APE Master Program

Forecasting the French GDP:

Essay on statistical models to forecast aggregate macroeconomic variables

August 24, 2015

Nicolas Saleille
PSE, ENS Cachan and ENSAE
nicolas.saleille@ensae.fr

Supervisor :

Stéphane Gregoir
Directeur, DMCSI, INSEE
stephane.gregoir@insee.fr

Acknowledgments

I am extremely grateful to my advisor Mr. Stéphane Gregoir, head of the Methodology, Statistical Coordination and International Relations Directorate at INSEE, for his support, guidance, valuable comments and suggestions throughout the year 2014/2015. His technical expertise and sharp advices helped me in all the time of research and writing of this master thesis.

Contents

Introduction

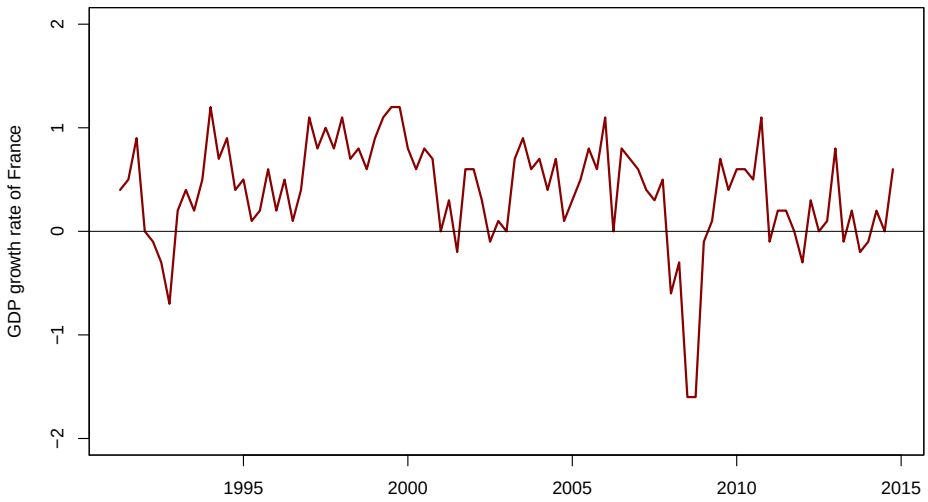
Macroeconomic forecasting methods evolve quickly and constitute an active field of research. Researchers now have a lot of data at their disposal, and increasing computational capacities makes it possible to consider estimation techniques far more complex than only a few years ago. In this context, statistical institutions regularly investigate options to produce accurate forecasts of economic indicators such as the Gross Domestic Product (GDP) or inflation rates. Hence, the main motivations of this master thesis are to study, provide intuitions and compare the performances of existing methods to forecast these macroeconomic variables.

In recent years, one of the main focuses of macroeconomic research has been Dynamic Stochastic General Equilibrium (DSGE) models. These micro-founded models typically involve a representative household and a representative firm with optimizing schemes. Solving the model analytically leads to a set of equations characterizing a steady-state. It is then possible to linearize the model around the steady-state, and to study the response of endogenous variables to various types of shocks. Today, complex DSGE models like the one of (Smets and Wouters, 2005) are used by central banks as powerful tools to conduct policy simulation, as well as medium to long term forecasts. Their main advantage is to integrate a high level of theory, in particular rational expectations of economic agents, imperfect market conditions such as monopolistic competition, and nominal rigidities. On the other hand, DSGE models are characterized by heavy assumptions regarding model structures, and their estimation often requires approximations. Thus, they do not always lead to satisfying parameter estimates in empirical applications. As an alternative to theoretically grounded models such as DSGEs, (Sims, 1980) suggested an approach to macroeconomic modeling based on vector autoregressive models (VARs). VARs are a class of model mixing the statistical analysis of correlations and the economic theory. Compared to DSGE models, they have a flexible functional form that allows better in-sample fit to the data, and provides in many cases interesting out-of-sample forecast performances. VARs now have been used for years to conduct both structural analysis and forecasts, and their efficiency has been demonstrated in many empirical studies. Research on VAR models and how to improve their short-term forecasting performance has been particularly active in the last decade, and a wide variety of modeling options are now available to the forecaster.

In this thesis, we explore several *short-term* forecasting methods, with forecasting horizons ranging from 1 to 8 month before the release of official figures by statistical institutions. We focus on the specific case of the quarterly French GDP growth rate. Forecasting the GDP is an important task for several French institutions involved in the construction of economic policies, including INSEE, DG Tresor, and Banque de France. Historically, many different methods have been used and combined. A popular one involves the construction of GDP forecasts from a combination of macro-sectorial previsions: the main components of the supply and demand sides are modeled separately, forecasts are conducted using historical data (mostly survey data), and combined using the relative weights of each sector in the national economy. Recently, other options such as Dynamic Factor Models have been investigated at Banque de France (Barhoumi et al., 2009) and at DG Tresor (Bessec and Doz, 2011). In this line of work, we focus on statistical models that can be used to nowcast and forecast the GDP efficiently. We discuss the pros and cons of various promising specifications, and compare their forecasting performances in a recursive exercise applied to the prediction of the French GDP growth rate. More specifically, our research investigates recent modeling options designed to handle two well-know issues: over-fitting in models with many predictors, and structural change.

In a forecast exercise, the macro-economist typically faces the *curse of dimensionality*. To forecast aggregate variables such as the GDP, a very large set of predictors might be relevant and enter the information set: one can think of various economic outlook surveys, real variables, nominal variables, as well as variables related to the international environment. On the other hand, the production of

Figure 1: Quarterly GDP growth rate for France - 1991Q1 to 2014Q4 - source: INSEE



macroeconomic data began in most developed countries after World War II, and most series are published at low frequencies (often monthly for indicators and quarterly for aggregates). As a result, the vast majority of macroeconomic datasets only include a few hundred observations, so that over-parametrization is an important concern in statistical models. Last but not least, macroeconomic predictors are in many cases highly correlated through time, as well as cross-sectionally, a feature that has to be taken into account to improve forecasting performances. Several methods have been proposed to overcome the curse of dimensionality, the most efficient ones being discussed in details in the following sections of this thesis. Dynamic Factor Models (DFMs) were initiated by (Geweke, 1977) and propose to shrink the information provided by a large set of predictors into fewer orthogonal factors estimated through a principal component analysis. More recently, (De Mol et al., 2008) and (Bańbura et al., 2010) set the theoretical foundations and proved the usefulness of Bayesian shrinkage to estimate large VARs.

When modeling phenomenons over long periods of time, parameters are likely to change substantially and this has to be taken into account by forecasters. For that reason, the forecasting literature made many efforts in order to incorporate the possibility for structural breaks in statistical models. To this regard, an interesting option is the Time-Varying Parameter VAR model (TVP-VAR). One other hot topic in modern statistics is to introduce uncertainty not only in the parameter space, but also in the model space (Varian, 2014). To this regard, an interesting method has been developed in (Koop and Korobilis, 2012) to perform dynamic model averaging (DMA). The authors rely on a flexible TVP-VAR specification that allows for gradual change in the coefficients, and dynamic variable selection based on the data. DMA has been proven to work well in several empirical cases, including the forecasting of real estate prices in the US, see (Bork and Møller, 2015).

The remaining sections are organized as follow. Section I presents the basics of econometrics applied to forecasting, while section II is dedicated to our empirical settings. Section III provides a detailed presentation of Dynamic Factor Models, and section IV shows how Bayesian shrinkage can be used as an alternative to principal component analysis to estimate large VARs. Sections V and VI present respectively TVP-VARs and DMA. Finally, we compare and discuss the performances of these different approaches to macroeconomic forecasting in an empirical application to the French GDP in section VII.

I An econometric approach of the forecasting problem

In this section, we present the statistical approach to macroeconomic forecasting and give insights on how one can use past data to build predictions. We also highlight the main characteristics of macroeconomic datasets, and we underline the main issues encountered by researchers in this field, namely short-series and variable selection.

I.1 Building economic predictions with statistical models

The problem of economic forecasting is *to use past and current information to generate a probability distribution for future events* (Litterman, 1986). This distribution is often called the *posterior predictive distribution* since it takes part of the past and current observed values of multiple economic variables and indicators. As of now, we will denote y_t the $(n \times 1)$ vector gathering the n-dimensional outcome we want to predict. Then the posterior predictive distribution is

$$\mathcal{P}\left(y_t | \Omega_{t-1}\right) \tag{1}$$

where Ω_{t-1} represents the information set at time $t-1$. In all of the following work, we will focus on how statistical models can help economists to derive it in the context they face.

Some econometric difficulties arise from the particular structure of macroeconomic datasets. Most time-series do not have more than a few hundreds observations, and degrees of freedom are typically scarce for the economist. Macroeconomic datasets have been produced and published by national institutes only for a few decades, and most indicators are released on a low frequency basis (each quarter, or, at best, each month). In France, quarterly national accounts were released on a quarterly basis since the beginning of the 50's. Hence, empirical macroeconomics must take specific care to ensure model parsimony.

Furthermore, multiple empirical analysis underlined the fact that economic structures are not stable but rather evolving through time. In practice, this means that some data might be too old to add any relevant predictive power regarding current economic variables. This is particularly true in countries that took part in the European Union and the Euro-zone, since macroeconomic conditions greatly changed since the beginning of the 90's. Thus, an other important feature of statistical models is the possibility to exploit efficiently the most recent data.

I.2 Forecasting with many predictors: the curse of dimensionality

In any econometric model, adding predictors is the most evident way to reduce the omission bias and to increase *in-sample* fit. However, additional predictors increase the dimension of the parameter that has to be estimated from the data and consumes additional degrees of freedom, so that the econometrician faces a *curse of dimensionality*. In a forecasting perspective, it is particularly important to use degrees of freedom carefully. For a given, limited number of observations, more parameters usually means a higher probability to *over-fit* and obtain poor *out-of-sample* performances. Hence, a particularly important feature in macroeconomic modeling is parsimony, and the ability to limit the dimension of parameters. Let's illustrate the curse of dimensionality starting from the reduced form of a VAR model:

$$Y = XA + \varepsilon \tag{2}$$

where $Y = (y_1', \ldots, y_T')'$ is a $(T \times n)$ matrix, y_t is a $(n \times 1)$ vector of n dependent variables at time t, and X is a $(T \times (np+1))$ matrix with rows containing a constant and the p lags of the n dependent variables at each date, i.e vectors $(1, y_{t-1}', \ldots, y_{t-p}')$. A is a matrix of coefficients and ε is a $(T \times n)$ matrix with t^{th} row given by $\varepsilon_t \sim \mathcal{N}(0, \Sigma)$. The parameters to be estimated are $\theta = vec(A)$, a vector with $n(np+1)$ components, and Σ, a matrix with $n(n+1)/2$ components. It is then straightforward to see that the dimension of the parameter space grows very fast in n, the number of predictors: setting for instance

$n = 100$ and $p = 4$ leads to more than 40 000 coefficients in θ, so that we need $T \approx 400$ to estimate the model from the nT observations. Such a configuration is not guaranteed, since a typical quarterly dataset will only have around 200 periods.

In the last years, the variety of economic time series available to forecasters has become incredibly huge. A typical illustration is the FRED database, that gather more than 250 000 time series. This means that economists have a very wide choice of predictors when conducting a forecasting exercise. Because of the curse of dimensionality, the most critical question from the forecaster's perspective lies in the selection of the most relevant ones among them. Traditionally, economists used to base their choice on economic theories to choose the variables best reflecting the macroeconomic environment. However in recent years, an important part of the literature as been dedicated to methods compatible with the inclusion of many predictors in forecasting models: (Bessec and Doz, 2014) consider 93 predictors for the French GDP, while (Bańbura et al., 2010) use a US macroeconomic dataset containing 131 variables, extended to 168 variables by (Koop, 2013). These methods are particularly interesting as they allow the econometrician to take advantage of the growing access to economic data. Handling many predictors can be achieved mainly in two different ways, namely dimensionality reduction and Bayesian shrinkage.

