On the Use of Machine Learning Techniques in the Forecasting of Macro Time Series

Marco Goretti

Department of Economics
HEC Lausanne

Supervisor: **Prof. Jean-Paul Renne**

A thesis submitted in partial fulfillment of the requirements for the degree of
Master in Economics

June 2017

Abstract

This master thesis compares the performance of several macroeconomic forecasting methods to methods coming from machine learning. The macroeconomic methods are standard ones, such as ARMA and VAR, as well as more advanced techniques, such as Markov Switching regression. The machine learning methods are a custom algorithm, named Split regression, that mixes clustering and classification in order to imitate the states of the Markov switching regression and neural networks under the form a multilayer perceptron. The methods are run on univariate, multivariate and factor-augmented data. The performance is evaluated on the forecast of the percentage change from last period of quarterly GDP, monthly CPI and monthly non-farm employment. The forecasts are performed for 1, 4 and 12 periods ahead predictions.

The use of factor-augmented data showed a clear improvement in the forecast of GDP, as well as showing an improvement for state-based models in the forecast of employment. They did not improve the performance of the forecast of the CPI.

State-based models (Markov switching regression and Split regression) are the top overall performers, which suggests the usefulness of the use of more advanced methods that simple VAR. However, the small quantity of data did not favour neural networks, which performed mediocrely.

https://github.com/mgoretti/master-thesis

Table of contents

Chapter 1

Introduction

Forecasting time series with a good accuracy plays an important role in a large number of domains. The obvious example is finance, where having a better prediction than the market allows to make a profit. Another domain, closely related to finance, where forecasting plays an important role is economics. More precisely macro-economics. The forecast of variables such as the GDP and the inflation is very important for both the central banks, which need to know the impact of a change of their influenced variables (such as the interest rate) in order to reach their targets. This allows them to keep the economy in the path they expect. It is also very important for the actors which adapt their structure to the expected future state of the economy.

Several papers, such as Mullainathan and Spiess (2017) and Kim et al. (2004) find interesting applications of machine learning methods (such as neural networks and random forests of decision trees) to economic problems. In regard to forecasting, previous literature findings are mixed: Tkacz and Hu (1999) and Moshiri and Cameron (2000) find neural networks useful in the forecasting of GDP and Nakamura (2005) finds them useful for forecasting inflation. However, in Stock and Watson (1998) they perform worse than several other methods in the univariate case.

The goal of this thesis is to study the utility of machine learning methods in the forecasting of macro-economics time series. However, one may wonder what machine learning means. Andrew Ng defines machine learning as "the science of getting computers to act without being explicitly programmed.". This definition is as broad as the field itself. However, only the methodology related to predicting continuous variables, such as regression analysis, is of interest for this work. It is a subset of supervised learning which is defined as "the machine learning task of inferring a function from labeled training data" by Mohri et al. (2012)[1].

The real difference between machine learning and standard econometrics is that machine learning only cares about the generalisation of the method to new data, which is evaluated

[1] In this context, labeled data means that the dependent variable is available.

as the out-of-sample performance of the model. Methods that operate as black boxes are not a problem in machine learning since only the final output is of interest. On the other hand, most econometric applications care about identification and statistical testing: this requires a theoretical foundation of the models in order to estimate the errors. A striking example of difference in the two methodologies is the bias: in regressions, econometrics tries to avoid bias by adding new variables while machine learning, with methods such as the ridge regression, actually adds bias in order to reduce the variance of the estimator which allows a better prediction. Of course, the coefficients of that method cannot be identified as the coefficients of the real process as the estimator is not consistent. However, if the goal is the better prediction, it may be worth using that approach. In the context of forecasting, this methodology is exactly what is wanted since only the performance of the prediction is of interest.

This thesis compares the performance of methods built and tuned using machine learning tools with methods commonly used in forecasting, such as ARIMA and VAR, as well as the more sophisticated Markov Switching regression (Hamilton (1990) and Hamilton (1994)). Multivariate models are run on both standard variables but also factors, yielding factor-augmented data, as is done in Bernanke et al. (2005) and Stock and Watson (2016). The machine learning methods used are neural networks implemented as a multilayer perceptron and a custom method that mixes clustering and classification to replicate the multi-state concept of the Markov Switching regression.

First the forecasted time series are presented, then the general methodology is explained. After that, the different methods used are explained and finally the performance results are commented.

Chapter 2

Experimental Framework

2.1 Data

The two presented datasets aim to cover the behaviour of the different methods when confronted with datasets of small size (the quarterly indicators), which should favour easier linear methods and of medium size (the monthly dataset) where more complex non-linear methods can profit from the bigger quantity of data. However, even in the monthly case, the total amount of data is pretty small compared to what is usually used to train neural networks.

Quarterly US Macroeconomics Indicators

The dataset is composed of 6 quarterly main variables going from 01.01.1959 to 01.10.2014 (223 observations): real GDP, total non-farm employment, the unemployment rate, a consumer price index (CPI), the effective federal funds rate and the industrial production. All the variables are with seasonality removed. Full descriptive statistics can be found in appendix C. The factors used by Stock and Watson (2016) were also added as they are supposed to summarise the general state of the economy and are used for the FAVAR data generation. Those factors are generated from a dataset of 201 quarterly variables. The variables have been transformed to make them stationary as describe in appendix D. A few variables with missing values where removed.

Monthly US Macroeconomics Indicators

The dataset is composed 5 monthly main variables going from 01.01.1959 to 01.11.2014 (671 observations): total non-farm employment, the unemployment rate, a consumer price index (CPI), the effective federal funds rate and the industrial production. All the variables

are with seasonality removed. Full descriptive statistics can be found in appendix C. Factors are generated from 133 monthly variables of the above dataset.

2.2 Forecasted variables

GDP

The percentage change of GDP from last period is used as the quarterly predicted time series.

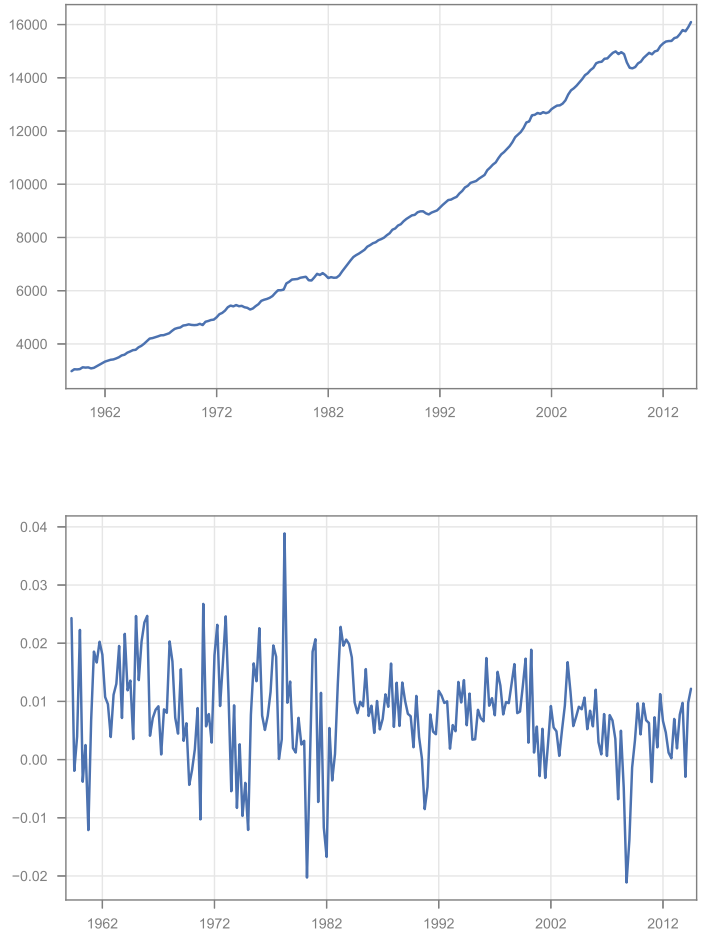

Fig. 2.1 GDP and percentage change of GDP from last period

CPI

The percentage change of CPI from last period is used as one of the monthly predicted time series.

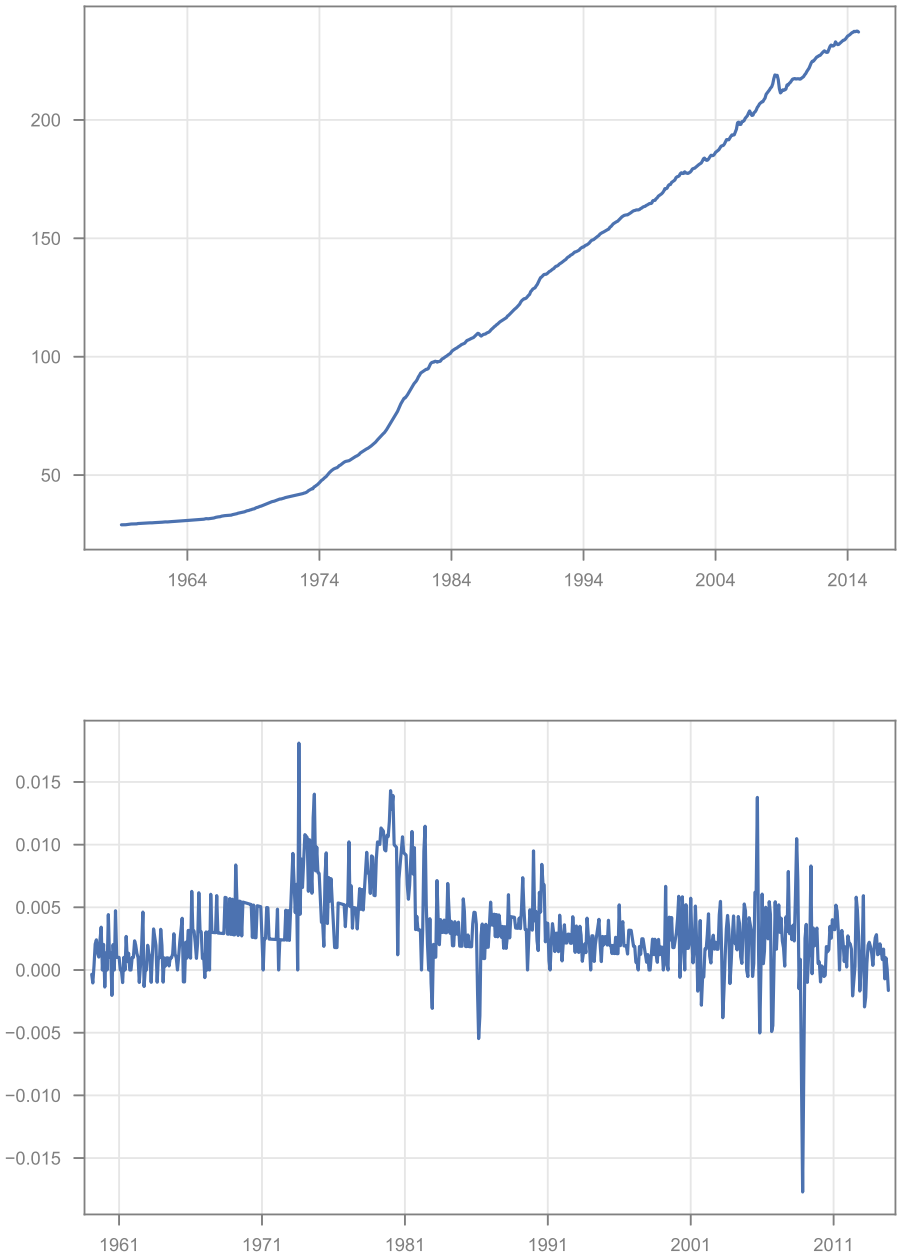

Fig. 2.2 CPI and percentage change of CPI from last period

Employment

The percentage change of non-farm employment from last period is used as the other monthly predicted time series.

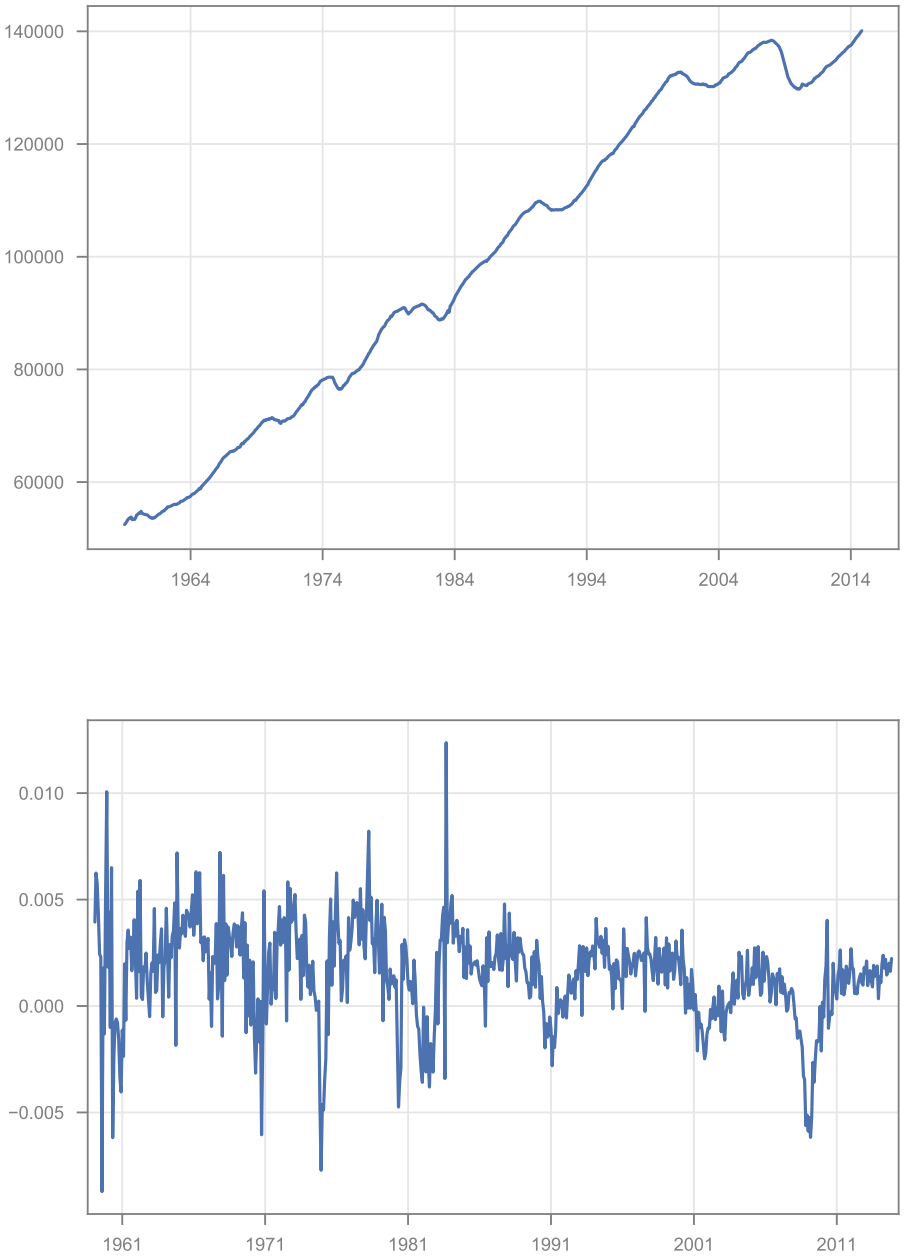

Fig. 2.3 Non-farm employment and percentage change of employment from last period

Chapter 3

Methodology

In this chapter, an overview of the prediction methodology is given by first presenting the easier task of predicting cross-sectional data and its performance scoring, then generalising the setup to time series.

3.1 The task of predicting

The idea behind predictions is to use already available data, for which we already know some wanted characteristic (that is for example measured a posteriori) to make assumptions on new data points for which we do not know that wanted characteristic.

We can define the wanted process as

$$f : \boldsymbol{X} \mapsto y$$

Where $\boldsymbol{X}$ is a vector of independent variables (variables observed a priori) and y is the dependent variable that we want to predict by approximating the mapping f which gives the real value of y.

We can separate the situation in two different cases depending on the nature of y.

Classification

y is discrete: we want to classify the observation into one of several classes. In this case, we are interested about the accuracy of our prediction, namely how often the correct class is found. No distinction is made between wrong classes (there is not a class that is less wrong than another).

Regression

y is continuous: we want to approximate the value of y. In this case, we are interested about how close the prediction is to the real value. The metric used is some function of the distance between the prediction and the real value, such as the mean absolute error (MAE) or, most of the time, the root mean squared error (RMSE).

3.2 Predicting Cross-sectional Data

The great advantage of cross-sectional data compared to time series is that they are not ordered since every observation is supposed to be independent from another [1] which allows the use of methods such as the cross-validation to validate models.

The need for a more sophisticated model than just training the model on all data and then reducing the error of the prediction on the same data (called training error) to validate the model comes from the problem known as overfit.

Bias-Variance Decomposition

If we consider the process

$$y = f(\mathbf{x}) + \varepsilon$$

Where ε is the noise of the process with variance σ^2.

The prediction error made by the model on a dataset that was not part of the training set, which is the error we want to minimise, can be decomposed in 3 parts:

- σ^2, the irreducible error caused by the noise in the real process
- $\mathrm{Var}[\hat{f}]$, the variance of the estimator of f
- $\mathrm{E}[f - \hat{f}]^2$, the squared bias of the estimator

Such that

$$\mathrm{E}[\mathrm{MSE}] = \mathrm{E}[(y - \hat{f})^2] = \sigma^2 + \mathrm{Var}[\hat{f}] + \mathrm{Bias}[\hat{f}]^2$$

[1] Even in the case of clustered data this only affects the errors and, thus, does not make it important for predictions.

