Fakultät für Wirtschaftswissenschaften
Facoltà di Economia
Faculty of Economics and Management

BEMPS –
Bozen Economics & Management
Paper Series

NO 59/ 2019

Density Forecasting

Federico Bassetti, Roberto Casarin
and Francesco Ravazzolo

Density Forecasting

Federico Bassetti, Roberto Casarin and Francesco Ravazzolo

Abstract This paper reviews different methods to construct density forecasts and to aggregate forecasts from many sources. Density evaluation tools to measure the accuracy of density forecasts are reviewed and calibration methods for improving the accuracy of forecasts are presented. The manuscript provides some numerical simulation tools to approximate predictive densities with a focus on parallel computing on graphical process units. Some simple examples are proposed to illustrate the methods.

Keywords: Density forecasting; density combinations; density evaluation; bootstrapping; Bayesian inference; Monte Carlo simulations; GPU computing.
JEL-codes: C10; C11; C15; C53; C63.

1 Introduction

Economic decision in real time are made under a high degree of uncertainty. One of the prominent feature of this uncertainty is that relevant information is missing at the moment of the decision. This requires to build forecasts to try to track the future evolution of the economic processes and to inform decision-makers. Researchers recognized the fundamental importance of forecasts a long time ago; but the focus was mainly on point forecasting. Point forecasting is often associated to the mean of a distribution and it is optimal for highly restricted loss functions, such as quadratic loss function. More generally, the value of a point forecast can be increased by supplementing it with some measure of uncertainty and complete probability distributions over outcomes provide information helpful for making economic decisions; see, for example, Anscombe (1968) and Zarnowitz (1969) for early works and the discussions in Granger and Pesaran (2000), Timmermann (2006) and Gneiting (2011). Recently, probabilistic forecasts in the form of predictive probability distributions have become prevalent in various fields, including macro economics with routine publications of fancharts from central banks, finance with asset allo-

Federico Bassetti
Politcenico di Milano, Milan Italy, e-mail: federico.bassetti@polimi.it

Roberto Casarin
University Ca' Foscari of Venice, Venice Italy, e-mail: r.casarin@unive.it

Francesco Ravazzolo
Free University of Bozen-Bolzano and BI Norwegian Business School, Bolzano Italy, e-mail: francesco.ravazzolo@unibz.it.

cation strategies based on higher-order moments, and meteorology with operational ensemble forecasts of future weather Tay and Wallis (2000), Gneiting and Katzfuss (2014). For example in central bank forecasting, the Bank of England, Norges Bank, Sveriges Riksbank publish so-called fan charts for macroeconomic variables such as inflation and GDP growth.

This paper reviews several methods to construct density forecasts for parametric models. The first method assumes a distribution for the errors and ignore parameter uncertainty; the second method, bootstrapping, accounts for parameter and error uncertainties in a frequentist environment; and the third method relies on Bayesian inference. The three methods rely on different assumptions. We describe them in the case of the simple linear regression models and provide tools to extend the analysis to more complex models. We also discuss density combinations as a tool to deal in the case there are several density forecasts and an *a priori* selection is difficult. And we provide some evaluation tools to measure the accuracy of density forecasts, accounting for the fact that the “true” density forecast is never observed, even *ex post*.

Moreover, in order to cope with the fact that relevant information is missing at the moment of decision, several papers (e.g., see Stock and Watson, 1999, 2002, 2005, 2014, and Bańbura et al., 2010) suggest to forecast with large sets of data. The recent fast growth in (real-time) big data allows researchers to forecast 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 forecasting using a large set of forecasts. However, forecasting with large data sets including many forecasts and high-dimensional models requires 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. In the application, we propose Graphical Processor Units (GPUs) as a tool to reduce computation time based on massively parallel computation and review the GPU computing functions introduced in the MATLAB parallel computing toolbox to reduce the steep learning curve of a dedicated programming language.

The structure of the paper is organized as follows. Section 2 presents the different methods to compute density forecasts. Section 3 describes density combinations and section 4 proposes different methods for density evaluation. Section 5 introduces GPU computing and applies to examples based on Monte Carlo (MC) simulations and an accept-reject algorithm to compute density. Section 6 concludes.

2 Computing Density Forecasts

This section reviews several methods to construct density forecasts. We discuss methodologies applied to a simple linear regression model:

$$y_t = x_t'\beta + \varepsilon_t,\ t = 1,\ldots,T,\ \varepsilon_t \sim i.i.d.(0,\sigma^2) \tag{1}$$

where $\theta = (\beta, \sigma^2)$ is a $((m+1) \times 1)$ vector of parameters; β a $(m \times 1)$ vector of coefficients; σ^2 the variance of the error term ε_t; and x_t is a $(m \times 1)$ vector of covariates, which can include exogenous variables z_t and lagged values of the dependent variable, y_{t-p}, $p > 0$.

We present three methods to deal with constructing density forecasts: assume a distribution for the errors and ignore parameter uncertainty; bootstrapping for accounting for parameter and error uncertainties in a frequentist environment; and Bayesian inference. The three methods rely on different assumptions. The first one requires to specify a distribution for a given model; the second one requires some assumptions and can be applied to any model that respects such assumptions; the third one requires prior information that are usually model dependent.

2.1 Distribution assumption

The easiest method to compute a density forecast is to assume a given distribution for the error term, e.g. $\varepsilon_t \sim N(0, \sigma^2)$ in (1), and to ignore parameter uncertainty. The $h-$step ahead density prediction, with $h > 1$, conditional to information available up to time T, $\mathscr{D}^T$, results to:

$$f(y_{T+h}|\mathscr{D}^T) = N(x'_{T+h}b, s^2) \tag{2}$$

where $b = (X^{'}X)^{-1}X^{'}y$, with $y = (y_1, \cdots, y_T)^{'}$ a $(T \times 1)$ vector, $X = (x_1, \cdots, x_T)^{'}$ a $(T \times m)$ matrix, and $s^2 = e'e/(T-m)$, with $e = (y - Xb)$. In the linear model (1) there is a closed form solution accounting for parameter uncertainty, see for example Hansen (2006). Simple modifications of that model have also closed form solution. For example, Clements and Galvao (2014) show how to compute the variance of the MIDAS predictive density to also account for parameter uncertainty.

The expression in (2) requires to know x_{T+h}. This is possible only in limited cases where the data generating process of X is known. In most cases, in particular when x_t includes also lags of y_t, this condition is not valid. There are several options to deal with it. For example, x_{T+h} can fixed to include only information up to time T, that is x_T is a function of exogenous $(z_1, \ldots, z_T)$ and lagged dependent $(y_T, \ldots, y_{T-p})$ variables. This strategy is often called direct forecasting and regressors in (1) should be changed accordingly. Otherwise, the system can be iterated to produce future values, that is x_{T+j} is computed conditional on x_{T+j-1} for $j = 1, \ldots, h$.

A special case is when x_t contains only lags of y_t. The density forecasts changes expression. The variable y_t can be expressed as a function of past errors and initial values as

$$y_t = \sum_{j=0}^{t-1} \phi_j \varepsilon_{t-j} + \pi, \ \varepsilon_t \sim \text{i.i.d.} N(0, \sigma^2),$$

where π summarizes the initial conditions. Assuming that the past errors and coefficients are known, the conditional expectation corresponds to the point forecast

$$y_{T+h} = \sum_{j=h}^{T-1} \phi_j \varepsilon_{T+h-j} + b\pi_0,$$

and the forecast error is $\sum_{j=0}^{h-1} \phi_j \varepsilon_{t+h-j}$. It follows that the forecast error variance is given by $s^2(h) = \sigma^2 \sum_{j=0}^{h-1} \phi_j^2$. The predictive density is therefore normally distributed with mean given by the usual point forecast and variance given by the above expression, $N(x'_{T+h}b, \sigma^2(h))$.

2.2 Bootstrapping

Ignoring parameter and distribution uncertainties can be very costly, in particular for small sample sizes and when the error distribution is not Gaussian, see Pascual et al. (2001). A solution to it is to apply a bootstrapping approach. The bootstrapping procedures are distribution-independent and account for parameter uncertainty.

Earlier studies in economics have mostly focused on bootstrapping in linear regressions and univariate autoregressions, see e.g., Berkowitz and Kilian (2000) and Clements and Taylor (2001). More recently, bootstrapping procedures for more advanced models have been proposed. These include models that deal with a large amount of data such as factor models, see e.g., Goncalves and Perron (2014), Djogbenou et al. (2015), Djogbenou et al. (2017), models with mixed frequency information, see Aastveit et al. (2014) and Mixed Data Sampling (MIDAS) models, see Aastveit et al. (2016).

2.2.1 A residual-based bootstrapping of density forecasts

We first consider a parametric residual-based bootstrap to derive forecast densities, accounting for both parameter and shock uncertainty as in Berkowitz and Kilian (2000) and Clements and Taylor (2001). The bootstrap procedure relies on the algorithm in Davison and Hinkley (1997) (Section 7.2.4) for prediction in generalized linear models. The residual-based bootstrap is valid under the following assumptions:

(A1) ε_t are i.i.d. with $E(\varepsilon_t) = 0$, $E(\varepsilon_t^2) = \sigma^2$ with $\sigma^2 < \infty$, and $E(\varepsilon_t^{2(s+1)}) < \infty$ for $s \geq 3$.

(A2) $(\varepsilon_1, \varepsilon_1^2)$ satisfies Cramer's condition, i.e., for every $d > 0$, there exists δ such that $sup_{||t||>d} |E \exp(it'(\varepsilon_1, \varepsilon_1^2))| \leq \exp(-\delta)$.

(A3) $x_{t_m+w-h_m}^{(m)}$ are exogenous fixed variables.

(A4) The process is stationary.

The steps conducted in the residual-based bootstrap are as follows.