Dimensionality reduction happens when the researcher tries to compress the information set in an accurate way in order to save degrees of freedom. A first popular approach to dimensionality reduction is Principal Component Analysis (PCA). PCA uses an orthogonal transformation to convert a set of observations of correlated variables into a set of values of linearly uncorrelated variables, called the principal components. Replacing the initial predictors by the few first principal components, one can achieve great reductions in the dimension of the parameter space, while keeping the information most relevant to build forecasts. This approach has been first applied to dimensionality reduction in economics in (Geweke, 1977) and (Sargent et al., 1977). These papers led to the more general framework of dynamic factor models (DFM), presented in details in section III. These models have been applied to various empirical problems and provide on average good forecasting performances (Stock and Watson, 2011). Further extensions relates to dimensionality reductions are based on the least absolute shrinkage and selection operator (LASSO) applied to VAR models, see (De Mol et al., 2008) and (Gefang, 2014).

Bayesian shrinkage is an other popular approach to overcome the curse of dimensionality. It is based on the idea that parameters are random variables on which the researcher can set a prior distribution and use the data to estimate a posterior distribution. Specifying priors amounts to shrink the value of model coefficients to a set of values perceived as plausible: it reduces uncertainty. In times-series applications, the most famous shrinkage method is probably the Minnesota prior of (Litterman, 1986), that shrinks the coefficients of a VAR model either to one (if the data favors a high degree of persistence) or zero (if it does not). Using this methodology, it is possible to keep a large number of predictors in the VAR, while controlling for over-fitting. Adding predictors is made possible by strengthening the degree of shrinkage. Recently, some other Bayesian priors have been successfully tested on VARs including very large sets of predictors: see for instance (Bańbura et al., 2010) and (De Mol et al., 2008). A popular option has been the stochastic search variable selection (SSVS) of (George et al., 2008). An application to macroeconomic data can be found in (Koop, 2013). Generally speaking, numerous empirical work showed that Bayesian shrinkage forecasts well.

I.3 Structural change and model uncertainty

Finally, a growing literature introduced structural change and model uncertainty into macroeconomic modeling. As underlined before, macroeconomic conditions change through time and it is important to take this into account when estimating models on long periods of time. In many cases, a standard approach with fixed parameters will not allow the econometrician to model structural change. To this regard, an interesting framework has been developed under the name of Time-Varying Parameter VARs

(TVP-VARs). These models assume that model parameters can be modeled as random walk processes, so that they vary progressively through time. TVP-VARs estimation is based on a state-space formulation and the Kalman filter, where the parameter θ plays the role of the latent variable.

Finally, model uncertainty has been introduced by (Koop and Korobilis, 2012), in an interesting extension of TVP-VARs called Dynamic Model Averaging (DMA). Rather than considering a very large set of predictors, the main idea of DMA is to consider many models, each of them including a limited number of predictors. This approach allows to consider many models at once, and constitutes a good way to control for shifts in relevant predictors through time. Estimation is made feasible thanks to forgetting factor assumptions, a parametric assumptions that allows to avoid heavy MCMC procedures. Even if DMA is not initially suited to handle large set of predictors, some recent contributions proposed useful adaptations: see (Koop and Korobilis, 2013) and (Belmonte et al., 2014). TVP-VARs and DMA are presented in details respectively in sections V and VI.

II Forecasting the French GDP: the empirical setting

The problem under consideration in this thesis is the short-term forecasting of the French GDP growth rate. To this regard, comparing forecasting models is particularly important since many methodologies have arisen and began to be used by French institutions (such as INSEE, DG Tresor or the Banque de France), as well as international ones (the ECB).

As highlighted in the previous section, forecasting macroeconomic variables is a difficult problem from an econometric standpoint. Furthermore, when estimating and comparing model performances, a particular attention has to be dedicated to the empirical framework. In this section, we present the dataset used in our empirical application. We also put emphasis on the experimental procedure used in the recursive exercise since careful attention has to be dedicated to the modeling of information sets available at each point in time to the forecaster.

II.1 The dataset

Forecasting an aggregate variable like GDP potentially involves to choose between thousands of economic indicators. Many empirical applications rely on economic theories, like the theory of business cycle, to choose which variable should or shouldn't enter the forecasting model. Although this is a legitimate approach when evaluating the explaining power of a given theory, this is probably not the best way to extract the most valuable information from the available data in a purely predictive perspective. As underlined by (Litterman, 1986), there are *"a multitude of economic theories of the business cycle, most of which focus on one part of a complex, multifaceted problem"*. Here we will use econometrics rather than theory to model the data, and rely on specifications that allow to keep a large set of predictors, without making too many a-priori assumptions about which ones are driving the GDP dynamics.

Our outcome variable is the French quarterly GDP growth rate. We run estimation and forecasts using a large dataset of time series related to the French activity, and published by various institutions, namely INSEE, Banque de France, Eurostat, the OECD and the FRED. Our final dataset includes 285 monthly observations for 51 indicators between February 1991 and October 2014. All our scripts and data files can be found online[1]. Following the work of (Bessec and Doz, 2014), we include four distinct categories of predictors in our models, all of them being potentially relevant to predict GDP growth:

- Economic outlook surveys published by INSEE, which provide useful business climate and turning point indicators, activity expectations by industry, as well as households surveys. Results of these various surveys are available at the end of the current month, so that this constitutes in a sense the

[1] https://github.com/nsaleille

"freshest" data at the the disposal of the forecaster. They also have the advantage to be relatively long series, most most of them being produced since the 70's.

- Real variables such that households consumption, new cars registrations, construction and industrial production indexes, as well as labor market variables. These contain information on the effective economic and productive environment in the recent past.

- Nominal variables, mostly related to financial and monetary conditions: interest rates, stock and bond market indexes, volatility indexes, monetary aggregates and price indexes.

- Various variables related to the international environment: Euro exchange rate, indicators on the German and American economies. These variables capture the interaction of France with other countries modeled as its main economic partners.

Description of all the series used in our estimations can be found in Table 1 on page 9. All variables are corrected for seasonal variations. Our outcome variable is the GDP growth rate, published on a quarterly basis at the end of quarter Q+1. Among the 51 predictors, two of them are only available at a quarterly frequency: we convert them to monthly frequency thanks to cubic splines (obviously, we do not extrapolate the GDP series which is our outcome). Real and financial series are taken in logarithm and tested for stationarity. Broadly speaking, economic outlook surveys are stationary while real and financial variables are $I(1)$. Integrated series are differentiated until stationarity. All variables are then centered and normalized to variance one in order to facilitate the inversion of the matrix containing our predictors. Normalization is done once and for all before we create the information sets used in the recursive forecast evaluation exercise, in order to avoid any instability coming from time-dependent transformations.

II.2 Forecasting or nowcasting ?

In order to evaluate the quality of predictions, we conduct a real-time forecasting exercise, i.e. we pay close attention to build coherent information sets and to use only the information that was available to the forecaster at each date. Information regarding the release schedule of time series can be found in Table 1. French GDP figures for quarter Q are released by INSEE at the end of M3/Q+1. The first indicators of quarter Q are released in M1/Q, while others are published only during Q+1. In this framework, it is important to underline the difference between the *release date* (M3/Q+1) and the *quarter of reference* (Q). Denoting T the release date, we will forecast the GDP growth rate with information sets at time $T-h$ where h lies between 1 and 8 months. Formally speaking, we are performing a "nowcasting" exercise rather than pure forecasting. Indeed, when forecasting with $h < 3$ we are trying to "predict" the value of y at quarter Q while the current quarter is Q+1. In this case, the GDP and its growth rate have fixed values corresponding to last quarter's production level and acceleration, but these values are not correctly measured yet.

Recently, a popular research direction has been the use of Google Trends data for nowcasting: see for instance the Bayesian structural time series framework in (Scott and Varian, 2014). Google Trends provide an index of the volume of Google queries on specific terms. Indexes are published weekly, and researchers figured out that some specific queries might add some predictive power in a nowcasting exercise, in particular thanks to the high release frequency. (Bortoli and Combes, 2015) applied the Bayesian structural approach to predict the level of consumption spendings. They conclude that, on average, the inclusion of Google Trends predictors in a forecasting model does not provide significant improvements in the quality of forecasts. The main reason they advance is the heterogeneity of the components explaining consumption spendings, while Google queries are related to very specific concepts. Better results can be achieved when trying to predict some specific categories of spendings, such as spendings related to house equipment. Based on this previous work, and since we work at a very aggregated level, we choose to ignore Google trends data.

Table 1: Presentation of the data set

Type	Sector	Name	Frequency	Release date	Source
Real	Production	GDP growth rate	Q	M+3	INSEE
Survey	Industry	Level of global order books (balance of opinion)	M	M+0	INSEE
Survey	Industry	Level of foreign order books (balance of opinion)	M	M+0	INSEE
Survey	Industry	Level of stocks of manufactured products (balance of opinion)	M	M+0	INSEE
Survey	Industry	Past trend in production (balance of opinion)	M	M+0	INSEE
Survey	Industry	Expected trend in production (balance of opinion)	M	M+0	INSEE
Survey	Services	Past trend of activity - All services activities	M	M+0	INSEE
Survey	Services	Expected trend of activity - All services activities	M	M+0	INSEE
Survey	Services	Expected trend of demand - All services activities	M	M+0	INSEE
Survey	Building Industry	Trend of expected activity - Overall	M	M+0	INSEE
Survey	Building Industry	Trend of past activity - Overall	M	M+0	INSEE
Survey	Building Industry	Trend of expected enrollment evolution - Overall	M	M+0	INSEE
Survey	Building Industry	Past trend in the workforce - Overall	M	M+0	INSEE
Survey	Building Industry	Expected trend of prices - Overall	M	M+0	INSEE
Survey	Building Industry	Judgment on the order book level - Overall	M	M+0	INSEE
Survey	Retail and Auto	Business development in trade during the next 3 months - All sectors	M	M+0	INSEE
Survey	Retail and Auto	Business development (sales) in the last 3 months - All sectors	M	M+0	INSEE
Survey	Retail and Auto	Intents for orders in the next 3 months - All sectors	M	M+0	INSEE
Survey	Households	Opinion on their past financial situation	M	M+0	INSEE
Survey	Households	Opinion on their future financial situation	M	M+0	INSEE
Survey	Households	Opinion on the past standard of living in France	M	M+0	INSEE
Survey	Households	Opinion on the future standard of living in France	M	M+0	INSEE
Survey	Households	Opinion on whether to make major purchases	M	M+0	INSEE
Survey	Services	Past trend of operating results - All services activities	Q	M+3	INSEE
Survey	Services	Expected trend of operating results - All services activities	Q	M+3	INSEE
Real	Transport	New passenger cars registrations	M	M+1	INSEE
Real	Households	Consumption expenditure on goods - Durable good	M	M+1	INSEE
Real	Households	Consumption expenditure on goods - Motor vehicles	M	M+1	INSEE
Real	Industry	Industrial Production Index - all sectors	M	M+2	INSEE
Real	Industry	Industrial Production Index - manufacturing	M	M+2	INSEE
Real	Industry	Industrial Production Index - construction	M	M+2	INSEE
Real	Households	Unemployment rate - less than 25 years	M	M+2	OECD
Real	Households	Unemployment rate - total	M	M+2	OECD
Nominal	Monetary mass	Loans to non-financial agents	M	M+2	BDF
Nominal	Monetary mass	Monetary aggregate - M1	M	M+2	BDF
Nominal	Monetary mass	Monetary aggregate - M2	M	M+2	BDF
Nominal	Monetary mass	Monetary aggregate - M3	M	M+2	BDF
Nominal	Interest Rates	Long-Term Government Bond Yields: 10-year	M	M+1	FRED
Nominal	Price	SP500	M	M+0	Yahoo!
Nominal	Price	SP500 volatility (VIX)	M	M+0	Yahoo!
Nominal	Price	Euro Stoxx 50	M	M+0	Yahoo!
Nominal	Price	Gold Fixing Price 10:30 A.M. (London time) in London	M	M+0	FRED
Nominal	Price	Crude Oil Prices: Brent - Europe	M	M+0	FRED
International	Exchange rate	Japan / U.S. Foreign Exchange Rate	M	M+0	FRED
International	Exchange rate	U.S. / Euro Foreign Exchange Rate	M	M+0	FRED
International	Germany	Production of Total Industry	M	M+2	FRED
International	Germany	Production of Total Industry	M	M+2	FRED
International	Germany	Total Retail Trade	M	M+2	FRED
International	USA	Industrial Production Index	M	M+1	FRED
International	USA	Civilian Unemployment Rate	M	M+1	FRED
International	USA	Retail Sales: Total (Excluding Food Services)	M	M+1	FRED

III Dynamic Factor Models

Dynamic Factor Models (DFMs) are powerful forecasting tools as they provide parsimonious representations of the information provided by a large number of *correlated* variables. Their use for short-term macroeconomic forecasting was initiated by (Anderson, 1958), developed by (Geweke, 1977) and is now widely spread in central banks such as the US federal reserve and the ECB. Interesting overviews of their use and estimation wan be found in (Stock and Watson, 2006) and (Bessec and Doz, 2014). These models have at least two interesting features. First, DFMs make it possible take into account the information contained in large sets of predictors without risking over-parametrization as it is the case with VAR models. Furthermore, they are compatible with the use of the Kalman filter, an algorithm particularly well suited to handle missing data, so that it is possible to integrate the most recent data in short-term forecasts.

