Quarter, which has a direct impact on GDP. In the same way, all the models which describes the inflation rate includes: Consumer Price Index for All Urban Consumers: All Items Less Food, which is consistent with the theoretical calculation of the inflation rate.

CHAPTER 9

CONCLUSION

This master's thesis empirically examen the macroeconomic model abilities for a range of different methods, designed to handle a large number of predictors. The included methods are: the dynamic factor model with principal component analysis, ridge regression, lasso, grouped lasso, elastic net, factor-augmented VAR, lasso-VAR, and Bayesian model average. When the optimal models are determined, the forecast accuracy of these models are examined.

We consider a constant forecast horizon of 2.5 years, and we evaluate the models in term of their ability to predict the gross domestic product, unemployment rate, and inflation rate for the U.S. When the optimal model for each variable are determined, we formally compare the forecasting accuracy of these models to the benchmark, by applying the Diebold-Mariano test.

According to our model selection the VAR models are the preferred models for describing the tree variables.

	GDP	Unemployment Rate	Inflation Rate
Dynamic Factor Model	4.909782e-05	1.006426e-06	2.333727e-05
Shrinkage Estimator	3.99307e-05	6.902786e-07	1.693075e-05
VAR Model	1.690936e-05	2.09088e-07	1.673245e-05
Bayesian Model Average	0.0001735943	6.453361e-07	2.947779e-05

Table 9.1: Prediction error for each of the selected models.

The GDP and the inflation rate are estimated by the FAVAR model, and the unemployment rate is estimated by the lasso-VAR model.

Furthermore, we produce the forecasts for the VAR models, and finds that non of them outperforms the benchmark, AR(p), model. Hence, our complex models do not improve upon the forecast made by the simple AR model.

These results are partially in agreement with other studies, such as [Stock and Watson, 1999], and [Kim and Swanson, 2014], in which the forecasts for the inflation rate do not outperform its benchmark. A different explanation could be that we have limited our analysis to only consider linear models.

According to [Kim and Swanson, 2014], the chosen models should be able to forecast macroeconomic variables, which could indicate that we might have to change our dependent variables in order to improve the forecasts.

An extension to this report is by including the shrinkage estimator and/or the Bayesian model average in the dynamic factors model, suggested by [Koop and Potter, 2004], [Kim and Swanson, 2014], [Stock and Watson, 2012].

An further extension is to apply a rolling forecast origin techniques, suggested by [?], because it moves the training and validation set in time, instead of sampling them random.
LIST OF FIGURES

0.1	Projektstruktur.	
2.1	The macroeconomic variable GDP, comprises four main components, that is con- sumption, investment, government purchase, and net exports. Consumption and investment can be further divided into three subcomponents.	5
3.1	The behaviour of prediction and training errors as the model complexity is varied. The solid blue curve show the expected prediction error, Err , and the solid red curve shows the expected training error, $E[\overline{err}]$. The light blue curves is the training	
	errors \overline{err} , and the light red curves is the conditional prediction errors $Err_{\mathcal{T}}$, for 100 training sets of size 50 each, as the model complexity is increased. Source: [Hastie et al., 2008]	10
3.2	The k-fold CV approach with k=5. In this case we fit the model to the first, second, fourth, and fifth group of the date, and calculate the prediction error of the fitted model by predicting on the third part of the data. Source: [Hastie et al., 2008] \ldots	13
5.1	Illustration of the estimation of the lasso (left) and the ridge regression (right). Here, the blue diamond and circle represents the constraint regions for the lasso and ridge regression respectively, the ellipses centered around $\hat{\beta}$ represent regions with constant residual sum of squares, and $\hat{\beta}$ represent the least squares solution. Source:[James	
5.2	et al., 2013]	28
5.3	ℓ_2 penalty. Source: [Yuan and Lin, 2007]	32
	et al., 2010]	34

6.1	The sparsity pattern for a lasso-VAR model with $p = 5$ and $k = 3$. The blue squares	
	are the coefficients that are included in the models, and the white are the coefficients	
	that are set to zero. Source: [Nicholson et al., 2014]	37
8.1	The three macroeconomic variables, constructed on quarterly measured U.S. time	
	series starting 2001-12-31 to 2015-12-31	43
8.2	A comparison between the predicted- and real values for the three macroeconomic	
	variables. The predicted values are based on the dynamic factor model with principal	
	components	46
8.3	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order. \ldots \ldots \ldots \ldots \ldots	47
8.4	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	48
8.5	A comparison between the forecasted and the real values for the three macroeconomic	
	variables. The forecasted values are based on the dynamic factor model with principal	
	components	49
8.6	The cross validation path, which shows the relationship between the mean squared	
	error and the λ values suggested to the model. The y-axis represents the mean	
	squared errors of the model, and the x-axis represents the $\log(\lambda)$ values tested in the	
	model. Furthermore, the top-line shows the number of predictors included in the	
	model	50
8.7	A comparison between the predicted- and real values for the GDP. The predicted	
	values are based on the models estimated by the shrinkage methods	53
8.8	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order. \ldots \ldots \ldots \ldots \ldots	54
8.9	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	55
8.10	A comparison between the forecasted and the real values for the GDP. The forecasted	
	values are based on the models estimated by the shrinkage methods	56

8.11	The cross validation path, which shows the relationship between the mean squared	
	error and the λ values suggested to the model. The y-axis represents the mean	
	squared errors of the model, and the x-axis represents the $\log(\lambda)$ values tested in the	
	model. Furthermore, the top-line shows the number of predictors included in the	
	model	57
8.12	A comparison between the predicted- and real values for the unemployment rate.	
	The predicted values are based on the models estimated by the shrinkage methods	60
8.13	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order	61
8.14	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	62
8.15	A comparison between the forecasted and the real values for the unemployment rate.	
	The forecasted values are based on the models estimated by the shrinkage methods.	63
8.16	The cross validation path, which shows the relationship between the mean squared	
	error and the λ values suggested to the model. The y-axis represents the mean	
	squared errors of the model, and the x-axis represents the $\log(\lambda)$ values tested in the	
	model. Furthermore, the top-line shows the number of predictors included in the	
	model	64
8.17	A comparison between the predicted- and real values for the inflation rate. The	
	predicted values are based on the models estimated by the shrinkage methods. \ldots	66
8.18	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order	67
8.19	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	68
8.20	A comparison between the forecasted and the real values for the inflation rate. The	
	forecasted values are based on the models estimated by the shrinkage methods	69
8.21	A comparison between the predicted- and real values for the GDP. The predicted	
	values are based on the FAVAR and lasso-VAR models.	71
8.22	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order	73

8.23	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	74
8.24	A comparison between the forecasted and the real values for the GDP. The forecasted	
	values are based on the FAVAR and lasso-VAR models	75
8.25	A comparison between the predicted- and real values for the unemployment rate.	
	The predicted values are based on the FAVAR and lasso-VAR models	76
8.26	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order	77
8.27	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	78
8.28	A comparison between the forecasted and the real values for the unemployment rate.	
	The forecasted values are based on the FAVAR and lasso-VAR models	78
8.29	A comparison between the predicted- and real values for the inflation rate. The	
	predicted values are based on the FAVAR and lasso-VAR models	79
8.30	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order	81
8.31	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	81
8.32	A comparison between the forecasted and the real values for the inflation rate. The	
	forecasted values are based on the FAVAR and lasso-VAR models	82
8.33	A comparison between the predicted- and real values for the three macroeconomic	
	variables. The predicted values are based on the Bayesian model average	84
8.34	Histograms of the residuals plotted together with their normal distribution. The	
	y-axis represents the number of residuals included in each column, and the x-axis	
	represents the value of the residuals in increasing order. \ldots \ldots \ldots \ldots \ldots \ldots	86
8.35	The residuals plotted against the fitted values. The y-axis represents the value of the	
	residuals, in increasing order, and the x-axis represents the value of the fitted values,	
	in increasing order	87
8.36	A comparison between the forecasted and the real values for the three macroeconomic	
	variables. The forecasted values are based on the Bayesian model average	88

LIST OF TABLES

8.1	The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance	
	in the dependent variable that is predictable from the predictors. The root mean	
	squared error (RMSE) is a quadratic scoring rule, which measures the average mag-	
	nitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green numbers represents the best measures. \ldots . \ldots . \ldots	45
8.2	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05 , the	
	null hypothesis is rejected	46
8.3	The root mean squared forecast error (RMSFE) and the mean absolute forecast error $% \left(\mathbf{RMSFE}\right) = \left(\mathbf{RMSFE}\right) \left(\mathbf{RMSFE}\right$	
	(MAFE) for the forecasted GDP, unemployment rate, and inflation rate. The green	
	numbers represents the lowest error	48
8.4	The optimal value for the tuning parameter λ , and the number of predictors included	
	in the optimal model. \ldots	50
8.5	The predictors included by the lasso, grouped lasso and elastic net	51
8.6	The goodness-of-fit and validation of the optimal models. \mathbb{R}^2 is a measure of the	
	variance in the dependent variable that is predictable from the predictors. The root	
	mean squared error (RMSE) is a quadratic scoring rule, which measures the average $% \left({{\rm{RMSE}}} \right)$	
	magnitude of the error. The \mathbf{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green and red numbers represents the best and worst measures,	
	respectively	52
8.7	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	53

8.8	The root mean squared forecast error (RMSFE) and the mean absolute forecast error $% \left(\mathbf{RMSFE}\right) = \left(\mathbf{RMSFE}\right) \left(\mathbf{RMSFE}\right$	
	(MAFE) for the forecasted GDP, using models constructed by the ridge regression,	
	lasso, grouped lasso and elastic net methods. The green and red numbers represents	
	the highest and lowest errors, respectively	56
8.9	The optimal value for the tuning parameter λ , and the number of predictors included	
	in the optimal model	57
8.10	The predictors included by the lasso, grouped lasso and elastic net	58
8.11	The goodness-of-fit and validation of the optimal models. \mathbf{R}^2 is a measure of the	
	variance in the dependent variable that is predictable from the predictors. The root	
	mean squared error (RMSE) is a quadratic scoring rule, which measures the average	
	magnitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green and red numbers represents the best and worst measures,	
	respectively	59
8.12	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05 , the	
	null hypothesis is rejected	60
8.13	The root mean squared forecast error (RMSFE) and the mean absolute forecast error $% \left(\mathbf{RMSFE}\right) = \left(\mathbf{RMSFE}\right) \right) \left(\mathbf{RMSFE}\right) \left(RMSF$	
	(MAFE) for the forecasted unemployment rate, using models constructed by the ridge	
	regression, lasso, grouped lasso and elastic net methods. The green and red numbers	
	represents the highest and lowest errors, respectively	62
8.14	The optimal value for the tuning parameter λ , and the number of predictors included	
	in the optimal model	64
8.15	The predictors included by the lasso, grouped lasso and elastic net	65
8.16	The goodness-of-fit and validation of the optimal models. \mathbf{R}^2 is a measure of the	
	variance in the dependent variable that is predictable from the predictors. The root	
	mean squared error (RMSE) is a quadratic scoring rule, which measures the average	
	magnitude of the error. The \mathbf{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green and red numbers represents the best and worst measures,	
	respectively	66

8.17	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	67
8.18	The root mean squared forecast error (RMSFE) and the mean absolute forecast	
	error (MAFE) for the forecasted inflation rate, using models constructed by the	
	ridge regression, lasso, grouped lasso and elastic net methods. The green and red	
	numbers represents the highest and lowest errors, respectively. \ldots	69
8.19	The shrinkage methods corresponding to the most accurate model and forecast, for	
	each of the three macroeconomic variable	70
8.20	The goodness-of-fit and validation of the models. \mathbf{R}^2 is a measure of the variance	
	in the dependent variable that is predictable from the predictors. The root mean	
	squared error (RMSE) is a quadratic scoring rule, which measures the average mag-	
	nitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green numbers represents the best measures. \ldots . \ldots . \ldots	71
8.21	The predictors included in the lasso-VAR model	72
8.22	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	72
8.23	The root mean squared forecast error (RMSFE) and the mean absolute forecast error $% \left(\mathbf{RMSFE}\right) = \left(\mathbf{RMSFE}\right) \left(\mathbf{RMSFE}\right$	
	(MAFE) for the forecasted GDP, for the FAVAR and lasso-VAR models. The green	
	numbers represents the lowest errors	74
8.24	The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance	
	in the dependent variable that is predictable from the predictors. The root mean	
	squared error (RMSE) is a quadratic scoring rule, which measures the average mag-	
	nitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green numbers represents the best measures	75
8.25	The predictors included in the lasso-VAR model	76
8.26	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	77

8.27	The root mean squared forecast error (RMSFE) and the mean absolute forecast	
	error (MAFE) for the forecasted unemployment rate, for the FAVAR and lasso-VAR	
	models. The green numbers represents the lowest errors	78
8.28	The goodness-of-fit and validation of the models. \mathbb{R}^2 is a measure of the variance	
	in the dependent variable that is predictable from the predictors. The root mean	
	squared error (RMSE) is a quadratic scoring rule, which measures the average mag-	
	nitude of the error. The \mathbb{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green numbers represents the best measures	79
8.29	The predictors included in the lasso-VAR model	80
8.30	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	80
8.31	The root mean squared forecast error (RMSFE) and the mean absolute forecast error	
	(MAFE) for the forecasted inflation rate, for the FAVAR and lasso-VAR models. The	
	green numbers represents the lowest errors.	82
8.32	The most accurate VAR model and forecast, for each of the three variable. \hdots	83
8.33	The goodness-of-fit and validation of the optimal models. \mathbb{R}^2 is a measure of the	
	variance in the dependent variable that is predictable from the predictors. The root	
	mean squared error (RMSE) is a quadratic scoring rule which measures the average	
	magnitude of the error. The \mathbf{R}^2 and the RMSE are computed using the training set.	
	The mean prediction error (MSPE) is the expected value of the squared difference	
	between the fitted values and the true values. The MSPE are computed using the	
	validation set. The green numbers represents the best measures. \ldots \ldots \ldots \ldots	83
8.34	The predictors included in Bayesian model average.	85
8.35	The Shapiro-Wilks test of normality. The null-hypothesis of the Shapiro-Wilks test	
	is that the residuals are normally distributed. If the p-values is lower than 0.05, the	
	null hypothesis is rejected	86
8.36	The root mean squared forecast error (RMSFE) and the mean absolute forecast error	
	(MAFE) for the forecasted GDP, unemployment rate, and inflation rate. The green	
	numbers represents the lowest error	88
8.37	Prediction error for each of the selected models.	89

8.38	The Diebold-Mariano test for predictive accuracy. The null-hypothesis of the Diebold-	
	Mariano test is that both forecast have the same accuracy. If the p-values is lower	
	than 0.05, the null-hypothesis is rejected.	89
8.39	The predictors which all the models has in common relative to the macroeconomic	
	variables.	90
9.1	Prediction error for each of the selected models.	91

BIBLIOGRAPHY

- Adams, R. A. and Essex, C. (2010). Calculus a Complete Course. Pearson, seventh edition. 26
- Arrow, K. J. and Intriligator, M. D. (2006). Handbook Of Economic Forecasting Volume 1. Elsevier. 15, 39
- Bai, J. and Ng, S. (2002). Determining the number of flactor in approximate factor models. *Econometrica*, Vol. 70. 19
- Bakin, S. (1999). Adaptive Regression and Model Selection in Data Mining Problems. PhD thesis, The Australian national University. 1, 32
- Bernanke, B. S., Boivin, J., and Eliasz, P. (2005). Measuring the effects of monetary policy: A factor-augmented vector autoregressive (favar) approach. *The Quarterly Journal of Economics*, pages 387–422. 35, 36
- Blanchard, O., Amighini, A., and Giavazzi, F. (2010). Macroeconomics a European Perspective. Pearson. 3
- BLS (2016a). National unemployment rate (from the current population survey). http://www. bls.gov/cps/. Retrieved 2016-04-15. 5
- BLS (2016b). Overview of bls statistics on inflation and prices. http://www.bls.gov/bls/ inflation.htm. Retrieved 2016-04-15. 6
- Brillinger, D. R. (1964). A frequency approach to the techniques of principal components, factor analysis and canonical variates in the case of stationary time series. *Unknown*. 15
- Diebold, F. X. and Mariano., R. S. (1995). Comparing predictive accuracy. Journal of Business and Economic Statistics. 1
- Diebold, F. X., Rudebusch, G. D., and Aruoba, S. B. (2006). The macroeconomy and the yield curve: A dynamic latent factor approach. *Journal of Econometrics*, (131):309–338. 17
- Economics, P. (2016). In search of the perfect recession indicator. http://www.philosophicaleconomics.com/2016/02/uetrend/. Retrieved 2016-05-26. 44

- Fernandez, C., Ley, E., and Steel, M. F. J. (2001). Benchmark priors for bayesian model averaging. *Journal of Econometrics*, (381-427). 39, 40, 41
- Fornaro, P. (2011). Dynamic factor models and forecasting finnish macroeconomic variables. Master's Thesis 2011. 15
- Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Statistical Software*, 33(1). Retrieved 2016-05-09. 33
- Froyen, R. T. (2013). Macroeconomics Theoremies and Policies. Pearson, tenth edition. 3
- Geweke, J. (1977). The dynamic factor analysis of economic time series. Unknown. 15
- Hastie, T., Tibshirani, R., and Friedman, J. (2008). The Elements of Statistical Learning. Springer, second edition. 9, 10, 13, 21, 93
- Hastie, T., Tibshirani, R., and Friedman, J. (2010). A note on the group lasso and a sparse group lasso. Retrieved 2016-05-08. 32, 34, 93
- Hastie, T., Tibshirani, R., and Wainwright, M. (2015). Statistical Learning with Sparsity: The Lasso and Generalizations. CRS Press. 27, 29
- Hayashi, F. (2000). Econometrics. Princeton University Press. 13
- Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. 1, 22
- Hsu, N. J., Hung, H. L., and Chang, Y. M. (2008). Subset selection for vector autoregressive processes using lasso. *Computational Statistics and Data Analysis*, 52(7):3645–3657. 35
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer. 9, 21, 28, 93
- Josse, J. and Husson, F. (2012). Selecting the number of components in principal component analysis usin cross-validation approximations. *Computational Statistics and Data Analysis*. 19
- Kim, H. H. and Swanson, N. R. (2014). Forecasting financial and macroeconomic variables using data reduction methods: New empirical evidence. 91, 92
- Koop, G. and Potter, S. (2004). Forecasting in dynamic factor models using bayesian model averaging. The Econometrics Journal. 39, 41, 92

- Leamer, E. E. (1978). Specification searches. Technical report. 39
- Litterman, R. B. (1979). Techniques of forecasting using vector autoregressions. 35
- Lutkepohl, H. (2005). New Introduction to Multiple Time Series Analysis. Springer. 13
- Nicholson, W. B., Matteson, D. S., and Bien, J. (2014). Structured regularization for large vector autoregression. Technical report, ,Cornell University. 36, 37, 94
- Sims, C. A. (1980). Macroeconomics and reality. Econometrica: Journal of the Econometric Society, 48(1):1–48. 35
- Stock, J. H. and Watson, M. W. (1989). New indexes of coincident and leading economic indicators. NBER Macroeconomics Annual 1989, 4:351–409. 17
- Stock, J. H. and Watson, M. W. (1999). Forecasting inflation. Journal of Monetary Economics, pages 293 – 335. 91
- Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*. 1, 17, 19
- Stock, J. H. and Watson, M. W. (2010). Dynamic factor models. 15, 16
- Stock, J. H. and Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business and Economic Statistics*. 92
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, 58(1):267–288. 1, 27
- Watson, M. W. (2000). Macroeconomic forecasting using many predictors. https://www.princeton.edu/~mwatson/papers/wc_3.pdf. Retrieved 2016-05-08. 44
- Weatherwax, J. L. and Epstein, D. (2013). A solution manual and notes for: The elements of statistical learning. Technical report. 25
- Wikipedia (2016). Law of total expectation. https://en.wikipedia.org/wiki/Law_of_total_ expectation. Retrieved 2016-06-07. 40
- Wright, S. J. (2015). Coordinate descent algorithms. Mathematical Programming, 151(101017):3–34. Retrieved 2016-05-07. 27, 29
- Yuan, M. and Lin, Y. (2007). Model selection and estimation in regression with grouped variables. Royal Statistical Society, pages 49–67. Retrieved 2016-05-08. 32, 93

- Zellner, A. (1986). On assessing prior distributions and bayesian regression analysis with g-prior distributions. 1, 40
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, (301-323). 1, 33

APPENDIX A

DATA

The first column contains the name of the time series used in the r-code, the second column contains a number, which corresponds to the following transformations: (1) level of the series; (2) first difference, (3) logarithm of the series; (4) first difference of the logarithm; (6) the second difference of the logarithm. The following abbreviations appear in the last column; SA = seasonally adjusted, NSA = not seasonally adjusted, SAAR = seasonally adjusted by an annual rate.