1. Estimate equation (1), and obtain b.
2. For $r = 1,\dots,R$, simulate $\widetilde{y}_{r,t} = x_t b + \widetilde{e}_{r,t}$, where $\widetilde{e}_{r,t}$ is resampled from $\check{e}_t \equiv \left(\frac{n}{n-k}\right)^{0.5} e_t$.[1]
3. Re-estimate (1) for each $\widetilde{y}_{r,t}$, and obtain $\widetilde{y}_{r,T+h}$, where the shock uncertainty is included by resampling from $\check{e}_t$.

Davison and Hinkley (1997) fix the value of y_T equal to the value of the original series.

In practice, R vectors of pseudo-random numbers are generated to replicate the same properties of the residuals of the model, via the bootstrapping technique. For each $r = 1,\dots,R$ replications, a new set of simulated data is generated, and a new forecast $\widetilde{y}_{r,T+h}$ is obtained. The empirical distribution of $\left\{\widetilde{y}_{r,T+h}\right\}_{r=1}^{R}$ is then our density.

If the error terms in equation (1) are independent and identically distributed with common variance, then we can generally make very accurate inferences by using the residual bootstrap. Given the assumptions (A1)-(A4), Davison and Hinkley (1997) discuss how the method is a generalization of the bootstrapping algorithm for linear models and Bose (1988) provides proofs of its convergence.[2]

2.2.2 Accounting for autocorrelated or heteroskedastic errors

One limitation of the standard residual-based bootstrapping method above is that it treats the errors as i.i.d. The i.i.d. assumption does not follow naturally from economic models, and in many empirical applications the actual data are not well represented by models with i.i.d. errors, see e.g. Goncalves and Kilian (2004) and Davidson and MacKinnon (2006). Typically, economic and financial variables exhibit evidence of autocorrelation and/or conditional heteroskedasticity. In these cases, assumption **(A1)** is violated and the residual-based bootstrap is not valid.

Block bootstrap methods, suggested by Hall (1985) and Kunsch (1989), account for autocorrelated errors. The block bootstrap divides the quantities that are being re-sampled into blocks of b consecutive observations. The blocks can be either overlapping or non-overlapping, nevertheless Andrews (2002) finds small differences in performance between the two methods.

The wild bootstrap suggested by Wu (1986) and Liu (1988) is specifically designed to handle heteroskedasticity in regression models. Goncalves and Kilian (2004) have also shown that heteroskedasticity is an important feature in many

[1] Davidson and MacKinnon (2006) suggest to rescale the residuals so that they have the correct variance by $\check{e}_t \equiv \left(\frac{n}{n-k}\right)^{0.5} \widehat{e}_t$.

[2] Bose (1988) focuses on linear AR models with imposed stationarity (see assumption (A4) above). For an extension accounting for a possible unit root, see Inoue and Kilian (2002).

macroeconomic and financial series and apply the wild bootstrap to autoregressive models.

2.2.3 A block wild bootstrapping of density forecasts

To account for both autocorrelation and heteroskedasticity at the same time, we suggest using a block wild bootstrap, first proposed by Yeh (1998). Djogbenou et al. (2015, 2017) have recently proposed adapting the block wild bootstrap to the case of factor models and Aastveit et al. (2016) to to MIDAS models. Non-overlapping blocks of size n_T of consecutive residuals are formed. Assume that $(T-h)/n_T = k_T$, where k_T is an integer and denotes the number of blocks of size n_T. For $l = 1,\dots,b_T$ and $j = 1,\dots,k_T$, we let

$$y^*_{(j-1)n_T+l+h} = x_{(j-1)n_T+l}b + e^*_{(j-1)n_T+l+h}, \tag{3}$$

where

$$e^*_{(j-1)n_T+l+h} = \check{e}_{(j-1)n_T+l+h} \cdot \nu_j. \tag{4}$$

There are various ways to specify the distribution of ν_j. Davidson and Flachaire (2008) assume that ν_j is a Rademacher random variable

$$\nu_j = \begin{cases} 1 \text{ with probability } 1/2 \\ -1 \text{ with probability } 1/2 \end{cases} \tag{5}$$

Davidson and Flachaire (2008) study the wild bootstrap in the context of regression models with heteroskedastic disturbances and find that, among several popular candidates, this has the most desirable properties.

By replacing step 2 and 3 in the residual-based bootstrap above with the block wild bootstrap it is possible to accommodate both serial correlation and heteroskedasticity in $\widetilde{e}_{r,t}$. Djogbenou et al. (2017) set the block size equal to h.

2.3 *Bayesian inference*

A different approach to construct density forecasts rely on Bayesian inference. Bayesian analysis formulates prior distributions on parameters that multiplied by the likelihood results on parameter posterior distributions. Accounting for the uncertainty on parameter posterior distribution, future probabilistic statements derive without any further assumption. Moreover, prior distributions allow to impose restrictions on the parameters if useful and necessary. However, a user must specify prior statements before to start the analysis.

As example, we present the main derivation for model (1). The objective of Bayesian inference is to compute a predictive density:

$$f(y_{T+h}|\mathscr{D}^T) = \int p(y_{T+h}, X_{T+h}, \theta|\mathscr{D}^T)d\theta = \int l(y_{T+h}|X_{T+h}, \theta, \mathscr{D}^T)p(\theta|\mathscr{D}^T)d\theta \tag{6}$$

where $\mathscr{D}^T = (Y, X, X_{T+h})$ is the information set, $l(y_{T+h}|X_{T+h}, \theta)$ is the likelihood of the model for time $T+h$, $p(\theta|\mathscr{D}^T)$ is the parameter marginal distribution computed with information up to time T.

Regarding the choice of the prior distribution, if the prior is conjugate then the posterior and the predictive distribution can be computed analytically. If non-conjugate priors are used, then posterior and predictive are in integral form and need to be evaluated by means of numerical methods such as Monte Carlo simulation methods. In the regression model, in practice one usually defines $\tau = 1/\sigma^2$ and assumes a conjugate normal-gamma prior:

$$\beta|\tau \sim N(\underline{\beta}, \tau^{-1}\underline{V}), \quad \tau \sim G(\underline{s}^{-2}, \underline{\nu}), \quad \beta, \tau \sim NG(\underline{\beta}, \underline{V}, \underline{s}^{-2}, \underline{\nu})$$

where $\underline{\beta}$, $\underline{V}$, $\underline{s}^{-2}$ and $\underline{\nu}$ are parameters of the normal and gamma prior distributions. Define $\overline{V} = (V^{-1} + X'X)^{-1}$, $\overline{\beta} = \overline{V}(\underline{V}^{-1}\underline{\beta} + bX'X)$, $\overline{\nu} = \underline{\nu} + T$, $\overline{\nu s}^2 = \underline{\nu s}^2 + \nu s^2 + (b - \underline{\beta})'(\underline{V} + (X'X)^{-1})^{-1}(b - \underline{\beta})$, $\nu s^2 = (y - Xb)'(y - Xb)$, the conditional posteriors of β given σ^2 and σ^2 given β are:

$$p(\beta|\tau, y) \sim N(\overline{\beta}, \tau^{-1}\overline{V}), \quad p(\tau|\beta, y) \sim G(\overline{\nu s}^2, \overline{\nu})$$

See Koop (2003). The target is the marginal posterior distribution, that has a closed-form solution for model (1) and Normal-gamma priors:

$$\begin{aligned} \beta|\mathscr{D}^T &\sim t(\overline{\beta}, \overline{s}^2\overline{V}, \overline{\nu}) \\ \tau|\mathscr{D}^T &\sim G(\overline{\nu s}^2, \overline{\nu} - 2) \end{aligned}$$

The conditional and marginal predictive densities have also a closed-form solution, see Koop (2003):

$$f(y_{T+h}|\beta, \sigma, \mathscr{D}^T) \sim N(X_{T+h}\overline{\beta}, \overline{s}^2\widetilde{X}'\widetilde{X}) \tag{7}$$

$$f(y_{T+h}|\mathscr{D}^T) \sim t(X_{T+h}\overline{\beta}, \overline{s}^2(I_T + \widetilde{X}\overline{V}\widetilde{X}), \overline{\nu}) \tag{8}$$

As in the normal and bootstrapping cases, computation is more complex when X_{T+h} is not available at time T and direct forecasting is avoided. The algorithm in (6) generalizes to:

$$f(y_{T+h}|\mathscr{D}^T) = \int l(y_{T+h}|X_{T+h}, \theta)p(X_{T+h}|X_T, \theta)p(\theta|\mathscr{D}^T)d\theta \tag{9}$$

Closed-form solutions do not exist for most of economic models, but simulation methods can be used to compute the integral and derive the marginal predictive density. Assume a set of random samples θ_r, $r = 1, \ldots, R$ from $p(\theta|\mathscr{D}^T)$ is available, then the predictive density in equation (9) can be approximated as follows

$$\hat{f}_R(y_{T+h}|\mathscr{D}^T) = \frac{1}{R}\sum_{r=1}^{R} l(y_{T+h}|X_{T+h},\theta_r)p(X_{T+h}|X_T,\theta_r). \qquad (10)$$

See Section 5 for an introduction to simulation methods.

3 Density combinations

When multiple forecasts are available from different models or sources it is possible to combine these in order to make use of all relevant information on the variable to be predicted and, as a consequence, to produce better forecasts. This is particular important when working with large database and selection of relevant information *a priori* is not an easy task. Early papers on forecasting with model combinations are Barnard (1963), who considered air passenger data, and Roberts (1965) who introduced a distribution which includes the predictions from two experts (or models). This latter distribution is essentially a weighted average of the posterior distributions of two models and is similar to the result of a Bayesian Model Averaging (BMA) procedure. See Raftery et al. (1997) for a review on BMA, with a historical perspective. Raftery et al. (2005) and Sloughter et al. (2010) extend the BMA framework by introducing a method for obtaining probabilistic forecasts from ensembles in the form of predictive densities and apply it to weather forecasting. McAlinn and West (2018) extend it to Bayesian predictive synthesis.