This decomposition shows the basic problem of predictions: the complexity of the model has an opposite effect on the two non-irreducible components. The more the model is complex, the more it can vary to fit the training data (even the noise, as it can be seen in figure 3.1), thus increasing its variance. Conversely, the simpler the model, the less it can adapt to fit the real form, thus increasing the bias.
There is an optimal value for each hyper-parameter of the model, at which the error is minimal, as illustrated by figure 3.2

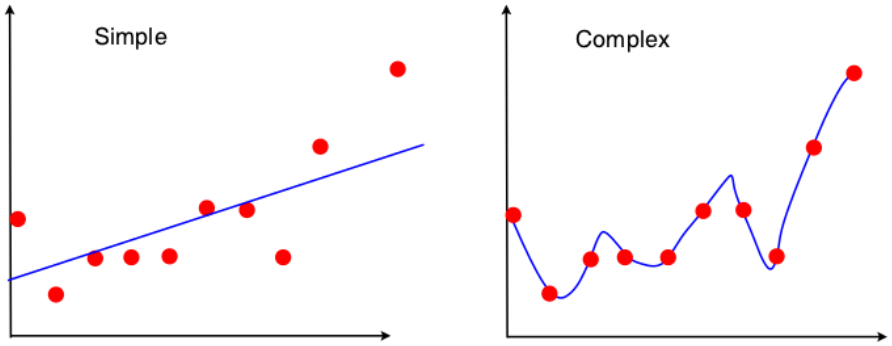

Fig. 3.1 The left model fits the real process, while the right model is overfitting

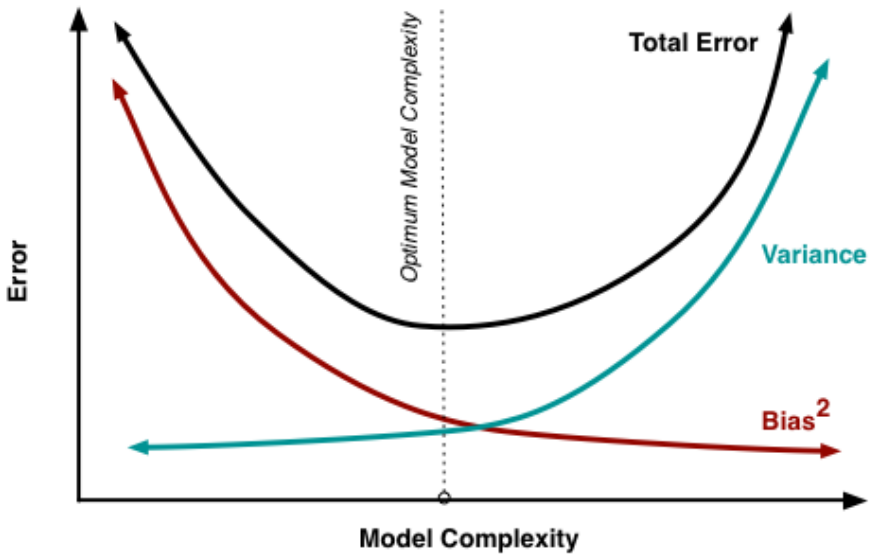

Fig. 3.2 Evolution of the decomposition of the prediction error as a function of the complexity

If the model is not complex enough (left side of figure 3.2), it is underfitting, This is caused by a bias increase that is bigger than the decrease in variance. It results in an error higher than the optimum. If the model is too complex, the opposite happens and the increase in

variance does not justify the decrease in bias anymore (right side of figure 3.2). This also results in an error higher than the optimum. A visualisation of the two phenomena can be seen in figure 3.3.

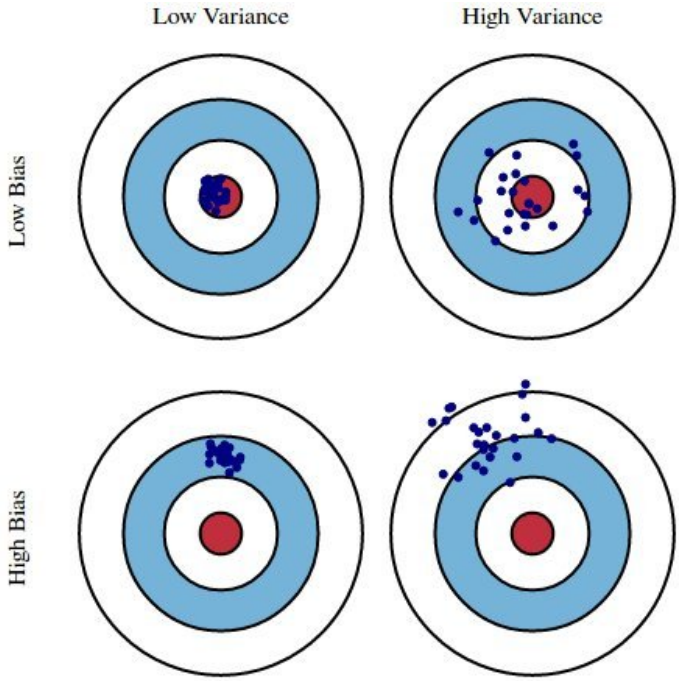

Fig. 3.3 Visualisation of the effect of bias and variance

The above mentioned overfit problem is only observed when testing on unseen data, as those have different values of noise than the one the model is fitted on. This results in a train error that keeps decreasing the greater the complexity of the model, while the error on an unseen dataset starts increasing after a certain point, as it can be seen in figure 3.4. A simple example is the R^2 of an OLS that keeps decreasing the more variables are added to the model. To circumvent this problem, a method called cross-validation (CV) is used.

3.2.1 Cross-Validation

Since only the training data is available, a way to still be able to test how the model performs on unseen data is to hide some of the data from it and then use it to assess its performance. The simplest way of doing this, called hold-out cross-validation, is by randomly picking (without replacement) a certain number of observations as the test data and using the rest to train the model. A commonly used value is 20% or 30% of the dataset as test data.

This process can be summarised as such:

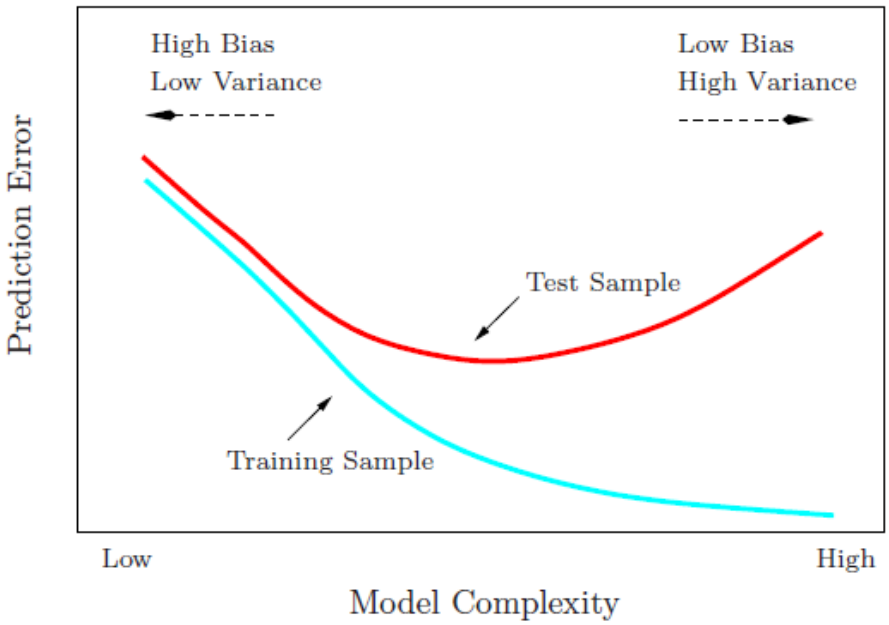

Fig. 3.4 The test error decreases until the optimum, where it stopped underfitting, but increases again after that because it starts overfitting

- Train the model on $\sim 70\%$ of the data (train dataset)
- Test the performance on the remaining $\sim 30\%$ of the data (test dataset)

However, two problems still exist. First, we are evaluating the performance of a model that is trained on only $70-80\%$ of the total dataset, while in practice it will be trained on 100%. Also, we could have been unlucky with the splits and over/underrepresented some special pattern in the testing data. To avoid so, an improvement to the hold-out CV can be made by performing the same process, but this time only taking one observation as testing and repeating the train-test process for every observation and, finally, averaging the obtained score. This is called **leave-one-out CV**. However, as it can be expected, this process is very resource consuming and in practice a compromise between the two methods is used.

k-fold cross-validation

This method uses the same idea of splitting into train and test and averaging the performance over all splits. However, instead of having N splits, it only has k (usually 5 or 10). The splits are illustrated in figure 3.5.

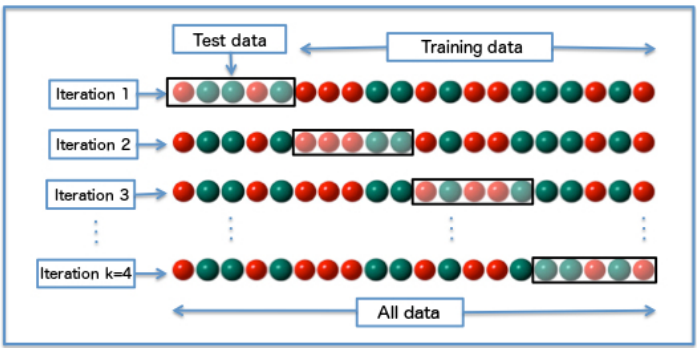

Fig. 3.5 Splits into train and test performed by a 4-fold cross-validation

3.2.2 Model Selection and Parameter Tuning

To best approximate real life situations, when assessing the best model to use, the data is setup as such:

- **Training set:** Used to train the final model
- **Dev set:** Used to select the best parameters (variables, but also hyper-parameters for more complex models), operation that is called model tuning
- **Test set:** used only to compare the performance of the final models. It is never used to tune the models. It emulates a real life situation where the new data has never been seen before

Most of the time, the dev and training set are the same since it allows to work with the biggest quantity of data.

Inside the dev set, parameter tuning is done as such:

- Split the dataset in a train set and a validation set (by using either k-fold CV or leave-one-out CV).
- Select a range of values for a parameter we want to tune.
- Train and validate the model for each fold and each value in the range.
- Find the value of the parameter that gives the lowest error (usually performed by either finding a global minimum, as shown in figure 3.4, or by generating a distribution of each error and then comparing the distributions of errors. This can be done, for example, by performing t-tests or by having some special loss function (that takes into account the variance of the error).

3.3 Predicting Time Series

Time series come with an added constraint: the data is ordered and comes from the same phenomenon at different times instead of having different things observed (different individuals, companies, country, etc.) at the same time. This causes the hypothesis of the observations being independent to never hold anymore. The influence of this is very strong as, very often, the best predictor of an explained value at a given time is the same variable in the previous period. This is called a **lagged variable**. Most of the time, several lags are used.

The described prediction process here could be to identify some model of the form:

$$y_t = \beta_1 y_{t-1} + \cdots + \beta_k y_{t-k} + \Gamma_0 X_t + \Gamma_1 X_{t-1} + \cdots + \Gamma_k X_{t-k} + \varepsilon_t$$

However, this assumes that the only variable we do not know at time t is y, while still being able to identify it in the previous period (or a few periods before, without loss of generality). This can happen with variables that are not immediate to identify but is not the most interesting application of prediction methods.

3.3.1 Forecasting Time Series

One can write a model such as:

$$y_t = \beta_1 y_{t-1} + \cdots + \beta_k y_{t-k} + \Gamma_1 X_{t-1} + \cdots + \Gamma_k X_{t-k} + \varepsilon_t$$

In this case, the value of y_t only depends on past values, thus we can predict the future based on present values. This is very interesting as it can be used to predict the future of the economy of countries, but also the stock market.

This can be taken one step further to predict more than one period ahead, namely

$$y_{t+h} = f(\boldsymbol{x_t}) + \varepsilon_{t+h}$$

This can be done in two different ways.

Direct h-step ahead prediction

Directly predict the value several periods ahead from the available data until t (this means that a different model is trained for each i-step ahead prediction):

$$y_{t+h} = \beta_0 y_t + \cdots + \beta_k y_{t-k} + \Gamma_0 X_t + \cdots + \Gamma_k X_{t-k} + \varepsilon_{t+h}$$

Dynamic Forecasting

Predict each variable one step ahead and keep iterating until $t+h$. This is simple in the case of univariate models (such as the AR model) where there is only one variable to propagate forward but can become very complicated with multivariate models that need a separate tuning for each variable[2].

3.3.2 Performance Evaluation

Given that the hypothesis that the observations are i.i.d does not hold anymore, as they are ordered, we cannot simply shuffle and split the observations in train and test datasets, as is done with cross-validation. However, the idea of training on train data and predicting unseen data still holds and is called out-of-sample forecasting [3]. The process is to set a date n at which the data will be split. Everything before n is going to be used to train the model and everything after is used to test it by evaluating it's performance. However, given that the most useful information is usually in the periods before the prediction (which is available in real life), not using all the data until the prediction would be unfair to some models. The correct process consists in using the information until time m when predicting the $m+h$ for the wanted h steps ahead forecast. It has two common implementations:

Rolling Window

Set a window of size D and always use the observations from $m-D$ to $m-1$ when predicting at time m, for $m \in [n, T]$. The different h-step ahead predictions will the be put together and used as different time series when computing the prediction error.

[2] In the case of the VAR model it is direct since there is no tuning to do but, for example, with neural networks it becomes very time consuming.

[3] The literature uses the definition of in-sample data for the train data and out-of-sample data for the test data. For simplicity, the train and test terminology will be kept.

Expanding Window

The idea is basically the same as the rolling window but instead of only using the last D observations, use all the information from the beginning.

3.3.3 Loss Function

The rolling or expanding window is going to give a time series for each h-step ahead prediction. This prediction will be named $\{\hat{y}_t\}_{t=n}^{T}$ for $N = T - n$ predictions. This prediction can be used to assess the performance score by comparing it to the real value of y_t with a loss function. The following loss functions are approached in this thesis. For the sake of simplicity, the time series boundaries are moved be from $[n, T]$ to $[1, N]$.

Root-mean-square error (RMSE)

The most common loss function is the **R**oot-**M**ean-**S**quare **E**rror (RMSE) as it is equivalent to the metric optimised by ordinary least squares (the mean square error). It is computed as

$$\mathrm{RMSE}(\hat{y}, y) = \sqrt{\frac{1}{N}\sum_{n}^{N}(\hat{y}_n - y_n)^2}$$

This loss function gives a high weight to large prediction errors. This is sometimes a wanted result but most of time simply makes the score more sensible to outliers. Its strict convexity makes MSE the favourite loss function for the training of methods with a variant of gradient descent (for training regressions, neural networks, etc.).

Mean absolute error (MAE)

The **M**ean **A**bsolute **E**rror is another very commonly used loss function that is less sensible to outliers. It is computed as

$$\mathrm{MAE}(\hat{y}, y) = \frac{1}{N}\sum_{n}^{N}|\hat{y}_n - y_n|$$

3.3.4 Diebold-Mariano test

As presented in Diebold and Mariano (1995), the Diebold-Mariano test allows to test the null hypothesis of no difference in the performance of two forecasts for a given loss function, g.

We define the loss-differential time series as the difference between the loss of the two tests forecasts at each period

$$d_t = g(\hat{y_{1,t}}, y_t) - g(\hat{y_{2,t}}, y_t)$$

Under the assumption that the loss-difference is covariance stationary and short memory, its distribution can be written as

$$\sqrt{T}(\bar{d} - \mu) \xrightarrow{d} \mathcal{N}(0, 2\pi f_d(0))$$

Where $f_d(0)$ is the spectral density of d at frequency 0 which is given by

$$f_d(0) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} \gamma_d(\tau)$$

Where $\gamma_d(\tau)$ is the autocovariance of difference τ (which does not depend on t given the covariance stationarity of d)

$$\gamma_d(\tau) = \mathrm{E}[(d_t - \mu)(d_{t-\tau} - \mu)]$$

This yields the final distribution of d:

$$\sqrt{T}(\bar{d} - \mu) \xrightarrow{d} \mathcal{N}(0, \sum_{\tau=-\infty}^{\infty} \gamma_d(\tau))$$

A simple Z-test can be performed by using the test statistic

$$\mathrm{DM} = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{T}}}$$

The long-run variance can be estimated using the consistent Newey-West estimator which only has non-negative estimates

$$2\pi \hat{f}_d(0) = \hat{\gamma}_0 + 2\sum_{v=1}^{q}\left(1 - \frac{v}{q+1}\right)\hat{\gamma}_v$$

where the sample autocovariance is computed as

$$\hat{\gamma}_\tau = \frac{1}{T}\sum_{\tau+1}^{T}(y_t - \bar{y})(y_{t-\tau} - \bar{y})$$

The value of q is chosen based on the hypothesis that in a h-step-ahead forecast, the forecast errors are at most $h-1$ dependent. Thus, only the covariances until $h-1$ need to be computed since the others will be 0.

$$q = h - 1$$

3.4 Transformations

Several transformations can be applied to the data. However, those will be reverted on the final data so that the performance on the raw data (what is really observed) can be evaluated.

The process of a transformation is:

- Set the parameters of the transformation on the train data
- Perform the transformation on the train data
- Train the model
- Perform the same transformation on the test data (using the parameters of the train data)
- Predict the dependent variable of the test dataset
- Invert the transformation
- Evaluate the performance

Transformations are pushed in a stack when applied and popped when inverted (meaning that the last transformation applied is the first one inverted).

3.4.1 i-th order difference

The transformation consists in computing the absolute difference between periods, for each order of difference, the difference is recomputed based on the output of the previous one.

first order difference: $\Delta X_t = X_t - X_{t-1} = (1-L)X_t$

second order difference: $\Delta(\Delta X_t) = \Delta X_t - \Delta X_{t-1} = (1-L)^2 X_t$

i-th order difference: = $\Delta^i X_t = (1-L)^i X_t$

Inverting

The inversion requires reconstructing all the differences using the real data up to the previous period and then propagating the reconstruction. To rebuild the value $\hat{X}_{t+1}$ (which assumes knowing the values up to t:

$$\Delta^{i-1}\hat{X}_{t+1} = \Delta^{i-1}X_t + \Delta^i \hat{X}_{t+1}$$

This can be iterated until the original non-differentiated value.

