TI 2022-053/III
Tinbergen Institute Discussion Paper

A Flexible Predictive Density Combination for Large Financial Data Sets in Regular and Crisis Periods

Roberto Casarin[1]
Stefano Grassi[2]
Francesco Ravazzolo[3]
Herman K. van Dijk[4]

1 University Ca' Foscari of Venice

2 University of Rome Tor Vergata

3 BI Norwegian Business School, Free University of Bozen-Bolzano and RCEA

4 Erasmus University Rotterdam, Tinbergen Institute and Norges Bank

A Flexible Predictive Density Combination for Large Financial Data Sets in Regular and Crisis Periods*

Roberto Casarin† Stefano Grassi‡
Francesco Ravazzolo§ Herman K. van Dijk¶

†University Ca' Foscari of Venice
‡University of Rome Tor Vergata
§BI Norwegian Business School, Free University of Bozen-Bolzano and RCEA
¶Erasmus University Rotterdam, Tinbergen Institute and Norges Bank

July 14, 2022

Abstract

A flexible predictive density combination is introduced for large financial data sets which allows for model set incompleteness. Dimension reduction procedures that include learning allocate the large sets of predictive densities and combination weights to relatively small subsets. Given the representation of the probability model in extended nonlinear state-space form, efficient simulation-based Bayesian inference is proposed using parallel dynamic clustering as well as nonlinear filtering, implemented on graphics processing units. The approach is applied to combine predictive densities based on a large number of individual US stock returns of daily observations over a period that includes the Covid-19 crisis period. Evidence on dynamic cluster composition, weight patterns and model set incompleteness gives valuable signals for improved modelling. This enables higher predictive accuracy and better assessment of uncertainty and risk for investment fund management.

JEL codes: C11, C15, C53, E37.
Keywords: Density Combination, Large Set of Predictive Densities, Dynamic Factor Models, Nonlinear state-space, Bayesian Inference.

*This paper should not be reported as representing the views of Norges Bank. The views expressed are those of the authors and do not necessarily reflect those of Norges Bank. Paper was presented at the ISBA2022 World Meeting. The authors are indebted to Mike West, James Mitchell and several participants for very useful discussions and comments. We are also indebted to Jamie Cross, Lennart Hoogerheide and, in particular, to the editor, Torben Andersen, and three anonymous referees for valuable comments on earlier versions of this paper, Casarin et al. (2020) which led to a substantial revision and extension. Roberto Casarin's research is supported by the Venice center of Economic and Risk Analytics. Stefano Grassi gratefully acknowledges financial support from the University of Rome 'Tor Vergata' under Grant "Beyond Borders" (CUP: E84I20000900005). Roberto Casarin and Francesco Ravazzolo acknowledge financial support from Italian Ministry MIUR under the PRIN project Hi-Di NET - Econometric Analysis of High Dimensional Models with Network Structures in Macroeconomics and Finance (grant 2017TA7TYC).

1 Introduction

Predicting with large sets of data involving many model structures and explanatory variables is a topic of substantial interest to academic researchers as well as to professional forecasters. It has been studied in several papers (e.g., see Stock and Watson, 1999, 2002, 2005, 2014, and Bańbura et al., 2010). The recent fast growth in big data allows researchers to predict variables of interest more accurately (e.g., see Choi and Varian, 2012; Varian, 2014; Varian and Scott, 2014; Einav and Levin, 2014). Stock and Watson (2005, 2014), Bańbura et al. (2010) and Koop and Korobilis (2013) suggest that there are also potential gains from predicting using a large set of predictors. However, such predictions require new modelling strategies, efficient inference methods and extra computing power possibly resulting from parallel computing. We refer to Granger (1998) for an early discussion of these issues.

Given a large financial micro-data set consisting of US individual stock returns of daily observations over an extended period which includes the Covid-19 crisis period, we propose a flexible predictive density combination in order to approximate such data accurately while allowing for model set incompleteness and combination weight learning. A major motivation for our approach is that in portfolio analysis the process of collecting such a large data set and clustering it in a relatively small number of groups where each group has typical data characteristics is a popular strategy. Next, one replicates an aggregate index through a weighted combination of assets which can be used for a predictive asset allocation strategy in investment funds (e.g., see Corielli and Marcellino, 2006; Kim and Kim, 2020). Our approach provides tools for the quantification of predictive uncertainty and risk for such funds which may be useful management information. A second related motivation is the appearance of fat tail behaviour (and possibly skewness) in most financial data distributions. These data features are usually ignored in point prediction and this may give wrong signals about financial uncertainty and risk.

In terms of methodology, we extend the combination strategies of Billio et al. (2013) and McAlinn and West (2019). McAlinn and West (2019) is founded on a decision theory framework providing a coherent Bayesian interpretation for Bayesian Model Averaging (BMA). However, when the time series tend to become large BMA tends to select one model, supposedly the true model, see Amisano and Geweke (2010). In empirical analysis it is, however, known that the concept of a true model is not very realistic, see Geweke (2010). Our aim is not to pursue the approach of finding the true model, but rather to introduce a flexible predictive density combination that provides an accurate approximation to nonstandard data distributions in finance such as bimodal ones and/or distributions with heavy tails. We refer to Hall and Mitchell (2007); Amisano and Geweke (2010); Billio et al. (2013); Gneiting and Ranjan (2013); Yao et al. (2018) and the discussion therein for further insight on the different combination approaches.

Our extensions to Billio et al. (2013) are threefold. First, replace their normal combination model with a flexible mixture of predictive distributions to account for possible multimodality and heavy tails and specify a measure for

model incompleteness or misspecification. Second, make the combination approach operational for large data sets by applying dimension reduction which includes learning about the large sets of predictive densities as well as the combination weights. A major reason for these different dimension reductions is that investment fund managers are interested to obtain not only predictions of the fund but also to learn about the composition over time of the clusters of assets and, also, to learn about the behaviour of weights of these clusters. This may lead to improved modelling, prediction and policy. Third, for efficient inferential purposes make use of a recent parallel Sequential Monte Carlo method.

Extension one is described in the beginning of Section 2, here we explain extensions two and three. Our empirical example contains a large data set of individual US stock returns of daily observations over an extended time period which includes the Covid-19 crisis. As a consequence, we deal with a large set of predictive densities in the combination process, which makes the inference task a substantial challenge. Inference based on the normal combination model from Billio et al. (2013) is not operational in large panels since the latent space of combination weights is high dimensional and overparameterization and overfitting issues can easily arise. We extend Billio et al. (2013) with two dimension reductions. The first one is based on dynamic clustering of the large set of predictive densities exploiting common data features in the large set of stock return series such as wide data bands and time-varying volatility. The dynamic clustering maps, at each time t, the large set of predictive densities to a much smaller subset where the cluster composition at time t depends on the past composition. We note that clustering strategies have been successfully used in other models to cope with high dimensional parameter spaces, (e.g., see MacLehose and Dunson, 2010; Billio et al., 2019). The second dimension reduction deals with the large number of combination weights. A nonlinear dynamic factor model is specified for these weights which contains learning, possibly, using information about past predictive performance. An alternative is to shrink the combination weights to zero as in the sparse factor model literature, (e.g., see Carvalho et al., 2008; Kaufmann and Schumacher, 2017, 2019) but learning is not included. We note that our approach contributes to the literature on time series models with time-varying parameters that take values on a bounded domain, see, e.g., Aitchinson and Shen (1980) and Aitchinson (1982), and applies it to large financial data extending the intuition in Stock and Watson (2014).

With respect to the inference method we show that our mixture model allows for an extended nonlinear state-space representation. This enables us to construct an efficient simulation-based Bayesian inference procedure, where parallel Sequential Monte Carlo is used to filter the set of probabilistic weights and integrate the set of random parameters. Here we make use of the recently developed M-Filter, see Baştürk et al. (2019) and Hoogerheide et al. (2012).

In terms of empirical analysis we provide three contributions: accuracy gains in predictive moments compared to benchmark results; time-varying composition of clusters and cluster weights; diagnostic information on model set incompleteness.

Evidence on substantial accuracy gains in predictive means, volatilities and tail events is presented compared to the no-predictive ability benchmark and predictions from individual models and combination methods as BMA and Equal Weights (EW). The time-varying composition of the set of clusters shows learning. Individual stocks may switch across clusters or eventually exit them, for example, during and after a crisis like the financial crisis and the Covid-19 crisis. Measures of model set incompleteness and dynamic patterns in the cluster-based weights give valuable diagnostic signals. These empirical results may provide useful information for improved financial modelling and policy analysis by investment fund management.

For a recent survey about the evolution of predictive density combinations we refer to Aastveit et al. (2019), for background to Billio et al. (2013), McAlinn and West (2019) and for a policy application to Baştürk et al. (2019).

The contents of this paper are structured as follows. Section 2 provides details of the methodological contributions of our approach. Section 3 contains an efficient simulation-based Bayesian inference procedure. Section 4 contains results of the empirical application using a large set of US stock return data in regular and crisis periods. Section 5 presents conclusions and suggestions for further research. Some additional results are given in Supplementary Material that serves as an online Appendix.

2 Mixture process with model set incompleteness, dimension reduction and time-varying component weights

We start with extending a standard mixture process for predictive densities to include model set incompleteness. Let the conditional predictive probability distribution of a financial variable of interest, y_t, be specified as a mixture of conditional predictive probability distributions of y_t coming from a large set of n individual financial models with information sets $\mathfrak{I}_{i,t-1}$, where the information set $\mathfrak{I}_{i,t-1}$ includes data information as well as model structure, denoted by M_i, with $i = 1, \ldots, n$. Define weights w_{it} that form a convex combination of the conditional predictive probabilities. In terms of densities this implies a standard mixture process:

$$f(y_t|\mathfrak{I}_{t-1}) = \sum_{i=1}^{n} w_{it} f(y_t|\mathfrak{I}_{i,t-1}), \quad 0 \leq w_{it} \leq 1, \quad \sum_{i=1}^{n} w_{it} = 1, \tag{1}$$

where $\mathfrak{I}_{t-1}$ is the joint information set.

A key step is to give specific content to the i-th mixture component $f(y_t|\mathfrak{I}_{i,t-1})$. Let $y_t = y_{it}^*$ with probability w_{it}, where y_{it}^* is defined for all models M_i as the sum of the following two random variables:

$$y_{it}^* = \tilde{y}_{it} + \varepsilon_{it}, \tag{2}$$

where $\tilde{y}_{it}$ is a generated draw from the predictive distribution with density $f\left(\tilde{y}_{it}|\mathfrak{I}_{i,t-1}\right)$ from model M_i. A new feature is the addition of the disturbance ε_{it}. It points towards two sources of error. There may be misspecification errors due to model set incompleteness and prediction errors due to, for instance, sudden shocks in the series. In this paper we focus on the former, that is, a larger specification error implies a larger error ε_{it}. Investigating the relative importance of a prediction error component is a topic for further research. We note that Terui and van Dijk (2002); Hoogerheide et al. (2010); Takanashi and McAlinn (2019), McAlinn and West (2019) and recently Aastveit et al. (2022) make use of a combination equation which can be interpreted as a linear regression model with time-varying parameters and a time-varying constant and one disturbance term. In contrast, we work with a flexible mixture approach in the combination process where for each component of the mixture there exists a time-varying weight and a disturbance.