Bates and Granger (1969) deal with the combination of predictions from different forecasting models using descriptive regression. Granger and Ramanathan (1984) extend this and propose to combine forecasts with unrestricted regression coefficients as weights. Terui and van Dijk (2002) generalize the least square weights by representing the dynamic forecast combination as a state space with weights that are assumed to follow a random walk process. Guidolin and Timmermann (2009) introduce Markov-switching weights, and Hoogerheide et al. (2010) propose robust time-varying weights and account for both model and parameter uncertainty in model averaging. Raftery et al. (2010) derive time-varying weights in “dynamic model averaging”, following the spirit of Terui and van Dijk (2002), and speed up computations by applying forgetting factors in the recursive Kalman filter updating.

A different line was started by Ken Wallis in several papers, see for example Wallis (2003), Wallis (2005), Wallis (2011) and Mitchell and Wallis (2011). Here the use of the full predictive distribution is proposed when forecasting. Benefits and problems related to it are discussed in detail. One focus has been to measure to the importance of density combinations. Hall and Mitchell (2007) introduce the Kullback-Leibler divergence as a unified measure for the evaluation and suggest weights that maximize such a distance, see also Amisano and Geweke (2010) and Geweke and Amisano (2011) for a compresive discussion on how such weights are robust to model incompleteness, that is the true model is not included in the model set. Gneiting and Raftery (2007) recommend strictly proper scoring rules, such as the cumulative rank probability score. Billio et al. (2013) develops a general method that can deal with most of issues discussed above, including time-variation in combination weights, learning from past performance, model incompleteness, correlations among weights and joint combined predictions of several variables. See, also Waggoner and Zha (2012), Kapetanios et al. (2015), Pettenuzzo and Ravazzolo (2016), Aastveit et al. (2018) and Del Negro et al. (2016).

We refer to Aastveit et al. (2019) for a recent survey on the evolution of forecast density combinations in economics. In the following we provide some details on two basic methodologies, the Bayesian Model Averaging (BMA) and the linear opinion pool (LOP), and discuss briefly some extensions.

3.1 Bayesian model averaging

Let $\mathscr{D}^T$ be the set of information available up to time t, then BMA combines the individual forecast densities $f(Y_{T+h}|\mathscr{D}^T, M_j)$, $i = 1, \dots, N$, into a composite-weighted predictive distribution $f(Y_{T+h}|\mathscr{D}^T)$ given by

$$f(Y_{T+h}|\mathscr{D}^T) = \sum_{j=1}^{N} P\left(M_j \middle| \mathscr{D}^T\right) f(Y_{T+h}|\mathscr{D}^T, M_j) \tag{11}$$

where $P\left(M_j \middle| \mathscr{D}^T\right)$ is the posterior probability of model j, derived by Bayes' rule,

$$P\left(M_j \middle| \mathscr{D}^T\right) = \frac{P\left(\mathscr{D}^T \middle| M_j\right) P(M_j)}{\sum_{j=1}^{N} P(\mathscr{D}^T | M_j) P(M_j)}, \qquad j = 1, \dots, N \tag{12}$$

and where $P(M_j)$ is the prior probability of model M_j, with $P\left(\mathscr{D}^T \middle| M_j\right)$ denoting the corresponding marginal likelihood. We shall notice that the model posterior probability can be written in terms of Bayes factors

$$P\left(M_j \middle| \mathscr{D}^T\right) = \frac{\alpha_j B_{1j}}{\sum_{j=2}^{N} \alpha_j B_{1j}} \tag{13}$$

where $\alpha_j = P(M_j)/P(M_1)$ and $B_{1j} = P\left(\mathscr{D}^T \middle| M_j\right) / P\left(\mathscr{D}^T \middle| M_1\right)$, $j = 2, \dots, N$ are the Bayes factors. An alternative averaging weighting scheme can be define by using the predictive distributions:

$$P\left(M_j \middle| \mathscr{D}^T\right) = \frac{P\left(Y_T \middle| \mathscr{D}^{T-1}, M_j\right) P(M_j)}{\sum_{j=1}^{N} P(Y_T | \mathscr{D}^{T-1}, M_j) P(M_j)}, \qquad j = 1, \dots, N. \tag{14}$$

3.2 Linear opinion pool

LOP gives a predictive density $f(y_{T+h}|\mathscr{D}^T)$ for the variable of interest to be predicted at horizon $T+h$ with $h>0$, y_{T+h}, using the information available up to time T, $\mathscr{D}^T$, from a set of predictions generated by the models M_j, $j=1,\dots,N$.

$$f(y_{T+h}|\mathscr{D}^T)=\sum_{j=1}^{N} w_{j,T+h} f(y_{T+h}|\mathscr{D}^T,M_j) \tag{15}$$

where $w_{j,T+h}$ is the $(0,1)$-valued weight given to model M_j computed at time T and $f(y_{T+h}|\mathscr{D}^T,M_j)$ is the density forecast of y_{T+h} conditional on predictor M_j, and on the information available up to time T. The individual prediction can be model based, parametric or non-parametric, or individual subjective predictions. Each of these predictive densities must be non-negative for all the support of y_{T+h} and their cumulative density functions must add to 1. To guarantee that the combined forecast density $f(y_{T+h})$ also satisfies these features, some restrictions can be imposed to the combination weights $w_{j,T+h}$, $j=1,\dots,N$. Sufficient conditions are that weights are non-negative, $w_{j,T+h}\geq 0$, $j=1,\dots,N$, and that add to unity, $\sum_{j=1}^{N} w_{j,T+h}=1$.

Standard practice, see for example Hall and Mitchell (2007), Kascha and Ravazzolo (2010) and Mazzi et al. (2014), is to use the cumulative log score, see equation (26). The combination weights are computed as

$$w_{j,T+h}^{LS}=\frac{\exp(\eta_{j,T}^{LS})}{\sum_{j=1}^{N}\exp(\eta_{j,T}^{LS})} \tag{16}$$

where $\eta_{j,T}^{LS}$ is the cumulative log score for model M_j at time T. We note that at time T when predictions are made, the cumulative log score can be computed up to the same time and therefore weights are based on the statistic $\eta_{j,T}^{LS}$, $j=1,\dots,N$. Such statistic contains information on how the predictor M_j associated to prediction $f(y_{T+h}|\mathscr{D}^T,M_j)$ has performed in the past in terms of forecasting. Therefore, the major difference between LOP and BMA is in weights definition. In LOP, weights are computed using some statistics; in BMA weights depend on model posterior probabilities.

3.3 Generalized opinion pool

Following the notation used in Gneiting and Ranjan (2013b), it is possible to define a general pooling method as a parametric family of combination formulas. Let $F_{jT}(y_{T+h})=F(_{T+h}|\mathscr{D}^T,M_j)$ denote the cdf of the density $f(y_{T+h}|\mathscr{D}^T,M_j)$, a generalized pool is a map

$$H : \left[\begin{array}{l} \times^N \mathscr{F} \to \mathscr{F} \\ (F_{1T}(\cdot), \cdots, F_{NT}(\cdot)) \mapsto F(\cdot|\xi, \mathscr{D}^T) = H(F_{1T}(\cdot), \ldots, F_{NT}(\cdot), \xi) \end{array} \right.$$

indexed by the parameter $\xi \in \Xi$, where Ξ is a parameter space and $\mathscr{F}$ is a suitable space of distributions. Following (see DeGroot and Mortera, 1991; DeGroot et al., 1995) we consider pooling scheme of the form:

$$H(F_{1T}(\cdot), \ldots, F_{NT}(\cdot), \xi) = \varphi^{-1} \left(\sum_{j=1}^{N} \omega_j \varphi(F_{jT}(\cdot) \right) \tag{17}$$

where φ is a continuous increasing monotone function with inverse φ^{-1} and $\xi = (\omega_1, \cdots, \omega_N)'$ is a vector of combination weights, with $\omega_1 + \ldots + \omega_N = 1$ and $\omega_j \geq 0$, for all i. If $\varphi(x) = x$ then we obtain the Linear Opinion Pool

$$F(y_{T+h} vert \mathscr{D}^T, \xi) = \sum_{j=1}^{N} \omega_j F(y_{T+h}|\mathscr{D}^T, M_j) \tag{18}$$

The harmonic opinion pool is obtained for $\varphi(x) = 1/x$

$$F(y_{T+h}|\mathscr{D}^T, \xi) = \left(\sum_{j=1}^{N} \omega_j F(y_{T+h}|\mathscr{D}^T, M_j)^{-1} \right)^{-1} \tag{19}$$

whereas by choosing $\varphi(x) = \log(x)$ one obtains the logarithmic opinion pool

$$F(y_{T+h}|\mathscr{D}^T, \xi) = \prod_{j=1}^{N} F(y_{T+h}|\mathscr{D}^T, M_j)^{\omega_j}. \tag{20}$$

If φ is differentiable then the generalized combination model can be re-written in terms of pdf as follows

