



HAL
open science

CRPS-based online learning for nonlinear probabilistic forecast combination

Dennis van der Meer, Pierre Pinson, Simon Camal, Georges Kariniotakis

► **To cite this version:**

Dennis van der Meer, Pierre Pinson, Simon Camal, Georges Kariniotakis. CRPS-based online learning for nonlinear probabilistic forecast combination. *International Journal of Forecasting*, inPress, 10.1016/j.ijforecast.2023.12.005 . hal-04408320

HAL Id: hal-04408320

<https://hal.science/hal-04408320>

Submitted on 20 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Highlights

CRPS-based online learning for nonlinear probabilistic forecast combination

Dennis van der Meer, Pierre Pinson, Simon Camal, Georges Kariniotakis

- Linear combination of calibrated probabilistic forecasts leads to overdispersion.
- We develop a flexible nonlinear forecast combination method for the online setting.
- The model converges to the best fixed strategy in two simulation studies.
- The model outperforms the best fixed strategy in a nonstationary real-world study.

CRPS-based online learning for nonlinear probabilistic forecast combination

Dennis van der Meer^{a,*}, Pierre Pinson^{b,c}, Simon Camal^a and Georges Kariniotakis^a

^aMines Paris, PSL University, Centre for processes, renewable energy and energy systems (PERSEE), 06904, Sophia Antipolis, France

^bImperial College London, Dyson School of Design Engineering, South Kensington Campus, London, SW7 2AZ, United Kingdom

^cTechnical University of Denmark, Department of Technology, Management and Economics, Akademivej 325, Kgs. Lyngby, 2800, Denmark

ARTICLE INFO

Keywords:

Beta-transform

Linear opinion pool

Continuous ranked probability score

Post-processing

Online convex optimization

ABSTRACT

Forecast combination improves upon the component forecasts. Most often, combination approaches are restricted to the linear setting only. However, theory shows that if the component forecasts are neutrally dispersed—a requirement for probabilistic calibration—linear forecast combination will only increase dispersion and thus lead to miscalibration. Furthermore, the accuracy of the component forecasts may vary over time and the combination weights should vary accordingly, necessitating updates as time progresses. In this paper, we develop an online version of the beta-transformed linear pool, which theoretically can transform the probabilistic forecasts such that they are neutrally dispersed. We show that, in case of stationary synthetic time series, the performance of the developed method converges to that of the optimal combination in hindsight. Moreover, in case of nonstationary real-world time series from a wind farm in mid-west France, the developed model outperforms the optimal combination in hindsight.


1. Introduction

The combination of probability distributions issued by experts has a long history that can be traced back to at least Stone (1961). The linear opinion pool, as labelled by Stone (1961), is the convex combination of component probability distributions. In probabilistic forecasting, forecasters aim to maximize the sharpness of the forecasts, subject to calibration (Gneiting, Balabdaoui and Raftery, 2007). Calibration refers to the agreement between the forecasts and observed probabilities; for instance, when a forecaster predicts daily overcast conditions with 80% probability, cloudiness should actually occur on 80 of the 100 days with such conditions. Ensemble forecasts from numerical weather prediction (NWP) models tend to exhibit underdispersion, which implies that the forecasts are overconfident (Wilks, 2018). Conversely, probabilistic forecasts can be overdispersed, which means that the forecaster is underconfident and issues forecasts with too much variance. These types of miscalibration can negatively affect decision-making based on such forecasts and care needs to be taken to ensure proper calibration. On this point, Hora (2004) notes that “there are theoretical reasons for questioning the use of linear combinations of experts’ probabilities. This concern stems from understanding that well-calibrated experts cannot be combined without introducing miscalibration.” Nevertheless, linear forecast combination is commonly used, especially in combination with batch learning. Following the above, it is important to note that the context in this manuscript is different from that of deterministic, or point, forecast combination. In the context of the latter, the observation that the simple average of experts’ forecasts tends to be more accurate than the optimally weighted combination is referred to as the “forecast combination puzzle” and is attributed to the combination variance that is affected by weight estimation (Claeskens, Magnus, Vasnev and Wang, 2016).

1.1. Batch learning

Batch learning requires a separate data set that accurately represents the test set to optimize the combination weights. Consequently, the optimally weighted forecast combination does not necessarily outperform the best component forecast if the separate data sets are not comparable (e.g., Hall and Mitchell, 2007). Nevertheless, there is value to optimizing the weights as shown by Martin, Loaiza-Maya, Maneesoonthorn, Frazier and Ramírez-Hassan (2022) who consistently outperform the naive forecast combination with equal weights. A notable example frequently applied

*Corresponding author

 dennis.van_der_meer[at]minesparis.psl.eu (D.v.d. Meer)

ORCID(s): 0000-0002-9473-4536 (D.v.d. Meer)

to NWP forecasts is Bayesian Model Averaging (BMA), where ensemble members are dressed with a probability density function (PDF) and the weights optimized using the logarithmic score (Raftery, Gneiting, Balabdaoui and Polakowski, 2005). In their application, BMA is an efficient method to calibrate the ensemble because the ensemble under investigation is underdispersed. To reduce overdispersion, Bracale, Carpinelli and De Falco (2017) minimize a weighted sum of the continuous ranked probability score (CRPS) and deviation from calibration. Similarly, Jose, Grushka-Cockayne and Lichtendahl (2014) propose the exterior-trimmed opinion pool heuristic method that effectively removes expert forecasts with low and high means or cumulative distribution function (CDF) values. Additionally, they introduce the interior-trimmed opinion pool that removes expert forecasts with moderate means or CDF values to increase the dispersion in case the experts are overconfident.

Averaging probabilities, which is also referred to as vertical averaging, can be shown to be at least as accurate in terms of CRPS as the average CRPS of the experts (Lichtendahl, Grushka-Cockayne and Winkler, 2013). Besides averaging probabilities, it is also possible to average quantiles, which can be referred to as horizontal averaging. In their comparative study, Lichtendahl et al. (2013) show that the average quantile forecast is always sharper than the average probability forecast and that the former is therefore better suited in case the component forecasts are well calibrated. In the area of load forecasting, Wang, Zhang, Tan, Hong, Kirschen and Kang (2019) linearly combine quantiles and show that their approach does not always outperform the component models although they do not evaluate forecast calibration. In a similar fashion, Bracale, Carpinelli and De Falco (2019) minimize the pinball loss to optimally combine quantile forecasts of photovoltaic (PV) power and improve accuracy. However, similar to Wang et al. (2019), the authors do not evaluate the calibration of the component nor the combined forecasts. Taylor and Taylor (2023) forecast the cumulative COVID-19 mortality and face interesting challenges such as a lack of historical forecasts, and therefore use trimming techniques as well as the simple average, the median forecast and weights based on the inverse quantile score to find that the latter performs at least as well as the simple average.

Besides trimming and averaging quantiles, one can also apply nonlinear transformations to the linear opinion pool to improve probabilistic calibration. Gneiting and Ranjan (2013) describe two such methods, namely the spread-adjusted linear pool (SLP) and beta-transformed linear pool (BLP). SLP adjusts the spread of the component forecasts and can consequently mitigate—to a certain extent—overdispersion caused by linearly combining calibrated forecasts. Möller and Groß (2020) apply SLP to post-processed temperature forecasts issued by the European Center for Medium-range Weather Forecasts (ECMWF) ensemble prediction system and show that it effectively lowers CRPS compared to the component forecasts. However, a limitation of SLP is that the method fails to be flexibly dispersive, which is to say that it is unable to sufficiently adjust the spread to produce neutrally dispersed forecasts, especially when the component forecasts are neutrally dispersed or underdispersed (Gneiting and Ranjan, 2013). In contrast, BLP is exchangeably flexibly dispersive and as such is able to transform the predictive distributions such that the second moment of the resulting probability integral transform (PIT) can attain any value in the open interval $(0, 1/4)$, with $1/12$ indicating neutral dispersion (Gneiting and Ranjan, 2013). Van der Meer, Camal and Kariniotakis (2022) apply both SLP and BLP to combine PV power forecasts and show that SLP outperforms BLP, which is caused by a lack of representativeness of the training data that affects parameter learning more in case of BLP than in case of SLP. Finally, Bassetti, Casarin and Ravazzolo (2018) develop a Bayesian nonparametric approach that extends the parametric class of calibration functions, i.e., BLP, using a possibly unknown number of beta mixtures, which can be interpreted as a mixture of local combination models.

1.2. Online learning

Nonstationary data, extensive training times for complex machine learning models, and data storage present challenges in batch learning. To illustrate this further, consider batch gradient descent. In this method, the objective is to update the model parameter θ by calculating the loss gradient $\nabla L(f(x_i, \theta), y_i)$ across a training set indexed by $i \in 1, \dots, N$, which consists of input-output pairs (x_i, y_i) . The parameter update, after processing a batch containing N data points, is computed as follows: $\theta_{\text{new}} = \theta_{\text{old}} - \eta \cdot \frac{1}{N} \sum_{i=1}^N \nabla L(f(x_i, \theta_{\text{old}}), y_i)$, with η representing the learning rate. The aforementioned expression highlights that batch learning necessitates the storage of all N data points and the consecutive computation of the gradient N times. Unlike batch learning, online learning is computationally less expensive and does not require to store historical data. Moreover, it integrates real-time data and therefore adapts to trends and seasonalities. Returning to the example of gradient descent, the online version instead updates θ as soon as a new input-output pair (x_i, y_i) becomes available as $\theta_{\text{new}} = \theta_{\text{old}} - \eta \cdot \nabla L(f(x_i, \theta_{\text{old}}), y_i)$. In the context of forecast combination, such real-time adaptation would be relevant when one of the experts is better at forecasting in a declining market rather than a rising market (Winkler, Grushka-Cockayne, Lichtendahl and Jose, 2019).

Recent years have seen a stark increase in the number of publications on online forecast combination. For instance, Thorey, Mallet and Baudin (2017) consider online ridge regression and exponentiated gradient to update weights based on CRPS minimization so as to linearly combine the ensemble members into a CDF. In a subsequent study, Thorey, Chaussin and Mallet (2018) extend their research by applying the ML-Poly algorithm introduced by Gaillard, Stoltz and van Erven (2014) to linearly combine ensemble members from ECMWF and Météo France. This extension demonstrated enhanced performance compared to using raw ensembles, although it should be noted that their forecasts still exhibited underdispersion. In the aforementioned studies, the authors show that the regret of their algorithm is logarithmic as a function of time. It is important to note that in the field of online learning, here used interchangeably with Online Convex Optimization (OCO), and of which forecast combination is a subset, the primary objective is to minimize regret rather than loss. Regret is defined as the difference in performance between the online player and the optimal fixed strategy in hindsight (Hazan, 2021). Instead of combining ensemble members, V'yugin and Trunov (2019) combine predictive CDFs using Vovk's Aggregating Algorithm and show that their method offers a time-independent upper bound on regret. Zamo, Bel and Mestre (2021) modify the objective function to comprise the CRPS and the Jolliffe-Primo test for rank histogram flatness to improve calibration of their linearly combined forecasts. The aforementioned studies concern linear forecast combination, which, as previously discussed, leads to overdispersion.

Regarding quantile forecasting, Berrisch and Ziel (2021) observe potential differences in accuracy across various segments of the experts' predictive distributions. To address this, they propose a linear pointwise combination algorithm that aggregates quantiles based on CRPS minimization. Their research reveals that this approach yields a more uniformly distributed loss throughout the predictive distribution. In a related study, Krannichfeldt, Wang, Zufferey and Hug (2022) adapt the pinball loss to remain "passive" when the loss is below a threshold but "aggressively" adjusts the weights when a new sample causes the loss to exceed the threshold, resulting in improved CRPS and pinball loss although their method appears to be outperformed by benchmark models in terms of calibration.