III.1 Model assumptions

The general idea behind Dynamic Factor Models (DFMs) is that the observable variables can be decomposed in two orthogonal unobserved processes: first, a common component that drives the bulk of the covariation between time series, and second, an idiosyncratic component (Doz et al., 2011). From now on, let's denote by T the number of observations in the training sample (i.e. the subsample used to estimate our model - remaining observations are left in a test sample, designed to evaluate the quality of our predictions). Starting from a large set of n predictors casted in the $(T \times n)$ matrix $X = (x_1', \dots, x_T')$, principal component analysis (PCA) uses an orthogonal transformation to convert the initial observations into a set of linearly uncorrelated variables called principal components. This transformation is defined so that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. By selecting the q first principal components, PCA allows to achieve dimensionality reduction by replacing the initial set of predictors by a small number of orthogonal regressors.

PCA is a classical methodology and has been extended to take into account the ordered nature of time-series: see (Doz et al., 2011). This approach gave rise to the following DFM representation:

$$x_t = \Lambda_0 f_t + \ldots + \Lambda_s f_{t-s} + \varepsilon_t \tag{3}$$

$$f_t = \sum_{i=1}^{p} A_i f_{t-i} + \eta_t \tag{4}$$

where x_t is a $(1 \times n)$ vector of predictors and f_t is a $(1 \times q)$ vector of factors, and ε_t is a white noise. These factors are dynamic (since they follow a VAR process of order p), and are orthogonal in the sens

$$\mathbb{E}[f_t f_t^{'}] = I_q \qquad \mathbb{E}[f_t f_s^{'}] = 0 \qquad \mathbb{E}[f_t \varepsilon_s^{'}] = 0 \qquad \forall s \neq t \tag{5}$$

Setting $F_t = (f_t', \dots, f_{t-p+1}')'$ a $(pq \times 1)$ vector, we can rewrite

$$x_t = \Lambda F_t + \varepsilon_t \tag{6}$$

$$F_t = A F_{t-1} + B \eta_t \tag{7}$$

where $\mathbb{V}(\varepsilon_t) = diag(\varphi_1, \dots, \varphi_n)$, B is a $(n \times pq)$ matrix with rank pq, and $\mathbb{V}(B\eta_t) = \Sigma$. The model is said to have $r = pq$ static factors (the components of F_t) and q dynamic factors (the components of f_t). Compared to a classical PCA, this model has two interesting features: first, it explicitly takes into account the time-series structure of factors; second, factors can be estimated even when some observations are missing for some predictors in the training sample. This last feature is of particular interest since, as forecasters, we are interested in using the whole information set at our disposal despite non-matching release date of specific series.

III.2 Two-step estimation procedure

One great advantage of this last formulation comes from the possibility to cast it into state space form. The state-space formulation enables the use of the Kalman filter and makes it possible to recover the dynamic factors consistently, thanks to the estimation method first introduced in (Doz et al., 2011). In a first step, the parameters of the model are estimated from an OLS on principal components. Then in a second step, the values of factors for the dates where the set of predictors is incomplete are estimated via the Kalman filter.More precisely, the two step procedure works as follow:

(1) In the first step, PCA is conducted on dates for which all the predictors values in x_t are available. The PCA estimates are given by

$$\hat{\Lambda}_0 = PD^{\frac{1}{2}} \tag{8}$$

$$\hat{f}_t = D^{-\frac{1}{2}} P x_t \tag{9}$$

where $D = diag(d_1, \dots, d_q)$, d_i is the i-th eigenvalue of the estimated covariance matrix of X, and $P = [u_1, \dots, u_q]$ where u_i are the corresponding eigenvectors. Using this first estimates of the factors, the full matrix $\Lambda = [\Lambda_0, \dots, \Lambda_s]$ is estimated through an OLS regression conducted on equation (3). We then estimate $\mathbb{V}(\varepsilon_t) = diag(\varphi_1, \dots, \varphi_n)$ using an unbiased estimator of the variance on estimated residuals $\hat{\varepsilon}_t$. Parameters of the VAR equation (4) are estimated thanks to regression of factor estimates on their own past (the order p of the VAR is selected via the AIC criterion). Finally, the residuals $\hat{\xi}_t = B\hat{\eta}_t$ are used to estimate the *diagonal* covariance matrix Σ_ξ and we set $B = \Sigma_\xi^{-1/2}$ so that $\mathbb{V}(\eta_t) = I_{q \times s}$.

(2) In the second step, the factors are estimated a second time using all available information in the dataset. Missing values (i.e. values for which factors can't be directly estimated through the PCA) are replaced by the optimal approximation provided by the Kalman filter. At each date t, we set

$$\mathbb{E}[e_{i,t}^2] = \begin{cases} \varphi_i & \text{if } x_{i,t} \text{ is observed} \\ +\infty & \text{else} \end{cases} \tag{10}$$

where φ_i is the estimated covariance of $x_{i,t}$. This second steps sets to zero the weight attached to unobserved variables at time t in the Kalman filter algorithm. The estimated factors integrate all the information set and are optimal.

III.3 Forecasting with DFMs

Once estimated, the dynamic factors are used to construct short-term forecasts. First of all, monthly factors are aggregated to quarterly values. For that we set the value of the factors at quarter Q to the value estimated at M1/Q. The forecasting exercise then consists in the selection of a forecast horizon $h > 0$ and the estimation by OLS the parameters in the regression

$$y_{t+h} = \sum_{i=1}^{q} \delta_i f_{i,t} + e_{t+h} \qquad \forall\, t = 1, \dots, T-h \tag{11}$$

where y_t is the variable we want to forecast. The h steps-ahead forecast is then simply given by the projection

$$\hat{y}_{T+h|T} = \sum_{i=1}^{q} \hat{\delta}_i f_{i,T} \tag{12}$$

As underlined in (Stock and Watson, 2011), DFMs make an efficient use of the information in the many predictors for most macroeconomic series. This approach works less well for some series known to be difficult to predict (exchange rates, price inflation, stock prices), and new techniques described in the

following sections have been proven to provide better forecasting results. However, DFMs are relatively simple to estimate (mostly based on OLS regressions or maximum likelihood) compared to recent models that are often based on Bayesian priors and require MCMC algorithm to estimate posterior predictive densities.

IV Large Bayesian VARs

Since the founding paper of (Sims, 1980), vector autoregressive models (VARs) have been used for many years to forecast macroeconomic variables. These models are powerful forecasting tools, but they are affected by two well-identified issues: a huge number of parameters and structural change. As highlighted in section I.1, VARs with a reasonable amount of series have so many parameters that over-fitting is a serious risk. In a typical VAR, many coefficients are close to zero or poorly estimated, leading to large standard deviations and forecast uncertainty. Bayesian VARs took the lead in recent research precisely because they address the *curse of dimensionality*.

IV.1 Bayesian shrinkage in VAR models

Informative hierarchal Bayesian priors are a popular way to shrink the coefficients of VARs including dozens of variables in order to reduce over-fitting issues. Prior distributions $\mathcal{P}(\theta)$ are specified on the parameters of the VAR to bring additional information coming from the prior beliefs of the researcher in the estimation process. They are used to derive posterior distributions $\mathcal{P}(\theta|y^T)$ once we observed the data. To this regard, it is convenient to work with *conjugate* priors, i.e. priors that lead to posteriors in the same family of distributions. Conjugate priors are a nice way to avoid Monte-Carlo Markov-Chain algorithms, that are required to simulate from the posterior when it is not known analytically.

The most famous prior in the Bayesian VAR literature is probably the Minnesota prior, first introduced by (Litterman, 1986). This prior shrinks every coefficients of the VAR toward zero, except the ones corresponding to the first lags of the dependent variable in each equation, which are shrunk to one. In a nutshell, each equation of the VAR is shrunk by the prior toward a random walk, the most basic process that can be used in forecasting. The covariance matrix of the residuals Σ is viewed as a fixed parameter and assumed to be diagonal, with diagonal elements estimated by OLS in each equation. The Minnesota prior is powerful to increase the forecasting power of VARs, and has the great advantage to be a conjugate prior. However, it is based on rather restrictive assumptions and many extensions have been suggested since the founding article of Litterman. A particularly interesting approach was presented in (Bańbura et al., 2010). While researchers working with many variables traditionally used DFMs to forecast, they find that Bayesian VARs can perform better on a dataset of up to 130 predictors. They work with a conjugate extension of the Minnesota prior, that includes a Wishart prior on the covariance matrix of the residuals Σ. Their promising empirical results further strengthened the Bayesian approach as an efficient way to control over-fitting in large VARs, and led to a rich literature in the last years. In particular, the recent contribution of (Koop, 2013) suggested to estimate large VARs thanks to a conjugate version of the stochastic search variable selection (SSVS) algorithm of (George et al., 2008). This approach is conjugate and more flexible than the Minnesota prior; we present it in the next subsections.

Last but not least, it is also worth emphasizing an additional benefit of the Bayesian approach (Litterman, 1986). In a forecasting perspective, DFMs only provided point forecasts. In the Bayesian framework, this is not true anymore, and we can associate an entire probability distribution to the forecast. This distribution is called the posterior predictive density, and brings an additional information about the degree of certainty that we may have in a point forecast (which is often given by the posterior median or mean).

IV.2 Natural conjugate prior for VARs

Natural conjugate priors are those where the prior, likelihood and posterior come from the same family of distributions. The reduced form VAR model writes

$$y_t = a_0 + \sum_{i=1}^{p} A_i y_{t-i} + \varepsilon_t \qquad \forall t = 1, \dots, T \tag{13}$$

where y_t is a $(n \times 1)$ vector containing observations of n time series, a_0 is a $(n \times 1)$ constant term, $A_1, \dots, A_p$ are $(n \times n)$ coefficients matrices, and $\varepsilon_t \sim^{i.i.d} \mathcal{N}(0, \Sigma)$. Let's define $Y = (y_1, \dots y_T)'$, the $(T \times n)$ matrix which stacks the observations, $x_t = (1, y_{t-1}, \dots, y_{t-p})'$ and $X = (x_1, \dots x_T)'$. Finally, let's set $A = (a_0, A_1, \dots, A_p)$ a $(n \times (np+1))$ matrix and $\alpha = vec(A)$. Then the VAR rewrites

$$y = (I_n \otimes X)\alpha + \varepsilon \tag{14}$$

where $\varepsilon \sim \mathcal{N}(0, \Sigma \otimes I_n)$. The likelihood writes

$$\mathcal{L}(y, \alpha) = \log l(y_1, \dots, y_T | y_{-p+1}, y_0, \alpha) = \sum_{t=1}^{T} \log l(y_t | y^{t-1}, \alpha) \tag{15}$$

$$= \frac{-nT}{2}\log(2\pi) - \frac{T}{2}\log\det\Sigma - \frac{1}{2}tr\left[\Sigma^{-1}\sum_{t=1}^{T}(y_t - m_t(\alpha))(y_t - m_t(\alpha))'\right] \tag{16}$$

where $m_t(\alpha) = a_0 + \sum_{i=1}^{p} A_i y_{t-i}$. From the Zellner theorem in SURE models, we know the maximum likelihood estimator is equivalent to the OLS estimator $\hat{A} = (X'X)^{-1}X'y$. The natural conjugate prior has the form

$$\alpha | \Sigma \sim \mathcal{N}(\underline{\alpha}, \Sigma \otimes \underline{V}) \qquad \Sigma^{-1} \sim W(\underline{S}^{-1}, \underline{\nu}) \tag{17}$$

where $\underline{\alpha}, \underline{V}, \underline{S}, \underline{\nu}$ are hyper-parameters chosen by the researcher. Combined with the expression of the likelihood, one finds the conjugate expression of the posterior distributions

$$\alpha | \Sigma, y \sim \mathcal{N}(\overline{\alpha}, \Sigma \otimes \overline{V}) \qquad \Sigma^{-1} | y \sim W(\overline{S}^{-1}, \overline{\nu}) \tag{18}$$

where the posterior parameters are given by

$$\begin{cases} \overline{V} = \left[\underline{V}^{-1} + X'X\right]^{-1} \\ \overline{A} = \overline{V}\left[\underline{V}^{-1}\underline{A} + X'X\hat{A}\right] \\ \overline{\alpha} = vec(\overline{A}) \\ \overline{\nu} = T + \underline{\nu} \\ \overline{S} = S + \underline{S} + \hat{A}'X'X\hat{A} + \underline{A}'\underline{V}^{-1}\underline{A} - \overline{A}'(\underline{V}^{-1} + X'X)\overline{A} \end{cases} \tag{19}$$

and $S = (Y - X\hat{A})'(Y - X\hat{A})$. Those expressions are simplified if we set a very diffuse prior value for $\underline{V}$, so that $\underline{V}^{-1} \to 0$.