The probability density function of ε_{it} is given as:

$$\varepsilon_{it} \sim \mathcal{N}\left(0, \sigma_{it}^2\right), \tag{3}$$

where

$$\sigma_{it}^2 = \sigma_i^2 \exp(h_{it}), \quad h_{it} = h_{i,t-1} + \zeta_{it}, \quad \zeta_{it} \sim \mathcal{N}(0, \sigma_{\zeta,i}^2), \tag{4}$$

for each $i = 1, \ldots, n$. Given equation (2), the probability density function of y_{it}^* is the convolution of two densities given as:

$$f(y_{it}^*|\mathfrak{I}_{i,t-1}, \sigma_{it}^2) = \int_{\mathbb{R}} \frac{1}{\sigma_{it}} \phi\left(\frac{y_{it}^* - \tilde{y}_{it}}{\sigma_{it}}\right) f(\tilde{y}_{it}|\mathfrak{I}_{i,t-1}) d\tilde{y}_{it} \tag{5}$$

where $\phi(\cdot)$ is the standard normal density. So, for our mixture of n models we have

$$f(y_t|\mathfrak{I}_{t-1}, \sigma_t^2) \quad = \quad \sum_{i=1}^{n} w_{it} \int_{\mathbb{R}} \frac{1}{\sigma_{it}} \phi\left(\frac{y_t - \tilde{y}_{it}}{\sigma_{it}}\right) f(\tilde{y}_{it}|\mathfrak{I}_{i,t-1}) d\tilde{y}_{it} \tag{6}$$

where $\sigma_t^2 = \{\sigma_{1t}^2, \ldots, \sigma_{nt}^2\}$. One may interpret the information from the predictive densities $f\left(\tilde{y}_{it}|\mathfrak{I}_{i,t-1}\right)$, $i = 1, \ldots, n$ as *prior-predictive* information that is fed into the predictive density combination (6).

However, a large set of predictive densities for each time observation allowing for time-varying combination weights is not easy to handle in terms of econometric inference. Reduction of the large information set is necessary. Dimension reduction techniques are widely used in machine learning to reduce the information of high-dimensional datasets (see e.g., Varian, 2014; Casarin and Veggente, 2020, and references cited therein). We make use of dimension reduction steps for the large number of components in the predictive mixture and the accompanying large number of latent probabilistic weights. The different steps are schematically shown in Table 1.

In our financial case, we start with a preliminary step using diagnostic graphical evidence about typical data features in individual financial series of stock returns like high and low time-varying volatility and wide and narrow data bands. Our motivation for this is that in financial applications large differences across

predictions occur in the higher moments and tail behaviour. This leads in our case to the, *a priori*, choice of a Normal density with high and low volatility and a Student-t density with large and small degrees of freedom. We use these data features for dimension reduction of the large number of components in the predictive mixture into four groups using dynamic K-means clustering where the current cluster composition depends on the past composition. This allows for learning about model dependence and cluster grouping, see e.g. Varian (2014); Casarin and Veggente (2020), and for more general applications of machine learning methods to financial predictions (Gu et al., 2020; Bianchi et al., 2020). This is well documented in data but largely ignored in the predictive density combination literature. We note that Bianchi and McAlinn (2018) and Takanashi and McAlinn (2019) also discuss clustering procedures. Usually, these methods make use of constant clustering. The clustering process is depicted in the second row of Table 1 and empirical results are presented in Section 4. For general background on K-Means clustering we refer to (Frühwirth-Schnatter, 2006, pp. 97) and Malsiner Walli et al. (2016) and for details about the implementation of our dynamic clustering procedure we refer to Appendix B.1 in the Supplementary Material.

Large set of financial series leading to n predictive densities
$\searrow \quad \downarrow \quad \downarrow \quad \swarrow$
Dynamic clustering allocates n densities to $m \ll n$ groups
$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
Cluster 1 Cluster 2 $\cdots$ Cluster m
Cluster weights z_{jt} are logistic transformations of unrestricted random walk variables v_{jt}:
$(z_{1t}, \ldots, z_{mt}) = g(v_{1t}, \ldots, v_{mt})$
$\swarrow \quad \downarrow \quad \downarrow \quad \searrow$
Mapping m cluster weights z_{jt} to n model weights w_{it}
$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
Construction of convex combination of large set of n predictive densities

Table 1: Dimension reduction and computation of latent weights for predictive density combination using a large financial data set. The function $g(\cdot)$ refers to the logistic transformation given in equation (8).

In order to complete the specification of the probability model given in equation (6), we specify a law of motion, which involves dimension reduction and learning, for the latent probabilistic weights w_{it}, $i = 1, \ldots, n$, $t = 1, \ldots, T$ with large n and large T. The weight model, given in equations (7)-(9), is a nonlinear dynamic factor model of w_{it}. Factor models are well-known as a vehicle for dimension reduction, see Anderson (1984) and Lopes and West (2004) and we make use of a simple nonlinear extension. Let

$$v_{jt} = v_{j,t-1} + \eta_{jt}, \quad \eta_{jt} \sim \mathcal{N}(0, \sigma_\eta^2), \quad j = 1, \ldots, m, \tag{7}$$

$$z_{jt} = \frac{\exp(v_{jt})}{\sum_{i=1}^{m} \exp(v_{it})}, \quad 0 \leq z_{jt} \leq 1, \quad \sum_{j=1}^{m} z_{jt} = 1 \tag{8}$$

$$w_{it} = \sum_{j=1}^{m} b_{ijt} z_{jt}, \qquad i = 1, \ldots, n. \tag{9}$$

The unrestricted latent variables v_{jt}, $j = 1, \ldots, m$ in the $m = 4$ clusters are normal random walk variables, see equation (7). This dynamic specification can be generalised, see Billio et al. (2013). Next, a logistic transformation is applied from the unrestricted latent variables $v_{jt}, j = 1, \ldots, m$ to an *auxiliary* set of probabilistic cluster weights $z_{jt}, j = 1, \ldots, m$, see equation (8), which is depicted in the bottom part of Table 1. The purpose of the factor loadings b_{ijt} is to transform the m weights z_{jt} of the convex combination of m clusters to the n weights w_{it} of the convex combination of $n \gg m$ models. Therefore, the factor loadings b_{ijt} are restricted to be nonnegative and sum to 1. We assume in our case, for convenience, that each model i has an equal weight within its cluster j. To allow for learning one may consider a case where the weights are driven by model-specific predictive performance using the log score, see Billio et al. (2013) and also Mitchell and Hall (2005). Note that in our case there exists no noise in the connection between w_{it}, z_{jt} and v_{jt}, but this assumption can be relaxed to account for contemporaneous uncertainty with the addition of logistic-normal noise in the equation for w_{it}. This is left for further research.

Given equations (7)-(9) we can complete the specification of the probability model given in (6) with the specification of the law of motion for the probabilistic weights w_{it}, $i = 1, \ldots, n$. Consider the $m \times 1$ vector $\mathbf{v}_t = (v_{1t}, \ldots, v_{mt})'$ with the multivariate normal distribution $\mathbf{v}_t \sim \mathcal{N}_m(\mathbf{v}_{t-1}, \mathbf{\Sigma}_\eta)$ where $\mathbf{\Sigma}_\eta$ is a diagonal matrix. Given that the $m \times 1$ vector $\mathbf{z}_t = (z_{1t}, \ldots, z_{mt})'$ is given as a logistic transformation, see equation (8), the vector $\tilde{\mathbf{z}}_t = (z_{1t}, \ldots, z_{m-1,t})'$ follows a logistic-normal distribution given as $\tilde{\mathbf{z}}_\mathbf{t} \sim \mathcal{L}_{m-1}(\mathbf{D}\mathbf{v}_{t-1}, \mathbf{D}\mathbf{\Sigma}_\eta\mathbf{D}')$ with $z_{mt} = 1 - \sum_{j=1}^{m-1} z_{jt}$, where the expression of the matrix $\mathbf{D}$ is given in the proof of the result which is presented in the Supplementary Material, Appendix A.

Since the large set of n probabilistic weights w_{it}, $i = 1, \ldots, n$ consists of linear combinations of the small set of m logistic-normal probabilistic weights z_{jt}, $j = 1, \ldots, m$, see equation (9), the distribution of the weights w_{it} is logistic-normal. Here, use is made of the class preserving property of the logistic-normal distribution. In matrix notation, consider the vector $\tilde{\mathbf{w}}_t = (w_{1t}, \ldots, w_{n-1,t})'$ with $w_{nt} = 1 - \sum_{i=1}^{n-1} w_{it}$. Then we make use of $\tilde{\mathbf{w}}_t = \tilde{\mathbf{B}}_t \tilde{\mathbf{z}}_t$ where $\tilde{\mathbf{B}}_t$ is an $(n-1) \times m$ matrix that contains an appropriate subset of elements b_{ijt}. As a consequence, one can write that the implied logistic-normal distribution of $\tilde{\mathbf{w}}_\mathbf{t}$ is given as :

$$\tilde{\mathbf{w}}_t \sim \mathcal{L}_{n-1}\left(\tilde{\mathbf{B}}_t \mathbf{D}\mathbf{v}_{t-1}, \tilde{\mathbf{B}}_t \mathbf{D}\mathbf{\Sigma}_\eta \mathbf{D}'\tilde{\mathbf{B}}_t'\right), \tag{10}$$

for details on this result, see the Supplementary Material, Appendix A. Note that the density of the complete vector $\mathbf{w}_t = (w_{1t}, ..., w_{nt})'$ is singular due to the adding-up restriction of the n weights. A second source of degeneracy is intrinsic to our projection strategy which implies rank deficiency of the matrix $\tilde{\mathbf{B}}_t \mathbf{D}\mathbf{\Sigma}_\eta \mathbf{D}'\tilde{\mathbf{B}}_t'$.[1]

[1] Other distributions can be used for weights such as the Dirichlet distribution, but as shown in Aitchinson and Shen (1980) this distributional assumption can be too restrictive in our analysis

The analytic solution of the probability model of equations (6) and (10) is generally not known but the model can be represented in extended nonlinear state-space form given as:

$$y_t = \sum_{i=1}^{n} (\tilde{y}_{it} + \varepsilon_{it}) s_{it}, \qquad \varepsilon_{it} \sim \mathcal{N}(0, \sigma_{it}^2) \tag{11}$$

$$(s_{1t}, \ldots, s_{nt}) \sim \mathcal{M}_n(1, (w_{1t}, \ldots, w_{nt})), \tag{12}$$

$$w_{it} = \sum_{j=1}^{m} b_{ijt} z_{jt}, \qquad i = 1, \ldots, n, \tag{13}$$

$$z_{jt} = \frac{\exp(v_{jt})}{\sum_{i=1}^{m} \exp(v_{it})}, \qquad j = 1, \ldots, m, \tag{14}$$

$$v_{jt} = v_{j,t-1} + \eta_{jt}, \qquad \eta_{jt} \sim \mathcal{N}(0, \sigma_\eta^2), \qquad j = 1, \ldots, m. \tag{15}$$

The measurement equation (11) has as extension that the right-hand side variable $\tilde{y}_{it}$ is not an observation from our dataset but refers to a random draw from the predictive distribution of model M_i and ε_{it} gives an indication of possible incompleteness. Using the Multinoulli distribution (also known as the Categorical distribution) of $(s_{1t}, \ldots, s_{nt})$ with parameter vector $(w_{1t}, \ldots, w_{nt})$ given in equation (12), one generates draws from the i-th component of the mixture distribution with probability w_{it} where $(s_{1t}, \ldots, s_{nt})$ contains $n-1$ 0's and one element equal to 1. That is, $s_{it} = 1$ means that model i is selected.

