Ensemble Postprocessing Using Quantile Function Regression Based on Neural Networks and Bernstein Polynomials

JOHN BJØRNAR BREMNES

Norwegian Meteorological Institute, Oslo, Norway

(Manuscript received 5 July 2019, in final form 6 September 2019)

ABSTRACT

The value of ensemble forecasts is well documented. However, postprocessing by statistical methods is usually required to make forecasts reliable in a probabilistic sense. In this work a flexible statistical method for making probabilistic forecasts in terms of quantile functions is proposed. The quantile functions are specified by linear combinations of Bernstein basis polynomials, and their coefficients are assumed to be related to ensemble forecasts by means of a highly adaptable neural network. This leads to many parameters to estimate, but a recent learning algorithm often applied to deep-learning problems makes this feasible and provides robust estimates. The method is applied to ~2 yr of ensemble wind speed forecasting data at 125 Norwegian stations for lead time +60 h. An intercomparison with two quantile regression methods shows improvements in quantile skill score of nearly 1%. The most appealing feature of the method is arguably its versatility.

1. Introduction

Ensemble forecasts have now been generated for many years and are the main source for weather information beyond a few hours ahead (Toth and Kalnay 1993; Molteni et al. 1996; Richardson 2000). To facilitate decision-making processes the scenarios are often given a probabilistic interpretation. However, model biases and insufficient description of local processes tend to limit their direct use. Statistical methods are therefore widely applied to make ensemble forecasts more reliable. A comprehensive review of the subject is given by Vannitsem et al. (2018).

In statistical calibration of ensemble forecasts the objective is to estimate how the probability distribution of the observational quantity varies with the ensemble forecast. During the last decades several parametric probability distributions as well as mixtures of these have been proposed. The tendency in recent years has been toward using more flexible distributions in order to better describe the underlying forecast uncertainty. For example, a shifted censored gamma distribution was proposed by Scheuerer and Hamill (2015) for precipitation forecasting, while a mixture of truncated normal and lognormal distributions was applied to forecast wind speed by Baran and Lerch (2016). Various mixture distributions with one component distribution for each ensemble member have also been applied since the initial work of Raftery et al. (2005). More flexible statistical models, however, lead to challenging estimation and modeling problems, especially since more or less all distribution parameters naturally depend on the ensemble output in an unknown way and also interact. Simple linear relations are not necessarily appropriate for all parameters. As a consequence, more automated methods based on regression trees, like random forest, have received increased attention (e.g., Taillardat et al. 2016).

Another class of methods for ensemble calibration is quantile regression approaches. These focus only on a set of quantiles in the forecast distributions and put few or no constraints on the overall shape. Quantile regression was first applied to weather forecasting by Bremnes (2004). Here, two other quantile regression methods will be used as benchmarks. The first, constrained quantile regression splines Bremnes (2019), models the relation between observations and ensemble forecasts by means of constrained spline functions, while the second, monotonic composite quantile regression neural network Cannon (2011, 2018), is based

Denotes content that is immediately available upon publication as open access.

Corresponding author: John Bjørnar Bremnes, j.b.bremnes@met.no

DOI: 10.1175/MWR-D-19-0227.1

on adaptable neural network functions. To the author's knowledge the latter's potential in calibrating ensemble forecasts has not yet been fully explored.

The main basis behind this article is the work by Rasp and Lerch (2018) on using neural networks for ensemble postprocessing. For temperature forecasting they considered the normal distribution and let its parameters, the mean and the variance, be output from a neural network. In that way very flexible and complex relations to the ensemble forecasts can be allowed for. By using the continuous ranked probability score as loss function and an estimation method developed for deep-learning problems (Goodfellow et al. 2016) they demonstrated that the parameters could be efficiently estimated and skillful probabilistic forecasts could be made.

The method proposed in this article can be seen as a generalization of the work by Rasp and Lerch (2018) and also as a way to deal with challenges in quantile regression methods. Predictive distributions are here characterized in form of adaptable quantile functions. These are defined by Bernstein polynomials whose coefficients are assumed to be output from a neural network. By adjusting the degree of the polynomials, or equivalently the number of coefficients, the shape of predictive distributions can be made arbitrarily flexible. Parameter estimation is carried out with the same approach as in Rasp and Lerch (2018), but with the quantile score averaged over relevant quantile levels as loss function. Direct use of all ensemble members instead of only a few ensemble statistics is also suggested. The method will be referred to as quantile function regression (QFR). A dataset composed of ensemble wind speed forecasting data at 125 Norwegian sites is used to test the method.

2. Data

The skill of the QFR method is tested on the same data as in Bremnes (2019), and the description of the dataset is derived from that with minor modifications. The dataset is composed of forecasts of wind speed at 10-m height at 125 Norwegian synoptic stations from the European Center for Medium Range Forecasts (ECMWF) ensemble prediction system (ECMWF ENS) and corresponding measurements of hourly maximum 10-min-averaged wind speed. For the period under study, the ECMWF ENS had a horizontal resolution of about 32 km and consisted of 51 members: 1 control member and 50 perturbed members. All forecasts were generated at 0000 UTC with a lead time of +60 h and bilinearly interpolated to the locations of the stations. The observations and corresponding forecasts are from 21 November 2013 to 31 December 2015.

The locations of the stations are shown in Fig. 1 along with the topography, the average observed wind speed, and the linear correlation between the control run forecast and the observed wind speed. Norway has a varied topography, with valleys, fjords, and mountains with peaks up to almost 2500 m above the sea level. The strongest wind speeds are observed along the coast and in mountainous areas, while the more sheltered stations are exposed to considerably weaker wind speeds. The skill of the ECMWF ENS forecasts varies strongly with the locations. In terms of correlation the highest values are seen close to the coast line, while the lowest mostly occur in narrow valleys that are not well resolved in the ECMWF model. At a few stations the observed wind speeds are also strongly influenced by local effects and not very representative of their surrounding areas.

3. Methods

In this section the QFR method based on neural networks is described in detail along with brief descriptions of two methods that are used as benchmarks, the constrained quantile regression splines (CQRS) and the monotonic composite quantile regression neural networks (MCQRNN). At the end, the verification procedure for evaluating the forecast models is outlined.

a. Quantile function regression based on neural networks

In quantile regression the dependency between the observational variable and the ensemble is modeled only for a finite set of quantiles and then most often separately for each quantile. As a consequence, quantiles may at some points cross and be invalid unless precautions are taken in the optimization. The main challenge in quantile regression is to allow for sufficiently flexible relations between the observation and the ensemble and at the same time be able to generate valid quantiles. In this work a slightly different approach is taken. Instead of considering a set of quantiles, all quantiles are modeled simultaneously through a flexible quantile function. Since the quantile function proposed here can be characterized by a few parameters the approach also bears close resemblance to other distributional regression approaches.