$$\begin{cases} \overline{V} = [X'X]^{-1} \\ \overline{A} = (X'X)^{-1}X'y \\ \overline{\alpha} = vec(\overline{A}) \\ \overline{\nu} = T + \underline{\nu} \\ \overline{S} = (Y - X\hat{A})'(Y - X\hat{A}) \end{cases} \tag{20}$$

After integrating out for Σ, the marginal posterior for α is a multivariate student distribution with mean $\overline{\alpha}$, $\overline{\nu}$ degrees of freedom, and covariance matrix

$$\mathbb{V}(\alpha | Y) = \frac{1}{\overline{\nu} - n - 1}\overline{S} \otimes \overline{V} \tag{21}$$

Furthermore the one-step ahead posterior predictive distribution is also a multivariate student with $\overline{\nu}$ degrees of freedom, posterior predictive mean $\mathbb{E}[y_{T+1}|Y] = (x_{T+1}\overline{A})'$, and covariance

$$\mathbb{V}\left(y_{T+1}|Y\right) = \frac{1}{\overline{\nu}-2}[1 + x_{T+1}\overline{V}x'_{T+1}]\overline{S} \tag{22}$$

As underlined in (Koop, 2013), a great advantage of natural conjugate priors is the existence of an analytical formula for the one-step ahead posterior predictive density. When forecasting more than one period ahead, this is not the case anymore and simulation is required. Typically, in large VARs, posterior simulation is too computationally demanding. A similar issue arises when working with non-conjugate priors, such as the SSVS prior of (George et al., 2008), since very large matrices have to be inverted in the posterior computation.

On the other hand, the natural conjugate prior has a restrictive property that $\mathbb{V}(\alpha|\Sigma) = \Sigma \otimes \underline{V}$. This formulation implies that the prior variance of the coefficients on the same explanatory variable in any two equations must be proportional. To this regard, the standard Minnesota prior (where Σ is not treated as random but assumed to be diagonal and estimated through OLS) is more flexible since coefficients on own lags have a larger prior variance than coefficients on other lags. (Bańbura et al., 2010) applies the same degree of shrinkage to all parameters.

IV.3 Stochastic search variable selection (SSVS) prior

The stochastic search variable selection (SSVS) prior was initiated by (George et al., 2008) and provides an interesting way to perform shrinkage in VAR models. However this prior is not conjugate and requires simulation, so that it is not adapted to VARs with more than 30 variables. To overcome this difficulty, (Koop, 2013) proposed a conjugate version of the SSVS prior.

The SSVS prior is hierarchical involving the mixture of two Normal distributions.

$$\alpha_j|\gamma_j \sim (1-\gamma_j)\,\mathcal{N}\left(\underline{\alpha}_j, \kappa_{0,j}^2\right) + \gamma_j\,\mathcal{N}\left(\underline{\alpha}_j, \kappa_{1,j}^2\right) \tag{23}$$

where $\gamma_j = \mathbb{1}\{\alpha_j \neq 0\}$ is a random parameter that indicates if variable j enters the model or not. The SSVS prior is based on the spike and slab specification, i.e. the hyper-parameters $\kappa_{0,j}$ and $\kappa_{1,j}$ are chosen to be respectively small and large, while traditionally $\underline{\alpha}_j = 0$. This prior can be rewritten

$$\alpha|\gamma \sim \mathcal{N}\left(\underline{\alpha}, D\right) \tag{24}$$

where D is a diagonal matrix with elements given by

$$d_j = \begin{cases} \kappa_{0,j}^2 & \text{if} \quad \gamma_j = 0 \\ \kappa_{1,j}^2 & \text{if} \quad \gamma_j = 1 \end{cases} \tag{25}$$

If variable j is in the spike of the distribution ($\gamma_j = 0$), the coefficient is constraint to be very close to $\underline{\alpha}_j$ ($\kappa_{0,j}$ is not exactly set to zero to avoid inversion issues). If it is in the slab ($\gamma_j = 1$), then the prior is relatively non-informative. Finally, the hierarchical SSVS prior is completed with independent Bernouilli priors on the elements of γ:

$$\mathcal{P}\left(\gamma_j = 1\right) = q_j, \qquad \mathcal{P}\left(\gamma_j = 0\right) = 1 - q_j \tag{26}$$

where $q_j = 0.5$ so that each coefficient is a priori equally likely to be included or not in the model.

This specification has been found in (George et al., 2008), (Jochmann et al., 2010) and (Korobilis, 2013) to be an efficient way to shrink coefficients and improve the forecasting power of small VARs. However the SSVS prior is not a natural conjugate, and the posterior predictive distribution is not known analytically. In order to perform Bayesian inference, we need to evaluate the posterior using

MCMC, an infeasible task if the dimension of the VAR is too large. Indeed, in a recursive forecasting exercise, the MCMC algorithm has to be repeated many times (once for each date in the testing dataset). In the case of the SSVS prior, this algorithm involves the computation of

$$\mathbb{V}(\alpha|Y,\Sigma,\gamma) = \left[\Sigma^{-1}\otimes(X'X)+D^{-1}\right]^{-1} \tag{27}$$

i.e. a $n(pn+1)\times n(pn+1)$ has to be inverted for each MCMC draw.

To overcome these computational issues in large VARs, (Koop, 2013) suggests to use a conjugate version of the SSVS prior. Denoting $\tilde{\gamma}$ a $(n\times 1)$ vector of dummy variables (while γ was $(np+1\times 1)$), the SSVS conjugate prior is

$$\alpha|\Sigma,\tilde{\gamma}\sim\mathcal{N}(\underline{\alpha},\Sigma\otimes D) \tag{28}$$

where D is a $(np+1\times np+1)$ diagonal matrix with elements defined as before in equation (25). Conditionally to $\tilde{\gamma}$, equations (18) holds. The trick in this conditional approach is that the posterior distribution for models is easy to evaluate. Using a standard formula for the marginal likelihood and a non-informative prior for $\tilde{\gamma}$, (Koop, 2013) gets

$$\mathcal{P}(\tilde{\gamma}|Y)\propto\left(|D|\left|\overline{V}^{-1}\right|\right)^{-\frac{n}{2}}\left|\overline{S}\right|^{\frac{1}{2}(-T+n+\underline{\nu}-1)} \tag{29}$$

Since 2^K models are possible, this posterior can't be fully evaluated if K is large. The author propose a simulation strategy.

V Time Varying Parameter VAR models

Time varying parameters VAR models (TVP-VARs) are an interesting alternative to standard constant parameter representations, such as VAR models. When analyzing time series on long periods of time, allowing for some flexibility in the parameter space is important if ones wants to capture structural change in macroeconomic dynamics. The forecasting performance of these models have been studied recently in (Koop and Korobilis, 2013) and (Belmonte et al., 2014).

V.1 The TVP-VAR setting

Suppose we want to predict the value of a variable y_t using n predictors stacked in a $(n\times 1)$ vector x_t. The standard state-space representation of a TVP-VAR model writes

$$y_t = \theta_t x_t+\varepsilon_t \tag{30}$$
$$\theta_t = \theta_{t-1}+\eta_t \tag{31}$$

where the unobserved parameter θ_t is a $(n\times 1)$ vector. The error terms in the measurement and state equations are assumed to be independent Gaussian processes:

$$\varepsilon_t\sim\mathcal{N}(0,H_t)\qquad \eta_t\sim\mathcal{N}(0,Q_t) \tag{32}$$

In this setting, the main parameter θ_t varies gradually over time. Note that here, we modeled θ_t as a random walk, i.e. a random process centered in θ_0 with growing variance. However this doesn't mean that we are in a Bayesian framework, since we did not specified any prior linked to the *uncertainty* on θ_t; rather, we imposed a structure defined by the state equation (31). Furthermore, the model includes stochastic volatility through the sequence of matrices H_t and Q_t, so that we leave the simple homoscedatic framework.

A first approach to estimate such models requires specifications for the processes (H_t) and (Q_t), as well as priors for parameters θ_0, H_0 and Q_0. Then, it is possible to use MCMC methods to draw from

this particular state-space model using recursive computations of the likelihood provided by the Kalman filter, and to estimate the posterior predictive distribution. This implies to draw sequentially, for all dates, in the conditional distributions of $(\theta_t|H_t, Q_t)$, $(H_t|\theta_t, Q_t)$, and $(Q_t|\theta_t, H_t)$. This task is feasible only for TVP-VARs including a small set of predictors (and thus a limited parameter size). Even for small TVP-VARs, a recursive forecasting exercise is very computationally demanding since the number of parameters is huge and the posterior simulation algorithm must be repeated many times.

Hence, as underlined in (Koop and Korobilis, 2013), forecasting with medium or large TVP-VARs is, in practice, computationally infeasible using MCMC methods. The solution they provide consists of the use of approximations based on *forgetting factors*, that leads to closed form solutions for the posterior distributions in the Kalman filter equations. The resulting solutions can be computed quickly and allow to introduce uncertainty on the model and *model averaging*. Forgetting factors are described in the next subsection, while the pros and cons of model averaging will be discussed in section VI.

V.2 Estimation through the Kalman filter

In the state-space setting described by equations (30) and (31), the relevant tool to learn form the data is the Kalman filter. As we show in the following discussion, this two-steps algorithm allows to recover the parameters of the parameter posterior distribution $\mathcal{P}\left(\theta_t|y^{t-1}\right)$ at every date t, as well as parameters of the posterior predictive distribution $\mathcal{P}(y_t|y^{t-1})$. From now on, let's define:

$$\hat{\theta}_{t|s} = \mathbb{E}\left[\theta_t|y^s\right], \quad \Sigma_{t|s} = \mathbb{V}\left(\theta_t|y^s\right), \quad \text{and} \quad F_t = \mathbb{V}\left(y_t|y^{t-1}\right)$$

The Kalman filter relies on the two following sequential steps:

(1) **Predict**. In the prediction step, we compute the parameters of the posterior distributions of θ_t and $y_t$conditional to y^{t-1}:

$$\begin{cases} \hat{\theta}_{t|t-1} = \hat{\theta}_{t-1|t-1} \\ \Sigma_{t|t-1} = \Sigma_{t-1|t-1} + Q_t \\ F_{t|t-1} = x_t \Sigma_{t|t-1} x_t' + H_t \\ Cov\left(y_t, \theta_t|y^{t-1}\right) = \Sigma_{t|t-1} x_t' \end{cases} \tag{33}$$

The distribution of the couple $(\theta_t, y_t)|y^{t-1}$ is Gaussian and given by

$$(\theta_t, y_t)|y^{t-1} \sim \mathcal{N}\left(\begin{pmatrix} \hat{\theta}_{t|t-1} \\ \hat{y}_{t|t-1} \end{pmatrix}, \begin{pmatrix} \Sigma_{t|t-1} & \Sigma_{t|t-1} x_t' \\ x_t \Sigma_{t|t-1} & F_{t|t-1} \end{pmatrix}\right) \tag{34}$$

(2) **Update**. Once we have observed y_t, we can compute the moments of the full conditional distribution of interest

$$\theta_t|y^t \sim \mathcal{N}\left(\hat{\theta}_{t|t}, \Sigma_{t|t}\right)$$

Indeed, we know that the couple $(\theta_t, y_t)|y^{t-1}$ is Gaussian, so that $\theta_t|y^t$ is also Gaussian with parameters given by the standard formula for conditional distributions of Gaussian processes. Using this trick we easily compute the updating equations:

$$\hat{\theta}_{t|t} = \hat{\theta}_{t|t-1} + \Sigma_{t|t-1} x_t' F_{t|t-1}^{-1}\left(y_t - x_t \hat{\theta}_{t|t-1}\right) \tag{35}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1} x_t' F_{t|t-1}^{-1} x_t \Sigma_{t|t-1} \tag{36}$$

These two steps are conducted sequentially for every observation in the training sample in order to get the sequence of filtered states $(\hat{\theta}_{t|t})$ and covariance matrices $(\Sigma_{t|t})$. Forecasting is finally achieved based on the closed form posterior predictive distribution:

$$y_{T+1}|y^T \sim \mathcal{N}\left(x_T \hat{\theta}_{T+h|T}, \quad x_t \Sigma_{T|T-1} x_t' + H_T\right) \tag{37}$$

The Kalman filter is a nice way to derive a closed form one-step-ahead predictive distribution. However, we still need to find a way to compute the sequences (H_t) and (Q_t) in order to filter the unobserved values of the parameter (θ_t) from the data. These can't be considered as model parameters, since estimation would require n^2T degrees of freedom for each sequence. The *forgetting factor* approach is based on an explicit specification of their dynamics, so that the only parameters left to be specified are the hyper-parameters θ_0, H_0 and Q_0.

