Nonhomogeneous Boosting for Predictor Selection in Ensemble Postprocessing

JAKOB W. MESSNER,a GEORG J. MAYR, AND ACHIM ZEILEIS
University of Innsbruck, Innsbruck, Austria

(Manuscript received 4 March 2016, in final form 9 September 2016)

ABSTRACT

Nonhomogeneous regression is often used to statistically postprocess ensemble forecasts. Usually only ensemble forecasts of the predictand variable are used as input, but other potentially useful information sources are ignored. Although it is straightforward to add further input variables, overfitting can easily deteriorate the forecast performance for increasing numbers of input variables. This paper proposes a boosting algorithm to estimate the regression coefficients, while automatically selecting the most relevant input variables by restricting the coefficients of less important variables to zero. A case study with ensemble forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF) shows that this approach effectively selects important input variables to clearly improve minimum and maximum temperature predictions at five central European stations.

1. Introduction

Over the past decades ensemble forecasts have become an important tool for estimating the uncertainty of numerical weather prediction models. To account for initial condition and model errors, numerical models are integrated several times with slightly different initial conditions and sometimes different parameterization schemes. However, because of insufficient representation of these errors such ensembles of predictions are often biased and do not fully represent the forecast uncertainty. Therefore, ensemble forecasts are often statistically postprocessed to obtain unbiased and calibrated probabilistic forecasts.

Over the past years a variety of different ensemble postprocessing methods have been proposed. Aside from ensemble dressing (Roulston and Smith 2003), Bayesian model averaging (Raftery et al. 2005), or (extended) logistic regression (Hamill et al. 2004; Wilks 2009; Messner et al. 2014b), nonhomogeneous regression (Gneiting et al. 2005) is particularly popular. It assumes a parametric predictive distribution and models the distribution parameters as linear functions of predictor variables such as the ensemble mean and ensemble standard deviation. In recent years it has been used for several different forecast variables (e.g., Thorarinsdottir and Gneiting 2010; Scheuerer 2014; Scheuerer and Hamill 2015) and has been extended to account for covariance structures (Pinson 2012; Schuhen et al. 2012; Schefzik et al. 2013; Feldmann et al. 2015) or to predict full spatial fields (Scheuerer and Büermann 2014; Feldmann et al. 2015; Dabernig et al. 2016; Stauffer et al. 2017). In most publications only the ensemble forecast of the predictand variable was used as input for the nonhomogeneous regression model. However, Scheuerer (2014) and Scheuerer and Hamill (2015) showed that additional input variables can be easily incorporated and can clearly improve the forecast performance. The set of potentially useful input variables is huge and includes, among others, ensemble forecasts for other variables or locations, deterministic forecasts, current observations, transformations, and interactions of all of these. Since using too many input variables can deteriorate the forecast accuracy through overfitting, the input variables should be selected carefully. Doing this by hand can be a cumbersome task that requires expert knowledge and should be done separately for each forecast variable, station, and lead time.

DOI: 10.1175/MWR-D-16-0088.1

© 2017 American Meteorological Society
For postprocessing of deterministic predictions, stepwise regression has commonly been used to automatically select the most important input variables (e.g., Glahn and Lowry 1972; Wilson and Vallé 2002). However, to our knowledge, automatic variable selection has not yet been used for ensemble postprocessing with nonhomogeneous regression. In this paper we propose a boosting algorithm to automatically select the most relevant predictor variables in nonhomogeneous regression. Boosting has originally been proposed for classification problems (Freund and Schapire 1997) but has also been extended and used for regression (Friedman et al. 2000; Bühlmann and Yu 2003; Bühlmann and Hothorn 2007; Hastie et al. 2013). Like other optimization algorithms boosting finds the minimum of the loss function iteratively, but in each step it only updates the coefficient that improves the current fit most. Thus, if it is stopped before convergence, only the most important predictor variables have nonzero coefficients so that less relevant variables are ignored. To investigate this novel boosting approach and to compare its performance against ordinary nonhomogeneous regression we use maximum and minimum temperature forecasts at five stations in central Europe. As potential input variables we use ensemble forecasts for different weather variables from the European Centre for Medium-Range Weather Forecasts (ECMWF).

The remainder of this paper is structured as follows: the following section describes the nonhomogeneous regression approach and introduces the boosting algorithm to estimate the regression coefficients. Subsequently section 3 describes the data that are used to compute the results that are presented in section 4. Finally, section 5 provides a summary and conclusions.