Let x denote the vector of covariates—also referred to as predictors, input variables, or features—and $\tau \in [0, 1]$ denote the quantile level. The conditional

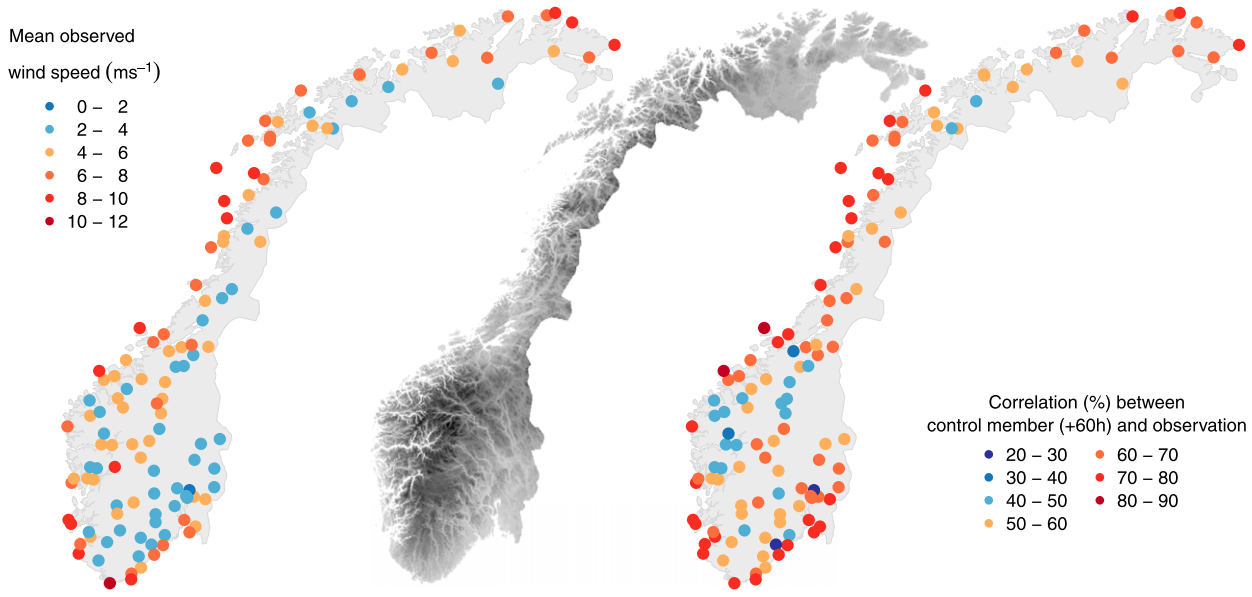

FIG. 1. (left) Mean observed wind speed (m s^{-1}) for each of the 125 sites. (center) Elevation, with the darkest gray level representing about 2500 m above sea level. (right) Linear correlation between the ECMWF ENS control member (%) and the observation for each site for lead time +60 h.

quantile function $Q(\tau|x)$ is then assumed to be a linear combination of some basis functions and expressed by

$$Q(\tau|x) = \sum_{j=0}^{d} \alpha_j(x) B_{jd}(\tau), \tag{1}$$

where the regression coefficients $\alpha_j()$ vary with the covariates while the basis functions only depend on the quantile level τ. The basis functions $B_{jd}(\tau)$ are here chosen to be the Bernstein basis polynomials of degree d, which are defined as

$$B_{jd}(\tau) = \binom{d}{j} \tau^j (1-\tau)^{d-j}, \quad j = 0, 1, \ldots, d, \tag{2}$$

on the interval [0, 1], where $(d\ j)^{-1}$ is the binomial coefficient. The number of basis polynomials is equal to the degree plus one and determines the flexibility of the quantile functions. The Bernstein polynomials have several useful properties. Of particular importance is that if its coefficients are in nondecreasing order, $\alpha_0(x) \le \alpha_1(x) \le \cdots \le \alpha_d(x)$, then the resulting function also will be nondecreasing (e.g., Wang and Ghosh 2012), which is a fundamental requirement for any quantile function. In Fig. 2 the nine Bernstein basis polynomials of degree eight are shown. It can be noticed that they are smooth, positive and bounded. In addition, each basis function has most of its support on a small part of the interval, that is, most of the impact is distributed on a limited range of quantiles. Bernstein polynomials have previously also been used to estimate marginal quantile functions (e.g., Perez and Palacín (1987); Cheng (1995)), as regression functions (e.g., Curtis and Ghosh (2011); Wang and Ghosh (2012); Tan (2016)) and in transformation models [Hothorn et al. 2018; T. Hothorn and A. Zeileis 2017, unpublished manuscript (arXiv: 1701.02110)], among other things.

To estimate the regression coefficients a training dataset is needed. Let $y_1, y_2, \ldots, y_n$ denote the observations and $x_1, x_2, \ldots, x_n$ the covariates, which in our case, are ensemble forecasts. The obvious scoring rule for quantiles is the quantile score (Koenker and Bassett 1978; Gneiting and Raftery 2007), which for implementation convenience is here averaged over a set of quantile

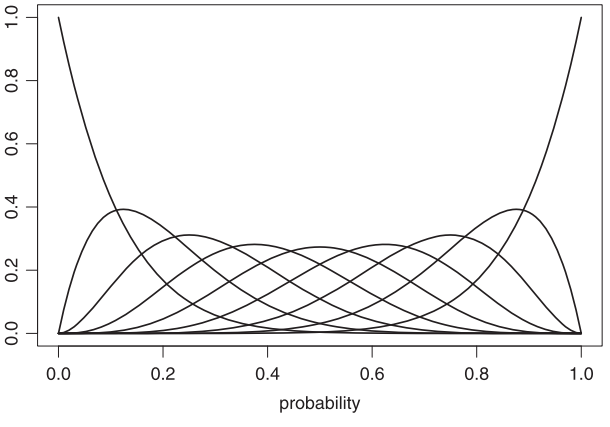

FIG. 2. Bernstein polynomial basis functions of eighth degree.

levels $\tau_1, \tau_2, \ldots, \tau_T$. The coefficients can then be estimated by solving the following optimization problem:

$$\min \sum_{i=1}^{n} \sum_{t=1}^{T} \rho_{\tau_t}(y_i - Q[\tau_t | x_i]) \tag{3}$$

with respect to the underlying coefficients α. The quantile loss function is defined by

$$\rho_\tau(u) = \begin{cases} u(\tau - 1) & u < 0 \\ u\tau & u \geq 0 \end{cases}. \tag{4}$$

Note that it would be possible to replace the sum over the quantile levels by integration over all levels. Since the quantile function is just a linear combinations of polynomials, an explicit expression of the integral is available. The score would then be proportional to the continuous ranked probability score (CRPS; Gneiting and Ranjan 2011). By using many quantile levels, optimizing Eq. (3) would be approximately the same as minimizing the CRPS. In the discrete case the selection of quantile levels can be used to focus on particular parts of the predictive distribution.

