GDP growth rate nowcasting and forecasting

A system averaging model implementation

Author: Fredrik Björnfot
Supervisor: Øystein Børsum
Examiner: Markus Ådahl

Umeå university
February 2017
Abstract

English

The main purpose of this project was to help Swedbank get a better understanding of how gross domestic product growth rate develops in the future from a data set of macroeconomic variables. Since GDP values are released long after a quarter has ended Swedbank would like to have a model that could predict upcoming GDP from these data sets. This was solved by a combination of growth rate predictions from a dynamic factor model, a vector autoregressive model and two machine learning models. The predictions were combined using a weighting method called system averaging model where the model prediction with least historical error receives the largest weight in the final future prediction. In previous work a simple moving average model has been implemented to achieve this effect however there are several flaws in a simple moving average model. Most of these defects could in theory be avoided by using an exponential weighting scheme instead. This resulted in the use of an exponential weighting method that is used to calculate weights for future predictions. The main conclusions from this project were that some predictions could get better when removing bad performing models which had too large of a weight. Putting too high weight on a single well performing model is also not optimal since the predictions could get very unstable because of varying model performance. The exponential weighting scheme worked well for some predictions however when the parameter $\lambda$, that controls how the weight is distributed between recent and historical errors, got too small a problem arose. Too few values were used to form the final weights for the prediction and the estimate got unsteady results.
Contents

1 Introduction 1
  1.1 Scope ................................................. 1
  1.2 Limitations ........................................... 1
  1.3 Background ............................................ 1
  1.4 Previous work ......................................... 2

2 Theory 3
  2.1 Dynamic factor analysis ................................. 3
    2.1.1 Method of principal components ................. 4
    2.1.2 Example of factor analysis ...................... 4
  2.2 Random forests .......................................... 5
    2.2.1 Bootstrap sampling ............................... 5
    2.2.2 Bootstrap aggregation ............................ 5
    2.2.3 Regression tree .................................. 5
    2.2.4 Definition of random forest ..................... 6
    2.2.5 Example of random forest ....................... 6
  2.3 Support vector machine .................................... 8
    2.3.1 Kernel functions ................................ 10
    2.3.2 Support vector machine example ................ 10
  2.4 System averaging model .................................. 10

3 Method 11
  3.1 Data .................................................. 12

4 Result 13
  4.1 Sweden Data October 27 ................................. 13
    4.1.1 Dynamic factor analysis .......................... 13
    4.1.2 Machine learning .................................. 14
    4.1.3 Error surface with VAR ........................... 16
    4.1.4 Error surface without VAR ....................... 17
    4.1.5 Combining all models with SAM .................... 18
  4.2 Sweden Data December 08 ............................... 20
    4.2.1 Dynamic factor analysis .......................... 20
    4.2.2 Machine learning .................................. 21
    4.2.3 Error surface with VAR ........................... 23
    4.2.4 Error surface without VAR ....................... 24
    4.2.5 Combining all model with SAM .................... 25
  4.3 United states Data October 27 .......................... 28
    4.3.1 Machine learning .................................. 29
    4.3.2 Error surface with VAR ........................... 31
    4.3.3 Error surface without VAR ....................... 32
    4.3.4 Combining all models with SAM .................... 33
  4.4 United states Data December 08 .......................... 36
    4.4.1 Dynamic factor analysis .......................... 36
1 Introduction

1.1 Scope

The main goal of this project is to create a model that can predict and nowcast gross domestic product growth rate from a large data set of macroeconomic variables. This will be done by first creating a random forest model, a support vector machine model and a dynamic factor model which combined with a vector autoregressive model created by Otto Lundberg should be able to estimate GDP growth rate values from these data sets. Each country do also have a unique set of data since the economies function differently, the final model should be able to use these different sets of underlying variables and still calculate a GDP growth rate estimation. The combination of all the models together should be able to estimate GDP growth rate with the assistance of a system of averaging model. Since there might arise some problems with data supply for developing countries the model will be created to predict GDP of developed countries with access to good data sets as a start.

1.2 Limitations

The main objective of the project is to create a model that could estimate GDP even though similar methods can be used to predict other economic variables such as inflation, purchasing managers index etc. The ability to get a solid GDP prediction from the program is most important and there is low priority in creating a user friendly program however it should be able to run without any setup in Matlab.

1.3 Background

The problem that many companies encounter when interested in the financial situation worldwide is that some financial variables are released months after a quarter has past. One of the most important of these variables is gross domestic product, GDP, which is a measure of the value of all final goods and services produced in a region during a period of time. Most commonly a region is defined by a country and the period of time in this report will be a quarter for convenience since GDP is released once every quarter. GDP is one of the primary indicators of an economies health. High GDP growth indicates that the economy is doing well and usually means that the companies and thereby the stocks in a country is rising in value. This typically means that unemployment rate is low and that wages are rising in the country. On the other hand if GDP growth is not increasing or increasing slowly it is a sign of a country where the economy is not doing so great and thereby company stocks in that country is probably not performing so well either.

In the textbook "Economics" Samuelson and Nordhaus described the importance of GDP. They made a comparison between how GDP had the ability to provide a good measure of the state of an economy to that of a weather satellite
which can examine weather over entire continents. Studying GDP value could make a great foundation for politicians and central banks to decide if an economy is contracting or expanding, if the economy needs a boost or restraint and whether a threat such as a recession or inflation is waiting around the corner. This is the reason that Swedbank needs a reliable method to calculate the gross domestic product in real time from other observed macroeconomic variables.