2. Methods

This section first describes the nonhomogeneous regression approach of Gneiting et al. (2005) and subsequently presents a boosting algorithm to automatically select the most relevant input variables.

a. Nonhomogeneous regression

Nonhomogeneous regression, sometimes also called ensemble model output statistics, was first proposed by Gneiting et al. (2005) for normally distributed predictands such as temperature and sea level pressure. Later publications extended this method to variables described by nonnormal distributions, for example, wind [truncated normal; Thorarinsdottir and Gneiting (2010)], or precipitation [generalized extreme value; Scheuerer (2014), censored logistic; Messner et al. (2014a), or censored gamma; Scheuerer and Hamill (2015)]. In the following, we only regard nonhomogeneous Gaussian regression (NGR), but most concepts can easily be transferred to other distributions as well.

NGR assumes the observations $y$ to follow a normal distribution $\mathcal{N}$ with mean $\mu$ (location) and variance $\sigma^2$ (squared scale):

$$y \sim \mathcal{N}(\mu, \sigma^2),$$  \hspace{1cm} (1)

where the location $\mu$ and the logarithm of the scale $\sigma$ are expressed as

location: \quad \mu = \mathbf{x}^T \mathbf{\beta}, \hspace{1cm} (2)

log-scale: \quad \log(\sigma) = \mathbf{z}^T \mathbf{\gamma}, \hspace{1cm} (3)

with $\mathbf{x} = (1, x_1, x_2, \ldots)^T$ and $\mathbf{z} = (1, z_1, z_2, \ldots)^T$ being vectors of predictor variables, and $\mathbf{\beta} = (\beta_0, \beta_1, \beta_2, \ldots)^T$ and $\mathbf{\gamma} = (\gamma_0, \gamma_1, \gamma_2, \ldots)^T$ the corresponding coefficient vectors. Note that $y$, $\mathbf{x}$, $\mathbf{z}$, $\mu$, and $\sigma$ are event specific (i.e., different for each forecast event) but indices were omitted to enhance the readability. The logarithmic link function in Eq. (3) $[\log(\sigma)]$ is used to assure positive values for $\sigma$. Alternatively, often also $\sigma^2$ is modeled where all coefficients in $\mathbf{\gamma}$ are restricted to be positive (e.g., Gneiting et al. 2005).

The coefficients $\mathbf{\beta}$ and $\mathbf{\gamma}$ are estimated by minimizing a loss function such as the negative log-likelihood or the continuous ranked probability score (CRPS). In the following, we use the negative log-likelihood, but all concepts can be easily transferred to any other differentiable loss function as well. The negative log-likelihood ($L$) for a single event is given by

$$L(\mu, \sigma) = -\log \left[ \frac{1}{\sigma} \phi \left( \frac{y - \mu}{\sigma} \right) \right],$$  \hspace{1cm} (4)

where $\phi(\cdot)$ is the probability density function of the normal distribution. The full negative log-likelihood, that is used to estimate $\mathbf{\beta}$ and $\mathbf{\gamma}$, is derived by taking the sum of $L(\mu, \sigma)$ over the training data. We perform this optimization with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm as implemented in R (R Core Team 2015), similar to Gneiting et al. (2005), Thorarinsdottir and Gneiting (2010), and Scheuerer (2014). For an increased efficiency of this optimization we also use analytical gradients and Hessian matrices of the log-likelihood (Messner et al. 2016). In most studies, $\mathbf{x}$ is a vector including different ensemble member forecasts or the ensemble mean forecast while $\mathbf{z}$ usually contains the ensemble variance or standard deviation. Scheuerer (2014) and Scheuerer and Hamill (2015) also
included further input variables; however, typically only
ensemble forecasts of the predictand variable have been
used (e.g., only ensemble predictions of temperature are
included in x and z for temperature forecasts).

Clearly, many more information sources could be
used as inputs (e.g., different ensemble forecast
variables, current observations, deterministic forec-
casts, or transformations or interactions of all of
these). However, adding too many variables can
easily result in overfitting so that the input variables
must be selected carefully. Considering the huge set
of candidate variables it is clear that selecting them
by hand can be very cumbersome, especially if forec-
sasts for many different predictands, stations, and
lead times are required.

Thus, algorithms to automatically select the most
important variables are highly desirable. Section 2b
introduces a boosting algorithm that can be employed
for this purpose.

b. Nonhomogeneous boosting

This subsection introduces an alternative algorithm to
the BFGS optimization to estimate the coefficients β
and γ. This algorithm is based on boosting and can au-
tomatically select the most important predictor vari-
ables. Like other optimization algorithms, boosting finds
the minimum of the loss function [e.g., the negative log-
likelihood sum over all events]. This algorithm is based on boosting and can au-
tomatically select the most important variables.