A schematic figure of the extended nonlinear state-space representation is given in Figure 1. The sequence of steps starts with the dynamic clustering step shown at the bottom left, then one proceeds upwards and next to the middle center where the nonlinear dynamic factor model is shown which is constructed at the bottom right and going upwards. At the top, the measurement equation is shown from the finite mixture process with the n generated predictive draws and disturbances that follow a stochastic volatility process. The complete state-space representation enables us to make use of filtering methods of the nonlinear time series literature to evaluate and update the unobserved weight components in the predictive density combination.

3 Bayesian inference applying the M-Filter

The analytic solution of the optimal filtering problem is in most applications not known, except for the case of a Kalman filter where use is made of well-known properties of the multivariate normal distribution. For our large finite mixture process with time-varying weights based on a nonlinear dynamic factor model, we make use of simulation-based numerical methods to tackle the filtering problem and

since the components of a Dirichlet composition have a correlation structure determined solely by the normalisation operation, so that only negative correlations are possible (and given the means, there is only one free parameter for the variances and covariances).

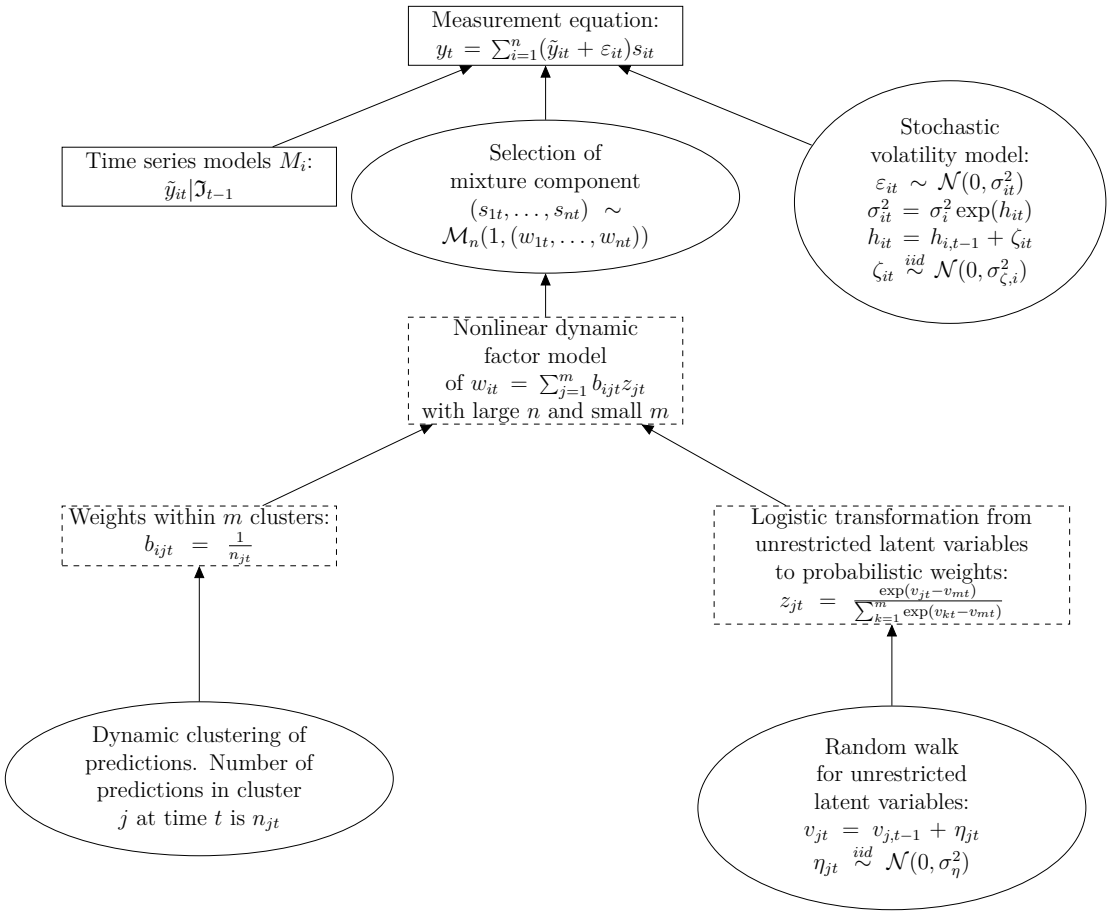

Figure 1: Directed acyclic graph of the extended nonlinear state-space model. It shows the connections between the variable of interest y_t, the predicted variables $\tilde{y}_{it}$ (rectangles, solid line), the dynamic clustering, the random walk process, the categorical Multinoulli distribution, the stochastic volatility specification (ellipses), the link functions $\frac{\exp(v_{jt}-v_{mt})}{\sum_{k=1}^{m}\exp(v_{kt}-v_{mt})}$, b_{ijt} and the nonlinear dynamic factor model of the latent probabilistic weights w_{it} (rectangles, dashed line). The directed arrows show the dependence structure.

we make use of Bayesian inference to update information about the random model parameters. For a fundamental discussion about a coherent Bayesian framework for evaluation, calibration, and data-informed combination of multiple predictive densities, see McAlinn and West (2019) and McAlinn et al. (2020).

Apart from the fundamental motivation for applying Bayesian inference there exists a practical one which is based on simulation-based Bayesian methods. That is, the generated draws $\tilde{y}_{it}$ from the predictive distributions of the different models are carried forward into the predictive combination density. Thus, the uncertainty in the predictions of the different models carries directly forward into the uncertainty of the combined predictive density. In contrast, frequentist methods like method of moments or maximum likelihood proceed in a two-step fashion by first computing point predictions for the different models and substituting these in a second stage into the combined predictive density. As

such the second stage results suffer from the generated regressor problem, that is, uncertainty measures of the second-stage predictions are sensitive to the estimation procedure used for the model predictions, see Pagan (1984) for background details.

In order to filter the latent weights in equations (11)-(14) we make use of the recently developed M-Filter introduced in Baştürk et al. (2019) and based on Hoogerheide et al. (2012). We restrict ourselves to a summary of the novel, efficient and robust properties of this method and refer to the cited references and Appendix B.2 in the Supplementary Material for technical background.

The M-Filter is a member of the class of Sequential Monte Carlo (SMC) algorithms that are suitable to solve filtering problems in nonlinear state-space models, see Creal (2007) and Herbst and Schorfheide (2014) for background. The basic idea of SMC is that a set of draws, labelled particles, is generated from an approximation to the so-called target density, in our case, the combined predictive density, in a sequential way at each time t. That is, such an SMC method consists in general at each time point t of two steps: a sampling step to generate particles from the approximate density and next a correction step to adjust for the distance between the approximate density and the target. However, the propagation of particles over time points leads to the situation that after many iterations one particle receives all weights which implies degeneracy of the approximate density. In order to avoid this, a so-called resampling step is introduced where the approximate density is updated in such a way that degeneracy does not occur.

The M-filter has two innovations. First, this rather cumbersome resampling procedure in the propagation step is replaced by independent sampling at each time t. This also reduces Monte Carlo variation. Second, the independent sampling step occurs in an online-fashion, contrary to other methods that make use of independent sampling like Efficient Importance Sampling of Richard and Zhang (2007) and Liesenfeld and Richard (2003), or Numerically Accelerated Importance Sampling of Koopman et al. (2015) that are off-line methods.

The choice of an accurate approximate density is crucial for the performance of any filter method and has received considerable attention in the SMC literature, see Doucet et al. (2001), Liu (2001), Kunsch (2005) and Creal (2012). The M-Filter method approximates a target density using the *Mixture of t by Importance Sampling Weighted Expectation–Maximization* (MitISEM) algorithm proposed by Hoogerheide et al. (2012) and further developed in Baştürk et al. (2016). MitISEM has been shown to be able to effectively approximate complex, non-elliptical distributions due to two features of this algorithm: the class of importance distributions (mixtures of multivariate Student's t distributions), and their joint optimisation (with the Expectation-Maximisation, EM, algorithm). The former allows to closely track distributions of nonstandard shape (multimodal, skewed). The latter is iteratively carried out with the objective of minimising the Kullback-Leibler (KL) divergence between target density and approximate density. This robustness and flexibility of the M-Filter in constructing approximate densities is particularly important in econometrics where breaks in time series are often observed.

The application of the M-Filter requires to choose the values of a few tuning parameters. First, the number of components of the Student-t mixture to approximate the target density is restricted to be no larger than a maximum of four. For each component the initial mean, variance and degrees of freedom parameter are set to zero, one and three, respectively. Those parameters are then updated with an EM step. Second, the number of draws is fixed to 10000 which is sufficient for good convergence in our case. Third, the measure that is used in order to check the convergence of the algorithm is the relative change in the Coefficient of Variation (CoV) of the Importance Sampling weights, where the Importance Sampling weight is the ratio of target and approximate density. The default convergence criterion is chosen as the change of the CoV being smaller than 0.5%.

The M-Filter is easy to parallelise, this enables our approach to speed up the computations using Multiple CPUs or GPUs. In Table 2 we report results from an experiment about the computing time of the M-Filter with different numbers of threads, numbers of draws and maximum numbers of components of the Student-t mixture. All the values reported are in ratio with the benchmark model given by 5000 draws, one thread and two threads. For a comparison of the computing time between the M-Filter and particle filters we refer to Baştürk et al. (2019), where extensive Monte Carlo studies in benchmark exercises are presented. The results

	1 Thread			*6 Threads*			*12 Threads*			*24 Threads*		
C	5000	10000	20000	5000	10000	20000	5000	10000	20000	5000	10000	20000
2	1.000	1.953	3.773	0.171	0.314	0.725	0.093	0.171	0.377	0.083	0.164	0.325
3	1.067	2.086	3.970	0.178	0.317	0.750	0.098	0.181	0.379	0.083	0.165	0.326
4	1.068	2.097	3.981	0.186	0.322	0.782	0.099	0.185	0.383	0.083	0.168	0.331

Table 2: Computing time for number of threads $(1, 6, 12, 24)$*, number of draws* $(5000, 10000, 20000)$ *and maximum number of Student-t components (C). All values are in ratio with the benchmark model given by 5000 draws and one thread. Values higher that 1 means that the computing time is higher than the benchmark. Values lower than 1 means that the computing time is lower than the benchmark.*

reported in Table 2 show that using multiple threads reduces the computing time considerably in our case. Results are not sensitive to the choice of the maximum number of components in the mixture approximation, while doubling the number of draws also doubles the computing time.