1.3. Contributions

In some applicative fields like wind power forecasting, the value of forecast combination was recognized already 20 years ago, with the first operational models based on spot forecast combination set-up by the Spanish Transmission System Operator (see, e.g., Sánchez, 2008). Today, this is considered as a mainstream approach in business practices in renewable energy forecasting. Though when it comes to probabilistic forecasting, several research challenges remain. As outlined in the previous sections, the literature on probabilistic forecast combination—both batch and online—is expanding rapidly, which has been predicted based on the popularity of forecasting competitions and developments in machine learning and expert forecasting (Winkler et al., 2019). In this work, we extend the beta-transformed linear pool proposed by Gneiting and Ranjan (2013) to the online setting to mitigate miscalibration caused by linear forecast combination. The beta-transformed linear pool is exchangeably flexibly dispersive, meaning that there exists a set of parameters that ensures that the combined forecasts are probabilistically calibrated while the component forecasts are exchangeable (Gneiting and Ranjan, 2013). The method that we develop is able to adapt in nonstationary contexts and is integrated into the Online Newton Step (ONS) algorithm that moves in the direction of an approximate Hessian and the gradient, the former of which is additionally used to project the weights back onto the simplex. The CRPS is employed to guide the learning process as it is exponentially-concave, an attribute of the CRPS that permits logarithmic regret when used as a cost function in the ONS algorithm. To summarize, we contribute to the state of the art of online probabilistic forecast combination in the following ways:

- We develop a nonlinear and online method to combine probabilistic forecasts that is exchangeably flexibly dispersive.
- The proposed method relies on the CRPS, an exponentially-concave function, in conjunction with the Online Newton Step algorithm. This combination permits logarithmic regret and accommodates the most comprehensive scenario, wherein experts provide full predictive distributions.
- We demonstrate the effectiveness of our method through two simulation studies from the literature and a real-world wind power forecasting study, and make the code publicly available to facilitate its uptake.

The remainder of this work is organized as follows. Section 2 describes the probabilistic forecast combination framework, i.e., the linear and beta-transformed linear pool, as well as the CRPS that is used to guide the learning process. Next, we introduce online convex optimization, develop the necessary mathematics and introduce the ONS

algorithm in Section 3. We present the results of two simulation studies and a real-world wind power case study in Section 4 and conclude this work in Section 5.

2. Probabilistic forecast combination framework

In this section, we provide a comprehensive explanation of forecast combination. Initially, we delve into the concept of linear combination, followed by an exploration of how the Beta distribution is employed to transform this linear combination. Throughout the remainder of this work, we consider a total of m experts indexed by j and denote a predictive distribution by \hat{F} . Furthermore, we use lower case normal font for realizations of scalar variables and upper case for scalar variables, lower case bold font for vectors and upper case bold font for matrices.

2.1. Classical linear combination

The linear opinion pool is defined as (e.g., Gneiting and Ranjan, 2013):

$$\hat{F}(y) = \sum_{j=1}^m w_j \hat{F}_j(y). \quad (1)$$

We adopt the abbreviations of Gneiting and Ranjan (2013) in this work. Consequently, in case of equal weights, i.e., $w_j = 1/m$, we refer to (1) as the ordinary linear pool (OLP), which was labelled the linear opinion pool by Stone (1961). In contrast, when the weights are optimized with respect to a score, we refer to (1) as the traditional linear pool (TLP). The main motivation to combine forecasts is to harness the wisdom of the crowd, which is an effective method in point forecasting due to the expertise and diversity of the crowd (Soll, Mannes and Larrick, 2012). Instead, linearly combining diverse probabilistic forecasts further increases the dispersion, which may lead to calibration issues (Hora, 2004).

2.1.1. Beta-transformed linear pool

To mitigate miscalibration caused by linear combination, Gneiting and Ranjan (2013) propose a nonlinear transformation of the linear pool (1) by means of the Beta CDF. The beta-transformed linear pool (BLP) forecast encapsulates (1) and is defined as (Gneiting and Ranjan, 2013):

$$\hat{F}_{a,b}(y) = I_{a,b} \left(\sum_{j=1}^m w_j \hat{F}_j(y) \right), \quad (2)$$

where $I_{a,b}$ is the regularized incomplete beta function with shape parameters a and b . Recall that BLP is able to transform the predictive distributions such that the second moment of the resulting PIT can attain any value in the open interval $(0, 1/4)$ with fixed weights and $a > 0$ and $b > 0$ (Gneiting and Ranjan, 2013). Note that when $a = b = 1$, (2) equals (1). Throughout the remainder of this work, we will abbreviate the linear opinion pool as $z = \sum_{j=1}^m w_j \hat{F}_j(y)$. The regularized incomplete beta function, which is also known as the Beta CDF, is defined as:

$$I_{a,b}(z) = \frac{B_{a,b}(z)}{B_{a,b}}, \quad (3)$$

where

$$\begin{aligned} B_{a,b}(z) &= \int_0^z u^{a-1} (1-u)^{b-1} du \\ &= \Gamma(a) z^a {}_2\tilde{F}_1(a, 1-b; a+1; z), \end{aligned} \quad (4)$$

$$\begin{aligned} B_{a,b} &= \int_0^1 u^{a-1} (1-u)^{b-1} du \\ &= \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}. \end{aligned} \quad (5)$$

In the above equations, $B_{a,b}(z)$ is the incomplete Beta function, $B_{a,b}$ is the complete Beta function, Γ is the gamma function and ${}_2\tilde{F}_1$ is a regularized hypergeometric function. These identities are useful to derive the gradient in Section 3.1.

2.2. CRPS-based learning

Strictly proper scoring rules minimize in expectation under the true model. These rules are therefore recommended for forecast evaluation because they encourage truth telling from a forecaster (e.g., Gneiting and Raftery, 2007). Conversely, a forecaster may minimize such a rule to identify the true model in the learning stage. The CRPS is a strictly proper scoring rule that evaluates the entire predictive distribution and therefore permits the most general setting where experts issue complete predictive distributions. Moreover, the CRPS is exponential-concave in case of bounded support, which implies that the score is strongly convex in the direction of the gradient but not necessarily elsewhere (Korotin, V'yugin and Burnaev, 2021; Hazan, 2021). Exp-concavity is an important attribute of the CRPS because—in combination with particular online learning methods—it allows for accelerated learning, as Section 3.2 will clarify. The CRPS is defined as follows:

$$\text{CRPS}(\hat{F}_t, y_t) = \int_0^1 (\hat{F}_t(x) - \mathbb{1}\{x \geq y_t\})^2 dx, \quad (6)$$

where $\mathbb{1}$ denotes the indicator function that is 1 when the condition inside the curly brackets is true. The limits of the integral are taken to be 0 and 1, respectively, since the data are normalized. Note that (6) is an instantaneous value and that these are averaged over a test set to rank competing forecasts. Similarly, one can minimize the expected value of (6) to learn the true parameters of a forecast model or, as in our case, learn the optimal combination parameters. Henceforth we will omit the time index to simplify the notation.

2.3. Distribution-oriented forecast verification

While the CRPS can be used to rank competing forecasts, it does not reveal specific types of miscalibration. The PIT is a common verification tool to visualize the calibration of probabilistic forecasts and can be computed as $Z = \hat{F}(Y)$, where \hat{F} and Y are series of predictive distributions and observations, respectively. \hat{F} is said to be probabilistically calibrated if $Y \sim \hat{F}$, which implies that Z is a standard uniform distribution (Rosenblatt, 1952). Consequently, the variance of Z is constrained to the closed interval $[0, 1/4]$ and the distribution is flat when $\text{var}(Z) = 1/12$.

Nevertheless, a PIT histogram can deviate from flatness due to randomness induced by a test set of limited length. To account for randomness, Bröcker and Smith (2007) propose consistency bands that represent the maximum estimation uncertainty that can be expected for a test set of specific length. In other words, as long as the deviation from flatness remains within the consistency bands, one cannot reject the hypothesis that the probabilistic forecasts are reliable. Herein, we use dotted lines to visualize the consistency bands.

3. Online convex optimization

Online convex optimization (OCO) can be seen as a game where a player repeatedly makes decisions over time. At time step t , the player chooses from the n -dimensional convex decision set $\mathbf{x}_t \subseteq \mathbb{R}^n$. After the player's decision, a convex cost function f_t is revealed and the player incurs a loss $f_t(\mathbf{x}_t)$. The performance of OCO algorithms is measured in terms of regret, which is defined as the difference in performance between the online player and the optimal fixed strategy in hindsight. In the context of online forecast combination, the forecast aggregator is the online player who adapts the aggregation function based on the most recent performance and who regrets—in hindsight—not choosing the fixed aggregation function that optimizes performance. The reason for using regret is that OCO permits an adversary who can choose different cost functions as the game progresses. In this framework, OCO is concerned with algorithms that realize sublinear regret with increasing test length, implying that, on average, the algorithms perform as well as the best strategy in hindsight. For an in-depth treatment of OCO and its various algorithms, the reader is referred to Orabona (2019) and Hazan (2021).

Although there are settings in OCO, such as Multi-Armed Bandit, where a player does not have access to the loss gradient, we focus on gradient-based online learning. One advantage of gradient-based online learning is that it generally permits tighter bounds on regret. The properties of the CRPS described in Section 2.2 make it an attractive scoring rule in an online learning context as well as for evaluating forecasts. What follows is the derivation of the CRPS gradient with respect to the shape parameters of the Beta CDF and the linear combination weights. Afterwards, we introduce the OCO algorithm that we employ.

3.1. Derivation of the CRPS gradient

Recall that we consider m experts indexed by j ; the vector of weights is subsequently defined as $\mathbf{w} = (w_1, \dots, w_m)^\top$ and the vector of parameters is defined as $\mathbf{x} = (a, b, \mathbf{w}^\top)^\top$. The gradient of the CRPS with respect to \mathbf{x} is then defined as follows:

$$\nabla \text{CRPS}(\hat{F}_{a,b}, y) = \left(\frac{\partial \text{CRPS}}{\partial a}, \frac{\partial \text{CRPS}}{\partial b}, \frac{\partial \text{CRPS}}{\partial w_1}, \dots, \frac{\partial \text{CRPS}}{\partial w_m} \right)^\top. \quad (7)$$

3.1.1. Partial derivative with respect to the weights

In deriving the partial derivatives, we first consider the linear combination of the component forecasts as it is encapsulated in the beta-transformed linear pool. This is relevant because the linear pool can act as a benchmark for the beta-transformed linear pool.

Linear The combination weights are contained within the integral, cf. (6). We therefore use the Leibniz integral rule for differentiation under the integral sign to derive the partial derivatives with respect to weights w_j since they are not integration variables. To improve the numerical properties of the learning process, Pinson and Madsen (2012) propose a logit transform \tilde{w}_j of w_j , such that:

$$\tilde{w}_j = \ln \left(\frac{w_j}{1 - w_j} \right), \quad (8)$$

which constrains w_j to the open interval $(0, 1)$. Using the chain rule, the partial derivative of the CRPS with respect to w_j becomes:

$$\frac{\partial \text{CRPS}}{\partial w_j} = \frac{\partial \text{CRPS}}{\partial \tilde{w}_j} \frac{\partial \tilde{w}_j}{\partial w_j},$$

which means that:

$$\frac{\partial \text{CRPS}}{\partial \tilde{w}_j} = \frac{\partial \text{CRPS}}{\partial w_j} \frac{\partial w_j}{\partial \tilde{w}_j}.$$

Notice that $\frac{\partial \tilde{w}_j}{\partial w_j} = \frac{1}{w_j - w_j^2}$. We have:

$$\begin{aligned} \frac{\partial \text{CRPS}}{\partial w_j} &= \int_0^1 2 (\hat{F}_{\text{TLP}}(x) - \mathbb{1}\{x \geq y\}) \frac{\partial \hat{F}_{\text{TLP}}(x)}{\partial w_j} dx \\ &= 2 \int_0^1 \hat{F}_j(x) (\hat{F}_{\text{TLP}}(x) - \mathbb{1}\{x \geq y\}) dx, \end{aligned} \quad (9)$$

and therefore:

$$\frac{\partial \text{CRPS}}{\partial \tilde{w}_j} = 2 (w_j - w_j^2) \int_0^1 \hat{F}_j(x) (\hat{F}_{\text{TLP}}(x) - \mathbb{1}\{x \geq y\}) dx. \quad (10)$$

Nonlinear The partial derivative in the nonlinear setting is similar to (9), except that $\hat{F}_{\text{TLP}}(x)$ is replaced with $\hat{F}_{a,b}(z)$. Notice that z appears in the upper limit of the integral in (4) and we therefore require Leibniz's integral rule. We defer the derivation to Appendix A.1 and instead present the final result below:

$$\frac{\partial \text{CRPS}}{\partial \tilde{w}_j} = \frac{2 (w_j - w_j^2)}{B_{a,b}} \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) z^{a-1} (1-z)^{b-1} \hat{F}_j(x) dx. \quad (11)$$

3.1.2. Partial derivative with respect to shape parameter a

Similarly, shape parameter a is not an integration variable and we therefore apply the Leibniz integral rule again. Important to note here is that both a and b are strictly positive. Similar to weight w_j , we apply a change of variable $\tilde{a} = \ln(a)$ to improve the stability of the algorithm (Pinson and Madsen, 2012). The partial derivative with respect to shape parameter a is presented below, whereas the derivation is deferred to Appendix A.2:

$$\begin{aligned} \frac{\partial \text{CRPS}}{\partial \tilde{a}} &= 2a \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \\ &\quad \times \left(\hat{F}_{a,b}(z) (\ln(z) - \psi(a) + \psi(a+b)) \right. \\ &\quad \left. - \frac{\Gamma(a)\Gamma(a+b)}{\Gamma(b)} z^a {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) \right) dx, \end{aligned} \quad (12)$$

where ψ is the digamma function.