Industrial Production Growth:

GDP	4	Gross Domestic Product, 1 Decimal	(Billions of Dollars SA)
IP	4	Industrial Production Index	(Index 2007=100 SA)
IP.IE	4	Industrial Production: Industrial equipment	(Index 2007=100 SA)
IP.FP	4	Industrial Production: Final Product	(Index 2007=100 SA)
IP.Man	4	Industrial Production: Manufacturing (NAICS)	(Index 2007=100 SA)
IP.BE	4	Industrial Production: Business Equipment	(Index 2007=100 SA)
IP.CG	4	Industrial Production: Consumer Goods	(Index 2007=100 SA)
IP.Min	4	Industrial Production: Mining	(Index 2007=100 SA)
IP.M	4	Industrial Production: Materials	(Index 2007=100 SA)
IP.DM	4	Industrial Production: Durable Materials	(Index 2007=100 SA)
IP.NdM	4	Industrial Production: nondurable Materials	(Index 2007=100 SA)
IP.DMan	4	Industrial Production: Durable manufacturing	(Index 2007=100 SA)
IP.NdMan	4	Industrial Production: Nondurable manufacturing	$(Index \ 2007 = 100 \ SA)$

CUR		1 Civilian Unemployment Rate	(\mathbf{P})	ct., SA)
CLF		4 Civilian Labor Force	(Th. of	Persons, SA)
CE		4 Civilian Employment	(Th.s of	Persons, SA)
AMDU		1 Average (Mean) Duration of Unemployment		(SA)
E.T		4 All Employees: Total nonfarm	(Th. of	Persons SA)
E.Man		4 All Employees: Manufacturing	(Th. of	Persons SA)
E.DG		4 All Employees: Durable goods	(Th. of	Persons SA)
E.Min		4 All Employees: Mining and Logging: Mining	(Th	. of SA)
E.CCB		4 All Employees: Construction: Construction of Build	lings (Th. of	Persons SA)
E.NdG		4 All Employees: Nondurable goods	(Th. of	Persons SA)
E.WT		4 All Employees: Wholesale Trade	(Th. of	Persons SA)
E.RT		4 All Employees: Retail Trade	(Th. of	Persons SA)
E.IS		4 All Employees: Information Services	(Th. of	Persons SA)
E.OS		4 All Employees: Other Services	(Th. of	Persons SA)
E.SG		4 All Employees: Government: State Government	(Th. of	Persons SA)
E.LG		4 All Employees: Government: Local Government	(Th. of	Persons SA)
AWHP.Ma	n	1 Average Weekly Hours of Production: Manufactur	ing (Ho	ours, SA)
Real Perso	ona	l Income and consumption Growth:		
PI	4	Real Disposable Personal Income	(Billions,	SAAR)
PCE.DG	4	Personal Consumption Expenditures: Durable Good	ls (Billions,	SAAR)
PCE.NdG	4	Personal Consumption Expenditures: Nondurable Go	ods (Billions,	SAAR)
PCE.S	4	Personal Consumption Expenditures: Services	(Billions,	SAAR)
PCE.NC	4	Personal consumption expenditures: New autos		
Housing S	tar	ts Growth:		
HS.T	3	Housing Starts: Total: New Privately Owned Housing	Units Started	(Th.Units SAAR)
HS.NE	3	Housing Starts in Northeast Census Region	1	(Th.Units SAAR)
HS. MW	3	Housing Starts in Midwest Census Region		(Th.Units SAAR)
HS.S	3	Housing Starts in South Census Region		(Th.Units SAAR)
HS.W	3	Housing Starts in West Census Region		(Th.Units SAAR)
Manufactu	urin	ng and retail sale:		
MTS.I 4	R	Real Manufacturing and Trade Industries Sales (Million	ns, SA)	
RT.T 4		Retail Trade: Total (Million	ns, SA)	

Employment and Unemployment Growth

$Real\ inventories\ and\ inventory-sales\ ratios:$

IB.T	1	Total Business Inventories	(Pct. Change SA)
MWI	4	Merchant Wholesalers Inventories	(Millions, SA)
VMNO.ManI	4	Value.Man' New Orders for All Manufacturing Industries	(Million, SA)
VMNO.ManIUO	4	Value.Man' New Orders A.M. Industries With Unfilled Orders	(Million, SA)
VMNO.DGI	4	Value.Man' New Orders: Consumer Durable Goods Industries	(Million, SA)
VMNO.DGIUO	4	Value.Man' New Orders: C.D.G.I With Unfilled Orders	(Million, SA)
VMNO.NdGI	4	Value.Man' New Orders for Nondurable Goods Industries	(Million, SA)

Prices, Wages and Inflation:

5	Consumer Price Index: All Items	(Index 1982-84=100 SA)
5	Producer Price Index: All Commodities	(Index 1982=100 Not SA)
5	Producer Price Index: Finished Goods	(Index 1982 = 100 SA)
5	Producer Price Index: Finished Consumer Goods	(Index 1982 = 100 SA)
5	Producer Price Index: Intermediate Materials	(Index 1982 = 100 SA)
5	Producer Price Index: Crude Energy Materials	(Index 1982 = 100 SA)
5	Producer Price Index: C. Materials for Further Processing	(Index 1982 = 100 SA)
5	Consumer Price Index for All Urban Consumers: Apparel	$(Index 1982-84{=}100 \text{ SA})$
5	Consumer Price Index for A. Urban Consumers: Transportation	$(Index 1982-84{=}100 \text{ SA})$
5	Consumer Price Index for All Urban Consumers: Durables	$(Index 1982-84{=}100 \text{ SA})$
5	Consumer Price Index for All Urban Consumers: Services	$(Index 1982-84{=}100 \text{ SA})$
5	C.P.I for All Urban Consumers: All Items Less Food	$(Index 1982-84{=}100 \text{ SA})$
	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	 5 Consumer Price Index: All Items 5 Producer Price Index: All Commodities 5 Producer Price Index: Finished Goods 5 Producer Price Index: Finished Consumer Goods 5 Producer Price Index: Intermediate Materials 5 Producer Price Index: Crude Energy Materials 5 Producer Price Index: C. Materials for Further Processing 5 Consumer Price Index for All Urban Consumers: Apparel 5 Consumer Price Index for All Urban Consumers: Services 5 Consumer Price Index for All Urban Consumers: Services 5 Consumer Price Index for All Urban Consumers: Services

$Money\ and\ credit\ quantity\ aggregates:$

M1	5	M1 Money Stock	(Billions of Dollars SA)
M2	5	M2 Money Stock	(Billions of Dollars SA)
RDI.T	5	Reserves Of Depository Institutions, Total	(Millions of Dollars SA)
RDI.N	5	Reserves Of Depository Institutions, Nonborrowed	(Millions of Dollars SA)
CIL	5	Commercial and Industrial Loans, All Commercial Banks	(Billions of Dollars SA)
MB.T	5	Monetary Base; Total	(Millions of Dollars SA)

Interest Rates:

IR.FF	2	Effective Federal Funds Rate	(Percent NSA)
IR.3MTB	2	3-Month Treasury Bill: Secondary Market Rate	(Percent NSA, Discount Basis)
IR.6MTB	2	6-Month Treasury Bill: Secondary Market Rate	(Percent NSA, Discount Basis)
IR.1TCM	2	1-Year Treasury Constant Maturity Rate	(Percent NSA)
IR.5TCM	2	5-Year Treasury Constant Maturity Rate	(Percent NSA)
IR.10TCM	2	10-Year Treasury Constant Maturity Rate	(Percent NSA)
TEDS	2	TED Spread	(Percent NSA)

Exchange rates and stock prices:

REPSG	S&P 500 Real Earnings Growth by Year	
RSPSG	S&P 500 Real Sales Growth by Quarter	
FE.Jap	Japan / U.S. Foreign Exchange Rate	(Japanese Yen to One U.S. Dollar NSA)
FE.Can	Canada / U.S. Foreign Exchange Rate	(Canadian Dollars to One U.S. Dollar NSA)
FE.Swit	Switzerland / U.S. Foreign Exchange Rate	(Swiss Francs to One U.S. Dollar NSA)
FE.Euro	U.S. / Euro Foreign Exchange Rate	(U.S. Dollars to One Euro NSA)

APPENDIX B

R-CODES

B.1 Dynamic Factor Models

```
2
  library("plsdof")
3 library("forecast")
4
5
  6
  #Using PCR with cross-validation
7
  Principal.component.regression.CUR <- pcr.cv(training.data.CUR.predictors,</pre>
8
                                            training.data.CUR.dependent.variable, k = 5)
9
10 Principal.component.regression.CUR$m.opt
11
12
  #Model adequacy
13
  PCR.CUR.intercept <- Principal.component.regression.CUR$intercept
  PCR.CUR.coefficients <- Principal.component.regression.CUR$coefficients
14
15 PCR.CUR.beta <- c(PCR.CUR.intercept, PCR.CUR.coefficients)
16 PCR.CUR <- cbind(rep(1, each = 31), training.data.CUR.predictors) %*% PCR.CUR.beta
17
18 residuals.PCR.CUR <- training.data.CUR.dependent.variable - PCR.CUR
19
20
  #Test for normal errors
21
  shapiro.test(residuals.PCR.CUR)
22
23
  #Fitted values
24 fit <-as.vector(PCR.CUR)
25
26 plot(residuals.PCR.CUR<sup>~</sup>fit,
       xlab = "", ylab = "")
27
28
  mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
29
  mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
30
  abline(h = 0, col = "red")
31
32
33 #Histogram of the residuals compared with the normal-distribution
34 h <- hist(residuals.PCR.CUR, xlab = "", ylab = "")
35 mtext("Index", side = 2, line = 2.2, cex = 1.5)
36
  mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
37
  xfit <- seq(min(residuals.PCR.CUR), max(residuals.PCR.CUR), length = 40)</pre>
38 yfit <- dnorm(xfit, mean = mean(residuals.PCR.CUR), sd = sd(residuals.PCR.CUR))
39 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.PCR.CUR)
40 lines(xfit, yfit, col = "blue", lwd = 2)
41
```

```
42 #Coefficient of determination
43 mean.CUR <- mean(training.data.CUR.dependent.variable)
44 pred <- sum((PCR.CUR - mean.CUR)^2)
45 real <- sum((training.data.CUR.dependent.variable - mean.CUR)^2)
46 R.squared.CUR <- pred / real
47
48 #Root mean squared error
49 RMSE(residuals.PCR.CUR)
50
51 #Mean squared prediction error
52 PCR.CUR.validation <- cbind(rep(1, each = 15), validation.data.CUR.predictors) %*% PCR.CUR.
       beta
53 mean((validation.data.CUR.dependent.variable - PCR.CUR.validation)^2)
54
55 #Plot of the real vs. the predicted unemplyment rate
56 real <- data.frame(validation.data.CUR.dependent.variable)
57 fc <- data.frame(PCR.CUR.validation)
58 Comp <- cbind(fc, real)
59 colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
60 Quarter <- seq(1, 15, by = 1)
61
62 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
63
                                                         y = Comp$Predicted.Unemployment.Rate,
64
                                                         color = 'Predicted.Unemployment.Rate'),
                                                               size = 1) +
65
    geom_line(data = real, aes(x = Quarter,
66
                                y = Comp$Real.Unemployment.Rate,
67
                                color = 'Real.Unemployment.Rate'), size = 1) +
   labs(x = "Quarter", y = "Rate") +
68
69
   theme(axis.title = element_text(face = "bold", size = 20)) +
70
   scale_color_manual(name = "", values = c("red", "blue"),
71
                        breaks = c("Predicted.Unemployment.Rate",
72
                                  "Real.Unemployment.Rate")) +
73
   theme(legend.text = element_text(size = 20))
74
75 #Forecast 2,5 year ahead
76 PCR.V <- as.vector(PCR.CUR)
77 PCR.TS <- ts(PCR.V)
78 forecast <- forecast(PCR.TS, h=10)
79 summary <-summary(forecast)
80 | forecast.PCR.CUR <- summary$'Point Forecast'
81
82 RMSE(forecast.data.CUR.dependent.variable - forecast.PCR.CUR)
83 MAE(forecast.data.CUR.dependent.variable - forecast.PCR.CUR)
84
85 #Plot of the real vs. the forecasted unemplyment rate
86 real <- data.frame(forecast.data.CUR.dependent.variable)
87 fc <- data.frame(forecast.PCR.CUR)
88 Comp <- cbind(fc, real)
89 colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
90 Quarter <- seq(1, 10, by = 1)
91
```

```
92
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
 93
                                          y = Comp$Forecasted.Unemployment.Rate,
 94
                                          color = 'Forecasted.Unemployment.Rate'), size = 1) +
 95
                               geom_line(data = real, aes(x = Quarter,
 96
                                          y = Comp$Real.Unemployment.Rate,
 97
                                          color = 'Real.Unemployment.Rate'), size = 1) +
 98
                               labs(x = "Quarter", y = "Rate") +
                               theme(axis.title = element_text(face = "bold", size = 20)) +
 99
100
                               scale_color_manual(name = "", values = c("red", "blue"),
101
                                                   breaks = c("Forecasted.Unemployment.Rate",
102
                                                              "Real.Unemployment.Rate")) +
103
                               theme(legend.text = element_text(size = 20))
104
105
107 #Using PCR with cross-validation
108 Principal.component.regression.Infl <- pcr.cv(training.data.Infl.predictors,
109
                                                 training.data.Infl.dependent.variable, k = 5)
110
111
   Principal.component.regression.Infl$m.opt
112
113 #Model adequacy
114 PCR.Infl.intercept <- Principal.component.regression.Infl$intercept
115 PCR.Infl.coefficients <- Principal.component.regression.Infl$coefficients
116 PCR.Infl.beta <- c(PCR.Infl.intercept,PCR.Infl.coefficients)
117
   PCR.Infl <- cbind(rep(1, each = 31), training.data.Infl.predictors) %*% PCR.Infl.beta
118
119
120 residuals.PCR.Infl <- training.data.Infl.dependent.variable - PCR.Infl
121
122 #Test for normal errors
123 shapiro.test(residuals.PCR.Infl)
124
125 #Fitted values
   fit <-as.vector(PCR.Infl)
126
127
128 plot(residuals.PCR.Infl~fit,
129
        xlab = "", ylab = "")
130 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
131 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
132 abline(h = 0, col = "red")
133
134
   #Histogram of the residuals compared with the normal-distribution
135 h <- hist(residuals.PCR.Infl, xlab = "", ylab = "")
136 mtext("Index", side = 2, line = 2.2, cex = 1.5)
137 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
138 xfit <- seq(min(residuals.PCR.Infl), max(residuals.PCR.Infl), length = 40)
139 yfit <- dnorm(xfit, mean = mean(residuals.PCR.Infl), sd = sd(residuals.PCR.Infl))
140 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.PCR.Infl)
141 lines(xfit, yfit, col = "blue", lwd = 2)
142
143 #Coefficient of determination
```

```
144 mean.Infl <- mean(training.data.Infl.dependent.variable)
145 pred <- sum((PCR.Infl - mean.Infl)^2)
146 real <- sum((training.data.Infl.dependent.variable - mean.Infl)^2)
147
    R.squared.Infl <- real / pred
148
149 #Root mean squared error
150 RMSE(residuals.PCR.Infl)
151
152 #Mean squared prediction error
153 PCR.Infl.validation <- cbind(rep(1, each = 15), validation.data.Infl.predictors) %*%
154
                                                     PCR.Infl.beta
155
    mean((validation.data.Infl.dependent.variable - PCR.Infl.validation)^2)
156
157 #Plot of the real vs. the predicted unemplyment rate
158 real <- data.frame(validation.data.Infl.dependent.variable)
159 fc <- data.frame(PCR.Infl.validation)
160 Comp <- cbind(fc, real)
161 colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
    Quarter \leq seq(1, 15, by = 1)
162
163
164
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc,
                                            aes(x = Quarter, y = Comp$Predicted.Inflation.Rate,
165
166
                                                color = 'Predicted.Inflation.Rate'), size = 1) +
167
                                  geom_line(data = real, aes(x = Quarter,
168
                                                             y = Comp$Real.Inflation.Rate,
169
                                 color = 'Real.Inflation.Rate'), size = 1) +
170
                                  labs(x = "Quarter", y = "Rate") +
171
                                  theme(axis.title = element_text(face = "bold", size = 20)) +
172
                                  scale_color_manual(name = "", values = c("red", "blue"),
173
                                 breaks = c("Predicted.Inflation.Rate",
174
                                             "Real.Inflation.Rate")) +
175
                                 theme(legend.text = element_text(size = 20))
176
177 #Forecast 2,5 year ahead
178 PCR.V <- as.vector(PCR.Infl)
179 PCR.TS <- ts(PCR.V)
180 forecast <- forecast(PCR.TS, h=10)
181 summary <-summary(forecast)
182 forecast.PCR.Infl <- summary$'Point Forecast'</pre>
183
184 RMSE(forecast.data.Infl.dependent.variable - forecast.PCR.Infl)
185
    MAE(forecast.data.Infl.dependent.variable - forecast.PCR.Infl)
186
187
    \# {\tt Plot} of the real vs. the forecasted unemplyment rate
188 real <- data.frame(forecast.data.Infl.dependent.variable)
189 fc <- data.frame(forecast.PCR.Infl)</pre>
190 Comp <- cbind(fc, real)
191 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
192 Quarter <- seq(1, 10, by = 1)
193
194 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
195
                                            y = Comp$Forecasted.Inflation.Rate,
```

```
196
                                          color = 'Forecasted.Inflation.Rate'), size = 1) +
197
                               geom_line(data = real, aes(x = Quarter,
198
                                          y = Comp$Real.Inflation.Rate,
199
                                          color = 'Real.Inflation.Rate'), size = 1) +
200
                               labs(x = "Quarter", y = "Rate") +
201
                               theme(axis.title = element_text(face = "bold", size = 20)) +
202
                               scale_color_manual(name = "", values = c("red", "blue"),
203
                                                   breaks = c("Forecasted.Inflation.Rate",
204
                                                              "Real.Inflation.Rate")) +
205
                               theme(legend.text = element_text(size = 20))
206
207
    208
   #Using PCR with cross-validation
209 Principal.component.regression.GDP <- pcr.cv(training.data.GDP.predictors,
210
                                                training.data.GDP.dependent.variable, k = 5)
211
212 Principal.component.regression.GDP$m.opt
213
214 #Model adequacy
215 PCR.GDP.intercept <- Principal.component.regression.GDP$intercept
216 PCR.GDP.coefficients <- Principal.component.regression.GDP$coefficients
217 PCR.GDP.beta <- c(PCR.GDP.intercept, PCR.GDP.coefficients)
218
219 PCR.GDP <- cbind(rep(1, each = 31), training.data.GDP.predictors) %*% PCR.GDP.beta
220
221
   residuals.PCR.GDP <- training.data.GDP.dependent.variable - PCR.GDP
222
223
    #Test for normal errors
224 shapiro.test(residuals.PCR.GDP)
225
226 #Fitted values
227 fit <-as.vector(PCR.GDP)
228
229 plot(residuals.PCR.GDP~fit,
230
        xlab = "", ylab = "")
231
   mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
232 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
233 abline(h = 0, col = "red")
234
235 #Histogram of the residuals compared with the normal-distribution
236 h <- hist(residuals.PCR.GDP, xlab = "", ylab = "")
237
   mtext("Index", side = 2, line = 2.2, cex = 1.5)
238
   mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
239 xfit <- seq(min(residuals.PCR.GDP), max(residuals.PCR.GDP), length = 40)
240 yfit <- dnorm(xfit, mean = mean(residuals.PCR.GDP), sd = sd(residuals.PCR.GDP))
241 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.PCR.GDP)
242 lines(xfit, yfit, col = "blue", lwd = 2)
243
244 #Coefficient of determination
245 mean.GDP <- mean(training.data.GDP.dependent.variable)
246 pred <- sum((PCR.GDP - mean.GDP)^2)
247 real <- sum((training.data.GDP.dependent.variable - mean.GDP)^2)
```

```
248 R.squared.GDP <- pred / real
249
250 #Root mean squared error
251
    RMSE(residuals.PCR.GDP)
252
253 #Mean squared prediction error
254 PCR.GDP.validation <- cbind(rep(1, each = 15), validation.data.GDP.predictors) %*%
255
                                                   PCR.GDP.beta
256 mean((validation.data.GDP.dependent.variable - PCR.GDP.validation)^2)
257
258 #Plot of the real vs. the predicted unemplyment rate
259
    real <- data.frame(validation.data.GDP.dependent.variable)</pre>
260 fc <- data.frame(PCR.GDP.validation)
261 Comp <- cbind(fc, real)
262 colnames(Comp) <- c("Predicted.GDP", "Real.GDP")
263 Quarter <- seq(1, 15, by = 1)
264
265
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Predicted.GDP,
266
                                                           color = 'Predicted.GDP'), size = 1) +
267
      geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
268
                                 color = 'Real.GDP'), size = 1) +
    labs(x = "Quarter", y = "Rate") +
269
270
    theme(axis.title = element_text(face = "bold", size = 20)) +
271
    scale_color_manual(name = "", values = c("red", "blue"),
272
                         breaks = c("Predicted.GDP", "Real.GDP")) +
273
    theme(legend.text = element_text(size = 20))
274
275
    #Forecast 2,5 year ahead
276 PCR.V <- as.vector(PCR.GDP)
277 PCR.TS <- ts(PCR.V)
278 forecast <- forecast(PCR.TS, h=10)
279 summary <-summary(forecast)
280 forecast.PCR.GDP <- summary$'Point Forecast'
281
282 RMSE(forecast.data.GDP.dependent.variable - forecast.PCR.GDP)
283 MAE(forecast.data.GDP.dependent.variable - forecast.PCR.GDP)
284
285 #Plot of the real vs. the forecasted unemplyment rate
286 real <- data.frame(forecast.data.GDP.dependent.variable)
287 fc <- data.frame(forecast.PCR.GDP)
288 Comp <- cbind(fc, real)
289
    colnames(Comp) <- c("Forecasted.GDP", "Real.GDP")</pre>
290
    Quarter <- seq(1, 10, by = 1)
291
292
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Forecasted.GDP,
293
                                           color = 'Forecasted.GDP'), size = 1) +
294
                                geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
295
                                           color = 'Real.GDP'), size = 1) +
                                labs(x = "Quarter", y = "Rate") +
296
297
                                theme(axis.title = element_text(face = "bold", size = 20)) +
298
                                scale_color_manual(name = "", values = c("red", "blue"),
299
                                                    breaks = c("Forecasted.GDP", "Real.GDP")) +
```