The information on the priors for the SV model in equation (4) and for the random walk in equation (7) is given as follows. For the σ_{it}^2 we select a log-normal distribution with mean $\log(0.1)$ and and standard deviation 0.175; for σ_ζ^2, we also select a log-normal distribution with mean -2.3 and standard deviation 0.175. Our prior corresponds to an average incompleteness value that is equal to 10% of the unconditional volatility of our S&P500 data. For the σ_η^2, in equation (7), we select a log-normal distribution with mean -0.7 and standard deviation

equal to 0.175. Therefore the prior mean for σ_η^2 is 5 times higher than for σ_ζ^2, assuming bigger changes for the cluster weights than for the incompleteness over time. The standard deviation of 0.175 for both priors implies a relative loose prior in both cases. For the parameters in the four equation models we make use of rather uninformative proper priors. All our priors are default choices, that can be modified by the user, in the MATLAB toolbox that carries out the M-Filter, which is available at `http://www.francescoravazzolo.com/pages/DeCo.html`.

4 Predicting and tracking the S&P500

As discussed in the introduction many investors of mutual funds, hedge funds and exchange-traded funds try to replicate the performance of the S&P500 index by holding a set of stocks, which are not necessarily the exact same stocks included in the index.

We collected 496 individual daily stock prices, components of the S&P500, from Datastream over the sample January 2, 2014, to June 30, 2021, for a total of 1888 daily observations for each series. We computed the time series of log-returns for all stocks, see Figure 2. Table 3 reports several cross-section average statistics of the individual series together with the same statistics for S&P500 index. Some series have much lower average returns than the index with volatility up to 3 times higher than the index. Heterogeneity in skewness and kurtosis is also evident with the series with the lowest skewness equal to -1.829 and the highest skewness equal to 0.335 and with the lowest kurtosis equal to 9.135 and the highest kurtosis equal to 42.164.

The inclusion in our series of the Covid-19 pandemic explains such high values.[2] We report results on several features of the combined predictive density of the replication of the S&P500 index, including the economic value of tail events.

Diagnostic determination of four clusters and model estimation. Given the basic statistics about time-series and cross-section patterns, we have determined as typical data features of our financial micro-data set wide and narrow data bands and high and low time-varying volatility.

This led us to specify two clusters of predictive densities based on a Normal GARCH(1,1) model: one cluster with high volatility (labelled $n1$) and one cluster with low volatility (labelled $n2$). Next, two clusters based on a Student-t GARCH(1,1) model: one cluster with low degrees of freedom (labelled $t1$) and one cluster with high degrees of freedom (labelled $t2$).[3] Our motivation for this choice is to obtain a mixture of densities which fits both in the center of the empirical distribution (for mean prediction) and in the tails (for measuring uncertainty and

[2]It has been suggested to make use of the information about shares outstanding to determine better the time behaviour of weights. We consider only stocks that have survived in the S&P500 basket over the period of our sample, but the methodology can handle series with different sample lengths. We leave these as topics for further research.

[3]Low degrees of freedom occur jointly with a large scale and high degrees of freedom occur jointly with a low scale.

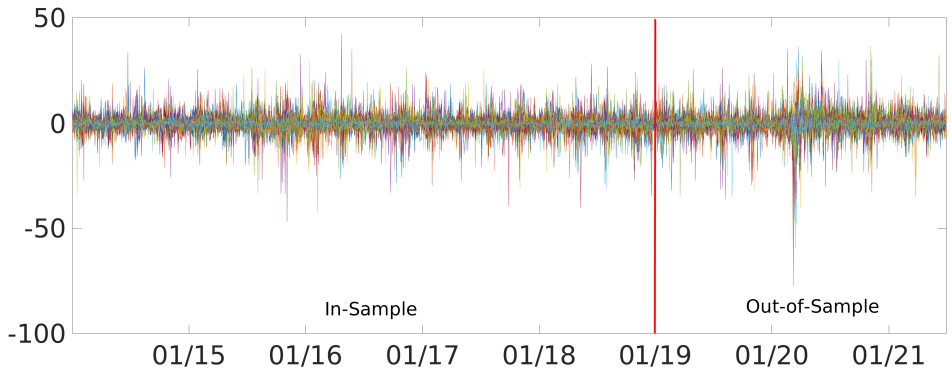

Figure 2: Daily (%) log-returns for 496 individual stock components of the S&P500 over the sample January 2, 2014 to June 30, 2021. The red line on January 2, 2019 indicates the beginning of the out-of-sample period.

risk).

To ease on the computational workload, we have applied an optimisation method to estimate the posterior modes of the parameters from a Normal GARCH(1,1) model and a Student-t GARCH(1,1) model. Given our rather uninformative proper priors, these mode estimates are equal to approximate Bayes mean estimates.

We use rolling samples of 1258 trading days (about five years) for each stock return in the model:

$$y_{it} = c_i + \kappa_{it}\xi_{it} \tag{16}$$

$$\kappa_{it}^2 = \theta_{i0} + \theta_{i1}(y_{i,t-1} - c_i)^2 + \theta_{i2}\kappa_{i,t-1}^2, \qquad i = 1, 2, \ldots, n, \tag{17}$$

where y_{it} is the log return of stock i at day t, $\xi_{it} \sim \mathcal{N}(0,1)$ and $\xi_{it} \sim \mathcal{T}(\nu_i)$ for the Normal and Student-t cases, respectively. The number of degrees of freedom ν_i is estimated in the latter model. We produce 629 one day ahead predictive densities from January 2, 2019 to June 30, 2021, see red line in Figure 2 for the sample split. Our out-of-sample period is associated with relatively low volatility in 2019 and high volatility from the end of February 2020 driven by the Covid-19 crisis. In the initial and most dramatic part of the pandemic, the lowest daily return is almost -80%.

Results are reported in Figure 3. It is seen that the Normal models in cluster $n2$ have a predictive variance (top left plot, dashed line) more than double in size than cluster $n1$ with several spikes over time and this increases further at the beginning of the Covid-19 pandemic. Cluster $n1$ has a relatively constant predicted variance (top left plot, solid line) over the entire period except at the beginning of the Covid-19 pandemic. The Student-t models in cluster $t1$ have a relatively constant thick tail just above 4 over the entire period (top right plot, solid line) while cluster $t2$ has values around 7 for the degrees of freedom (top right plot, dashed line) before the beginning of the Covid-19 which decreases to 6.5 after it. Some instability is also measured in the second wave of the Covid-19 at the end of

	Subcomponents			S&P500
	Lower	Median	Upper	
Average	-0.012	0.051	0.116	0.045
St dev	1.328	1.810	2.997	1.111
Skewness	-1.829	-0.469	0.335	-1.045
Kurtosis	9.135	16.707	42.164	24.640
Min	-34.907	-17.013	-10.182	-12.765
Max	9.551	14.662	26.722	8.968

Table 3: Average cross-section statistics for the 500 daily log-returns of individual stocks for the sample January 2, 2014 to June 30, 2021. The columns "Lower", "Median" and "Upper" refer to the cross-section 5% lower quantile, median and 95% upper quantile of the 500 statistics in rows, respectively. The rows "Average", "St dev", "Skewness", "Kurtosis", "Min" and "Max" refer to sample average, sample standard deviation, sample skewness, sample kurtosis, sample minimum and sample maximum statistics, respectively. The column "S&P500" reports the sample statistics for the log-returns of the aggregate S&P500 index.

2020 and beginning of 2021.

Time-varying cluster composition and weights. The clustering of the predictive densities is repeated at every time t and therefore the cluster composition and weights vary over time. The middle and bottom panels in Figure 3 present results about these features. The number of stocks varies across consecutive vintages of predictions with the clusters $n1$ and $t1$ being dominant in terms of attracting many individual stocks. The cluster $n1$ contains on average 400 stocks, whereas cluster $n2$ has the remaining 100. The Covid-19 pandemic creates some instability in the stock allocation, but similar patterns existed also in 2019. The cluster $t1$ includes on average 450 stocks, whereas cluster $t2$ has on average 50 stocks. The beginning of the Covid-19 pandemic is associated with a 5% increment in the allocation of stocks to cluster $t1$; the second wave at the end of 2020 and early months of 2021 produces some changes in the stock allocation.

Plots of the estimated cluster weights are shown at the bottom left panel in Figure 3. These weights are the average weights of w_{it} per cluster. In terms of the importance of the different clusters, we notice that the clusters $n1$ and $t1$ receive large part of the weight over the full sample, see the bottom panel in Figure 3. At the end of the period, they sum almost to 70% of the total weight. Therefore, the two clusters include most of the stocks and give the larger contribution to the combination. However, there is also evidence of time variations in the weights. The weights of clusters $n2$ and $t2$ are larger at the beginning of sample, but their size reduces from 50% to 30% just before the Covid-19 pandemic. The crisis increases substantially their weights, in particular for $n2$ when volatility increases. The weights of clusters $n2$ and $t2$ reduce after the first part of the Covid-19 period and in the final part of the sample their contribution is similar to the pre-Covid-19 period.