So far, it has not been specified how the Bernstein coefficients depend on the covariates. Neural networks are here chosen since they allow for highly flexible relations with the covariates and make it possible to estimate all at the same time. In the case study, the approach is applied to ensemble data at many sites. Instead of making separate models for each site the site identification number is included as an additional covariate. Rasp and Lerch (2018) demonstrated with success the use of so-called embedding to explain variability across sites. The embedding is just a simple mapping of site identification numbers into new variables. Let $s_1, \ldots, s_S$ be binary variables, one for each site, taking value 1 for the respective site and 0 otherwise. Further, let $e_{(1)}, \ldots, e_{(M)}$ denote the ordered ensemble of size M. A neural network with two hidden layers can then be formulated in terms of the following units and layers:

$$\text{input ensemble: } h_j^{(e)} = e_{(j)}, \quad j = 1, \ldots, M,$$

$$\text{site embedding: } h_j^{(s)} = \sum_{k=1}^{S} \omega_{jk}^{(s)} s_k, \quad j = 1, \ldots, N_S,$$

$$\text{1st hidden layer: } h_j^{(1)} = g\left(\omega_{0j}^{(1)} + \sum_{k=1}^{N_S} \omega_{jk}^{(1)} h_k^{(s)} + \sum_{k=N_S+1}^{N_S+M} \omega_{jk}^{(1)} h_{k-N_S}^{(e)}\right), \quad j = 1, \ldots, N_1,$$

$$\text{2nd hidden layer: } h_j^{(2)} = g\left(\omega_{0j}^{(2)} + \sum_{k=1}^{N_1} \omega_{jk}^{(2)} h_k^{(1)}\right), \quad j = 1, \ldots, N_2, \quad \text{and}$$

$$\text{output layer: } \alpha_j = \omega_{0j}^{(3)} + \sum_{k=1}^{N_2} \omega_{jk}^{(3)} h_k^{(2)}, \quad j = 0, \ldots, d,$$

where $g(u) = \max\{0, u\}$, N_S is the size of the embedding, and N_1 and N_2 are the number of units in the first and second hidden layers, respectively. The neural network function for the Bernstein coefficients is also illustrated as a graph in Fig. 3.

The ω parameters, also known as weights, are unknown and must be estimated from training data. In total there are $SN_S + (N_S + M + 1)N_1 + (N_1 + 1)N_2 + (N_2 + 1)(d + 1)$ parameters. In practice, the number of parameters is very high relative to other statistical methods that call for computationally efficient algorithms to estimate them. The Adam algorithm is here chosen because of its well-documented advantages in deep-learning problems (Kingma and Ba 2015). The algorithm is based on gradients and stochastic objective functions, which means that at each iteration the gradients are only computed for a small and random fraction of the n data cases. These small subsets are called *batches* and have a typical size of around 100 cases. To loop through all data cases, called an *epoch*, thus requires many iterations. Because of the nonconvexity of the optimization problem, randomness introduced by the Adam algorithm, and not iterating until convergence the optimal solution will in practice not be found. It is therefore generally recommended to repeat the optimization several times (here 10 times) and average the Bernstein coefficients to get quantile functions for prediction. Further computational details can be found in the appendix.

In addition to estimating weight parameters, there are several tuning parameters that need to be chosen more or less by trial and error. These include the size of the embedding N_S, the number of units N_1 and N_2 in the hidden layers, and also the degree of the Bernstein polynomial d, the batch size, and the number of epochs. In practice predictions based on several choices for these parameters including combinations need to be generated and evaluated. The best configuration is then chosen in the end.

b. Benchmark methods

As benchmarks for the QFR method two quantile regression methods have been chosen. The first, CQRS by Bremnes (2019), assumes that each quantile of interest is represented by a spline function. By putting

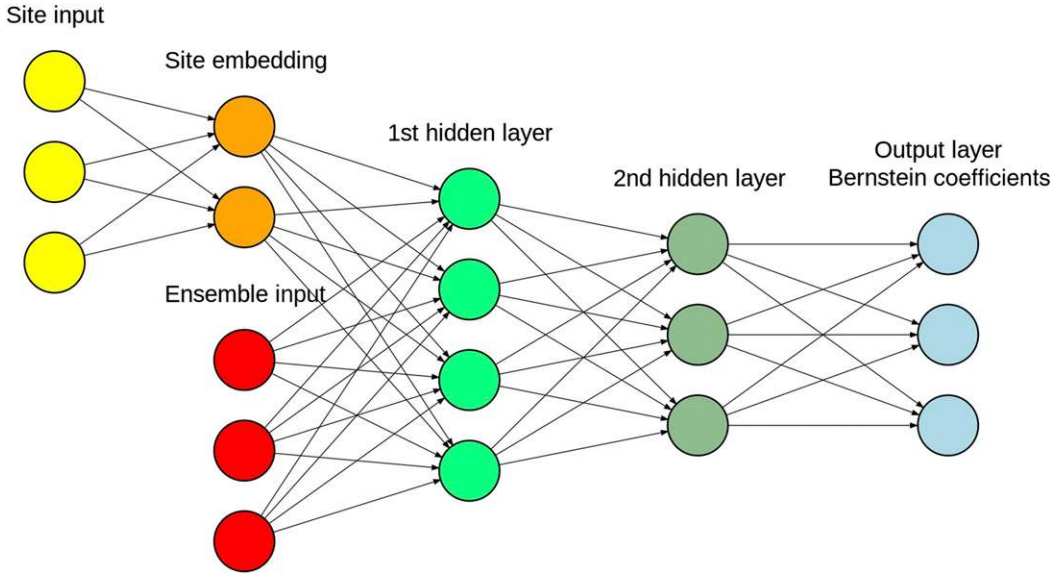

FIG. 3. Illustration of neural network architecture for the Bernstein coefficients: binary representation of S sites (yellow), site embedding of dimension N_S (orange), ensemble member input of size M (red), hidden layers with N_1 and N_2 units (green colors), and output layer with $d + 1$ Bernstein coefficients (light blue). In each of the three middle layers there is also an intercept term/unit.

constraints on the spline coefficients crossing quantiles can be avoided and at the same time nonlinear relations to the ensemble are allowed. In contrast to the QFR, the CQRS is only able to make predictions for the chosen set of quantiles, but any intermediate quantile can be obtained by interpolation if necessary. For the given case study with many sites CQRS is applied separately to each site in order to account for spatial variability across the sites. The CQRS is implemented in the R programming language (the R code is available online at https://github.com/johnbb02/cqrs_mwr/).