3.4.2 Function Application

The transformation consists in applying any invertible function. Most of the time it is the log which allows to approximate a growth. It can easily be inverted by applying the inverse function (such as the exponential in the case of the log).

3.4.3 Normalisation

The transformation scales every variable to have the same range going from -1 to 1. It is performed by computing the minimum and maximum values x_{min} and x_{max} of the variable in the train dataset.

The transformation is then given by:

$$x' = \frac{x - \frac{x_{max}+x_{min}}{2}}{\frac{x_{max}-x_{min}}{2}}$$

3.4.4 Standardisation

This transformation is an alternative to the above normalisation. It consists in the same concept of rescaling the data. However, this time it is done to obtain a mean of 0 and a standard deviation of 1. It is performed by computing the mean μ and standard deviation σ of the variable in the train dataset.

The transformation is then given by:

$$x' = \frac{x - \mu}{\sigma}$$

Both methods should be applied on all variables and have the objective of making them have the same importance in the sense that only the relative variation in the variable will count: a small relative change in a variable with a wide range is not going to be more important than a big relative change in a variable with a small range once they are rescaled. This also allows the optimisation algorithm (such as gradient descent) to converge faster as it is more likely to point a local optimum as shown in figure 3.6. Finally, standardisation is often advised over normalisation as the later is very sensible to outliers. However, for completeness, the performance of both transformations is checked.

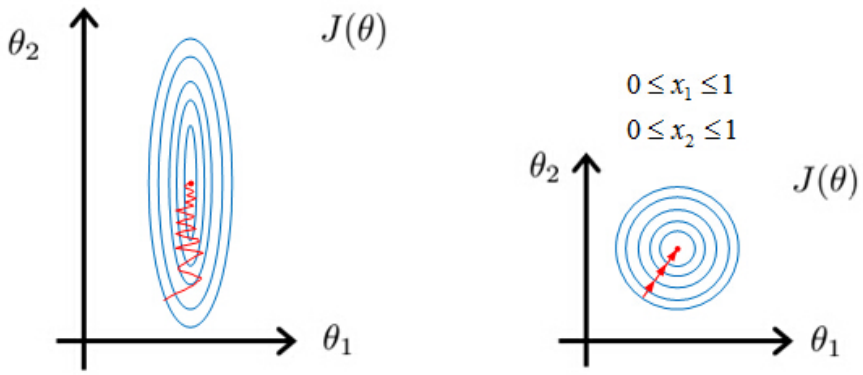

Fig. 3.6 The gradient in the right-hand plot points closer to the optimum which allows the gradient descent to converge faster

Chapter 4

Methods

In this chapter, the theory behind the methods used in this thesis is explored.

4.1 General Tools

In this section, an overview of generic methods and concepts that are used in several final models is given.

4.1.1 AR

The **A**uto**R**egressive model of order p, AR(p), writes the dependent variable as a linear combination of its past values.

It can be written as

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t$$

4.1.2 MA

The **M**oving-**A**verage model of order q, MA(q), writes the dependent variable as a linear combination of the previous shocks. It can be seen as a finite impulse response filter.

It can be written as:

$$x_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

4.1.3 Principal Component Analysis (PCA)

Principal **C**omponent **A**analysis is a dimensionality reduction method which projects a dataset into an orthogonal basis which is composed of the basis vectors that maximise the variance of the projected dataset for each ordered subset of basis vectors used as basis.

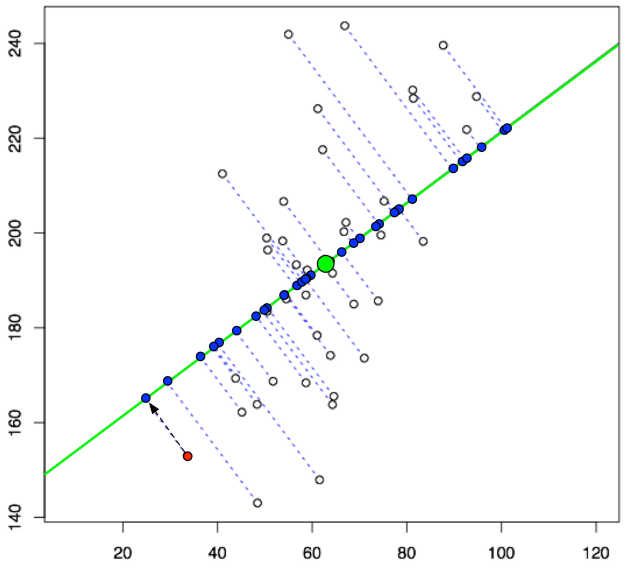

Fig. 4.1 Example of 2D to 1D projection on the first principal component

Estimation

For a standardised matrix of observations X. The covariance matrix is

$$A = \mathrm{Cov}(X) = X^T X$$

Given that the covariance matrix is symmetric, the spectral theorem says that it allows a spectral decomposition

$$A = ODO^T$$

Where D is the diagonal matrix of eigenvalues and O is the orthogonal matrix of eigenvectors. Furthermore, the variance represented by each eigenvector simply consists in its eigenvalue as proved below:

For a matrix $Y = XO$, which represents the projection of X on the eigenvectors:

$$\mathrm{Var}(Y) = Y^T Y = (XO)^T XO = O^T \underbrace{X^T X}_{\mathrm{Var}(X)} O = \cancel{O^T} \cancel{O} D \cancel{O^T} \cancel{O} = D$$

This allows to say that to get the next highest variance basis vector, the remaining eigenvector with the biggest eigenvalue must be chosen.

Finally, a subset of the ordered eigenvectors, $\hat{O}$ is chosen as the projection basis. Giving the projected dataset

$$X' = X\hat{O}$$

In practice, the eigendecomposition can be performed without computing the covariance matrix as this causes precision problems. The process used is the Singular Value Decomposition (SVD).

4.1.4 Clustering

Clustering is an unsupervised learning technique which attributes labels to unlabelled observations by grouping them is classes based on their similarity. The most common similarity metric is the euclidean distance computed using the different variables.

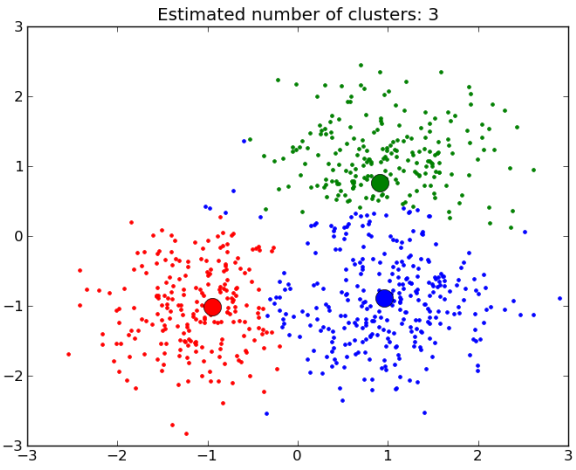

Fig. 4.2 Example of clustering obtained with k-mean. The bigger dot is the centroid of each cluster

Performance metrics

Within-cluster sum of squares: to characterise how similar the elements in clusters are, the squared distance between each point in a cluster is computed for each cluster:

$$\sum_{\text{Cluster } C_i} \sum_{x,y \in C_i} \|x-y\|^2$$

Silhouette score: this value is a comparison between the cohesion of clusters (how similar observations are in their cluster) and their separation (how different observation are when compared to other clusters). The score is the average of the values, with each value ranging from -1 to 1, where a high value indicates that the observation is similar to others in its cluster and different from observations in other clusters, while a low value indicates the opposite.

For any dissimilarity measure (e.g. as the euclidean distance) and any observation i,

- $a(i)$ average dissimilarity with respect to the other points in the same cluster
- $b(i)$ lowest average dissimilarity with respect to all the points in another cluster

We get the silhouette value

$$s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$$

The average of the silhouette values is the silhouette score.

Centroid-based clustering

The most common clustering methods are part of the Centroid-based clustering family. The principle of the methods is to move the center of the clusters and put the point in the cluster has that the closest center. There are several methods such as

- **k-means** which allows the centers to have any coordinates and minimises the within-cluster sum of squares which can be seen as minimising the variance
- **k-median** which is equivalent to k-means but uses the median of the within-cluster squares instead of its mean
- **k-medoids** which is similar to k-means but only allows observations to be used as centers and uses pairwise dissimilarities as the distance

Overall, these methods tend to create clusters of equal size (in term of distance, as it can be seen in figure 4.3) which can be a problem if the variance of the clusters is different (as in figure 4.4.

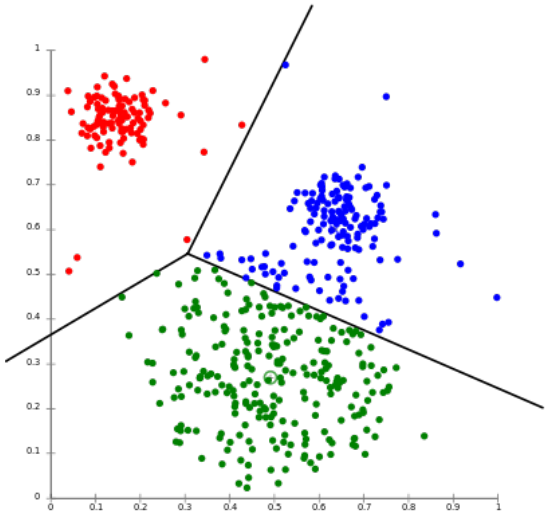

Fig. 4.3 The k-mean clustering splits the area in equal sized partitions

Distribution-based clustering

Those models are based on the idea that points in the same cluster come from the same distribution and the parameters estimated by the model are the parameters of the distributions. For example, in the case of a multinomial distribution, it would estimate the vector of means $\boldsymbol{\mu_i}$ and the covariance matrix $\boldsymbol{\Sigma_i}$ of each distribution i.

Gaussian mixture models clustering (GMM)

The most used application of this family of methods is the **G**aussian **M**ixture **M**odel clustering. It is a method of soft-clustering, in the sense that it attributes to each point a probability r_{nk} of being in cluster k, instead of attributing it a cluster. It is also a latent variable model as only the posterior probabilities of the latent variable r_n, named p_n, are estimated given the parameters, instead of themselves being a parameter of the model. This makes the problem solvable since, otherwise, there would be $N \cdot K$ (N observations and K clusters) parameters for only N points which would make the problem unidentified.

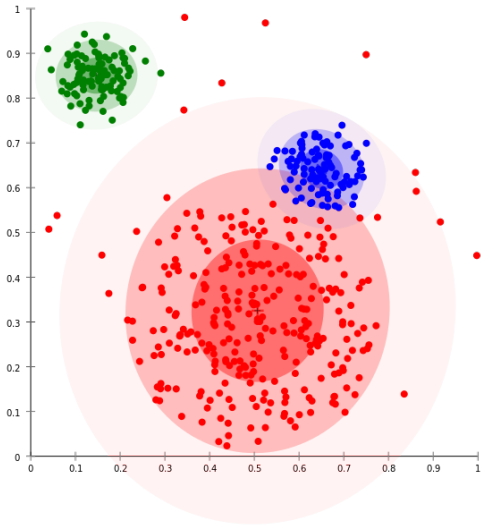

Fig. 4.4 Example of clustering obtained using EM clustering with a gaussian distribution

For each point x_n (vector of coordinates), the probability of being in a cluster r is given by the distribution of the custers which is given by the parameters of the clusters (mean, covariance and prior distribution π_k which is the ratio of observations in the cluster):

$$p_{nk} = \Pr(r_n = k|x_n, \theta) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j^K \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

The details and estimation method for the parameters of the model are explained in appendix A.

4.1.5 Classification

Decision Tree

A decision tree is a directed graph summarising a set of rules used to classify a dependent variable. Figure 4.5 shows a simple example of a decision tree using the X variables to find the class y. In that case, the final split is perfect (there are no misclassified training observations). However, in practice it is perfectly conceivable to obtain misclassified observations as some observations simply cannot be separated (same X but different y). It is also often useful to stop the splitting before reaching perfect purity in order to avoid overfit.

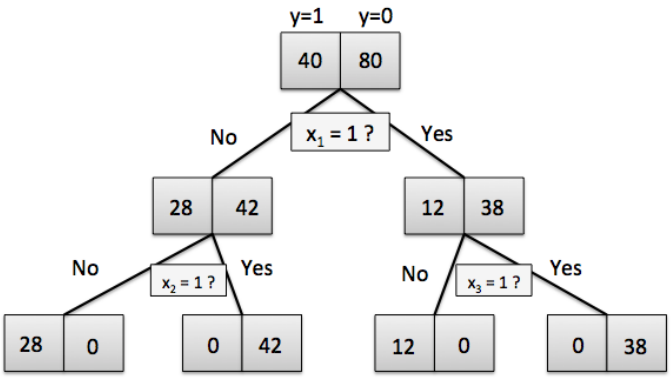

Fig. 4.5 Example of a simple decision tree with the number of training observations in each class shown in the vertices

The construction of a decision tree from training data is performed as such: try splitting (adding one level of depth to the tree) on an attribute and select the split that minimises impurity following a criterion (compare the performance of splitting on every variable and select the best in each leaf). The split can be of several types:

- variable equal to a value. This leads to 2 paths in the case of a binomial variable and to several in the case of a multinomial variable (one path for each possible value)
- variable is greater than or greater and equal to. This is done with continuous variables

The two most common criterion are:

For J classes and f_i the frequency of items having the i label in class i (the fraction of correctly placed items in that branch):

Gini Impurity:

$$I_G = \sum_{i=1}^{J} f_i(1 - f_i)$$

Entropy:

$$S = -\sum_{i=1}^{J} f_i \log_2 f_i$$

Random Forest

Random Forest is a method that limits the overfit of decisions trees by training several of them on a bootstrapped subsample of the observations as well as a subset of the features at

each split (in order to avoid having highly correlated trees due to strong predictors, meaning variables that generate very good splits). The classified class is then chosen by taking the majority vote of the trained trees. The algorithm can also generate probabilities (soft classification) by weighting the votes.

4.2 Models

4.2.1 Baseline

To have a reference when comparing the performance of other models, a baseline is created with the simplest model possible: the simplest prediction is to keep the value at the previous period. Given the stationarity of the processes (after the transformations). If the real process is a random walk, more sophisticated models cannot beat the baseline. Expecting the mean could have also been an alternative given that the dependent variables are stationary.

4.2.2 ARMA

The **A**uto**R**egressive-**M**oving-**A**verage model, ARMA(p, q) is the composition of an autoregressive model of order p, AR(p) and a moving-average model of order q, MA(q).

The model can be written as:

$$x_t = \underbrace{c + \sum_{i=1}^{p} \phi_i x_{t-i}}_{\text{AR}} + \underbrace{\mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-1}}_{\text{MA}} + \varepsilon_t$$

The parameters are evaluated by fitting a Kalman Filter with maximum likelihood estimation.

The coefficients can then be applied to the latest observation to obtain the next period prediction and the forecast can be dynamically propagated to several periods ahead by using the predicted future values of x.

4.2.3 Vector Autoregression (VAR)

A **V**ector **A**uto**R**egression model of order p, VAR(p), can be seen as the composition of several AR(p) models where each also uses the variables of the others. For D variables, it can be written as several equations:

$$\begin{cases} y_{1,t} = c_1 + (a^1_{1,1} y_{1,t-1} + \cdots + a^1_{1,D} y_{D,t-1}) + \cdots + (a^p_{1,1} y_{1,t-p} + \cdots + a^p_{1,D} y_{D,t-p}) + \varepsilon_{1,t} \\ \vdots \\ y_{D,t} = c_1 + (a^1_{D,1} y_{1,t-1} + \cdots + a^1_{D,D} y_{D,t-1}) + \cdots + (a^p_{D,1} y_{1,t-p} + \cdots + a^p_{D,D} y_{D,t-p}) + \varepsilon_{D,t} \end{cases}$$

Which can be rewritten under matrix form:

$$\underbrace{\begin{bmatrix} y_{1,t} \\ \vdots \\ y_{D,t} \end{bmatrix}^T}_{Y} = \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_D \end{bmatrix}^T}_{C} + \underbrace{\begin{bmatrix} y_{1,t-1} \\ \vdots \\ y_{D,t-1} \end{bmatrix}^T}_{X^1} \underbrace{\begin{bmatrix} A^1_{1,1} & \cdots & A^1_{1,D} \\ \vdots & \ddots & \vdots \\ A^1_{D,1} & \cdots & A^1_{D,D} \end{bmatrix}}_{A^1} + \cdots + \underbrace{\begin{bmatrix} y_{1,t-p} \\ \vdots \\ y_{D,t-p} \end{bmatrix}^T}_{X^p} \underbrace{\begin{bmatrix} A^p_{1,1} & \cdots & A^p_{1,D} \\ \vdots & \ddots & \vdots \\ A^p_{D,1} & \cdots & A^p_{D,D} \end{bmatrix}}_{A^P} + \underbrace{\begin{bmatrix} e_{1,t} \\ \vdots \\ e_{D,t} \end{bmatrix}^T}_{E}$$

Which can be rewritten under compact matrix form (by simply concatenating the X^i along the second axis and and A^i along the first axis) as:

$$Y = AX + E$$

This is simply several regression equations in a row vector and can be estimated as a multivariate least squares

$$A = (X^T X)^{-1} X^T Y$$

The coefficients can then be applied to the latest observation to obtain the next period prediction and, since all the variables are being predicted, the forecast can be dynamically propagated to several periods ahead.

4.2.4 Factor-Augmented VAR (FAVAR)

The **F**actor-**A**ugmented **V**ector **A**uto**R**regression model is simply a VAR model which also integrates artificial unobserved variables that summarise a great collection of other variables. Those are called factors. Those factors are added as if they were standard variables. The advantage of including this information as a few factors, instead of the many variables is that it reduces overfitting by having a simpler model while still including most of the information.

The easiest way to obtain those factors is to run PCA on the variables we want to add and select the number of principal components as an hyper-parameter of the model.