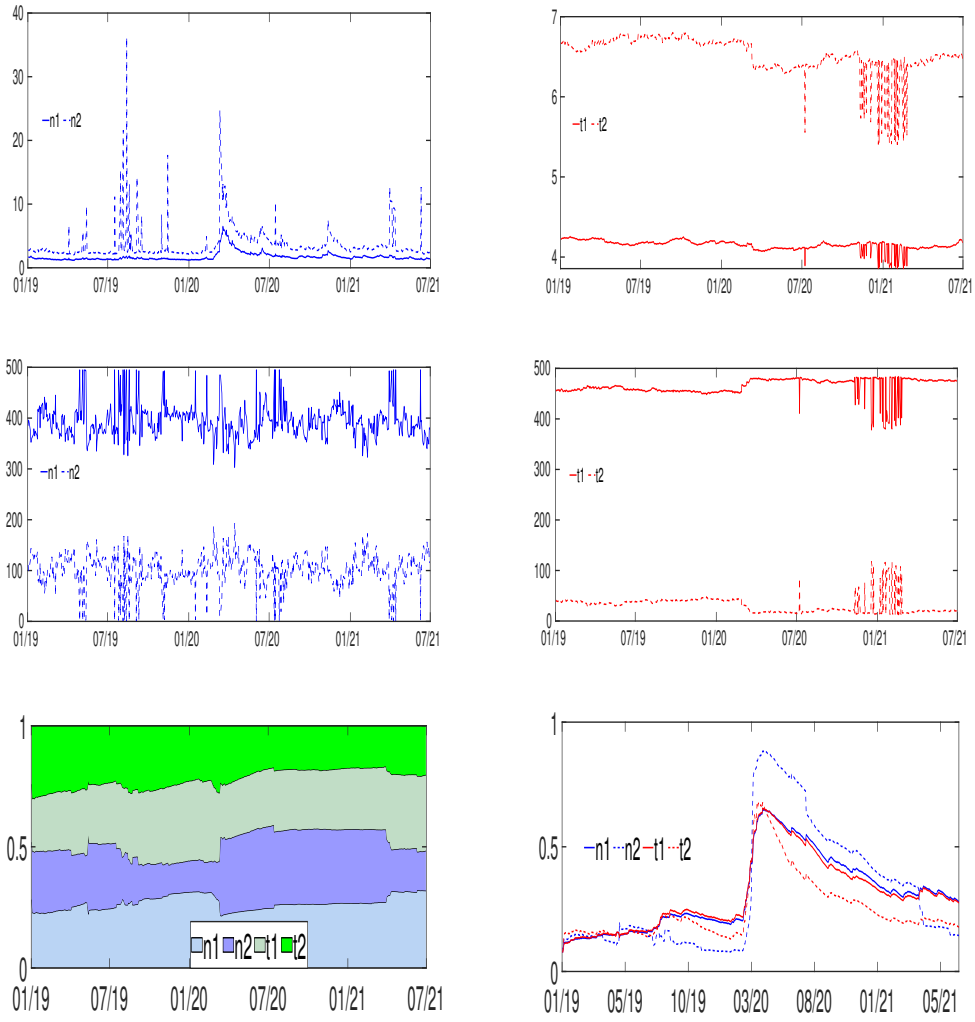

Figure 3: Top left: the average variance of the predictions from the $n1$ (solid blue line) and $n2$ (dashed blue line) clusters. Top right: the average degree of freedom of the predictions from the $t1$ (solid red line) and $t2$ (dashed red line) clusters. The degrees of freedom are bounded to 30. Middle left: cluster allocation over time between the two clusters for the Normal case: $n1$ (solid blue line) and $n2$ (dashed blue line). Middle right: cluster allocation over time between the two clusters for the Student's t case: $t1$ (solid red line) and $t2$ (dashed red line). Bottom left: the mean logistic-normal weights for the $n1$, $n2$, $t1$ and $t2$ clusters. Bottom right: posterior mean estimates of the incompleteness measures in the four clusters.

Measures of incompleteness. We measure incompleteness for the model set Density Combination with Equal Weights and Stochastic Volatility, (DCEW-SV) at the bottom right panel in Figure 3. We plot the average estimate of the four clusters incompleteness. The estimates are similar over time and they show a 500% increase in February-March 2020, which is due to the Covid-19 pandemic. In particular, the incompleteness for cluster $n2$ has the larger increase. Interestingly, the volatilities start to reduce quite fast and they do over the remaining part of the sample. The values for clusters $n1$ and $t1$ in June are still twice the value pre-Covid-19; whereas the values for clusters $n2$ and $t2$ are closer to pre-Covid-19

Models	RMSPE	LS	CRPS	avQS-T	avQS-L	Violation
WN	1.518	-2.129	0.689	0.085	0.114	7.15%
Normal GARCH	1.513	-1.532^{**}	0.638^{**}	0.072^{**}	0.104^{**}	5.73%
Student-t GARCH	1.525	-1.420^{**}	0.649^{**}	0.074^{**}	0.106^{**}	3.50%
GJR GARCH	1.512	-1.517^{**}	0.639^{**}	0.072^{**}	0.105^{**}	5.56%
EW	1.522	-14.303	0.804	0.119	0.130	32.11%
BMA	1.525	-21.095	0.822	0.116	0.126	32.47%
DCEW-SV	$\mathbf{1.509}^{*}$	$\mathbf{-1.372}^{**}$	$\mathbf{0.557}^{**}$	$\mathbf{0.065}^{**}$	$\mathbf{0.090}^{**}$	**4.97%**

Table 4: Predictive results for next day S&P500 log-returns. Root mean square prediction error (RMSPE), logarithmic score (LS) and the continuous rank probability score (CRPS) are reported. Bold numbers indicate the best statistic for each loss function. One or two asterisks indicate that differences in accuracy from the white noise (WN) benchmark are credibly different from zero at 5%, and 1%, respectively, using Bayes estimates of the Diebold-Mariano t-statistic for equal loss. The underlying p-values are based on t-statistics computed with a serial correlation-robust variance, using the pre-whitened quadratic spectral estimator of Andrews and Monahan (1992). The alternative models considered are: GARCH model with Normal error for the aggregate index (Normal GARCH); GARCH model with Student-t error for the aggregate index (Student-t GARCH); Glosten-Jagannathan-Runkle GARCH model for the aggregate index (GJR GARCH, see Glosten et al., 1993); Equal Weights of all disaggregate models (EW); Bayesian Model Averaging of the models of all disaggregate models (BMA); Density Combination with Equal Weights within the clusters and Stochastic Volatility (DCEW-SV). The column "Violation" shows the number of times the realised value exceeds the 5% Value-at-Risk (VaR) predicted by the different models over the sample.

values.

Predictive accuracy of center and shape of the distribution. We compare the performance of our combination approach with results from five different basic models applied to the S&P500 log-returns: a white noise model in mean, often used as the main benchmark in equity premium predictability; the Normal GARCH(1,1) and the Student-t GARCH(1,1) models described above and applied to the aggregate S&P500 log-returns.

To explore the sensitivity of our results for model set incompleteness in more detail, we include the Normal GJR GARCH(1,1) model in Glosten et al. (1993) that includes leverage effects in the model set. This model is a richer model than the standard GARCH and should fit the data better. The leverage effect is considered among the stylized facts of financial returns and the added feature may become relevant in our analysis. Finally, given that simple combination methods might handle uncertainty accurately, we apply an equal weight combination of the all disaggregated GARCH models, labeled EW; and Bayesian model averaging, labelled BMA. In Section C in the Supplementary Material we also run AR models, AR GARCH models, different EW and BMA combinations of the models in the different clusters, see Table C.1. None of these provides very accurate predictions and are therefore excluded from the main text.

Out-of-sample predictive result are presented in Table 4. The first three columns deal with location and shape features of the predictive densities. It is seen that our combination scheme produces the lowest Root Mean Squared Prediction Error (RMSPE) and Cumulative Rank Probability Score (CRPS) and the highest Log Score (LS). The results indicate that the combination scheme is statistically superior to the no-predictability WN benchmark and it offers the most accurate statistics. The Normal GARCH(1,1) model, the Student-t GARCH(1,1) model and the Normal GJR GARCH(1,1) model fitted on the index also provide more accurate density predictions than the WN in terms of density prediction, but not on point prediction while our DCEW-SV is the only model that is statically superior at 5% level. For all three score criteria, the statistics given by BMA and EW are inferior to our combination scheme. In particular the density performance is very inferior. The lack of time-varying learning weights appears responsible for the poor performance in our data set.

Tail estimates and their economic value. We consider two statistics that refer to tails of the predictive densities. These statistics are the weighted averages of the Gneiting and Raftery (2007) quantile scores that are based on quantile predictions that correspond to the predictive densities from the different models. In the Supplementary Material, it is shown that avQS-T emphasises both tails and avQS-L the left tail of the predictive density relative to the realisation 1-step ahead. The fourth and fifth columns of Table 4 show results for tail evaluation. Our scheme provides the lowest avQS-T and avQS-L statistics, confirming the accuracy of our method in the tails of the distribution.

As an economic measure, we apply a Value-at-Risk (VaR) based measure, see Jorion (2006). We compare the accuracy of our models in terms of violations, that is the number of times that negative returns exceed the VaR predictions at time t, with the implication that actual losses on a portfolio are worse than was predicted. Higher accuracy results in numbers of violations close to the nominal value of 5%.[4] When looking at VaR violations, reported in the final column of Table 4, the number for all individual models is not very accurate, with the WN higher than 7%, the normal GARCH almost at 6%, the other two GARCH models at 3.5% indicating too large and a conservative density. Our DCEW-SV is the only one having a realised violation close to the 5% nominal value. The dramatic events in particular at the beginning of the Covid-19 pandemic in February/March 2020 drive the results. The property of our combination scheme to increase volatility in both normal clusters, and moreover, allocating more stock series to the fat tail cluster and a larger weight to the high volatility normal one, helps to model more accurately the lower tail of the index returns and covers more adequately risk.

[4]We choose a 5% nominal value and not the standard 1% due to a large number of negative returns and the high variability at the beginning of the Covid-19 pandemic. Restricting to 1% will limit the analysis.

5 Conclusions

We proposed in this paper a flexible Bayesian modelling approach with the construction of a predictive density combination with model set incompleteness and combination weight learning that can deal with large data sets in finance. The approach makes use of dimension reduction by dynamic clustering of the large number of components of the predictive mixture in mutually exclusive small subsets. Using a nonlinear dynamic factor model reduces the dimension of the large number of combination weights to a small set of cluster weights where a learning step is added. A parallel Sequential Monte Carlo algorithm is introduced for efficient Bayesian inference.

We applied the methodology to a large financial data set of individual stock returns which includes the Covid-19 crisis period. Empirical results show that our approach yields substantial accuracy gains in predictive means, volatilities and tail events compared to predictions from individual models and combination methods as Bayesian Model Averaging (BMA) and Equal Weights (EW). Measures of model set incompleteness and dynamic patterns in the cluster weights give valuable signals for improved financial modelling and policy analysis. These empirical results may provide useful information for investment fund management.

The line of research presented in this paper can be extended in several directions. For example, the cluster-based weights contain relevant signals about the importance of the predictive performance and composition of each of the clusters. Some clusters have a substantial weight while others have only little weight and such a pattern may vary over long time periods. This may lead to the construction of alternative model combinations for more accurate out-of-sample prediction and improved policy analysis. Another suggestion is to make use of judgemental information from individual forecasters, see McAlinn and West (2019), and combine this with the predictive information based on our modelling approach. Finally, we emphasise a fruitful connection and possible extension of our approach with work in the field of dynamic portfolio allocation, see Baştürk et al. (2019) for a basic analysis of a momentum strategy with a relatively small set of financial assets, and for a connection with work on machine learning methods for stock and bond market predictions, see Gu et al. (2020); Bianchi et al. (2020).

References

Aastveit, K. A., Cross, J. L., and van Dijk, H. K. (2022). Quantifying time-varying forecast uncertainty and risk for the real price of oil. *Journal of Business & Economic Statistics*, 0(0):1–15.

Aastveit, K. A., Mitchell, J., Ravazzolo, F., and van Dijk, H. K. (2019). The Evolution of Forecast Density Combinations in Economics. In *Oxford Research Encyclopedia of Economics and Finance*. Oxford University Press.

Aitchinson, J. (1982). The Statistical Analysis of Compositional Data. *Journal of the Royal Statistical Society Series, Series B*, 44:139–177.

Aitchinson, J. and Shen, S. M. (1980). Logistic-Normal Distributions: Some Properties and Uses. *Biometrika*, 67:261–272.

Amisano, G. and Geweke, J. (2010). Comparing and Evaluating Bayesian Predictive Distributions of Asset Returns. *International Journal of Forecasting*, 26:216–230.

Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, New York.

Andrews, D. W. K. and Monahan, J. C. (1992). An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator. *Econometrica*, 60:953–966.

Baştürk, N., Borowska, A., Grassi, S., Hoogerheide, L., and van Dijk, H. K. (2019). Forecast Density Combinations of Dynamic Models and Data Driven Portfolio Strategies. *Journal of Econometrics*, 210:170–186.

Bańbura, M., Giannone, D., and Reichlin, L. (2010). Large Bayesian Vector Auto Regressions. *Journal of Applied Econometrics*, 25:71–92.