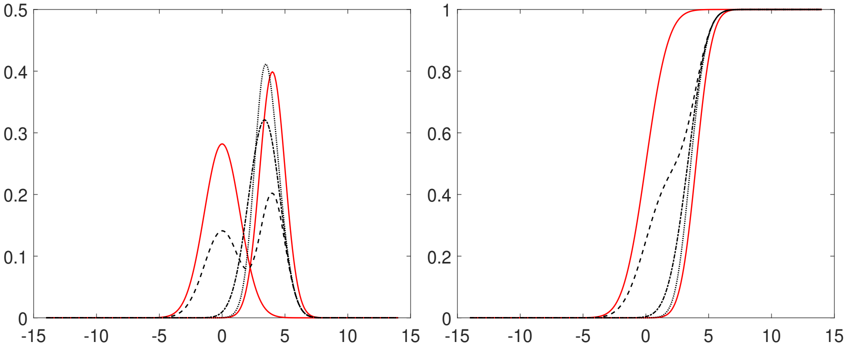
$$f(y_{T+h}|\mathscr{D}^T, \xi) = \frac{1}{\varphi'(F(y_{T+h}|\mathscr{D}^T, \xi))} \sum_{j=1}^{N} \omega_j \varphi'(F(y|\mathscr{D}^T, M_j)) f(y|\mathscr{D}^T, M_j) \tag{21}$$

where φ' denotes the first derivative of φ. The related density functions are:

$$\begin{aligned} f(y_{T+h}|\mathscr{D}^T, \xi) &= \sum_{j=1}^{N} \omega_j f(y_{T+h}|\mathscr{D}^T, M_j) \\ f(y_{T+h}|\mathscr{D}^T, \xi) &= F(y_{T+h}|\mathscr{D}^T, \xi)^2 \sum_{j=1}^{N} \omega_j F(y_{T+h}|\mathscr{D}^T, M_j)^{-2} f(y_{T+h}|\mathscr{D}^T, M_j) \\ f(y_{T+h}|\mathscr{D}^T, \xi) &= F(y_{T+h}|\mathscr{D}^T, \xi) \sum_{j=1}^{N} \omega_j F(y_{T+h}|\mathscr{D}^T, M_j)^{-1} f(y_{T+h}|\mathscr{D}^T, M_j) \end{aligned}$$

for the linear opinion pool, harmonic opinion pool and logarithmic opinion pool, respectively. Generalized combination schemes have developed further in Kapetanios et al. (2015) and Bassetti et al. (2018). We illustrate the three combination methods by assuming that two density forecasts are available, $F(y_{T+h}|\mathscr{D}^T, M_1) \sim N(4,1)$ and $F(y_{T+h}|\mathscr{D}^T, M_2) \sim \mathscr{N}(0,2)$, and a equally weighted pooling is used ($\omega_1 = \omega_2 = 0.5$). From Fig. 1 one can see that harmonic and logarithmic pools concentrate the probability mass on one of the model in the pool.

Fig. 1 Pdfs (left) and cdfs (right) of the two forecasting models $F(y|\mathscr{D}^T, M_1) \sim N(4,1)$ and $F(y|\mathscr{D}^T, M_2) \sim N(0,2)$ (red solid lines) and of their linear (dashed), harmonic (dotted) and logarithmic (dotted-dashed) combination.



4 Density forecast evaluation

The density of the variable of interest y_{T+h} at given time $T+h$ is never observed. This complicates the evaluation of density forecasts. In economics, there are two main approaches to evaluate density forecasts. The first one is based on properties of a density and refers to absolute accuracy. The second one is based on comparison of different forecasts and refers to relative accuracy.

4.1 Absolute accuracy

The absolute accuracy can be studied by testing forecast accuracy relative to the "true" but unobserved density. Dawid (1982) introduced the criterion of *complete calibration* for comparing prequential probabilities with binary random outcomes. This criterion requires that the averages of the prequential probabilities and of the binary outcomes converges to the same limit. For continuous random variables Dawid (1982) exploited the concept of probability integral transform (PIT) that is the value that a predictive cdf attains at the observations. The PITs summarize the properties of the densities and may help us judge whether the densities are biased in a particular direction and whether the width of the densities has been roughly correct on average Diebold et al. (1998). More precisely, the PITs represents the ex-ante inverse predictive cumulative distributions, evaluated at the ex-post actual observations. The PIT at time T are:

$$PIT_{T+h} = \int_{-\infty}^{y_{T+h}} f(y|\mathscr{D}^T)dy \tag{22}$$

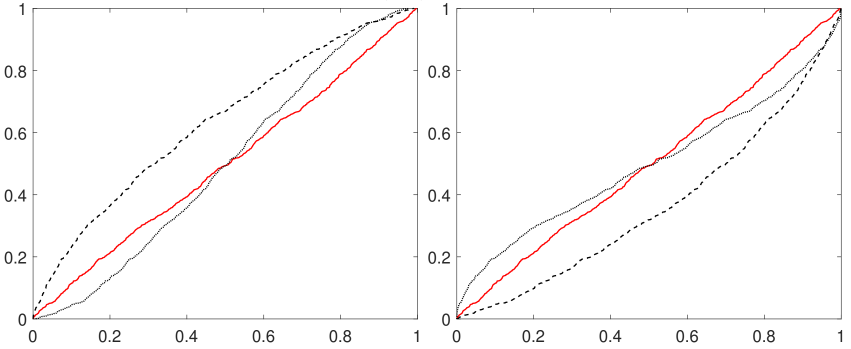
and should be uniformly, independently and identically distributed if the h-step-ahead forecast densities $f(y_{T+h}|\mathscr{D}^T)$ conditional on the information set available at time T, are correctly calibrated.

As an example assume that a set of observations are generated from a standard normal, $Y_t \sim N(0,1)$, i.i.d. $t = T+1,\ldots,T+1000$ and that four predictive cdfs are used:

$$\begin{array}{ll} F(y_{T+h}|\mathscr{D}^T, M_1) \sim N(0.5,1), & F(y_{T+h}|\mathscr{D}^T, M_2) \sim N(0,2) \\ F(y_{T+h}|\mathscr{D}^T, M_3) \sim N(-0.5,1), & F(y_{T+h}|\mathscr{D}^T, M_4) \sim N(0,0.5) \end{array}$$

The first model is wrong in predicting the mean of the distribution, the second one is wrong in predicting the variance. In Fig. 2, which show the cdfs of PITs. In each plot the red line indicates the PITs of the true model. Errors in mean induce a cdf that overestimate (left plot) or underestimate (right plot), depending on error sign, the "true" cumulative density function. Variance overestimation appears as an underestimate in the left side of the distribution, and an overestimate in the right side, whereas variance underestimation appears as an overestimate in the left side of the distribution, and an underestimate in the right side. In both cases, the discontinuity point corresponds at the mean, in which the two line intersect.

Fig. 2 Empirical cdfs of the PITs. Left: PITs generated by $F(y_{T+h}|\mathscr{D}^T, M_1) \sim N(0.5,1)$ (dashed line), $F(y_{T+h}|\mathscr{D}^T, M_2) \sim N(0,2)$ (dotted line). Right: PITs generated by $F(y_{T+h}|\mathscr{D}^T, M_3) \sim N(-0.5,1)$ (dashed line), $F(y_{T+h}|\mathscr{D}^T, M_4) \sim N(0,0.5)$ (dotted line). In each plot the red solid line indicates the PITS of the true model ($N(0,1)$).



Calibration can be gauge by testing jointly for uniformity and (for one-step ahead forecasts) independence of the PITs, applying the tests proposed by Berkowitz (2001) and Knuppel (2015).[3] Rossi and Sekhposyan (2013) extend the evaluation in the presence of instabilities; Rossi and Sekhposyan (2014) apply to large database and Rossi and Sekhposyan (2016) compare alternative tests for correct specification of density forecasts.

4.2 Relative accuracy

When moving to relative comparison, density forecasts can be evaluated by the Kullback Leibler Information Criterion (KLIC)-based measure, utilising the expected difference in the Logarithmic Scores of the candidate forecast densities; see, for example, Mitchell and Hall (2005), Hall and Mitchell (2007), Kascha and Ravazzolo (2010) and Billio et al. (2013). The KLIC is the distance between the true density $p(y_{T+h}|\mathscr{D}^T)$ of a random variable y_{T+h} and some candidate density $f(y_{T+h}|\mathscr{D}^T, M_j)$ obtained from the model M_j and chooses the model that on average gives the higher probability to events that actually occurred. An estimate of it can be obtained from the average of the sample information, $y_{\underline{T}+1}, \ldots, y_{\overline{T}+1}$, on $p(y_{T+h})$ and $f(y_{T+h}|\mathscr{D}^T, M_j)$:

$$\overline{KLIC}_{j,h} = \frac{1}{T^*} \sum_{T=\underline{T}}^{\overline{T}} [\ln p(y_{T+h}|\mathscr{D}^T) - \ln f(y_{T+h}|\mathscr{D}^T, M_j)] \tag{23}$$

where $T^* = (\overline{T} - \underline{T} + 1)$. Although we do not know the true density, we can still compare different densities, $f(y_{T+h}|M_j)$. For the comparison of two competing models, it is sufficient to consider the Logarithmic Score (LS) given as:

$$LS_{j,h} = -\frac{1}{T^*} \sum_{T=\underline{t}}^{\overline{T}} \ln f(y_{T+h}|\mathscr{D}^T, M_j) \tag{24}$$

for all j and choose the model for which this score is minimal.

Alternative, density forecasts can be evaluated on the continuous rank probability score (CRPS); see, for example, Gneiting and Raftery (2007), Gneiting and Ranjan (2013a), Groen et al. (2013) and Ravazzolo and Vahey (2014). The CRPS for the model j measures the average absolute distance between the empirical cumulative distribution function (CDF) of y_{T+h}, which is simply a step function in y_{T+h}, and the empirical CDF that is associated with model j's predictive density:

$$\text{CRPS}_{j,T+h} = \int_{-\infty}^{+\infty} \left(F(y|\mathscr{D}^T, M_j) - \mathbb{I}_{[Y_{T+h},+\infty)}(y) \right)^2 \text{d}y \tag{25}$$

[3] For longer horizons, test for independence is skipped.

where F is the CDF from the predictive density $f(y_{t+h}|\mathscr{D}^T, M_j)$ of model j. The sample average CRPS is computed as:

$$\mathrm{CRPS}_{j,h} = -\frac{1}{T^*}\sum_{T=\underline{t}}^{\overline{T}} \mathrm{CRPS}_{j,T+h} \tag{26}$$

Smaller CRPS values imply higher precisions.

Finally, the Diebold and Mariano (1995) and West (1996) t-tests for equality of the average loss (with loss defined as log score, or CRPS) can be applied.