The model can be written

$$\begin{bmatrix} X_t \\ f_t \end{bmatrix} = C + A_1 \begin{bmatrix} X_{t-1} \\ f_{t-1} \end{bmatrix} + \cdots + A_p \begin{bmatrix} X_{t-p} \\ f_{t-p} \end{bmatrix} + \eta_t$$

Where X_t is the vector of standard parameters (as in a VAR) and f_t is a vector of added factors. This model can be estimated exactly as the standard VAR(p).

4.2.5 Markov Switching Regression

This model is based on the idea that some processes have different persistent states and they will switch between them which alters their behaviour, in the form of the parameters of the regression.

For N different states, the regressions can be written as

$$\begin{aligned} y_{t+1} &= \mu_1 + \beta^{(1)} X_t + \varepsilon_{1,t} \\ &\vdots \\ y_{t+1} &= \mu_n + \beta^{(N)} X_t + \varepsilon_{N,t} \end{aligned}$$

With $\varepsilon_{i,t}$ following a custom distribution. In the used implementation, $\varepsilon_{i,t} \sim \mathcal{N}(0, \sigma_i^2)$, with only the variance changing, but the model is not limited to normal distributions.

A latent variable, s_t, states the regime of the process at period t. If the variable were known, we could simply train each regression using only the observations in their regime and then forecast using the correct regression. However, as its name indicates, the latent variable

is unknown and needs to be marginalised. The Markov Switching model uses a Markov chain to estimate the prior probability of being in each regime $\xi_{t|t-1}$ (vector of size 1xN). Which allows to sum over the probabilities conditional on the latent variable to marginalise it. Overall, the idea and application is almost the same as with GMM clustering.

The transition matrix of the Markov chain can be written as

$$\Pi = \begin{bmatrix} p_{11} & \cdots & p_{j1} & \cdots & p_{N1} \\ \vdots & \ddots & & & \vdots \\ p_{1i} & \cdots & p_{ji} & \cdots & p_{Ni} \\ \vdots & & & \ddots & \vdots \\ p_{1N} & \cdots & p_{ji} & \cdots & p_{NN} \end{bmatrix}$$

where p_{ab} gives the probability of switching from state a to state b. Which can be written as

$$p_{ij} = \Pr(s_{t+1} = j | s_t = i)$$

We can apply Bayes rule:

$$\Pr(X|B) \cdot \Pr(B) = \Pr(X, B)$$

Which also works with conditional probabilities:

$$\underbrace{\Pr(X|B,C)}_{\text{likelihood}} \cdot \underbrace{\Pr(B|C)}_{\text{prior}} = \Pr(X,B|C) = \underbrace{\Pr(B|X,C)}_{\text{posterior}} \cdot \underbrace{\Pr(X|C)}_{\text{marginal lik}}$$

Which allows to remove B.

We can use the above formula to obtain the value of the posterior, since it is equal to the likelihood times the prior divided by the marginal likelihood. The marginal likelihood can be obtained by marginalising B, which consists in summing over all its occurrences (which gives a sum if it discrete or an integral otherwise).

By replacing terms

- y_t: the observation
- $s_t = j$: the event of being in the j state
- x_t: the exogenous variables of the regression
- $\underline{y}_{t-1}$ all the previous values of y (before t)

- θ the parameters of the model (regressors, Π and σ_i

We get:

$$\underbrace{\Pr(y_t, s_t = j | x_t, \underline{y}_{t-1}; \theta)}_{\text{joint}} = \underbrace{\Pr(y_t | s_t = j, x_t, \underline{y}_{t-1}; \theta)}_{\text{likelihood}} \cdot \underbrace{\Pr(s_t = j | x_t, \underline{y}_{t-1}; \theta)}_{\text{prior}}$$
$$= \underbrace{\Pr(s_t = j | x_t, \overbrace{y_t, \underline{y}_{t-1}}^{\underline{y}_t}; \theta)}_{\text{posterior}} \cdot \underbrace{\Pr(y_t | x_t, \underline{y}_{t-1}; \theta)}_{\text{marginal lik}}$$

The marginal likelihood can be rewritten as the marginalisation of s_t from the joint probability (by summing every regime in the likelihood times the prior):

$$\Pr(y_t | x_t, \underline{y}_{t-1}; \theta) = \sum_i^N \Pr(s_t = i | x_t, \underline{y}_{t-1}; \theta) \cdot \Pr(y_t | s_t = i, x_t, \underline{y}_{t-1}; \theta)$$

Using this, the posterior can be rewritten as:

$$\Pr(s_t = j | x_t, \underline{y}_t; \theta) = \frac{\Pr(y_t | s_t = j, x_t, \underline{y}_{t-1}; \theta) \cdot \Pr(s_t = j | x_t, \underline{y}_{t-1}; \theta)}{\Pr(y_t | x_t, \underline{y}_{t-1}; \theta)}$$
$$= \frac{\Pr(y_t | s_t = j, x_t, \underline{y}_{t-1}; \theta) \cdot \Pr(s_t = j | x_t, \underline{y}_{t-1}; \theta)}{\sum_i^N \Pr(s_t = i | x_t, \underline{y}_{t-1}; \theta) \cdot \Pr(y_t | s_t = i, x_t, \underline{y}_{t-1}; \theta)}$$

If we consider the vector η_t (where f is the pdf of the error of y)

$$\eta_t = \begin{bmatrix} f(y_t | s_t = 1, \underline{y}_{t-1}; \theta) \\ \vdots \\ f(y_t | s_t = N, \underline{y}_{t-1}; \theta) \end{bmatrix}$$

We can rewrite the marginal likelihood as:

$$\Pr(y_t | x_t, \underline{y}_{t-1}; \theta) = \sum_i^N \Pr(s_t = i | x_t, \underline{y}_{t-1}; \theta) \cdot \Pr(y_t | s_t = i, x_t, \underline{y}_{t-1}; \theta) = \mathbf{1}'(\xi_{t|t-1} \odot \eta_t)$$

By putting each posterior in a vector, we get:

$$\xi_{t|t} = \frac{\xi_{t|t-1} \odot \eta_t}{\mathbf{1}'(\xi_{t|t-1} \odot \eta_t)}$$

Where $\odot$ is the Hadamard product (element-wise multiplication)

Given that the process is a Markov chain, the expected value of ξ is:

$$\mathrm{E}(\xi_{t+1}|\underline{y}_t) = \Pi \cdot \mathrm{E}(\xi_t|\underline{y}_t) \iff \xi_{t+1|t} = \Pi \cdot \xi_{t|t}$$

The two above equations allow to find the filtered and expected values of ξ. A recursion over those (using an initial guess for the first value, which can for example be the unconditional probabilities of the Markov chain, which would be the probabilities of staying in each state) allows to find every expected value of ξ.

This finally allows to compute the log-likelihood:

$$\log \mathcal{L}(\theta|\underline{y}_T) = \sum_i^T \log(\Pr(y_t|x_t, \underline{y}_{t-1}; \theta)) = \sum_i^T \log(\mathbf{1}'(\xi_{t|t-1} \odot \eta_t))$$

Which is used to find all the parameters.

Once the model has been estimated, the h period-ahead prediction (out-of-sample) can be estimated using the expected ξ:

$$\xi_{T+h|T} = \Pi^h \xi_{T|T}$$

which allows to get the prediction at $T+h$:

$$\hat{y}_{T+h} = \xi'_{T+h|T} \begin{bmatrix} \mu_1 + \beta^{(1)} X_T \\ \vdots \\ \mu_N + \beta^{(N)} X_T \end{bmatrix}$$

Smoother posterior can be computed by also using the future information and back-propagating the filtering. This is called Kim filter and is used to get better regime attributions (for example for identifying when crises happened in the past)

$$\xi_{t|T} = \xi_{t|t} \odot [\Pi'(\xi_{t+1|T} \oslash \xi_{t+1|t})]$$

Where $\oslash$ is the element-wise division.

The model can be either estimated by straight likelihood, using expectation-maximisation (as shown for the similar problem of GMM clustering in appendix A) or by a combination of the two methods.

4.2.6 Split Regression

The Split regression is a custom model created in the scope of this thesis. It is similar to the above Markov Switching Regression but uses clustering and classification to find the latent regime variable instead of a Markov Chain and maximum likelihood. The main difference in hypothesis is that the future regime also depends on the other variables in the regression instead of only depending on the previous states of the dependent variable.

Training

The training of the method is performed in 2 steps:

For a given number of states (hyper-parameter of the model), a clustering method is applied on the matrix of train observations which contains the dependent variable (for L lags).

$$X = \begin{bmatrix} y_T \\ \vdots \\ y_t \\ \vdots \\ y_L \end{bmatrix}$$

Any clustering method can be used. In this thesis, both k-means and Gaussian Mixture Model clustering are tried.

After attributing a class to each observation (c_t), a classification model can be trained on the following data.

$$y = \begin{bmatrix} c_T \\ \vdots \\ c_L \end{bmatrix} \quad X = \begin{bmatrix} y_{T-1} & \cdots & y_{T-L} & X_{T-1} & \cdots & X_{T-L} \\ \vdots & & \vdots & & \vdots & \\ y_{L-1} & \cdots & y_0 & X_{L-1} & \cdots & X_0 \end{bmatrix}$$

In this thesis, the used classification method is random forests.

The classification model is then reused on the training Xs to compute the probabilities of being in each class and those probabilities are used to give weights to observations when training N (for N states, which is an hyper-parameter of the model) VAR models using Weighted Least Squares.

The classification model is then used on the test Xs to compute the probability of being in each state and that probability is used to weight the contribution of each VAR result when computing the final prediction, which is equivalent to what is done in the Markov switching model with the prior probabilities.

4.2.7 Neural Networks

Neural Networks are a family of methods inspired from the composition of animal brains. They are composed of computation units called artificial neurons that have connections between them (the equivalent of synapses). They can be seen as a weighted graph where the neurons are vertices and the connections are edges. Each connection contains a weight that is used to compute the influence of the input neuron of the edge on the output one.

The graph is composed of three families of neurons:

- the input neurons, which contain the input values of the model (the Xs)
- the hidden neurons, which are used for the computations and are not seen from outside the model
- the output neurons, which contain the result of the model (the ys)

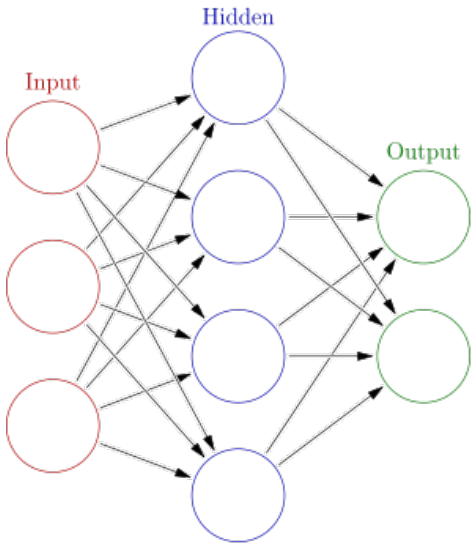

Fig. 4.6 Graph structure of a neural network with one hidden layer

The value of each vertex v_i is computed by applying the following, with I being the set of input vertices of that vertex (first vertex in the edges where the second vertex is the one for

which the value is computed):

$$v_i = \sum_{v_j \in I} f(v_j \cdot w_{j,i})$$

Where $w_{i,j}$ is the weight of the edge connecting the vertices j and i and f is the activation function.

The two most common activation functions are:

Sigmoid: the Sigmoid activation function is the original activation function used in the first neural network concept. It mimics the behaviour of biological neurons.

$$f(x) = \frac{1}{1+e^{-x}}$$

Rectifier Linear Unit: The ReLU activation function is computationally lighter and thus preferred to the sigmoid for large datasets.

$$f(x) = \max(x, 0)$$

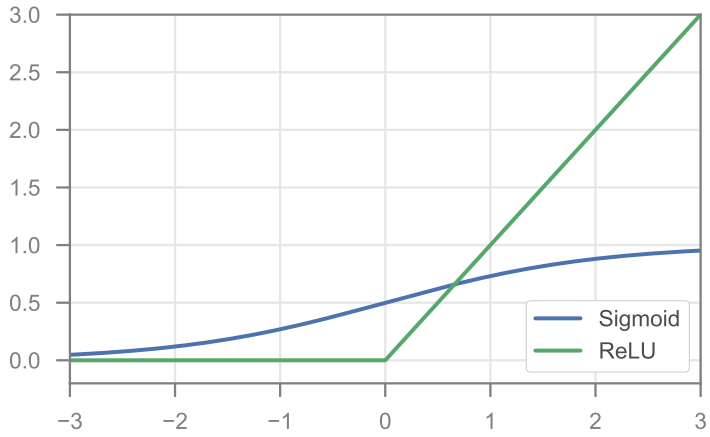

Fig. 4.7 Plot of the Sigmoid and Rectifier Linear Unit activation functions

Multilayer Perceptron

The **M**ulti**L**ayer **P**reception is a feedforward neural network model that connects input neurons to output neurons through one or several layers of hidden neurons. The feedfor-

ward denomination comes from the fact that there are no cycles in the graph which makes computation and training easier.

Biases are added to each layer in order to have the equivalent of the constant term in linear regressions. It also allows the neural network to have an output different from 0 in case all the neurons are turned-off by the ReLU activation function.

The optimal weights and biases of the model are found using a modified version of gradient descent called Adam (Kingma and Ba, 2014). The gradient itself is computed through a method called backpropagation which starts computing the derivatives of the cost at the output and propagates them backward to all neurons. This is vastly more efficient than computing it starting from each neuron (forward propagation) as it avoids recomputing several times the same derivatives for the layers in the middle.

L2 regularisation

As the method is prone to overfitting, given its incredibly high number of parameters, a regularisation term is added to the loss in order to reduce overfit by shutting down connections.

The resulting optimised loss function becomes:

$$\mathcal{L} = \underbrace{\sqrt{\frac{1}{N}\sum_{n}^{N}(\hat{y}_n - y_n)^2}}_{\text{RMSE}} + \underbrace{\beta \sum_{i.j} w_{i,j}^2}_{\text{L2}}$$

Where β is an hyper-parameter of the model.

Chapter 5

Results

The evaluated out-of-sample forecast performance of the models on the 3 time series is presented in this chapter. The forecasts were done for 1, 4 and 12 periods ahead. The tables all contain the results scaled to have the baseline model (random walk) as 1. The best performance for each column (h-step-ahead and loss function) is marked in red. A plot of the performance of each model is also shown for each time series to have a more intuitive view.

The name code of the models is composed of, first, the specific model used and, second, the type of data used as shown in table 5.1.

Data	Code	Model	line
Univariate	BASE	Random Walk	:
	AR	AutoRegressive	-
	MA	Moving Average	-
	ARMA	AutoRegressive Moving Average	-
	MSAR	Markov Switching AutoRegressive	–
	SPLITAR	Split AutoRegressive	-.
	MLPAR	MultiLayer Perceptron AutoRegressive	:
Multivariate	VAR	Vector AutoRegressive	-
	MSVAR	Markov Switching Vector AutoRegressive	–
	SPLITVAR	Split Vector AutoRegressive	-.
	MLPVAR	MultiLayer Perceptron Vector AutoRegressive	:
Factors	FAVAR	Factor-Augmented Vector AutoRegressive	-
	MSFAVAR	Markov Switching Factor-Augmented Vector AutoRegressive	–
	SPLITFAVAR	Split Factor-Augmented Vector AutoRegressive	-.
	MLPFAVAR	MultiLayer Perceptron Factor-Augmented Vector AutoRegressive	:

Table 5.1 List of model codes used and their corresponding model, data and line type

5.1 GDP

Loss	RMSE			MAE		
Periods ahead	1	4	12	1	4	12
BASE	1.0	1.29	1.45	1.0	1.15	1.17
AR	0.92	1.09	1.1	0.82	0.9	**0.88**
MA	0.92	1.1	1.12	0.8	0.87	0.88
ARMA	0.92	1.08	1.12	**0.77**	0.85	0.88
MSAR	0.94	1.08	1.12	0.8	0.94	0.92
SPLITAR	0.91	1.11	1.17	0.79	0.92	0.94
MLPAR	1.12	1.11	1.16	1.02	0.98	0.91
VAR	0.92	1.07	1.12	0.79	0.84	0.89
MSVAR	0.95	1.06	1.1	0.82	0.92	0.88
SPLITVAR	0.9	1.1	1.18	0.83	0.98	0.94
MLPVAR	0.94	1.13	1.2	0.8	0.93	0.98
FAVAR	0.91	0.99	1.06	0.86	0.86	0.88
MSFAVAR	**0.86**	0.96	1.05	0.84	0.84	0.89
SPLITFAVAR	0.89	**0.93**	**1.05**	0.84	**0.83**	0.9
MLPFAVAR	0.98	1.17	1.11	0.91	1.09	0.95

Table 5.2 out-of-sample performance of the models when forecasting the GDP growth. Tested on the period 2004-04-01 to 2014-07-01

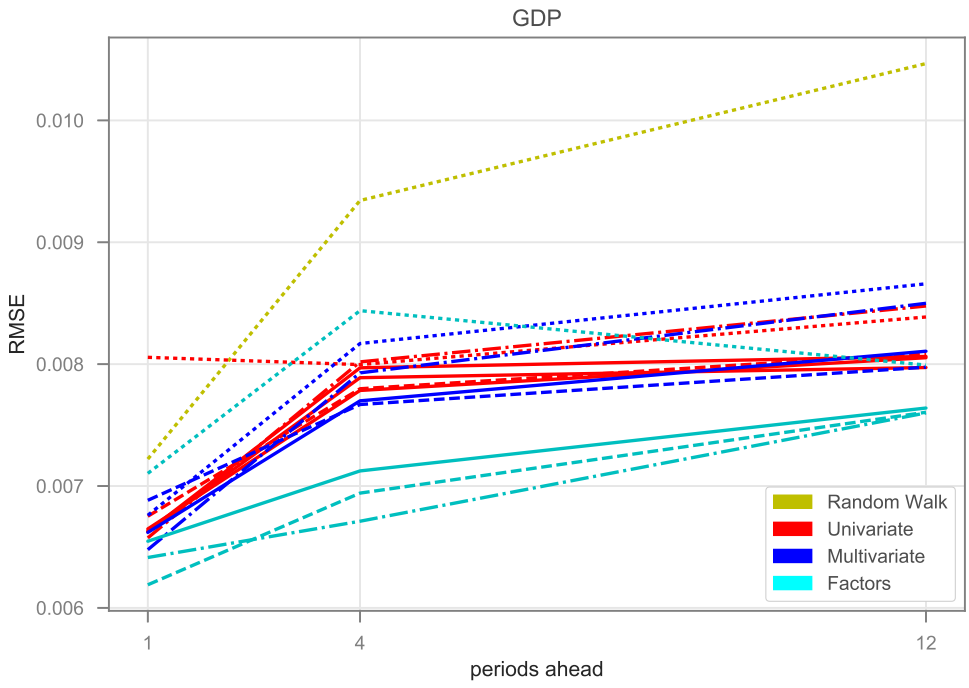

Fig. 5.1 Plot of the above RMSE

When predicting the GDP, the use of one factor greatly improves the performance. However, using more than one results in a decreased performance.The Markov Switching regression only starts improving over the simpler models (such as AR or VAR) when using factors. The Split FAVAR model performs as well as the Markov Switching model.