Baştürk, N., Grassi, S., Hoogerheide, L., and Van Dijk, H. K. (2016). Parallelization Experience with Four Canonical Econometric Models Using ParMitISEM. *Econometrics*, 4:1–11.

Bianchi, D., Büchner, M., and Tamoni, A. (2020). Bond Risk Premiums with Machine Learning. *The Review of Financial Studies*, 34:1046–1089.

Bianchi, D. and McAlinn, K. (2018). Divide and Conquer: Financial Ratios and Industry Returns Predictability. Working papers, SSRN.

Billio, M., Casarin, R., Ravazzolo, F., and van Dijk, H. K. (2013). Time-Varying Combinations of Predictive Densities using Nonlinear Filtering. *Journal of Econometrics*, 177:213–232.

Billio, M., Casarin, R., and Rossini, L. (2019). Bayesian nonparametric sparse var models. *Journal of Econometrics*, 212(1):97–115.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456.

Casarin, R., Grassi, S., Ravazzolo, F., and van Dijk, H. K. (2020). A Bayesian Dynamic Compositional Model for Large Density Combinations in Finance. Working paper series 20-27, Rimini Centre for Economic Analysis.

Casarin, R. and Veggente, V. (2020). Random Projection Methods in Economics and Finance. In Petr, H., Uddin, M., and Abedin, M. Z., editors, *The Essentials of Machine Learning in Finance and Accounting*, pages 1–20. Routledge Taylor & Francis.

Choi, H. and Varian, H. (2012). Predicting the Present with Google Trends. *Economic Record*, 88:2–9.

Corielli, F. and Marcellino, M. (2006). Factor based index tracking. *Journal of Banking & Finance*, 30(8):2215–2233.

Creal, D. (2007). Sequential Monte Carlo Samplers for Bayesian DSGE Models. Working papers, Vrije Universitiet.

Creal, D. (2012). A Survey of Sequential Monte Carlo Methods for Economics and Finance. *Econometric Reviews*, 31:245–296.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA.

Einav, L. and Levin, J. (2014). Economics in The Age of Big Data. *Science*, 346:715–718.

Favirar, R., Rebolledo, D., Chan, E., and Campbell, R. (2008). A Parallel Implementation of K-Means Clustering on GPUs. *Proceedings of 2008 International Conference on Parallel and Distributed Processing Techniques and Applications*, 2:14–17.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models: Modeling and Applications to Random Processes*. Springer Series in Statistics. Springer.

Geweke, J. (2010). *Complete and Incomplete Econometric Models*. The Econometric and Tinbergen Institutes Lectures. Princeton University Press.

Glosten, L. R., Jagannathan, R., and Runkle, D. E. (1993). On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. *Journal of Finance*, 48:1779–1801.

Gneiting, T. and Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102:359–378.

Gneiting, T. and Ranjan, R. (2013). Combining Predictive Distributions. *Electronic Journal of Statistics*, 7:1747–1782.

Granger, C. W. J. (1998). Extracting Information from Mega-Panels and High-Frequency Data. *Statistica Neerlandica*, 52:258–272.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, 33(5):2223–2273.

Hall, S. G. and Mitchell, J. (2007). Combining density forecasts. *International Journal of Forecasting*, 23:1–13.

Herbst, E. and Schorfheide, F. (2014). Sequential Monte Carlo Sampling for DSGE Models. *Journal of Applied Econometrics*, 29:1073–1098.

Hoogerheide, L., Kleijn, R., Ravazzolo, R., van Dijk, H. K., and Verbeek, M. (2010). Forecast Accuracy and Economic Gains from Bayesian Model Averaging using Time Varying Weights. *Journal of Forecasting*, 29(1-2):251–269.

Hoogerheide, L., Opschoor, A., and Van Dijk, H. K. (2012). A Class of Adaptive Importance Sampling Weighted EM Algorithms for Efficient and Robust Posterior and Predictive Simulation. *Journal of Econometrics*, 171:101–120.

Jorion, P. (2006). *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, New York.

Kaufmann, S. and Schumacher, C. (2017). Identifying relevant and irrelevant variables in sparse factor models. *Journal of Applied Econometrics*, 32(6):1123–1144.

Kaufmann, S. and Schumacher, C. (2019). Bayesian estimation of sparse dynamic factor models with order-independent and ex-post mode identification. *Journal of Econometrics*, 210(1):116–134.

Kim, S. and Kim, S. (2020). Index tracking through deep latent representation learning. *Quantitative Finance*, 20(4):639–652.

Koop, G. and Korobilis, D. (2013). Large Time-Varying Parameter VARs. *Journal of Econometrics*, 177:185–198.

Koopman, S. J., Lucas, A., and Scharth, M. (2015). Numerically Accelerated Importance Sampling for Nonlinear Non-Gaussian State Space Models. *Journal of Business and Economic Statistics*, 33:114–127.

Kunsch, H. R. (2005). Recursive Monte Carlo Filters: Algorithms and Theoretical Analysis. *Annals of Statistics*, 33:1983–2021.

Liesenfeld, R. and Richard, J. F. (2003). Univariate and Multivariate Stochastic Volatility Models: Estimation and Diagnostics. *Journal of Empirical Finance*, 10:505–531.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Verlag, New York, USA.

Lopes, H. F. and West, M. (2004). Bayesian Model Assessment in Factor Analysis. *Statistica Sinica*, 1:41–68.

MacLehose, R. and Dunson, D. (2010). Bayesian semiparametric multiple shrinkage. *Biometrics*, 66(2):455–462.

Malsiner Walli, G., Frühwirth-Schnatter, S., and Grün, B. (2016). Model-Based Clustering Based on Sparse Finite Gaussian Mixtures. *Statistics and Computing*, 26:303–326.

McAlinn, K., Aastveit, K. A., Nakajima, J., and West, M. (2020). Multivariate bayesian predictive synthesis in macroeconomic forecasting. *Journal of the American Statistical Association*, 115(531):1092–1110.

McAlinn, K. and West, M. (2019). Dynamic Bayesian Predictive Synthesis in Time Series Forecasting. *Journal of Econometrics*, 210:155 – 169.

Mitchell, J. and Hall, S. G. (2005). Evaluating, Comparing and Combining Density Forecasts Using the KLIC with an Application to the Bank of England and NIESER "Fan" Charts of Inflation. *Oxford Bulletin of Economics and Statistics*, 67:995–1033.

Pagan, A. (1984). Econometric Issues in the Analysis of Regressions with Generated Regressors. *International Economic Review*, 25:221–247.

Richard, J. and Zhang, W. (2007). Efficient High-Dimensional Importance Sampling. *Journal of Econometrics*, 141:1385–1411.

Stock, J. H. and Watson, W. M. (1999). Forecasting Inflation. *Journal of Monetary Economics*, 44:293–335.

Stock, J. H. and Watson, W. M. (2002). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of American Statistical Association*, 97:1167–1179.

Stock, J. H. and Watson, W. M. (2005). Implications of Dynamic Factor Models for VAR Analysis. Technical report, NBER Working Paper No. 11467.

Stock, J. H. and Watson, W. M. (2014). Estimating Turning Points Using Large Data Sets. *Journal of Econometris*, 178:368–381.

Takanashi, K. and McAlinn, K. (2019). Predictive Properties of Forecast Combination, Ensemble Methods, and Bayesian Predictive Synthesis. Working papers, ArXiv.

Terui, N. and van Dijk, H. K. (2002). Combined forecasts from linear and nonlinear time series models. *International Journal of Forecasting*, 18:421–438.

Varian, H. (2014). Machine Learning: New Tricks for Econometrics. *Journal of Economics Perspectives*, 28:3–28.

Varian, H. and Scott, S. (2014). Predicting the Present with Bayesian Structural Time Series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5:4–23.

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018). Using Stacking to Average Bayesian Predictive Distributions (with Discussion). *Bayesian Analysis*, 13(3):917 – 1007.

Supplementary Material for the paper "A Flexible Predictive Density Combination for Large Financial Data Sets in Regular and Crisis Periods"

A Derivation of the probability density function of the logistic normal distribution of the weight vector $\mathbf{w}_t$.

We present the proof for the vector of m probabilistic weights $\mathbf{z}$ on the low dimensional space and then make use of the specification that the n probabilistic weights $\mathbf{w}$ are linear combinations of the weights $\mathbf{z}$. For notational convenience we delete the subindex t.

Proposition A.1. *Given that the $m\times 1$ vector $\mathbf{v} = (v_1, \ldots, v_m)'$ has a multivariate normal distribution, see equation (7):*

$$\mathbf{v} \sim \mathcal{N}_m(\mathbf{v}_{-1}, \boldsymbol{\Sigma}_\eta),$$

where $\boldsymbol{\Sigma}_\eta$ is a diagonal matrix and the $m \times 1$ vector $\mathbf{z} = (z_1, \ldots, z_m)'$ is given as:

$$z_j = \frac{\exp(v_j)}{\sum_{i=1}^{m} \exp(v_i)}, \qquad j = 1, \ldots, m, \tag{A.1}$$

see equation (8), the vector $\tilde{\mathbf{z}} = (z_1, \ldots, z_{m-1})'$ follows a logistic-normal distribution $\tilde{\mathbf{z}} \sim \mathcal{L}_{m-1}(\mathbf{D}\mathbf{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}')$ with density function:

$$f(\tilde{\mathbf{z}}|\mathbf{D}\boldsymbol{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}') = (2\pi)^{-(m-1)/2}|\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'|^{-1/2}\left(\prod_{j=1}^{m} z_j\right)^{-1} \times$$
$$\exp\left(-\frac{1}{2}\left(\log\left(\frac{\tilde{\mathbf{z}}}{z_m}\right) - \mathbf{D}\mathbf{v}_{t-1}\right)'(\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}')^{-1}\left(\log\left(\frac{\tilde{\mathbf{z}}}{z_m}\right) - \mathbf{D}\mathbf{v}_{-1}\right)\right), \tag{A.2}$$

where $z_m = 1 - \sum_{i=1}^{m-1} z_i$, and the $(m-1)\times m$ matrix $\mathbf{D}$ is given by

$$\mathbf{D} = (\mathbf{I}_{m-1} \;\; | \;\; -\iota_{m-1}),$$

with $\mathbf{I}_{m-1}$ equal to the $(m-1)\times(m-1)$ identity matrix, and ι_{m-1} is the $(m-1)\times 1$ vector containing only ones.