3.1.3. Partial derivative with respect to shape parameter b

A similar approach to derive $\partial \text{CRPS} / \partial b$ can be used as was used in the previous section. Furthermore, an identical change of variable $\tilde{b} = \ln(b)$ is used to improve the stability of the algorithm. Equation (13) presents the result, while the derivation is deferred to Appendix A.3.

$$\begin{aligned} \frac{\partial \text{CRPS}}{\partial \tilde{b}} &= 2b \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \\ &\quad \times \left(\frac{\Gamma(a+b)\Gamma(b)}{\Gamma(a)} (1-z)^b {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) \right. \\ &\quad \left. + \hat{F}_{b,a}(1-z) (\psi(b) - \psi(a+b) - \ln(1-z)) \right) dx. \end{aligned} \quad (13)$$

3.2. Online Newton step

Several algorithms have been proposed in the field of online convex optimization that are analogues to well-known offline algorithms (see, e.g., Hazan, 2021, for an overview). Online Gradient Descent (OGD), as described in Section 1.2, is the analogue to Gradient Descent in which the algorithm moves in the direction of the gradient at every iteration and is therefore a first-order algorithm (Zinkevich, 2003). Despite its simplicity, OGD is linear in time and achieves sublinear regret in the worst case and logarithmic regret for strongly convex loss functions (Hazan, 2021).

One difficulty associated with OGD is the need for precise tuning of the step size or learning rate to achieve the desired regret. Additionally, it may be that the requirement of strong convexity is too stringent in practice. In such instances, it is possible to utilize exp-concave loss functions that are strongly convex in the direction of the gradient but not necessarily elsewhere (Hazan, 2021). The Online Newton Step (ONS) is an algorithm that guarantees logarithmic regret for exp-concave loss functions and therefore does not require an adaptive step size (Hazan, Agarwal and Kale, 2007). ONS is analogues to the Newton-Raphson method in that it moves in the direction of an approximated Hessian and the gradient, i.e., $\mathbf{A}_t^{-1} \nabla_t$, yet is based only on first-order information. The operation $\mathbf{A}_t^{-1} \nabla_t$ can potentially yield a set of weights that lie beyond the boundaries of the unit simplex. To avoid constraints, Pinson and Madsen (2012) parameterize the transition probabilities on the m -dimensional unit sphere. Instead, we perform a projection onto the unit simplex with a norm induced by \mathbf{A}_t rather than the Euclidean norm (Hazan et al., 2007). More details on this projection will be given below.

Recall that we consider m experts, resulting in a weight vector $\mathbf{w} \in \mathbb{R}^m$ and a parameter vector $\mathbf{x} \in \mathbb{R}^n$ that includes the shape parameters of the Beta distribution (3). Algorithm 1 summarizes the ONS algorithm in detail. Note that γ is a scaling factor of the starting point of the update step, i.e., \mathbf{A}_0^{-1} , whereas η represents. The step size remains constant for two reasons: (i) ONS guarantees logarithmic regret for exp-concave losses and (ii) maintaining a fixed step size allows the algorithm to adapt effectively in non-stationary environments, whereas reducing step sizes could hinder this adaptability. To avoid inverting a potentially large matrix, Hazan et al. (2007) recommend a recursion step for \mathbf{A}_t^{-1} using the Sherman-Morrison formula (Sherman and Morrison, 1950):

$$\mathbf{A}_t^{-1} = (\mathbf{A}_t + \nabla_t \nabla_t^\top)^{-1}$$

Algorithm 1: Online Newton step (Hazan, 2021; Wintenberger, 2021)

Data: convex set \mathcal{K} , T , $\mathbf{x}_1 \in \mathcal{K} \subseteq \mathbb{R}^n$, parameters $\gamma, \eta > 0$, $\mathbf{A}_0 = 1/\gamma^2 \mathbf{I}_n$, $\mathbf{A}_0^{-1} = \gamma^2 \mathbf{I}_n$

for $t \leftarrow 1$ **to** T **do**

Play \mathbf{x}_t and observe cost $f_t(\mathbf{x}_t)$;

$\mathbf{A}_t = \mathbf{A}_{t-1} + \nabla_t \nabla_t^\top$;

$\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1} \nabla_t \nabla_t^\top \mathbf{A}_{t-1}^{-1}}{1 + \nabla_t^\top \mathbf{A}_{t-1}^{-1} \nabla_t}$;

Newton step: $\mathbf{y}_{t+1} = \mathbf{x}_t - \eta \frac{1}{\gamma} \mathbf{A}_t^{-1} \nabla_t$;

$\mathbf{v}_{t+1} \in \mathbb{R}^m \subset \mathbf{y}_{t+1} \in \mathbb{R}^n$;

Projection (weights only) with weighted norm $\|\cdot\|_{\mathbf{D}_t}$: $\mathbf{w}_{t+1} = \frac{1}{2} \arg \min_{\mathbf{w} \in \mathcal{K}} \|\mathbf{w} - \mathbf{v}_{t+1}\|_{\mathbf{D}_t}^2$

end

$$= \mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1} \nabla_t \nabla_t^\top \mathbf{A}_{t-1}^{-1}}{1 + \nabla_t^\top \mathbf{A}_{t-1}^{-1} \nabla_t}. \quad (14)$$

In this work, the weights should sum to 1. Although not a strict requirement, we additionally ensure that the weights are positive. Whereas OGD uses a Euclidean projection step, ONS requires a generalized projection onto the unit simplex Λ that is in the vector norm induced by \mathbf{A}_t , which is a convex program (Hazan et al., 2007). Nonetheless, the presence of off-diagonal elements within \mathbf{A}_t introduces complexity to the optimization problem. Therefore, it is more advantageous to approximate it by employing $\mathbf{D}_t = \text{diag}(\mathbf{A}_t)$. Consequently, the generalized projection onto Λ breaks down into a series of independent scalar minimization problems for each component of \mathbf{w} . These can be resolved by solving a piecewise linear equation through sorting, as described by Held, Wolfe and Crowder (1974). Specifically, we want to solve the following convex optimization problem (Hazan et al., 2007)

$$\mathbf{w}_{t+1} = \frac{1}{2} \arg \min_{\mathbf{w} \in \mathcal{K}} \|\mathbf{w} - \mathbf{v}_{t+1}\|_{\mathbf{D}_t}^2, \quad (15)$$

where $\mathbf{v}_{t+1} \in \mathbb{R}^m$ comprises only the weights and $\|\mathbf{v}_{t+1} - \mathbf{w}\|_{\mathbf{D}_t}^2 := (\mathbf{w} - \mathbf{v}_{t+1})^\top \mathbf{D}_t (\mathbf{w} - \mathbf{v}_{t+1})$. The piecewise linear equation can be derived by solving a system of linear equations obtained from the Karush–Kuhn–Tucker (KKT) optimality conditions (see Appendix B), which results in the weighted soft-threshold (Wintenberger, 2021)

$$\mathbf{w}^* = \mathbf{D}_t^{-1} \text{SoftThreshold}(\mathbf{D}_t \mathbf{v}_{t+1}, \mathbf{v}^*). \quad (16)$$

Algorithm 2 summarizes the aforementioned steps. Note that in our application, the convex set $\mathcal{K} \subseteq \mathbb{R}^n$ is unbounded and open since the shape parameters a and b are strictly larger than 0. We therefore forego a regret analysis and instead empirically show the efficacy of the proposed method.

4. Results

We illustrate the efficacy of the proposed method through three case studies. The first two case studies are based on synthetic data sets adapted from relevant studies, a time-invariant (Section 4.2.1) and a time-varying process (Section 4.2.2). The synthetic case studies enable us to assess the performance of the proposed method within a controlled environment. To facilitate uptake of the proposed method, we make the code to generate the synthetic data and run the experiments available on GitHub¹. Finally, we apply the proposed method to a real-world wind power forecast case study in Section 4.3.

4.1. Benchmarks

In order to evaluate the effectiveness of the proposed online combination method, referred to as BLP, we incorporate several benchmark models. The most obvious benchmarks are the component models that BLP is designed to combine,

¹www.github.com/xyz

Algorithm 2: Simplex projection with weighted norm $\|\cdot\|_{\mathbf{D}_t}$ (Wintenberger, 2021)

Data: $\mathbf{w} \in \mathbb{R}^m$ and $\mathbf{D}_t = \text{diag}(\mathbf{A}_t)$

if $\mathbf{w} \in \Lambda$ **then**

Return \mathbf{w} ;

else

Sort $(d_t v_{t+1})_1 \geq \dots \geq (d_t v_{t+1})_m$;

Find $d_0 = \max \left\{ 1 \leq j \leq m; (d_t v_{t+1})_j - \frac{1}{\sum_{i=1}^{d_0} d_{t,i}^{-1}} \left(\sum_{i=1}^{d_0} v_{t+1,i} - 1 \right) \right\}$;

Define $\mathbf{v}^* = \frac{1}{\sum_{i=1}^{d_0} d_{t,i}^{-1}} \left(\sum_{i=1}^{d_0} v_{t+1,i} - 1 \right)$;

Return $\mathbf{w}^* = \mathbf{D}_t^{-1} \text{SoftThreshold}(\mathbf{D}_t \mathbf{v}_{t+1}, \mathbf{v}^*)$

end

and these will be elaborated upon in the respective sections. Furthermore, there are three combination models characterized by fixed weights, meaning that their weights remain constant throughout the entire duration. These combination models are OLP with equal weights, TLP* (cf. (1)), and BLP* (cf. (2)). Note that the asterisk indicates that the model parameters are optimized in hindsight, meaning that on average they are optimal for the testing data set. Finally, we include two benchmarks based on online learning: (i) TLP, and (ii) the CRPS learning approach by Berrisch and Ziel (2021), available on CRAN², with all options set to the default values and which we refer to as PFC. Note that the parameters required by the online learning methods are updated at every time step.

4.2. Synthetic data

For both synthetic case studies, we initialize the weights as $1/m$ and the shape parameters $a = b = 1$, which effectively is the ordinary linear pool (OLP). In total, we run 150 simulations for 11,500 time steps and report the median as well as the 90% confidence interval of the parameter estimates. Regarding the hyperparameters of the ONS algorithm, we perform an exhaustive grid search for γ and η and select the hyperparameters that minimize CRPS. Because of the different nature of the case studies, these parameter grids are not identical.

4.2.1. Time-invariant process

We adapt a simulation study from Gneiting and Ranjan (2013), in which the data generating process is the combination of standard normal random variables X_0, X_1, X_2, X_3 and ϵ without a temporal relation:

$$Y = X_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + \epsilon, \quad (17)$$

where ϵ represents an error term. Gneiting and Ranjan (2013) note that X_0 may represent public information, such as weather forecasts from a meteorological office, while X_1, X_2 and X_3 represent private information measured by forecasters 1, 2 and 3, respectively. These experts issue the following probabilistic forecasts:

$$f_1 = \mathcal{N}(X_0 + a_1 X_1, 1 + a_2^2 + a_3^2), \quad (18)$$

$$f_2 = \mathcal{N}(X_0 + a_2 X_2, 1 + a_1^2 + a_3^2), \quad (19)$$

$$f_3 = \mathcal{N}(X_0 + a_3 X_3, 1 + a_1^2 + a_2^2), \quad (20)$$

where $a_1 = a_2 = 1$ and $a_3 = 1.1$. Consequently, the component forecasts are probabilistically calibrated by design and their linear combination leads to overdispersion (Gneiting and Ranjan, 2013). As mentioned, we compare our method with linear and nonlinear combinations whose weights are static over time, denoted with an asterisk. To that end, the parameters are optimized over the test data by minimizing the logarithmic score as in Gneiting and Ranjan (2013).