The second method, MCQRNN, was proposed by Cannon (2018). In MCQRNN neural networks are applied to describe the relation between a set of quantiles and the covariates/ensemble. To avoid crossing quantiles the quantile level is included as an additional covariate and enforced to be monotonic. For further details see Cannon (2018). Fitting MCQRNN models is here carried out using version 2.0.3 of the R-package qrnn Cannon (2011). Note that, unlike the software for QFR, this version of the qrnn library does not have direct support for embedding of discrete variables like site numbers. It is also much slower.

c. *Forecast verification methods*

The QFR generates forecasts in terms of quantile functions, but in order to compare these with predictions made by the benchmark methods attention is only paid to evaluation at given quantile levels. The reliability and the sharpness of the quantile forecasts are examined in addition to the quantile score (Gneiting and Raftery 2007) and defined as follows. Let y denote the observation and q_τ the τth quantile of the forecast distribution. The quantile score (QS) for the τth quantile is then defined as $\rho_\tau(y - q_\tau)$, which is identical to the loss function used in quantile regression, see Eqs. (3) and (4). For many forecasts the score is averaged over the cases. Depending on the purpose the score is reported for each quantile level and also averaged over the levels. It is worth noting that the quantile score can vary considerably with the level and is often by far largest for central quantiles. Thus, quantiles in the tails of the distribution have low impact on the overall averaged score. It could also be mentioned that by integrating over all levels the quantile score is equal to half the continuous ranked probability score (Gneiting and Ranjan 2011). In the intercomparison of forecast models the quantile scores are not reported as absolute values, but as relative improvements over a chosen reference forecast system (here CQRS). The scores are then referred to as skill scores where the skill score of the reference forecast system is defined to be zero (Murphy 1988).

The reliability of quantile forecasts concerns the degree the probabilistic statement can be trusted. For the τth quantile the probability of observing a value below or equal to the quantile should be τ by definition. The obvious statistic for reliability is therefore the proportion of observations below or equal to the quantile. The closer the proportion is to τ the more reliable the

forecast model is. The deviation between the observed proportion and the quantile level is here applied to assess reliability. Negative deviations thus indicate that predicted quantiles are lower than they should be and the opposite for positive deviations.

Forecast uncertainty should ideally be as low as possible, which implies that predictive probability distributions should be sharp. For quantile forecasts sharpness is usually defined as the average length between two quantiles. Here, average lengths of forecast intervals with coverage probabilities 50%, 88%, and 96% are chosen to quantify sharpness. Since this definition favors unimodal forecast distributions a more general way to compute interval lengths suggested by Bremnes (2019) is also applied. Assume a forecast of M quantiles is given and that these form $M + 1$ subintervals each with probability mass $1/(M + 1)$. The composite interval length is then defined as the total length of the m shortest subintervals such that $m/(M + 1)$ equals the interval coverage probability. For example, to obtain composite intervals with coverage probabilities 26/52 (50%), 46/52 (≈88%), and 50/52 (≈96%) m should be 26, 46, and 50, respectively. The composite interval metric also gives credit to multimodal predictive distribution that in some weather situations may better describe the underlying forecast uncertainty.

The verification scores are averaged over all forecasts and also computed for each site. In addition they are computed for subsets of the forecasts defined by the ECMWF ENS ensemble mean, ensemble standard deviation, ensemble skewness, and the Hartigan's dip test statistic of unimodality (Hartigan and Hartigan 1985) of the ensemble. Various ways of grouping the forecasts have been considered, but results are here reported for ECMWF ENS ensemble statistics discretized into three groups, low, medium, and high, defined by the 10 and 90 percentiles of the ensemble statistics at each station.

4. Experiments and results

In this section QFR is tested and compared against the benchmark methods on the wind speed forecasting data. It could also be mentioned that on exactly the same data CQRS compared favorably in Bremnes (2019) against the mixture of lognormal and truncated normal approach of Baran and Lerch (2016) and the lognormal approach of Baran and Lerch (2015). Hence, QFR is indirectly compared with these methods as well.

a. *Tuning of models*

For the methods based on neural networks, QFR and MCQRNN, there are many options to consider and several parameters that needs to be tuned in order to get best possible predictions. The tuning process was here carried out on the data for 2013 and 2014 (about 14 months), which further was split into training and validation sets. Days 4, 8, 12, . . . , 28 of each month were set aside for validation, and the rest were available for training. In total this resulted in 38 568 cases for training and 11 438 for validation. The CQRS was tested in depth on the same data in Bremnes (2019) and, hence, the best configuration from that study was also adopted here.

For QFR it was decided to use all 51 ensemble members in an ascending order as covariates without further experimentation. After brief explorations an extensive experiment was setup to determine a favorable configuration of the neural network. The following setups were considered:

1) Neural networks with one and two hidden layers were considered. For only one layer, the number of hidden units $N_1 \in \{16, 32, 64\}$ was assessed; for two layers, eight combinations of $N_1 \in \{16, 32, 64\}$ and $N_2 \in \{8, 16, 32, 64\}$ under the restriction $N_1 \geq N_2$ were tested.
2) For embedding size, $N_S \in \{2, 4, 6, 8\}$ was looked at.
3) Batch sizes of 64 and 128 were examined.
4) The degree of Bernstein polynomials $d \in \{4, 6, 8\}$ was considered.

This set of choices led to, in total, 264 QFR configurations that were all trained for 250 epochs. Quantile scores averaged over levels 1/52, 2/52, . . . , 51/52 were then calculated for each combination and epoch, and the best number of epochs was recorded for each combination. It turned out that the quantile scores were remarkably stable over the 264 variants; the best one was only about 1.3% better than the worst in terms of the quantile score. The largest variation was seen across the embedding size N_S. The best score was obtained with $N_1 = 64$, $N_2 = 32$, $N_S = 8$, $d = 8$, a batch size of 128, and 90 epochs. This set resulted, in total, in 7217 weight parameters. From these results, a less extensive study exploring more extreme values for the tuning parameters was performed. However, there were no indications of further improvements. Regularization approaches such as one- and two-norm penalties on the weight parameters and dropout of neural network units were not tried since Rasp and Lerch (2018) did not report any successful results by including them.

In the MCQRNN method there are many tuning parameters as well. Initially all combinations in the following setup were examined:

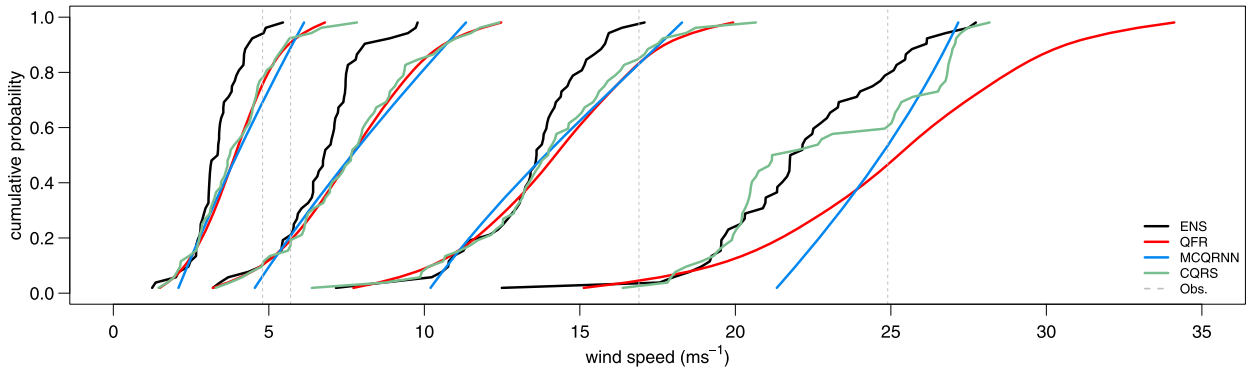

FIG. 4. Four forecast examples for models ECMWF ENS, QFR, MCQRNN, and CQRS given by the cumulative distribution functions (inverse quantile functions) and observed wind speed (dashed vertical lines).