The θs are vectors of zeros, mstop is a predefined
number of boosting iterations, ρ(xj, r) is the correla-
tion calculated by averaging xj × r over the training data
(note that r and s are also different for each forecast
event), and ν is a predefined step size between 0 and 1.

Bühlmann and Hothorn (2007) showed that the choice
of the step size is only of minor importance as long as it is
small and we follow their suggestion of ν = 0.1.

Because xj and zk have zero mean and unit variance,
ρ(xj, r) and ρ(zk, s) are simple linear regression co-
eficients of r given xj and s given zk, respectively. Thus,
ρ(xj, r) and ρ(zk, s) can be viewed as linear approxi-
mations of the negative gradients and the update in
steps 2ii and 2iv proceeds along the steepest of these
approximated negative gradients (Bühlmann and Hothorn 2007).

If mstop is selected to be very large, the estimated
coefficients β and γ approximate the maximum likeli-
hood estimates from the model in the previous sub-
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
However, overfitting is prevented with unimportant
variables having zero coefficients so that the predictive
performance might be improved. To get the best pre-
dictive performance an appropriate mstop has to be
found. This is achieved by optimizing the cross-validated
log-likelihood estimates from the model in the previous
section. When choosing a smaller mstop the likelihood
for the training data has not reached its maximum.
In addition to automatically selecting the most important input variables, boosting also regularizes the nonzero coefficients (i.e., the coefficients are shrunk compared to their maximum-likelihood values). Hastie et al. (2013) showed that this regularization is similar to that of the absolute shrinkage and selection operator (LASSO; Tibshirani 1994) and also helps to reduce overfitting, especially for highly correlated input variables.

In the following we investigate nonhomogeneous Gaussian boosting (NGB) in a case study and compare its performance with that of NGR with only the ensemble forecast of the predictand variable as input. To assess the influence of the regularization in boosting, we also compare a further NGR model with the subset of input variables that were selected by boosting.

3. Data

This section describes the data that are used for the case study in the following section. We considered minimum and maximum temperatures at the five central European SYNOP stations: Wien Schwechat (48.110°N, 16.570°E), Innsbruck Airport (47.260°N, 11.357°E), Berlin Tegel (52.566°N, 13.311°E), Leipzig Halle (51.436°N, 12.241°E), and Zürich Kloten (47.480°N, 8.536°E). Minimum temperatures are for periods between 1800 and 0600 UTC, and maximum temperatures between 0600 and 1800 UTC.

As numerical predictions we employed the 51-member ensemble predictions from the ECMWF. In addition to the direct forecast of minimum and maximum temperatures, we used various predictions for different parameters (e.g., temperatures, wind, precipitation) from the surface level and pressure levels at 1000, 850, 700, and 500 hPa. The regarded 12-h time windows (1800–0600 or 0600–1800 UTC) span several (3 h) time steps of the ECMWF model. For accumulated quantities (e.g., precipitation) we computed means, maxima, and minima over the regarded time windows for each parameter and member, respectively.

Subsequently ensemble means and log standard deviations were derived. The logarithm of the ensemble standard deviations is used to be consistent with the log scale that is modeled in Eq. (3). Zero standard deviations sometimes occur for variables with a limited range such as precipitation. These variables are almost never selected by our models, but to avoid infinite numbers, we set standard deviations that are 0 to a very small value (0.0001).

For each accumulated parameter, this results in two variables (ensemble means and log standard deviation) and for each other parameter in six variables (ensemble means and log standard deviations for 12-hourly means, minima, and maxima). In the following, these variables are labeled according to following rule: parameter_aggregation_ensemble-statistic, [e.g., t2m_dmax_mean is the ensemble mean of daily (12 hourly) maximum temperature ensemble forecasts at 2m above ground].

In addition to the ensemble predictions from the numerical weather forecasting model, the last observed minimum or maximum temperature is used as potential predictor variable. Overall 335 input variables are available to the NGB model.

We regarded lead times from 1 to 5 days (30–138 h) and use data from January 2011 to January 2016 (approximately 1700 days).