The neural network is always performing poorly. This suggests that quarterly data is just not enough to train the model correctly.

5.2 CPI

Loss	RMSE			MAE		
Periods ahead	1	4	12	1	4	12
BASE	1.0	1.38	1.48	1.0	1.3	1.49
AR	0.92	1.09	1.02	0.86	1.02	0.95
MA	0.93	**1.04**	1.04	0.87	1.01	1.01
ARMA	0.91	1.04	1.02	0.84	0.94	0.95
MSAR	0.93	1.12	1.1	0.84	1.11	1.1
SPLITAR	0.93	1.06	1.03	0.85	**0.92**	0.97
MLPAR	**0.89**	1.07	**1.01**	**0.84**	0.97	0.96
VAR	0.93	1.09	1.06	0.86	1.03	1.04
MSVAR	0.95	1.07	1.06	0.86	1.0	1.02
SPLITVAR	0.92	1.07	1.05	0.86	0.97	1.01
MLPVAR	0.93	1.07	1.05	0.85	0.98	1.0
FAVAR	0.93	1.11	1.06	0.86	1.01	1.02
MSFAVAR	0.96	1.08	1.05	0.86	0.98	1.0
SPLITFAVAR	0.92	1.08	1.03	0.85	0.93	**0.95**
MLPFAVAR	0.91	1.06	1.03	0.85	0.92	0.98

Table 5.3 out-of-sample performance of the models when forecasting the CPI growth. Tested on the period 2004-09-01 to 2014-11-01

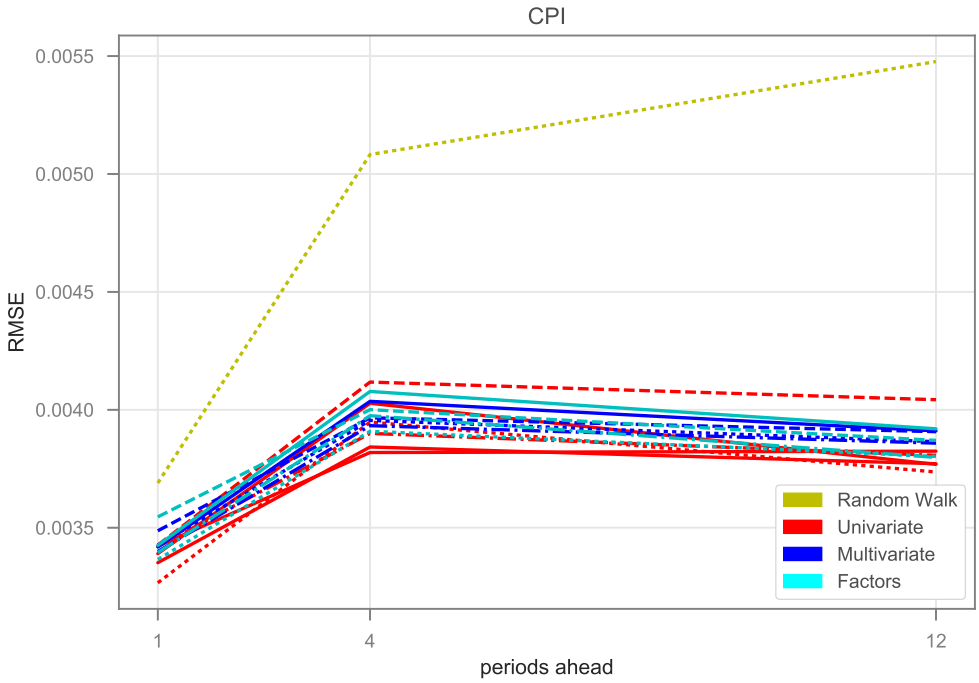

Fig. 5.2 Plot of the above RMSE

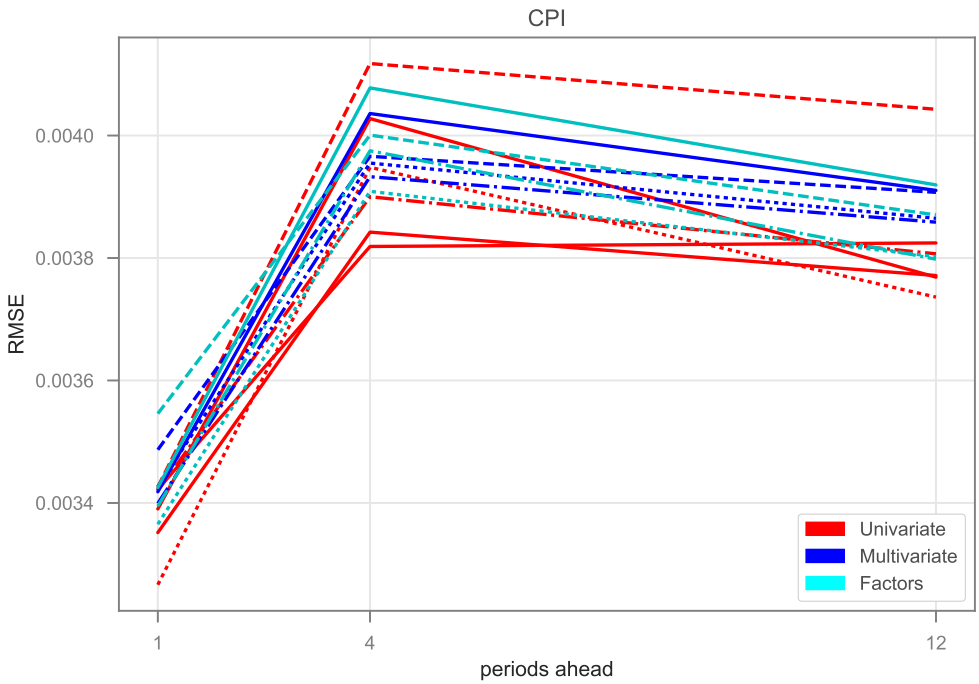

Fig. 5.3 Zoomed version of the above plot (without the random walk)

In the case of CPI, the only clear result is the strong autoregressive part which makes the random walk perform terribly. Otherwise, there does not seem to be any clear difference. However, the neural networks start performing as well as other methods and even manage to get most of the best scores, even if by only a small margin. The factors, or even the other variables, do not seem to play any role, unlike with GDP where they brought a strong performance improvement.

5.3 Employment

Loss	RMSE			MAE		
Periods ahead	1	4	12	1	4	12
BASE	1.0	1.4	2.21	1.0	1.37	2.02
AR	1.0	1.36	1.94	1.0	1.23	1.62
MA	1.12	2.11	2.13	1.09	1.77	1.79
ARMA	1.0	1.39	1.97	0.99	1.23	1.63
MSAR	0.98	1.63	2.45	0.97	1.46	2.34
SPLITAR	1.06	1.47	**1.79**	1.04	1.36	1.84
MLPAR	1.09	1.45	1.85	1.04	1.31	1.64
VAR	1.01	1.42	2.0	1.0	1.25	1.66
MSVAR	0.99	1.38	1.99	0.99	1.26	1.65
SPLITVAR	1.06	1.46	1.8	1.06	1.3	1.8
MLPVAR	1.04	1.45	2.1	1.04	1.31	1.79
FAVAR	1.02	1.41	2.0	1.08	1.32	1.68
MSFAVAR	**0.93**	1.27	1.85	**0.97**	1.24	**1.54**
SPLITFAVAR	1.0	**1.25**	1.93	1.02	**1.21**	1.64
MLPFAVAR	1.05	1.49	2.02	1.1	1.43	1.78

Table 5.4 out-of-sample performance of the models when forecasting the employment growth. Tested on the period 2004-09-01 to 2014-11-01

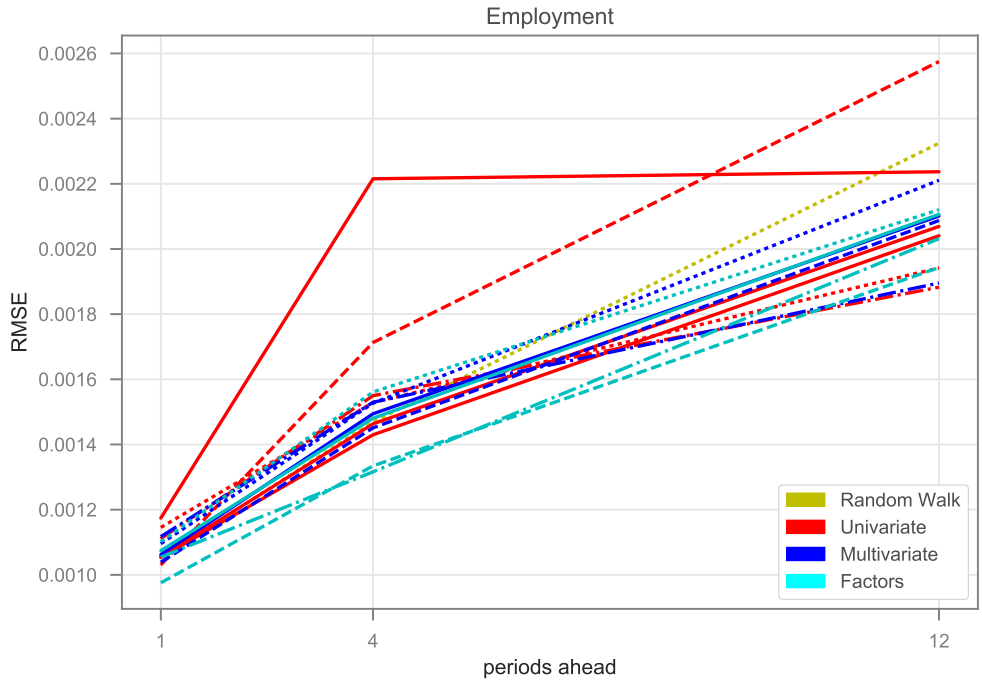

Fig. 5.4 Plot of the above RMSE

The most flagrant result is the great performance of the Markov Switching method with factors while it performs poorly in the univariate case for the 12 periods ahead forecast. Conversely, The Split method manages to get the best performance for the 12 periods ahead window using only the univariate data and seems to only get any improvement from factors in the near predictions.

It is interesting to note that this is the only time series where the Split regression performed consistently better when using the GMM clustering than with k-means, suggesting a gaussian distribution of the dependent variable based on states (which also explains the very good performance of the Markov Switching model).

5.4 Aggregated

The scaled scores of the 3 time series can be summed to get an overview of the average performance of the methods.

Loss	RMSE			MAE		
Periods ahead	1	4	12	1	4	12
BASE	3.0	4.07	5.15	3.0	3.83	4.68
AR	2.84	3.54	4.07	2.67	3.15	3.44
MA	2.96	4.25	4.28	2.76	3.65	3.68
ARMA	2.83	3.51	4.11	**2.6**	3.02	3.46
MSAR	2.84	3.83	4.67	2.61	3.5	4.36
SPLITAR	2.89	3.64	4.0	2.68	3.2	3.74
MLPAR	3.09	3.63	4.02	2.9	3.25	3.51
VAR	2.85	3.58	4.18	2.65	3.12	3.58
MSVAR	2.89	3.52	4.15	2.67	3.19	3.55
SPLITVAR	2.88	3.62	4.03	2.75	3.24	3.76
MLPVAR	2.91	3.66	4.35	2.69	3.22	3.77
FAVAR	2.86	3.5	4.12	2.81	3.19	3.58
MSFAVAR	**2.75**	3.31	**3.95**	2.68	3.06	**3.43**
SPLITFAVAR	2.81	**3.26**	4.01	2.71	**2.97**	3.49
MLPFAVAR	2.94	3.71	4.15	2.86	3.44	3.71

Table 5.5 Sum of the 3 out-of-sample scores

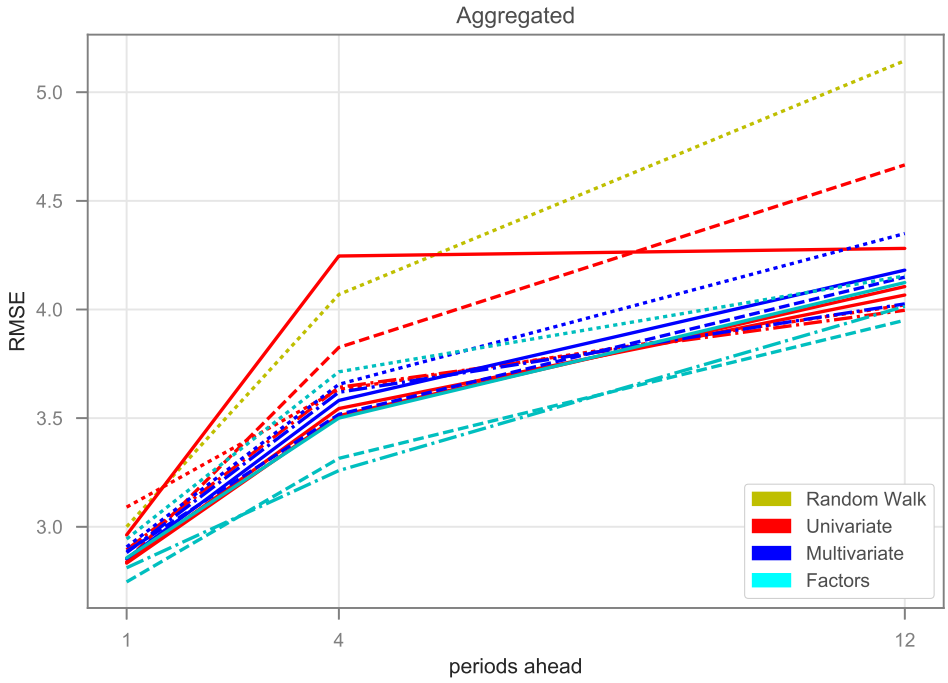

Fig. 5.5 Plot of the above sum of scores

The aggregate score shows several interesting results: the most reliable method for next period predictions seems to be the Markov Switching regression with factors, which is

outside of the group. The Split regression performs slightly better than the Markov Switching regression for 4 periods ahead, but both are outside the group. For 12 periods ahead those methods go back to the group but are still the top performers.

5.5 Summary

The use of factors often improves the forecast over univariate methods. Besides, the Markov Switching regression improves over single state models in the case of regressions with factors but not with others. The Split regression performs as well as the Markov Switching regression, each method beating the other several times. Finally, the neural networks model do not perform well as they have horrible performance with the quarterly and average performance with monthly data. Their use does not seem justified in this context.

5.6 Robustness

Appendix B shows the result of the Diebold-Mariano test. Given the small testing window, the lack of significance of the comparisons was expected. Nevertheless, the use of 3 different time series makes the results more trustworthy as there is no big discrepancy between them. However, given the lack of significant differences between the forecasts, one must be careful when trying to generalise the findings of this thesis.

Chapter 6

Conclusion

The 3 tested time series, the percentage change from last period of quarterly GDP, monthly CPI and monthly non-farm employment, did not show many significant forecast differences according to the Diebold-Mariano test statistic for 1, 4 and 12 periods ahead forecasts. However, it is still possible to make suppositions based on the RMSE performance.

The use of multi-state models, such as the Markov Switching regression as well as the Split regression, combined with factor-augmented data, provide an improvement in the forecasting performance over standard univariate models such as ARMA. Especially for more than a period ahead.

The utility of factors is clear in the case of GDP, where even the FAVAR beats every non-factor-augmented method. They also bring an improvement in the case of the employment, however, only for state based models. Finally, factors do not seem to bring any improvement in the case of the CPI.

The fact that multi-state models did not bring any improvement in the case of the CPI seems to indicate an absence of defined and persistent state, at least for monthly periods. Conversely, it seems to confirm the existence of states in the case of both GDP and employment. Even more, the great performance of gaussian distributed clustering for the Split FAVAR method in the case of employment makes a strong case for the assumption of gaussian errors.

Even though the Split regression performed very well, the other machine learning model tested in this thesis, the neural network multilayer perceptron, did not perform as well and was always one of the worst overall performers. This seems to confirm that the method is simply not made to work with such a small sample as its flexibility makes it more prone to overfitting and, in the end, ends up being severely restrained by a high L2 regularisation factor.

Given the very good performance of both Markov Switching and Split regressions it may be useful to combine the two, as their states as well as predictions are not perfectly correlated, using an ensemble learning method.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning.

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*, 120(1):387–422.

Boot, T. and Pick, A. (2016). Optimal forecasts from markov switching models. *Journal of Business & Economic Statistics*, (just-accepted).

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chen, H., Wan, Q., and Wang, Y. (2014). Refined diebold-mariano test methods for the evaluation of wind power forecasting models. *Energies*, 7(7):4185–4198.

Diebold, F. X. (2015). Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of diebold–mariano tests. *Journal of Business & Economic Statistics*, 33(1):1–1.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.

Gonzalez, S., Economic, C., and Branch, F. P. (2000). *Neural networks for macroeconomic forecasting: a complementary approach to linear regression models*. Department of Finance Canada.