Simulation results Given the stationary and time-invariant nature of this case study, the hyperparameter grid includes low γ values for increased stability and relatively high η values to learn quickly. Specifically, the grid comprises the

²<https://cran.r-project.org/web/packages/profoc/index.html>

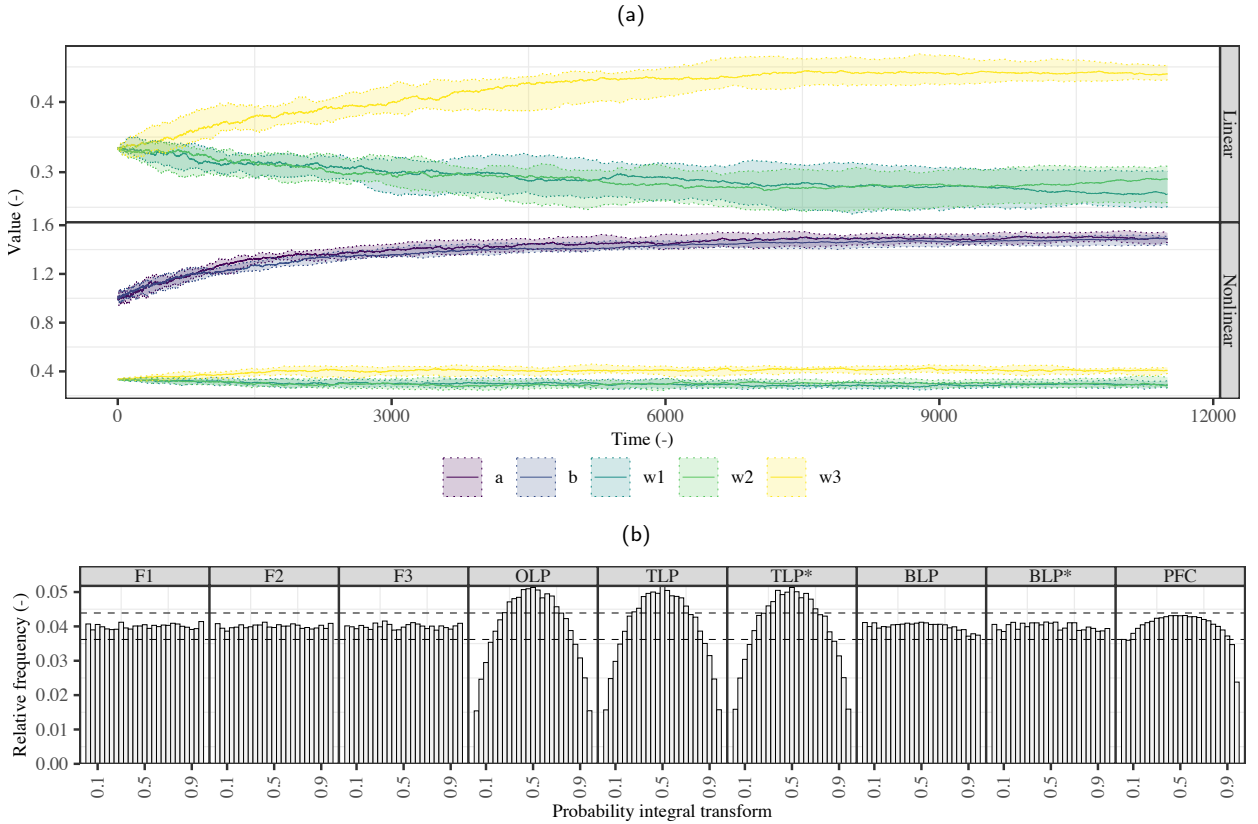


Figure 1: In (a), the evolution of the weights over time organized by linear and nonlinear combination (rows), where the shaded area represents the 90% confidence interval computed over 150 experiments. In (b), the PIT histograms of the experts and combination methods again with the asterisk indicating the optimal parameters in hindsight.

combinations of $\gamma \in \{1/16, 1/21, 1/26, 1/31, 1/36\}$ and $\eta \in \{0.150, 0.175, 0.200, 0.225, 0.250\}$. The combination that minimizes CRPS is $\gamma = 1/26$ and $\eta = 0.25$.

Figure 1a presents the evolution of the parameters over time during linear and nonlinear forecast combination. The shaded area represents the 90% confidence interval associated to the parameter, computed over 150 experiments. When considering linear forecast combination, it becomes apparent that the third expert typically receives the highest weight. This pattern is expected because the third expert consistently provides slightly sharper predictive distributions. The weights of experts 1 and 2 are similar, which can be explained by the fact that their forecasts are identical in expectation. We observe a similar pattern in the case of nonlinear forecast combination. However, both shape parameters are larger than 1, indicating that the Beta transformation sharpens the predictive distributions. Crucially, the shape parameters are very similar and this is expected as their primary purpose is to enhance sharpness.

Figure 1b presents the PIT histograms for the experts and the combination methods. Recall that departures from flatness may arise from a limited-sized test set, which is why Fig. 1b also includes consistency bars. This figure demonstrates that the experts exhibit the intended probabilistic calibration, highlighting that the linear combination methods (namely, OLP, TLP, and TLP*) result in overdispersed forecasts. In contrast, both the online BLP and the fixed BLP* effectively transform the combined forecasts to generate calibrated forecasts, whereas the PFC model tends to be slightly miscalibrated at the outer edge of the distributions.

Considering CRPS, Table 1 shows that all combination methods substantially outperform the component models. Furthermore, it can be seen that OLP, TLP and TLP* perform quite similar. BLP and BLP* perform identically, which can partly be attributed to the fact that the process is time-invariant. In other words, the ONS algorithm converges to the true weights, which implies that logarithmic regret is hereby empirically shown. Surprisingly, PFC underperforms on this synthetic data set although it is important to recall that all settings have been left to their default values.

Table 1

Realized CRPS for the component models and combination methods. The results are presented as the mean and standard error of 150 experiments based on synthetic data adapted from Gneiting and Ranjan (2013). Note that the asterisk indicates a combination method with optimal parameters in hindsight.

F1	F2	F3	OLP	TLP	TLP*	BLP	BLP*	PFC
1.012±0.725	1.01±0.724	0.978±0.700	0.895±0.553	0.893±0.553	0.894±0.555	0.880±0.618	0.880±0.626	0.931±0.639

Finally, we evaluate the computational complexity of Algorithm 1, which is dominated by the computation of the integrals (11), (12) and (13). On a 2020 M1 MacBook Pro running R 4.2.2, it takes approximately 10 ms (per component model), 33 ms and 44 ms to evaluate the aforementioned integrals, respectively. One iteration of Algorithm 1 requires, on average, 110 ms when considering 3 component models.

4.2.2. Time-varying process

We additionally illustrate the efficacy of the proposed method on a simulation study adapted from Berrisch and Ziel (2021). Specifically, the data generating process is defined as:

$$Y_t = \mathcal{N}(0.15 \operatorname{asinh}(\mu_t), 1), \quad (21)$$

where $\mu_t = 0.99\mu_{t-1} + \varepsilon_t$ and ε_t a standard normal random variable. In this study, two experts provide constant probabilistic forecasts:

$$f_1 = \mathcal{N}(-1, 1), \quad (22)$$

$$f_2 = \mathcal{N}(3, 4). \quad (23)$$

Similar to the previous case study, we compare our method with linear and nonlinear combination methods whose parameters are optimal in hindsight, learned by minimizing the logarithmic score over the test set and indicated by an asterisk.

Simulation results Here, we select a grid that includes higher γ values than in the previous case study to increase adaptation speed, while retaining the same η values. Specifically, the grid comprises the combinations of $\gamma \in \{0.175, 0.200, 0.225, 0.250, 0.275, 0.300, 0.325\}$ and $\eta \in \{0.150, 0.175, 0.200, 0.225, 0.250\}$. The combination that minimizes CRPS is $\gamma = 0.275$ and $\eta = 0.175$.

The forecasts by the two experts in this case study are both biased; however, the first expert does issue forecasts that are correctly dispersed, whereas the second expert issues forecasts that are overdispersed. Consequently, Fig. 2a shows that both linear and nonlinear combination weigh the first expert more than the second (recall that the shaded area represents the 90% confidence interval associated with the parameters). To optimize the calibration of the resulting combination, TLP (the linear combination method) distributes the weights to include both experts to a degree. In contrast, BLP quickly disregards the second expert and instead uses the shape parameters to modify the predictive distributions. Note that the shape parameters are quite different from each other; this is because they have to account for the bias of the first expert.

Figure 2b reveals the miscalibration of the two experts. Their miscalibration is such that linearly combining the forecasts does not result in calibrated forecasts, which can be expected because of the overdispersion of the second expert and their opposing biases. Nonlinear combination significantly improves calibration and the online method is preferred over the static method (indicated with an asterisk). Nevertheless, these combined forecasts are not perfectly calibrated at the edges of the distributions. In this case study, the PFC generates forecasts that are closest to perfect calibration although also here some deviation can be observed at the edges.

In terms of CRPS, Table 2 shows that BLP* outperforms the online method, i.e., BLP. This is not entirely unexpected since the data generating process is stationary, meaning that the best strategy in hindsight can be expected to be highly competitive. The PFC method outperforms others, primarily because it combines quantiles, allowing it to incorporate the strengths of component forecasts exhibiting biases in opposing directions. In terms of linear combination, we observe that TLP outperforms TLP*, indicating that online learning does benefit linear forecast combination.

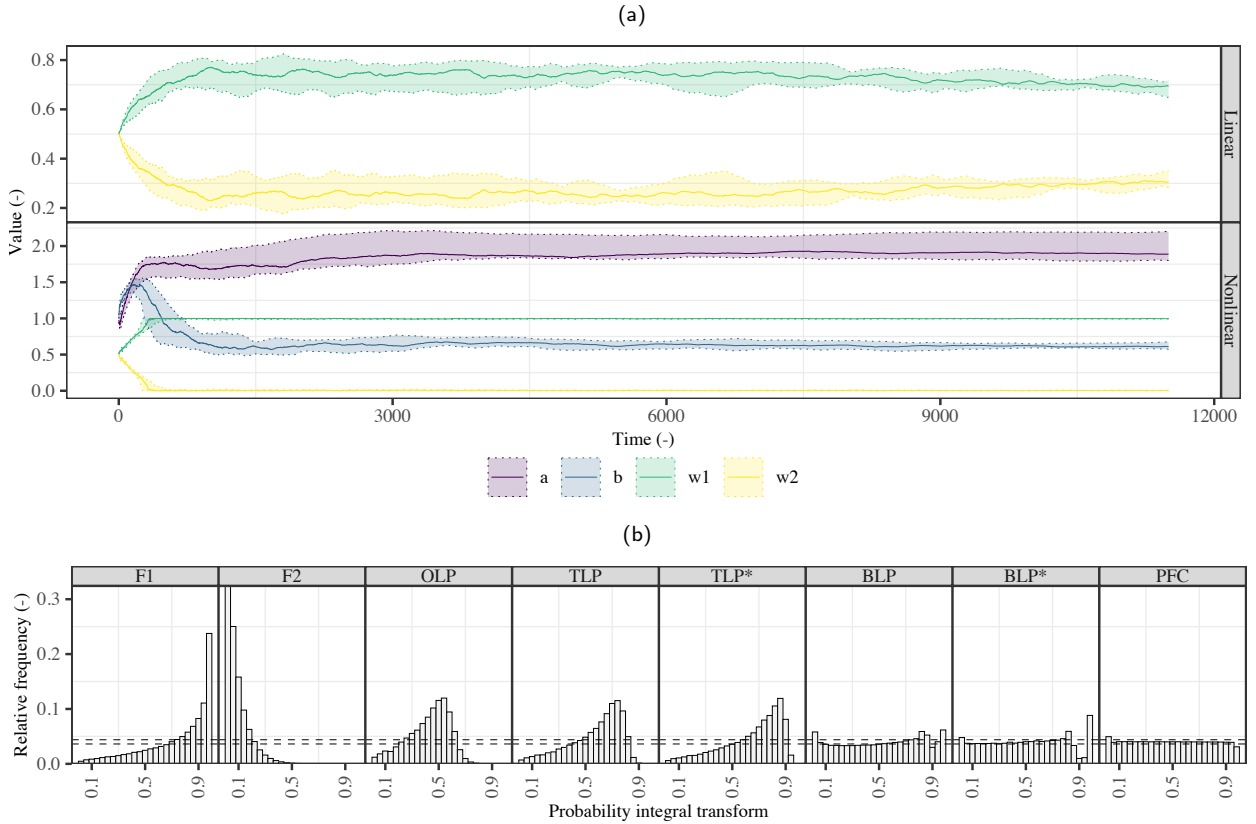


Figure 2: In (a), the evolution of the weights over time organized by linear and nonlinear combination (rows), where the shaded area represents the 90% confidence interval computed over 150 experiments. In (b), the PIT histograms of the experts and combination methods again with the asterisk indicating the optimal parameters in hindsight.