V.3 Approximate solution using forgetting factors

Going back to system (33), one can see that the stochastic volatility matrices Q_t and H_t enter the Kalman filtering formulas only in the predict equations for $\Sigma_{t|t-1}$ and $F_{t|t-1}$. In a Bayesian perspective, these matrices are treated as model parameters and, without any further assumptions, they have to be drawn in the posterior simulation steps. As underlined in (Koop and Korobilis, 2012), huge computational gains can be achieved thanks to the two following assumptions.

Assumption: *Forgetting factor for $\Sigma_{t|t-1}$ - (Raftery et al., 2010)*

The conditional covariance matrix of the state equation can be approximated by

$$\Sigma_{t|t-1} = \frac{1}{\lambda}\Sigma_{t-1|t-1} \qquad \lambda \in [0,1] \tag{38}$$

where λ is a forgetting factor. Implicitly, this is equivalent to assuming that the covariance matrix of the state equation writes

$$Q_t = \left(\frac{1}{\lambda} - 1\right)\Sigma_{t-1|t-1} \tag{39}$$

This specification is very handy. Intuitively, it implies that the current parameter value is estimated using past observations with exponentially decreasing weights: observations j periods in the past have weight λ^j. The effective window size is given by $\frac{1}{1-\lambda}$, so λ has to be set close to 1 if we want to obtain a gradual evolutions rather than parameters that vary a lot between periods. A similar assumptions is made on H_t, the conditional covariance matrix of the measurement equation.

Assumption: *Exponentially weighted moving average for H_t - (Koop and Korobilis, 2012)*

The conditional covariance matrix of the measurement equation can be estimated through an exponentially weighted moving average

$$\hat{H}_{t+1|t} = \left[(1-\kappa)\sum_{j=1}^{t}\kappa^{j-1}\varepsilon_t\varepsilon_t'\right]^{\frac{1}{2}} \tag{40}$$

where $\kappa \in [0,1]$ is an other forgetting factor. An attractive feature of this specification is the following recursive approximation

$$\hat{H}_{t+1|t} = \kappa\hat{H}_{t|t-1} + (1-\kappa)\varepsilon_t\varepsilon_t' \tag{41}$$

Once again, the forgetting factor specification implies that (H_t) is subject to gradual change, and follows a process parametrized by the "persistence" parameter κ.

The forgetting factor specification allows to avoid likelihood maximization since the model doesn't have any parameters left to be estimated (other than the state variables). In this framework the estimation strategy only involves on a single run of the Kalman filter on the training sample to filter the state variables: since the Kalman filter is very efficiently implemented in most programming languages, this means that estimation comes at a very low cost. Forecasting is based on the last estimated state and the

posterior predictive distribution given by equation (37). The hyper-parameters left to be specified are θ_0, H_0 and Q_0, as well as the forgetting factors λ and κ; regarding this last choice, (Koop and Korobilis, 2012) conduct some sensitivity exercise, but no systematic rule is proposed. One has to keep in mind that forgetting factor approximations are justified by computational gains rather than model needs. This means that close attention should be dedicated to robustness checks in order to verify that these rather strong assumptions are plausible.

VI Dynamic Model Averaging

Dynamic model averaging (DMA) is a flexible forecasting approach, introduced by (Raftery et al., 2010) in the engineering literature, and applied to inflation forecasts by (Koop and Korobilis, 2012). DMA introduces two kinds of flexibilities. First, the model rely on the TVP-VAR specification described in equations (30) and (31), so that *gradual change* in the statistical relationship is made possible. The second source of flexibility comes from the possibility for predictors to enter and leave the model dynamically, i.e the model performs *dynamic variable selection*. On the other hand, estimation of DMA models requires the use of approximate solutions and forgetting factors as described in the previous section in order to obtain computationally tractable solutions. The next subsection provides statistical motivations regarding the usefulness of model averaging in a forecasting exercise. We then present the multimodel setting of (Koop and Korobilis, 2012).

VI.1 Model averaging

The general idea behind model averaging is presented in (Hastie and Tibshirani, 2001). Suppose we have a sequence of candidate models $(\mathcal{M}_m)$, $m = 1, \dots, M$ and a training set Z, which we will use to estimate the posterior distribution of a given random variable ξ. We can write

$$\mathcal{P}(\xi|Z) = \sum_{i=1}^{m} \mathcal{P}(\xi|\mathcal{M}_i, Z)\,\mathcal{P}(\mathcal{M}_i|Z) \tag{42}$$

where the right and side of (42) depends on posterior probabilities for ξ conditional to the model, weighted by model posterior probabilities. Usually, we will be interested in the posterior mean:

$$\mathbb{E}[\xi|Z] = \sum_{i=1}^{m} \mathbb{E}[\xi|\mathcal{M}_i, Z]\,\mathcal{P}(\mathcal{M}_i|Z) \tag{43}$$

The posterior mean is simply a weighted average of individual model predictions, with weights equal to the posterior probability of each model. Applied to the specific issue of macroeconomic forecasting, this means that we make forecasts based on

$$\mathbb{E}\left[y_t|y^{t-1}\right] = \sum_{i=1}^{m} \mathbb{E}\left[y_t|\mathcal{M}_i, y^{t-1}\right]\mathcal{P}\left(\mathcal{M}_i|y^{t-1}\right) \tag{44}$$

How should this increased complexity help improving forecasts ? In econometrics, most of the work is traditionally done admitting a specific parametric model, the whole uncertainty and statistical inference focusing only on the parameter value. Specifications are often tested but it is not standard practice to focus on model uncertainty. As underlined in (Varian, 2014), the new challenge for modern econometrics is to consider both types of uncertainty simultaneously. In the context of macroeconomic forecasting, we are not specifically interested in the explaining power of a given model as it is typically the case in structural analysis. Rather, we are focused on improving the forecasting performance, and considering many models at once is a good way to enrich the analysis and avoid to base the whole prediction on a single, possibly misspecified model.

The benefits of model averaging are easily identified in a frequentist perspective. Given predictions $\hat{F}'(x) = [\hat{f}_1(x), \ldots, \hat{f}_M(x)]$ in the M different models, frequentist model averaging amounts to look for the best linear combination under squared-error loss (at a given point x)

$$\hat{w} = \arg\min_w \mathbb{E}\left[y - \sum_{i=1}^{M} w_i \hat{f}_i(x)\right] \tag{45}$$

The solution is a linear regression estimator $\hat{w} = \mathbb{E}\left[\hat{F}(x)\hat{F}'(x)\right]^{-1} \mathbb{E}\left[\hat{F}(x)y\right]$. It is then straightforward to see that the risk associated to the full regression is always smaller or equal to the one associated to any single model:

$$\mathbb{E}\left[y - \sum_{i=1}^{M} w_i \hat{f}_i(x)\right]^2 \leq \mathbb{E}\left[y - \hat{f}_m(x)\right]^2 \quad \forall m \tag{46}$$

At the population level, combining models never make things worse. However in practice, the linear regression has to be carried on an incomplete sample (the training set), and there are simple examples where this doesn't work well. For instance if $\hat{f}_m(x)$ is the prediction from the best subset of predictors with size m, then linear regression will put all the weight on the largest model. In practice, model complexity will have to be taken into account in order to penalize in some way non-parsimonious models.

Going back to a Bayesian perspective, and assuming that individual predictions are easy to compute, the difficult step is to find a strategy to estimate posterior model probabilities. If each model $\mathcal{M}_i$ is parametrized by θ_i, posterior model probabilities rewrites

$$\mathcal{P}(\mathcal{M}_i|Z) \propto \mathcal{P}(\mathcal{M}_i)\,\mathcal{P}(Z|\mathcal{M}_i) \tag{47}$$

$$\propto \mathcal{P}(\mathcal{M}_i) \int \mathcal{P}(Z|\theta_i, \mathcal{M}_i)\,\mathcal{P}(\theta_i|\mathcal{M}_i)\,d\theta_i \tag{48}$$

Estimating posterior probabilities requires to specifies priors for the models $\mathcal{P}(\mathcal{M}_i)$ and for the parameters $\mathcal{P}(\theta_i|\mathcal{M}_i)$. Then, it is possible to numerically compute the posterior probabilities. A simpler approach would be to estimate this probabilities using the BIC criterion.

VI.2 Multi-model setting

Suppose we want to predict the value of a variable y_t at time t, based on a maximum number of n predictors. In this setting, the model space dimension is $M = 2^n$. Let's define $\mathcal{M}_t = (\mathcal{M}_{t,1} \ldots, \mathcal{M}_{t,M})$ the sequence of models at time t. Then, the conditional TVP-VAR representation writes

$$y_t = X_t^{(k)}\theta_t^{(k)} + \varepsilon_t^{(k)} \tag{49}$$

$$\theta_{t+1}^{(k)} = \theta_t^{(k)} + \eta_t^{(k)} \tag{50}$$

where subscripts (k) indicates $x_t^{(k)} = x_t|\mathcal{M}_{t,k}$. In each model, the error terms in the measurement and state equations are supposed to be independent and Gaussian:

$$\varepsilon_t^{(k)} \sim \mathcal{N}\left(0, H_t^{(k)}\right) \qquad \eta_t^{(k)} \sim \mathcal{N}\left(0, Q_t^{(k)}\right) \tag{51}$$

In this multi-model setup, the underlying state variable consists in the pair $(\theta_t, \mathcal{M}_t)$. Ultimately, DMA constructs point forecasts using the full posterior predictive distribution:

$$\mathbb{E}\left[y_t|y^{t-1}\right] = \sum_{k=1}^{M} \mathbb{E}\left[\,y_t \,|\, \mathcal{M}_{t,k},\, y^{t-1}\right] \mathcal{P}\left(\mathcal{M}_{t,k} \,|\, y^{t-1}\right) \tag{52}$$

This is obviously not the only way to use model probabilities to construct forecasts. To this regard, an other popular approach is Dynamic Model Selection (DMS), which set the point forecast to the forecast obtained in the model with the higher probability:

$$\mathbb{E}\left[y_t|\mathcal{M}_{t,k^*},\, y^{t-1}\right] \quad \text{where} \quad \mathcal{P}\left(\mathcal{M}_{t,k^*}\,|\,y^{t-1}\right) = \max_{j=1,\dots,K} \mathcal{P}\left(\mathcal{M}_{t,j}\,|\,y^{t-1}\right) \tag{53}$$

VI.3 Estimation of model probabilities

So far the DMA specification lacks a process that specifies how predictors enter and leave the model. We need to define a process that indicates how we jump from one model to another between periods, i.e. we need to specify a transition matrix P with elements $\mathcal{P}(\mathcal{M}_{t,i}|\mathcal{M}_{t-1,j})$. Modeling unobserved jumps between states has been traditionally addressed through Markov Switching processes. However as underlined in (Koop and Korobilis, 2012), this approach is not feasible in the case of DMA. Since 2^n models are possible, the transition matrix P has dimension $(2^n \times 2^n)$. Estimation of such a large matrix is typically infeasible unless n is very small since over-parametrization would lead to huge computational costs and large imprecisions.