Hamilton, J. D. (1990). Analysis of time series subject to changes in regime. *Journal of econometrics*, 45(1-2):39–70.

Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton university press Princeton.

Ho, T. K. (1995). Random decision forests. 1:278–282.

Kim, T. Y., Oh, K. J., Sohn, I., and Hwang, C. (2004). Usefulness of artificial neural networks for early warning system of economic crisis. *Expert Systems with Applications*, 26(4):583 – 590.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

Moshiri, S. and Cameron, N. (2000). Neural network versus econometric models in forecasting ination. *J. Forecast*, 19:201–217.

Mullainathan, S. and Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106.

Nakamura, E. (2005). Inflation forecasting using a neural network. *Economics Letters*, 86(3):373–378.

Ng, A. (2016). Cs229 machine learning. http://cs229.stanford.edu/.

Ouyang, M., Welsh, W. J., and Georgopoulos, P. (2004). Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, 20(6):917–923.

Stock, J. H. and Watson, M. W. (1998). A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. Technical report, National Bureau of Economic Research.

Stock, J. H. and Watson, M. W. (2016). Factor models and structural vector autoregressions in macroeconomics.

Swanson, N. R. and White, H. (1997). A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. *The Review of Economics and Statistics*, 79(4):540–550.

Tkacz, G. and Hu, S. (1999). Forecasting gdp growth using artificial neural networks. Bank of Canada Working Paper 99-3.

Van Norden, S. and Vigfusson, R. (1996). Regime-switching models: A guide to the bank of canada gauss procedures.

Varian, H. R. (2013). Big data: New tricks for econometrics.

Vila, J. P. and Schniter, P. (2013). Expectation-maximization gaussian-mixture approximate message passing. *IEEE Transactions on Signal Processing*, 61(19):4658–4672.

Wilson, D. R. and Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451.

Appendix A

GMM and EM

A.1 Estimation

For N observations and K clusters, the likelihood of the model can be written as

$$\mathcal{L}(\mu,\Sigma,r|X) = p(X|\mu,\Sigma,r) = \prod_{n}^{N}\prod_{k}^{K}[\mathcal{N}(x_n|\mu_k,\Sigma_k)]^{r_{nk}}$$

where r_{nk} represents the real probabilities of the realisations of the latent variable r_n:

$$r_{nk} = \Pr(r_n = k)$$

Bayesian models require a specified prior distribution which we define as

$$\Pr(r_n = k) = \pi_k$$

This represents the frequency of each cluster. ($\sum_k \pi_k = 1$).

We define the set of all parameters $\theta = \{\mu,\Sigma,\pi\}$.

This set of parameters can be optimised by maximum likelihood:

$$\max_{\theta}\mathcal{L}(\theta|X) = \max_{\theta}\log\mathcal{L}(\theta|X) = \max_{\theta}\log\prod_{n}^{N}\Pr(x_n|\theta) = \max_{\theta}\sum_{n}^{N}\log\Pr(x_n|\theta)$$

Bayes rule gives

$$\underbrace{\Pr(x_n, r_n = k|\theta)}_{\text{joint}} = \underbrace{\Pr(x_n|r_n = k,\theta)}_{\text{likelihood}}\underbrace{\Pr(r_n = k|\theta)}_{\text{prior}=\pi_k} = \underbrace{\Pr(r_n = k|x_n,\theta)}_{\text{posterior}=p_{nk}}\underbrace{\Pr(x_n|\theta)}_{\text{marginal lik}}$$

The posterior can be written as the likelihood times the prior divided by the marginal likelihood. The marginal likelihood can be obtained by marginalising r_n out of the joint distribution by summing over all its possible occurrences:

$$\begin{aligned}\Pr(x_n|\theta) &= \sum_k^K \Pr(x_n, r_n = k|\theta) = \sum_k^K \underbrace{\Pr(x_n|r_n = k, \theta)}_{\text{likelihood}} \underbrace{\Pr(r_n = k|\theta)}_{\text{prior}} \\ &= \sum_k^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\end{aligned}$$

This gives the posterior

$$p_{nk} = \Pr(r_n = k|x_n, \theta) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j^K \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

The above marginal likelihood can be plugged in the likelihood maximisation problem:

$$\max_\theta \mathcal{L}(\theta|X) = \max_\theta \sum_n^N \log \Pr(x_n|\theta) = \max_\theta \sum_n^N \log \sum_k^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

The above optimisation can be performed using the **E**xpectation-**M**aximisation (EM) algorithm. However, given that the negative log-likelihood function is not convex, the algorithm is only guaranteed to finish in a local optimum, not in the global one.

A.2 EM

This methods alternatively optimises a part of the parameters while keeping the other part constant and iterates the two processes until the optimum is reached.

A.2.1 E-step

Keeping the θ constant, compute the posterior probabilities

$$p_{nk} = \Pr(r_n = k|x_n, \theta) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j^K \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

A.2.2 M-step

This time, keeping the posterior probabilities constant, optimise the parameters θ by maximum likelihood.

Jensen inequality applied to the log gives

$$\log\left(\sum_i a_i b_i\right) \geq \sum_i a_i \log b_i$$

This can be used on a modified form of the log-likelihood (by optimising a lower bound, which ensures that the likelihood cannot decrease during this step)

$$\begin{aligned}\mathcal{L}(\theta|X) &= \sum_n^N \log \sum_k^K \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k) = \sum_n^N \log \sum_k^K \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{p_{kn}} p_{kn} \\ &\geq \sum_n^N \sum_k^K p_{kn} \log \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{p_{kn}} = \sum_n^N \sum_k^K p_{kn}[\log \pi_k + \log \mathcal{N}(x_n|\mu_k,\Sigma_k) - \log p_{nk}]\end{aligned}$$

The FOCs of this form (under the constraints that the $\sum_k \pi_k = 1$ and $\pi_k > 0$) can be taken to find the optimal parameters given the posterior probabilities

$$\begin{aligned}\mu_k &= \frac{\sum_n^N p_{kn} x_n}{p_{kn}} \\ \Sigma_k &= \frac{\sum_n^N p_{kn}(x_n - \mu_k)(x_n - \mu_k)^T}{p_{kn}} \\ \pi_k &= \frac{1}{N}\sum_n^N p_{kn}\end{aligned}$$

The E and M steps are iterated until the local optimum is reached with the required precision. The method can be run several times with different initial seeds to avoid being stuck in an inappropriate local optimum.

Appendix B

Diebold-Mariano Results

GDP: 1 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	-0.70	-0.65	-0.61	-0.52	-0.77	0.77	-0.64	-0.41	-1.00	-0.53	-1.20	-1.94	-1.39	-0.14
AR	-	0.01	0.13	0.65	-0.26	**2.81**	-0.04	0.66	-0.50	0.56	-0.10	-0.54	-0.30	0.36
MA	-	-	0.18	0.73	-0.22	**3.16**	-0.06	0.62	-0.43	0.48	-0.10	-0.51	-0.29	0.35
ARMA	-	-	-	0.77	-0.31	**3.14**	-0.20	0.56	-0.45	0.46	-0.12	-0.50	-0.29	0.32
MSAR	-	-	-	-	-0.76	**2.68**	-0.56	0.34	-0.79	0.03	-0.25	-0.67	-0.46	0.26
SPLITAR	-	-	-	-	-	**2.80**	0.19	1.04	-0.43	0.76	-0.04	-0.48	-0.23	0.40
MLPAR	-	-	-	-	-	-	**-2.96**	-1.96	**-2.57**	**-2.41**	-1.52	-1.78	-1.76	-0.67
VAR	-	-	-	-	-	-	-	0.63	-0.38	0.53	-0.08	-0.46	-0.25	0.34
MSVAR	-	-	-	-	-	-	-	-	-1.43	-0.37	-0.47	-0.98	-0.78	0.18
SPLITVAR	-	-	-	-	-	-	-	-	-	0.96	0.10	-0.44	-0.11	0.52
MLPVAR	-	-	-	-	-	-	-	-	-	-	-0.28	-0.73	-0.50	0.27
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-1.62	-0.71	0.72
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	0.88	1.18
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	0.80

Table B.1 DM test statistics for the GDP out-of-sample forecast performed with an expanding window ranging from 2004-04-01 to 2014-07-01. A positive number indicates that the model on the row outperforms the model on the column. Bolded statistics are significant at 95%

GDP: 4 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	-0.67	-0.61	-0.71	-0.74	-0.61	-0.65	-0.75	-0.81	-0.70	-0.55	-1.14	-1.25	-1.27	-0.59
AR	-	0.62	-1.76	-0.29	0.84	0.21	**-2.03**	-0.48	0.06	0.61	-0.66	-0.80	-0.92	0.45
MA	-	-	-1.33	-0.44	0.24	0.04	-1.95	-0.55	-0.05	0.37	-0.68	-0.80	-0.91	0.36
ARMA	-	-	-	0.05	**2.05**	0.42	-1.28	-0.26	0.21	0.81	-0.58	-0.72	-0.85	0.55
MSAR	-	-	-	-	0.76	0.44	-0.31	-0.38	0.22	0.80	-0.70	-0.86	-1.00	0.60
SPLITAR	-	-	-	-	-	-0.05	**-1.97**	-0.87	-0.14	0.35	-0.84	-0.97	-1.09	0.36
MLPAR	-	-	-	-	-	-	-0.54	-1.14	-0.19	0.39	-1.29	-1.44	-1.64	0.46
VAR	-	-	-	-	-	-	-	-0.06	0.30	0.87	-0.49	-0.62	-0.75	0.61
MSVAR	-	-	-	-	-	-	-	-	0.79	1.29	-0.76	-0.98	-1.15	0.80
SPLITVAR	-	-	-	-	-	-	-	-	-	0.59	-1.42	-1.73	-1.79	0.53
MLPVAR	-	-	-	-	-	-	-	-	-	-	-1.13	-1.30	-1.38	0.23
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-1.44	**-2.07**	1.64
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-0.93	1.91
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	**2.08**

GDP: 12 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	-1.77	-1.70	-1.71	-1.67	-1.42	-1.45	-1.66	-1.91	-1.37	-1.52	**-3.33**	**-3.82**	**-3.45**	**-2.53**
AR	-	**3.28**	**3.04**	0.38	**3.27**	**2.93**	**3.18**	0.02	**3.73**	**2.39**	-0.41	-0.38	-0.42	0.03
MA	-	-	**-2.84**	0.08	**3.11**	**2.78**	**2.31**	-0.52	**3.62**	**2.14**	-0.52	-0.47	-0.52	-0.11
ARMA	-	-	-	0.12	**3.21**	**2.84**	**2.75**	-0.46	**3.64**	**2.16**	-0.51	-0.46	-0.50	-0.09
MSAR	-	-	-	-	1.00	0.88	0.05	-0.41	1.18	1.75	-0.59	-0.53	-0.60	-0.17
SPLITAR	-	-	-	-	-	-1.30	**-2.93**	**-2.20**	0.14	0.59	-0.99	-0.87	-0.95	-0.72
MLPAR	-	-	-	-	-	-	**-2.57**	-1.83	0.94	0.95	-0.87	-0.77	-0.84	-0.58
VAR	-	-	-	-	-	-	-	-0.70	**3.73**	1.94	-0.56	-0.50	-0.55	-0.17
MSVAR	-	-	-	-	-	-	-	-	**1.98**	**3.27**	-0.51	-0.45	-0.51	0.04
SPLITVAR	-	-	-	-	-	-	-	-	-	0.54	-0.97	-0.86	-0.93	-0.71
MLPVAR	-	-	-	-	-	-	-	-	-	-	-1.59	-1.33	-1.46	-1.33
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-0.21	-0.39	1.50
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-0.05	1.03
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	1.35

Table B.2 DM test statistics for the GDP out-of-sample forecast performed with an expanding window ranging from 2004-04-01 to 2014-07-01. A positive number indicates that the model on the row outperforms the model on the column. Bolded statistics are significant at 95%

CPI: 1 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	-0.80	-0.85	-0.89	-0.79	-0.75	-1.29	-0.89	-0.67	-0.82	-0.75	-0.86	-0.37	-0.83	-1.02
AR	-	0.21	-0.33	0.24	0.20	-0.97	0.17	0.55	0.06	0.25	0.21	0.93	0.03	-0.15
MA	-	-	-0.66	0.04	-0.02	-0.93	-0.03	0.53	-0.22	0.04	0.02	0.90	-0.25	-0.42
ARMA	-	-	-	0.85	0.90	-0.53	0.52	0.98	0.61	1.04	0.62	**2.29**	0.58	0.11
MSAR	-	-	-	-	-0.12	-1.04	-0.10	0.86	-0.52	0.00	-0.02	1.24	-0.59	-0.69
SPLITAR	-	-	-	-	-	-0.97	-0.02	0.61	-0.59	0.10	0.06	1.39	-0.91	-0.61
MLPAR	-	-	-	-	-	-	0.95	1.31	0.81	0.94	1.03	1.28	0.80	0.81
VAR	-	-	-	-	-	-	-	1.21	-0.20	0.10	0.21	0.87	-0.23	-0.51
MSVAR	-	-	-	-	-	-	-	-	-0.91	-0.66	-1.12	0.40	-0.86	-1.15
SPLITVAR	-	-	-	-	-	-	-	-	-	0.46	0.31	1.51	-0.11	-0.36
MLPVAR	-	-	-	-	-	-	-	-	-	-	-0.02	1.50	-0.51	-0.56
FAVAR	-	-	-	-	-	-	-	-	-	-	-	0.89	-0.36	-0.70
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-1.67	-1.13
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.35

CPI: 4 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	-1.94	**-2.31**	**-2.45**	-1.67	**-2.45**	**-2.14**	**-2.47**	**-2.47**	**-2.36**	**-2.40**	**-2.38**	**-2.35**	**-2.33**	**-2.44**
AR	-	-0.79	-0.96	0.45	-0.50	-0.51	0.04	-0.26	-0.39	-0.32	0.20	-0.11	-0.18	-0.45
MA	-	-	0.15	1.04	0.35	0.60	1.33	1.02	0.58	0.79	1.24	0.87	0.55	0.43
ARMA	-	-	-	1.09	0.44	0.66	1.93	1.50	0.84	1.48	**2.08**	1.39	0.74	0.54
MSAR	-	-	-	-	-0.61	-1.18	-0.29	-0.52	-0.55	-0.53	-0.12	-0.34	-0.36	-0.59
SPLITAR	-	-	-	-	-	0.19	0.95	0.52	0.69	0.65	**1.97**	1.81	1.24	0.17
MLPAR	-	-	-	-	-	-	0.47	0.09	-0.06	0.03	0.59	0.22	0.09	-0.15
VAR	-	-	-	-	-	-	-	-1.62	-0.85	-1.06	0.57	-0.33	-0.33	-0.99
MSVAR	-	-	-	-	-	-	-	-	-0.34	-0.21	1.41	0.37	0.05	-0.54
SPLITVAR	-	-	-	-	-	-	-	-	-	0.38	1.63	1.19	0.43	-0.51
MLPVAR	-	-	-	-	-	-	-	-	-	-	1.84	0.78	0.15	-0.66
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-1.47	-0.87	**-2.04**
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-0.29	**-2.18**
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	-0.75

Table B.3 DM test statistics for the CPI out-of-sample forecast performed with an expanding window ranging from 2004-09-01 to 2014-11-01. A positive number indicates that the model on the row outperforms the model on the column. Bolded statistics are significant at 95%

CPI: 12 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	**-2.78**	**-2.71**	**-2.82**	**-2.11**	**-2.79**	**-2.86**	**-2.58**	**-2.60**	**-2.74**	**-2.67**	**-2.61**	**-2.66**	**-2.82**	**-2.78**
AR	-	0.59	0.04	0.63	0.29	-0.53	1.73	1.52	0.65	1.09	**2.18**	1.53	0.16	0.26
MA	-	-	-0.50	0.56	-0.14	-0.94	**2.11**	1.13	0.28	0.49	1.14	0.60	-0.13	-0.17
ARMA	-	-	-	0.66	0.42	-0.46	1.68	1.71	0.85	1.47	**2.39**	1.67	0.18	0.39
MSAR	-	-	-	-	-0.60	-0.76	-0.35	-0.35	-0.51	-0.46	-0.30	-0.42	-0.59	-0.62
SPLITAR	-	-	-	-	-	-0.51	1.08	1.48	0.78	0.98	1.31	0.74	-0.10	-0.09
MLPAR	-	-	-	-	-	-	**1.97**	1.57	0.96	1.30	1.84	1.48	0.30	0.50
VAR	-	-	-	-	-	-	-	-0.08	-0.57	-0.95	0.15	-0.79	-0.67	-1.06
MSVAR	-	-	-	-	-	-	-	-	-0.59	-1.43	0.24	-0.79	-0.82	-1.30
SPLITVAR	-	-	-	-	-	-	-	-	-	0.09	0.56	0.12	-0.46	-0.78
MLPVAR	-	-	-	-	-	-	-	-	-	-	1.06	0.13	-0.52	-0.94
FAVAR	-	-	-	-	-	-	-	-	-	-	-	**-3.15**	-0.85	-1.37
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-0.49	-0.83
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	0.04

emp: 1 ahead	**AR**	**MA**	**ARMA**	**MSAR**	**SPLITAR**	**MLPAR**	**VAR**	**MSVAR**	**SPLITVAR**	**MLPVAR**	**FAVAR**	**MSFAVAR**	**SPLITFAVAR**	**MLPFAVAR**
BASE	0.02	1.15	0.04	-0.24	0.58	0.79	0.11	-0.15	0.62	0.42	0.24	-0.99	0.03	0.55
AR	-	**2.59**	0.15	-0.83	1.28	1.58	0.59	-0.60	1.49	1.18	0.36	-1.45	0.01	0.77
MA	-	-	**-2.96**	**-2.23**	-1.46	-0.57	**-2.78**	**-2.23**	-1.26	-1.64	-1.45	**-2.35**	-1.53	-1.10
ARMA	-	-	-	-0.75	1.38	1.64	0.75	-0.57	1.60	1.25	0.35	-1.41	-0.01	0.77
MSAR	-	-	-	-	1.37	1.59	0.96	1.06	1.48	1.34	0.70	-1.27	0.32	1.03
SPLITAR	-	-	-	-	-	0.70	-1.28	-1.30	0.34	-0.40	-0.66	-1.90	-0.82	-0.18
MLPAR	-	-	-	-	-	-	-1.51	-1.53	-0.53	-0.81	-0.84	-1.81	-0.92	-0.51
VAR	-	-	-	-	-	-	-	-0.83	1.52	1.11	0.24	-1.59	-0.12	0.69
MSVAR	-	-	-	-	-	-	-	-	1.43	1.28	0.62	-1.50	0.22	0.96
SPLITVAR	-	-	-	-	-	-	-	-	-	-0.66	-0.80	**-2.03**	-0.95	-0.31
MLPVAR	-	-	-	-	-	-	-	-	-	-	-0.40	-1.90	-0.65	0.11
FAVAR	-	-	-	-	-	-	-	-	-	-	-	**-2.16**	-0.39	0.87
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	1.09	**2.29**
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	0.72