Table 2

Realized CRPS for the component models and combination methods. The results are presented as the mean and standard error of 150 experiments based on synthetic data adapted from Berrisch and Ziel (2021). Note that the asterisk indicates a combination method with optimal parameters in hindsight.

F1	F2	OLP	TLP	TLP*	BLP	BLP*	PFC
0.858±0.645	2.07±0.871	0.871±0.283	0.718±0.327	0.741±0.456	0.609±0.446	0.603±0.437	0.588±0.418

4.3. Real-word time series

We apply the proposed combination method on probabilistic power forecasts issued for a wind farm located in mid-west France. The wind farm has a nominal capacity of 16,000 kW and the data set ranges from 2018-09-30 until 2020-09-30 at 15 min resolution. To flag anomalies, we use the OpenOA library in Python and linearly interpolate flagged instances (Perr-Sauer, Optis, Fields, Bodini, Lee, Todd, Simley, Hammond, Phillips, Lunacek, Kemper, Williams, Craig, Agarwal, Sheng and Meissner, 2021).

For brevity, we consider only 15 min ahead, 3 h, 6 h and 24 h ahead forecasts. Especially further into the future, NWP forecasts are essential. Furthermore, Winkler et al. (2019) argue that weighted forecast combination of experts that are highly correlated can become unstable. We therefore include forecasts from three NWP models, namely (i) the High-Resolution Forecast (HRES) from the European Centre for Medium-Range Weather Forecasts (ECMWF), (ii) the Global Forecast System (GFS) from the National Centers for Environmental Prediction (NCEP), and (iii) the global Arpège model from Météo France. The average of the four NWP grid points closest to the wind farm, as well as the last observed power and wind speed, serve as input to three machine learning models that we describe next.

Table 3

The combination of γ and η that minimize CRPS for linear and nonlinear forecast combination, i.e., TLP and BLP respectively, and for all forecast horizons.

	TLP				BLP			
	15 min	3 h	6 h	24 h	15 min	3 h	6 h	24 h
γ	0.58	0.46	0.70	0.58	0.22	0.70	0.46	0.22
η	0.58	0.22	0.58	0.34	0.46	0.70	0.70	0.70

The last stage before forecast combination is to use post-processing models to convert the wind speed forecasts into probabilistic wind power forecasts that are to be combined. For this, we employ three models, specifically (i) Generalized Boosted Regression Models (GBM), (ii) Quantile Regression Forests (QRF), and (iii) Quantile Regression (QR). GBM is a technique where simple models, here regression trees, are repeatedly combined in a stage-wise fashion to minimize the prediction loss (Friedman, 2001). In contrast, QRF is based on random forests in which independent regression trees are fitted on subsets of the feature space and sample space, making them easily parallelizable (Meinshausen, 2006). Finally, QR is a linear model where the pinball loss is minimized to predict conditional quantiles (Koenker and Bassett, 1978). A thorough treatment of these models is out of the scope of this manuscript and we refer to standard textbooks such as Hastie, Tibsharani and Friedman (2008). Furthermore, it is important to note that the aim of this study is not to generate the most accurate component forecasts; rather, the aim is to show the effectiveness of probabilistic forecast combination with our proposed method. The aforementioned models have been selected because they are well documented and can easily be used in R or Python. We use the R packages `gbm` to implement GBM (Greenwell, Boehmke, Cunningham and Developers, 2022), `quantregForest` to implement QRF (Meinshausen, 2017) and `quantreg` to implement QR (Koenker, 2022). Finally, we use the function `contCDF` of the R package `ProbCast` to convert quantile forecasts into continuous CDFs with generalized Pareto distribution tails (Browell, Gilbert, McFadzean and Tawn, 2022).

We perform an exhaustive grid search to select the hyperparameters that minimize CRPS for each machine learning model. Given that there are in total 216 combinations to be validated³, we restrict ourselves to a single forecast horizon (3 h) and a single wind park, thus resulting in 54 combinations. This strategy can be motivated by the fact that the post-processing models are expected to have a similar learning pattern across horizons, except for perhaps the first forecast horizon. For hyperparameter validation, we train each model on 5 weeks of data and validate on the following 3 weeks, which is repeated 12 times in a rolling fashion to cover the entire first year. Similarly, we perform an exhaustive grid search to determine the hyperparameters that minimize CRPS when running the ONS algorithm. Specifically, the grid comprises the combinations of $\gamma = \eta \in \{0.10, 0.22, 0.34, 0.46, 0.58, 0.70\}$. The first four months of the test set are used to validate the ONS algorithm hyperparameters, whereas the remaining eight months are used to test the selected ONS hyperparameters. Table 3 presents the hyperparameter combinations for the linear and nonlinear forecast combination methods and the forecast horizons that minimize CRPS. It is interesting to note that the hyperparameters vary as a function of combination methods and the forecast horizon, indicating that careful validation is necessary. As before, we compare the linear and nonlinear combination methods with the optimal parameters in hindsight, i.e., the parameters are learned based on out-of-sample test data. These static models are denoted with an asterisk.

4.3.1. Measure-oriented forecast analysis

Table 4 presents forecast results across the entire test set for all combinations of weather models, static and online combination methods, as well as the post-processing methods, in terms of CRPS. In the table, the best performing combination of weather model and post-processing model, i.e., expert, is underlined while the best performing combination method is in bold font. Table 4 indicates that there is minimal distinction among the models for the initial forecast horizon. It is worth highlighting that, in this case, the standard deviation of the CRPS surpasses its mean value. An in-depth analysis (which is not presented here) unveiled the skewness in the CRPS distribution. Specifically, the 25th percentile and median CRPS values are considerably lower than the mean CRPS, and there is also a noteworthy maximum value, approaching approximately 0.76 (varies depending on the model).

³Four forecast horizons, one wind park, three data sources, three forecast models and six hyperparameter combinations results in 216 combinations.

Table 4

Average and standard deviation of the CRPS in percentage of nominal capacity, computed over the entire test set. The best performing expert is underlined, whereas the best performing combination method is in bold. Note that the asterisk indicates a combination method with optimal parameters in hindsight.

		Post-process model	Horizon			
			15 min	3 h	6 h	24 h
ECMWF	QRF	2.64±3.17	<u>6.33±6.44</u>	6.70±6.67	7.13±6.80	
	QR	2.66±3.44	<u>8.62±8.46</u>	10.9±9.69	13.2±12.0	
	GBM	<u>2.61±3.30</u>	6.67±6.40	<u>6.47±6.55</u>	<u>6.57±6.85</u>	
GFS	QRF	2.66±3.15	7.11±7.00	7.75±7.38	8.53±7.89	
	QR	2.66±3.42	8.61±8.39	10.9±9.65	13.2±11.9	
	GBM	2.62±3.30	7.34±6.76	7.55±7.12	8.01±7.80	
MF	QRF	2.64±3.15	6.52±6.45	6.89±6.64	7.67±7.35	
	QR	2.66±3.45	8.62±8.45	10.9±9.66	13.1±11.9	
	GBM	2.62±3.31	6.87±6.34	6.77±6.46	7.13±7.21	
Combination	OLP	2.60±3.24	6.83±6.29	7.28±6.20	7.86±6.33	
	TLP	2.60±3.20	6.26±5.92	6.33±5.84	6.44±5.96	
	TLP*	2.60±3.22	6.15±5.94	6.21±5.87	6.32±6.15	
	BLP	2.58±3.28	6.07±6.36	6.00±6.11	6.20±6.07	
	BLP*	2.58±3.30	6.08±6.22	6.31±6.34	6.33±6.25	
	PFC	2.58±3.22	5.97±6.00	6.01±5.97	6.24±6.11	

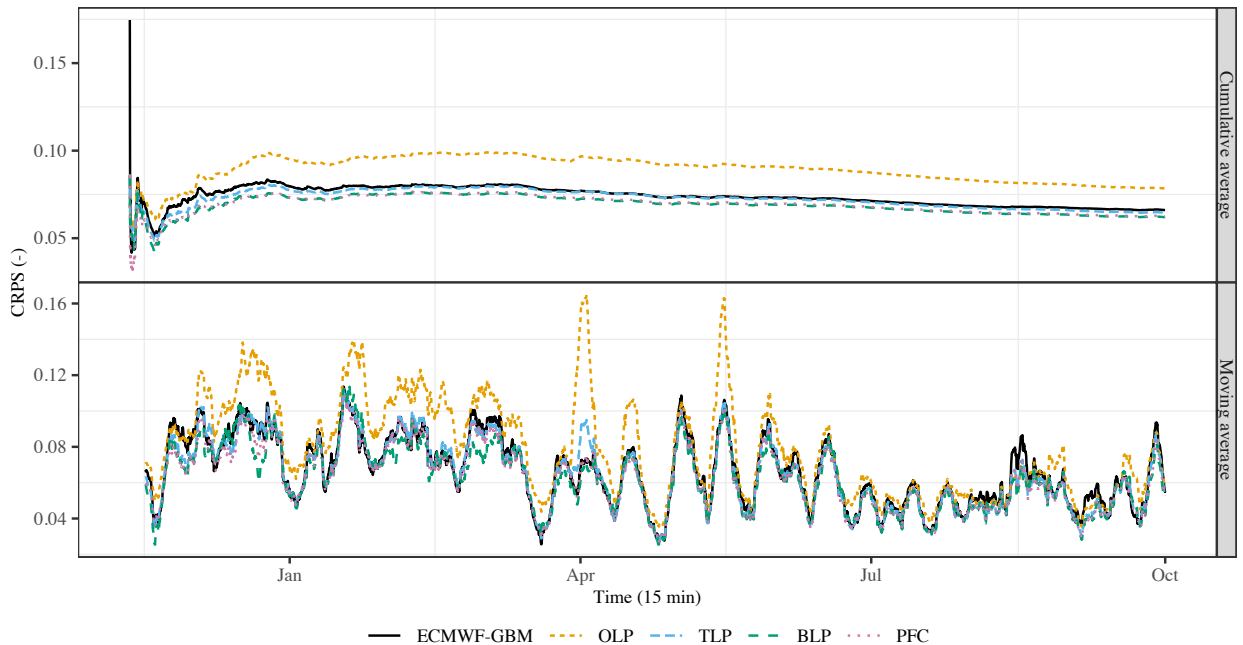


Figure 3: Cumulative average and moving average of the best performing expert as well as the online combination methods for the 24 h ahead forecast horizon.

As expected, Table 4 shows that the CRPS increases with the forecast horizon, with the largest relative increase occurring from 15 min to 3 h ahead. The relative increase is especially noticeable for QR, which can be expected since the linear model fails to accurately represent the nonlinear relationship between wind speed and wind power.

Furthermore, the table shows that weather forecasts from ECMWF generally result in more accurate wind power forecasts, whereas weather forecasts from GFS result in less accurate wind power forecasts. Of the combination methods, it can be seen that the linear static method with optimal parameters in hindsight, i.e., TLP*, achieves lower CRPS than the online method, i.e., TLP. Conversely for nonlinear combination, the proposed method BLP improves CRPS up to 4.91% compared to BLP*, indicating that online learning combined with the additional flexibility of the Beta transformation is a valuable improvement when dealing with nonstationary time series. Moreover, the proposed method improves CRPS up to 7.26% when compared to the most accurate expert. In comparison with PFC, the difference in CRPS varies per forecast horizon and is minimal.

Figure 3 presents the CRPS as well, except as cumulative and five-day moving averages for the most accurate expert and the combination methods, and only for the 24 h ahead horizon. In terms of cumulative average, the figure shows that BLP is consistently on par with PFC and that together they outperform the other combination methods, as well as the most accurate expert. Although OLP performs worst in this figure, it is worth noting that it is a competitive method when compared to the experts (cf. Table 4). Nonetheless, when examining the moving average, it becomes evident that adaptive techniques prove beneficial in instances where one or more experts underperform. This is notably the case prior to January and around the outset of April. Particularly during the latter period, it is evident that the fixed method OLP exhibits significantly poorer performance compared to the top-performing expert, along with TLP, PFC, and BLP, with the latter demonstrating the best performance during this period. Concerning the period prior to January, it is noteworthy that BLP exhibits a considerably superior performance compared to the other methods, suggesting the important influence of shape parameters. This aspect will be explored further in the upcoming section.