1.4 Previous work

In a report from Norges bank it was described how three different classes of methods together created a prediction with the use of a system averaging model or SAM. These three methods are a dynamic factor analysis, a vector autoregressive model and an indicator model. The system averaging model uses historical errors from past predictions to calculate weights for each model in upcoming future estimations. System averaging model provides some advantages as a weighting scheme mainly by simple calculations of weights for each model and the ease of adding additional models to the prediction. Norges bank came to the conclusion that a combination of different prediction methods created the best prediction of GDP. Norges bank model could also create predictions from one to four quarters into the future. (Aastveit, 2011) Swedbank is only interested in a model that could predict GDP growth rate for the not yet released GDP values and the upcoming GDP values from the current quarter. One of the most rapidly growing areas in big data analysis is machine learning. The idea here is simple, first let the machine find patterns in data sets then let the machine use these patterns to predict values from new data sets. In this report dynamic factor analysis will be examined closer together with two machine learning methods, random forest and support vector machine.

In a report from Stanford university the support vector machine was compared to neural networks and ARIMA(1,1,1) when predicting recessions with GDP growth. From this study the main results were that support vector machine performed better than both neural networks and ARIMA(1,1,1) at GDP prediction. However a relatively small set of macroeconomic variables was used in this study in opposite to the big data sets that are used in this report. (Islam, 2013)

Studying this previous work that has been done in this field of research there seem to be a good potential for the models used in this report to come up with a valuable estimate of GDP growth rate.
2 Theory

2.1 Dynamic factor analysis

The idea behind factor analysis is to describe the covariance relationships between a large set of variables with a few unobserved underlying factors. Basically variables with high correlations are grouped together to form a new variable, or factor, which can describe the variance in all of the variables in a group. (Johnson, 2007)

Dynamic factor analysis is a further development of this idea with the addition of dynamic factors that change over time. The model describes how a vector of \( N \) observed time series \( X_t \) evolves over time in terms of unobserved factors together with uncorrelated random terms that describe deviations such as measurement errors. The two ways of writing the DFM that is going to be discussed in this report is the dynamic form where \( X_t \) depends on lags of factors explicitly and the static form of the DFM where \( X_t \) depends on lags of factors implicitly.

There are both advantages and disadvantages of each form of the DFM depending on what the use of the model is. Dynamic factor analysis belong to the larger class of methods called hidden Markov models where observable variables are expressed in the form of hidden or unobserved variables. These properties that a small number of factors can explain changes in big data sets over time are what makes dynamic factor analysis well fitted for analyzing macroeconomic data sets. (Stock, 2016)

The dynamic factor model can be expressed as:

\[
X_t = \lambda(L) f_t + e_t \quad (1)
\]

\[
f_t = \Psi(L) f_{t-1} + \eta_t \quad (2)
\]

where \( e_t \) and \( \eta_t \) are idiosyncratic disturbances, \( X_t \) is a vector of time series variables, \( f_t \) is a matrix of latent factors and \( \lambda(L) \) is a matrix where each row represent the loadings for each series in \( X_t \). The lag operator is defined as:

\[
\alpha(L) X_t = \sum_{i=0}^{\infty} \alpha_i X_{t-i} \quad (3)
\]

where \( \alpha_i \) is a matrix at the \( i \)th lag and \( X_{t-i} \) is the time series vector of values known at time \( t-i \).

Rewriting Equation (1) and (2) into a static form of the DFM that depend on static factors \( F_t \) instead of dynamic factors \( f_t \) makes it susceptible to principal component analysis. The dynamic factor analysis model in Equation (1) and (2) can be rewritten as:

\[
X_t = \Lambda F_t + e_t \quad (4)
\]

\[
F_t = \Phi(L) F_{t-1} + G \eta_t \quad (5)
\]

where \( G = [I_q \ 0_{q(x-r-q)}] \), \( p \) is the degree of the lag polynomial matrix, \( F_t \) is a vector of static factor \( F_t = (f_t', f_{t-1}', ..., f_{t-p}') \), \( \lambda_h \) is the matrix of coefficients on the \( h \)th lag in \( \lambda(L) \) in \( \Lambda = (\lambda_0, \lambda_1, ..., \lambda_p) \), let \( \Phi(L) \) be the matrix of 1’s, 0’s
and values from $\Psi(L)$ so that the vector autoregression in Equation (2) could be written in terms of $F_t$. The value $r$ is the number of static factors $F_t$ and the value $q$ is the number of dynamic factors $f_t$.

2.1.1 Method of principal components

The method of principal components is a linear orthogonal transformation of the data set. This creates a set of principal components in which there is no covariance and hence no correlation within the components. Minimizing the variance, $V_r$, within the data set the least square problem in Equation (7) can be solved with principal components.

$$\min_{F_1, \ldots, F_T} V_r(\Lambda, F)$$

where

$$V_r(\Lambda, F) = \frac{1}{NT} \sum_{t=1}^{T} (X_t - \Lambda F_t)'(X_t - \Lambda F_t)$$

The solution to Equation (7) is a least squares problem and can be estimated by $\hat{F}_t = \frac{1}{T} \hat{\Lambda}'X_t$. The eigenvectors of the sample variance matrix of $X_t$ is $\hat{\Lambda}$ that belongs to the largest eigenvalues of $\hat{\Sigma}_x$. The sample variance matrix can be estimated from $\hat{\Sigma}_x = \frac{1}{T} \sum_{t=1}^{T} X_t X_t'$. (Stock, 2016)