Table B.4 DM test statistics for the CPI and employment out-of-sample forecast performed with an expanding window ranging from 2004-09-01 to 2014-11-01. A positive number indicates that the model on the row outperforms the model on the column. Bolded statistics are significant at 95%

emp: 4 ahead	AR	MA	ARMA	MSAR	SPLITAR	MLPAR	VAR	MSVAR	SPLITVAR	MLPVAR	FAVAR	MSFAVAR	SPLITFAVAR	MLPFAVAR
BASE	-0.19	1.61	-0.03	1.09	0.36	0.24	0.11	-0.10	0.28	0.27	0.06	-1.10	-0.93	0.48
AR	-	**2.18**	1.19	**1.99**	1.05	1.13	1.59	0.50	1.20	1.94	1.14	-0.90	-1.10	1.69
MA	-	-	**-2.23**	-1.58	**-2.13**	**-2.29**	**-2.21**	**-2.13**	**-2.10**	**-2.14**	**-2.06**	**-2.03**	**-2.04**	**-1.96**
ARMA	-	-	-	1.77	0.80	0.94	**2.11**	-0.21	0.85	1.66	0.33	-0.98	-1.16	1.28
MSAR	-	-	-	-	-1.18	-1.24	-1.53	**-1.99**	-1.45	-1.39	-1.53	**-2.20**	**-2.09**	-1.08
SPLITAR	-	-	-	-	-	-0.32	-0.51	-0.95	-0.48	-0.26	-0.54	-1.33	-1.25	0.15
MLPAR	-	-	-	-	-	-	-0.51	-0.82	0.01	-0.04	-0.46	-1.15	-1.18	0.40
VAR	-	-	-	-	-	-	-	-0.66	0.44	0.84	-0.25	-1.14	-1.30	0.87
MSVAR	-	-	-	-	-	-	-	-	0.95	1.48	0.57	-1.37	-1.35	1.79
SPLITVAR	-	-	-	-	-	-	-	-	-	-0.04	-0.48	-1.37	-1.32	0.47
MLPVAR	-	-	-	-	-	-	-	-	-	-	-0.71	-1.48	-1.52	0.60
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-1.42	-1.74	1.11
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-0.29	**1.96**
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	1.80

emp: 12 ahead	AR	MA	ARMA	MSAR	SPLITAR	MLPAR	VAR	MSVAR	SPLITVAR	MLPVAR	FAVAR	MSFAVAR	SPLITFAVAR	MLPFAVAR
BASE	-0.62	-0.17	-0.54	0.56	-1.10	-0.92	-0.46	-0.54	-1.08	-0.25	-0.45	-0.90	-0.58	-0.42
AR	-	1.44	1.01	1.53	-0.46	-0.89	1.02	1.65	-0.48	**2.75**	1.08	-1.16	-0.09	1.49
MA	-	-	-1.50	0.88	-0.78	-1.25	-1.64	-1.22	-0.82	-0.27	-1.59	-1.35	-1.08	-0.87
ARMA	-	-	-	1.42	-0.51	-0.93	1.01	0.53	-0.53	**2.90**	1.10	-1.13	-0.36	1.13
MSAR	-	-	-	-	-1.84	-1.86	-1.30	-1.39	-1.92	-1.04	-1.29	-1.85	-1.47	-1.29
SPLITAR	-	-	-	-	-	0.25	0.56	0.58	0.24	0.86	0.57	0.23	0.42	0.67
MLPAR	-	-	-	-	-	-	0.96	1.16	-0.23	1.65	1.00	0.03	0.57	1.28
VAR	-	-	-	-	-	-	-	-0.24	-0.58	**2.17**	0.46	-1.11	-0.58	0.33
MSVAR	-	-	-	-	-	-	-	-	-0.61	**2.66**	0.33	-1.46	-0.48	0.47
SPLITVAR	-	-	-	-	-	-	-	-	-	0.92	0.60	0.21	0.43	0.71
MLPVAR	-	-	-	-	-	-	-	-	-	-	**-1.96**	**-2.00**	-1.57	-1.19
FAVAR	-	-	-	-	-	-	-	-	-	-	-	-1.15	-0.63	0.25
MSFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	0.73	1.58
SPLITFAVAR	-	-	-	-	-	-	-	-	-	-	-	-	-	1.28

Table B.5 DM test statistics for the employment out-of-sample forecast performed with an expanding window ranging from 2004-09-01 to 2014-11-01. A positive number indicates that the model on the row outperforms the model on the column. Bolded statistics are significant at 95%

Appendix C

Descriptive Statistics

C.1 Quarterly

	GDP	CPI	FF	IP	emp	unemp
count	223.0	223.0	223.0	223.0	223.0	223.0
mean	8689.309	115.181	5.342	63.461	99537.444	6.096
std	4035.425	68.047	3.643	25.473	27953.039	1.585
min	2976.6	28.98	0.07	22.725	52480.0	3.4
max	16094.5	237.424	19.08	105.492	139068.0	10.4

Table C.1 Descriptive Statistics of the quarterly time series

	GDP	CPI	FF	IP	emp	unemp
count	221.0	221.0	221.0	221.0	221.0	221.0
mean	0.008	0.019	-0.013	0.007	0.004	6.1
std	0.008	0.015	1.197	0.018	0.006	1.591
min	-0.021	-0.033	-8.58	-0.083	-0.017	3.4
max	0.039	0.073	6.27	0.092	0.018	10.4
adfuller	0.0	0.363	0.0	0.002	0.0	0.005

Table C.2 Descriptive Statistics of the quarterly time series after transformations. The adfuller row shows the p-value of the existence of an unit-root

C.2 Monthly

	CPI	FF	IP	emp	unemp
count	671.0	671.0	671.0	671.0	671.0
mean	115.846	5.319	63.732	99789.542	6.096
std	68.241	3.595	25.52	27936.982	1.587
min	28.97	0.07	22.725	52480.0	3.4
max	237.506	19.1	106.613	140117.0	10.8

Table C.3 Descriptive Statistics of the monthly time series

	CPI	FF	IP	emp	unemp
count	670.0	670.0	670.0	670.0	670.0
mean	0.003	-0.004	0.002	0.001	6.096
std	0.003	0.533	0.008	0.002	1.588
min	-0.018	-6.63	-0.044	-0.009	3.4
max	0.018	3.06	0.06	0.012	10.8
adfuller	0.101	0.0	0.0	0.0	0.029

Table C.4 Descriptive Statistics of the monthly time series after transformations. The adfuller row shows the p-value of the existence of an unit-root

Appendix D

Stock & Watson Transformations

This appendix shows the transformations performed on the dataset used to generated the factors, as is done in Stock and Watson (2016).

Factor Models and Structural Vector Autoregressions in Macroeconomics

James H. Stock and Mark W. Watson

March 11, 2016

–- Data Appendix –-

Table A.1 below lists the 207 data series used in Sections 6 and 7. The data series can be found in the Excel file *hom_fac_1.xlsx* in the replication materials. Unless otherwise noted in Table A.1 or *hom_fac_1.xlsx*, the data were downloaded from FRED. We carried out several transformations in the order listed below.

Seasonal Adjustment: Many of the series were seasonally adjusted at the source, but a few were not and yet showed substantial seasonality. We seasonally adjusted these series using the X-11 feature in RATS. Where applicable, this is noted in the series description in Table A.1.

The results Sections 6 and 7 were computed using a series of Matlab programs available in the replication materials. The data are input from the Excel file into Matlab using the programs *datain_all.m* and *datain_real.m*. These programs carry out the following transformations.

Price Deflation: Many of the nominal series were deflated using the price deflator for person consumption expenditures (PCE) or PCE less food and energy (core-PCE). Where applicable, this is noted in the series description in Table A.1.

Temporal Aggregation: Many of the series were available monthly, and these series were converted to quarterly values by averaging the quarter's three monthly values. (See the program *mtoq.m* in the replication files.)

Logs, Differences, etc.: The series were converted to first differences, growth rates, etc. using the transformation listed in the column labeled "T" in Table A.1. (See the program *transx.m* in the replication files.) Letting x denote the original series and y denote the transformed series these transformations are: (1) $y_t = x_t$, (2) $y_t = (1\text{-L})x_t$, (3) $y_t = (1\text{-L})^2 x_t$, (4) $y_t = \ln(x_t)$, (5) $y_t = (1\text{-L})\ln(x_t)$, (6) $y_t = (1\text{-L})^2\ln(x_t)$.

Outliers: Some series were examined for outliers. These series are indicated by a non-zero value in the column labeled "O" on the table. Outliers were defined as values satisfying $|y_t - m|/iqr > \kappa$, where m is the sample median, iqr is the sample interquartile

range, and κ is a parameter. Series with O = 1 in the table used $\kappa = 4.5$, and series with O=2 used $\kappa = 3$. Outliers were replaced with the median of the 5 preceding observations. (See the program *adjout.m* in the replication files.)

Table A.1: Data Series

	Name	Description	Sample Period	T	O	F
	(1) NIPA					
1	GDP	Real Gross Domestic Product 3 Decimal	1959:Q1-2014:Q4	5	0	0
2	Consumption	Real Personal Consumption Expenditures	1959:Q1-2014:Q4	5	0	0
3	Cons:Dur	Real Personal Consumption Expenditures: Durable Goods Quantity Index	1959:Q1-2014:Q4	5	0	1
4	Cons:Svc	Real Personal Consumption Expenditures: Services Quantity Index	1959:Q1-2014:Q4	5	0	1
5	Cons:NonDur	Real Personal Consumption Expenditures: Nondurable Goods Quantity Index	1959:Q1-2014:Q4	5	0	1
6	Investment	Real Gross Private Domestic Investment 3 Decimal	1959:Q1-2014:Q4	5	0	0
7	FixedInv	Real Private Fixed Investment Quantity Index	1959:Q1-2014:Q4	5	0	0
8	Inv:Equip	Real Nonresidential Investment: Equipment Quantity Idenx	1959:Q1-2014:Q4	5	0	1
9	FixInv:NonRes	Real Private Nonresidential Fixed Investment Quantity Index	1959:Q1-2014:Q4	5	0	1
10	FixedInv:Res	Real Private Residential Fixed Investment Quantity Index	1959:Q1-2014:Q4	5	0	1
11	Ch. Inv/GDP	Change in Inventories /GDP	1959:Q1-2014:Q4	1	0	1
12	Gov.Spending	Real Government Consumption Expenditures & Gross Investment 3 Decimal	1959:Q1-2014:Q4	5	0	0
13	Gov:Fed	Real Federal Consumption Expenditures Quantity Index	1959:Q1-2014:Q4	5	0	1
14	Real_Gov Receipts	Government Current Receipts (Nominal) Defl by GDP Deflator	1959:Q1-2014:Q3	5	0	1
15	Gov:State&Local	Real State & Local Consumption Expenditures Quantity Index	1959:Q1-2014:Q4	5	0	1
16	Exports	Real Exports of Goods & Services 3 Decimal	1959:Q1-2014:Q4	5	0	1
17	Imports	Real Imports of Goods & Services 3 Decimal	1959:Q1-2014:Q4	5	0	1
18	Disp-Income	Real Disposable Personal Income	1959:Q1-2014:Q4	5	0	0
19	Ouput:NFB	Nonfarm Business Sector: Output	1959:Q1-2014:Q4	5	0	0
20	Output:Bus	Business Sector: Output	1959:Q1-2014:Q4	5	0	0
	(2) Industrial Production					
21	IP: Total index	IP: Total index	1959:Q1-2014:Q4	5	0	0
22	IP: Final products	Industrial Production: Final Products (Market Group)	1959:Q1-2014:Q4	5	0	0
23	IP: Consumer goods	IP: Consumer goods	1959:Q1-2014:Q4	5	0	0
24	IP: Materials	Industrial Production: Materials	1959:Q1-2014:Q4	5	0	0
25	IP: Dur gds materials	Industrial Production: Durable Materials	1959:Q1-2014:Q4	5	0	1
26	IP: Nondur gds materials	Industrial Production: nondurable Materials	1959:Q1-2014:Q4	5	0	1
27	IP: Dur Cons. Goods	Industrial Production: Durable Consumer Goods	1959:Q1-2014:Q4	5	0	1
28	IP: Auto	IP: Automotive products	1959:Q1-2014:Q4	5	0	1
29	IP:NonDur Cons God	Industrial Production: Nondurable Consumer Goods	1959:Q1-2014:Q4	5	0	1

30	IP: Bus Equip	Industrial Production: Business Equipment	1959:Q1-2014:Q4	5	0	1
31	Capu Tot	Capacity Utilization: Total Industry	1967:Q1-2014:Q4	1	0	1
	(3) Employment and Unemployment					
32	Emp:Nonfarm	Total Nonfarm Payrolls: All Employees	1959:Q1-2014:Q4	5	0	0
33	Emp: Private	All Employees: Total Private Industries	1959:Q1-2014:Q4	5	0	0
34	Emp: mfg	All Employees: Manufacturing	1959:Q1-2014:Q4	5	0	0
35	Emp:Services	All Employees: Service-Providing Industries	1959:Q1-2014:Q4	5	0	0
36	Emp:Goods	All Employees: Goods-Producing Industries	1959:Q1-2014:Q4	5	0	0
37	Emp: DurGoods	All Employees: Durable Goods Manufacturing	1959:Q1-2014:Q4	5	0	1
38	Emp: Nondur Goods	All Employees: Nondurable Goods Manufacturing	1959:Q1-2014:Q4	5	0	0
39	Emp: Const	All Employees: Construction	1959:Q1-2014:Q4	5	0	1
40	Emp: Edu&Health	All Employees: Education & Health Services	1959:Q1-2014:Q4	5	0	1
41	Emp: Finance	All Employees: Financial Activities	1959:Q1-2014:Q4	5	0	1
42	Emp: Infor	All Employees: Information Services	1959:Q1-2014:Q4	5	1	1
43	Emp: Bus Serv	All Employees: Professional & Business Services	1959:Q1-2014:Q4	5	0	1
44	Emp:Leisure	All Employees: Leisure & Hospitality	1959:Q1-2014:Q4	5	0	1
45	Emp:OtherSvcs	All Employees: Other Services	1959:Q1-2014:Q4	5	0	1
46	Emp: Mining/NatRes	All Employees: Natural Resources & Mining	1959:Q1-2014:Q4	5	1	1
47	Emp:Trade&Trans	All Employees: Trade Transportation & Utilities	1959:Q1-2014:Q4	5	0	1
48	Emp: Gov	All Employees: Government	1959:Q1-2014:Q4	5	0	0
49	Emp:Retail	All Employees: Retail Trade	1959:Q1-2014:Q4	5	0	1
50	Emp:Wholesal	All Employees: Wholesale Trade	1959:Q1-2014:Q4	5	0	1
51	Emp: Gov(Fed)	Employment Federal Government	1959:Q1-2014:Q4	5	2	1
52	Emp: Gov (State)	Employment State government	1959:Q1-2014:Q4	5	0	1
53	Emp: Gov (Local)	Employment Local government	1959:Q1-2014:Q4	5	0	1
54	Emp: Total (HHSurve)	Emp Total (Household Survey)	1959:Q1-2014:Q4	5	0	0
55	LF Part Rate	LaborForce Participation Rate (16 Over) SA	1959:Q1-2014:Q4	2	0	0
56	Unemp Rate	Urate	1959:Q1-2014:Q4	2	0	0
57	Urate_ST	Urate Short Term (< 27 weeks)	1959:Q1-2014:Q4	2	0	0
58	Urate_LT	Urate Long Term (>= 27 weeks)	1959:Q1-2014:Q4	2	0	0
59	Urate: Age16-19	Unemployment Rate - 16-19 yrs	1959:Q1-2014:Q4	2	0	1
60	Urate:Age>20 Men	Unemployment Rate - 20 yrs. & over Men	1959:Q1-2014:Q4	2	0	1
61	Urate: Age>20 Women	Unemployment Rate - 20 yrs. & over Women	1959:Q1-2014:Q4	2	0	1
62	U: Dur<5wks	Number Unemployed for Less than 5 Weeks	1959:Q1-2014:Q4	5	0	1
63	U:Dur5-14wks	Number Unemployed for 5-14 Weeks	1959:Q1-2014:Q4	5	0	1