4.3.2. Combination weights

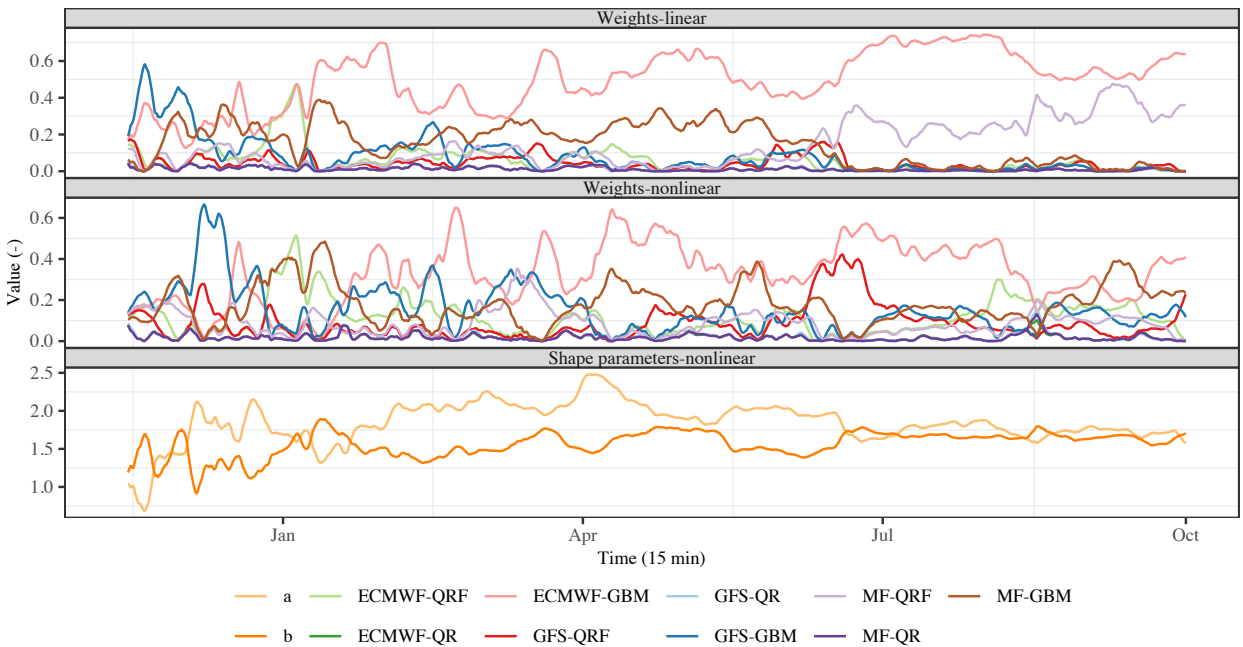


Figure 4: The evolution of linear and nonlinear parameters over time, color coded according to the combination of weather model and post-processing model for the 24 h forecast horizon.

Figure 4 presents the three-day moving average of the weights as they evolve throughout the test period, organized by linear and nonlinear weights, as well as the shape parameters used by the nonlinear forecast combination algorithm for the 24 h forecast horizon.

For the case of linear combination, Fig. 4 shows that expert ECMWF-GBM generally is given the most weight, which is consistent with the results from Table 4. It is worth highlighting, however, the varying weights assigned to experts during different periods. Notably, expert GFS-GBM carries the highest weight during the initial period, while expert MF-QRF assumes significant weighting in the later stages of the test set. Particularly noteworthy is the fact that

GFS-GBM maintains a weight exceeding 0.3 for an extended duration, despite not performing as competitively across the entire test set (as shown in Table 4).

When we consider nonlinear combination, as illustrated in Fig. 4, it is evident that expert ECMWF-GBM maintains the highest weighting. However, the remaining experts are allocated substantial weights, with the exception of those relying on QR as a post-processing model. Notably, during the last period of the test set, 4 experts are assigned weights more than 0.1 while the shape parameters are close to being identical and larger than 1. The latter suggests that these experts are overdispersed during the final period and that the proposed algorithm leverages the shape parameters to enhance the probabilistic calibration, similar as in Section 4.2.1.

4.3.3. Distribution-oriented forecast analysis

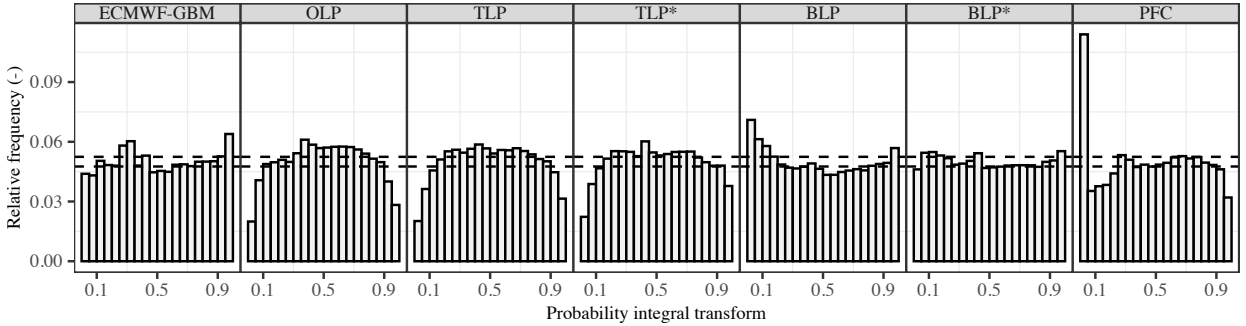


Figure 5: PIT histograms of the component and combination models again with the asterisk indicating the optimal combination with static weights.

Finally, we examine the probabilistic calibration using PIT histograms, which are presented in Fig. 5. Recall that deviations from flatness can be due to a test set of limited size, which is why Fig. 5 additionally presents the consistency bars. The figure illustrates that the most accurate expert aligns closely with probabilistic calibration. Although not shown here, it is worth noting that the majority of experts exhibit satisfactory calibration in the central portion of their predictive distributions. However, all experts do experience slight miscalibration in the outer regions of their predictive distributions.

As the experts are already approaching probabilistic calibration, employing linear combination methods such as OLP, TLP and TLP* leads to overdispersed forecasts. The optimal nonlinear combination in hindsight, BLP*, is closest to perfect calibration, again indicating that the Beta transformation is capable of generating competitive forecasts that are calibrated as well. However, BLP shows signs of miscalibration at the lower end of the predictive distributions, suggesting a positive bias—a characteristic not observed in the predictions of the experts. This discrepancy can likely be attributed to the period up to and including April, during which the shape parameter a exceeds shape parameter b , as depicted in Fig. 4. In general, setting $a > b$ shifts the predictive distribution upwards, potentially introducing a positive bias when the component forecasts are unbiased. One possible remedy for this issue could involve periodic recalibration of the hyperparameters of ONS. It is worth noting that benchmark PFC lags in terms of probabilistic calibration, exhibiting a notable deviation at the lower end of the predictive distributions.

5. Conclusions

In this study, we have expanded the application of the beta-transformed linear pool to the online setting by deriving the gradient of the continuous ranked probability score (CRPS) with respect to the shape parameters of the Beta distribution and the combination weights. We have selected the CRPS because it evaluates the entire predictive distribution and therefore yields the most general setting where experts issue complete predictive distributions. In addition, the CRPS is exponential-concave, which allows for accelerated learning in combination with the Online Newton Step (ONS) algorithm. The motivation for our approach is that linear combination, which is the predominant form of forecast combination, always leads to overdispersed forecasts in case the experts are probabilistically calibrated, which is a requirement in probabilistic forecasting. In a time-invariant simulation study using synthetic data, we have shown that the proposed method converges to the optimal combination strategy in hindsight, meaning that the average

regret goes to zero. In another simulation study with time-varying synthetic data, we have shown that the proposed method approaches the optimal combination strategy in hindsight but does not achieve zero average regret. Finally, we employed the proposed method to combine probabilistic forecasts of nine experts, resulting from all combinations of three weather models and three post-processing methods, on a real-world wind power data set. We showed that the online and offline combination methods, except the naive version, always perform as well as the best expert. More importantly, the proposed method outperformed the most accurate expert by up to 7.26% and the optimal combination strategy in hindsight by up to 4.91% in terms of CRPS, indicating that online learning combined with the additional flexibility of the Beta transformation is a valuable improvement for nonstationary time series. In all case studies, we observed that the ONS algorithm, which uses the derived gradient to update the parameters, is sensitive to its hyperparameters. Hence, it is important to carefully validate these to optimize the performance of the algorithm.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors acknowledge ECMWF, NCEP and Météo France for providing numerical weather predictions. The present research was carried as part of the Smart4RES Project (European Union's Horizon 2020, No. 864337).

CRedit authorship contribution statement

Dennis van der Meer: Conceptualization, Data Curation, Investigation, Methodology, Software, Visualization, Writing - Original Draft, Writing - Review & Editing. **Pierre Pinson:** Conceptualization, Funding Acquisition, Writing - Original Draft, Writing - Review & Editing. **Simon Camal:** Data Curation, Funding Acquisition, Project Administration, Software, Writing - Original Draft. **Georges Kariniotakis:** Funding Acquisition, Project Administration, Resources, Writing - Original Draft.

A. Derivation of the gradient

A.1. Partial derivative with respect to the weights

Recall that we can compute $\frac{\partial \text{CRPS}}{\partial w_j}$ as follows:

$$\frac{\partial \text{CRPS}}{\partial w_j} = \int_0^1 2 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \frac{\partial \hat{F}_{a,b}(z)}{\partial w_j} dx, \quad (24)$$

Also recall that the CDF of the Beta distribution is a quotient where the denominator only depends on a and b . Therefore, we focus on the incomplete beta function $B_{a,b}(z)$. As mentioned, the weights appear in the upper limit of the integral in (4) as $z = \sum_{j=1}^m w_j \hat{F}_j(y)$ and we therefore require Leibniz's integral rule. In general terms, it is defined as follows:

$$\frac{d}{dx} \left(\int_{a(x)}^{b(x)} f(x, t) dt \right) = f(x, b(x)) \frac{d}{dx} b(x) - f(x, a(x)) \frac{d}{dx} a(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x, t) dt. \quad (25)$$

Now, we want to find partial derivative $\frac{\partial}{\partial w_j}$, so we plug (4) into (25):

$$\begin{aligned} \frac{\partial B_{a,b}(z)}{\partial w_j} &= \frac{\partial}{\partial w_j} \left(\int_0^z u^{a-1} (1-u)^{b-1} dt \right) \\ &= z^{a-1} (1-z)^{b-1} \underbrace{\frac{\partial}{\partial w_j} z - 0^{a-1} (1-0)^{b-1} \frac{\partial}{\partial w_j} 0}_{=0} + \underbrace{\int_0^z \frac{\partial}{\partial w_j} u^{a-1} (1-u)^{b-1} dt}_{=0} \end{aligned}$$

$$= \left(\sum_{j=1}^m w_j \hat{F}_j(y) \right)^{a-1} \left(1 - \sum_{j=1}^m w_j \hat{F}_j(y) \right)^{b-1} \hat{F}_j(y). \quad (26)$$

Accordingly, $\partial I_{a,b}(z)/\partial w_j$ requires multiplication of (26) with $1/B_{a,b}$ since the denominator in the latter term is not affected by the derivative:

$$\frac{\partial I_{a,b}(z)}{\partial w_j} = \frac{1}{B_{a,b}} z^{a-1} (1-z)^{b-1} \hat{F}_j(y). \quad (27)$$

Finally, we apply the logit transform (8) to attain the partial derivative of the CRPS with respect to the j^{th} weight by plugging (27) into (24):

$$\frac{\partial \text{CRPS}}{\partial \tilde{w}_j} = \frac{2(w_j - w_j^2)}{B_{a,b}} \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) z^{a-1} (1-z)^{b-1} \hat{F}_j(x) dx. \quad (28)$$

A.2. Partial derivative with respect to shape parameter a

The Leibniz integral rule of $\partial \text{CRPS}/\partial a$ results in:

$$\frac{\partial \text{CRPS}}{\partial a} = \int_0^1 2(\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \frac{\partial \hat{F}_{a,b}(z)}{\partial a} dx. \quad (29)$$