2.1.2 Example of factor analysis

Constructing an example to better understand factor analysis is done in Table (1). This example consists of six variables and two factors with related factor loadings.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Factor 1</th>
<th>Factor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Income</td>
<td>0.65</td>
<td>0.11</td>
</tr>
<tr>
<td>Education</td>
<td>0.59</td>
<td>0.25</td>
</tr>
<tr>
<td>Occupation</td>
<td>0.48</td>
<td>0.19</td>
</tr>
<tr>
<td>House value</td>
<td>0.38</td>
<td>0.60</td>
</tr>
<tr>
<td>Number of parks in neighbourhood</td>
<td>0.13</td>
<td>0.57</td>
</tr>
<tr>
<td>Number of crimes each year in neighbour</td>
<td>0.23</td>
<td>0.55</td>
</tr>
</tbody>
</table>

In Table (1) the results from the factor analysis can be viewed. Factor 1 seems to be related to income and education since it load the most on variables 1-4 while it does not load on the area specific variables that one can argue has a low correlation with education. The second factor, factor 2, seem to be an area related factor. This factor loads the most on variables 4-6 which all have relations to in which area the house stand.
2.2 Random forests

2.2.1 Bootstrap sampling

Suppose a model should be fit to a training set of data which is denoted by $Z = (z_1, z_2, ..., z_N)$ where $z_i = (x_i, y_i)$. Randomly draw $B$ new data sets with replacement from $Z$ where each set has the same size as $Z$. The model is then fit to each of these $B$ bootstrap data sets also known as bootstrap samples.

2.2.2 Bootstrap aggregation

Bootstrap aggregation can be used to improve an estimate or prediction by fitting the model to each bootstrap sample, $b = 1, 2, ..., B$ resulting in a prediction $\hat{f}^{*b}(x)$. Then the bootstrap aggregation estimate is defined by:

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x)$$

(8)

Hence the bagging estimate is an average over all model fits with every bootstrap sample. Equation (8) is approaching the true bagging estimate as $B \to \infty$.

2.2.3 Regression tree

Consider a data set of $p$ inputs, $x_{ip}$, and a response variable, $y_i$, each with number of observations equal to $N$. Dividing the response variable into $M$ regions $R_1, R_2, ..., R_M$ and in each region seeing the response variable as a constant $c_m$. This can be represented by:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

(9)

Using the minimization of sum of square error between $f(x)$ and the response $y_i$ it can be observed that the best constant $\hat{c}_m$ to choose for each interval is a simple average of the response in each region:

$$\hat{c}_m = \text{average}(y_i | x_i \in R_m)$$

(10)

However finding each binary partition in terms of minimum sum of squares is generally computationally infeasible. Because of this an approach with a greedy algorithm is used instead. From all the data a splitting variable $j$ and a split point $s$ should be found. This creates the half-planes:

$$R_1(j, s) = \{X | X_j \leq s\}$$

(11)

$$R_2(j, s) = \{X | X_j > s\}$$

(12)

Again using minimization of sum of square error to find the best $j$ and $s$ the following is obtained:
\[ \min_{j,s} \left[ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \]  

and then the inner minimization is solved by:

\[
\hat{c}_1 = \text{average}(y_i | x_i \in R_1(j,s)) 
\]

\[
\hat{c}_2 = \text{average}(y_i | x_i \in R_2(j,s)) 
\]

Scanning through all input variables the determination of the best pair of splitting point and splitting variable \((j,s)\) becomes feasible. This process can then be repeated for all resulting regions to grow a regression tree.

### 2.2.4 Definition of random forest

The idea behind random forests is to use bootstrap aggregation to reduce the variance of many noisy decision trees. Decision trees are perfect to use since they can capture complex interactions within the data while the noise is kept at a low level by the averaging process. The regression algorithm for random forests can be defined as:

1. For \(b = 1\) to \(B\):
   
   (a) Take a bootstrap sample \(Z^*\) from the training data.
   
   (b) Create a random forest of decision trees to the bootstrapped data set by repeating the following steps until the minimum node size is reached.
      
      i. Select \(m\) variables at random from all variables.
      
      ii. Find the best split-point in the variables \(m\).
      
      iii. Divide the node into two new nodes.

2. \(\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)\)

where \(2\) is used to find the regression from the random forest to regress a new value from the model. \((\text{Hastie, 2008})\)

### 2.2.5 Example of random forest

Suppose that there has been a random selection of two variables and some of the data that should be used to create a decision tree. This selection can be viewed in Figure (1).
Figure 1: Plot of two variables and the three classes of data namely green samples, blue samples and red samples which should be combined into a decision tree.

In Figure (2) the decision tree that is created from the data set in Figure (1) can be seen. Each split condition is also written above each node that is split into two new nodes. In this case the data set can be classified in such a way that every leaf only has one sample type. If a new value should be predicted it should first be controlled if the X-value is smaller or larger than 7.0 then depending on the result it should be checked if the Y-value is larger than 2.5 or 4.0 which will create a predicted type for the new sample. In a random forest regression this prediction is done with several different decision trees and the result is averaged from all the predictions this is the main idea of the random forest method.
2.3 Support vector machine

Support vector machine is a machine learning method that separates a data set with the construction of linear boundaries in a transformed version of the feature space.

According to the conditions in Figure (3) a criterion can be computed for positive respectively negative samples as follows:

\[ \omega \cdot u + b \geq 0 \quad (16) \]

where \( b \) is a constant. If this is true for a sample then it is a positive sample. This is the rule that decides which values are positive samples and which values are negative samples.