The proof that (A.2) is the density function of $\tilde{\mathbf{z}} = (z_1, \ldots, z_{m-1})'$ consists of three main steps. First, divide the numerator and denominator of (A.1) by

$\exp(v_m)$, which yields:

$$z_j = \frac{\exp(v_j - v_m)}{\sum_{i=1}^{m} \exp(v_i - v_m)} \qquad j = 1, \ldots, m-1,$$

where $z_m = 1 - \sum_{i=1}^{m-1} z_i$. Define:

$$\mathbf{u} = \begin{pmatrix} v_1 - v_m \\ \vdots \\ v_{m-1} - v_m \end{pmatrix} = \mathbf{D}\mathbf{v},$$

then $\tilde{\mathbf{z}} = g(\mathbf{u})$, where $g(\cdot)$ is a one-to-one or bijective function. Note that the symbol $\mathbf{u}$ is only used in this derivation. Further,

$$\mathbf{u} = \mathbf{D}\mathbf{v} \sim \mathcal{N}_{m-1}(\mathbf{D}\mathbf{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_{\eta}\mathbf{D}').$$

Second, the inverse transformation $\mathbf{u} = g^{-1}(\tilde{\mathbf{z}})$ is given as:

$$u_j = \log\left(\frac{z_j}{z_m}\right) = \log(z_j) - \log\left(1 - \sum_{i=1}^{m-1} z_i\right) \qquad j = 1, \ldots, m-1$$

with Jacobian matrix

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{z}}} &= \begin{pmatrix} z_1^{-1} & 0 & \cdots & 0 \\ 0 & z_2^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & z_{m-1}^{-1} \end{pmatrix} + \left(1 - \sum_{i=1}^{m-1} z_i\right)^{-1} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & & \vdots \\ \vdots & & \ddots & 1 \\ 1 & \cdots & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} z_1^{-1} & 0 & \cdots & 0 \\ 0 & z_2^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & z_{m-1}^{-1} \end{pmatrix} + z_m^{-1} \times \iota_{(m-1)\times(m-1)}, \end{aligned}$$

where $\iota_{(m-1)\times(m-1)}$ is the $(m-1) \times (m-1)$ matrix containing only ones.

The determinant of $\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{z}}}$ is

$$\left|\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{z}}}\right| = \prod_{j=1}^{m} z_j^{-1}, \tag{A.3}$$

where use is made of the following determinant rule [5]

$$|\mathbf{A} + \mathbf{x}\mathbf{y}'| = |\mathbf{A}| \times (1 + \mathbf{y}'\mathbf{A}^{-1}\mathbf{x})$$

[5] See Anderson (1984)

with

$$\begin{aligned}
\mathbf{A} &= \begin{pmatrix} z_1^{-1} & 0 & \cdots & 0 \\ 0 & z_2^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & z_{m-1}^{-1} \end{pmatrix} \\
\mathbf{x} &= z_m^{-1} \times \iota_{(m-1)} \\
\mathbf{y} &= \iota_{(m-1)}
\end{aligned}$$

where

$$\begin{aligned}
|\mathbf{A}| &= \prod_{j=1}^{m-1} z_j^{-1} \\
\mathbf{A}^{-1} &= \begin{pmatrix} z_1 & 0 & \cdots & 0 \\ 0 & z_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & z_{m-1} \end{pmatrix} \\
\mathbf{y}'\mathbf{A}^{-1}\mathbf{x} &= z_m^{-1} \sum_{j=1}^{m-1} z_j = \frac{1 - z_m}{z_m} \\
1 + \mathbf{y}'\mathbf{A}^{-1}\mathbf{x} &= 1 + \frac{1 - z_m}{z_m} = \frac{z_m + 1 - z_m}{z_m} = \frac{1}{z_m},
\end{aligned}$$

where pre and postmultiplying by ι'_{m-1} and ι_{m-1} obviously means that one can compute the sum of all elements of the matrix $\mathbf{A}^{-1}$ (where this sum is here equal to $\sum_{j=1}^{m-1} z_j$), and where $\sum_{j=1}^{m-1} z_j = 1 - z_m$, so that it follows that

$$|\mathbf{A}| \times (1 + \mathbf{y}'\mathbf{A}^{-1}\mathbf{x}) = \prod_{j=1}^{m} z_j^{-1}.$$

Third, given that $\mathbf{u}$ has multivariate normal $\mathcal{N}_{m-1}(\mathbf{D}\mathbf{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}')$ distribution with density

$$\begin{aligned}
f(\mathbf{u}|\mathbf{D}\mathbf{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}') = (2\pi)^{-(m-1)/2}|\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'|^{-1/2} \times \\
\exp\left(-\frac{1}{2}\left(\mathbf{u} - \mathbf{D}\mathbf{v}_{-1}\right)'\left(\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'\right)^{-1}\left(\mathbf{u} - \mathbf{D}\mathbf{v}_{-1}\right)\right).
\end{aligned}$$

Substituting $\mathbf{u} = \log\left(\frac{\tilde{\mathbf{z}}}{z_m}\right)$ into (A.4) and multiplying with $\left|\frac{\partial \mathbf{u}}{\partial \tilde{\mathbf{z}}}\right| = \prod_{j=1}^{m} z_j^{-1}$ yields:

$$f(\tilde{\mathbf{z}}|\mathbf{D}\mathbf{v}_{-1}, \mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}') = (2\pi)^{-(m-1)/2}|\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'|^{-1/2}\left(\prod_{j=1}^{m} z_j\right)^{-1} \times$$
$$\exp\left(-\frac{1}{2}\left(\log\left(\frac{\tilde{\mathbf{z}}}{z_m}\right) - \mathbf{D}\mathbf{v}_{-1}\right)'(\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}')^{-1}\left(\log\left(\frac{\tilde{\mathbf{z}}}{z_m}\right) - \mathbf{D}\mathbf{v}_{-1}\right)\right). \quad \text{(A.4)}$$

Q.E.D.

As a final step we make use of the specification that the $n-1$ weights $\tilde{\mathbf{w}} = (w_1, ..., w_{n-1})'$ are linear combinations of the logistic-normal weights $\tilde{\mathbf{z}}$. In matrix notation we use $\tilde{\mathbf{w}} = \tilde{\mathbf{B}}\tilde{\mathbf{z}}$ where $\tilde{\mathbf{B}}$ is an $(n-1)\times m$ matrix containing an appropriate subset of elements b_{ij} and $\sum_{i=1}^{n-1} w_i = 1 - w_n$. Then one can write that the logistic normal distribution of $\tilde{\mathbf{w}}$ is given as :

$$\tilde{\mathbf{w}} \sim \mathcal{L}_{n-1}\left(\tilde{\mathbf{B}}\mathbf{D}\mathbf{v}_{-1}, \tilde{\mathbf{B}}\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'\tilde{\mathbf{B}}'\right). \quad \text{(A.5)}$$

We note that the density of the complete vector $\mathbf{w}_t = (w_{1t}, ..., w_{nt})'$ is singular due to the adding-up restriction of all n weights. A second source of degeneracy is intrinsic to our projection strategy which implies rank deficiency of the matrix $\tilde{\mathbf{B}}\mathbf{D}\boldsymbol{\Sigma}_\eta\mathbf{D}'\tilde{\mathbf{B}}'$.

B Algorithm details

The analytical solution of the optimal filtering problem is generally not known. Also, the cluster-based mapping requires the solution of an optimisation problem which is not available in analytical form. Thus, we apply a parallel sequential clustering in order to determine the series allocation and the M-Filter in order to estimate the latent weights and combination parameters. The details of the algorithms are given in the following subsections.

B.1 Parallel dynamic clustering

The clustering step of classifying predictive draws to a particular group is done using the K-Means algorithm, which groups these draws based on their distance from the nearest cluster mean. These means are given in our dataset, see Section 4, as $m_{t1}, \ldots, m_{t4}$ corresponding to high and low volatility and large and small degrees of freedom. In each step of the algorithm the computation of the cluster mean, the distances from the cluster means and the selection of the cluster mean with minimum distance are easy to parallelise.

The generated draws from the predictive distributions are allocated according to the number of available cores, see Favirar et al. (2008) and the reference therein. More specifically, the parallel evaluation of the dynamic clustering process, using the K-Means algorithm is done as follows. The first step is to partition the generated draws of $\tilde{y}_{it}$, $i = 1, \ldots, n$ at each period t into P subsets, where P

is chosen according to the number of available threads. In our case we made use of AMD Ryzen 3900X that is a 12 core 24 threads processor, then $P = 24$.
In our empirical application the number of models are 4 and the number of series are 496, this brings to a total of $n = 1984$ predictive densities to cluster, with $\kappa = n/P = 83$.

Then for each $t = 1, \ldots, T$ one proceeds as follows: given initial guess of 4 means $m_{1t}, \ldots, m_{4t}$, the algorithm proceeds by alternating between two steps:

1) For each thread p, $p = 1, \ldots, P$, assign $\kappa = n/P$ generated draws $\tilde{y}_{it}$:

   2a) **Assignment step**: Assign each generated draw $\tilde{y}_{it}$, $i = 1, \ldots, n$ to the cluster S_{pt} with the nearest mean:

$$S_{pt} = \{\tilde{y}_{\kappa t} : ||\tilde{y}_{\kappa t} - m_{it}|| \leq ||\tilde{y}_{\kappa t} - m_{jt}|| \quad 1 \leq j \leq 4\}, \tag{B.6}$$

   2b) **Update step**: Recalculate means (centroids) for generated draws assigned to each thread:

$$m_{p(t+1)} = \frac{1}{|S_{pt}|} \sum_{\tilde{y}_{jt} \in S_{pt}} \tilde{y}_{jt}.$$

3) Uses all local means for each thread to find the global mean.

4) Repeat until convergence.

The dynamic clustering is parallel in point 2a) and 2b) and this can be done in the CPU or GPU context. The step 3) takes the clusters means and calculates the global mean.

B.2 M-Filter

Our Predictive Density Combination (PDC) has the general state-space representation

$$\begin{aligned} y_t &\sim f_t(y_t|\mathbf{v}_t), \\ \mathbf{v}_t &\sim p_t(\mathbf{v}_t|\mathbf{v}_{t-1}), \\ \mathbf{v}_0 &\sim p_0(\mathbf{v}_0), \end{aligned} \tag{B.7}$$

$t = 1, \ldots, T$, where $f_t(y_t|\mathbf{v}_t)$ is the measurement density, $p_t(\mathbf{v}_t|\mathbf{v}_{t-1})$ is the transition density, and $p_0(\mathbf{v}_0)$ the initial density of the state. The dependence on the static parameter vector $\boldsymbol{\theta}$ (which contains σ_η^2) and on the sets $\tilde{y}_{it}$ and σ_{it}^2 $(i = 1, \ldots, n; t = 1, \ldots, T)$ is omitted for notational convenience. In the application the parameters are included into an augmented state vector and a simulation-based filtering method is applied to the augmented state space model, which is presented, for our case, in equations (6) and (10) with state-space representation given in (11) to (15).

Our model is non-linear and non-Gaussian and, therefore, the standard Kalman filter cannot be used. We make use of a numerical method in the class of the Sequential Monte Carlo (SMC) algorithms, see Doucet et al. (2001) for a review. Specifically, we apply the M-Filter of Baştürk et al. (2019). This application assumes a Gaussian measurement density, thus it needs to be adapted to our PDC. The initial density $p_0(\mathbf{v}_0)$ and the transition density $p_t(\mathbf{v}_t|\mathbf{v}_{t-1})$ of the PDC are the Gaussian densities $\mathcal{N}_m(\mathbf{0}, \Sigma_\eta)$ and $\mathcal{N}_m(\mathbf{v}_{t-1}, \Sigma_\eta)$ respectively.