1) loss function with 11 quantile levels: 1/52, 5/52, . . . , 51/52,
2) three definitions of covariates: ensemble mean and ensemble standard deviation; ordered ensemble members $e_{(1)}$, $e_{(26)}$, $e_{(51)}$; and ordered ensemble members $e_{(1)}$, $e_{(13)}$, $e_{(26)}$, $e_{(39)}$, $e_{(51)}$,
3) all covariates assumed to be monotonic, and
4) one hidden layer with 2, 3, 4, and 5 hidden units.

Apart from this, all other tuning parameters were set to default values. Note that because of limitations in the software only simpler neural network models than for QFR could be considered. Separate models for each site also had to be fitted instead of one common model for all sites, as for QFR. It should be kept in mind that these restrictions may have implied that the potentially best MCQRNN model was not possible fit nor test. Conclusions concerning MCQRNN should therefore be drawn with caution. Using the ensemble mean and standard deviation as covariates and two hidden units turned out to give the best quantile score averaged over the same 51 quantile levels as for QFR. Next, the best combinations were tested without the monotonicity constraints, but worse quantile scores were seen. Further, it was verified that the default estimation algorithm gave more or less the same results as the Adam algorithm. Last, increasing the number of quantile levels used in the estimation did not improve the quantile score, and reducing the number gave worse results.

b. *Comparisons with quantile regression methods*

For the intercomparison all models were trained on the data from 2013 and 2014 while year 2015 was left for testing. In total this resulted in 49 883 cases for training and 45 477 for testing. The QFR was applied to all sites in one model, whereas the MCQRNN and the CQRS were trained separately for each site. Predictions were made for the 1/52, 2/52, . . . , 51/52 quantiles. Since the CQRS method has been extensively tested on the same data before, it was chosen as the main reference.

Before looking at verification scores other properties of the forecasting models are assessed. In Fig. 4 examples of four forecasts are shown in terms of their cumulative distribution functions (inverse quantile function). The cases are chosen based on the 50, 85, 99, and 100 percentiles of the ECMWF ENS ensemble mean. While the ECMWF ENS and CQRS give fairly rough and possibly multimodal distributions, QFR generates quite smooth ones. MCQRNN is also smooth, but the tails seem to be very short. For the most extreme case QFR indicates considerably more uncertainty than the other.

In QFR the Bernstein coefficients must be nondecreasing in order to generate valid quantile functions. This constraint was not added to the optimization problem, and, consequently, quantiles may cross. For the 10 individual models that happened in up to 100 (0.22%) cases. However, averaging over the 10 individual models resulted in no incidents. Thus, it can be concluded that it is not vital to include constraints in the estimation. In any case it is possible to rearrange quantiles in the end if necessary.

Wind speeds should obviously be enforced to be nonnegative. For QFR and MCQRNN a lower constraint at zero was not imposed in the estimation. For the individual QFR models this resulted in up to 84 (0.004%) negative quantiles, but again after averaging the Bernstein coefficients none was left. For MCQRNN 168 (0.007%) quantiles were less than zero. These cases were set equal to zero before computing the verification scores. A brief search for unrealistically high quantiles was also carried out. However, there were no suspiciously high quantiles for any of the methods. The highest predicted quantiles were 35.4, 37.2, and 31.1 m s^{-1} for models QFR, MCQRNN, and CQRS,

TABLE 1. Quantile skill scores (%) and proportions of stations with positive QSS (%) for models ECMWF ENS, QFR, and MCQRNN with CQRS as reference. Columns for QSS show the average over all forecasts, and the scores at the worst, median, and best station.

		Scores by station			
	Avg	Worst	Median	Best	QSS ≥ 0
ENS	−49.43	−186.16	−33.27	−1.23	0.00
QFR	0.92	−3.17	0.80	6.56	72.00
MCQRNN	−0.77	−11.71	−0.70	7.72	30.40

respectively, while the maximum observed wind speed in the test set was 30.1 ms^{-1}.

In Table 1 quantile skill scores (QSS) are reported. Clearly, the ECMWF ENS had the worst performance because of local biases and underestimation of forecast uncertainty. The QFR was 0.92% better than the CQRS, while MCQRNN was 0.77% worse. At station level the results vary as expected, but slightly more for the MCQRNN than for the QFR. The number of sites with positive skill scores was also computed. For QFR improvements were seen at 72% of the stations, while for MCQRNN the percentage was only 30%. The ECMWF ENS had negative skill scores for all sites. In Fig. 5 the QSS is computed for each quantile level. The QFR was consistently better than CQRS at all levels and largest for the extreme quantiles, except for the highest quantile level. For the individual QFR models the scores varied quite a bit, which confirms the importance of averaging neural network models and not least for improving the score. The MCQRNN had poor skill for the high and low quantiles, which suggests that further investigations should be made. A possible explanation could be that the neural network model is not flexible enough in combination with the monotonicity requirement. It should also be mentioned that high and low quantiles have only minor impact on the overall quantile score in the optimization and that poor fits for these are not strongly penalized. Figure 6 shows the QSS when the predictions are grouped by whether the ECMWF ENS ensemble mean is low, medium, and high at station level. The variations across the groups are rather small, but relative to CQRS QFR have slightly better scores for the low and high groups, which may indicate that it is advantageous to make one model for all sites instead of separate models. For MCQRNN the scores are worst for the high wind speed forecasts. Grouping by other ensemble statistics were also made, but no obvious variations in the scores were seen.