By introducing a new variable \( y_i \) such that \( y_i = 1 \) for positive samples and \( y_i = -1 \) for negative samples.

Equation (16) can then be rewritten as

\[ y_i (\omega \cdot x_i + b) - 1 = 0 \quad (17) \]

this is true for every sample \( x_i \) on the margin. By taking the difference between a positive and a negative sample on each side of the margin and then multiplying by a vector that is perpendicular to the margin the width of the margin can be expressed as the following:

\[ \text{Width} = (x_+ - x_-) \cdot \frac{\omega}{||\omega||} \quad (18) \]
by rearranging Equation (17) it can be deduced that $x_+$ and $x_-$ gives $1 - b$ and $1 + b$. Maximizing the width of the distance between the margins can then be rewritten as minimizing the following:

$$
\min \frac{1}{2} \left\| \omega \right\|^2
$$

(19)

Now the function should be minimized with constraints which can be solved by Lagrange multiplier method if written as:

$$
L = \frac{1}{2} \left\| \omega \right\|^2 - \sum \alpha_i \left[ y_i (\omega \cdot x_i + b) - 1 \right]
$$

(20)

solving this problem gives the following two expressions:

$$
\omega = \sum_i a_i y_i x_i
$$

(21)

$$
\sum_i a_i y_i = 0
$$

(22)

Inserting this into Equation (20) the final expression is obtained:

$$
L = \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j x_i \cdot x_j
$$

(23)

This expression can then be numerically analyzed to find the maximum value which is the solution to the problem.
2.3.1 Kernel functions

The problem that arises when an optimal solution cannot be found with linear vectors can be overcome by the introduction of kernel functions. These functions divides samples into arbitrary shaped groups instead of a linear divide. This is done by mapping the data set into another space where a linear division is possible. (Hastie, 2008)

2.3.2 Support vector machine example

Using the data set from Figure (3) as an example it can be observed how support vector machine divides data sets. In the case shown in the figure there are two variables and two different types of data in the data set namely positive and negative samples. Support vector machine divides these samples by two parallel margins that has a maximized distance between them without containing any sample in between. Predicting new samples in this case is just a matter of controlling if the two variable values are above of below the margins.

2.4 System averaging model

Suppose that there are a lot of estimations done by different prediction methods and all these should be weighted and summarized into a single prediction. This can be done by using a system averaging model or SAM which weight each model individually based on historical performance compared to other prediction models. The weights for each model can be derived by the following expression:

\[ \omega_{i, \tau, h} = \frac{1}{\sum_{i=1}^{N} \frac{1}{MSE_{i, \tau, h}}} \]  

(24)

where \( N \) is the number of models to combine, \( \tau \) is the time period at which to produce the forecast, \( i \) is the used model and \( h \) is the prediction horizon.

MSE or mean squared error is a commonly used measure of accuracy for a predicted mathematical model. It results in a single number and positive and negative deviations of the same size between samples and model have the same effect on MSE. (Aastveit, 2011) Mean squared error is defined as:

\[ MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 \]  

(25)

where \( Y_i \) is the true values to a time series and \( \hat{Y}_i \) is the predicted values to the time series. This exponent that in this standard case is equal to 2 will be varied in the results section and is known as SAM-exponent from here on. Another approach is to weigh resent data higher than past data since macroeconomic variance driving parameters tend to change slightly over time. Instead of using a mean error model an exponentially weighted error model could be used to make recent predictions more valuable. The weights for each model can be derived by the following:
\[ \omega_{i, \tau, h} = \frac{EWMA_{i, \tau, h}}{\sum_{i=1}^{N} EWMA_{i, \tau, h}} \]  

(26)

where the variables are defined above. EWMA or exponentially weighted moving average error is an error function that is appropriate for macroeconomic data since it weights more recent errors higher than past errors.

\[ EWMA = \frac{(1 - \lambda)}{\lambda(1 - \lambda^n)} \sum_{i=1}^{n} \lambda^i (\hat{Y}_i - Y_i)^2 \]  

(27)

where \( \lambda \) is the weight that decides how much more recent data should be weighted compared to past data. This weight is set between zero and one. Setting \( \lambda \) to a larger value that is close to one weights recent errors significantly higher than past errors.

### 3 Method

In this project Matlab is used for all of the coding. The data is imported to Matlab from Macrobond which is a service that provides macroeconomic data globally. Some of the data series that are imported to Matlab has to be season adjusted due to seasonal patterns in the data sets. This is done by season adjusting with ARIMA X11 seasonal adjusting. Each prediction is done at the end of a quarter to match up with GDP values from that quarter. The problem that arises is that during a quarter there are a lot of data missing until the end of the quarter. This is solved by using AR(1) on the data series of macroeconomic data to forecast the data set to the end of the quarter. When the dynamic factor model has found the principal component factors for a data set they are combined with a linear regression between the factors and the GDP values that are observed previously. The benchmark index that is used as a comparison for the result is a simple random walk prediction. This means that the predicted GDP growth in a period is the same as the period before. In Algorithm (1) a simplified version of how the program operates can be found.