As shown in (Raftery et al., 2010), it is possible to use the Kalman filter to compute model probabilities in a computationally tractable way. In this approach the computational effort is reduced to $M = 2^n$ runs of the Kalman filter without the need for a heavy MCMC algorithm. The key assumptions are forgetting factors specifications for the dynamics of matrices (H_t), (Q_t), and for the dynamics of posterior model probabilities.

Assumption: *Conditional independence*

> The posterior predictive density depends on $\theta_{t+1}^{(k)}$ only conditionally to $\mathcal{M}_{t,k}$, i.e. y_t depends on model k only conditionally to this model.

Under this last assumption, system (50) provides the conditional distribution of the parameter conditionally to the selected model and the observations:

$$\theta_{t-1}^{(k)}\,|\,y^{t-1} \sim \mathcal{N}\left(\hat{\theta}_{t-1|t-1}^{(k)},\, \Sigma_{t-1|t-1}^{(k)}\right) \tag{54}$$

$$\theta_{t}^{(k)}\,|\,y^{t-1} \sim \mathcal{N}\left(\hat{\theta}_{t|t-1}^{(k)},\, \Sigma_{t|t-1}^{(k)}\right) \tag{55}$$

where the covariance matrix are defined sequentially: $\Sigma_{t|t-1}^{(k)} = \Sigma_{t-1|t-1}^{(k)} + Q_t^{(k)}$. As for standard TVP-VARs, without further assumptions, one would have to draw $H_t^{(k)}$ and $Q_t^{(k)}$ in a forecasting exercise. As described in the previous section, estimation is made possible thanks to forgetting factors formulations. The resulting approximate solution is computational feasible, and the predict / update steps of the Kalman filter can be computed in the 2^n models. This task is heavy but feasible since no simulation is required. When a new observation y_t is available, we use updating equations (35) and (36) in each of the M models to learn about the moments of the posterior predictive distributions in each model; see equation (37).

These two distributions are conditional to the model. As underlined before, in the DMA perspective we are ultimately interested in the posterior distributions of $\theta_{t-1}|y^{t-1}$ and $y_t|y^{t-1}$, i.e. in the distributions

$$\mathcal{P}\Big(.\Big|y^{t-1}\Big) = \sum_{k=1}^{M} \mathcal{P}\Big(.\Big|\,\mathcal{M}_{t,k}\,y^{t-1}\Big)\mathcal{P}\Big(\mathcal{M}_{t,k}\Big|y^{t-1}\Big) \tag{56}$$

Conditional distributions $\mathcal{P}(\theta_t|\,\mathcal{M}_{t,k}\,y^{t-1}\,)$ and $\mathcal{P}(y_t|\,\mathcal{M}_{t,k}\,y^{t-1}\,)$ are given respectively by (55) and (37). Hence, the only thing missing in the model is a specification for posterior model probabilities

$$\pi_{t|s,k} = \mathcal{P}\left(\mathcal{M}_{t,k}\,|\,y^s\right) \tag{57}$$

In a standard Bayesian perspective we could specify a prior for the distribution of $\mathcal{M}_t$ and estimate a posterior using MCMC methods. However, the whole DMA exercise relies on simplifying assumptions that allows to avoid MCMC algorithms. Once again (Koop and Korobilis, 2012) suggests to use a prior only for the initial probabilities $\pi_{0|0,k}$, and then to model their evolution via forgetting factors.

Assumption: *A3 - Forgetting factor for the posterior distribution of models*

The posterior distribution of $\mathcal{M}_t$ is approximated recursively by

$$\pi_{t|t-1,k} = \frac{\pi_{t-1|t-1,k}^{\alpha}}{\sum_{l=1}^{m} \pi_{t-1|t-1,l}^{\alpha}} \tag{58}$$

where $0 < \alpha \leq 1$ is a forgetting factor. The model updating equation thus have the following closed form solution:

$$\pi_{t|t,k} = \frac{\pi_{t|t-1,k}\, \mathcal{P}(y_t|\mathcal{M}_{t,k},\, y_{t-1})}{\sum_{l=1}^{m} \pi_{t|t-1,l}\, \mathcal{P}(y_t|\mathcal{M}_{t,l},\, y_{t-1})} \tag{59}$$

Intuitively, model k will receive more weight at time t if its forecasts were accurate in the recent past (i.e. in the window controlled by the forgetting factor α). If $\alpha = 1$, then posterior model probabilities at time t are proportional to the marginal likelihood of each model using data up to $t-1$. In this case we are in the standard Bayesian Model Averaging framework.

VII Forecast evaluation

In this section, we implement the forecasting methods previously discussed and conduct a pseudo real-time forecasting exercise on French data. Our dataset includes 285 monthly observations between February 1991 and October 2014 for the 51 economic time series previously presented in Table 1 on page 9. We compare forecasting performances and discuss the main differences identified in each approach. The interested reader can reproduce these results using our dataset and scripts available at `www.github.com/nsaleille`.

VII.1 Model evaluation through recursive forecasting

We evaluate the out-of-sample performance of the forecasts produced by models presented in the previous section using a recursive approach. First, we take out 50% of the available observations to build an initial training sample, which is used to estimate the different models and to compute forecasts. The remaining observations are used to build a test sample, on which we evaluate the quality of forecasts. Once the first forecasts are computed, we extend the initial training set by one observation, and repeat estimation as well as forecasts. These operations are repeated sequentially until all available observations are in the training sample. Following a standard practice, we then compare models on the basis of the "root mean squared error" (RMSE), a measure defined as

$$\widehat{\text{RMSE}} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(\hat{y}_{T_i+h|T_i} - y_{T_i+h})^2} \tag{60}$$

where $(T_1, \ldots, T_K)$ are the forecast dates of the K different training samples in the recursive exercise. This approach is called "real-time" forecasting, since at all training sets i, we base forecasts on the information that would have been available for the forecaster at time T_i. However we shall also underline that some series in our dataset were revised substantially. We don't take revisions into account when building information sets, so that it is more appropriate to speak of a *pseudo* real-time forecasting exercise.

VII.2 Parameter and hyper-parameter selection

Following the models and estimation methodologies presented in sections III and VI, we run the DFM and DMA approach to forecast the quarterly French GDP. In order to evaluate their performance relative to simpler models, we also run recursive forecasting exercises with a simple random walk and a unidimensional auto-regressive process of order p. We estimate the RMSE for all these models for different forecasting horizons $h = 1, \dots, 8$. Note that the horizon is defined as the time period between T_i, the forecast date of training sample i, and $T_i + h$, the date where the GDP figure for the previous quarter is released.

Benchmark models. Random-walk and autoregressive forecasts are performed in order to provide benchmark RMSE values. The random walk forecast is simply defined with the following model

$$y_t = y_{t-1} + \varepsilon_t \tag{61}$$

$$\hat{y}_{t+h|t} = \mathbb{E}\left[y_{t+h}|y^t\right] = y_t \tag{62}$$

The random walk forecast is just the last observed value of the outcome. For the autoregressive models we use the maximum likelihood estimator. For each horizon, we select the order p of the AR model such that it minimizes the RMSE criterion. Predictions are then formulated using the standard formulas in the AR(p) context.

Dynamic Factor estimates. In order to recover dynamic factor estimates from the training samples, we apply the two-steps estimation method presented in section III. In order to run estimation we need to select the number of static and dynamic factors under consideration. We select the combination that provides the best relative fit in the forecast regression given by equation (11). The adjusted-R2 leads us to select $q = 8$ static factors and $s = 10$ dynamic factors. We then use the estimated factors to conduct the full recursive forecast analysis. The DFM forecasts are computed using equation (12), where estimated factors have been aggregated from monthly to quarterly observations.

Dynamic Model Averaging. We implement DMA on the same dataset to assess the quality of forecasts compared to DFM and benchmark models. Typically, DMA is not computationally feasible with hundreds of predictors since estimation requires 2^K runs of the Kalman filter, where K is the number of predictors. To overcome this issue, we use the dynamic factors previously estimated in the DFM approach as DMA predictors. As underlined before, dynamic factors sum up efficiently the information contained in a large set of predictors: this approach has the merit to overcome the curse of dimensionality. On the other hand, it makes it more difficult to interpret model *evolutions* through time, mainly because factors have no clear interpretations. We add an intercept to the predictor set and treat it as a mandatory predictor (i.e. every model has at least an intercept), so as to filter models which won't be effective for predictions. Finally, following the recent empirical literature on DMA we set forgetting factors for the parameter covariance $\Sigma_{t|t-1}$, the model probabilities $\pi_{t|t,k}$, and the measurement equation covariance matrix H_t respectively to $\lambda = 0.99$, $\alpha = 0.99$ and $\kappa = 0.99$. These parameters imply that values five years ago receive around 80% as much weight as last period values. Finally, we use a data-based prior for the initial parameter value:

$$\theta_0^{(k)} \sim \mathcal{N}\left(a_0^{(k)}, b_0^{(k)}\right) \tag{63}$$

where $a_0^{(k)} = (X'X)^{-1}X'y$ is the OLS estimator in the training set in model k, and $b_0^{(k)} = \hat{\sigma}^2(X'X)^{-1}$.

VII.3 Empirical results

The final results of our recursive exercise are summed up in table 2. Each line presents the RMSE of the four different approaches tested on the French dataset, for 8 different forecast horizons (forecast from 1 to 8 month before the official GDP release by INSEE). Unsurprisingly, the AR(p) model works a little bit better than the simple random walk benchmark, in particular when forecasting with longer horizons. DFM forecasts better than the AR(p) at all horizons, with gains in RMSE of up to 39% for $h = 1$. This confirms that short-term predictors add a lot of information in the forecasting process, and are thus essential to build accurate figures.

Table 2: Recursive forecast exercise - RMSE from various models

	Random Walk	AR(p)	DFM	DMA	DMS
h = 1	0.38	0.34	0.21	0.21	0.23
h = 2	0.38	0.34	0.25	0.22	0.22
h = 3	0.38	0.35	0.28	0.25	0.25
h = 4	0.42	0.39	0.35	0.26	0.28
h = 5	0.42	0.40	0.32	0.27	0.31
h = 6	0.42	0.40	0.29	0.28	0.32
h = 7	0.51	0.41	0.37	0.28	0.31
h = 8	0.51	0.41	0.30	0.29	0.32

Of all results, DMA and DMS provide the most promising ones. Both approaches forecast almost uniformly better than DFM, with gains in terms of RMSE ranging from -13% (DMS versus DFM for $h = 1$) to +25% (DMA versus DFM for $h = 4$). Most importantly, the overall average gains compared to DFM are significantly positive: +11.9% for DMA and +3.8% for DMS. DMA does uniformly better than both DMS and DFM. We conclude it is the approach best suited to the French GDP forecasts problem, at least among the ones we tested. In particular, DMA should be preferred to DMS since it comes with no extra computational cost (model probabilities are required anyway). These good forecasting results in terms of RMSE confirm the previous empirical findings of (Koop and Korobilis, 2012) and (Bork and Møller, 2015). However, it is also worth emphasizing that none of the two approaches is able to predict the huge, very abrupt drop in the GDP growth rate observed in 2008. To illustrate that phenomenon, we plotted the one-month ahead recursive forecasts in figure 3. This limitation comes in part from the very conservative forgetting factors retained in our estimation: despite the signals contained in short-term indicators, model probabilities evolve only gradually through time.