300

theme(legend.text = element_text(size = 20))

R-koder/PCR.R

B.2 Shrinkage Methods

```
2 library("parcor")
3 library("ggplot2")
4
5
  \#\mathbb{A} function mate by ... which creates the optimal lambda grid using the degrees off reedom
6
  opt_lambda_ridge <- function(X , multiple=1){</pre>
7
  s = svd(X)
8
  # the diagional elements d_j.
9 # Note that dj[1] is the largest value, while dj[end] is the smallest
10 dj = s$d
11 p = dim(X)[2]
12
13 # Do degrees_of_freedom = p first
14
  lambdas = 0
15
16 # Do all other values for the degrees_of_freedom next:
17 kRange = seq(p - 1, 1, by = (-1 / multiple))
18 for(ki in 1:length(kRange)){
19
   # solve for lambda in (via newton iterations):
   # k = \sum_{i=1}^p \frac{ d_j^2 }{ d_j^2 + \lambda }
20
21
    k = kRange[ki]
22
23
    # intialGuess at the root
24
   if(ki==1){
25
     xn = 0.0
26
    }else{
27
      xn = xnp1 # use the oldest previously computed root
28
    }
29
30
        = sum(dj^2 / (dj^2 + xn)) - k \# do the first update by hand
    f
31
    fp = - sum(dj^2 / (dj^2 + xn)^2)
32
    xnp1 = xn - f/fp
33
    while( abs(xn - xnp1) / abs(xn) > 10^{(-3)})
34
35
      xn = xnp1
36
      f
           = sum(dj^2 / (dj^2 + xn)) - k
37
          = - sum(dj^2 / (dj^2 + xn)^2)
      fp
38
      xnp1 = xn - f/fp
39
    }
40
41
    lambdas = c(lambdas,xnp1)
42 }
43
  # flip the order of the lambdas:
44 lambdas = lambdas[rev(1:length(lambdas))]
```

```
45 return(lambdas)
46 }
47
48
   49 #Creating the optimal lambda grid for the unemployment rate
50 optimal.lambdas.CUR <- opt_lambda_ridge(training.data.CUR.predictors, multiple = 1)
51
52 #Doing ridge regression for all the lambda values and using cross validation
53 #to determine the best lamba value.
54 ridge.regression.CUR <- ridge.cv(training.data.CUR.predictors,
55
                                    training.data.CUR.dependent.variable,
56
                                    optimal.lambdas.CUR, k = 5, scale = FALSE)
57
58 | lambda.CUR <- ridge.regression.CUR$lambda.opt
59
60 #Model adequacy
61 r.r.CUR.intercept <- ridge.regression.CUR$intercept
62 r.r.CUR.coefficients <- ridge.regression.CUR$coefficients
63 r.r.CUR.beta <- as.vector(c(r.r.CUR.intercept,r.r.CUR.coefficients))
64
65 ridge.regression.CUR <- cbind(rep(1, each = 31),
66
                                training.data.CUR.predictors) %*% r.r.CUR.beta
67
68 residuals.ridge.regression.CUR <- training.data.CUR.dependent.variable -
69
                                     ridge.regression.CUR
70
71
   #Test for normal errors
72
   shapiro.test(residuals.ridge.regression.CUR)
73
74 #Fitted values
75 fit <-as.vector(ridge.regression.CUR)
76
77 \left| \mbox{plot(residuals.ridge.regression.CUR~fit,} \right.
78
      xlab = "", ylab = "")
79 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
80 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
81 abline(h = 0, col = "red")
82
83 #Histogram of the residuals compared with the normal-distribution
84 h <- hist(residuals.ridge.regression.CUR, xlab = "", ylab = "")
85 mtext("Index", side = 2, line = 2.2, cex = 1.5)
   mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
86
87
   xfit <- seq(min(residuals.ridge.regression.CUR), max(residuals.ridge.regression.CUR),</pre>
88
              length = 40)
89 yfit <- dnorm(xfit, mean = mean(residuals.ridge.regression.CUR),
90
                sd = sd(residuals.ridge.regression.CUR))
91 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.ridge.regression.CUR)
92 lines(xfit, yfit, col = "blue", lwd = 2)
93
94 #Coefficient of determination
95 mean.CUR <- mean(training.data.CUR.dependent.variable)
96 pred <- sum((ridge.regression.CUR - mean.CUR)^2)
```

```
97 real <- sum((training.data.CUR.dependent.variable - mean.CUR)^2)
98 R.squared.CUR <- pred / real
99
100 #Root mean squared error
101 RMSE <- function(error){
     sqrt(mean(error^2))
102
103 }
104 RMSE(residuals.ridge.regression.CUR)
105
106 #Mean squared prediction error
107
   ridge.regression.CUR.validation <- cbind(rep(1, each = 15),
108
                                              validation.data.CUR.predictors) %*% r.r.CUR.beta
109 mean((validation.data.CUR.dependent.variable - ridge.regression.CUR.validation)^2)
110
111 #Plot of the real vs. the predicted unemplyment rate
112 real <- data.frame(validation.data.CUR.dependent.variable)
113 fc <- data.frame(ridge.regression.CUR.validation)
114 Comp <- cbind(fc, real)</pre>
115
   colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
116 Quarter <- seq(1, 15, by = 1)
117
118 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
119
                                                           y = Comp$Predicted.Unemployment.Rate,
120
                                                           color = 'Predicted.Unemployment.Rate'),
                                                                 size = 1) +
121
                                  geom_line(data = real, aes(x = Quarter,
122
                                                             y = Comp$Real.Unemployment.Rate,
123
                                                             color = 'Real.Unemployment.Rate'),
124
                                                             size = 1) +
125
                                  labs(x = "Quarter", y = "Rate") +
126
                                  theme(axis.title = element_text(face = "bold", size = 20)) +
127
                                  scale_color_manual(name = "", values = c("red", "blue"),
                                  breaks = c("Predicted.Unemployment.Rate",
128
129
                                             "Real.Unemployment.Rate")) +
130
                                  theme(legend.text = element_text(size = 20))
131
132
133 #Forecast 2,5 year ahead
134 ridge.regression.V <- as.vector(ridge.regression.CUR)
135 ridge.regression.TS <- ts(ridge.regression.V)
136 forecast <- forecast(ridge.regression.TS, h=10)
137
    summary <-summary(forecast)</pre>
138
   forecast.ridge.regression.CUR <- summary$'Point Forecast'</pre>
139 RMSE(forecast.data.CUR.dependent.variable - forecast.ridge.regression.CUR)
140
141 MAE <- function(error){
142
   mean(abs(error))
143 }
144 MAE(forecast.data.CUR.dependent.variable - forecast.ridge.regression.CUR)
145
146 #Plot of the real vs. the forecasted unemplyment rate
147 real <- data.frame(forecast.data.CUR.dependent.variable)
```

```
148 fc <- data.frame(forecast.ridge.regression.CUR)
149 Comp <- cbind(fc, real)
150
    colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")</pre>
151
    Quarter <- seq(1, 10, by = 1)
152
153
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
154
                                           y = Comp$Forecasted.Unemployment.Rate,
155
                                           color = 'Forecasted.Unemployment.Rate'), size = 1) +
156
                                 geom_line(data = real, aes(x = Quarter,
157
                                           y = Comp$Real.Unemployment.Rate,
158
                                           color = 'Real.Unemployment.Rate'), size = 1) +
159
                                labs(x = "Quarter", y = "Rate") +
160
                                theme(axis.title = element_text(face = "bold", size = 20)) +
161
                                scale_color_manual(name = "", values = c("red", "blue"),
162
                                                    breaks = c("Forecasted.Unemployment.Rate",
163
                                                               "Real.Unemployment.Rate")) +
164
                                theme(legend.text = element_text(size = 20))
165
166
    167
    #Creating the optimal lambda grid for the inflation rate
168 optimal.lambdas.Infl <- opt_lambda_ridge(training.data.Infl.predictors, multiple = 1)
169
170 #Doing ridge regression for all the lambda values and using cross validation
171 #to determine the best lamba value and the beste ridge regression model.
172 ridge.regression.Infl <- ridge.cv(training.data.Infl.predictors,
173
                                      training.data.Infl.dependent.variable,
174
                                      optimal.lambdas.Infl, k = 5, scale = FALSE)
175
176 lambda.Infl <- ridge.regression.Infl$lambda.opt
177
178 #Model adequacy
179 r.r.Infl.intercept <- ridge.regression.Infl$intercept
180 r.r.Infl.coefficients <- ridge.regression.Infl$coefficients
181 r.r.Infl.beta <- c(r.r.Infl.intercept, r.r.Infl.coefficients)</pre>
182
    ridge.regression.Infl <- cbind(rep(1, each = 31),</pre>
183
                                   training.data.Infl.predictors) %*% r.r.Infl.beta
184
185 residuals.ridge.regression.Infl <- training.data.Infl.dependent.variable -
186
                                       ridge.regression.Infl
187
    #Test for normal errors
188
189
    shapiro.test(residuals.PCR.CUR)
190
191
    #Fitted values
192 fit <-as.vector(ridge.regression.Infl)
193
194 plot(residuals.ridge.regression.Infl~fit,
195
        xlab = "", ylab = "")
196 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
197 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
198 abline(h = 0, col = "red")
199
```

```
200 #Histogram of the residuals compared with the normal-distribution
201 h <- hist(residuals.ridge.regression.Infl, xlab = "", ylab = "")
202 mtext("Index", side = 2, line = 2.2, cex = 1.5)
203
    mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
204 xfit <- seq(min(residuals.ridge.regression.Infl), max(residuals.ridge.regression.Infl),
205
                length = 40)
206 yfit < -dnorm(xfit, mean = mean(residuals.ridge.regression.Infl),
207
                  sd = sd(residuals.ridge.regression.Infl))
208 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.ridge.regression.Infl)
209 lines(xfit, yfit, col = "blue", lwd = 2)
210
211
   #Coefficient of determination
212 mean.Infl <- mean(training.data.Infl.dependent.variable)
213 pred <- sum((ridge.regression.Infl - mean.Infl)^2)
214 real <- sum((training.data.Infl.dependent.variable - mean.Infl)^2)
215 R.squared.Infl <- real / pred
216
217 #Root mean squared error
218 RMSE(residuals.ridge.regression.Infl)
219
220 #Mean squared prediction error
221 ridge.regression.Infl.validation <- cbind(rep(1, each = 15),
222
                                               validation.data.Infl.predictors) %*%
223
                                               r.r.Infl.beta
224 mean((validation.data.Infl.dependent.variable - ridge.regression.Infl.validation)^2)
225
226 #Plot of the real vs. the predicted unemplyment rate
227
    real <- data.frame(validation.data.Infl.dependent.variable)</pre>
228 fc <- data.frame(ridge.regression.Infl.validation)
229 Comp <- cbind(fc, real)
230 colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
231 Quarter <- seq(1, 15, by = 1)
232
233 ggplot(Comp, aes(Quarter)) + geom_line(data = fc,
234
                                            aes(x = Quarter, y = Comp$Predicted.Inflation.Rate,
235
                                                color = 'Predicted.Inflation.Rate'), size = 1) +
236
                                  geom_line(data = real, aes(x = Quarter,
237
                                                             y = Comp$Real.Inflation.Rate,
238
                                  color = 'Real.Inflation.Rate'), size = 1) +
239
                                  labs(x = "Quarter", y = "Rate") +
240
                                  theme(axis.title = element_text(face = "bold", size = 20)) +
241
                                  scale_color_manual(name = "", values = c("red", "blue"),
242
                                                     breaks = c("Predicted.Inflation.Rate",
243
                                                                "Real.Inflation.Rate")) +
244
                                  theme(legend.text = element_text(size = 20))
245
246 #Forecast 2,5 year ahead
247 ridge.regression.V <- as.vector(ridge.regression.Infl)
248 ridge.regression.TS <- ts(ridge.regression.V)
249 forecast <- forecast(ridge.regression.TS, h=10)
250
   summary <-summary(forecast)</pre>
251 forecast.ridge.regression.Infl <- summary$'Point Forecast'
```

```
252
253 RMSE(forecast.data.Infl.dependent.variable - forecast.ridge.regression.Infl)
254
    MAE(forecast.data.Infl.dependent.variable - forecast.ridge.regression.Infl)
255
256
    #Plot of the real vs. the forecasted unemplyment rate
257 real <- data.frame(forecast.data.Infl.dependent.variable)
258 fc <- data.frame(forecast.ridge.regression.Infl)
259 Comp <- cbind(fc, real)
260
    colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")</pre>
    Quarter <- seq(1, 10, by = 1)
261
262
263
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
264
                                           y = Comp$Forecasted.Inflation.Rate,
265
                                           color = 'Forecasted.Inflation.Rate'), size = 1) +
266
                                geom_line(data = real, aes(x = Quarter,
267
                                           y = Comp$Real.Inflation.Rate,
                                           color = 'Real.Inflation.Rate'), size = 1) +
268
                                labs(x = "Quarter", y = "Rate") +
269
270
                                theme(axis.title = element_text(face = "bold", size = 20))+
271
                                scale_color_manual(name = "", values = c("red", "blue"),
                                                    breaks = c("Forecasted.Inflation.Rate",
272
273
                                                               "Real.Inflation.Rate")) +
274
                                theme(legend.text = element_text(size = 20))
275
276
    277
    #Creating the optimal lambda grid for the gross domestic product
278
    optimal.lambdas.GDP <- opt_lambda_ridge(training.data.GDP.predictors, multiple = 1)</pre>
279
280
    #Doing ridge regression for all the lambda values and using cross validation
281
    #to determine the best lamba value and the beste ridge regression model.
    ridge.regression.GDP <- ridge.cv(training.data.GDP.predictors,</pre>
282
283
                                     training.data.GDP.dependent.variable,
284
                                     optimal.lambdas.GDP, k = 5, scale = FALSE)
285
286
    optimal.lambdas.GDP <- ridge.regression.GDP$lambda.opt</pre>
287
288 #Model adequacy
289 r.r.GDP.intercept <- ridge.regression.GDP$intercept
290 r.r.GDP.coefficients <- ridge.regression.GDP$coefficients
291 r.r.GDP.beta <- c(r.r.GDP.intercept, r.r.GDP.coefficients)
    ridge.regression.GDP <- cbind(rep(1, each = 31),</pre>
292
293
                                  training.data.GDP.predictors) %*% r.r.GDP.beta
294
    residuals.ridge.regression.GDP <- training.data.GDP.dependent.variable -</pre>
295
296
                                      ridge.regression.GDP
297
298 #Test for normal errors
299 shapiro.test(residuals.ridge.regression.GDP)
300
301 #Fitted values
302 fit <-as.vector(ridge.regression.GDP)
303
```

```
304 plot(residuals.ridge.regression.GDP~fit,
        xlab = "", ylab = "")
305
306 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
307
    mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
308 abline(h = 0, col = "red")
309
310 #Histogram of the residuals compared with the normal-distribution
311 h <- hist(residuals.ridge.regression.GDP, xlab = "", ylab = "")
312 mtext("Index", side = 2, line = 2.2, cex = 1.5)
313 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
314 xfit <- seq(min(residuals.ridge.regression.GDP), max(residuals.ridge.regression.GDP),
315
                length = 40)
316 yfit <- dnorm(xfit, mean = mean(residuals.ridge.regression.GDP),
317
                  sd = sd(residuals.ridge.regression.GDP))
318 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.ridge.regression.GDP)
319 lines(xfit, yfit, col = "blue", lwd = 2)
320
321 #Coefficient of determination
322 mean.GDP <- mean(training.data.GDP.dependent.variable)
323 pred <- sum((ridge.regression.GDP - mean.GDP)^2)
324 real <- sum((training.data.GDP.dependent.variable - mean.GDP)^2)
325 R.squared.GDP <- pred / real
326
327 #Root mean squared error
328 RMSE(residuals.ridge.regression.GDP)
329
330 #Mean squared prediction error
331
   ridge.regression.GDP.validation <- cbind(rep(1, each = 15),
332
                                              validation.data.GDP.predictors) %*% r.r.GDP.beta
333 mean((validation.data.GDP.dependent.variable - ridge.regression.GDP.validation)^2)
334
335 #Plot of the real vs. the predicted unemplyment rate
336 real <- data.frame(validation.data.GDP.dependent.variable)</pre>
337 fc <- data.frame(ridge.regression.GDP.validation)</pre>
338
   Comp <- cbind(fc, real)</pre>
339
   colnames(Comp) <- c("Predicted.GDP", "Real.GDP")</pre>
340 Quarter <- seq(1, 15, by = 1)
341
342 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Predicted.GDP,
                                                           color = 'Predicted.GDP'), size = 1) +
343
344
      geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
345
                                  color = 'Real.GDP'), size = 1) +
346
     labs(x = "Quarter", y = "Rate") +
     theme(axis.title = element_text(face = "bold", size = 20)) +
347
348
     scale_color_manual(name = "", values = c("red", "blue"),
349
                         breaks = c("Predicted.GDP", "Real.GDP")) +
350
     theme(legend.text = element_text(size = 20))
351
352 #Forecast 2,5 year ahead
353 ridge.regression.V <- as.vector(ridge.regression.GDP)
354 ridge.regression.TS <- ts(ridge.regression.V)
355 forecast <- forecast(ridge.regression.TS, h=10)
```

```
356 summary <-summary(forecast)
357 forecast.ridge.regression.GDP <- summary$'Point Forecast'
358
359
    RMSE(forecast.data.GDP.dependent.variable - forecast.ridge.regression.GDP)
360 MAE(forecast.data.GDP.dependent.variable - forecast.ridge.regression.GDP)
361
362 #Plot of the real vs. the forecasted unemplyment rate
363 real <- data.frame(forecast.data.GDP.dependent.variable)
364 fc <- data.frame(forecast.ridge.regression.GDP)
365 Comp <- cbind(fc, real)
366
    colnames(Comp) <- c("Forecasted.GDP", "Real.GDP")</pre>
    Quarter <- seq(1, 10, by = 1)
367
368
369
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Forecasted.GDP,
370
                                           color = 'Forecasted.GDP'), size = 1) +
371
                                geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
372
                                           color = 'Real.GDP'), size = 1) +
                                labs(x = "Quarter", y = "Rate") +
373
374
                                theme(axis.title = element_text(face = "bold", size = 20)) +
375
                                scale_color_manual(name = "", values = c("red", "blue"),
376
                                                     breaks = c("Forecasted.GDP", "Real.GDP")) +
377
                                theme(legend.text = element_text(size = 20))
```