**Algorithm 1** Simplified procedure of program

1. Import data from Macrobond based on excel file
2. Season adjust data using ARIMA X11
3. Extend each data series to end of quarter with AR(1)
4. Every individual model creates its prediction
5. All predictions are combined using SAM
3.1 Data

Different number of variables are used for different countries due to the available data sets for each individual country. Each variable has to be at least 56 quarters of continuous data to be included in the prediction data set. There is a span of about 200-550 number of variables used for different countries. Some of the variables that are available has to be discarded due to the short length of the time series while other variables are discarded due to not having continuous data series. The variables do have different release intervals but the data is added up to form quarterly data. Some of the data series has to be differentiated to become stationary which is a requirement for the prediction to work properly. In Table (2) a brief summary of the most important statistics for the variables which have the highest correlation with GDP can be found. The data set is from Sweden and this typical behaviour with high correlation between macroeconomic variables can be observed.

Table 2: Brief summary of statistics from Swedish data set for the variables with strongest correlation to GDP.

<table>
<thead>
<tr>
<th></th>
<th>Sweden Equity Indices NasdaqOMX Industrial Goods</th>
<th>Sweden Production Approach Producers of Services</th>
<th>Sweden Expenditure Approach Import Goods Total</th>
<th>Sweden Business Approach Production Index Service Production</th>
<th>Sweden Domestic Trade Service Production Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>752</td>
<td>$7.28 \times 10^{10}$</td>
<td>$2.48 \times 10^{14}$</td>
<td>101</td>
<td>101</td>
</tr>
<tr>
<td>Standard dev.</td>
<td>300</td>
<td>$9.10 \times 10^{9}$</td>
<td>$2.77 \times 10^{10}$</td>
<td>8.19</td>
<td>14.3</td>
</tr>
<tr>
<td>Variance</td>
<td>$9.00 \times 10^5$</td>
<td>$8.28 \times 10^{19}$</td>
<td>$7.67 \times 10^{20}$</td>
<td>67.0</td>
<td>206</td>
</tr>
<tr>
<td>Correlation with GDP</td>
<td>0.978</td>
<td>0.977</td>
<td>0.972</td>
<td>0.971</td>
<td>0.966</td>
</tr>
</tbody>
</table>
4 Result

In this section four data sets will be reviewed. They are from Sweden and the United states of America at two different time periods. There was access to data sets from seven different countries but due to restrictions in time the above mentioned countries were the main focus of this project. Sweden and USA also had nice data sets with many variables and long time series. The first data set is from October 27th when the GDP values for the 3rd quarter is not yet released. This automatically creates two predictions from the model, one for the 3rd quarter and one for the 4th quarter. The second data set that is analyzed was imported on December 8th and here there is only one prediction which is the 4th quarter since 3rd quarter data at the time has been published.

4.1 Sweden Data October 27

4.1.1 Dynamic factor analysis

Figure 4: Data set from Sweden 27 October 2016. System averaging model used for DFM models with different number of principal components. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.
Using only a set of dynamic factor models that are combined by a system averaging model the results can be seen in Figure (4). Compared to mean square error for a random walk prediction of GDP growth the dynamic factor models has lower error in both previous quarter prediction and current quarter prediction. It can also be observed how DFM in most single predictions have a lower error than random walk. The previous quarter error is also lower than current quarter error which is reasonable since more data has been released for the 3rd quarter. The variance is lower in the dynamic factor models than in the real GDP growth and it misses some of the sharp changes in GDP growth. The system averaging model parameters $\lambda$ and SAM-exponent is based on the parameters that gave the smallest error in the prediction.

### 4.1.2 Machine learning

![Figure 5: Data set from Sweden 27 October 2016. System averaging model used for machine learning block which is a SAM that combines random forests and support vector machine. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.](image)

The machine learning block where random forests and support vector machine are combined to form a prediction with a system averaging model results can be viewed in Figure (5). Again comparing the error of the machine learning
methods to a random walk predictor the machine learning method has lower mean square error and is better in most periods. The machine learning method has a low variance and misses some sharp changes in GDP growth.

Figure 6: Bar plot of the weights distribution between the two Machine learning models. They are random forests and support vector machine.

Comparing between the two machine learning models both are fairly equally weighted and the plot does not seem too uneven with only a couple of jagged weights. This strengthens the importance of both machine learning models being equally useful for the prediction.
4.1.3 Error surface with VAR

Figure 7: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests, support vector machine and vector autoregressive model.

System averaging model has two important parameters that can be varied to obtain different predictions. These are $\lambda$ and SAM-exponent and the result can be seen in Figure (7). It can be observed in the figure that the error is smallest when using a high value of $\lambda$ and a small value of SAM-exponent. The largest errors are received when a high value of $\lambda$ and a high SAM-exponent is used. The smallest error has a value of about 85 percent of the largest ones.
4.1.4 Error surface without VAR

Figure 8: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests and support vector machine.

Removing the vector autoregressive model from the prediction the errors still behaves in a similar way. High values of $\lambda$ and small values of SAM-exponent still gives the smallest error values. However it can be noticed that all error values are smaller for the same combinations of $\lambda$ and SAM-exponent when the vector autoregressive model is left out of the prediction. Again in this case the smallest error is about 85 percent of the largest ones.
4.1.5 Combining all models with SAM

Figure 9: System averaging model with dynamic factor analysis, random forests, support vector machine and vector autoregressive model. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.

Studying the total system averaging model prediction the estimation has lower variance than the actual GDP growth. The error for the third quarter is lower than random walk in most periods and the mean square error for the 3rd period is lower as well. Fourth quarter error is also lower than random walk error in most time periods and mean square error is lower than random walk MSE.
Figure 10: System averaging model with dynamic factor analysis, random forests and support vector machine. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.