4.3 Forecast calibration

An expert is well-calibrated if the subjective predictive distribution (or density function) agrees with the sample distribution of the realizations of the unknown variable in the long run. When a predictive density $F(y|\mathscr{D}^T)$ is not well-calibrated, a calibration procedure can be applied, by introducing introducing a monotone non-decreasing map

$$\psi : \begin{bmatrix} [0,1] \rightarrow [0,1] \\ F(\cdot|\mathscr{D}^T,\xi) \mapsto F(\cdot|\mathscr{D}^T,\xi) = \psi(F(\cdot|\mathscr{D}^T)) \end{bmatrix} \tag{27}$$

such that $F(y_{T+h}|\mathscr{D}^T,\xi)$ is well calibrated. Bassetti et al. (2018) propose to use the cdf of a mixture of Beta II distributions as calibration functional, that is

$$F(y_{T+h}|\mathscr{D}^T,\xi) = \sum_{j=1}^{J} B_{\alpha_j,\beta_j}(F(y_{T+h}|\mathscr{D}^T)) \tag{28}$$

with $\xi = (\alpha_1,\ldots,\alpha_J,\beta_1,\ldots,\beta_J,\omega_1,\ldots,\omega_J)$ and $\alpha_j,\beta_j > 0$, $\omega_1+\ldots+\omega_J = 1$, $\omega_j \geq 0$ and $B_{\alpha,\beta}(u)$ the cdf of the Beta II distribution. This calibration functional has the beta calibration scheme of Ranjan and Gneiting (2010) and Gneiting and Ranjan (2013b) as special case for $J = 1$ and allows for more flexibility in calibrating in presence of fat tails, skewness and multiple-modes.

As an example assume that a set of observations are generated from a standard normal, $Y_t \sim N(0,1)$, i.i.d. $t = T+1,\ldots,T+n$, $n = 1000$ and that the predictive density results from the following linear pooling:

$$F(y_{T+h}|\mathscr{D}^T) \sim \frac{1}{3}N(0.5,1) + \frac{1}{3}N(0,2) + \frac{1}{3}N(-0.5,1).$$

Since the PITs of the density forecasts are not well-calibrated (dashed line, in the left panel of Fig. 3), we apply the following calibration functions:

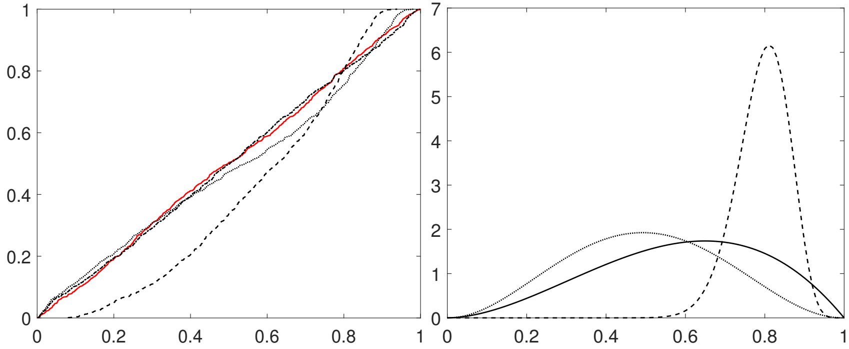
$$F(_{T+h}|\mathscr{D}^T,\xi) = B_{\alpha,\beta}(F(y_{T+h}|\mathscr{D}^T))$$
$$F(y_{T+h}|\mathscr{D}^T,\xi) = \omega B_{\alpha_1,\beta_1}(F(y_{T+h}|\mathscr{D}^T)) + (1-\omega)B_{\alpha_2,\beta_2}(F(y_{T+h}|\mathscr{D}^T))$$

where the parameters $\alpha = 2.81$ and $\beta = 2.01$ and $\alpha_1 = 23.13$, $\beta_1 = 6.61$, $\alpha_2 = 2.95$, $\beta_2 = 3.19$ and $\omega = 0.36$ have been optimally chosen by maximizing the likelihood function

$$L(Y^{T+n}|\xi) = \prod_{t=T+1}^{T+n} \left(\sum_{j=1}^{J} B_{\alpha_j,\beta_j}(F(Y_t|\mathscr{D}^T)) \right) \tag{29}$$

with respect to ξ. For a Bayesian approach to the estimation of the calibration function see Bassetti et al. (2018). The dashed line in the left panel suggests that the beta calibration model is not able to produce well-calibrated PITs, whereas the 2-component beta mixture functional (dotted-dashed line) allows for a better calibration. The first mixture component $B_{\alpha_1,\beta_1}(u)$ for $u \in (0,1)$ (dotted line in the right plot) is calibrating all the PITs, whereas the second component $B_{\alpha_2,\beta_2}(u)$ (dashed line) is reducing the value of the PITs below the 60%. A Bayesian approach to inference on the calibration functional can carried out by eliciting a prior on the parameter ξ and then using Markov-chain Monte Carlo methods for posterior simulation (e.g., see Robert and Casella, 2004). As an example consider the beta calibration exercise of this section. We assume $\alpha,\beta \sim Ga(2,4)$ where $Ga(c,d)$ is a gamma distribution with shape and scale parameters c and d, respectively and pdf

Fig. 3 PITs calibration exercise. Left: PITs generated by the true model (red solid), the forecasting model $F(y|\mathscr{D}^T)$ (dashed), the beta calibrated model (dotted) and the beta mixture calibrated model (dashed-dotted). Right: beta calibration function (solid) and the first (dashed) and second (dotted) component of the beta mixture calibration function.



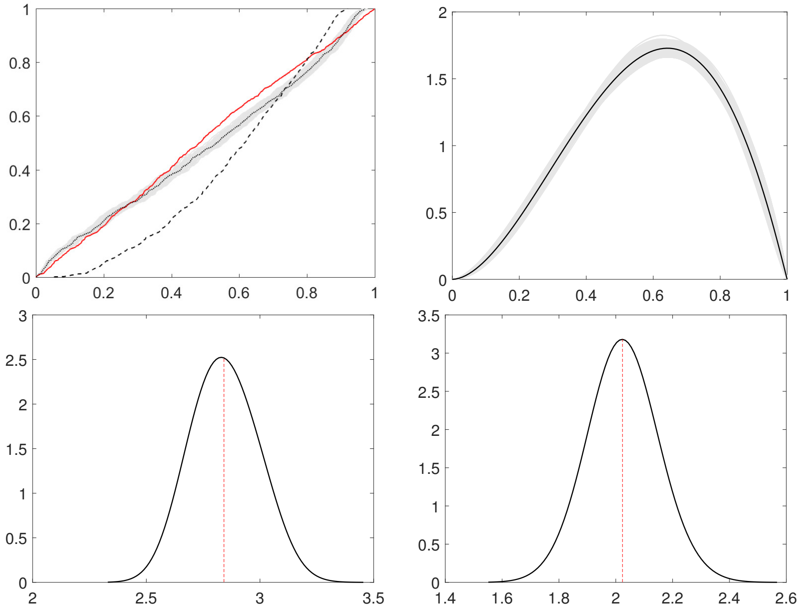
$$p(z) = \frac{1}{\Gamma(c)} c^{-d} \exp\left(-\frac{1}{d}z\right) z^{c-1}, \quad z > 0. \tag{30}$$

Let $p(\xi) = p(\alpha)p(\beta)$ be the joint prior with $\xi = (\alpha, \beta)$. The joint posterior distribution

$$p(\xi|\mathscr{D}^{T+n}) \propto L(Y^{T+n}|\xi)p(\xi) \tag{31}$$

is not tractable, thus we apply a Metropolis-Hastings simulation algorithm (see Section 5) which generates at the iteration r a candidate ξ^* from the random walk proposal $\log\xi^* = \log\xi + \eta_{r-1}$, $\eta_t \sim N_2(0, \mathrm{diag}\{0.05, 0.05\})$, where η_{r-1} is the previous iteration random sample from the simulation algorithm. The MH samples are used to estimate the posterior distribution of the calibrated PITs (left plot in Fig. 4) and the calibration parameters (right plot).

Fig. 4 PITs calibration exercise. Left: PITs generated by the true model (red solid), the forecasting model $F(y|\mathscr{D}^T)$ (dashed), the Bayesian beta calibrated model (dotted) and the MCMC posterior coverage (light gray lines). Right: beta calibration function (solid), he MCMC posterior coverage (light gray lines), posterior mean (vertical dashed).



5 Monte Carlo methods for predictive approximation

In the next sections, we report some Monte Carlo (MC) simulation methods which can be used for approximating predictive densities expressed in integral form. MC simulation is an approximation method to solve numerically several optimization and integration problems, and already found widespread application in economics and business, e.g., see Kloek and van Dijk, 1978 and Geweke, 1989.

5.1 Accept-reject

The Accept-Reject (AR) algorithm (Robert and Casella, 2004) is used to generate samples from a density $f(y)$ (called target density) by using an density $g(y)$ (called instrumental density). The AR algorithm iterates the following steps for $r = 1, \dots, R$

1. Generate X_r from g and a uniform U_r from $U_{[0,1]}$, .
2. Accept and set $Y_r = X_r$ if $U_r \leq f(X_r)/Cg(X_r)$

As an example consider the target density

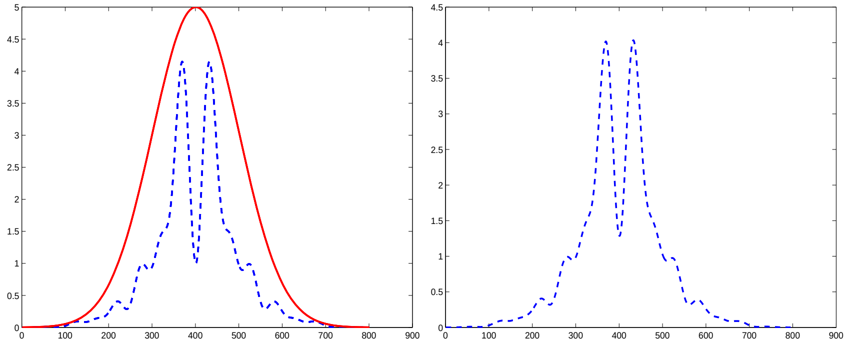
$$f(x) \propto \exp(-x^2/2)(\sin(6x)^2 + 3\cos(x)^2 \sin(4x)^2 + 1), \tag{32}$$

which is not easy to simulate, and assume the following instrumental density

$$g(x) \propto \exp(-x^2/2)/\sqrt{2\pi}, \tag{33}$$

which is the density of a standard normal distribution and is easier to simulate. The top panel in Figure 5 reports a graphical comparison of the two densities. The bottom panel of Figure 5 shows the simulated target density using the AR algorithm based on 1,000,000 draws. See Listing 3 in the Appendix for the MATLAB code.