R-koder/RidgeRegression.R

```
1
2
  library("glmnet")
3
  library("forecast")
4
5
  6 | lasso.CUR <- cv.glmnet(training.data.CUR.predictors, training.data.CUR.dependent.variable,
7
                     k =5, alpha = 1, type.measure = "mse")
8
  plot(lasso.CUR,
      xlab = "", ylab = "")
9
10
  mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
  mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
11
12 abline(h = 0, col = "red")
13
14 best.lambda.CUR <- lasso.CUR$lambda.min
15
16 #Model adequacy
17 1.CUR.coefficients <- coef(lasso.CUR)
18 lasso.CUR <- cbind(rep(1, each = 31), training.data.CUR.predictors) %*% 1.CUR.coefficients
19
20 residuals.lasso.CUR <- as.matrix(training.data.CUR.dependent.variable - lasso.CUR)
21
22 #Test for normal errors
23 shapiro.test(residuals.lasso.CUR)
24
25 #Fitted values
26 fit <-as.vector(lasso.CUR)
27
```

```
28 plot(residuals.lasso.CUR<sup>~</sup>fit,
        xlab = "", ylab = "")
29
30 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
31
   mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
32
   abline(h = 0, col = "red")
33
34 #Histogram of the residuals compared with the normal-distribution
35 h <- hist(residuals.lasso.CUR, xlab = "", ylab = "")
36 mtext("Index", side = 2, line = 2.2, cex = 1.5)
37 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
38 xfit <- seq(min(residuals.lasso.CUR), max(residuals.lasso.CUR), length = 40)
39
   yfit <- dnorm(xfit, mean = mean(residuals.lasso.CUR), sd = sd(residuals.lasso.CUR))</pre>
40
  yfit <- yfit * diff(h$mids[1:2]) * length(residuals.lasso.CUR)</pre>
41 lines(xfit, yfit, col = "blue", lwd = 2)
42
43 #Coefficient of determination
44 mean.CUR <- mean(training.data.CUR.dependent.variable)
45 pred <- sum((lasso.CUR-mean.CUR)^2)
46
  real <- sum((training.data.CUR.dependent.variable - mean.CUR)^2)
47
   R.squared.CUR <- pred / real
48
49 #Root mean squared error
50 RMSE(residuals.lasso.CUR)
51
52 #Mean squared prediction error
53
  lasso.CUR.validation <- cbind(rep(1, each = 15),validation.data.CUR.predictors) %*%
54
                           1.CUR.coefficients
55
  mean((validation.data.CUR.dependent.variable - lasso.CUR.validation)^2)
56
57
  #Plot of the real vs. the predicted unemplyment rate
58 real <- data.frame(validation.data.CUR.dependent.variable)
59 fc <- as.vector(lasso.CUR.validation)
60 fc <- data.frame(fc)
61
  Comp <- cbind(fc, real)</pre>
62
   colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")</pre>
63
  Quarter <- seq(1, 15, by = 1)
64
65
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
66
                                                           y = Comp$Predicted.Unemployment.Rate,
67
                                                           color = 'Predicted.Unemployment.Rate'),
                                                                 size = 1) +
     geom_line(data = real, aes(x = Quarter,
68
69
                                 y = Comp$Real.Unemployment.Rate,
70
                                 color = 'Real.Unemployment.Rate'),
71
               size = 1) +
72
    labs(x = "Quarter", y = "Rate") +
73
     theme(axis.title = element_text(face = "bold", size = 20)) +
74
     scale_color_manual(name = "", values = c("red", "blue"),
75
                         breaks = c("Predicted.Unemployment.Rate",
76
                                    "Real.Unemployment.Rate")) +
77
     theme(legend.text = element_text(size = 20))
78
```

```
79 #Forecast 2,5 year ahead
 80 lasso.V <- as.vector(lasso.CUR)
 81 lasso.TS <- ts(lasso.V)
 82 forecast <- forecast(lasso.TS, h=10)
 83 summary <-summary(forecast)
 84 forecast.lasso.CUR <- summary$'Point Forecast'
 85
 86 RMSE(forecast.data.CUR.dependent.variable - forecast.lasso.CUR)
 87 MAE(forecast.data.CUR.dependent.variable - forecast.lasso.CUR)
 88
 89 #Plot of the real vs. the forecasted unemplyment rate
 90
    real <- data.frame(forecast.data.CUR.dependent.variable)</pre>
 91 fc <- data.frame(forecast.lasso.CUR)
 92 Comp <- cbind(fc, real)
 93 colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
 94 Quarter <- seq(1, 10, by = 1)
 95
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
 96
 97
                                          y = Comp$Forecasted.Unemployment.Rate,
 98
                                          color = 'Forecasted.Unemployment.Rate'), size = 1) +
                              geom_line(data = real, aes(x = Quarter,
 99
100
                                          y = Comp$Real.Unemployment.Rate,
101
                                          color = 'Real.Unemployment.Rate'), size = 1) +
102
                              labs(x = "Quarter", y = "Rate") +
103
                              theme(axis.title = element_text(face = "bold", size = 20))+
104
                              scale_color_manual(name = "", values = c("red", "blue"),
105
                                                   breaks = c("Forecasted.Unemployment.Rate",
106
                                                              "Real.Unemployment.Rate"))+
107
                              theme(legend.text = element_text(size = 20))
108
109
111 lasso.Infl <- cv.glmnet(training.data.Infl.predictors,
112
                           training.data.Infl.dependent.variable,
113
                           k = 5, alpha = 1, type.measure = "mse")
114 plot(lasso.Infl,
        xlab = "", ylab = "")
115
116 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
117 mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
118 abline(h = 0, col = "red")
119
120 best.lambda.Infl <- lasso.Infl$lambda.min
121
122 #Model adequacy
123 1.Infl.coefficients <- coef(lasso.Infl)
124 lasso.Infl <- cbind(rep(1, each = 31),
125
                       training.data.Infl.predictors) %*% l.Infl.coefficients
126
127 residuals.lasso.Infl <- as.matrix(training.data.Infl.dependent.variable - lasso.Infl)
128
129 #Test for normal errors
130 shapiro.test(residuals.lasso.Infl)
```

```
131
132 #Fitted values
133 fit <-as.vector(lasso.Infl)
134
135 plot(residuals.lasso.Infl~fit,
        xlab = "", ylab = "")
136
137 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
138 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
139 abline(h = 0, col = "red")
140
141 #Histogram of the residuals compared with the normal-distribution
   h <- hist(residuals.lasso.Infl, xlab = "", ylab = "")</pre>
142
143 mtext("Index", side = 2, line = 2.2, cex = 1.5)
144 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
145 xfit < -seq(min(residuals.lasso.Infl), max(residuals.lasso.Infl), length = 40)
146 yfit <- dnorm(xfit, mean = mean(residuals.lasso.Infl), sd = sd(residuals.lasso.Infl))
147 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.lasso.Infl)
148 lines(xfit, yfit, col = "blue", lwd = 2)
149
150 #Coefficient of determination
151 mean.Infl <- mean(training.data.Infl.dependent.variable)
152 pred <- sum((lasso.Infl-mean.Infl)^2)
153 real <- sum((training.data.Infl.dependent.variable - mean.Infl)^2)
154 R.squared.Infl <- real/pred
155
156 #Root mean squared error
157
   RMSE(residuals.lasso.Infl)
158
159 #Mean squared prediction error
160 lasso.Infl.validation <- cbind(rep(1, each = 15),</pre>
                                   validation.data.Infl.predictors) %*% l.Infl.coefficients
161
162 mean((validation.data.Infl.dependent.variable - lasso.Infl.validation)^2)
163
164 #Mean squared prediction error
165 ridge.regression.Infl.validation <- cbind(rep(1, each = 15),</pre>
166
                                               validation.data.Infl.predictors) %*%
167
    r.r.Infl.beta
168 mean((validation.data.Infl.dependent.variable - ridge.regression.Infl.validation)^2)
169
170 #Plot of the real vs. the predicted unemplyment rate
171 real <- data.frame(validation.data.Infl.dependent.variable)
172 fc <- as.vector(lasso.Infl.validation)
173
   fc <- data.frame(fc)</pre>
174 Comp <- cbind(fc, real)
175 | colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
176 Quarter <- seq(1, 15, by = 1)
177
178 ggplot(Comp, aes(Quarter)) + geom_line(data = fc,
179
                                            aes(x = Quarter, y = Comp$Predicted.Inflation.Rate,
180
                                                color = 'Predicted.Inflation.Rate'), size = 1) +
181
     geom_line(data = real, aes(x = Quarter,
182
                                 y = Comp$Real.Inflation.Rate,
```

```
183
                                color = 'Real.Inflation.Rate'), size = 1) +
     labs(x = "Quarter", y = "Rate") +
184
185
      theme(axis.title = element_text(face = "bold", size = 20)) +
186
      scale_color_manual(name = "", values = c("red", "blue"),
187
                        breaks = c("Predicted.Inflation.Rate",
188
                                   "Real.Inflation.Rate")) +
189
     theme(legend.text = element_text(size = 20))
190
191 #Creating the forecast
192 #Forecast 2,5 year ahead
193 lasso.V <- as.vector(lasso.Infl)
194
    lasso.TS <- ts(lasso.V)</pre>
195 forecast <- forecast(lasso.TS, h=10)
196 summary <-summary(forecast)
197 forecast.lasso.Infl <- summary$'Point Forecast'
198
199 RMSE(forecast.data.Infl.dependent.variable - forecast.lasso.Infl)
200 MAE(forecast.data.Infl.dependent.variable - forecast.lasso.Infl)
201
202 #Plot of the real vs. the forecasted unemplyment rate
203 real <- data.frame(forecast.data.Infl.dependent.variable)
204 fc <- data.frame(forecast.lasso.Infl)
205 Comp <- cbind(fc, real)
206 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
207
    Quarter <- seq(1, 10, by = 1)
208
209
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
210
                                          y = Comp$Forecasted.Inflation.Rate,
211
                                          color = 'Forecasted.Inflation.Rate'), size = 1) +
212
                               geom_line(data = real, aes(x = Quarter,
213
                                          y = Comp$Real.Inflation.Rate,
214
                                          color = 'Real.Inflation.Rate'), size = 1) +
                               labs(x = "Quarter", y = "Rate") +
215
216
                               theme(axis.title = element_text(face = "bold", size = 20)) +
217
                                scale_color_manual(name = "", values = c("red", "blue"),
218
                                                   breaks = c("Forecasted.Inflation.Rate",
219
                                                   "Real.Inflation.Rate")) +
220
                               theme(legend.text = element_text(size = 20))
221
222
224
    lasso.GDP <- cv.glmnet(training.data.GDP.predictors, training.data.GDP.dependent.variable,</pre>
225
                          k = 5, alpha = 1, type.measure = "mse")
226 plot(lasso.GDP,
        xlab = "", ylab = "")
227
228 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
229 mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
230 abline(h = 0, col = "red")
231 best.lambda.GDP <- lasso.GDP$lambda.min
232
233 #Model adequacy
234 1.GDP.coefficients <- coef(lasso.GDP)
```
```
235 |lasso.GDP <- cbind(rep(1, each = 31),training.data.GDP.predictors) %*% 1.GDP.coefficients
236
237
   residuals.lasso.GDP <- as.matrix(training.data.GDP.dependent.variable - lasso.GDP)
238
239
   #Test for normal errors
240 shapiro.test(residuals.lasso.GDP)
241
242 #Fitted values
243 fit <-as.vector(lasso.GDP)
244
245
   plot(residuals.lasso.GDP~fit,
        xlab = "", ylab = "")
246
247
   mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
248 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
249 abline(h = 0, col = "red")
250
251 #Histogram of the residuals compared with the normal-distribution
252 h <- hist(residuals.lasso.GDP, xlab = "", ylab = "")
253 mtext("Index", side = 2, line = 2.2, cex = 1.5)
254
   mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
255 xfit <- seq(min(residuals.lasso.GDP), max(residuals.lasso.GDP), length = 40)
256 yfit <- dnorm(xfit, mean =mean(residuals.lasso.GDP), sd = sd(residuals.lasso.GDP))
257 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.lasso.GDP)
258 lines(xfit, yfit, col = "blue", lwd = 2)
259
260 #Coefficient of determination
    mean.GDP <- mean(training.data.GDP.dependent.variable)</pre>
261
262 pred <- sum((lasso.GDP-mean.GDP)^2)
263 real <- sum((training.data.GDP.dependent.variable - mean.GDP)^2)
264 R.squared.GDP <- pred / real
265
266 #Root mean squared error
267 RMSE(residuals.lasso.GDP)
268
269 #Mean squared prediction error
270 | lasso.GDP.validation <- cbind(rep(1, each = 15), validation.data.GDP.predictors) %*%
271
                                  l.GDP.coefficients
272 mean((validation.data.GDP.dependent.variable-lasso.GDP.validation)^2)
273
274 #Plot of the real vs. the predicted unemplyment rate
275 real <- data.frame(validation.data.GDP.dependent.variable)
276 fc <- as.vector(lasso.GDP.validation)
277
   fc <- data.frame(fc)</pre>
278 Comp <- cbind(fc, real)
279 colnames(Comp) <- c("Predicted.GDP", "Real.GDP")
280 Quarter <- seq(1, 15, by = 1)
281
282 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Predicted.GDP,
                                                           color = 'Predicted.GDP'), size = 1) +
283
284
     geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
285
                                 color = 'Real.GDP'), size = 1) +
286
     labs(x = "Quarter", y = "Rate") +
```

```
287
     theme(axis.title = element_text(face = "bold", size = 20)) +
288
     scale_color_manual(name = "", values = c("red", "blue"),
                     breaks = c("Predicted.GDP", "Real.GDP")) +
289
290
     theme(legend.text = element_text(size = 20))
291
292 #Forecast 2,5 year ahead
293 lasso.V <- as.vector(lasso.GDP)
294 lasso.TS <- ts(lasso.V)
295 forecast <- forecast(lasso.TS, h=10)
296 summary <-summary(forecast)
297
   forecast.lasso.GDP <- summary$'Point Forecast'</pre>
298
299
   RMSE(forecast.data.GDP.dependent.variable - forecast.lasso.GDP)
300 MAE(forecast.data.GDP.dependent.variable - forecast.lasso.GDP)
301
302 #Plot of the real vs. the forecasted unemplyment rate
303 real <- data.frame(forecast.data.GDP.dependent.variable)
304 fc <- data.frame(forecast.lasso.GDP)
305 Comp <- cbind(fc, real)
306
   colnames(Comp) <- c("Forecasted.GDP", "Real.GDP")</pre>
   Quarter <- seq(1, 10, by = 1)
307
308
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Forecasted.GDP,
309
310
                                                  color = 'Forecasted.GDP'), size = 1) +
311
                           geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
312
                                    color = 'Real.GDP'), size = 1) +
313
                           labs(x = "Quarter", y = "Rate") +
314
                           theme(axis.title = element_text(face = "bold", size = 20)) +
                           scale_color_manual(name = "", values = c("red", "blue"),
315
316
                                           breaks = c("Forecasted.GDP", "Real.GDP")) +
317
                           theme(legend.text = element_text(size = 20))
318
319
320
   321
   library("grpreg")
322
   323
324
   325
326
                10,10,10,10)
327
328
   grouped.lasso.CUR <- cv.grpreg(training.data.CUR.predictors,</pre>
329
                              training.data.CUR.dependent.variable,
330
                              k = 5, group = group.CUR, penalty = "grLasso")
331
332 plot(grouped.lasso.CUR,
333
       xlab = "", ylab = "")
334 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
335 mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
336 abline(h = 0, col = "red")
337
338 best.grouped.lambda.CUR <- grouped.lasso.CUR$lambda.min
```

```
339
340 #Model adequacy
341 gl.CUR.coefficients <- coef(grouped.lasso.CUR)
342
    grouped.lasso.CUR <- cbind(rep(1, each = 31),</pre>
343
                               training.data.CUR.predictors) %*% gl.CUR.coefficients
344
345 residuals.grouped.lasso.CUR <- as.matrix(training.data.CUR.dependent.variable -
346
                                              grouped.lasso.CUR)
347 #Test for normal errors
348 shapiro.test(residuals.grouped.lasso.CUR)
349
350 #Fitted values
351 fit <-as.vector(grouped.lasso.CUR)
352
353 plot(residuals.grouped.lasso.CUR<sup>~</sup>fit,
        xlab = "", ylab = "")
354
355 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
356 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
357
   abline(h = 0, col = "red")
358
359 #Histogram of the residuals compared with the normal-distribution
360 h <- hist(residuals.grouped.lasso.CUR, xlab = "", ylab = "")
361 mtext("Index", side = 2, line = 2.2, cex = 1.5)
362 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
363 xfit < -seq(min(residuals.grouped.lasso.CUR), max(residuals.grouped.lasso.CUR), length = 40)
364
   yfit <- dnorm(xfit, mean = mean(residuals.grouped.lasso.CUR),</pre>
365
                  sd = sd(residuals.grouped.lasso.CUR))
366
   yfit <- yfit * diff(h$mids[1:2]) * length(residuals.grouped.lasso.CUR)</pre>
367
   lines(xfit, yfit, col = "blue", lwd = 2)
368
369 #Coefficient of determination
370 mean.CUR <- mean(training.data.CUR.dependent.variable)
371 pred <- sum((grouped.lasso.CUR-mean.CUR)^2)
372 real <- sum((training.data.CUR.dependent.variable - mean.CUR)^2)
373 R.squared.CUR <- pred / real
374
375 #Root mean squared error
376 RMSE(residuals.grouped.lasso.CUR)
377
378 #Mean squared prediction error
379
   grouped.lasso.CUR.validation <- cbind(rep(1, each = 15),</pre>
380
                                           validation.data.CUR.predictors) %*%
381
                                           gl.CUR.coefficients
382 mean((validation.data.CUR.dependent.variable-grouped.lasso.CUR.validation)^2)
383
384 #Plot of the real vs. the predicted unemplyment rate
385 real <- data.frame(validation.data.CUR.dependent.variable)
386 fc <- data.frame(grouped.lasso.CUR.validation)</pre>
387 Comp <- cbind(fc, real)
388 colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
389 Quarter <- seq(1, 15, by = 1)
390
```

```
391
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
392
                                                       y = Comp$Predicted.Unemployment.Rate,
393
                                                       color = 'Predicted.Unemployment.Rate'),
                                                            size = 1) +
394
     geom_line(data = real, aes(x = Quarter,
395
                               y = Comp$Real.Unemployment.Rate,
396
                               color = 'Real.Unemployment.Rate'),
397
               size = 1) +
398
     labs(x = "Quarter", y = "Rate") +
399
     theme(axis.title = element_text(face = "bold", size = 20)) +
      scale_color_manual(name = "", values = c("red", "blue"),
400
401
                       breaks = c("Predicted.Unemployment.Rate",
402
                                  "Real.Unemployment.Rate")) +
403
     theme(legend.text = element_text(size = 20))
404
405 #Forecast 2,5 year ahead
406 grouped.lasso.V <- as.vector(grouped.lasso.CUR)
   grouped.lasso.TS <- ts(grouped.lasso.V)</pre>
407
408
   forecast <- forecast(grouped.lasso.TS, h=10)</pre>
409
   summary <-summary(forecast)</pre>
410 forecast.grouped.lasso.CUR <- summary$'Point Forecast'
411
412 RMSE(forecast.data.CUR.dependent.variable - forecast.grouped.lasso.CUR)
413 MAE(forecast.data.CUR.dependent.variable - forecast.grouped.lasso.CUR)
414
415
   #Plot of the real vs. the forecasted unemplyment rate
416
   real <- data.frame(forecast.data.CUR.dependent.variable)</pre>
417
   fc <- data.frame(forecast.grouped.lasso.CUR)</pre>
418 Comp <- cbind(fc, real)
419 colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
   Quarter <- seq(1, 10, by = 1)
420
421
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
422
423
                                        y = Comp$Forecasted.Unemployment.Rate,
424
                                        color = 'Forecasted.Unemployment.Rate'), size = 1) +
425
                              geom_line(data = real, aes(x = Quarter,
                                        y = Comp$Real.Unemployment.Rate,
426
427
                                        color = 'Real.Unemployment.Rate'), size = 1) +
                              labs(x = "Quarter", y = "Rate") +
428
429
                              theme(axis.title = element_text(face = "bold", size = 20)) +
430
                              scale_color_manual(name = "", values = c("red", "blue"),
431
                                                 breaks = c("Forecasted.Unemployment.Rate",
432
                                                            "Real.Unemployment.Rate")) +
433
                              theme(legend.text = element_text(size = 20))
434
435
   436
   437
                   10,10,10,10,10)
438
439
440
   grouped.lasso.Infl <- cv.grpreg(training.data.Infl.predictors,</pre>
441
                                  training.data.Infl.dependent.variable,
```

```
442
                                     k = 5, group = group.Infl, penalty="grLasso")
443
444
   plot(grouped.lasso.Infl,
445
         xlab = "", ylab = "")
446 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
447
   mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
448 abline(h = 0, col = "red")
449
450 best.grouped.lambda.Infl <- grouped.lasso.Infl$lambda.min
451
452 #Model adequacy
453
    gl.Infl.coefficients <- coef(grouped.lasso.Infl)</pre>
    grouped.lasso.Infl <- cbind(rep(1, each = 31),</pre>
454
455
                                training.data.Infl.predictors) %*% gl.Infl.coefficients
456
457 residuals.grouped.lasso.Infl <- as.matrix(training.data.Infl.dependent.variable -
458
                                               grouped.lasso.Infl)
459
460 #Test for normal errors
461
    shapiro.test(residuals.grouped.lasso.Infl)
462
463 #Fitted values
464 fit <-as.vector(grouped.lasso.Infl)
465
466 plot(residuals.grouped.lasso.Infl~fit,
467
         xlab = "", ylab = "")
468
    mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
469
    mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
470 abline (h = 0, col = "red")
471
472 #Histogram of the residuals compared with the normal-distribution
473 h <- hist(residuals.grouped.lasso.Infl, xlab = "", ylab = "")
474 mtext("Index", side = 2, line = 2.2, cex = 1.5)
475 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
476 xfit < -seq(min(residuals.grouped.lasso.Infl), max(residuals.grouped.lasso.Infl),
477
                length = 40)
478 yfit <- dnorm(xfit, mean = mean(residuals.grouped.lasso.Infl),
479
                 sd = sd(residuals.grouped.lasso.Infl))
480 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.grouped.lasso.Infl)
481 lines(xfit, yfit, col = "blue", lwd = 2)
482
483 #Coefficient of determination
484
    mean.Infl <- mean(training.data.Infl.dependent.variable)</pre>
485 pred <- sum((grouped.lasso.Infl - mean.Infl)^2)
486 real <- sum((training.data.Infl.dependent.variable - mean.Infl)^2)
487 R.squared.Infl <- real / pred
488
489 #Root mean squared error
490 RMSE(residuals.grouped.lasso.Infl)
491
492 #Mean squared prediction error
493 grouped.lasso.Infl.validation <- cbind(rep(1, each = 15),
```

```
494
                                            validation.data.Infl.predictors) %*%
495
                                            gl.Infl.coefficients
496
    mean((validation.data.Infl.dependent.variable-grouped.lasso.Infl.validation)^2)
497
498
    #Plot of the real vs. the predicted unemplyment rate
499 real <- data.frame(validation.data.Infl.dependent.variable)
500 fc <- data.frame(grouped.lasso.Infl.validation)
501 Comp <- cbind(fc, real)
502
    colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")</pre>
    Quarter <- seq(1, 15, by = 1)
503
504
505
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc,
506
                                            aes(x = Quarter, y = Comp$Predicted.Inflation.Rate,
507
                                                color = 'Predicted.Inflation.Rate'), size = 1) +
508
      geom_line(data = real, aes(x = Quarter,
509
                                 y = Comp$Real.Inflation.Rate,
510
                                 color = 'Real.Inflation.Rate'), size = 1) +
511
      labs(x = "Quarter", y = "Rate") +
512
      theme(axis.title = element_text(face = "bold", size = 20)) +
513
      scale_color_manual(name = "", values = c("red", "blue"),
514
                         breaks = c("Predicted.Inflation.Rate".
515
                                    "Real.Inflation.Rate")) +
516
     theme(legend.text = element_text(size = 20))
517
518 #Creating the forecast
519
    #Forecast 2,5 year ahead
520
    grouped.lasso.V <- as.vector(grouped.lasso.Infl)</pre>
521
    grouped.lasso.TS <- ts(grouped.lasso.V)</pre>
522 forecast <- forecast(grouped.lasso.TS, h=10)
523 summary <-summary(forecast)
524 forecast.grouped.lasso.Infl <- summary$'Point Forecast'
525
526 RMSE(forecast.data.Infl.dependent.variable - forecast.grouped.lasso.Infl)
527 MAE(forecast.data.Infl.dependent.variable - forecast.grouped.lasso.Infl)
528
529 #Plot of the real vs. the forecasted unemplyment rate
530 real <- data.frame(forecast.data.Infl.dependent.variable)
531 fc <- data.frame(forecast.grouped.lasso.Infl)
532 Comp <- cbind(fc, real)
533 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
    Quarter <- seq(1, 10, by = 1)
534
535
536
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
537
                                            y = Comp$Forecasted.Inflation.Rate,
538
                                            color = 'Forecasted.Inflation.Rate'), size = 1) +
539
                                 geom_line(data = real, aes(x = Quarter,
540
                                            y = Comp$Real.Inflation.Rate,
541
                                            color = 'Real.Inflation.Rate'), size = 1) +
542
                                 labs(x = "Quarter", y = "Rate") +
543
                                 theme(axis.title = element_text(face = "bold", size = 20)) +
544
                                 scale_color_manual(name = "", values = c("red", "blue"),
545
                                                     breaks = c("Forecasted.Inflation.Rate",
```