The reliability of the quantile predictions is first assessed at station level for all quantiles. In Fig. 7 the

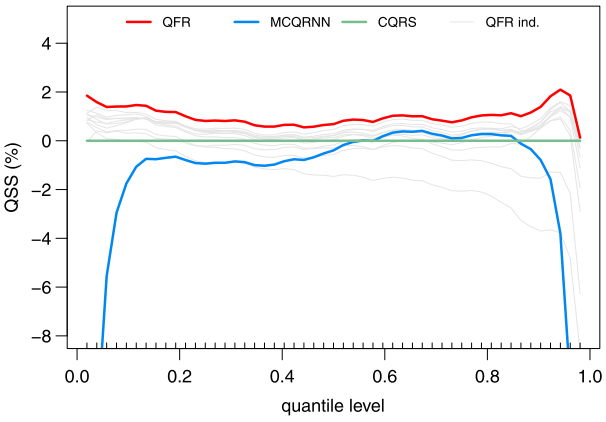

FIG. 5. QSS (%) for models QFR and MCQRNN with CQRS as reference forecast system. Gray lines show the 10 individual QFR models (label QFR ind.). The vertical axis is cut at −8%.

deviation from perfect reliability is computed for each station and its variability over the stations is shown. For both QFR and CQRS the proportions of observations below each quantile level are mostly a little low, which implies that the predicted quantiles are somewhat low on average. The high variability also indicates that the reliability is not good at all sites. For MCQRNN there is a clear systematic variation with the quantile level. The lower quantiles are too high, while the higher quantiles are too low. In other words, the forecast uncertainty is underestimated, which confirms the suspiciously short tails in the forecast examples shown in Fig. 4. For ECMWF ENS (not shown) there were in general far too many observations above and below the extreme quantiles, indicating that the forecast uncertainty is underestimated. In Fig. 8 the average deviation from reliability is shown when the predictions are grouped according to whether the ECMWF ENS ensemble mean is low, medium, and high at site level. For QFR the predicted quantiles are on average mostly a bit high for weak wind speed forecasts and slightly too low for stronger wind speeds. The MCQRNN has the same systematic deviations for all wind speed groups, but with different amplitudes.

Scores for sharpness are reported in Table 2. The ECMWF ENS clearly has the sharpest and narrowest forecast intervals, but it also underestimates uncertainty severely. For the central intervals MCQRNN has the shortest forecast intervals except for the 50% coverage, probably mainly due underestimation of the tails in the predictive distributions. QFR has consistently slightly sharper forecast intervals than CQRS (up to about 9%). For the composite intervals MCQRNN has also the best score except for the 50% intervals where CQRS benefits from more

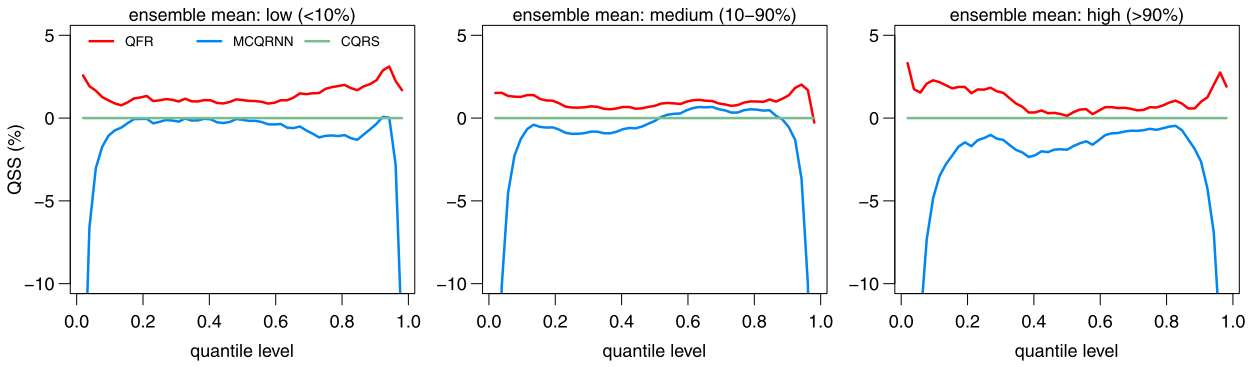

FIG. 6. QSS (%) for models QFR, MCQRNN, and CQRS (reference forecast system) grouped by whether the raw ensemble mean is (left) low (less than the 10th percentile), (center) medium (between the 10th and 90th percentiles), or (right) high (larger than the 90th percentile).

clustered quantiles. QFR generates quite smooth quantile functions.

5. Discussion

The QFR can be categorized as a distributional regression method where the conditional probability distribution is specified by means of the quantile function. Since the quantile function is assumed to be a linear combination of some basis functions, the QFR has some similarity with another neural network method for estimating conditional distributions. In mixture density networks (Jacobs et al. 1991; Bishop 1994) the mixture is represented by a linear combination of Gaussian distributions where the coefficients are assumed to be output from a neural network. Estimates of parameters are obtained by maximizing the likelihood. The difference is thus the choice of basis functions or component distributions and the optimization criterion. The approach of Rasp and Lerch (2018) can be seen as a mixture density network with one component distribution only, but with maximum likelihood estimation replaced by optimizing the continuous ranked probability score. Mixture density networks have even more in common with Bayesian model averaging (BMA) Raftery et al. (2005). The difference is essentially that the neural network in the former allows for more flexible relations to the covariates.

In this study it was decided to use all ensemble members in sorted order as covariates. The main motivation was that the perturbed members can be treated as exchangeable and that each Bernstein basis polynomial has more support for certain quantile levels than others. However, in the case of the ECMWF ENS the unperturbed member has higher skill than a perturbed member and this has been ignored. An option would be to sort the perturbed members and leave the unperturbed as a separate unit in the neural network. In the more general case of multimodel ensembles sorting members could preferably be applied to each submodel.

In operational environments it is vital that postprocessing systems are able to deal with for example

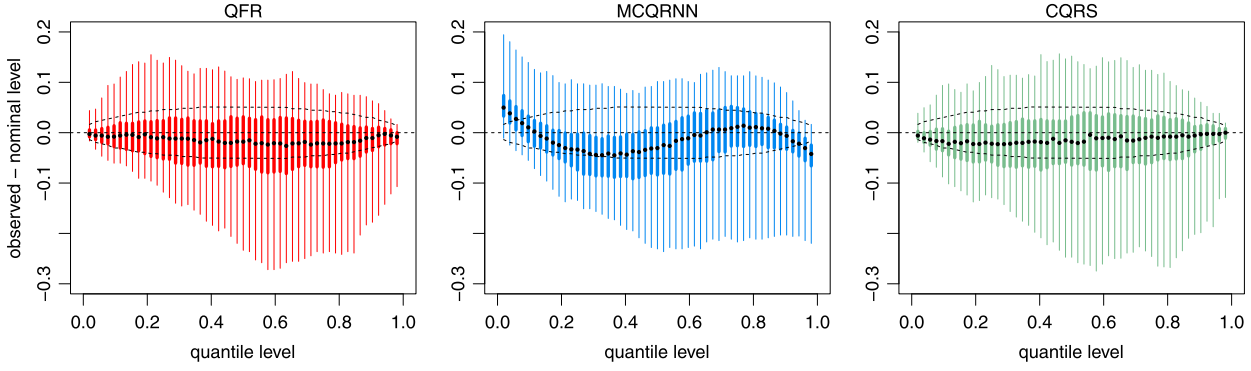

FIG. 7. Deviation in reliability quantified by the difference between observed and nominal level computed for each station. The ranges show the 0, 25, 50, 75, and 100 percentiles for models (left) QFR, (center) MCQRNN, and (right) CQRS. The dashed black lines show the 95% reference interval derived from the binomial distribution with 365 cases.