Fig. 5 Accept-Reject example. Left: target (dashed) and instrumental (solid) density. Right: target histogram approximated with 1,000,000 draws.



5.2 Importance sampling

Let $f(y)$ be a target density function, h a measurable function and

$$\mathfrak{I} = \int h(y) f(y) dy \tag{34}$$

the integral of interest. In importance sampling (IS) (see Robert and Casella, 2004, chapter 3)) a distribution g (called importance distribution or instrumental distribution) is used to apply a change of measure

$$\mathfrak{I} = \int \frac{f(y)}{g(y)} h(y) g(y) dy. \tag{35}$$

The resulting integral is then evaluated numerically by using i.i.d. samples $Y_1, \ldots, Y_R$ from g, that is

$$\mathfrak{I}_R^{IS} = \frac{1}{R} \sum_{r=1}^{R} w(Y_r) h(Y_r) \tag{36}$$

where $w(Y_r) = f(Y_r)/g(Y_r)$, $r = 1, \ldots, R$ are called importance weights. A set of sufficient conditions for the IS estimators to have finite variance is the following:

(B1) $f(y)/g(y) < M \; \forall y \in \mathscr{Y}$ and $\mathbb{V}_f(h) < \infty$

(B1) $\mathscr{Y}$ is compact, $f(y) < C$ and $g(y) > \varepsilon \; \forall y \in \mathscr{Y}$.

The condition **(B1)** implies that the distribution g has thicker tails than f. If the tails of the importance density are lighter than those of the target then the importance weight $w(Y)$ is not a.e. bounded and the variance of the estimator will be infinite for many functions h. A way to address this issue is to consider the self-normalized importance sampling (SNIS) estimator

$$\mathfrak{I}_R^{SNIS} = \frac{\sum_{r=1}^{R} w(Y_r) h(Y_r)}{\sum_{r=1}^{R} w(Y_r)} \tag{37}$$

It is biased on a finite sample, but it converges to $\mathfrak{I}$ by the strong law of large number.

As an example let $h(y) = \sqrt{|y/(1-y)|}$ and y follow a Student-t distribution $\mathscr{T}(\nu, \theta, \sigma^2)$ with density

$$f(y) = \frac{\Gamma((\nu+1)/2)}{\sigma\sqrt{\nu\pi}\Gamma(\nu/2)} \left(1 + \frac{(y-\theta)^2}{\nu\sigma^2}\right)^{-(\nu+1)/2} \mathbb{I}_{\mathbb{R}}(y). \tag{38}$$

We study the performance of the importance sampling estimator when the following instrumental distributions are used:

1. Student-*t*, $t(\nu^*, 0, 1)$ with $\nu^* < \nu$ (e.g. $\nu^* = 7$);
2. Cauchy, $C(0, 1)$.

We shall recal that the Cauchy distribution $\mathscr{C}(\alpha, \beta)$ has density function

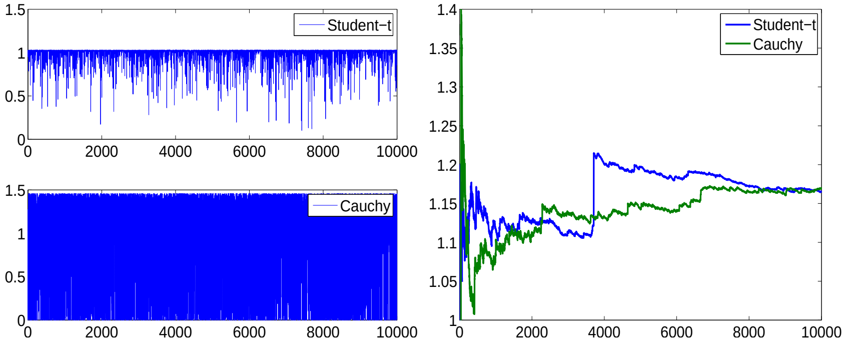
$$g(y) = \frac{1}{\pi\beta(1+((y-\alpha)/\beta)^2)}\mathbb{I}_{\mathbb{R}}(y),$$

where $-\infty < \alpha < +\infty$ and $\beta > 0$ and cumulative distribution function

$$G(y) = \left(\frac{1}{2} + \frac{1}{\pi}\arctan\frac{y-\alpha}{\beta}\right)\mathbb{I}_{\mathbb{R}}(y).$$

The inverse cdf method can be applied in order to generate from the Cauchy: if$Y = G^{-1}(U)$, where $U \sim \mathscr{U}_{[0,1]}$, then $Y \sim \mathscr{C}(\alpha,\beta)$. See Listing 4 in Appendix for a MATLAB code. We generate 10000 draws from the instrumental distributions. Figure 6 shows that the importance weights for Student-t and Cauchy are stable (left panel), but the Cauchy proposal seems to converge faster than the Student-t (right panel).

Fig. 6 Importance sampling draws for the two different instrumental distributions. Left: importance sampling weights $w(Y_j)$. Right: importance sampling estimator.



5.3 Metropolis-Hastings

In IS and AR samples from a target distribution can be generated by using a different distribution. A similar idea motivates the use of Markov chain Monte Carlo methods, where samples are generated from an ergodic Markov chain process with the target as a stationary distribution. A general MCMC method is the Metropolis-Hastings (MH) algorithm. Let $f(y)$ be the target distribution and $q(x|y)$ a proposal distribution. The MH algorithm (see Ch. 6-10 in Robert and Casella, 2004) generates a sequence of samples $Y_1,\ldots,Y_R$ by iterating the following steps. At the r-th iteration, given Y_{r-1} from the previous iteration:

1. generate $X^* \sim q(x|Y_{r-1})$;
2. set

$$Y_r = \begin{cases} X^* & \text{with probability } \quad \alpha(X^*, Y_{r-1}) \\ Y_{r-1} & \text{with probability } 1 - \alpha(X^*, Y_{r-1}) \end{cases}$$

where

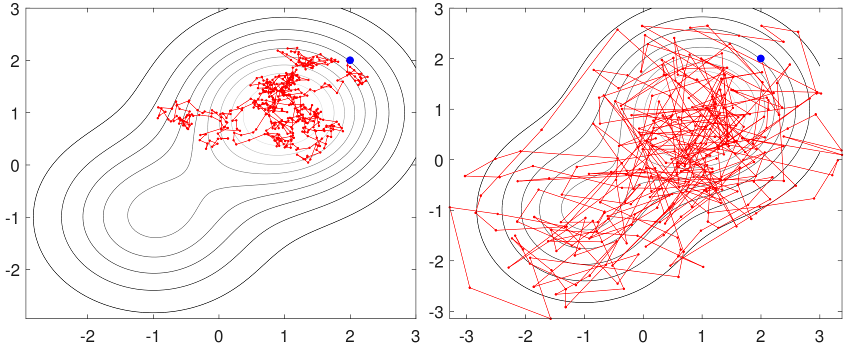
$$\alpha(x,y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1\right\}$$

The generality of the MH relies on the assumption that the target density is known up to a normalizing constant, which is common in many Bayesian inference problems. A drawback of the MH method is that the sequence of samples is not independent and the degree of dependence depends on the choice of the proposal distribution. In order to illustrate this aspect, we consider a toy example. Assume the target distribution is a bivariate normal mixture $1/3N_2(-\iota, I_2) + 2/3N_2(\iota, I_2)$ where $\iota = (1,1)'$ and I_2 is the 2-dimensional identity matrix and design a random-walk MHalgorithm with candidate samples X^* generated from $N_2(Y_{r-1}, \tau^2 I_2)$.

Fig. 7 shows the output of 500 iterations of the MH sampler for different values of the scale parameter τ (different panels). In each plot, the 2-dimensional random vectors Y_r, $r = 1, \ldots, 500$ (red dots), the trajectory of the M.-H. chain (red line connecting the dots), the initial value of the algorithm (blue dot) and the level sets of the target distribution (solid balck lines).

Left plot shows an example of missing mass problem. The scale of the proposal is too small ($\tau^2 = 0.01$), thus the M.-H.chain gets trapped by one of the mode and is not able to visit the other mode. In this case one expects that the results of the approximated inference procedure are sensitive to the choice of the initial condition of the MH chain. The MH chain in the right plot has a better mixing and is able to generate samples from the two components of the mixture.

Fig. 7 Output of the Metropolis-Hastings for different choices of the random walk scale parameter, $\tau^2 = 0.01$ (left) and $\tau = 1$ (right). In each plot: the trajectory of the M.-H. chain (red line), the initial value of the algorithm (blue dot) and the level sets of the target distribution (solid balck lines).