64	U:dur>15-26wks	Civilians Unemployed for 15-26 Weeks	1959:Q1-2014:Q4	5	0	1
65	U: Dur>27wks	Number Unemployed for 27 Weeks & over	1959:Q1-2014:Q4	5	0	1
66	U: Job losers	Unemployment Level - Job Losers	1967:Q1-2014:Q4	5	0	1
67	U: LF Reenty	Unemployment Level - Reentrants to Labor Force	1967:Q1-2014:Q4	5	1	1
68	U: Job Leavers	Unemployment Level - Job Leavers	1967:Q1-2014:Q4	5	0	1
69	U: New Entrants	Unemployment Level - New Entrants	1967:Q1-2014:Q4	5	1	1
70	Emp:SlackWk	Employment Level - Part-Time for Economic Reasons All Industries	1959:Q1-2014:Q4	5	1	1
71	EmpHrs:Bus Sec	Business Sector: Hours of All Persons	1959:Q1-2014:Q4	5	0	0
72	EmpHrs:nfb	Nonfarm Business Sector: Hours of All Persons	1959:Q1-2014:Q4	5	0	0
73	AWH Man	Average Weekly Hours: Manufacturing	1959:Q1-2014:Q4	1	0	1
74	AWH Privat	Average Weekly Hours: Total Private Industry	1964:Q1-2014:Q4	2	0	1
75	AWH Overtime	Average Weekly Hours: Overtime: Manufacturing	1959:Q1-2014:Q4	2	0	1
76	HelpWnted	Index of Help-Wanted Advertising in Newspapers (Data truncated in 2000)	1959:Q1-1999:Q4	1	0	0
	(4) Orders, Inventories, and Sales					
77	MT Sales	Manufacturing and trade sales (mil. Chain 2005 $)	1959:Q1-2014:Q3	5	0	0
78	Ret. Sale	Sales of retail stores (mil. Chain 2000 $)	1959:Q1-2014:Q3	5	0	1
79	Orders (DurMfg)	Mfrs' new orders durable goods industries (bil. chain 2000 $)	1959:Q1-2014:Q4	5	0	1
80	Orders (Cons. Gds & Mat.)	Mfrs' new orders consumer goods and materials (mil. 1982 $)	1959:Q1-2014:Q4	5	0	1
81	UnfOrders(DurGds)	Mfrs' unfilled orders durable goods indus. (bil. chain 2000 $)	1959:Q1-2014:Q4	5	0	1
82	Orders(NonDefCap)	Mfrs' new orders nondefense capital goods (mil. 1982 $)	1959:Q1-2014:Q4	5	0	1
83	VendPerf	ISM Manufacturing: Supplier Deliveries Index©	1959:Q1-2014:Q4	1	0	1
84	NAPM:INV	ISM Manufacturing: Inventories Index©	1959:Q1-2014:Q4	1	0	1
85	NAPM:ORD	ISM Manufacturing: New Orders Index©; Index;	1959:Q1-2014:Q4	1	0	1
86	MT Invent	Manufacturing and trade inventories (bil. Chain 2005 $)	1959:Q1-2014:Q3	5	0	1
	(5) Housing Starts and Permits					
87	Hstarts	Housing Starts: Total: New Privately Owned Housing Units Started	1959:Q1-2014:Q3	5	0	0
88	Hstarts >5units	Privately Owned Housing Starts: 5-Unit Structures or More	1959:Q1-2014:Q3	5	0	0
89	Hpermits	New Private Housing Units Authorized by Building Permit	1960:Q1-2014:Q4	5	0	1
90	Hstarts:MW	Housing Starts in Midwest Census Region	1959:Q1-2014:Q3	5	0	1
91	Hstarts:NE	Housing Starts in Northeast Census Region	1959:Q1-2014:Q3	5	0	1
92	Hstarts:S	Housing Starts in South Census Region	1959:Q1-2014:Q3	5	0	1
93	Hstarts:W	Housing Starts in West Census Region	1959:Q1-2014:Q3	5	0	1
94	Constr. Contracts	Construction contracts (mil. sq. ft.) (Copyright McGraw-Hill)	1963:Q1-2014:Q4	4	0	1
	(6) Prices					
95	PCED	Personal Consumption Expenditures: Chain-type Price Index	1959:Q1-2014:Q4	6	0	0

96	PCED_LFE	Personal Consumption Expenditures: Chain-type Price Index Less Food and Energy	1959:Q1-2014:Q4	6	0	0
97	GDP Defl	Gross Domestic Product: Chain-type Price Index	1959:Q1-2014:Q4	6	0	0
98	GPDI Defl	Gross Private Domestic Investment: Chain-type Price Index	1959:Q1-2014:Q4	6	0	1
99	BusSec Defl	Business Sector: Implicit Price Deflator	1959:Q1-2014:Q4	6	0	1
100	PCED_Goods	Goods	1959:Q1-2014:Q4	6	0	0
101	PCED_DurGoods	Durable goods	1959:Q1-2014:Q4	6	0	0
102	PCED_NDurGoods	Nondurable goods	1959:Q1-2014:Q4	6	0	0
103	PCED_Serv	Services	1959:Q1-2014:Q4	6	0	0
104	PCED_HouseholdServices	Household consumption expenditures (for services)	1959:Q1-2014:Q4	6	0	0
105	PCED_MotorVec	Motor vehicles and parts	1959:Q1-2014:Q4	6	0	1
106	PCED_DurHousehold	Furnishings and durable household equipment	1959:Q1-2014:Q4	6	0	1
107	PCED_Recreation	Recreational goods and vehicles	1959:Q1-2014:Q4	6	0	1
108	PCED_OthDurGds	Other durable goods	1959:Q1-2014:Q4	6	0	1
109	PCED_Food_Bev	Food and beverages purchased for off-premises consumption	1959:Q1-2014:Q4	6	0	1
110	PCED_Clothing	Clothing and footwear	1959:Q1-2014:Q4	6	0	1
111	PCED_Gas_Enrgy	Gasoline and other energy goods	1959:Q1-2014:Q4	6	0	1
112	PCED_OthNDurGds	Other nondurable goods	1959:Q1-2014:Q4	6	0	1
113	PCED_Housing-Utilities	Housing and utilities	1959:Q1-2014:Q4	6	0	1
114	PCED_HealthCare	Health care	1959:Q1-2014:Q4	6	0	1
115	PCED_TransSvg	Transportation services	1959:Q1-2014:Q4	6	0	1
116	PCED_RecServices	Recreation services	1959:Q1-2014:Q4	6	0	1
117	PCED_FoodServ_Acc.	Food services and accommodations	1959:Q1-2014:Q4	6	0	1
118	PCED_FIRE	Financial services and insurance	1959:Q1-2014:Q4	6	0	1
119	PCED_OtherServices	Other services	1959:Q1-2014:Q4	6	0	1
120	CPI	Consumer Price Index For All Urban Consumers: All Items	1959:Q1-2014:Q4	6	0	0
121	CPI_LFE	Consumer Price Index for All Urban Consumers: All Items Less Food & Energy	1959:Q1-2014:Q4	6	0	0
122	PPI:FinGds	Producer Price Index: Finished Goods	1959:Q1-2014:Q4	6	0	0
123	PPI	Producer Price Index: All Commodities	1959:Q1-2014:Q3	6	0	0
124	PPI:FinConsGds	Producer Price Index: Finished Consumer Goods	1959:Q1-2014:Q4	6	0	1
125	PPI:FinConsGds (Food)	Producer Price Index: Finished Consumer Foods	1959:Q1-2014:Q4	6	0	1
126	PPI:IndCom	Producer Price Index: Industrial Commodities	1959:Q1-2014:Q4	6	0	1
127	PPI:IntMat	Producer Price Index: Intermediate Materials: Supplies & Components	1959:Q1-2014:Q4	6	0	1

128	Real_P:SensMat	Index of Sensitive Matrerials Prices (Discontinued) Defl by PCE(LFE) Def	1959:Q1-2004:Q1	5	0	1
129	Real_Commod: spot price	Spot market price index:BLS & CRB: all commodities(1967=100) Defl by PCE(LFE)	1959:Q1-2009:Q1	5	0	0
130	NAPM com price	ISM Manufacturing: Prices Paid Index©	1959:Q1-2014:Q4	1	0	1
131	Real_Price:NatGas	PPI: Natural Gas Defl by PCE(LFE)	1967:Q1-2014:Q4	5	0	1
	(7) Productivity and Earnings					
132	Real_AHE:PrivInd	Average Hourly Earnings: Total Private Industries Defl by PCE(LFE)	1964:Q1-2014:Q4	5	0	0
133	Real_AHE:Const	Average Hourly Earnings: Construction Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
134	Real_AHE:MFG	Average Hourly Earnings: Manufacturing Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
135	CPH:NFB	Nonfarm Business Sector: Real Compensation Per Hour	1959:Q1-2014:Q4	5	0	1
136	CPH:Bus	Business Sector: Real Compensation Per Hour	1959:Q1-2014:Q4	5	0	1
137	OPH:nfb	Nonfarm Business Sector: Output Per Hour of All Persons	1959:Q1-2014:Q4	5	0	1
138	OPH:Bus	Business Sector: Output Per Hour of All Persons	1959:Q1-2014:Q4	5	0	0
139	ULC:Bus	Business Sector: Unit Labor Cost	1959:Q1-2014:Q4	5	0	0
140	ULC:NFB	Nonfarm Business Sector: Unit Labor Cost	1959:Q1-2014:Q4	5	0	1
141	UNLPay:nfb	Nonfarm Business Sector: Unit Nonlabor Payments	1959:Q1-2014:Q4	5	0	1
	(8) Interest Rates					
142	FedFunds	Effective Federal Funds Rate	1959:Q1-2014:Q4	2	0	1
143	TB-3Mth	3-Month Treasury Bill: Secondary Market Rate	1959:Q1-2014:Q4	2	0	1
144	TM-6MTH	6-Month Treasury Bill: Secondary Market Rate	1959:Q1-2014:Q4	2	0	0
145	EuroDol3M	3-Month Eurodollar Deposit Rate (London)	1971:Q1-2014:Q4	2	0	0
146	TB-1YR	1-Year Treasury Constant Maturity Rate	1959:Q1-2014:Q4	2	0	0
147	TB-10YR	10-Year Treasury Constant Maturity Rate	1959:Q1-2014:Q4	2	0	0
148	Mort-30Yr	30-Year Conventional Mortgage Rate	1971:Q2-2014:Q4	2	0	0
149	AAA Bond	Moody's Seasoned Aaa Corporate Bond Yield	1959:Q1-2014:Q4	2	0	0
150	BAA Bond	Moody's Seasoned Baa Corporate Bond Yield	1959:Q1-2014:Q4	2	0	0
151	BAA_GS10	BAA-GS10 Spread	1959:Q1-2014:Q4	1	0	1
152	MRTG_GS10	Mortg-GS10 Spread	1971:Q2-2014:Q4	1	0	1
153	tb6m_tb3m	tb6m-tb3m	1959:Q1-2014:Q4	1	0	1
154	GS1_tb3m	GS1_Tb3m	1959:Q1-2014:Q4	1	0	1
155	GS10_tb3m	GS10_Tb3m	1959:Q1-2014:Q4	1	0	1
156	CP_Tbill Spread	CP3FM-TB3MS	1959:Q1-2014:Q4	1	0	1
157	Ted_spr	MED3-TB3MS (Version of TED Spread)	1971:Q1-2014:Q4	1	0	1
158	gz_spread	Gilchrist-Zakrajsek Spread (Unadjusted)	1973:Q1-2012:Q4	1	0	0
159	gz_ebp	Gilchrist-Zakrajsek Excess Bond Premium	1973:Q1-2012:Q4	1	0	1
	(9) Money and Credit					

160	Real_mbase	St. Louis Adjusted Monetary Base; Bil. of $; M; SA; Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
161	Real_InsMMF	Institutional Money Funds Defl by PCE(LFE)	1980:Q1-2014:Q4	5	0	0
162	Real_m1	M1 Money Stock Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
163	Real_m2	M2SL Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
164	Real_mzm	MZM Money Stock Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
165	Real_C&Lloand	Commercial and Industrial Loans at All Commercial Banks Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	1
166	Real_ConsLoans	Consumer (Individual) Loans at All Commercial Banks/ Outlier Code because of change in data in April 2010. See FRB H8 Release Defl by PCE(LFE)	1959:Q1-2014:Q4	5	1	1
167	Real_NonRevCredit	Total Nonrevolving Credit Outstanding Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	1
168	Real_LoansRealEst	Real Estate Loans at All Commercial Banks Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	1
169	Real_RevolvCredit	Total Revolving Credit Outstanding Defl by PCE(LFE)	1968:Q1-2014:Q4	5	1	1
170	Real_ConsuCred	Total Consumer Credit Outstanding Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	0
171	FRBSLO_Consumers	FRB Senior Loans Officer Opions. Net Percentage of Domestic Respondents Reporting Increased Willingness to Make Consumer Installment Loans (Fred from 1982:Q2 on Earlier is DB series)	1970:Q1-2014:Q4	1	0	1
	(10) International Variables					
172	Ex rate: major	FRB Nominal Major Currencies Dollar Index (Linked to EXRUS in 1973:1)	1959:Q1-2014:Q4	5	0	1
173	Ex rate: Euro	U.S. / Euro Foreign Exchange Rate	1999:Q1-2014:Q4	5	0	1
174	Ex rate: Switz	Foreign exchange rate: Switzerland (Swiss franc per U.S.$) Fred 1971. EXRSW previous	1971:Q1-2014:Q4	5	0	1
175	Ex rate: Japan	Foreign exchange rate: Japan (yen per U.S.$) Fred 1971- EXRJAN previous	1971:Q1-2014:Q4	5	0	1
176	Ex rate: UK	Foreign exchange rate: United Kingdom (cents per pound) Fred 1971-> EXRUK Previous	1971:Q1-2014:Q4	5	0	1
177	EX rate: Canada	Foreign exchange rate: Canada (Canadian $ per U.S.$) Fred 1971 -> EXRCAN previous	1971:Q1-2014:Q4	5	0	1
178	OECD GDP	OECD: Gross Domestic Product by Expenditure in Constant Prices: Total Gross; Growth Rate (Quartely); Fred Series NAEXKP01O1Q657S	1961:Q2-2013:Q4	1	0	1
179	IP Europe	OECD: Total Ind. Prod (excl Construction) Europe Growth Rate (Quarterly); Fred Series PRINTO01OEQ657S	1960:Q2-2013:Q4	1	0	1
180	Global Ec Activity	Kilian's estimate of glaobal economic activity in industrial commodity markets (Kilian website)	1968:Q1-2014:Q4	1	0	1
	(11) **Asset Prices, Wealth, and Household Balance Sheets**					
181	S&P 500	S&P's Common Stock Price Index: Composite (1941-43=10)	1959:Q1-2014:Q4	5	0	1
182	Real_HHW:TA	Households and nonprofit organizations; total assets (FoF) Seasonally Adjusted (RATS X11) Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	0
183	Real_HHW:TL	Households and nonprofit organizations; total liabilities Seasonally Adjusted (RATS X11) Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	1
184	liab_PDI	Liabilities Relative to Person Disp Income	1959:Q1-2014:Q3	5	0	0
185	Real_HHW:W	Households and nonprofit organizations; net worth (FoF) Seasonally Adjusted (RATS X11)	1959:Q1-2014:Q3	5	0	1

		Defl by PCE(LFE)				
186	W_PDI	Networth Relative to Personal Disp Income	1959:Q1-2014:Q3	1	0	0
187	Real_HHW:TFA	Households and nonprofit organizations; total financial assets Seasonally Adjusted (RATS X11) Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	0
188	Real_HHW:TA_RE	TotalAssets minus Real Estate Assets Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	1
189	Real_HHW:TNFA	Households and nonprofit organizations; total nonfinancial assets (FoF) Seasonally Adjusted (RATS X11) Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	0
190	Real_HHW:RE	Households and nonprofit organizations; real estate at market value Seasonally Adjusted (RATS X11) Defl by PCE(LFE)	1959:Q1-2014:Q3	5	0	1
191	DJIA	Common Stock Prices: Dow Jones Industrial Average	1959:Q1-2014:Q4	5	0	1
192	VXO	VXO (Linked by N. Bloom) .. Average daily VIX from 2009 ->	1962:Q3-2014:Q4	1	0	1
193	Real_Hprice:OFHEO	House Price Index for the United States Defl by PCE(LFE)	1975:Q1-2014:Q4	5	0	1
194	Real_CS_10	Case-Shiller 10 City Average Defl by PCE(LFE)	1987:Q1-2014:Q4	5	0	1
195	Real_CS_20	Case-Shiller 20 City Average Defl by PCE(LFE)	2000:Q1-2014:Q4	5	0	1
	(12) Other					
196	Cons. Expectations	Consumer expectations NSA (Copyright University of Michigan)	1959:Q1-2014:Q4	1	0	1
197	PoilcyUncertainty	Baker Bloom Davis Policy Uncertainty Index	1985:Q1-2014:Q4	2	0	1
	(13) Oil Market Variables					
198	World Oil Production	World Oil Production.1994:Q1 on from EIA (Crude Oil including Lease Condensate); Data prior to 1994 from From Baumeister and Peerlman (2013)	1959:Q1-2014:Q3	5	0	0
199	World Oil Production	World Oil Production.1994:Q1 on from EIA (Crude Oil including Lease Condensate); Data prior to 1994 from From Baumeister and Peerlman (2013); Seasonally adjusted using RATS X11 (note seasonality before 1970)	1959:Q1-2014:Q3	5	0	1
200	IP: Energy Prds	IP: Consumer Energy Products	1959:Q1-2014:Q4	5	0	1
201	Petroleum Stocks	U.S. Ending Stocks excluding SPR of Crude Oil and Petroleum Products (Thousand Barrels); SA using X11 in RATS	1959:Q1-2014:Q4	5	0	1
202	Real_Price:Oil	PPI: Crude Petroleum Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	1
203	Real_Crudeoil Price	Crude Oil: West Texas Intermediate (WTI) - Cushing Oklahoma Defl by PCE(LFE)	1986:Q1-2014:Q4	5	0	1
204	Real_CrudeOil	Crude Oil Prices: Brent - Europe Defl by PCE(LFE) Def	1987:Q3-2014:Q4	5	0	1
205	Real_Price Gasoline	Conventional Gasoline Prices: New York Harbor Regular Defl by PCE(LFE)	1986:Q3-2014:Q4	5	0	1
206	Real_Refiners Acq. Cost (Imports)	U.S. Crude Oil Imported Acquisition Cost by Refiners (Dollars per Barrel) Defl by PCE(LFE)	1974:Q1-2014:Q4	5	0	1
207	Real_CPI Gasoline	CPI Gasoline (NSA) BLS: CUUR0000SETB01 Defl by PCE(LFE)	1959:Q1-2014:Q4	5	0	1