Consequently, it is necessary to compute the partial derivative of (3) with respect to a . Given that the regularized incomplete Beta function is the ratio of the incomplete Beta function and the complete Beta function, we use the quotient rule. First, we compute $\partial B_{a,b}/\partial a$ by applying the product rule to the final equality in (5):

$$\begin{aligned} \frac{\partial B_{a,b}}{\partial a} &= \frac{\partial}{\partial a} \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \\ &= \frac{\partial}{\partial a} \Gamma(a)\Gamma(b)\Gamma(a+b)^{-1}. \end{aligned}$$

The partial derivative of $\Gamma(a)$ with respect to parameter a is:

$$\begin{aligned} \frac{\partial}{\partial a} \Gamma(a) &= \frac{\partial}{\partial a} \int_0^\infty u^{(a-1)} \exp(-u) du \\ &= \int_0^\infty \frac{\partial}{\partial a} u^{(a-1)} \exp(-u) du \\ &= \int_0^\infty u^{(a-1)} \ln(u) \exp(-u) du \\ &= \psi(a)\Gamma(a), \end{aligned} \quad (30)$$

The partial derivative with respect to a of $\Gamma(a+b)$ can be computed in a similar manner such that $\partial \Gamma(a+b)/\partial a = \psi(a+b)\Gamma(a+b)$. Using this result and (30) in the product rule we find:

$$\begin{aligned} \frac{\partial B_{a,b}}{\partial a} &= \psi(a)\Gamma(a)\Gamma(b)\Gamma(a+b)^{-1} - \Gamma(a)\Gamma(b)\Gamma(a+b)^{-2} \frac{\partial}{\partial a} \Gamma(a+b) \\ &= \frac{\psi(a)\Gamma(a)\Gamma(b)}{\Gamma(a+b)} - \frac{\Gamma(a)\Gamma(b)\psi(a+b)}{\Gamma(a+b)} \\ &= \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} (\psi(a) - \psi(a+b)). \end{aligned} \quad (31)$$

The next step is to compute the partial derivative of the incomplete Beta function in (4):

$$\frac{\partial B_{a,b}(z)}{\partial a} = \frac{\partial}{\partial a} \int_0^z u^{a-1} (1-u)^{b-1} du$$

$$\begin{aligned}
 &= \int_0^z \frac{\partial}{\partial a} u^{a-1} (1-u)^{b-1} du \\
 &= \int_0^z u^{a-1} \ln(u) (1-u)^{b-1} du.
 \end{aligned} \tag{32}$$

The partial derivative can be found through integration by parts and setting $v = \ln(u)$ and $dw = u^{a-1}(1-u)^{b-1}$, which gives $dv = 1/u$ and $w = u^a {}_2F_1(a, 1-b; a+1; u)/a$. Plugging this in to (32) gives:

$$\begin{aligned}
 \int_0^z \ln(u) u^{a-1} (1-u)^{b-1} du &= \frac{\ln(u) u^a {}_2F_1(a, 1-b; a+1; u)}{a} - \int \frac{u^a {}_2F_1(a, 1-b; a+1; u)}{au} du \Big|_0^z \\
 &= \frac{1}{a} \left(\ln(u) u^a {}_2F_1(a, 1-b; a+1; u) - \int u^{a-1} {}_2F_1(a, 1-b; a+1; u) du \right) \Big|_0^z \\
 &= \frac{1}{a} \left(\ln(u) u^a {}_2F_1(a, 1-b; a+1; u) - \frac{u^a}{a} {}_3F_2(a, a, 1-b; a+1, a+1; u) \right) \Big|_0^z \\
 &= \frac{1}{a} \left(\ln(u) u^a \Gamma(a+1) {}_2\tilde{F}_1(a, 1-b; a+1; u) - \frac{u^a}{a} {}_3F_2(a, a, 1-b; a+1, a+1; u) \right) \Big|_0^z \\
 &= \frac{1}{a} \left(\ln(u) u^a a \Gamma(a) {}_2\tilde{F}_1(a, 1-b; a+1; u) - \frac{u^a}{a} {}_3F_2(a, a, 1-b; a+1, a+1; u) \right) \Big|_0^z \\
 &= \frac{1}{a} \left(a B_{a,b}(z) \ln(z) - z^a \frac{1}{a} {}_3F_2(a, a, 1-b; a+1, a+1; z) \right) \\
 &= \frac{1}{a} \left(a B_{a,b}(z) \ln(z) - z^a \frac{1}{a} {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) \Gamma(a+1) \Gamma(a+1) \right) \\
 &= \frac{1}{a} \left(a B_{a,b}(z) \ln(z) - z^a a \Gamma(a)^2 {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) \right) \\
 &= B_{a,b}(z) \ln(z) - z^a \Gamma(a)^2 {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z).
 \end{aligned} \tag{33}$$

The final step to compute the partial derivative of the regularized incomplete Beta function with respect to parameter a is, as mentioned at the beginning of this section, to use the quotient rule:

$$\begin{aligned}
 \frac{\partial I_{a,b}(z)}{\partial a} &= \frac{\left(B_{a,b}(z) \ln(z) - z^a \Gamma(a)^2 {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) \right) \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}}{\left(\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \right)^2} \\
 &\quad - \frac{\Gamma(a) z^a {}_2\tilde{F}_1(a, 1-b; a+1; z) \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} (\psi(a) - \psi(a+b))}{\left(\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \right)^2} \\
 &= \frac{B_{a,b}(z) \ln(z)}{B_{a,b}} - \frac{z^a \Gamma(a)^2 {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z)}{\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}} - \frac{B_{a,b}(z) (\psi(a) - \psi(a+b))}{B_{a,b}} \\
 &= I_{a,b}(z) \ln(z) - \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} z^a \Gamma(a)^2 {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) - I_{a,b}(z) (\psi(a) - \psi(a+b)) \\
 &= I_{a,b}(z) (\ln(z) - \psi(a) + \psi(a+b)) - \frac{\Gamma(a)\Gamma(a+b)}{\Gamma(b)} z^a {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z).
 \end{aligned} \tag{34}$$

As mentioned in Section 3.1.2, the change of variable $\tilde{a} = \ln(a)$ improves the stability of the algorithm. Given that $\partial a / \partial \tilde{a} = a$ and plugging (34) in to (29) in combination with the definition of the beta-transformed linear pool forecast in (2), we find the partial derivative of the CRPS with respect to \tilde{a} :

$$\begin{aligned}
 \frac{\partial \text{CRPS}}{\partial \tilde{a}} &= a \frac{\partial \text{CRPS}}{\partial a} \\
 &= a \int_0^1 2 \left(\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\} \right) \frac{\partial \hat{F}_{a,b}(z)}{\partial a} dx.
 \end{aligned}$$

$$\begin{aligned}
 &= 2a \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \\
 &\times \left(\hat{F}_{a,b}(z) (\ln(z) - \psi(a) + \psi(a+b)) - \frac{\Gamma(a)\Gamma(a+b)}{\Gamma(b)} z^a {}_3\tilde{F}_2(a, a, 1-b; a+1, a+1; z) \right) dx. \quad (35)
 \end{aligned}$$

A.3. Partial derivative with respect to shape parameter b

The aim is to derive the following:

$$\frac{\partial \text{CRPS}}{\partial b} = \int_0^1 2 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \frac{\partial \hat{F}_{a,b}(z)}{\partial b} dx. \quad (36)$$

Since the partial derivative of the gamma function with respect to b is similar to that in (30), we find that:

$$\begin{aligned}
 \frac{\partial B_{a,b}}{\partial b} &= \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} (\psi(b) - \psi(a+b)) \\
 &= B_{a,b} (\psi(b) - \psi(a+b)). \quad (37)
 \end{aligned}$$

For the partial derivative of the incomplete beta function with respect to b , we use the following useful property:

$$I_{a,b}(z) = 1 - I_{b,a}(1-z) \Leftrightarrow \frac{B_{a,b}(z)}{B_{a,b}} = 1 - \frac{B_{b,a}(1-z)}{B_{a,b}}. \quad (38)$$

From equality (38) it follows that $B_{a,b}(z) = B_{a,b} - B_{b,a}(1-z)$. Considering the definition of the incomplete beta function in (4), switching a and b is important to attain a solution that depends predominantly on b . Then:

$$\begin{aligned}
 \frac{\partial B_{a,b}(z)}{\partial b} &= \frac{\partial B_{a,b}}{\partial b} - \frac{\partial B_{b,a}(1-z)}{\partial b} \\
 &= B_{a,b} (\psi(b) - \psi(a+b)) - \int_0^{1-z} \ln(u) u^{b-1} (1-u)^{a-1} du. \quad (39)
 \end{aligned}$$

Similar to the partial derivative with respect to a , we use integration by parts and some algebra to find that:

$$\begin{aligned}
 \frac{\partial B_{a,b}(z)}{\partial b} &= B_{a,b} (\psi(b) - \psi(a+b)) - \frac{1}{b} \left(\ln(u) u^b {}_2F_1(b, 1-a; b+1; u) - \int u^{b-1} {}_2F_1(b, 1-a; b+1; u) du \right) \Big|_0^{1-z} \\
 &= B_{a,b} (\psi(b) - \psi(a+b)) - \frac{1}{b} \left(\ln(u) u^b {}_2F_1(b, 1-a; b+1; u) - u^b b \Gamma(b)^2 {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; u) \right) \Big|_0^{1-z} \\
 &= B_{a,b} (\psi(b) - \psi(a+b)) + (1-z)^b \Gamma(b)^2 {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) - \ln(1-z) B_{b,a}(1-z). \quad (40)
 \end{aligned}$$

Using (37), (40) and the quotient rule, we can compute the partial derivative of the regularized incomplete beta function with respect to b :

$$\begin{aligned}
 \frac{\partial I_{a,b}(z)}{\partial b} &= \frac{(B_{a,b} (\psi(b) - \psi(a+b)) + (1-z)^b \Gamma(b)^2 {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) - \ln(1-z) B_{b,a}(1-z)) B_{a,b}}{B_{a,b}^2} \\
 &\quad - \frac{B_{a,b}(z) B_{a,b} (\psi(b) - \psi(a+b))}{B_{a,b}^2} \\
 &= \frac{\Gamma(a+b)\Gamma(b)}{\Gamma(a)} (1-z)^b {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) - \ln(1-z) \frac{B_{b,a}(1-z)}{B_{a,b}} \\
 &\quad + (\psi(b) - \psi(a+b)) \left(\frac{B_{a,b}}{B_{a,b}} - \frac{B_{a,b}(z)}{B_{a,b}} \right)
 \end{aligned}$$

$$= \frac{\Gamma(a+b)\Gamma(b)}{\Gamma(a)}(1-z)^b {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) + I_{b,a}(1-z)(\psi(b) - \psi(a+b) - \ln(1-z)). \quad (41)$$

The final step is to plug (41) into (36) in combination with the change of variable:

$$\begin{aligned} \frac{\partial \text{CRPS}}{\partial \tilde{b}} &= b \frac{\partial \text{CRPS}}{\partial b} \\ &= b \int_0^1 2(\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \frac{\partial \hat{F}_{a,b}(z)}{\partial b} dx. \\ &= 2b \int_0^1 (\hat{F}_{a,b}(z) - \mathbb{1}\{x \geq y\}) \\ &\quad \times \left(\frac{\Gamma(a+b)\Gamma(b)}{\Gamma(a)}(1-z)^b {}_3\tilde{F}_2(b, b, 1-a; b+1, b+1; 1-z) \right. \\ &\quad \left. + \hat{F}_{b,a}(1-z)(\psi(b) - \psi(a+b) - \ln(1-z)) \right) dx. \end{aligned} \quad (42)$$

B. Weighted projection on the simplex

Recall that the projection of the updated weights \mathbf{v}_{t+1} onto Λ with weighted norm $\|\cdot\|_{\mathbf{D}_t}$, where $\mathbf{D}_t = \text{diag}(\mathbf{A}_t)$, is the following convex optimization problem (Wintenberger, 2021)

$$\mathbf{w}_{t+1} = \frac{1}{2} \arg \min_{\mathbf{w} \in \mathcal{K}} \|\mathbf{w} - \mathbf{v}_{t+1}\|_{\mathbf{D}_t}^2. \quad (43)$$