Figure 2: Posterior probabilities for DMA models - most probable models, evolution through time

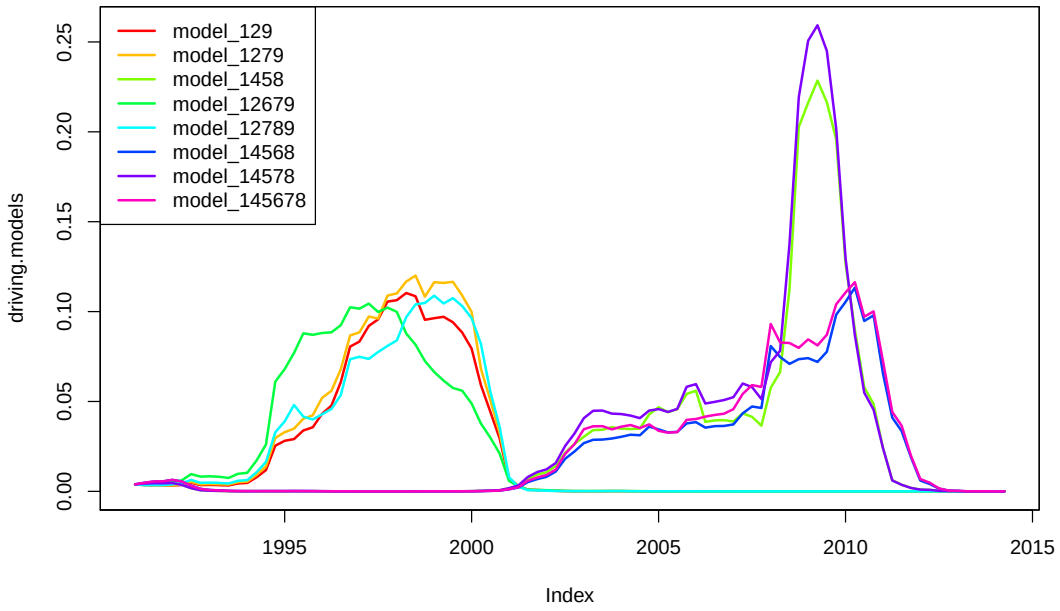
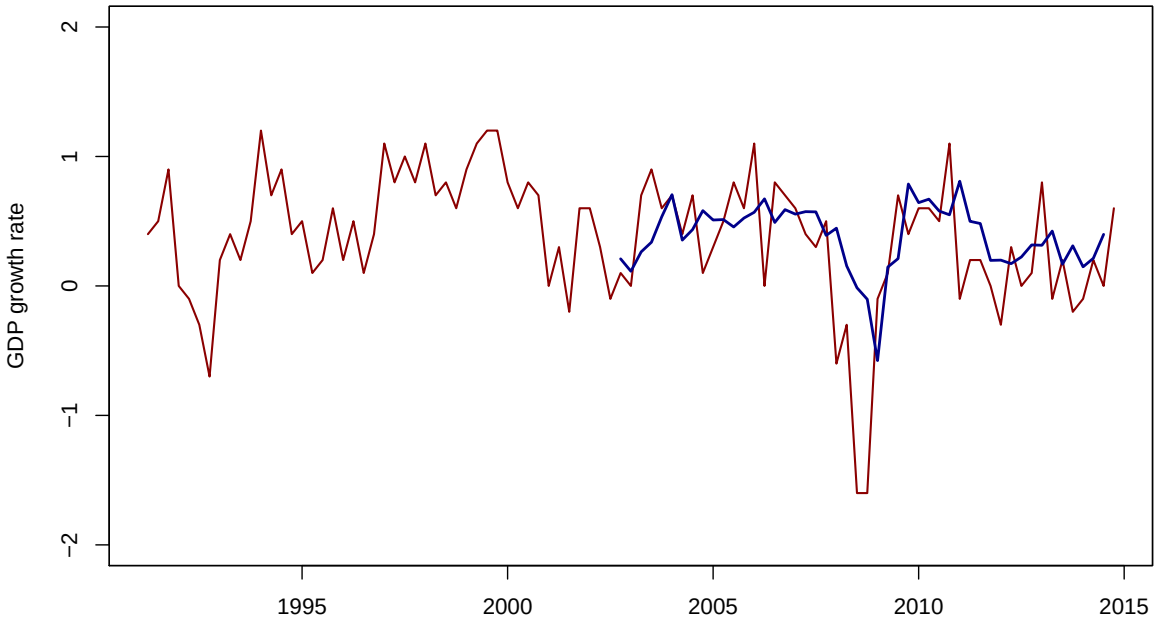
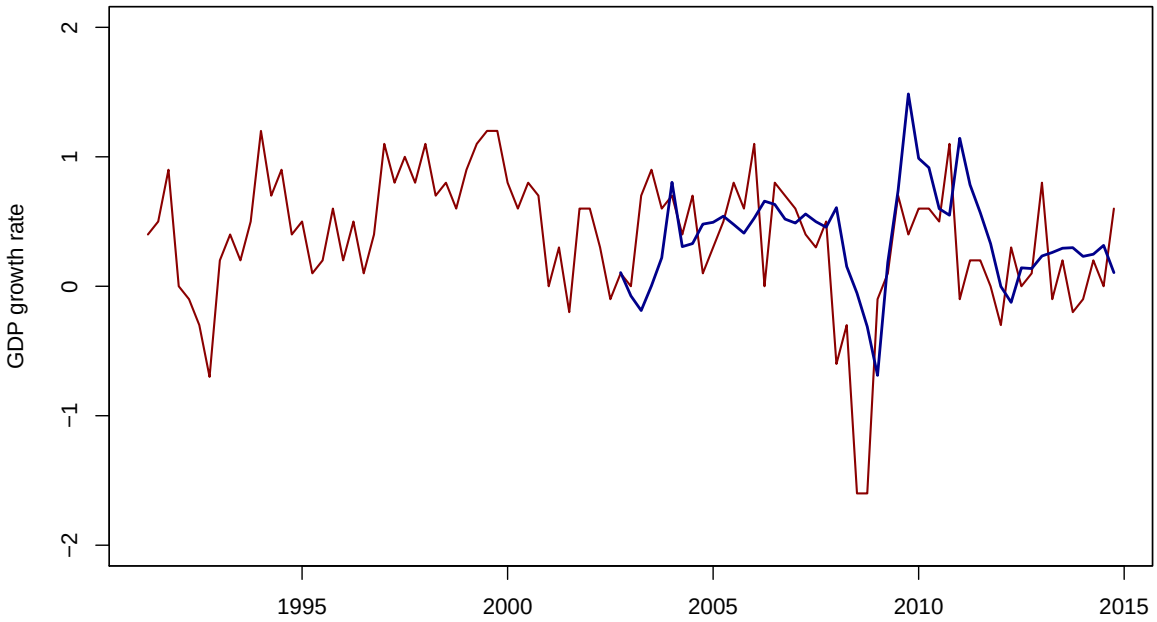


Figure 2 provides intuitions about model change through time. More precisely, we plotted a subset of posterior probabilities estimates, namely the ones that reach 10% at least at one point in time. Unlike (Koop and Korobilis, 2012), we do not find that DMA shrinks the model space and leads to parsimonious solutions. Rather, figure 2 shows that groups of similar models tend to have similar posterior probabilities patterns, with higher probabilities associated to models with more predictors. Far from surprising, this result directly comes from the fact that models with more predictors are not penalized. This 'lack' of shrinkage within the model space explains why DMS is slightly less efficient than DMA in a pure forecasting exercise: many models have similar probability patterns, and DMS won't switch efficiently between them. A solution to this problem would be to divide the model space into clusters based on probability patterns, and then choose one model by cluster to represent each group.

Finally, it is also important to underline the limits of our empirical results. First, we do not conduct any sensitivity analysis with respect to the forgetting factors α, λ and κ, and we don't use any test to fix them in a statistically justified way. Even if the previous literature has underlined low sensitivity of forecasts performances to these parameters, some further investigation might be useful. Second, we only look at a single evaluation measure (the RMSE), while measures based on the whole posterior predictive distribution might add some information to our interpretation. Third, the aggregation method we apply to factors might slightly disadvantage the DFM approach, since part of the most recent information is lost in the process. These three limits should be addressed by future empirical studies.

Figure 3: DMA vs DMS recursive forecasts based on dynamic factors with $h = 1$





Conclusion

Forecasters build their predictions from growing but imperfect datasets. The challenges they face include handling series with a relatively small number of observations, mixed frequencies, and missing values. Many modeling options have been proposed in the existing literature, but few are flexible enough to address these issues all together.

Dynamic factor models are convenient since they can be adapted to handle missing values and mixed frequency datasets. Indeed, the methodology of (Bessec and Doz, 2014) based on the Kalman filter makes it possible to estimate factors using the whole information set, even in the case of missing values. In practice, this means that forecasters are able to estimate factors even when predictors cover different time-spans, a useful feature since some series are released at the end of the current month (for instance INSEE's monthly surveys), while others are published with a two or three month lag. In the specific case of GDP forecasts, this gain on the factor estimation is partially lost during our forecasting exercise since we need to aggregate monthly factors to quarterly observations in order to estimate the DFM regression equation (11).

Comparatively, Bayesian VARs require a very clean dataset, which most of the time isn't available. As underlined in table 1, the vast majority of our predictors are monthly time series while the predicted outcome is released on a quarterly basis. A standard VAR approach on such data structure requires to aggregate monthly time-series to quarterly frequency, so that we loose a valuable part of the information set, in particular the most recent monthly observations. Recent research on BVARs brought many modeling options, in particular to perform variable selection on large predictors sets. Most of these methods still require heavy MC-MC computations to derive the posterior predictive densities. Even in the interesting case of the natural conjugate prior, some simulation is required to evaluate the posterior.

DMA and DMS models are quickly estimated and provide good out-of-sample results. Both of them are built on the TVP-VAR framework and thus integrate gradual changes in model parameters and in the model through time-varying probabilities. Thanks to specifications in terms of forgetting factors, the parameter rich TVP-VAR converts in a model with a constrained structure but flexible coefficients and predictors. In fact, assumptions on the structure of the model are so detailed that the estimation procedure amounts to a simple run of the Kalman filter: all parameters are modeled as dynamic processes and have well behaved close form distributions, so that there is no need for optimization as it is commonly the case with likelihood based estimators. Thus, before "fitting" such models, one has to be sure the underlying structure is plausible. The next steps regarding DMA applications deals with the curse of dimensionality: more predictors leads to an explosive number of parameters in TVP-VAR specification, as well as in the number of models to explore. In our approach, we overcame this issue with a simple replacement of the initial set of predictors with dynamic factor estimates. Some work has been recently dedicated to the estimation of DMA models with many predictors - see for instance (Belmonte et al., 2014) or (Onorante and Raftery, 2014). Future theoretical developments and empirical applications might provide better insight with regard to their performances.

Our empirical results plead in favor of DMA to forecast aggregate variables like the GDP. However, it is important to underline the limits of our work. First, we do not conduct any sensitivity analysis with respect to the forgetting factors, and some further work on this matter might be useful to check the robustness of DMA's strength over other methods. Second, we only look at a single evaluation measure, while measures based on the whole posterior predictive distribution might add some information to our interpretation. Finally, the aggregation method we apply to factors might slightly disadvantage the DFM approach, since part of the most recent information is lost in the process. These three limits should be addressed in future empirical studies.

VIII Bibliography

Anderson, T. W. (1958). *An Introduction to Multivariate Statistic Analysis*. Wiley, New York.

Bańbura, M., Giannone, D., and Reichlin, L. (2010). Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92.

Barhoumi, K., Runstled, G., Cristadoro, R., and Reijer, A. (2009). Short-term forecasting of gdp using large monthly data sets: a pseudo real-time forecast exercise. *Journal of forecasting*, 28 (7).

Belmonte, M., Koop, G., and Korobilis, D. (2014). Hierarchical shrinkage in timevarying parameter models. *Journal of Forecasting*, 33(1):80–94.

Bessec, M. and Doz, C. (2011). Prévision de court terme de la croissance du pib français à l'aide de modèles à facteurs dynamiques. *Documents de travail de la DG Trésor*.

Bessec, M. and Doz, C. (2014). Short-term forecasting of french gdp growth using dynamic factor models. *OECD Journal: Journal of Business Cycle Measurement and Analysis*, 2013(2):11–50.

Bork, L. and Møller, S. V. (2015). Forecasting house prices in the 50 states using dynamic model averaging and dynamic model selection. *International Journal of Forecasting*, 31(1):63–78.

Bortoli, C. and Combes, S. (2015). Apports de google trends pour prévoir la conjoncture française : des pistes limitées. *Note de conjoncture de l'INSEE, mars 2015*.