5.4 Constructing density forecasting using GPU

There is a recent trend in using Graphical Processor Units (GPUs) for general, non-graphics, applications (prominently featuring those in scientific computing) the so-called General-Purpose computing on Graphics Processing Units (GPGPU). GPGPU has been applied successfully in different fields such as astrophysics, biology, engineering, and finance, where quantitative analysts started to use this technology ahead of academic economists, see Morozov and Mathur (2011) for a literature review. To date, the adoption of GPU computing technology in economics and econometrics has been relatively slow compared to other fields. There are just a few papers that deal with this interesting topic (e.g., see Morozov and Mathur, 2011; Geweke and Durham, 2012; Durham and Geweke, 2014; Casarin et al., 2015; Vergé et al., 2015; Casarin et al., 2016). This is odd given the fact that parallel computing in economics has a long history and specifically for this paper computing density forecasts based on bootstrapping or Bayesian inference requires extensive computation that can be paralleled. The low diffusion of this technology in the economics and econometrics literature, according to Creel (2005), is related to the steep learning curve of a dedicated programming language and expensive hardware. Modern GPUs can easily solve the second problem (hardware costs are relatively low), but the the first issue still remains open. Amont the popular softwares used in econometrics (e.g., see LeSage, 1998), MATLAB has introduced from the version R2010b the support to GPU computing in its parallel computing toolbox. This allows for using raw CUDA code within a MATLAB code and MATLAB functions that are executed on the GPU. See Geweke and Durham (2012) for a discussion about CUDA programming in econometrics. As showed in the Appendix, using the build-in functions, GPGPU can be almost effortless where the only knowledge required is a decent programming skill in MATLAB.

6 Conclusion

This paper reviews different methods to construct density forecasts based on error assumptions, bootstrapping and Bayesian inference. We describe different assumptions of the three methods in the case of the simple linear regression models and provide tools to extend the analysis to more complex models. We also discuss density combinations as a tool to deal in the case there are several density forecasts and an *a priori* selection is difficult. And we provide some evaluation tools to measure the accuracy of density forecasts, accounting for the fact that the “true” density forecast is never observed, even *ex post*.

As example, we present how to use GPU computing almost effortless with MATLAB. The only knowledge required is a decent programming skill and a knowledge of the GPU computing functions introduced in the MATLAB parallel computing toolbox. We generate random numbers, estimate a linear regression model and present a Monte Carlo simulation based on accept/rejection algorithm. We expect large benefits in computational time when dealing with big database with GPU computing.

Appendix

There is little difference between a CPU and a GPU MATLAB code as listings 1 and 2, for example, show. The pseudo code, reported in the listings, generates random variables Y and X and estimates the linear regression model $Y = X\beta + \varepsilon$, on CPU and GPU, respectively.

The GPU code, Listing 2, uses the command gpuArray.randn to generate a matrix of normal random numbers. The build-in function is handled by the NVIDIA plug-in that generates the random number with an underline raw CUDA code. Once the variables *vY* and *mX* are created and saved in the GPU memory all the related calculations are automatically executed on the GPU, e.g., *inv* is executed directly on the GPU. This is completely transparent to the user.

If further calculations are needed on the CPU then the command *gather* transfers the data from GPU to the CPU, see line 5 of Listing 2. There exist already a lot of supported functions and this number continuously increases with new releases.[4]

```
1  iRows = 1000; iColumns = 5; % number of rows and columns
2  mX = randn(iRows, iColumns); % generate random numbers
3  vY = randn(iRows, 1);
4  vBeta = inv(mX ' * mX) * mX' * vY;
```

Listing 1 MATLAB CPU code that generate random numbers and estimate a linear regression model.

```
1  iRows = 1000; iColumns = 5; % number of rows and columns
2  mX = gpuArray.randn(iRows, iColumns); % generate random
       numbers
3  vY = gpuArray.randn(iRows, 1);
4  vBeta = inv(mX ' * mX) * mX' * vY;
5  vBeta = gather (vBeta); % transfer data to CPU
```

Listing 2 MATLAB GPU code that generate random numbers and estimate a linear regression model.

As further examples in Listings 3 and 4 we show the GPU implementation of the accept/reject and the importance sampling algorithms presented in Section 5.

```
1 sampsize = 1000000;      % sample size to use for examples
```

[4] See for the complete list of functions http://www.mathworks.com/help/distcomp/using-gpuarray.html

```
sig = 1;                  % standard deviation of the instrumental density
samp = gpuArray.randn(sampsize, 1) .* sig; % step 1 in the A/R algorithm
ys = exp((-samp.^2)/2) .* (sin(6 * samp).^2 + 3 *((cos(samp).^2).*(sin(4*samp).^2)) + 1);
wts = (1/sqrt(2*pi)) .* exp(-samp.^2/2);
samp2 = gpuArray.rand(sampsize, 1);
dens = samp(samp2<=(ys)./wts);   % step 2 in the A/R algorithm
target = gather(dens);           % step 3 in the A/R algorithm
```

Listing 3 Accept/reject MATLAB GPU code.

```
nIS = 10000; nu = gpuArray(12); nustar = gpuArray(7);% number of simulations; degree of freedom of the target density; degree of freedom of the proposal
muIS = gpuArray.nan(nIS, 2);
wIS = gpuArray.nan(nIS, 2);
x1 = rant_GPU(nIS, nustar);                        % Student t proposals
x2 =  tan((gpuArray.rand(nIS, 1) - 0.5) * pi); % Cauchy proposals
wIS(:, 1) = w1_GPU(x1, nu, nustar);                % Importance weights
wIS(:, 2) = w3_GPU(x2, nu);                        % Importance weights
muIS(:, 1) = sqrt(abs(x1./(1-x1)));
muIS(:, 2) = sqrt(abs(x2./(1-x2)));
muIScum(:,1)=cumsum(muIS(:,1).*wIS(:,1))./(1:nIS)';
muIScum(:,2)=cumsum(muIS(:,2).*wIS(:,2))./(1:nIS)';
%
% Additional functions
function w = w1_GPU(x,nu,nustar)     % Student's t weights
   w = tpdf_GPU(x, nu)./tpdf_GPU(x, nustar);
end
function w=w3_GPU(x,nu)              % Cauchy weights
    w = tpdf_GPU(x, nu)./ pdfcauchy_GPU(x, 0, 1);
end
function f = tpdf_GPU(x,v)           % Student's t GPU pdf
k = find(v>0 & v<Inf);
    if any(k)
        term = exp(gammaln((v(k) + 1) / 2) - gammaln(v(k)/2));
        f(k) = term ./ (sqrt(v(k)*pi) .* (1 + (x(k) .^ 2) ./ v(k)) .^ ((v(k) + 1)/2));
    end
end
function f = pdfcauchy_GPU(x, a, b)  % Cauchy GPU pdf
         f = 1./(pi .* b .* (1 + ((x - a)./b).^2));
end
```

Listing 4 Importance sampling GPU code.

References

Aastveit, K., J. Mitchell, F. Ravazzolo, and H. van Dijk (2019). The evolution of forecast density combinations in economics. In forthcoming (Ed.), *Oxford Research Encyclopedia of Economics and Finance*. North-Holland.

Aastveit, K., F. Ravazzolo, and H. van Dijk (2018). Combined density nowcasting in an uncertain economic environment. *Journal of Business and Economic Statistics 36*(1), 131–145.

Aastveit, K. A., C. Foroni, and F. Ravazzolo (2016). Density forecasts with MIDAS models. *Journal of Applied Econometrics 32*(4), 783–801.

Aastveit, K. A., K. Gerdrup, A. S. Jore, and L. A. Thorsrud (2014). Nowcasting GDP in real time: A density combination approach. *Journal of Business and Economic Statistics 32*(1), 48–68.

Amisano, G. and J. Geweke (2010, April). Comparing and evaluating bayesian predictive distributions of asset returns. *International Journal of Forecasting 26*(2), 216–230.

Andrews, D. W. K. (2002). Higher-Order Improvements of a Computationally Attractive “k”-Step Bootstrap for Extremum Estimators. *Econometrica 70*(1), 119–162.

Anscombe, F. (1968). Topics in the investigation of linear relations fitted by the method of least squares. *Journal of the Royal Statistical Society B 29*, 1–52.

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

Barnard, G. A. (1963). New methods of quality control. *Journal of the Royal Statistical Society, Series A 126*, 255–259.

Bassetti, F., R. Casarin, and F. Ravazzolo (2018). Bayesian nonparametric calibration and combination of predictive distributions. *Journal of the American Statistical Association 113*(522), 675–685.

Bates, J. and C. Granger (1969). The combination of forecasts. *Operations Research Quarterly 20*(4), 451–468.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business and Economic Statistics 19*(4), 465–474.

Berkowitz, J. and L. Kilian (2000). Recent developments in bootstrapping time series. *Econometric Reviews 19*(1), 1–48.

Billio, M., R. Casarin, F. Ravazzolo, and H. K. van Dijk (2013). Time-varying combinations of predictive densities using nonlinear filtering. *Journal of Econometrics 177*, 213–232.

Bose, A. (1988). Edgeworth corrections by bootstrap in autoregressions. *Annals of Statistics 16*(4), 1709–1722.

Casarin, R., R. V. Craiu, and F. Leisen (2016). Embarrassingly parallel sequential Markov-chain Monte Carlo for large sets of time series. *Statistics and Its Interface 9*(4), 497–508.

Casarin, R., S. Grassi, F. Ravazzolo, and H. K. van Dijk (2015). Parallel sequential monte carlo for efficient density combination: The deco matlab toolbox. *Journal of Statistical Software 68(3).*

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

Clements, M. P. and A. B. Galvao (2014). Measuring macroeconomic uncertainty: US inflation and output growth. ICMA Centre Discussion Papers in Finance 2014/04, Henley Business School, Reading University.

Clements, M. P. and N. Taylor (2001). Bootstrapping prediction intervals for autoregressive models. *International Journal of Forecasting 17*(2), 247–267.

Creel, M. (2005). User-Friendly Parallel Computations with Econometric Examples. *Computational Economics 26*, 107–128.

Davidson, R. and E. Flachaire (2008). The wild bootstrap, tamed at last. *Journal of Econometrics 146*(1), 162–169.

Davidson, R. and J. G. MacKinnon (2006). Bootstrap methods in econometrics. In *Palgrave Handbooks of Econometrics: Volume 1 Econometric Theory*, pp. 812–838. Basingstoke: Palgrave Macmillan.

Davison, A. and D. Hinkley (1997). *Bootstrap Methods and their Applications*. Cambridge University Press.

Dawid, A. P. (1982). Intersubjective statistical models. *Exchangeability in Probability and Statistics*, 217–232.