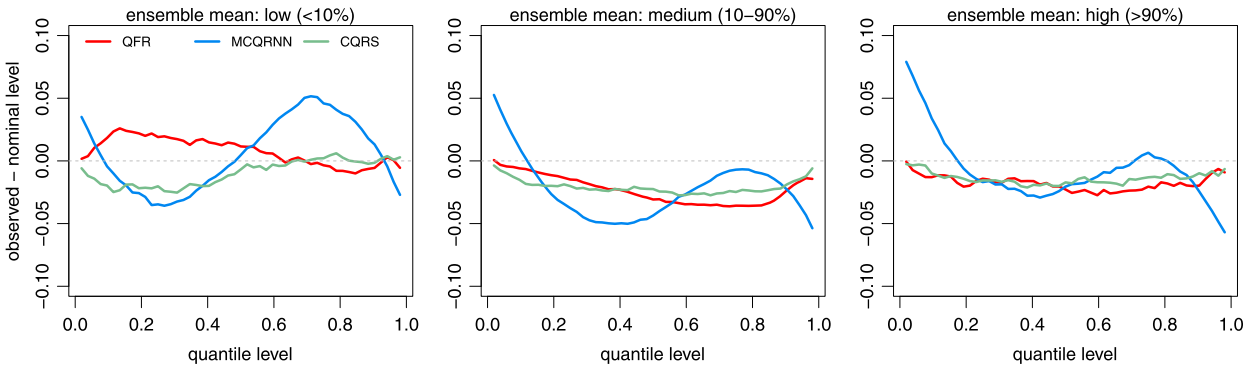

FIG. 8. Deviation in reliability, quantified by the difference between observed and nominal level, averaged over all stations and grouped by whether the raw ensemble mean is (left) low (less than the 10th percentile), (center) medium (between the 10th and 90th percentiles), or (right) high (larger than the 90th percentile). The statistic is shown for models QFR, MCQRNN, and CQRS.

missing ensemble members. There are several alternative approaches to handle such situations. The most simple solution would be to duplicate random members to keep the number of members fixed. Another possibility would be to use interpolation of ordered members to attain the desired ensemble size. If missing members are a persistent problem one may decide in the first place to use less members in the QFR model and thereby allow missing members up to a point without further actions needed in the operational setting. The latter would also reduce the computational burden in the training process. Using less members does not necessarily reduce the forecast quality either. For example, Bremnes (2019) demonstrated that for CQRS slightly improved forecasts skill was obtained by not using all ensemble members. It may also be the case for QFR and could be investigated further.

Including additional NWP model variables in the QFR models has not been looked into here, but can often improve the forecast skill significantly, in particular if long series of training data are available. Rasp and Lerch (2018) included many variables, but only ensemble means and standard deviations. It would be interesting to use all ensemble members also. One possible way could be to form clusters of covariates, one for each ensemble member, and order these with respect to the dominant variable (wind speed in this case). Maybe the ordering of clusters is not necessary. In the case of exchangeable ensemble members it would then be natural to share weight parameters in the neural networks across the clusters.

In the implementation of the QFR no constraints on the quantile functions were put in the optimization. Hence, there is no guarantee that the resulting quantile functions would be valid, that is not cross, or that they are within certain bounds. As is shown, this turned out to be of no concern on the wind speed forecasting data, but it would nevertheless be of interest to include such constraints in general especially if the cost is low. Constraining quantile function to be within a certain range could likely be obtained by applying a suitable function to the mapping between the last hidden layer and the output coefficients. Imposing the Bernstein coefficients to be in nondecreasing order, which is a sufficient condition for valid quantiles, is more challenging. One option would be to reparameterize the Bernstein coefficients such that they instead are represented by sequential differences and force these to be nonnegative by including an appropriate function, for example the exponential, between the last hidden layer and the output.

For many meteorological variables there is no finite upper limit on the values they can take. Quantile functions generated by the QFR will on the other hand always be bounded since the Bernstein basis polynomials are bounded. As consequence, the probability of exceeding a certain value will be 0%. In practice such statements should be avoided and extreme quantiles should therefore be presented with caution. If it is of interest to let quantile functions go to infinity as the probability level approaches one, other basis functions need to be considered. There is, however, not only negative aspects of having a finite unknown upper limit. It may for example prevent highly improbable quantiles.

TABLE 2. Sharpness for models ENS, QFR, MCQRNN, and CQRS in terms of average lengths ($m\,s^{-1}$) of central and composite forecast intervals with coverage probabilities 50%, 88%, and 96%.

	Central intervals			Composite intervals		
	50%	88%	96%	50%	88%	96%
ENS	1.22	2.91	4.00	0.53	2.45	3.90
QFR	2.49	5.80	7.32	2.41	5.63	7.32
MCQRNN	2.81	5.20	5.74	2.28	4.95	5.74
CQRS	2.62	6.01	8.02	1.57	5.40	7.86

In operational settings this would be an attractive feature.

6. Conclusions

In this article a quantile function regression method for making probabilistic forecasts for continuous variables has been proposed. Predictive probability distributions are specified in terms of their quantile function, which is assumed to be a linear combination of Bernstein basis polynomials. The Bernstein coefficients are linked to ensemble forecasts by means of highly adaptable neural networks, and the number of coefficients controls the flexibility of the predictive distributions. In summary, the method provides a very general and robust approach for statistical postprocessing of ensemble forecasts.

The method was tested on ensemble wind speed forecasting data and demonstrated positive skill relative to methods such as constrained quantile regression splines. The improvement of nearly 1% in quantile score may seem small, but one has to keep in mind that there is a finite amount of information in the data and that the scope for further improvements on these data may be small. It can be argued that the most attractive features with the QFR are its flexibility and potential for extensions in various directions. For example, in contrast to the CQRS it seems straightforward to include more covariates. Future studies include applying the method to other variables like precipitation and to problems where predictions are needed at any spatial location and not only at points with measurements.

Acknowledgments. The author thanks the three reviewers, Thorsten Simon plus two anonymous ones, for their comments on the paper.

APPENDIX

Further Computational Details

The QFR method has been implemented in R using the Keras interface (Chollet et al. 2015) to Google's TensorFlow library (Abadi et al. 2015), a highly versatile software library for neural network models among others. The library enables users to design new loss functions, has good support for embedding of discrete variables like site identification numbers, and provides in general an easy interface to building a wide range of neural network models. In this study the Adam algorithm (Kingma and Ba 2015) has been employed for parameter estimation with the following arguments: 1) learning rate of 0.001, 2) exponential decay rate for the first moment estimates of 0.9, 3) exponential decay rate for the second moment estimates of 0.999, and 4) learning rate decay of 0. All arguments are recommended default values. For further information it is referred to the given reference.