546 "Real.Inflation.Rate")) + 547 theme(legend.text = element_text(size = 20)) 548 549 550 551 552 10, 10, 10, 10)553 554 grouped.lasso.GDP <- cv.grpreg(training.data.GDP.predictors, 555 training.data.GDP.dependent.variable, 556 k = 5, group = group.GDP, penalty = "grLasso") 557 558 plot(grouped.lasso.GDP, 559 xlab = "", ylab = "") 560 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5) 561 mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5) 562 abline(h = 0, col = "red") 563 564 best.grouped.lambda.GDP <- grouped.lasso.GDP\$lambda.min 565 566 #Model adequacy 567 gl.GDP.coefficients <- coef(grouped.lasso.GDP) 568 grouped.lasso.GDP <- cbind(rep(1, each = 31), 569 training.data.GDP.predictors) %*% gl.GDP.coefficients 570 571 residuals.grouped.lasso.GDP <- as.matrix(training.data.GDP.dependent.variable grouped.lasso.GDP) 572 573 #Test for normal errors 574 shapiro.test(residuals.grouped.lasso.GDP) 575 576 #Fitted values 577 fit <-as.vector(grouped.lasso.GDP) 578 579 plot(residuals.grouped.lasso.GDP~fit, 580 xlab = "", ylab = "") 581 mtext("Residuals", side = 2, line = 2.2, cex = 1.5) 582 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5) 583 abline(h = 0, col = "red") 584 585 #Histogram of the residuals compared with the normal-distribution 586 h <- hist(residuals.grouped.lasso.GDP, xlab = "", ylab = "") 587 mtext("Index", side = 2, line = 2.2, cex = 1.5) 588 mtext("Residuals", side = 1, line = 2.2, cex = 1.5) 589 xfit <- seq(min(residuals.grouped.lasso.GDP), max(residuals.grouped.lasso.GDP), length = 40) 590 yfit <- dnorm(xfit, mean = mean(residuals.grouped.lasso.GDP), 591 sd = sd(residuals.grouped.lasso.GDP)) 592 yfit <- yfit * diff(h\$mids[1:2]) * length(residuals.grouped.lasso.GDP) 593 lines(xfit, yfit, col = "blue", lwd = 2) 594 595 #Coefficient of determination 596 mean.GDP <- mean(training.data.GDP.dependent.variable) 597 pred <- sum((grouped.lasso.GDP-mean.GDP)^2)

```
598 real <- sum((training.data.GDP.dependent.variable - mean.GDP)^2)
599 R.squared.GDP <- pred/real
600
601
    #Root mean squared error
602
    RMSE(residuals.grouped.lasso.GDP)
603
604
    #Mean squared prediction error
605
    grouped.lasso.GDP.validation <- cbind(rep(1, each = 15),</pre>
606
                                           validation.data.GDP.predictors) %*%
607
                                           gl.GDP.coefficients
608
    mean((validation.data.GDP.dependent.variable-grouped.lasso.GDP.validation)^2)
609
610
    #Plot of the real vs. the predicted unemplyment rate
611 real <- data.frame(validation.data.GDP.dependent.variable)
612 fc <- data.frame(grouped.lasso.GDP.validation)
613 Comp <- cbind(fc, real)
614 colnames(Comp) <- c("Predicted.GDP", "Real.GDP")
615 Quarter <- seq(1, 15, by = 1)
616
617
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Predicted.GDP,
618
                                                            color = 'Predicted.GDP'), size = 1) +
619
      geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
                                  color = 'Real.GDP'), size = 1) +
620
621
    labs(x = "Quarter", y = "Rate") +
622
     theme(axis.title = element_text(face = "bold", size = 20)) +
623
      scale_color_manual(name = "", values = c("red", "blue"),
624
                         breaks = c("Predicted.GDP", "Real.GDP")) +
625
     theme(legend.text = element_text(size = 20))
626
627 #Creating the forecast
628 #Forecast 2,5 year ahead
629 grouped.lasso.V <- as.vector(grouped.lasso.GDP)
630
    grouped.lasso.TS <- ts(grouped.lasso.V)</pre>
631 forecast <- forecast(grouped.lasso.TS, h=10)
632
    summary <-summary(forecast)</pre>
633 forecast.grouped.lasso.GDP <- summary$'Point Forecast'
634
635 RMSE(forecast.data.GDP.dependent.variable - forecast.grouped.lasso.GDP)
636 MAE(forecast.data.GDP.dependent.variable - forecast.grouped.lasso.GDP)
637
638 #Plot of the real vs. the forecasted unemplyment rate
639
    real <- data.frame(forecast.data.GDP.dependent.variable)</pre>
640
    fc <- data.frame(forecast.grouped.lasso.GDP)</pre>
641
    Comp <- cbind(fc, real)</pre>
    colnames(Comp) <- c("Forecasted.GDP", "Real.GDP")</pre>
642
643
    Quarter <- seq(1, 10, by = 1)
644
645
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Forecasted.GDP,
646
                                                           color = 'Forecasted.GDP'), size = 1) +
647
                                 geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
648
                                                             color = 'Real.GDP'), size = 1) +
649
                                labs(x = "Quarter", y = "Rate") +
```

650	theme(axis.title = element_text(face = "bold", size = 20)) +
651	<pre>scale_color_manual(name = "", values = c("red", "blue"),</pre>
652	<pre>breaks = c("Forecasted.GDP", "Real.GDP")) +</pre>
653	<pre>theme(legend.text = element_text(size = 20))</pre>