De Mol, C., Giannone, D., and Reichlin, L. (2008). Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2):318–328.

Doan, T., Litterman, R., and Sims, C. (1984). Forecasting and conditional projection using realistic prior distributions. *Econometric reviews*, 3(1):1–100.

Doz, C., Giannone, D., and Reichlin, L. (2011). A two-step estimator for large approximate dynamic factor models based on kalman filtering. *Journal of Econometrics*, 164(1):188–205.

Gefang, D. (2014). Bayesian doubly adaptive elastic-net lasso for var shrinkage. *International Journal of Forecasting*, 30(1):1–11.

George, E. I., Sun, D., and Ni, S. (2008). Bayesian stochastic search for var model restrictions. *Journal of Econometrics*, 142(1):553–580.

Geweke, J. (1977). kthe dynamic factor analysis of economic time series, l in latent variables in socio economic models, edited by d. aigner and a. gold" berger.

Hastie, T. and Tibshirani, R. (2001). *The elements of statistical learning: data mining, inference and prediction*. Springer.

Jochmann, M., Koop, G., and Strachan, R. W. (2010). Bayesian forecasting using stochastic search variable selection in a var subject to breaks. *International Journal of Forecasting*, 26(2):326–347.

Koop, G. and Korobilis, D. (2012). Forecasting inflation using dynamic model averaging. *International Economic Review*, 53(3):867–886.

Koop, G. and Korobilis, D. (2013). Large time-varying parameter vars. *Journal of Econometrics*.

Koop, G. M. (2013). Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*, 28(2):177–203.

Korobilis, D. (2013). Var forecasting using bayesian variable selection. *Journal of Applied Econometrics*, 28(2):204–230.

Kydland, F. E. and Prescott, E. C. (1982). Time to build and aggregate fluctuations. *Econometrica: Journal of the Econometric Society*, pages 1345–1370.

Litterman, R. B. (1986). Forecasting with bayesian vector autoregressions—five years of experience. *Journal of Business & Economic Statistics*, 4(1):25–38.

Onorante, L. and Raftery, A. E. (2014). Dynamic model averaging in large model spaces using dynamic occam's window. *arXiv preprint arXiv:1410.7799*.

Raftery, A. E., Kárnỳ, M., and Ettler, P. (2010). Online prediction under model uncertainty via dynamic model averaging: Application to a cold rolling mill. *Technometrics*, 52(1):52–66.

Sargent, T. J., Sims, C. A., et al. (1977). Business cycle modeling without pretending to have too much a priori economic theory. *New methods in business cycle research*, 1:145–168.

Scott, S. L. and Varian, H. R. (2014). Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1):4–23.

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pages 1–48.

Smets, F. and Wouters, R. (2005). Comparing shocks and frictions in us and euro area business cycles: a bayesian dsge approach. *Journal of Applied Econometrics*, 20(2):161–183.

Stock, J. H. and Watson, M. W. (2006). Forecasting with many predictors. *Handbook of economic forecasting*, 1:515–554.

Stock, J. H. and Watson, M. W. (2011). Dynamic factor models. *Oxford Handbook of Economic Forecasting*, 1:35–59.

Varian, H. R. (2014). Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, pages 3–27.

A R code

All the figures presented in this thesis have been computed using the R programming language. The source files can be found at https://github.com/nsaleille. We include the most interesting functions we developed in the following subsections.

A.1 Dynamic factor model

```
dynamicFactorsEstimates <- function(X, q, s, lag.max = 5){

    # X_t = lambda * F_t + epsilon_t
    # F_t = A F_{t-1} + xi_t
    #
    # F_t = [f_t', ..., f_{t-s+1}']'
    # f_t is a (q x 1) vector
    # F_t is a (s * q) vector
    #
    # q is the number of static factors
    # s is the number of dynamic factors
    #
    # var(espilon_t) = diag(phi_1, ..., phi_n) = Phi
    # var(xi_t) = diag(sigma_1, ..., sigma_(p*q))
    # xi_t = P nu_t
    # where var(nu_t) = I_(q x s), var(xi_t) = P %*% t(P) = Sigma

    require(vars)
    require(zoo)
    require(FKF)

    X.omit <- na.omit(X)
    print(paste(nrow(X) - nrow(X.omit), 'observations dropped to estimate factors'))

    ##############################################################
    ################  FACTOR ESTIMATES (Step 1)  #################
    ##############################################################

    T <- dim(X.omit)[1]; N <- dim(X.omit)[2]
    S <- (1/T) * Reduce("+", lapply(as.data.frame(t(X.omit)), function(x){x%*%t(x)}))
    D <- diag(eigen(S)$values[1:q]) # (q x q)
    P <- eigen(S)$vectors[, 1:q] # (N x q)
    f <- t(solve(D)^{1/2} %*% t(P) %*% t(X.omit))
    f <- zoo(f, order.by = index(X.omit))

    ## model parameter estimates - measurement equation

    # matrix lambda
    lambda.0 <- P %*% D^{1/2}
    F <- bindLags(f, s, na.omit = TRUE, bind.original = TRUE)
    regs <- lapply(X.omit, function(x) lm(x ~ -1 + ., data = merge(x, F)))
    lambda <- t(sapply(regs, coefficients))

    # covariance of residuals
    epsilon <- X.omit - zoo(F %*% t(lambda), index(F))
    Phi <- unlist(apply(epsilon, MARGIN = 2, var))

    ## model parameter estimates - state equation

    # Matrix A
    p <- VARselect(as.data.frame(f), lag.max = lag.max, type= "none")$selection['AIC(n)']
    f.var <- VAR(as.data.frame(f), p = p, type = "none")
```

```
    A <- t(sapply(f.var$varresult, function(x) coefficients(x)))
    A <- cbind(A, matrix(0, q, (s - p + 1) * q))
    A <- rbind(A, cbind(diag(s) %x% diag(q), matrix(0, s * q, q)))

    # Matrix P = chol(Sigma)
    xi <- F - zoo(t(A %*% t(lag(F, -1))), order.by = index(F))
    Sigma <- diag(diag(cov(xi)))
    P <- solve(Sigma)^{1/2}

    ############################################################
    ################ FACTOR ESTIMATES (Step 2) ##########
    ############################################################

    ## covariance of the measure equation
    # set to +\infty when the outcome is not observed
    covs <- array(dim = c(N, N, nrow(X)))
    for (t in 1:nrow(X)){
      covs[,,t] <- measure.cov(X[t, ], Phi)
    }

    ## Kalman filter to get better estimates of factors
    # using the previously estimated parameters
    X.kalman <- fkf(
        a0 = as.vector(F[1,]), # ???? NA dans F
        P0 = diag((s+1)*q),
        dt = matrix(0, (s+1)*q, 1),
        ct = matrix(0, N, 1),
        Tt = array(A, dim = c(dim(A), 1)),
        Zt = array(lambda, dim = c(dim(lambda), 1)),
        HHt = array(P, dim = c(dim(P), 1)),
        GGt = covs,
        yt = t(as.matrix(X))
        )

    F.kalman <- zoo(t(X.kalman$att), order.by = index(X))
    f.kalman <- F.kalman[,1:q] # remove f_{t-1} from what we call f.kalman
    X.hat <- zoo(t(lambda %*% t(F.kalman)), order.by = index(X))

    return(list(f = f, f.kalman = f.kalman, lambda = lambda, X.hat = X.hat))

}
```

A.2 Dynamic model averaging

```
dma_estimates <- function(y, X, lambda, alpha, kappa){

  y.train <- na.omit(y)

## construction of the model space

  T <- dim.zoos(X)[1]
  m <- dim.zoos(X)[2]
  d <- dim.zoos(y.train)[1]
  Ty <- dim.zoos(y.train)[2]

  models <- lapply(1:m, FUN = combn, x = m, simplify = FALSE)
  models <- unlist(models, recursive = FALSE)
  models <- models[sapply(models, function(x) 1 %in% x)] # delete models without intercept
  models <- lapply(models, FUN = function(x){return(list(name = paste('model', paste(x,
      collapse=''), sep = '_'), vars = x))})
```

```
  model_names <- sapply(models, FUN = function(x){return(x$name)})
  vars <- sapply(models, function(x) x$vars)
  K <- length(models)

  print(paste('DMA with', m, 'predictors and ', K, 'models'))
  print(paste('Training set size :', Ty, 'observations'))

## priors

  # model probabilities : uninformative uniform prior
  prob <- matrix(nrow = T, ncol = K)
  colnames(prob) <- model_names
  prob[1, ] <- rep(1/K, K) # uniform prior on models

  # parameter theta : OLS estimator
  regs.ols <- lapply(models, function(model) lm(y.train ~ -1 + ., data = merge(y.train, X
      [,model$vars])))
  theta_init <- lapply(regs.ols, coefficients)
  theta <- lapply(models, FUN = function(x){return(matrix(nrow = T, ncol = length(x$vars))
      )})
  theta <- mapply(function(x,y) {x[1,] <- y; return(x)}, theta, theta_init)
  names(theta) <- model_names

  # measurement equation covariance matrix : homscedastic OLS estimator
  H_init <- lapply(regs.ols, function(x) (1 / (T - m - 1)) * sum(residuals(x)^2))
  names(H_init) <- model_names

  # covariance of the parameter: homoscedatic OLS estimator
  sigma_init <- mapply(function(h, model) solve(t(X[,model$vars])%*%X[,model$vars]) * h, H
      _init, models)

## initialisation

  sigma <- sigma_init
  H <- H_init

## compute estimates recursively
  print(paste('Kalman filter on', Ty, 'observations: Start...'))
  for (t in 2:Ty){

    # apply predict_update over model space
    updates <- lapply(models, FUN = predict_update, t = t, outcome = y, predictors = X,
                      theta = theta, sigma = sigma, H = H, prob_last = prob[t-1,],
                      lambda = lambda, alpha = alpha, kappa = kappa)
    # update theta
    theta_up <- sapply(updates, FUN = function(x){return(x$theta_up)})
    theta <- mapply(FUN = function(x,y){x[t,] <- y; return(x)}, theta, theta_up)

    # update probabilities
    weights_up <- sapply(updates, FUN = function(x){return(x$weight)})
    prob[t,] <- weights_up / sum(weights_up)

    # update covariance matrices
    H <- lapply(updates, FUN = function(x){return(x$H)}) # prediction covariance matrix
    sigma <- sapply(updates, FUN = function(x){return(x$sigma)}) # parameter covariance
        matrix
    names(H) <- model_names; names(sigma) <- model_names


  }
  print('Done')
```

```
# wrap up results nicely

  theta <- lapply(theta, zoo, order.by = index(X))
  prob <- zoo(prob, order.by = index(X))

  return(list(parameters = theta, sigma = sigma, probs = prob, models = models))

}
```

```
predict_update <- function(t, model, outcome, predictors, theta, sigma, H, prob_last,
    lambda, alpha, kappa){

  # select model
  y_current <- outcome[t]
  X_current <- predictors[t, model$vars]
  theta_last <- theta[[model$name]][t-1,]
  H_last <- H[[model$name]]
  sigma_last <- sigma[[model$name]]

  # predict for one model / one step ahead
  theta_pred <- theta_last # F: identity
  S_pred <- (1/lambda) * sigma_last # predicted variance for (theta | y^t)
  prob_pred <- prob_last^alpha / sum(prob_last) # predicted probability for model k
  y_pred <- X_current %*% theta_pred # forecast

  # Update for one model / one step ahead
  error <- y_current - y_pred # forecast error
  xSx <- X_current %*% S_pred %*% t(X_current)
  H_up <- kappa * H_last + (1-kappa) * t(error) %*% error # predicted variance for (y_t |
      y^t-1)
  F_inv <- solve(H_up + xSx)
  theta_up <- theta_pred + S_pred %*% t(X_current) %*% F_inv %*% (y_current - X_current
      %*% theta_pred)
  sigma_up <- S_pred - S_pred %*% t(X_current) %*% F_inv %*% X_current %*% S_pred
  weight <- pnorm(q = y_current, mean = X_current %*% theta_pred, sd = sqrt(H_up + xSx)) *
       prob_pred[model$name]

  return(list(weight = weight, theta_up = theta_up, sigma_up = sigma_up, y_pred = y_pred,
              H = H_up, sigma = sigma_up))
}
```