DeGroot, M. H., A. P. Dawid, and J. Mortera (1995). Coherent combination of experts' opinions. *Test 4*, 263–313.

DeGroot, M. H. and J. Mortera (1991). Optimal linear opinion pools. *Management Science 37*(5), 546–558.

Del Negro, M., B. R. Hasegawa, and F. Schorfheide (2016). Dynamic prediction pools: An investigation of financial frictions and forecasting performance. *Journal of Econometrics 192*(2), 391–405.

Diebold, F. and R. Mariano (1995). Comparing Predictive Accuracy. *Journal of Business and Economic Statistics 13*, 253–263.

Diebold, F. X., T. A. Gunther, and A. S. Tay (1998, November). Evaluating density forecasts with applications to financial risk management. *International Economic Review 39*(4), 863–83.

Djogbenou, A., S. Goncalves, and B. Perron (2015). Bootstrap inference in regressions with estimated factors and serial correlation. *Journal of Time Series Analysis 36*(3), 481–502.

Djogbenou, A., S. Goncalves, and B. Perron (2017). Bootstrap prediction intervals for factor models. *Journal of Business and Economic Statistics 35*(1), 53–69.

Durham, G. and J. Geweke (2014). Adaptive sequential posterior simulators for massively parallel computing environments. In I. Jeliazkov and D. J. Poirier (Eds.), *Bayesian Model Comparison (Advances in Econometrics*, Chapter 34. Emerald Group Publishing Limited.

Einav, L. and J. Levin (2014). Economics in the age of big data. *Science 346*(6210), 715–718.

Geweke, J. (1989). Bayesian Inference in Econometric Models using Monte Carlo Integration. *Econometrica 57*, 1317–1340.

Geweke, J. and G. Amisano (2011). Optimal prediction pools. *Journal of Econometrics 164*(1), 130 – 141.

Geweke, J. and G. Durham (2012). Massively Parallel Sequential Monte Carlo for Bayesian Inference. Working papers, University of Technology Sydney.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association 106*, 746–762.

Gneiting, T. and M. Katzfuss (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application 1*, 125–151.

Gneiting, T. and A. E. Raftery (2007, March). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*, 359–378.

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

Gneiting, T. and R. Ranjan (2013b). Combining predictive distributions. *Electronic Journal of Statistics 7*, 1747–1782.

Goncalves, S. and L. Kilian (2004). Bootstrapping autoregressions with conditional heteroskedasticity of unknown form. *Journal of Econometrics 123*(1), 89–120.

Goncalves, S. and B. Perron (2014). Bootstrapping factor-augmented regression models. *Journal of Econometrics 182*(1), 156–173.

Granger, C. W. J. (1998). Extracting information from mega-panels and high-frequency data. *Statistica Neerlandica 52*, 258–272.

Granger, C. W. J. and M. H. Pesaran (2000). Economic and statistical measures of forecast accuracy. *Journal of Forecasting 19*, 537–560.

Granger, C. W. J. and R. Ramanathan (1984). Improved Methods of Combining Forecasts. *Journal of Forecasting 3*, 197–204.

Groen, J. J. J., R. Paap, and F. Ravazzolo (2013). Real-Time Inflation Forecasting in a Changing World. *Journal of Business & Economic Stastistics 31*, 29–44.

Guidolin, M. and A. Timmermann (2009). Forecasts of US Short-term Interest Rates: A Flexible Forecast Combination Approach. *Journal of Econometrics 150*, 297–311.

Hall, P. (1985). Resampling a coverage process. *Stochastic Processes and their Applications 20*(2), 231–246.

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

Hansen, B. (2006). Interval forecasts and parameter uncertainty. *Journal of Econometrics 135*, 377–398.

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

Inoue, A. and L. Kilian (2002). Bootstrapping autoregressive processes with possible unit roots inoue. *Econometrica*, 377–391.

Kapetanios, G., J. Mitchell, S. Price, and N. Fawcett (2015). Generalised density forecast combinations. *Journal of Econometrics 188*, 150–165.

Kascha, C. and F. Ravazzolo (2010). Combining inflation density forecasts. *Journal of Forecasting 29*(1-2), 231–250.

Kloek, T. and H. van Dijk (1978). Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo. *Econometrica 46*, 1–19.

Knuppel, M. (2015). Evaluating the Calibration of Multi-Step-Ahead Density Forecasts Using Raw Moments. *Journal of Business & Economic Statistics 33*(2), 270–281.

Koop, G. (2003). *Bayesian Econometrics*. John Wiley and Sons.

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

Kunsch, H. R. (1989). The Jackknife and the Bootstrap for General Stationary Observations. *The Annals of Statistics 17*(3), 1217–1241.

LeSage, J. P. (1998, December). Econometrics: Matlab Toolbox of Econometrics Functions. Statistical Software Components, Boston College Department of Economics.

Liu, R. (1988). Bootstrap procedures under some non-i.i.d. models. *Annals of Statistics 16*, 1696–1708.

Mazzi, G., J. Mitchell, and G. Montana (2014). Density nowcasts and model combination: nowcasting euro-area gdp growth over the 2008-9 recession. *Oxford Bulletin of Economics and Statistics 76(2)*, 233–256.

McAlinn, K. and M. West (2018). Dynamic bayesian predictive synthesis in time series forecasting. *Journal of Econometrics forthcoming*.

Mitchell, J. and S. G. Hall (2005, December). Evaluating, comparing and combining density forecasts using the klic with an application to the bank of england and niesr 'fan' charts of inflation. *Oxford Bulletin of Economics and Statistics 67*(s1), 995–1033.

Mitchell, J. and K. Wallis (2011). Evaluating density forecasts: Forecast combinations, model mixtures, calibration and sharpness. *Journal of Applied Econometrics 26*(6), 1023–1040.

Morozov, S. and S. Mathur (2011). Massively Parallel Computation Using Graphics Processors with Application to Optimal Experimentation in Dynamic Control. *Computational Economics*, 1–32.

Pascual, L., J. Romo, and E. Ruiz (2001). Effects of parameter estimation on prediction densities: a bootstrap approach. *International Journal of Forecasting 17*(1), 83–103.

Pettenuzzo, D. and F. Ravazzolo (2016). Optimal portfolio choice under decision-based model combinations. *Journal of Applied Econometrics 31*(7), 1312–1332.

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

Raftery, A. E., T. Gneiting, F. Balabdaoui, and M. Polakowski (2005). Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review 133*, 1155–1174.

Raftery, A. E., D. Madigan, and J. A. Hoeting (1997, March). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association 92*(437), 179–91.

Ranjan, R. and T. Gneiting (2010). Combining probability forecasts. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(1), 71–91.

Ravazzolo, F. and S. V. Vahey (2014). Forecast densities for economic aggregates from disaggregate ensembles. *Studies in Nonlinear Dynamics and Econometrics 18*, 367–381.

Robert, C. P. and G. Casella (2004). *Monte Carlo Statistical Methods*. Springer, Berlin, Second Edition.

Roberts, H. V. (1965). Probabilistic prediction. *Journal of American Statistical Association 60*, 50–62.

Rossi, B. and T. Sekhposyan (2013). Conditional predictive density evaluation in the presence of instabilities. *Journal of Econometrics 177*(2), 199–212.

Rossi, B. and T. Sekhposyan (2014). Evaluating predictive densities of us output growth and inflation in a large macroeconomic data set. *International Journal of Forecasting 30*(3), 662–682.

Rossi, B. and T. Sekhposyan (2016). Alternative tests for correct specification of conditional predictive densities. Working Paper 758, Barcelona GSE.

Sloughter, J., T. Gneiting, and A. E. Raftery (2010). Probabilistic Wind Speed Forecasting Using Ensembles and Bayesian Model Averaging. *Journal of the American Statistical Association 105*, 25–35.

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

Stock, J. H. and W. M. Watson (2002). Forecasting using principal components from a large number of predictors. *Journal of American Statistical Association 97*, 1167–1179.

Stock, J. H. and W. M. Watson (2005). Implications of dynamic factor models for VAR analysis. Technical report, NBER Working Paper No. 11467.

Stock, J. H. and W. M. Watson (2014). Estimating turning points using large data sets. *Journal of Econometris 178*, 368–381.

Tay, A. and K. F. Wallis (2000). Density Forecasting: A Survey. *Journal of Forecasting 19*, 235–254.

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

Timmermann, A. (2006). Forecast combinations. In G. Elliot, C. W. J. Granger, and A. Timmermann (Eds.), *Handbook of Forecasting*, Chapter 4. Elsevier.

Varian, H. (2014). Machine learning: New tricks for econometrics. *Journal of Economics Perspectives 28*, 3–28.

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

Vergé, C., C. Dubarry, P. Del Moral, and E. Moulines (2015, Mar). On parallel implementation of sequential monte carlo methods: the island particle model. *Statistics and Computing 25*(2), 243–260.

Waggoner, D. F. and T. Zha (2012). Confronting model misspecification in macroeconomics. *Journal of Econometrics 171*, 167–184.

Wallis, K. F. (2003). Chi-squared tests of interval and density forecasts, and the bank of england's fan charts. *International Journal of Forecasting 19*(3), 165–175.

Wallis, K. F. (2005). Combining density and interval forecasts: A modest proposal. *Oxford Bulletin of Economics and Statistics 67*(s1), 983–994.

Wallis, K. F. (2011). Combining forecasts - forty years later. *Applied Financial Economics 21*(1-2), 33–41.

West, K. (1996). Asymptotic inference about predictive ability. *Econometrica 64*, 1067–1084.

Wu, C. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *Annals of Statistics 14*, 1261–1295.

Yeh, A. B. (1998). A bootstrap procedure in linear regression with nonstationary errors. *The Canadian Journal of Statistical Association 26*(1), 149–160.

Zarnowitz, V. (1969). Topics in the investigation of linear relations fitted by the method of least squares. *American Statistician 23*, 12–16.