Concerning computing time for training the implementation of CQRS was by far the fastest. On a single core/thread on an Intel Core i7–8700K 3.7-GHz CPU it took 13 s to train the model in section 4b. The QFR with 10 individual fits and the MCQRNN were roughly 50 and 150 times slower, respectively. For all methods it is possible to parallelize the computations at various levels to speed up the training process considerably.

REFERENCES

Abadi, M., and Coauthors, 2015: TensorFlow: Large-scale machine learning on heterogeneous systems. Google, Inc., accessed 1 July 2019, https://www.tensorflow.org/.

Baran, S., and S. Lerch, 2015: Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quart. J. Roy. Meteor. Soc.*, **141**, 2289–2299, https://doi.org/10.1002/qj.2521.

——, and ——, 2016: Mixture EMOS model for calibrating ensemble forecasts of wind speed. *Environmetrics*, **27**, 116–130, https://doi.org/10.1002/env.2380.

Bishop, C. M., 1994: Mixture density networks. Aston University Neural Computing Research Group Rep. NCRG/94/004, 26 pp., https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf.

Bremnes, J. B., 2004: Probabilistic forecasts of precipitation in terms of quantiles using NWP model output. *Mon. Wea. Rev.*, **132**, 338–347, https://doi.org/10.1175/1520-0493(2004)132<0338:PFOPIT>2.0.CO;2.

——, 2019: Constrained quantile regression splines for ensemble postprocessing. *Mon. Wea. Rev.*, **147**, 1769–1780, https://doi.org/10.1175/MWR-D-18-0420.1.

Cannon, A. J., 2011: Quantile regression neural networks: Implementation in R and application to precipitation downscaling. *Comput. Geosci.*, **37**, 1277–1284, https://doi.org/10.1016/j.cageo.2010.07.005.

——, 2018: Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes. *Stochastic Environ. Res. Risk Assess.*, **32**, 3207–3225, https://doi.org/10.1007/s00477-018-1573-6.

Cheng, C., 1995: The Bernstein polynomial estimator of a smooth quantile function. *Stat. Probab. Lett.*, **24**, 321–330, https://doi.org/10.1016/0167-7152(94)00190-J.

Chollet, F., and Coauthors, 2015: Keras documentation, accessed 1 July 2019, https://keras.io.

Curtis, S. M., and S. K. Ghosh, 2011: A variable selection approach to monotonic regression with Bernstein polynomials. *J. Appl. Stat.*, **38**, 961–976, https://doi.org/10.1080/02664761003692423.

Gneiting, T., and A. E. Raftery, 2007: Strictly proper scoring rules, prediction, and estimation. *J. Amer. Stat. Assoc.*, **102**, 359–378, https://doi.org/10.1198/016214506000001437.

——, and R. Ranjan, 2011: Comparing density forecasts using threshold- and quantile-weighted scoring rules. *J. Bus. Econ. Stat.*, **29**, 411–422, https://doi.org/10.1198/jbes.2010.08110.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, 800 pp.

Hartigan, J. A., and P. M. Hartigan, 1985: The dip test of unimodality. *Ann. Stat.*, **13**, 70–84, https://doi.org/10.1214/aos/1176346577.

Hothorn, T., L. Möst, and P. Bühlmann, 2018: Most likely transformations. *Scand. J. Stat.*, **45**, 110–134, https://doi.org/10.1111/sjos.12291.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton, 1991: Adaptive mixtures of local experts. *Neural Comput.*, **3**, 79–87, https://doi.org/10.1162/neco.1991.3.1.79.

Kingma, D. P., and J. Ba, 2015: Adam: A method for stochastic optimization. *Proc. Third Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, ISCA, https://dblp.org/db/conf/iclr/iclr2015.

Koenker, R., and J. Bassett, 1978: Regression quantiles. *Econometrica*, **46**, 33–50, https://doi.org/10.2307/1913643.

Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliagis, 1996: The ECMWF ensemble prediction system: Methodology and validation. *Quart. J. Roy. Meteor. Soc.*, **122**, 73–119, https://doi.org/10.1002/qj.49712252905.

Murphy, A. H., 1988: Skill scores based on the mean square error and their relationships to the correlation coefficient. *Mon. Wea. Rev.*, **116**, 2417–2424, https://doi.org/10.1175/1520-0493(1988)116<2417:SSBOTM>2.0.CO;2.

Perez, J. M., and A. F. Palacín, 1987: Estimating the quantile function by Bernstein polynomials. *Comput. Stat. Data Anal.*, **5**, 391–397, https://doi.org/10.1016/0167-9473(87)90061-2.

Raftery, A. E., T. Gneiting, F. Balabdaoui, and M. Polakowski, 2005: Using Bayesian model averaging to calibrate forecast ensembles. *Mon. Wea. Rev.*, **133**, 1155–1174, https://doi.org/10.1175/MWR2906.1.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, https://doi.org/10.1175/MWR-D-18-0187.1.

Richardson, D. S., 2000: Skill and relative economic value of the ECMWF ensemble prediction system. *Quart. J. Roy. Meteor. Soc.*, **126**, 649–667, https://doi.org/10.1002/qj.49712656313.

Scheuerer, M., and T. M. Hamill, 2015: Statistical postprocessing of ensemble precipitation forecasts by fitting censored, shifted Gamma distributions. *Mon. Wea. Rev.*, **143**, 4578–4596, https://doi.org/10.1175/MWR-D-15-0061.1.

Taillardat, M., O. Mestre, M. Zamo, and P. Naveau, 2016: Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Mon. Wea. Rev.*, **144**, 2375–2393, https://doi.org/10.1175/MWR-D-15-0260.1.

Tan, M. H. Y., 2016: Monotonic quantile regression with Bernstein polynomials for stochastic simulation. *Technometrics*, **58**, 180–190, https://doi.org/10.1080/00401706.2015.1027066.

Toth, Z., and E. Kalnay, 1993: Ensemble forecasting at NMC—The generation of perturbations. *Bull. Amer. Meteor. Soc.*, **74**, 2317–2330, https://doi.org/10.1175/1520-0477(1993)074<2317:EFANTG>2.0.CO;2.

Vannitsem, S., D. Wilks, and J. Messner, Eds., 2018: *Statistical Postprocessing of Ensemble Forecasts*. 1st ed. Elsevier, 362 pp.

Wang, J., and S. Ghosh, 2012: Shape restricted nonparametric regression with Bernstein polynomials. *Comput. Stat. Data Anal.*, **56**, 2729–2741, https://doi.org/10.1016/j.csda.2012.02.018.