R-koder/Lasso.R

```
1
 2
  library("grpreg")
 3
 4
   5
   elastic.net.CUR <- cv.glmnet(training.data.CUR.predictors,</pre>
 6
                              training.data.CUR.dependent.variable, k = 5, alpha = 0.5,
 7
                               type.measure = "mse")
 8
 9
  plot(elastic.net.CUR,
       xlab = "", ylab = "")
10
11 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
  mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
12
13
   abline(h = 0, col = "red")
14
15
  best.elastic.net.CUR <- elastic.net.CUR$lambda.min</pre>
16
17 #Model adequacy
18 en.CUR.coefficients <- coef(elastic.net.CUR)
19
   elastic.net.CUR <- cbind(rep(1, each = 31),</pre>
20
                           training.data.CUR.predictors) %*% en.CUR.coefficients
21
22
  residuals.elastic.net.CUR <- as.matrix(training.data.CUR.dependent.variable-elastic.net.CUR)
23
24 #Test for normal errors
25 shapiro.test(residuals.elastic.net.CUR)
26
27
  #Fitted values
28
  fit <-as.vector(elastic.net.CUR)</pre>
29
30
  plot(residuals.elastic.net.CUR<sup>~</sup>fit,
31
       xlab = "", ylab = "")
32 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
33 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
  abline(h = 0, col = "red")
34
35
36
  #Histogram of the residuals compared with the normal-distribution
  h <- hist(residuals.elastic.net.CUR, xlab = "", ylab = "")</pre>
37
38 mtext("Index", side = 2, line = 2.2, cex = 1.5)
39 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
40 xfit <- seq(min(residuals.elastic.net.CUR), max(residuals.elastic.net.CUR), length = 40)
41 yfit <- dnorm(xfit, mean = mean(residuals.elastic.net.CUR),
42
                sd = sd(residuals.elastic.net.CUR))
43 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.elastic.net.CUR)
44
  lines(xfit, yfit, col = "blue", lwd = 2)
45
```

```
46 #Coefficient of determination
47 mean.CUR <- mean(training.data.CUR.dependent.variable)
48 pred <- sum((elastic.net.CUR-mean.CUR)^2)
49 real <- sum((training.data.CUR.dependent.variable - mean.CUR)^2)
50 R.squared.CUR <- pred / real
51
52 #Root mean squared error
53 RMSE(residuals.elastic.net.CUR)
54
55 #Mean squared prediction error
56 elastic.net.CUR.validation <- cbind(rep(1, each = 15),
57
                                       validation.data.CUR.predictors) %*% en.CUR.coefficients
58 mean((validation.data.CUR.dependent.variable-elastic.net.CUR.validation)^2)
59
60 #Plot of the real vs. the predicted unemplyment rate
61 real <- data.frame(validation.data.CUR.dependent.variable)
62 fc <- as.vector(elastic.net.CUR.validation)
63 fc <- data.frame(fc)
64 Comp <- cbind(fc, real)
65
   colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
66 Quarter <- seq(1, 15, by = 1)
67
68 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
69
                                                          y = Comp$Predicted.Unemployment.Rate,
70
                                                          color = 'Predicted.Unemployment.Rate'),
                                                                size = 1) +
71
     geom_line(data = real, aes(x = Quarter,
72
                                y = Comp$Real.Unemployment.Rate,
73
                                color = 'Real.Unemployment.Rate'),
74
               size = 1) +
75
   labs(x = "Quarter", y = "Rate") +
76
   theme(axis.title = element_text(face = "bold", size = 20)) +
     scale_color_manual(name = "", values = c("red", "blue"),
77
78
                        breaks = c("Predicted.Unemployment.Rate",
79
                                   "Real.Unemployment.Rate")) +
80
    theme(legend.text = element_text(size = 20))
81
82 #Forecast 2,5 year ahead
83 elastic.net.V <- as.vector(elastic.net.CUR)
84 elastic.net.TS <- ts(elastic.net.V)
85 forecast <- forecast(elastic.net.TS, h=10)</pre>
   summary <-summary(forecast)</pre>
86
87
   forecast.elastic.net.CUR <- summary$'Point Forecast'</pre>
88
89 RMSE(forecast.data.CUR.dependent.variable - forecast.elastic.net.CUR)
90 MAE(forecast.data.CUR.dependent.variable - forecast.elastic.net.CUR)
91
92 #Plot of the real vs. the forecasted unemplyment rate
93 real <- data.frame(forecast.data.CUR.dependent.variable)
94 fc <- data.frame(forecast.elastic.net.CUR)
95 Comp <- cbind(fc, real)
96 colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
```

```
97 Quarter <- seq(1, 10, by = 1)
98
99
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
100
                                          y = Comp$Forecasted.Unemployment.Rate,
101
                                          color = 'Forecasted.Unemployment.Rate'), size = 1) +
102
                               geom_line(data = real, aes(x = Quarter,
103
                                          y = Comp$Real.Unemployment.Rate,
104
                                          color = 'Real.Unemployment.Rate'), size = 1) +
105
                               labs(x = "Quarter", y = "Rate") +
106
                               theme(axis.title = element_text(face = "bold", size = 20)) +
                                scale_color_manual(name = "", values = c("red", "blue"),
107
108
                                breaks = c("Forecasted.Unemployment.Rate",
109
                                           "Real.Unemployment.Rate")) +
110
                               theme(legend.text = element_text(size = 20))
111
113 elastic.net.Infl <- cv.glmnet(training.data.Infl.predictors,</pre>
114
                                 training.data.Infl.dependent.variable,
115
                                 k = 5, alpha = 0.5, type.measure = "mse")
116 plot(elastic.net.Infl,
        xlab = "", ylab = "")
117
118 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
119 mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
120 abline(h = 0, col = "red")
121
122
   best.elastic.net.Infl <- elastic.net.Infl$lambda.min
123
124
    #Model adequacy
   en.Infl.coefficients <- coef(elastic.net.Infl)</pre>
125
126 elastic.net.Infl <- cbind(rep(1, each = 31), training.data.Infl.predictors) %*%
127
                             en.Infl.coefficients
128
129 residuals.elastic.net.Infl <-
130
                             as.matrix(training.data.Infl.dependent.variable-elastic.net.Infl)
131
   #Test for normal errors
132 shapiro.test(residuals.elastic.net.Infl)
133
134 #Fitted values
135 fit <-as.vector(elastic.net.Infl)
136
137
   plot(residuals.elastic.net.Infl~fit,
138
        xlab = "", ylab = "")
139
   mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
140 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
141 abline(h = 0, col = "red")
142
143 #Histogram of the residuals compared with the normal-distribution
144 h <- hist(residuals.elastic.net.Infl, xlab = "", ylab = "")
145 mtext("Index", side = 2, line = 2.2, cex = 1.5)
146 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
147
   xfit <- seq(min(residuals.elastic.net.Infl), max(residuals.elastic.net.Infl), length = 40)
148 yfit <- dnorm(xfit, mean = mean(residuals.elastic.net.Infl),
```

```
149
                  sd = sd(residuals.elastic.net.Infl))
150 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.elastic.net.Infl)
151 lines(xfit, yfit, col = "blue", lwd = 2)
152
153 #Coefficient of determination
154 mean.Infl <- mean(training.data.Infl.dependent.variable)
155 pred <- sum((elastic.net.Infl-mean.Infl)^2)
156 real <- sum((training.data.Infl.dependent.variable - mean.Infl)^2)
157 R.squared.Infl <- real / pred
158
159 #Root mean squared error
160
    RMSE(residuals.elastic.net.Infl)
161
162 #Mean squared prediction error
163 elastic.net.Infl.validation <- cbind(rep(1, each = 15),</pre>
164
                                         validation.data.Infl.predictors) %*%
165
                                          en.Infl.coefficients
166 mean((validation.data.Infl.dependent.variable - elastic.net.Infl.validation)^2)
167
168 #Plot of the real vs. the predicted unemplyment rate
169 real <- data.frame(validation.data.Infl.dependent.variable)
170 fc <- as.vector(elastic.net.Infl.validation)
171 fc <- as.data.frame(fc)
172 Comp <- cbind(fc, real)
173 colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
174
    Quarter <- seq(1, 15, by = 1)
175
176
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc,
177
                                           aes(x = Quarter, y = Comp$Predicted.Inflation.Rate,
178
                                                color = 'Predicted.Inflation.Rate'), size = 1) +
179
     geom_line(data = real, aes(x = Quarter,
180
                                 y = Comp$Real.Inflation.Rate,
                                 color = 'Real.Inflation.Rate'), size = 1) +
181
182
    labs(x = "Quarter", y = "Rate") +
183
     theme(axis.title = element_text(face = "bold", size = 20)) +
184
     scale_color_manual(name = "", values = c("red", "blue"),
185
                         breaks = c("Predicted.Inflation.Rate",
186
                                    "Real.Inflation.Rate")) +
     theme(legend.text = element_text(size = 20))
187
188
189 #Forecast 2,5 year ahead
190
    elastic.net.V <- as.vector(elastic.net.Infl)</pre>
191
    elastic.net.TS <- ts(elastic.net.V)</pre>
192 forecast <- forecast(elastic.net.TS, h=10)
193 summary <-summary(forecast)
194 forecast.elastic.net.Infl <- summary$'Point Forecast'
195
196 RMSE(forecast.data.Infl.dependent.variable - forecast.elastic.net.Infl)
197 MAE(forecast.data.Infl.dependent.variable - forecast.elastic.net.Infl)
198
199 #Plot of the real vs. the forecasted unemplyment rate
200 real <- data.frame(forecast.data.Infl.dependent.variable)
```

```
201 fc <- data.frame(forecast.elastic.net.Infl)
202 Comp <- cbind(fc, real)
203
   colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")</pre>
204
    Quarter <- seq(1, 10, by = 1)
205
206 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
207
                                          y = Comp$Forecasted.Inflation.Rate,
208
                                          color = 'Forecasted.Inflation.Rate'), size = 1) +
209
                                geom_line(data = real, aes(x = Quarter,
210
                                          y = Comp$Real.Inflation.Rate,
211
                                          color = 'Real.Inflation.Rate'), size = 1) +
212
                                labs(x = "Quarter", y = "Rate") +
                               theme(axis.title = element_text(face = "bold", size = 20)) +
213
214
                                scale_color_manual(name = "", values = c("red", "blue"),
215
                                breaks = c("Forecasted.Inflation.Rate", "Real.Inflation.Rate"))
                                     +
216
                                theme(legend.text = element_text(size = 20))
217
219
    elastic.net.GDP <- cv.glmnet(training.data.GDP.predictors,</pre>
220
                                training.data.GDP.dependent.variable,
221
                                k = 5, alpha = 0.5, type.measure = "mse")
222 plot(elastic.net.GDP,
223
        xlab = "", ylab = "")
224 mtext("Mean-Squared-Error", side = 2, line = 2.2, cex = 1.5)
   mtext("log(Labda)", side = 1, line = 2.2, cex = 1.5)
225
    abline(h = 0, col = "red")
226
227
228 best.elastic.net.GDP <- elastic.net.GDP$lambda.min
229
230 #Model adequacy
231 en.GDP.coefficients <- coef(elastic.net.GDP)
232 elastic.net.GDP <- cbind(rep(1, each = 31),
233
                            training.data.GDP.predictors) %*% en.GDP.coefficients
234
235 residuals.elastic.net.GDP <- as.matrix(training.data.GDP.dependent.variable-elastic.net.GDP)
236
237 #Test for normal errors
238 shapiro.test(residuals.elastic.net.GDP)
239
240 #Fitted values
241
   fit <-as.vector(elastic.net.GDP)</pre>
242
243 plot(residuals.elastic.net.GDP~fit,
244
        xlab = "", ylab = "")
245 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
246 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
247 abline(h = 0, col = "red")
248
249 #Histogram of the residuals compared with the normal-distribution
250 h <- hist(residuals.elastic.net.GDP, xlab = "", ylab = "")
251 mtext("Index", side = 2, line = 2.2, cex = 1.5)
```

```
252 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
253 xfit <- seq(min(residuals.elastic.net.GDP), max(residuals.elastic.net.GDP),length = 40)
254 yfit <- dnorm(xfit, mean = mean(residuals.elastic.net.GDP),
255
                  sd = sd(residuals.elastic.net.GDP))
256 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.elastic.net.GDP)
257 lines(xfit, yfit, col = "blue", lwd = 2)
258
259 #Coefficient of determination
260 mean.GDP <- mean(training.data.GDP.dependent.variable)
261 pred <- sum((elastic.net.GDP-mean.GDP)^2)
262 real <- sum((training.data.GDP.dependent.variable - mean.GDP)^2)
    R.squared.GDP <- pred / real
263
264
265 #Root mean squared error
266 RMSE(residuals.elastic.net.GDP)
267
268 #Mean squared prediction error
269 elastic.net.GDP.validation <- cbind(rep(1, each = 15),
270
                                        validation.data.GDP.predictors) %*% en.GDP.coefficients
271
    mean((validation.data.GDP.dependent.variable - elastic.net.GDP.validation)^2)
272
273 #Plot of the real vs. the predicted unemplyment rate
274 real <- data.frame(validation.data.GDP.dependent.variable)
275 fc <- as.vector(elastic.net.GDP.validation)
276 | fc <- data.frame(fc)
277 Comp <- cbind(fc, real)
    colnames(Comp) <- c("Predicted.GDP", "Real.GDP")</pre>
278
279
    Quarter <- seq(1, 15, by = 1)
280
281
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Predicted.GDP,
282
                                                          color = 'Predicted.GDP'), size = 1) +
283
    geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
284
                                 color = 'Real.GDP'), size = 1) +
285
    labs(x = "Quarter", y = "Rate") +
286
     theme(axis.title = element_text(face = "bold", size = 20)) +
287
     scale_color_manual(name = "", values = c("red", "blue"),
288
                        breaks = c("Predicted.GDP", "Real.GDP")) +
289
    theme(legend.text = element_text(size = 20))
290
291 #Creating the forecast
292 #Forecast 2,5 year ahead
293
    elastic.net.V <- as.vector(elastic.net.GDP)</pre>
294
    elastic.net.TS <- ts(elastic.net.V)</pre>
295 forecast <- forecast(elastic.net.TS, h=10)
296 summary <-summary(forecast)
297 forecast.elastic.net.GDP <- summary$'Point Forecast'
298
299 RMSE(forecast.data.GDP.dependent.variable - forecast.elastic.net.GDP)
300 MAE(forecast.data.GDP.dependent.variable - forecast.elastic.net.GDP)
301
302 #Plot of the real vs. the forecasted unemplyment rate
303 real <- data.frame(forecast.data.GDP.dependent.variable)
```

```
304 fc <- data.frame(forecast.elastic.net.GDP)
305 Comp <- cbind(fc, real)
306
    colnames(Comp) <- c("Forecasted.GDP", "Real.GDP")</pre>
307
    Quarter <- seq(1, 10, by = 1)
308
309
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter, y = Comp$Forecasted.GDP,
310
                                            color = 'Forecasted.GDP'), size = 1) +
                                 geom_line(data = real, aes(x = Quarter, y = Comp$Real.GDP,
311
312
                                            color = 'Real.GDP'), size = 1) +
313
                                labs(x = "Quarter", y = "Rate") +
314
                                 theme(axis.title = element_text(face = "bold", size = 20)) +
315
                                 scale_color_manual(name = "", values = c("red", "blue"),
                                                     breaks = c("Forecasted.GDP", "Real.GDP")) +
316
317
                                theme(legend.text = element_text(size = 20))
```

```
R-koder/ElasticNet.R
```