Removing the vector autoregressive model and creating a system averaging model prediction the problem with low variability still remains in the prediction. However the mean square error for both 3rd and 4th quarter prediction is smaller without the vector autoregressive model.

When machine learning is set as a block the weight distribution between machine learning and random forests are quite equal which can be seen in Figure (11). Though it can be observed that one method have a tendency to be better than the other which results in a slightly jagged weight distribution.
4.2 Sweden Data December 08

4.2.1 Dynamic factor analysis

The first results from this data set is again using only various versions of dynamic factor models which are combined by a system averaging model which can be seen in Figure (12). System averaging model for the dynamic factor models prediction is still better than random walk when comparing mean square error. Looking at the mean square errors in the bottom plot there is small difference between the comparison of the errors. Some periods the SAM has lower error and some periods the random walk has lower error. The system averaging model parameters $\lambda$ and SAM-exponent are chosen from the prediction with the smallest error.

Figure 11: Bar plot of the weights distribution between dynamic factor model and Machine learning in the system averaging model.
4.2.2 Machine learning

Combining both machine learning models, random forests and support vector machine, into a single prediction the result can be viewed in Figure (13). The machine learning prediction has barely lower values than the random walk benchmark. It can also be observed how the machine learning prediction has a low variance and appears to miss most major movements in GDP growth.
Figure 13: Data set from Sweden 08 December 2016. System averaging model used for machine learning block which is a SAM with random forests and support vector machine. First prediction is for 4th quarter. In the bottom there is a mean square error plot for the 4th quarter prediction.

In Figure (14) it can be seen how the weights are distributed over both the machine learning models for this second data set. The distribution is fairly equal however the graph is a bit jagged which means that the distribution of the weight shifts between the two models rather fast and even between two adjacent quarters. Comparing this to the first data set and Figure (6) the more recent quarterly errors are weighted higher in this case because of the lower value of $\lambda$. 
4.2.3 Error surface with VAR

System averaging model parameters are varied to obtain a low prediction error. The surface plot in Figure (15) shows this. The error is smallest for low values of $\lambda$ and a small SAM-exponent. This means that the most recent values have high weights in the prediction and no model is punished particularly hard for high error predictions.
Figure 15: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests, support vector machine and vector autoregressive model.

4.2.4 Error surface without VAR

Similar plot arises when the vector autoregressive model is removed which can be seen in Figure (16). Smallest errors can be obtained with low values of $\lambda$ and low values of SAM-exponent. The size of the errors are of about the same magnitude as before the vector autoregressive model was removed. An interesting feature to notice in Figure (15) and 16 is that the smallest errors are obtained when weighting the most recent data heavily which will be discussed later on.
Figure 16: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests and support vector machine.

4.2.5 Combining all model with SAM

For the second data set that was analyzed in the results the total system averaging prediction can be viewed in Figure (17). As mentioned in this case there is only a single prediction being made since the third quarter GDP values has been released by the time of obtaining the data. There is a close to even distribution of lowest error between random walk and total system averaging model however in total the mean square error is smaller for SAM.
Figure 17: System averaging model with dynamic factor analysis, random forests, support vector machine and vector autoregressive model. First prediction is for 4th quarter. Bottom plot is 4th quarter prediction error and mean square error.

In Figure (18) the prediction is done without the vector autoregressive method. In the majority of the predictions the error is smaller for the SAM-prediction than the random walk error. The mean square error is also lower than random walk and comparing to the prediction with the VAR model in Figure (17) the mean square error is smaller in this case without the use of VAR however there is still too low variability in the prediction.
Figure 18: System averaging model with dynamic factor analysis, random forests and support vector machine. First prediction is for the 4th quarter. In the bottom there is an error plot for the 4th quarter prediction.

The comparison of machine learning and dynamic factor model is shown in Figure (19) where the total weight distribution is about equal. This jagged pattern is very clear in this figure as one method dominates the other between consecutive periods. This is a consequence of a very small $\lambda$ and a larger SAM-exponent which makes the model quite unstable between adjacent periods and the weights could shift very quickly. The parameters that were chosen for the prediction gave the lowest error however it can be discussed how stable the method is for future predictions.
4.3 United states Data October 27

Studying the first American data set when only using a system averaging model to combine dynamic factor models the results can be viewed in Figure (20). The first period prediction finds the 2009 bottom quite well and seems acceptable overall however the second period prediction is not quite as well fit. In the bottom left plot it can be observed that the error in most periods are smaller than random walk error and the mean square error for the prediction is about 30 percent lower than random walk error. System averaging model parameters $\lambda$ and SAM-exponent are in this data set chosen as $\lambda$ equals to 0.1 and SAM-exponent equal to 3.5.
Figure 20: Data set from USA 27 October 2016. System averaging model used for DFM models with different number of principal components. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.

4.3.1 Machine learning

The machine learning methods, random forest and support vector machine, combined with a system averaging model results can be seen in Figure (21). Again the problem that can be seen in the figure is that machine learning has too low variance. Compared to random walk the error is smaller in most periods for the machine learning prediction. Mean square error for machine learning is about 15 percent lower than random walk error.
Figure 21: Data set from Sweden 08 December 2016. System averaging model used for machine learning which is a SAM with random forests and support vector machine. First prediction is for 4th quarter. In the bottom there is an error plot for the 4th quarter prediction.