From this, the Lagrangian can be derived:

$$L(\mathbf{w}, \lambda, \nu) = \frac{1}{2} (\mathbf{w} - \mathbf{v}_{t+1})^\top \mathbf{D}_t (\mathbf{w} - \mathbf{v}_{t+1}) - \sum_{j=1}^m \lambda_j w_j + \nu (\mathbf{b}^\top \mathbf{w} - 1), \quad (44)$$

where $\mathbf{b} = [\mathbf{1}_m]^\top$ is a vector of ones, λ_j is a dual variable associated with the nonnegativity constraint of the j^{th} component forecast weight and ν the dual variable associated with the equality constraint. The next step is to compute the gradient:

$$\nabla L(\mathbf{w}, \lambda, \nu) = \begin{pmatrix} \mathbf{D}_t (\mathbf{w} - \mathbf{v}_{t+1}) - \lambda + \nu \mathbf{b}^\top \\ -\mathbf{w} \\ \mathbf{b}^\top \mathbf{w} - 1 \end{pmatrix}. \quad (45)$$

Setting the gradient (45) to 0, we find the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{cases} \mathbf{w}^* = \mathbf{v}_{t+1} - \mathbf{D}_t^{-1} (\nu^* \mathbf{b}^\top + \lambda^*) \\ \mathbf{b}^\top \mathbf{w}^* = 1 \\ w_j^* = 0 \text{ or } w_j^* > 0 \text{ and } \lambda_j^* = 0 \end{cases}, \quad (46)$$

which results in the weighted soft-threshold (Wintenberger, 2021)

$$\begin{aligned} \mathbf{w}^* &= \max(\mathbf{v}_{t+1} - \mathbf{D}_t^{-1} \nu^* \mathbf{b}^\top, 0) \\ &= \mathbf{D}_t^{-1} \text{SoftThreshold}(\mathbf{D}_t \mathbf{v}_{t+1}, \nu^*). \end{aligned} \quad (47)$$

Subsequently, set $\|\mathbf{w}^*\|_0 = d_0$ to establish the relation

$$1 = \sum_{i=1}^{d_0} w_i^* = \sum_{i=1}^{d_0} \mathbf{D}_t^{-1} \text{SoftThreshold}(\mathbf{D}_t \mathbf{v}_{t+1}, \nu^*) = \sum_{i=1}^{d_0} v_{t+1,i}^* - \sum_{i=1}^{d_0} d_{t,i}^{-1} \nu^*, \quad (48)$$

where $d_{t,i}$ is the i^{th} diagonal element of \mathbf{D}_t with identical ordering, which implies that

$$v^* = \frac{1}{\sum_{i=1}^{d_0} d_{t,i}^{-1}} \left(\sum_{i=1}^{d_0} v_{t+1,i} - 1 \right). \quad (49)$$

References

- Bassetti, F., Casarin, R., Ravazzolo, F., 2018. Bayesian nonparametric calibration and combination of predictive distributions. *Journal of the American Statistical Association* 113, 675–685. doi:10.1080/01621459.2016.1273117.
- Berrisch, J., Ziel, F., 2021. CRPS learning. *Journal of Econometrics* doi:doi.org/10.1016/j.jeconom.2021.11.008.
- Bracale, A., Carpinelli, G., De Falco, P., 2017. A probabilistic competitive ensemble method for short-term photovoltaic power forecasting. *IEEE Transactions on Sustainable Energy* 8, 551–560. doi:10.1109/TSTE.2016.2610523.
- Bracale, A., Carpinelli, G., De Falco, P., 2019. Developing and comparing different strategies for combining probabilistic photovoltaic power forecasts in an ensemble method. *Energies* 12. doi:10.3390/en12061011.
- Bröcker, J., Smith, L.A., 2007. Increasing the reliability of reliability diagrams. *Weather Forecasting* 22, 651–661. doi:10.1175/WAF993.1.
- Browell, J., Gilbert, C., McFadzean, G., Tawn, R., 2022. ProbCast. doi:10.5281/zenodo.6035270.
- Claeskens, G., Magnus, J.R., Vasnev, A.L., Wang, W., 2016. The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting* 32, 754–762. doi:https://doi.org/10.1016/j.ijforecast.2015.12.005.
- Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 1189–1232. doi:10.1214/aos/1013203451.
- Gaillard, P., Stoltz, G., van Erven, T., 2014. A second-order bound with excess losses, in: Balcan, M.F., Feldman, V., Szepesvári, C. (Eds.), *Proceedings of The 27th Conference on Learning Theory, PMLR, Barcelona, Spain*. pp. 176–196.
- Gneiting, T., Balabdaoui, F., Raftery, A.E., 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69, 243–268. doi:https://doi.org/10.1111/j.1467-9868.2007.00587.x.
- Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102, 359–378. doi:10.1198/016214506000001437.
- Gneiting, T., Ranjan, R., 2013. Combining predictive distributions. *Electronic Journal of Statistics* 7, 1747–1782. doi:doi.org/10.1214/13-EJS823.
- Greenwell, B., Boehmke, B., Cunningham, J., Developers, G., 2022. gbm: Generalized Boosted Regression Models. URL: https://CRAN.R-project.org/package=gbm. r package version 2.1.8.1.
- Hall, S.G., Mitchell, J., 2007. Combining density forecasts. *International Journal of Forecasting* 23, 1–13. doi:https://doi.org/10.1016/j.ijforecast.2006.08.001.
- Hastie, T., Tibsharani, R., Friedman, J., 2008. *The Elements of Statistical Learning*. 2 ed., Springer Series in Statistics.
- Hazan, E., 2021. *Introduction to Online Convex Optimization*. The MIT Press, Cambridge, Massachusetts.
- Hazan, E., Agarwal, A., Kale, S., 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69, 169–192. doi:10.1007/s10994-007-5016-8.
- Held, M., Wolfe, P., Crowder, H., 1974. Validation of subgradient optimization. *Mathematical Programming* 6, 62–88. doi:10.1007/BF01580223.
- Hora, S.C., 2004. Probability judgments for continuous quantities: Linear combinations and calibration. *Management Science* 50, 597–604. doi:10.1287/mnsc.1040.0205.
- Jose, V.R.R., Grushka-Cockayne, Y., Lichtendahl, K.C., 2014. Trimmed opinion pools and the crowd’s calibration problem. *Management Science* 60, 463–475. doi:10.1287/mnsc.2013.1781.
- Koenker, R., 2022. quantreg: Quantile Regression. URL: https://CRAN.R-project.org/package=quantreg. r package version 5.94.
- Koenker, R., Bassett, G., 1978. Regression Quantiles. *Econometrica* 46, 33–50. doi:10.2307/1913643.
- Korotin, A., V’yugin, V., Burnaev, E., 2021. Mixability of integral losses: A key to efficient online aggregation of functional and probabilistic forecasts. *Pattern Recognition* 120, 108175. doi:https://doi.org/10.1016/j.patcog.2021.108175.
- Krannichfeldt, L.V., Wang, Y., Zufferey, T., Hug, G., 2022. Online ensemble approach for probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy* 13, 1221–1233. doi:10.1109/TSTE.2021.3124228.
- Lichtendahl, K.C., Grushka-Cockayne, Y., Winkler, R.L., 2013. Is it better to average probabilities or quantiles? *Management Science* 59, 1594–1611. doi:10.1287/mnsc.1120.1667.
- Martin, G.M., Loaiza-Maya, R., Maneesoonthorn, W., Frazier, D.T., Ramírez-Hassan, A., 2022. Optimal probabilistic forecasts: When do they work? *International Journal of Forecasting* 38, 384–406. doi:https://doi.org/10.1016/j.ijforecast.2021.05.008.
- Van der Meer, D., Camal, S., Kariniotakis, G., 2022. Generalizing renewable energy forecasting using automatic feature selection and combination, in: 2022 17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), pp. 1–6. doi:10.1109/PMAPS53380.2022.9810647.
- Meinshausen, N., 2006. Quantile regression forests. *Journal of Machine Learning Research* 7, 983–999. URL: http://jmlr.org/papers/v7/meinshausen06a.html.
- Meinshausen, N., 2017. quantregForest: Quantile Regression Forests. URL: https://CRAN.R-project.org/package=quantregForest. r package version 1.3-7.
- Möller, A., Groß, J., 2020. Probabilistic temperature forecasting with a heteroscedastic autoregressive ensemble postprocessing model. *Quarterly Journal of the Royal Meteorological Society* 146, 211–224. doi:https://doi.org/10.1002/qj.3667.
- Orabona, F., 2019. *A modern introduction to online learning*. URL: https://arxiv.org/abs/1912.13213, doi:10.48550/ARXIV.1912.13213.

- Perr-Sauer, J., Optis, M., Fields, J.M., Bodini, N., Lee, J.C., Todd, A., Simley, E., Hammond, R., Phillips, C., Lunacek, M., Kemper, T., Williams, L., Craig, A., Agarwal, N., Sheng, S., Meissner, J., 2021. OpenOA: An Open-Source Codebase For Operational Analysis of Wind Farms. *Journal of Open Source Software* 6, 2171. doi:10.21105/joss.02171.
- Pinson, P., Madsen, H., 2012. Adaptive modelling and forecasting of offshore wind power fluctuations with markov-switching autoregressive models. *Journal of Forecasting* 31, 281–313. doi:doi.org/10.1002/for.1194.
- Raftery, A.E., Gneiting, T., Balabdaoui, F., Polakowski, M., 2005. Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review* 133, 1155–1174. doi:https://doi.org/10.1175/MWR2906.1.
- Rosenblatt, M., 1952. Remarks on a multivariate transformation. *Annals of Mathematical Statistics* 23, 470–472. doi:https://doi.org/10.1214/aoms/1177729394.
- Sherman, J., Morrison, W.J., 1950. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics* 21, 124 – 127. doi:10.1214/aoms/1177729893.
- Soll, J.B., Mannes, A.E., Larrick, R.P., 2012. The “wisdom of crowds” effect, in: Pashler, H. (Ed.), *Encyclopedia of Mind*, Sage Publications.
- Stone, M., 1961. The opinion pool. *Annals of Mathematical Statistics* 32, 1339–1342. doi:https://doi.org/10.1214/aoms/1177704873.
- Sánchez, I., 2008. Adaptive combination of forecasts with application to wind energy. *International Journal of Forecasting* 24, 679–693. doi:https://doi.org/10.1016/j.ijforecast.2008.08.008. energy Forecasting.
- Taylor, J.W., Taylor, K.S., 2023. Combining probabilistic forecasts of covid-19 mortality in the united states. *European Journal of Operational Research* 304, 25–41. doi:https://doi.org/10.1016/j.ejor.2021.06.044.
- Thorey, J., Chaussin, C., Mallet, V., 2018. Ensemble forecast of photovoltaic power with online crps learning. *International Journal of Forecasting* 34, 762–773. doi:https://doi.org/10.1016/j.ijforecast.2018.05.007.
- Thorey, J., Mallet, V., Baudin, P., 2017. Online learning with the continuous ranked probability score for ensemble forecasting. *Quarterly Journal of the Royal Meteorological Society* 143, 521–529. doi:https://doi.org/10.1002/qj.2940.
- V'yugin, V.V., Trunov, V.G., 2019. Online learning with continuous ranked probability score, in: Gammerman, A., Vovk, V., Luo, Z., Smirnov, E. (Eds.), *Proceedings of the Eighth Symposium on Conformal and Probabilistic Prediction and Applications*, PMLR. pp. 163–177.
- Wang, Y., Zhang, N., Tan, Y., Hong, T., Kirschen, D.S., Kang, C., 2019. Combining probabilistic load forecasts. *IEEE Transactions on Smart Grid* 10, 3664–3674. doi:10.1109/TSG.2018.2833869.
- Wilks, D.S., 2018. Chapter 3 - univariate ensemble postprocessing, in: Vannitsem, S., Wilks, D.S., Messner, J.W. (Eds.), *Statistical Postprocessing of Ensemble Forecasts*. Elsevier, pp. 49–89. doi:https://doi.org/10.1016/B978-0-12-812372-0.00003-0.
- Winkler, R.L., Grushka-Cockayne, Y., Lichtendahl, K.C., Jose, V.R.R., 2019. Probability forecasts and their combination: A research perspective. *Decision Analysis* 16, 239–260. doi:10.1287/deca.2019.0391.
- Wintenberger, O., 2021. Lecture notes in Online Convex Optimization. <http://wintenberger.fr/ens.html>. [Online; accessed 18-October-2022].
- Zamo, M., Bel, L., Mestre, O., 2021. Sequential aggregation of probabilistic forecasts—application to wind speed ensemble forecasts. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 70, 202–225. doi:https://doi.org/10.1111/rssc.12455.
- Zinkevich, M., 2003. Online convex programming and generalized infinitesimal gradient ascent., in: Fawcett, T., Mishra, N. (Eds.), *ICML, AAAI Press*. pp. 928–936.