B.3 VAR models

```
1
  2
3
  library("vars")
4 library("MTS")
5 library("ggplot2")
6
7
  8
9
  # Estimate factors from PCA
10
  pca.predictors <- apca(training.data.CUR.predictors, 5)</pre>
11
12 factor.CUR <- pca.predictors$factors</pre>
13 FAVAR <- as.matrix(cbind(factor.CUR, training.data.CUR.dependent.variable))
  colnames(FAVAR) <- c("f1", "f2", "f3", "f4", "f5", "CUR")
14
15
16
  #Determine the number of lags
17
  VARselect(training.data.CUR.dependent.variable, lag.max = 10)$select
18
19 FAVAR.model <- VAR(FAVAR, p=3)
20 summary (FAVAR.model)
21
22
  coefficients.CUR <- as.matrix(coef(FAVAR.model)$CUR)</pre>
23
  coef.CUR <- as.vector(coefficients.CUR[,1])</pre>
24
25
  #Residuals
26 residuals <- as.matrix(residuals(FAVAR.model))
27 residuals.CUR <- as.vector(residuals[,6])
28
29 #Fitted values
30 fitted.values <- as.matrix(fitted(FAVAR.model))
31 fitted.values.CUR <- as.vector(fitted.values[,6])</pre>
```

```
32
33 plot(residuals.CUR~fitted.values.CUR,
34
      xlab = "", ylab = "")
35
   mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
36 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
37 abline(h = 0, col = "red")
38
39 # Test for normality
40 shapiro.test(residuals.CUR)
41
42 #Histogram of the residuals compared with the normal-distribution
43 h <- hist(residuals.CUR, xlab = "", ylab = "")
44 mtext("Index", side = 2, line = 2.2, cex = 1.5)
45 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
46 xfit <- seq(min(residuals.CUR), max(residuals.CUR), length = 40)
47 yfit <- dnorm(xfit, mean = mean(residuals.CUR), sd = sd(residuals.CUR))
48 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.CUR)
49 lines(xfit, yfit, col = "blue", lwd = 2)
50
51
   #Coefficient of determination
52 summary (FAVAR.model)
53
54 #Root mean squared error
55 RMSE(residuals.CUR)
56
57 #Mean squared prediction error
58
   validation.favar.model <- predict(FAVAR.model, n.ahead = 15, ci = 0.95,
59
                               dumvar = validation.data.CUR.predictors)
60 validation.CUR.favar.model <- as.matrix(validation.favar.model$fcst$CUR)
61 validation.CUR <- validation.CUR.favar.model[,1]
62
63 # MSPE - prediction error
64 mean((validation.data.CUR.dependent.variable - validation.CUR)^2)
65
66 #Plot of the real vs. the predicted unemplyment rate
67 real <- data.frame(validation.data.CUR.dependent.variable)
68 fc <- data.frame(validation.CUR)
69 Comp <- cbind(fc, real)
70 colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
71 Quarter <- seg(1, 15, by = 1)
72
73
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
74
                                                         y = Comp$Predicted.Unemployment.Rate,
75
                                                          color = 'Predicted.Unemployment.Rate'),
76
                                          size = 1) +
77
                                geom_line(data = real, aes(x = Quarter,
78
                                                           y = Comp$Real.Unemployment.Rate,
79
                                                           color = 'Real.Unemployment.Rate'),
80
                                          size = 1) +
81
                                labs(x = "Quarter", y = "Rate") +
82
                                theme(axis.title = element_text(face = "bold", size = 20))+
83
                                scale_color_manual(name = "", values = c("red", "blue"),
```

```
84
                                 breaks = c("Predicted.Unemployment.Rate",
85
                                            "Real.Unemployment.Rate"))+
86
                                 theme(legend.text = element_text(size = 20))
 87
 88
89
90 #Forecast 2,5 year ahead
91 forecast.favar.model <- predict(FAVAR.model, n.ahead = 10, ci = 0.95, dumvar = NULL)
92 forecast.CUR.favar.model <- as.matrix(forecast.favar.model$fcst$CUR)
93 forecast.CUR <- forecast.CUR.favar.model[.1]
94
95
   RMSE(forecast.data.CUR.dependent.variable - forecast.CUR)
96
   MAE(forecast.data.CUR.dependent.variable - forecast.CUR)
97
98 #Plot of the real vs. the forecasted unemplyment rate
99 real <- data.frame(forecast.data.CUR.dependent.variable)
100 fc <- data.frame(forecast.CUR)
101 Comp <- cbind(fc, real)
102
   colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")</pre>
103
   Quarter <- seq(1, 10, by = 1)
104
105 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
106
                                                         y = Comp$Forecasted.Unemployment.Rate,
107
                                                      color = 'Forecasted.Unemployment.Rate'),
108
                                            size = 1) +
109
                                 geom_line(data = real, aes(x = Quarter,
110
                                                            y = Comp$Real.Unemployment.Rate,
111
                                                            color = 'Real.Unemployment.Rate'),
112
                                           size = 1) +
113
                                labs(x = "Quarter", y = "Rate") +
114
                                theme(axis.title = element_text(face = "bold", size = 20))+
                                scale_color_manual(name = "", values = c("red", "blue"),
115
                                breaks = c("Forecasted.Unemployment.Rate",
116
117
                                           "Real.Unemployment.Rate"))+
118
                                theme(legend.text = element_text(size = 20))
119
120
122
123 # Estimate factors from PCA
124 pca.predictors <- apca(training.data.Infl.predictors, 13)
125
126
   factor.Infl <- pca.predictors$factors</pre>
127
   FAVAR <- as.matrix(cbind(factor.Infl, training.data.Infl.dependent.variable))
128 colnames(FAVAR) <- c("f1","f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10",
129
                        "f11","f12", "f13", "Infl")
130
131 #Determine the number of lags
132 VARselect(training.data.Infl.dependent.variable, lag.max = 10)$select
133
134 FAVAR.model <- VAR(FAVAR, p=1)
135 summary (FAVAR.model)
```

```
136
137 coefficients.Infl <- as.matrix(coef(FAVAR.model)$Infl)
138 coef.Infl <- as.vector(coefficients.Infl[,1])
139
140 #Residuals
141 residuals <- as.matrix(residuals(FAVAR.model))
142 residuals.Infl <- as.vector(residuals[,14])
143
144 #Fitted values
145 fitted.values <- as.matrix(fitted(FAVAR.model))
146 fitted.values.Infl <- as.vector(fitted.values[,14])
147
148 plot(residuals.Infl~fitted.values.Infl,
149
       xlab = "", ylab = "")
150 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
151 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
152 abline(h = 0, col = "red")
153
154 # Test for normality
155 shapiro.test(residuals.Infl)
156
157 #Histogram of the residuals compared with the normal-distribution
158 h <- hist(residuals.Infl, xlab = "", ylab = "")
159 mtext("Index", side = 2, line = 2.2, cex = 1.5)
160 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
161 xfit <- seq(min(residuals.Infl), max(residuals.Infl), length = 40)
    yfit <- dnorm(xfit, mean = mean(residuals.Infl), sd = sd(residuals.Infl))</pre>
162
163 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.Infl)
164 lines(xfit, yfit, col = "blue", lwd = 2)
165
166 #Coefficient of determination
167 summary (FAVAR.model)
168
169 #Root mean squared error
170 RMSE(residuals.Infl)
171
172 #Mean squared prediction error
173 validation.favar.model <- predict(FAVAR.model, n.ahead = 15, ci = 0.95,
174
                                dumvar = validation.data.Infl.predictors)
175 validation.Infl.favar.model <- as.matrix(validation.favar.model$fcst$Infl)
176 validation.Infl <- validation.Infl.favar.model[,1]
177
178
    # MSPE - prediction error
179 mean((validation.data.Infl.dependent.variable - validation.Infl)^2)
180
181 #Plot of the real vs. the predicted Inflation rate
182 real <- data.frame(validation.data.Infl.dependent.variable)</pre>
183 fc <- data.frame(validation.Infl)
184 Comp <- cbind(fc, real)</pre>
185 colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
186 Quarter <- seq(1, 15, by = 1)
187
```

```
188
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
189
                                                          y = Comp$Predicted.Inflation.Rate,
190
                                                           color = 'Predicted.Inflation.Rate'),
191
                                           size = 1) +
192
                                 geom_line(data = real, aes(x = Quarter,
193
                                                            y = Comp$Real.Inflation.Rate,
194
                                                            color = 'Real.Inflation.Rate'),
                                           size = 1) +
195
196
                                 labs(x = "Quarter", y = "Rate") +
197
                                 theme(axis.title = element_text(face = "bold", size = 20))+
                                 scale_color_manual(name = "", values = c("red", "blue"),
198
199
                                 breaks = c("Predicted.Inflation.Rate",
200
                                            "Real.Inflation.Rate"))+
201
                                 theme(legend.text = element_text(size = 20))
202
203
204
205 #Forecast 2,5 year ahead
206 forecast.favar.model <- predict(FAVAR.model, n.ahead = 10, ci = 0.95, dumvar = NULL)
207
    forecast.Infl.favar.model <- as.matrix(forecast.favar.model$fcst$Infl)</pre>
208 forecast.Infl <- forecast.Infl.favar.model[.1]
209
210 RMSE(forecast.data.Infl.dependent.variable - forecast.Infl)
211 MAE(forecast.data.Infl.dependent.variable - forecast.Infl)
212
213 #Plot of the real vs. the forecasted inflation rate
214
    real <- data.frame(forecast.data.Infl.dependent.variable)</pre>
215
   fc <- data.frame(forecast.Infl)</pre>
216 Comp <- cbind(fc, real)
217 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
218 Quarter <- seq(1, 10, by = 1)
219
220 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
221
                                                          y = Comp$Forecasted.Inflation.Rate,
222
                                                          color = 'Forecasted.Inflation.Rate'),
223
                                           size = 1) +
224
                                 geom_line(data = real, aes(x = Quarter,
225
                                                            y = Comp$Real.Inflation.Rate,
                                                            color = 'Real.Inflation.Rate'),
226
227
                                           size = 1) +
228
                                 labs(x = "Quarter", y = "Rate") +
229
                                 theme(axis.title = element_text(face = "bold", size = 20))+
230
                                 scale_color_manual(name = "", values = c("red", "blue"),
231
                                 breaks = c("Forecasted.Inflation.Rate",
232
                                            "Real.Inflation.Rate"))+
233
                                 theme(legend.text = element_text(size = 20))
234
235
237
238
   # Estimate factors from PCA
239 pca.predictors <- apca(training.data.GDP.predictors, 13)
```

```
240
241 factor.GDP <- pca.predictors$factors
242 FAVAR <- as.matrix(cbind(factor.GDP, training.data.GDP.dependent.variable))
243
    colnames(FAVAR) <- c("f1","f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10",
244
                         "f11", "f12", "f13", "GDP")
245
246 #Determine the number of lags
247 VARselect(training.data.GDP.dependent.variable, lag.max = 10)$select
248
249 FAVAR.model <- VAR(FAVAR, p=1)
250
    summary(FAVAR.model)
251
252 coefficients.GDP <- as.matrix(coef(FAVAR.model)$GDP)
253 coef.GDP <- as.vector(coefficients.GDP[,1])
254
255 #Residuals
256 residuals <- as.matrix(residuals(FAVAR.model))
257 residuals.GDP <- as.vector(residuals[,14])
258
259 #Fitted values
260 fitted.values <- as.matrix(fitted(FAVAR.model))
261 fitted.values.GDP <- as.vector(fitted.values[,14])
262
263 plot(residuals.GDP~fitted.values.GDP,
264
        xlab = "", ylab = "")
265
    mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
    mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
266
267
    abline(h = 0, col = "red")
268
269 # Test for normality
270 shapiro.test(residuals.GDP)
271
\left. 272 \right| #Histogram of the residuals compared with the normal-distribution
273 h <- hist(residuals.GDP, xlab = "", ylab = "")
274 mtext("Index", side = 2, line = 2.2, cex = 1.5)
275 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
276 xfit <- seq(min(residuals.GDP), max(residuals.GDP), length = 40)
277 | yfit <- dnorm(xfit, mean = mean(residuals.GDP), sd = sd(residuals.GDP))
278 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.GDP)
279 lines(xfit, vfit, col = "blue", lwd = 2)
280
281
    #Coefficient of determination
282
    summary(FAVAR.model)
283
284 #Root mean squared error
285 RMSE(residuals.GDP)
286
287 #Mean squared prediction error
288 validation.favar.model <- predict(FAVAR.model, n.ahead = 15, ci = 0.95,
289
                                dumvar = validation.data.GDP.predictors)
290 validation.GDP.favar.model <- as.matrix(validation.favar.model$fcst$GDP)
291 validation.GDP <- validation.GDP.favar.model[,1]
```

```
292
293 # MSPE - prediction error
294 mean((validation.data.GDP.dependent.variable - validation.GDP)^2)
295
296 #Plot of the real vs. the predicted GDP rate
297
   real <- data.frame(validation.data.GDP.dependent.variable)</pre>
298 fc <- data.frame(validation.GDP)
299 Comp <- cbind(fc, real)
300
    colnames(Comp) <- c("Predicted.GDP.Rate", "Real.GDP.Rate")</pre>
301
    Quarter <- seq(1, 15, by = 1)
302
303
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
304
                                                            v = Comp$Predicted.GDP.Rate.
305
                                                            color = 'Predicted.GDP.Rate'),
306
                                            size = 1) +
307
                                  geom_line(data = real, aes(x = Quarter,
308
                                                              y = Comp$Real.GDP.Rate,
                                                              color = 'Real.GDP.Rate'),
309
310
                                            size = 1) +
311
                                  labs(x = "Quarter", y = "Rate") +
312
                                  theme(axis.title = element_text(face = "bold", size = 20))+
                                  scale_color_manual(name = "", values = c("red", "blue"),
313
314
                                  breaks = c("Predicted.GDP.Rate",
315
                                             "Real.GDP.Rate"))+
316
                                  theme(legend.text = element_text(size = 20))
317
318 #Forecast 2,5 year ahead
319
    forecast.favar.model <- predict(FAVAR.model, n.ahead = 10, ci = 0.95, dumvar = NULL)
320
   forecast.GDP.favar.model <- as.matrix(forecast.favar.model$fcst$GDP)
321 forecast.GDP <- forecast.GDP.favar.model[,1]
322
323 RMSE(forecast.data.GDP.dependent.variable - forecast.GDP)
324 MAE(forecast.data.GDP.dependent.variable - forecast.GDP)
325
326 #Plot of the real vs. the forecasted GDP rate
327
    real <- data.frame(forecast.data.GDP.dependent.variable)</pre>
328 fc <- data.frame(forecast.GDP)
329 Comp <- cbind(fc, real)
330 colnames(Comp) <- c("Forecasted.GDP.Rate", "Real.GDP.Rate")
331 Quarter <- seq(1, 10, by = 1)
332
333
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
334
                                                            y = Comp$Forecasted.GDP.Rate,
335
                                                            color = 'Forecasted.GDP.Rate'),
336
                                            size = 1) +
337
                                  geom_line(data = real, aes(x = Quarter,
338
                                                             y = Comp$Real.GDP.Rate,
339
                                                              color = 'Real.GDP.Rate'),
340
                                            size = 1) +
341
                                  labs(x = "Quarter", y = "Rate") +
342
                                  theme(axis.title = element_text(face = "bold", size = 20))+
343
                                  scale_color_manual(name = "", values = c("red", "blue"),
```

```
344breaks = c("Forecasted.GDP.Rate",345"Real.GDP.Rate"))+346theme(legend.text = element_text(size = 20))
```

```
R-koder/FAVAR.R
```

```
1
  2
3 devtools::install_github("lcallot/lassovar")
4 library("lassovar")
5 library("glmnet")
  library("vars")
6
7
  library("forecast")
8
9
  all.training <- as.matrix(cbind(training.data.CUR.dependent.variable,
10
                            training.data.CUR.predictors))
11
12 lasso.var.model <- lassovar(all.training)
13
14 summary.lassovar(lasso.var.model)
15 # GDP (diff.log.GDP.. ) - 26 non-zero - nr.2
16 # Infl (diff.log.Infl...degrees...2.) - 26 nonzero - nr.50
17 # CUR (training.data.CUR.dependent.variable ) - 10 non-zero - nr.1
18
19 coef.matrix <- as.matrix(coef(lasso.var.model))</pre>
20
21
22
  23
24 #Model adequacy
25
26 coef.CUR <- coef.matrix[,1]
27
28 #Residuals
29
  residuals.CUR <- residuals.lassovar(lasso.var.model)$training.data.CUR.dependent.variable
30
31 #Fitted values
32 fitted.values.CUR <- as.vector(residuals.CUR) +
33 as.vector(training.data.CUR.dependent.variable[2:31])
34
35 plot(residuals.CUR~fitted.values.CUR,
36
      xlab = "", ylab = "")
37
  mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
38 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
39 abline(h = 0, col = "red")
40
41 # Test for normality
42 shapiro.test(residuals.CUR)
43
44 #Histogram of the residuals compared with the normal-distribution
45 h <- hist(residuals.CUR, xlab = "", ylab = "")
46 mtext("Index", side = 2, line = 2.2, cex = 1.5)
```

150

```
47 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
48 xfit <- seq(min(residuals.CUR), max(residuals.CUR), length = 40)
49 yfit <- dnorm(xfit, mean = mean(residuals.CUR), sd = sd(residuals.CUR))
50
   yfit <- yfit * diff(h$mids[1:2]) * length(residuals.CUR)</pre>
   lines(xfit, yfit, col = "blue", lwd = 2)
51
52
53
  #Coefficient of determination
54 lasso.var.model
55
56 #Root mean squared error
57
  RMSE(residuals.CUR)
58
59
   #Mean squared prediction error
60 all.validation <- as.matrix(cbind(validation.data.CUR.dependent.variable,
61
                                      validation.data.CUR.predictors))
62
63 validation.matrix <- predict.lassovar(lasso.var.model, all.validation)
  validation.CUR <- validation.matrix[,1]
64
65
66
   mean((validation.data.CUR.dependent.variable - validation.CUR)^2)
67
68 #Plot of the real vs. the predicted unemplyment rate
69 real <- data.frame(validation.data.CUR.dependent.variable)
70 fc <- data.frame(validation.CUR)
71 Comp <- cbind(fc, real)</pre>
72
   colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")</pre>
73
   Quarter <- seq(1, 15, by = 1)
74
75
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
76
                                                           y = Comp$Predicted.Unemployment.Rate,
77
                                                           color = 'Predicted.Unemployment.Rate'),
78
                                           size = 1) +
79
                                 geom_line(data = real, aes(x = Quarter,
80
                                                             y = Comp$Real.Unemployment.Rate,
81
                                                             color = 'Real.Unemployment.Rate'),
82
                                           size = 1) +
83
                                 labs(x = "Quarter", y = "Rate") +
84
                                 theme(axis.title = element_text(face = "bold", size = 20))+
85
                                 scale_color_manual(name = "", values = c("red", "blue"),
86
                                 breaks = c("Predicted.Unemployment.Rate",
87
                                            "Real.Unemployment.Rate"))+
                                 theme(legend.text = element_text(size = 20))
88
89
90
   #Forecast 2,5 year ahead
91
92
  lasso.var.CUR <- cbind(rep(1, each = 31), all.training) %*% as.vector(coef.CUR)
93 ts.lasso.var.CUR <- ts(lasso.var.CUR)
94 forecast.CUR <- forecast(ts.lasso.var.CUR, h= 10)
  forecast.CUR.values <- summary(forecast.CUR)$"Point Forecast"</pre>
95
96
97
   RMSE(forecast.data.CUR.dependent.variable - forecast.CUR.values)
98 MAE(forecast.data.CUR.dependent.variable - forecast.CUR.values)
```

```
99
100 \#Plot of the real vs. the forecasted unemplyment rate
101 real <- data.frame(forecast.data.CUR.dependent.variable)
102 fc <- data.frame(forecast.CUR.values)</pre>
103 Comp <- cbind(fc, real)
104 | colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
105 Quarter <- seq(1, 10, by = 1)
106
107
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
108
                                                          y = Comp$Forecasted.Unemployment.Rate,
109
                                                        color = 'Forecasted.Unemployment.Rate'),
110
                                           size = 1) +
111
                                 geom_line(data = real, aes(x = Quarter,
112
                                                           y = Comp$Real.Unemployment.Rate,
113
                                                            color = 'Real.Unemployment.Rate'),
                                           size = 1) +
114
                                 labs(x = "Quarter", y = "Rate") +
115
                                 theme(axis.title = element_text(face = "bold", size = 20))+
116
117
                                 scale_color_manual(name = "", values = c("red", "blue"),
118
                                 breaks = c("Forecasted.Unemployment.Rate",
119
                                            "Real.Unemployment.Rate"))+
120
                                 theme(legend.text = element_text(size = 20))
121
122
123
    124
125
    #Model adequacy
126
127
    coef.Infl <- coef.matrix[,50]</pre>
128
129 #Residuals
130 residuals.Infl <- residuals.lassovar(lasso.var.model)$diff.log.Infl...degrees...2.
131
132 #Fitted values
133 fitted.values.Infl <- as.vector(residuals.Infl) +
134
     as.vector(training.data.Infl.dependent.variable[2:31])
135
136 plot(residuals.Infl~fitted.values.Infl,
       xlab = "", ylab = "")
137
138 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
139 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
140
    abline(h = 0, col = "red")
141
142 # Test for normality
143 shapiro.test(residuals.Infl)
144
145 #Histogram of the residuals compared with the normal-distribution
146 h <- hist(residuals.Infl, xlab = "", ylab = "")
147 mtext("Index", side = 2, line = 2.2, cex = 1.5)
148 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
149 xfit <- seq(min(residuals.Infl), max(residuals.Infl), length = 40)
150 yfit <- dnorm(xfit, mean = mean(residuals.Infl), sd = sd(residuals.Infl))
```

```
151 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.Infl)
152 lines(xfit, yfit, col = "blue", lwd = 2)
153
154
   #Coefficient of determination
155 lasso.var.model
156
157 #Root mean squared error
158 RMSE(residuals.Infl)
159
160 #Mean squared prediction error
161 validation.matrix <- predict.lassovar(lasso.var.model, all.validation)
    validation.Infl <- validation.matrix[,50]</pre>
162
163
164 mean((validation.data.Infl.dependent.variable - validation.Infl)^2)
165
166 #Plot of the real vs. the predicted Inflation rate
167 real <- data.frame(validation.data.Infl.dependent.variable)
168 fc <- data.frame(validation.Infl)
169 Comp <- cbind(fc, real)</pre>
170
    colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")</pre>
171 Quarter <- seq(1, 15, by = 1)
172
173 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
174
                                                            y = Comp$Predicted.Inflation.Rate,
175
                                                            color = 'Predicted.Inflation.Rate'),
176
                                            size = 1) +
177
                                  geom_line(data = real, aes(x = Quarter,
178
                                                              y = Comp$Real.Inflation.Rate,
179
                                                              color = 'Real.Inflation.Rate'),
180
                                            size = 1) +
181
                                  labs(x = "Quarter", y = "Rate") +
182
                                  theme(axis.title = element_text(face = "bold", size = 20))+
                                  scale_color_manual(name = "", values = c("red", "blue"),
183
184
                                  breaks = c("Predicted.Inflation.Rate",
185
                                             "Real.Inflation.Rate"))+
186
                                  theme(legend.text = element_text(size = 20))
187
188 #Forecast 2,5 year ahead
189
190 lasso.var.Infl <- cbind(rep(1, each = 31), all.training) %*% as.vector(coef.Infl)
191 ts.lasso.var.Infl <- ts(lasso.var.Infl)
192
    forecast.Infl <- forecast(ts.lasso.var.Infl, h= 10)</pre>
193
    forecast.Infl.values <- summary(forecast.Infl)$"Point Forecast"</pre>
194
195 RMSE(forecast.data.Infl.dependent.variable - forecast.Infl.values)
196 MAE(forecast.data.Infl.dependent.variable - forecast.Infl.values)
197
198 #Plot of the real vs. the forecasted inflation rate
199 real <- data.frame(forecast.data.Infl.dependent.variable)
200 fc <- data.frame(forecast.Infl.values)
201 Comp <- cbind(fc, real)
202 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
```

```
203 Quarter <- seq(1, 10, by = 1)
204
205
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
206
                                                         y = Comp$Forecasted.Inflation.Rate,
207
                                                          color = 'Forecasted.Inflation.Rate'),
208
                                          size = 1) +
209
                                 geom_line(data = real, aes(x = Quarter,
210
                                                           y = Comp$Real.Inflation.Rate,
211
                                                           color = 'Real.Inflation.Rate'),
212
                                          size = 1) +
                                 labs(x = "Quarter", y = "Rate") +
213
214
                                 theme(axis.title = element_text(face = "bold", size = 20))+
215
                                 scale_color_manual(name = "", values = c("red", "blue"),
216
                                 breaks = c("Forecasted.Inflation.Rate",
217
                                           "Real.Inflation.Rate"))+
218
                                 theme(legend.text = element_text(size = 20))
219
220
221
    222
223 #Model adequacy
224
225 coef.GDP <- coef.matrix[,2]
226
227 #Residuals
228
    residuals.GDP <- residuals.lassovar(lasso.var.model)$diff.log.GDP..</pre>
229
230
    #Fitted values
231 fitted.values.GDP <- as.vector(residuals.GDP) +
232
    as.vector(training.data.GDP.dependent.variable[2:31])
233
234 plot(residuals.GDP~fitted.values.GDP,
       xlab = "", ylab = "")
235
236 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
237
    mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
238 abline(h = 0, col = "red")
239
240 # Test for normality
241 shapiro.test(residuals.GDP)
242
243 #Histogram of the residuals compared with the normal-distribution
244 h <- hist(residuals.GDP, xlab = "", ylab = "")
245 mtext("Index", side = 2, line = 2.2, cex = 1.5)
246 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
247 xfit <- seq(min(residuals.GDP), max(residuals.GDP), length = 40)
248 yfit <- dnorm(xfit, mean = mean(residuals.GDP), sd = sd(residuals.GDP))
249 yfit <- yfit * diff(h$mids[1:2]) * length(residuals.GDP)
250 lines(xfit, yfit, col = "blue", lwd = 2)
251
252 #Coefficient of determination
253 lasso.var.model
254
```

```
255 #Root mean squared error
256 RMSE(residuals.GDP)
257
258 #Mean squared prediction error
259 validation.matrix <- predict.lassovar(lasso.var.model, all.validation)
260 validation.GDP <- validation.matrix[,2]
261
262 mean((validation.data.GDP.dependent.variable - validation.GDP)^2)
263
264 #Plot of the real vs. the predicted GDP rate
265 real <- data.frame(validation.data.GDP.dependent.variable)
266
    fc <- data.frame(validation.GDP)</pre>
267
    Comp <- cbind(fc, real)</pre>
268 colnames(Comp) <- c("Predicted.GDP.Rate", "Real.GDP.Rate")
269 Quarter <- seq(1, 15, by = 1)
270
271 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
272
                                                            y = Comp$Predicted.GDP.Rate,
273
                                                            color = 'Predicted.GDP.Rate'),
274
                                             size = 1) +
275
                                  geom_line(data = real, aes(x = Quarter,
276
                                                              y = Comp$Real.GDP.Rate,
277
                                                              color = 'Real.GDP.Rate'),
278
                                             size = 1) +
279
                                  labs(x = "Quarter", y = "Rate") +
280
                                  theme(axis.title = element_text(face = "bold", size = 20))+
281
                                  scale_color_manual(name = "", values = c("red", "blue"),
282
                                  breaks = c("Predicted.GDP.Rate",
283
                                              "Real.GDP.Rate"))+
284
                                  theme(legend.text = element_text(size = 20))
285
286 #Forecast 2,5 year ahead
287
288 |lasso.var.GDP <- cbind(rep(1, each = 31), all.training) %*% as.vector(coef.GDP)
289
    ts.lasso.var.GDP <- ts(lasso.var.GDP)</pre>
290
   forecast.GDP <- forecast(ts.lasso.var.GDP, h= 10)</pre>
291
   forecast.GDP.values <- summary(forecast.GDP)$"Point Forecast"</pre>
292
293 RMSE(forecast.data.GDP.dependent.variable - forecast.GDP.values)
294 MAE(forecast.data.GDP.dependent.variable - forecast.GDP.values)
295
296 #Plot of the real vs. the forecasted GDP rate
297
    real <- data.frame(forecast.data.GDP.dependent.variable)</pre>
298 fc <- data.frame(forecast.GDP.values)
299 Comp <- cbind(fc, real)
300 colnames(Comp) <- c("Forecasted.GDP.Rate", "Real.GDP.Rate")
301 Quarter <- seq(1, 10, by = 1)
302
303 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
304
                                                            y = Comp$Forecasted.GDP.Rate,
305
                                                            color = 'Forecasted.GDP.Rate'),
306
                                             size = 1) +
```