A bar plot over the weight distribution between the two machine learning models can be seen in Figure (22). When $\lambda$ is chosen as a lower value and SAM-exponent is larger this jagged pattern again appears where the weight shifts quite intensely between consecutive periods. Again the question is how stable the model would be for future predictions with such a small value of $\lambda$. 
4.3.2 Error surface with VAR

Varying the system averaging model parameters the errors can be seen in the surface plot in Figure (23). The smallest values for the error can be seen when $\lambda$ is equal to 0.1 and SAM-exponent is equal to 3.5. Smallest errors are about 85 percent of the largest errors in the figure however there is a very small difference between errors for all values of $\lambda$ when SAM-exponent is low for future predictions this could be an argument of choosing a higher value of $\lambda$ to get a more stable model.
4.3.3 Error surface without VAR

Removing the vector autoregressive model and again plotting the error as a surface plot the results can be seen in Figure (24). In this case the smallest error can be obtained when $\lambda$ is 0.1 and SAM-exponent is 1.5. Removing the vector autoregressive model lowers the error values which also can be observed in the figure. Again as in the previous figure the difference between errors for all values of $\lambda$ when SAM-exponent is small is very low.
4.3.4 Combining all models with SAM

In Figure (25) the total system averaging model prediction for all models can be observed. In almost every period SAM prediction error is lower than random walk error and mean square error for first period prediction is about 20 percent smaller than random walk. Looking in the bottom right plot the second period prediction error is still lower than random walk.

Figure 24: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests and support vector machine.
Figure 25: System averaging model with dynamic factor analysis, random forests, support vector machine and vector autoregressive model. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.

Removing the vector autoregressive model and plotting the results again can be seen in Figure (26). In the upper plot the prediction can be seen for both one and two period predictions. In the lower left plot the error is lower than random walk in most periods and the mean square error is about 25 percent lower than random walk error. In the last plot the second period prediction error can be seen which is in major lower than random walk error and mean square error is also here smaller than random walk mean square error.
Figure 26: System averaging model with dynamic factor analysis, random forests and support vector machine. First prediction is for 3rd quarter and second prediction is for 4th quarter. Left down plot is 3rd quarter prediction mean square error and right down plot is 4th quarter prediction mean square error.

The results from the distribution of weight from the SAM between both machine learning methods and dynamic factor analysis can be seen in Figure (27). The distribution between the two groups in total is fairly equal however one tends to dominate the other in some periods. In opposite to earlier data set this prediction uses SAM-exponent of 1.5 and and a λ of 0.1 which still produces this very jagged pattern between the weights.
Figure 27: Bar plot of the weights distribution between dynamic factor model and Machine learning in the system averaging model.

4.4 United states Data December 08

4.4.1 Dynamic factor analysis

Using only different versions of a dynamic factor model for the fourth data set that is calculated by a system averaging model the results can be viewed in Figure (28). The SAM parameters are chosen to get the smallest possible error which results in $\lambda$ equal to NaN which means that all data is weighted equally, it is basically a simple moving average model. SAM-exponent is chosen as 1.5 which means that no model is punished particularly hard for large error values. Looking at the bottom plot the system averaging model prediction error is smaller than random walk error in the majority of time periods. The mean square error is about 25 percent smaller than random walk mean square error.
Figure 28: Data set from USA 08 December 2016. System averaging model used for DFM models with different number of principal components. First prediction is for 4th quarter and this is the only prediction since 3rd quarter GDP values has been published. The bottom plot is prediction error and mean square error for the 4th quarter prediction.

4.4.2 Machine learning

The machine learning prediction shows in Figure (29) the prediction based on a system averaging model from both machine learning methods. In the lower plot it can be observed that machine learning still has smaller error in most periods compared to random walk. The mean square error is almost 20 percent lower than random walk mean square error. The major problem here as in some of the previous cases is the low variance in the machine learning prediction.
Figure 29: Data set from USA 08 December 2016. System averaging model used for machine learning which is a SAM with random forests and support vector machine. First prediction is for 4th quarter. In the bottom there is an error plot for the 4th quarter prediction.

The weight distribution between both machine learning models can be seen in Figure (30). In general the distribution is fairly even between both models but again this jagged pattern that was observed previously appears. This is the first time this pattern appears when a simple moving average model is used which normally has more gentle transition of weights.
4.4.3 Error surface with VAR

Surface plot of the variation in system averaging model parameters can be seen in Figure (31). The error is smallest for high values of $\lambda$ and small values of SAM-exponent which is an equal weighting of the data. In this case the difference between the largest and smallest error is the smallest where the smallest error is just over 90 percent of the largest error.
4.4.4 Error surface without VAR

Removing the vector autoregressive model creates a similar error surface however the errors in this case are lower which can be seen in Figure (32). In this case the smallest error can be found when $\lambda$ has a value close to one and SAM-exponent also has a value close to one. Even though the largest error is about the same as with the vector autoregressive model the smallest errors are a couple percent lower than with the VAR.
Figure 32: Surface plot of prediction errors for different values of SAM-exponent and $\lambda$ where NaN is a simple moving average model. Included models are dynamic factor model, random forests and support vector machine.

4.4.5 Combining all models with SAM

For the December 8 data set the final SAM prediction can be seen in Figure (33). In the lower plot it can be observed how first period prediction error is smaller in almost every single period compared to random walk error. The mean square error is thereby about 20 percent smaller than random walk MSE.
Figure 33: System averaging model with dynamic factor analysis, random forests, support vector machine and vector autoregressive model. First prediction is for 4th quarter. Bottom plot is 4th quarter prediction mean square error.