Clearly, many variables such as temperatures, have strongly pronounced seasonal patterns that probably affect the statistical properties of forecasts and observations. To only use training data that are representative for the current season, many studies use moving training windows of a certain number of days preceding the forecast date (e.g., Gneiting et al. 2005; Thorarinsdottir and Gneiting 2010; Scheuerer and Bürmann 2014). While this approach allows the model to adapt quickly to seasonal changes, it disregards large parts of available data. As an attractive alternative approach, Dabernig et al. (2016) showed that standardized anomalies can be used to remove seasonal patterns and allow for the use of substantially larger training datasets. For these standardized anomalies, first seasonally varying climatological means (location) and standard deviations (scale) were derived for the predictand and all input variables. Therefore, a nonhomogeneous regression model, similar to Eqs. (1)–(3), was fitted:

\[ a \sim \mathcal{N}(\mu_a, \sigma_a^2), \quad (11) \]

(location):

\[ m_a = \beta_0 + \beta_1 \sin(2\pi d/365) + \beta_2 \cos(2\pi d/365), \quad (12) \]

log-scale:

\[ \log(s_a) = \gamma_0 + \gamma_1 \sin(2\pi d/365) + \gamma_2 \cos(2\pi d/365), \quad (13) \]

where \( a \) is the respective variable (predictand or input variable) and \( d \) is the day of the year. Standardized anomalies are then derived by

\[ a = \frac{m_a}{s_a}. \quad (14) \]

As an example, Fig. 1 shows that the standardized anomalies of maximum temperatures in Wien Schwechat have no pronounced seasonal cycle anymore, neither in the mean nor in the variance.
Note that when anomalies are employed, location predictions $\hat{\mu}$ have to be transformed back by
\[ m_y + \hat{\mu} s_y \tag{15} \]
and scale predictions $\hat{\sigma}$ have to be transformed by
\[ \hat{\sigma} s_y. \tag{16} \]
Standardized anomalies work well for symmetrically distributed data such as temperatures, but could be less appropriate for variables like precipitation or wind speed. These may require more advanced approaches.

![Figure 1](image1.png)

**FIG. 1.** (left) Observed maximum temperatures (gray circles) in Wien Schwechat, fitted climatological mean (solid line), and climatological mean ± 1 climatological standard deviation (dashed lines). (right) Corresponding standardized anomalies derived by subtracting the mean and dividing by the standard deviation.

![Figure 2](image2.png)

**FIG. 2.** Paths of boosting coefficients for a +66-h maximum temperature forecast at Wien Schwechat. Coefficient paths for the location are shown as black lines and for the log scale they are shown as red lines. The optimum stopping iteration according to cross validation (cv) is shown as the dashed vertical line. The most important coefficients are labeled (see text).
4. Results

This section assesses the boosting algorithm on the data described in the previous section. To illustrate the boosting optimization, Fig. 2 shows a typical evolution of coefficients. Since the input variables all have unit variance their coefficient values can be directly compared and indicate their relevance. After all coefficients being zero in the beginning, the daily mean maximum temperature ensemble mean (tmax2m_dmean_mean) is the first variable that gets a nonzero coefficient, which indicates that it explains the observations best. With an increasing value of the corresponding coefficient, more and more of the variance in the observations is explained so that the intercept for the log scale decreases. After approximately 20 iterations the ensemble standard deviation of daily maximum evaporation (ske_dmax_sd) enters with a negative coefficient for the log scale.

A few steps later the daily maximum 2-m temperature ensemble mean (t2m_dmax_mean) is added to the equation for the location. Further selected variables are the daily minimum soil temperature ensemble mean (stl1_dmin_mean), the 700-hPa daily mean vorticity ensemble standard deviation (not labeled) for the log scale, and the daily minimum 1000-hPa temperature ensemble mean (not labeled) for the location. Further variables enter the regression equations later, but are not considered because the optimum cross-validation stopping iteration is already found at 31.

Figure 3 shows the boosting coefficients from the cross-validation stopping iteration at different lead times for maximum temperature forecasts in Wien Schwechat. Additionally, dashed lines show the NGR coefficients. As already indicated in Fig. 2, the daily maximum temperature ensemble forecast, which would be the direct predictor, is neither important for the location nor for the log scale. However, it is highly correlated (correlation coefficients > 0.9) to the daily mean maximum temperature, the daily maximum 2-m
temperature, or temperatures at 1000 hPa, so that these variables are virtually exchangeable without losing much information. For the log scale (Fig. 3, bottom), ensemble standard deviations of various variables are selected but also ensemble mean forecasts (e.g., of 1000-hPa divergence d1000_dmax_mean) seem to contain forecast uncertainty information. Interestingly, the NGR coefficient of the ensemble standard deviation in the scale equation is negative for short lead times indicating a negative spread–skill relationship (Wilks 2011).