307	<pre>geom_line(data = real, aes(x = Quarter,</pre>
308	y = Comp\$Real.GDP.Rate,
309	<pre>color = 'Real.GDP.Rate'),</pre>
310	size = 1) +
311	labs(x = "Quarter", y = "Rate") +
312	<pre>theme(axis.title = element_text(face = "bold", size = 20))+</pre>
313	<pre>scale_color_manual(name = "", values = c("red", "blue"),</pre>
314	<pre>breaks = c("Forecasted.GDP.Rate",</pre>
315	"Real.GDP.Rate"))+
316	<pre>theme(legend.text = element_text(size = 20))</pre>

R-koder/LassoVarLag.R

B.4 Bayesian Model Average

```
2 library("BMS")
3 library("ggplot2")
4
5
  6
7
  CUR.data <- as.matrix(cbind(training.data.CUR.dependent.variable,
8
                            training.data.CUR.predictors))
9
10 bms.CUR = bms(CUR.data, burn = 100000, iter = 200000, mprior = "random", g = "BRIC",
11
            nmodel = 2000, mcmc = "bd")
12
13
14 #Returns a matrix whose columns are the expected value of coefficients
15 #for the best models in a BMA object
16 best.model.CUR <- beta.draws.bma(bms.CUR[1], stdev = FALSE)
17
18 #A bma object stores several 'best' models it encounters (cf. argument nmodel in bms).
19 #as.zlm extracts a single model and converts it to an object of class zlm,
20
  #which represents a linear model estimated under Zellner's g prior.
21
22 zlm.model.CUR <- as.zlm(bms.CUR, model = 1)
23 summary(zlm.model.CUR)
24
25 #Residuals
26 residual.CUR <- residuals(zlm.model.CUR)
27
28
  #Fitted values
29 fitted.values.CUR <- fitted.values(zlm.model.CUR)
30
31 plot(residual.CUR<sup>~</sup>fitted.values.CUR,
32
      xlab = "", ylab = "")
33 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
34 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
35 abline(h = 0, col = "red")
```

```
36
  # Test for normality
37
38 shapiro.test(residual.CUR)
39
40
  #Histogram of the residuals compared with the normal-distribution
41 h <- hist(residual.CUR, xlab = "", ylab = "")
42 mtext("Index", side = 2, line = 2.2, cex = 1.5)
43 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
44 xfit <- seq(min(residual.CUR), max(residual.CUR), length = 40)
45 yfit <- dnorm(xfit, mean = mean(residual.CUR), sd = sd(residual.CUR))
46
  yfit <- yfit * diff(h$mids[1:2]) * length(residual.CUR)</pre>
47
   lines(xfit, yfit, col = "blue", lwd = 2)
48
49
  #Coefficient of determination
50 post.pr2(zlm.model.CUR) #pseudo R-squared
51
52 #Root mean squared error
53 RMSE(residual.CUR)
54
55
56
  #Mean squared prediction error
57
  validation.CUR.BMA <- predict(bms.CUR, newdata = validation.data.CUR.predictors,
58
                                  exact = TRUE, topmodels = 1)
59
60 # MSPE - prediction error
61
  mean((validation.data.CUR.dependent.variable - validation.CUR.BMA)^2)
62
63
   #Plot of the real vs. the Predicted unemplyment rate
64
  real <- data.frame(validation.data.CUR.dependent.variable)
65 fc <- data.frame(validation.CUR.BMA)
66 Comp <- cbind(fc, real)
67 colnames(Comp) <- c("Predicted.Unemployment.Rate", "Real.Unemployment.Rate")
68 Quarter <- seq(1, 15, by = 1)
69
70
   ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
71
                                                          y = Comp$Predicted.Unemployment.Rate,
72
                                                          color = 'Predicted.Unemployment.Rate'),
73
                                           size = 1) +
74
                                 geom_line(data = real, aes(x = Quarter,
75
                                                            y = Comp$Real.Unemployment.Rate,
76
                                                            color = 'Real.Unemployment.Rate'),
77
                                           size = 1) +
78
                                 labs(x = "Quarter", y = "Rate") +
79
                                 theme(axis.title = element_text(face = "bold", size = 20))+
                                 scale_color_manual(name = "", values = c("red", "blue"),
80
81
                                 breaks = c("Predicted.Unemployment.Rate",
82
                                            "Real.Unemployment.Rate"))+
83
                                 theme(legend.text = element_text(size = 20))
84
85
86
   #Forecast 2,5 year ahead
87 forecast.CUR <- predict(bms.CUR, newdata = NULL, exact = TRUE, topmodels = 1)
```

```
88
 89 RMSE(forecast.data.CUR.dependent.variable - forecast.CUR[1:10])
 90 MAE(forecast.data.CUR.dependent.variable - forecast.CUR[1:10])
 91
 92 #Plot of the real vs. the forecasted unemplyment rate
 93 real <- data.frame(forecast.data.CUR.dependent.variable)
 94 fc <- data.frame(forecast.CUR[1:10])
 95 Comp <- cbind(fc, real)
 96 colnames(Comp) <- c("Forecasted.Unemployment.Rate", "Real.Unemployment.Rate")
 97
    Quarter <- seq(1, 10, by = 1)
 98
 99
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
100
                                                          y = Comp$Forecasted.Unemployment.Rate,
101
                                                        color = 'Forecasted.Unemployment.Rate'),
102
                                           size = 1) +
103
                                 geom_line(data = real, aes(x = Quarter,
104
                                                            y = Comp$Real.Unemployment.Rate,
105
                                                            color = 'Real.Unemployment.Rate'),
106
                                           size = 1) +
107
                                 labs(x = "Quarter", y = "Rate") +
108
                                 theme(axis.title = element_text(face = "bold", size = 20))+
                                 scale_color_manual(name = "", values = c("red", "blue"),
109
110
                                 breaks = c("Forecasted.Unemployment.Rate",
111
                                              "Real.Unemployment.Rate"))+
112
                                 theme(legend.text = element_text(size = 20))
113
114
    115
116
117
    Infl.data <- as.matrix(cbind(training.data.Infl.dependent.variable,</pre>
118
                                training.data.Infl.predictors))
119
120 bms.Infl = bms(Infl.data, burn = 100000, iter = 200000, mprior = "random", g = "BRIC",
121
                  nmodel = 2000, mcmc = "bd")
122
123
124 #Returns a matrix whose columns are the expected value of coefficients
125 #for the best models in a BMA object
126 best.model.Infl <- beta.draws.bma(bms.Infl[1], stdev = FALSE)
127
128 #A bma object stores several 'best' models it encounters (cf. argument nmodel in bms).
129
    #as.zlm extracts a single model and converts it to an object of class zlm,
130
    #which represents a linear model estimated under Zellner's g prior.
131
132 zlm.model.Infl <- as.zlm(bms.Infl, model = 1)</pre>
133 summary(zlm.model.Infl)
134
135 #Residuals
136 residual.Infl <- residuals(zlm.model.Infl)
137
138 #Fitted values
139 fitted.values.Infl <- fitted.values(zlm.model.Infl)
```

```
140
141 plot(residual.Infl~fitted.values.Infl,
142
       xlab = "", ylab = "")
143 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
144 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
145 abline(h = 0, col = "red")
146
147 # Test for normality
148 shapiro.test(residual.Infl)
149
150 #Histogram of the residuals compared with the normal-distribution
   h <- hist(residual.Infl, xlab = "", ylab = "")</pre>
151
152 mtext("Index", side = 2, line = 2.2, cex = 1.5)
153 mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
154 xfit <- seq(min(residual.Infl), max(residual.Infl), length = 40)
155 yfit <- dnorm(xfit, mean = mean(residual.Infl), sd = sd(residual.Infl))
156 yfit <- yfit * diff(h$mids[1:2]) * length(residual.Infl)
157 lines(xfit, yfit, col = "blue", lwd = 2)
158
159 #Coefficient of determination
160 post.pr2(zlm.model.Infl) #pseudo R-squared
161
162 #Root mean squared error
163 RMSE(residual.Infl)
164
165
   #Mean squared prediction error
166
167
    validation.Infl.BMA <- predict(bms.Infl, newdata = validation.data.Infl.predictors,</pre>
168
                                   exact = TRUE, topmodels = 1)
169
170 # MSPE - prediction error
171 mean((validation.data.Infl.dependent.variable - validation.Infl.BMA)^2)
172
173 #Plot of the real vs. the Predicted Inflation rate
174 real <- data.frame(validation.data.Infl.dependent.variable)
175 fc <- data.frame(validation.Infl.BMA)
176 Comp <- cbind(fc, real)
177 colnames(Comp) <- c("Predicted.Inflation.Rate", "Real.Inflation.Rate")
178 Quarter <- seq(1, 15, by = 1)
179
180 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
181
                                                           y = Comp$Predicted.Inflation.Rate,
182
                                                           color = 'Predicted.Inflation.Rate'),
183
                                            size = 1) +
184
                                  geom_line(data = real, aes(x = Quarter,
185
                                                            y = Comp$Real.Inflation.Rate,
186
                                                             color = 'Real.Inflation.Rate'),
187
                                            size = 1) +
                                  labs(x = "Quarter", y = "Rate") +
188
189
                                  theme(axis.title = element_text(face = "bold", size = 20))+
                                  scale_color_manual(name = "", values = c("red", "blue"),
190
191
                                 breaks = c("Predicted.Inflation.Rate",
```

```
192
                                            "Real.Inflation.Rate"))+
193
                                 theme(legend.text = element_text(size = 20))
194
195
196
    #Forecast 2,5 year ahead
197
    forecast.Infl <- predict(bms.Infl, newdata = NULL, exact = TRUE, topmodels = 1)</pre>
198
199 RMSE(forecast.data.Infl.dependent.variable - forecast.Infl[1:10])
200 MAE(forecast.data.Infl.dependent.variable - forecast.Infl[1:10])
201
202
    #Plot of the real vs. the forecasted inflation rate
203
    real <- data.frame(forecast.data.Infl.dependent.variable)</pre>
204 fc <- data.frame(forecast.Infl[1:10])
205 Comp <- cbind(fc, real)
206 colnames(Comp) <- c("Forecasted.Inflation.Rate", "Real.Inflation.Rate")
207
    Quarter <- seq(1, 10, by = 1)
208
209
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
210
                                                          y = Comp$Forecasted.Inflation.Rate,
211
                                                          color = 'Forecasted.Inflation.Rate'),
                                           size = 1) +
212
213
                                 geom_line(data = real, aes(x = Quarter,
214
                                                            y = Comp$Real.Inflation.Rate,
215
                                                            color = 'Real.Inflation.Rate'),
216
                                           size = 1) +
217
                                 labs(x = "Quarter", y = "Rate") +
218
                                 theme(axis.title = element_text(face = "bold", size = 20))+
                                 scale_color_manual(name = "", values = c("red", "blue"),
219
220
                                 breaks = c("Forecasted.Inflation.Rate",
221
                                            "Real.Inflation.Rate"))+
222
                                 theme(legend.text = element_text(size = 20))
223
224
225
    226
227
    GDP.data <- as.matrix(cbind(training.data.GDP.dependent.variable,
228
                                 training.data.GDP.predictors))
229
    bms.GDP = bms(GDP.data, burn = 100000, iter = 200000, mprior = "random", g = "BRIC",
230
231
                   nmodel = 2000, mcmc = "bd")
232
233
234
    #Returns a matrix whose columns are the expected value of coefficients
235\ \mbox{\#for} the best models in a BMA object
236 best.model.GDP <- beta.draws.bma(bms.GDP[1], stdev = FALSE)
237
238 #A bma object stores several 'best' models it encounters (cf. argument nmodel in bms).
239 #as.zlm extracts a single model and converts it to an object of class zlm,
240 #which represents a linear model estimated under Zellner's g prior.
241
242 zlm.model.GDP <- as.zlm(bms.GDP, model = 1)
243 summary(zlm.model.GDP)
```

```
244
245 #Residuals
246 residual.GDP <- residuals(zlm.model.GDP)
247
248 #Fitted values
249 fitted.values.GDP <- fitted.values(zlm.model.GDP)
250
251 plot(residual.GDP~fitted.values.GDP,
252
        xlab = "", ylab = "")
253 mtext("Residuals", side = 2, line = 2.2, cex = 1.5)
254 mtext("Fitted Value", side = 1, line = 2.2, cex = 1.5)
255
    abline(h = 0, col = "red")
256
257
   # Test for normality
258 shapiro.test(residual.GDP)
259
260 #Histogram of the residuals compared with the normal-distribution
261 h <- hist(residual.GDP, xlab = "", ylab = "")
262 mtext("Index", side = 2, line = 2.2, cex = 1.5)
263
   mtext("Residuals", side = 1, line = 2.2, cex = 1.5)
264 xfit <- seq(min(residual.GDP), max(residual.GDP), length = 40)
265 yfit <- dnorm(xfit, mean = mean(residual.GDP), sd = sd(residual.GDP))
266 yfit <- yfit * diff(h$mids[1:2]) * length(residual.GDP)
267 lines(xfit, yfit, col = "blue", lwd = 2)
268
269 #Coefficient of determination
270
   post.pr2(zlm.model.GDP) #pseudo R-squared
271
272 #Root mean squared error
273 RMSE(residual.GDP)
274
275
276 #Mean squared prediction error
277 validation.GDP.BMA <- predict(bms.GDP, newdata = validation.data.GDP.predictors,
278
                                  exact = TRUE, topmodels = 1)
279
280 # MSPE - prediction error
281 mean((validation.data.GDP.dependent.variable - validation.GDP.BMA)^2)
282
283 #Plot of the real vs. the Predicted GDP rate
284 real <- data.frame(validation.data.GDP.dependent.variable)
285
   fc <- data.frame(validation.GDP.BMA)</pre>
286
   Comp <- cbind(fc, real)</pre>
    colnames(Comp) <- c("Predicted.GDP.Rate", "Real.GDP.Rate")</pre>
287
288 Quarter <- seq(1, 15, by = 1)
289
290 ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
291
                                                           y = Comp$Predicted.GDP.Rate,
292
                                                           color = 'Predicted.GDP.Rate'),
293
                                            size = 1) +
294
                                  geom_line(data = real, aes(x = Quarter,
295
                                                             y = Comp$Real.GDP.Rate,
```

```
296
                                                              color = 'Real.GDP.Rate'),
297
                                            size = 1) +
298
                                  labs(x = "Quarter", y = "Rate") +
299
                                  theme(axis.title = element_text(face = "bold", size = 20))+
300
                                  scale_color_manual(name = "", values = c("red", "blue"),
301
                                  breaks = c("Predicted.GDP.Rate",
302
                                             "Real.GDP.Rate"))+
303
                                  theme(legend.text = element_text(size = 20))
304
305
306
    #Forecast 2,5 year ahead
307
    forecast.GDP <- predict(bms.GDP, newdata = NULL, exact = TRUE, topmodels = 1)</pre>
308
309
    RMSE(forecast.data.GDP.dependent.variable - forecast.GDP[1:10])
310 MAE(forecast.data.GDP.dependent.variable - forecast.GDP[1:10])
311
312 #Plot of the real vs. the forecasted GDP rate
313 real <- data.frame(forecast.data.GDP.dependent.variable)
314 fc <- data.frame(forecast.GDP[1:10])
315
    Comp <- cbind(fc, real)</pre>
316 colnames(Comp) <- c("Forecasted.GDP.Rate", "Real.GDP.Rate")
317
    Quarter \leq seq(1, 10, by = 1)
318
319
    ggplot(Comp, aes(Quarter)) + geom_line(data = fc, aes(x = Quarter,
320
                                                            y = Comp$Forecasted.GDP.Rate,
321
                                                            color = 'Forecasted.GDP.Rate'),
322
                                            size = 1) +
323
                                  geom_line(data = real, aes(x = Quarter,
324
                                                             y = Comp$Real.GDP.Rate,
325
                                                             color = 'Real.GDP.Rate'),
326
                                            size = 1) +
                                  labs(x = "Quarter", y = "Rate") +
327
328
                                  theme(axis.title = element_text(face = "bold", size = 20))+
                                  scale_color_manual(name = "", values = c("red", "blue"),
329
330
                                  breaks = c("Forecasted.GDP.Rate",
331
                                             "Real.GDP.Rate"))+
332
                                  theme(legend.text = element_text(size = 20))
```

R-koder/BMA(BMS).R

B.5 Empirical Conclusion

```
9 forecast.GDP.benchmark <- forecast(GDP.benchmark, a.head = 10)
10 forecast.GDP.benchmark.values <- summary(forecast.GDP.benchmark)$"Point Forecast"
11
  forecast.error.GDP.benchmark <- forecast.data.GDP.dependent.variable -</pre>
12
                                  forecast.GDP.benchmark.values
13
14
   ## FAVAR
15
16
  forecast.error.GDP.FAVAR <- forecast.data.GDP.dependent.variable - forecast.GDP
17
  dm.test(forecast.error.GDP.benchmark, forecast.error.GDP.FAVAR, alternative = "greater",
18
19
          h=10)
20
21
   22
23
  CUR.benchmark <- ar(training.data.CUR.dependent.variable, method = "ols")
24 forecast.CUR.benchmark <- forecast(CUR.benchmark, a.head = 10)
25 forecast.CUR.benchmark.values <- summary(forecast.CUR.benchmark)$"Point Forecast"
  forecast.error.CUR.benchmark <- forecast.data.CUR.dependent.variable -
26
27
                                  forecast.CUR.benchmark.values
28
29
   ## Lasso-Var
30
31
  forecast.error.CUR.LassoVar <- forecast.data.CUR.dependent.variable - forecast.CUR.values
32
33
   dm.test(forecast.error.CUR.benchmark, forecast.error.CUR.FAVAR, alternative = "greater",
34
          h=10)
35
36
37
   38
39
  Infl.benchmark <- ar(training.data.Infl.dependent.variable, method = "ols")
40
  forecast.Infl.benchmark <- forecast(Infl.benchmark, a.head = 10)</pre>
  forecast.Infl.benchmark.values <- summary(forecast.Infl.benchmark )$"Point Forecast"</pre>
41
42
  forecast.error.Infl.benchmark <- forecast.data.Infl.dependent.variable -</pre>
43
                                   forecast.Infl.benchmark.values
44
45
   ## FAVAR
46
  forecast.error.Infl.FAVAR <- forecast.data.Infl.dependent.variable - forecast.Infl
47
48
49
  dm.test(forecast.error.Infl.benchmark, forecast.error.Infl.FAVAR, alternative = "greater",
50
          h=10)
```

R-koder/Benchmark.R