The filtering density is:

$$p(\mathbf{v}_t|y_{1:T}) = \frac{f_t(y_t|\mathbf{v}_t)p_t(\mathbf{v}_t|\mathbf{v}_{t-1})}{p_t(y_t|y_{1:t-1})}. \tag{B.8}$$

It is approximated by a set of random probability weights and variables (known as particles) $\{\mathbf{v}_t^{(i)}, \omega_t^i\}_{i=1}^N$. The usual SMC algorithms consist of sequences of propagation and updating steps. The propagation step relates to the way the sample is drawn at time t and the updating step provides an importance sampling (IS) correction for not using the true target density for sampling. It is well known that the propagation step leads to the necessity of resampling, as the sequential importance sampling is bound to lead to weight degeneracy problems. Moreover, the resampling may be time consuming but it also introduces additional MC variation.

The M-Filter tackles the filtering process differently. The propagation step is replaced by an IS step at each time point t, avoiding resampling. Moreover, it is well known that the choice of the proposal density is crucial for the performance of any IS scheme, see Doucet et al. (2001) for a general discussion and Section 3 for an application to our case. The M-Filter method approximates a target density in equation (B.7) using the *Mixture of t by Importance Sampling Weighted Expectation–Maximization* (MitISEM) algorithm proposed by Hoogerheide et al. (2012), where a mixture of Student-t distributions is used as the importance distribution. The MitISEM approach is able to effectively approximate complex, heavy-tailed, non-elliptical distributions and the M-Filter has been shown to be very fast and reliable, see Baştürk et al. (2019).

The M-Filter algorithm is presented below:

1) **Initialization**. Draw $\mathbf{v}_0^{(j)}$ for $j = 1, \ldots, M$, from the initial density[6] $p(\mathbf{v}_0)$, e.g. a uniform density.

2) **Recursion**. For $t = 1, \ldots, T$ construct the candidate density $g_t(\mathbf{v}_t)$ using the MitISEM algorithm as follows:

   a) *Initialization*: Simulate draws $\mathbf{v}_t^{(j)}$, $j = 1, \ldots, M$, from a naive candidate density $g_t^{(0)}(\cdot)$ (e.g. a Student-t with $\nu = 3$ degrees of freedom).

[6]Strictly speaking random draws are generated from a particular distribution with a corresponding density. For notational convenience and when there is no ambiguity, we make use of the short-hand given in the text.

Compute the corresponding IS weights:

$$\tilde{\omega}_t^{(j)} = \frac{f_T(y_t|\mathbf{v}_t^{(j)}, \tilde{y}_t) p_t(\mathbf{v}_t^{(j)}|\mathbf{v}_{t-1}^{(j)})}{g_t^{(0)}(\mathbf{v}_t^{(j)})},$$

where the target density kernel has the form $f(y_t|\mathbf{v}_t, \tilde{y}_t)p(\mathbf{v}_t|\mathbf{v}_{t-1}^{(j)})$. Normalize the weights to $\omega_t^{(j)}$.

b) *Adaptation*: Use the draws $\mathbf{v}_t^{(j)}$ and the weights $\omega_t^{(j)}$ from the naive density $g_t^{(0)}(\cdot)$ to estimate the mean and covariance matrix of the target distribution. Use these estimates as the mode and the scale matrix of the adapted Student-t density $g_t^{(a)}(\cdot)$. Draw a sample $\mathbf{v}_t^{(j)}$ from $g_t^{(a)}(\cdot)$ and compute the IS weights for this sample.

c) *Apply the IS weighted EM algorithm* (see Hoogerheide et al., 2012) given the sample from step b) and the corresponding IS weights. The output consists of the new candidate Student-t density with $H = 1$ component $g_t^{(1)}(\cdot)$ and optimized parameters (mean, variance and degrees of freedom, namely μ_1, Σ_1 and ν_1). Draw a new sample $\mathbf{v}_{t,H}^{(j)}$ from this candidate, and compute the corresponding IS weights. Calculate the coefficient of variation $CV^{(H)}$ $(H = 1)$ of the normalized weights $\omega_t^{(j)}$, $j = 1, \ldots, M$.

d) *Iterate on the number of mixture components.* Given the current mixture of H components with corresponding μ_h, Σ_h, ν_h and the mixture weight ξ_h $(h = 1, \ldots, H)$, add the next component $H + 1$ to the mixture in the following way:

   d.1) Select a sub-sample $\mathbf{v}_{t,h}^{(i_1)}, \ldots, \mathbf{v}_{t,h}^{(i_N)}$ of size N, where N is a certain small percentage of M, where the selection indexes $i_1, \ldots, i_N$ correspond to the particles with the highest IS weights, and estimate the mean and variance. Use these parameters as the starting mode and scale parameters for the new mixture component, μ_{H+1} and Σ_{H+1}. This step ensures that the new component covers a region where previous candidate mixture had a relatively low probability mass. Usually, two or three components are sufficient given the flexibility of the mixture of Student-t densities.

   d.2) Given the draws and weights from previous mixture $g_t^{(H)}(\cdot)$ apply the EM algorithm to optimize (again) each mixture component μ_h, Σ_h, ν_h and the ξ_h $(h = 1, \ldots, H + 1)$.

   d.3) Draw $\mathbf{v}_{t,H+1}^{(j)}$ from the new mixture $g_t^{(H+1)}(\cdot)$ of $H + 1$ Student-t densities from step d.2) and evaluate the corresponding normalized importance weights $\omega_t^{(j)}$, $j = 1, \ldots, M$.

   d.4) Calculate the coefficient of variation $CV^{(H+1)}$ of the normalised weights ω_t, $j = 1, \ldots, M$.

e) Assess convergence of the candidate density's quality by inspecting

whether the relative change between $CV^{(H)}$ and $CV^{(H+1)}$ is greater than the chosen threshold (e.g. 0.01) and return to step d) unless the algorithm has converged. Set $H = H + 1$.

3) **Draws**. Use the constructed mixture density in IS: Draw $\mathbf{v}_{t,H}^{(j)}$ from the mixture density $g_t^{(H)}(\cdot)$, compute the corresponding normalised IS weights $\omega_t^{(j)}$, and approximate $\mathrm{E}[h_t(\mathbf{v}_t)|y_{1:T}]$ by:

$$\hat{h}_t(\mathbf{v}_t) = \sum_{j=1}^{M} \omega_t^{(j)} h_t(\mathbf{v}_{t,H}^{(j)}). \tag{B.9}$$

The main advantages of the M-Filter are adaptation, which requires only candidate draws and IS weights and, by implication, that the M-Filter can simultaneously deal with several target densities over time, see (Baştürk et al., 2019) for a discussion. The computational efficiency gains are feasible by making use of parallel computing, for instance, using graphics processing units.

C Additional details on empirical results

Table C.1 reports the full sample of results for all alternative methods. We extend both models for the aggregate series by considering a white noise without mean; an AR model; AR-GARCH model with Normal error; AR-GARCH model with Student-t error. Moreover, we consider other equal weights and Bayesian model averaging combinations: EW of the models in the two Normal clusters; EW of the models in the two Student-t clusters; BMA of the models in the two Normal clusters; BMA of the models in the two Student-t cluster. Finally, we report EW of the models in the "n1" cluster; EW of the models in the "n2" cluster; EW of the models in the "t1" cluster; EW of the models in the "t2" cluster. None of the above models perform similarly to our DCEW-SV, neither to the other alternative models, in particular in terms of density predicting and violation. All the individual clusters do poorly, indicating that selecting only a sample of models is not a good strategy.

Models	RMSPE	LS	CRPS	avQS-T	avQS-L	Violation
WN	1.518	-2.129	0.689	0.085	0.114	7.15%
WN without mean	2.452	-2.213	1.075	0.135	0.167	6.36%
Normal GARCH	1.513	-1.532^{**}	0.638^{**}	0.072^{**}	0.104^{**}	5.73%
Student-t GARCH	1.525	-1.420^{**}	0.649^{**}	0.074^{**}	0.106^{**}	3.50%
GJR GARCH	1.512	-1.517^{**}	0.639^{**}	0.072^{**}	0.105^{**}	5.56%
AR	1.601	-2.266	0.716	0.089	0.119	7.15%
AR Normal GARCH	1.541	-1.544^{**}	0.639^{**}	0.072^{**}	0.105^{**}	6.52%
AR Student-t GARCH	1.543	-1.419^{**}	0.650^{**}	0.074^{**}	0.107^{**}	3.97%
EW	1.522	-14.303	0.804	0.119	0.130	32.11%
EW Normal	1.520	-13.560	0.784	0.116	0.126	31.63%
EW Student-t	1.525	-9.041	0.742	0.116	0.126	31.63%
BMA	1.525	-21.095	0.822	0.116	0.126	32.47%
BMA Normal	1.522	-20.810	0.839	0.130	0.137	38.47%
BMA Student-t	1.523	-20.503	0.839	0.130	0.137	37.83%
EW n1	1.523	-21.099	0.841	0.131	0.138	38.95%
EW n2	1.535	-5.921	0.802	0.111	0.128	21.46%
EW t1	1.528	-14.965	0.811	0.122	0.134	35.13%
EW t2	1.518	-3.434	0.699	0.089	0.114	17.17%
DCEW-SV	$\mathbf{1.509}^{*}$	$\mathbf{-1.372}^{**}$	$\mathbf{0.557}^{**}$	$\mathbf{0.065}^{**}$	$\mathbf{0.090}^{**}$	**4.97%**

Table C.1: Predicting results for next day S&P500 log-returns. Root mean square prediction error (RMSPE), logarithmic score (LS) and the continuous rank probability score (CRPS) are reported. Bold numbers indicate the best statistic for each loss function. One or two asterisks indicate that differences in accuracy from the white noise (WN) benchmark are credibly different from zero at 5%, and 1%, respectively, using the Diebold-Mariano t-statistic for equal loss.The underlying p-values are based on t-statistics computed with a serial correlation-robust variance, using the pre-whitened quadratic spectral estimator of Andrews and Monahan (1992). The alternative models considered are: a white noise without mean (WN without nmean); GARCH model with Normal error for the aggregate index (Normal GARCH); Garch model with Student-t error for the aggregate index (Student-t GARCH); Glosten-Jagannathan-Runkle Garch model for the aggregate index (GJR GARCH); AR model for the aggregate index (AR); AR-GARCH model with Normal error for the aggregate index (AR Normal GARCH); AR-GARCH model with Student-t error for the aggregate index (AR Student-t GARCH); Equal Weights of all disaggregate models (EW); Equal Weights of the models in the two Normal clusters (EW Normal); Equal Weights of the models in the two Student-t clusters (EW Student-t); Bayesian Model Averaging of all models (BMA); Bayesian Model Averaging of the models in the two Normal clusters (BMA Normal); Bayesian Model Averaging of the models in the two Student-t cluster (BMA Student-t); Equal Weights of the models in the "n1" cluster (EW n1); Equal Weights of the models in the "n2" cluster (EW n2); Equal Weights of the models in the "t1" cluster (EW t1); Equal Weights of the models in the "t2" cluster (EW t2); Density combination with equal weights and stochastic volatility (DCEW-SV). The column "Violation" shows the number of times the realised value exceeds the 5% Value-at-Risk (VaR) predicted by the different models over the sample.