In the plot where the vector autoregressive model is removed and the prediction is created again showed in Figure (34). Still the SAM prediction has lower error in nearly all periods and the mean square error is almost 25 lower than random walk mean square error. The mean square error is slightly higher in this case when comparing to the use of only a dynamic factor model which can be seen in Figure (28). This difference could be discussed as being small since it is about a 3 percent increase in error when using SAM compared to only using DFM.
Figure 34: System averaging model with dynamic factor analysis, random forests and support vector machine. First prediction is for the 4th quarter. In the bottom there is a mean square error plot for the 4th quarter prediction.

A bar plot that shows the distribution of weights between the machine learning block and the dynamic factor model can be observed in Figure (35). In total the distribution is again almost equal and some single periods are still heavily dominated by a method. Again this is a little bit of a surprising pattern since a low SAM-exponent is used in combination with a simple averaging model to predict GDP growth which normally gives a smoother shift of weights between adjacent quarters.
5 Discussion

5.1 Data

Data sets that are from the same country can still create very different results. This can be seen when comparing Figure (7) and (15) where in the first case a small as possible $\lambda$ obtains the smallest error and in the second case a large as possible $\lambda$ creates the smallest possible error. Studying differences between countries the results here are also quite unique. Looking at Figure (7) and (23) the errors for Sweden and the United stats has substantial differences even when the data sets are from the same time the error plots that are formed are completely different. There are four data sets examined in this study however it is hard to draw conclusions for other data set from this project due to the difference in behavior of the methods depending on the underlying data sets. In total there is 56 quarters of data used for each of all the time series. This is 14 years of data however since GDP data is only released once every quarter there is only 4 observations every year. This will result in most of the data sets having more variables than observations. Since there are two system averaging models run in two layers where each uses 10 quarters of data there is only 36 predictions being made which might be argued is a rather small amount to draw any deep
conclusions from. Since the examined methods are quite dependent from the data sets the predictions are not done for any longer period than one quarter into the future since the performance of these far into the future predictions tend to be guesses of the behaviour of random processes.

For some of the countries a problem arises where some of the macroeconomic variables are removed due to too short time series of data. This removed data could contain important variance that could boost the performance of the final prediction significantly. There might be methods to still use these short time series however if they are too short they have been removed altogether in this project. In some cases a large set of variables are removed due to short time series of macroeconomic data. This in combination with macroeconomic differences between countries will create very different data sets with irregular sizes for countries.

Some of the variables have been differentiated to make the time series stationary and more fit to use as independent variables for the prediction. These choices of which variables to differentiate has been made from macro economical theory. In the dynamic factor model after the dynamic factors are calculated they are combined by a linear regression which is a quite simple method that could be replaced or optimized to fit the data even better by using macroeconomic theory as an example.

5.2 Exponentially weighted moving average

An implementation that was not made in previous work was the exponentially weighted moving average method instead of a simple moving average method. The hopes were to create a prediction model that could respond faster to changes in new important variables and to avoid the so called ghost effect where old data that falls out from the moving window has a significant effect on the next prediction. However when \( \lambda \) gets to small all the weight in the prediction is put on a handful of values which could make the prediction very unstable. This phenomena can be seen in Figure (11) where \( \lambda \) is chosen as 0.01 since it gave the smallest error value. The weights could shift back and forth in consecutive time periods which is not so positive since the weights are calculated on historical values and a change in these recent errors for the models affects the final prediction in a strong sense.

5.2.1 Problem of instability

This problem can be seen in some of the bar plots are a jagged pattern where for a single period one model is very heavily weighted. This is most common when \( \lambda \) is small which means a few values in the most recent data has a powerful influence over the final prediction. The problem can also arise from a high SAM-exponent since it punishes a slightly worse prediction quite hard. When a method gets a low error in a prediction this method can get very heavily weighted in the next period which is very unstable to have a single method mostly create the entire final prediction by itself. Instead maybe an alternative weighting
system could be implemented where a single method could not dominate in such a way. In Figure (15) and (16) it can be observed how small the difference in error could be between high and low values of $\lambda$. Then it might not always be the best for future predictions to put all weight on a model that performed well in some few instances.

5.3 Random walk as benchmark

The benchmark that is used in this report to compare performance of the predictions is a simple random walk. This is a very trivial method and maybe there should be a slightly better and more complex method as benchmark since random walk tends to perform quite bad in volatile periods where GDP growth are varying a lot.

There are other indexes that are used when speaking of GDP growth such as California activity index. However comparing to this index is not a fair comparison since the standardization is made afterwards and previous values are updated as new values are added. This makes CAI less like a prediction and more like a hindsight index.

5.4 Machine learning

For some of the machine learning plots a very low variance in the predictions can be observed. This could be both due to a problem with the weighting algorithm or due to a problem with the machine learning methods for some of the data sets where they might not be as fit to use. There are many other machine learning methods to use for a prediction however these were chosen due to that they are normal machine learning methods that often are used with good results.

5.5 Parameter optimization

In the results section the model was chosen from the prediction with the smallest error depending of the value of SAM-exponent and $\lambda$ however each of the methods have underlying parameters that also could be tuned for optimization of the method. The problem here is that optimizing so many underlying parameters in a similar way as $\lambda$ and SAM-exponent would make a run of the program take more than a week. This optimization could probably boost the performance of the program significantly however it has to be done in a more efficient way.

5.6 Swedbank use of program

The program is written in such a way that it should be able to run without errors by just updating the excel file containing which data series that should be used in the prediction. As mentioned previously if random walk is the benchmark to beat this model does it in almost every single prediction however optimizing this model could improve the results even further in my opinion.
References