Figure 4 shows coefficients similar to Fig. 3 but for minimum temperatures. The direct predictor, the daily minimum temperature ensemble mean, is clearly the most relevant variable over all lead times unlike for maximum temperatures. However, various other variables seem to be more relevant for the log-scale equation, many also with negative coefficients. Note that for Wien Schwechat (Figs. 3 and 4), boosting selects relatively few variables. Many more variables are selected for some of the other stations (not shown).

Figures 2–4 show that boosting selects a meteorologically reasonable set of variables. In the following, we investigate how the increased number of input variables improves the forecast performance. To obtain independent training and test data, 10-fold cross validation is used again: for each station and lead time the data is split into 10 parts and for each part performance measures [squared errors, CRPS, or probability integral transforms (PITs)] are computed for models that were trained on the 9 remaining parts. The effective training data length is thus 9/10 of the full dataset length (approximately 1550 days). To estimate the sampling distribution of average squared errors and CRPS we computed means of 250 bootstrap samples.

Figure 5 shows the root-mean-squared error (RMSE) of the location forecasts \( \mu \) in Eq. (2)] of NGB, NGR, and the subset NGR, which is an NGR with the non-zero coefficients from boosting as input. For the two stations—Wien Schwechat and Innsbruck Airport—the RMSE of the minimum temperature forecast is always smaller for boosting than for NGR. As already indicated in Fig. 4, NGR and boosting differ only slightly for Wien minimum temperature forecasts. In contrast the differences are much larger for Innsbruck. In addition to selecting the most important variables, boosting also regularizes or shrinks the coefficients. The subset model
uses the same variables as boosting but does not regularize their coefficients, which results in very similar RMSE. The RMSE of the other stations and maximum temperatures look very similar to that of Wien Schwechat and Innsbruck Airport minimum temperatures and are therefore not shown.

While the RMSE shows the deterministic performance, we employ the CRPS (Hersbach 2000) to measure the probabilistic quality of the forecasts. Gneiting et al. (2005) provide a closed form for normal predictive distributions:

\[
\text{CRPS} = \sigma \left\{ \frac{y - \mu}{\sigma} \left[ 2\Phi \left( \frac{y - \mu}{\sigma} \right) - 1 \right] + 2\phi \left( \frac{y - \mu}{\sigma} \right) - \frac{1}{\sqrt{\pi}} \right\},
\]

where \( \Phi(\cdot) \) and \( \phi(\cdot) \) are the normal cumulative distribution function and probability density function, respectively; \( y \) is the observation; and \( \mu \) and \( \sigma \) are the predicted location and scale, respectively. Since we are mainly interested in improvements of boosting over NGR, Fig. 6 shows the continuous ranked probability skill score (CRPSS) relative to NGR:

\[
\text{CRPSS} = 1 - \frac{\text{CRPS}}{\text{CRPS}_{\text{NGR}}},
\]

where \( \text{CRPS} \) is the respective average CRPS and \( \text{CRPS}_{\text{NGR}} \) is the average CRPS of NGR. The values of the five stations are aggregated to summarize the overall performance of the different methods. For both minimum and maximum temperature forecasts, NGB performs clearly better than NGR over all lead times where for longer lead times this advantage is less pronounced. Contrary to the RMSE, the regularization in boosting slightly improves the forecast performance compared to the subset model.

To assess the reliability of the forecasts, Fig. 7 shows PIT histograms (Wilks 2011) of NGB and NGR. Both forecast methods seem to produce predictive distributions with too light left and too heavy right tails, indicating that actually a nonsymmetric distribution would better fit the data. However, the flatter PIT histogram of NGB indicates that using more variables partly compensates for this problem and increases the reliability.

Finally, Fig. 8 shows the CRPSS for different training data lengths. For shorter training data lengths the number of selected input variables decreases but is still proportionally high compared to the training data length. In the subset model this leads to overfitting that clearly deteriorates the predictive performance. In contrast, NGB regularizes the coefficients to largely prevent
overfitting so that, except for very short training data lengths, it outperforms NGR (i.e., positive CRPSS).

5. Summary and conclusions

Nonhomogeneous regression can easily be extended to use further predictor variables in addition to ensemble forecasts of the predictand variable. However, to avoid overfitting that can deteriorate the predictive performance, predictor variables have to be selected carefully.

In this paper we presented a boosting algorithm to estimate the regression coefficients that can be used for automatic variable selection. In addition to variable selection, this algorithm also regularizes or shrinks the regression coefficients to further prevent overfitting.
A case study for minimum and maximum temperatures at five central European stations showed clear improvements in the predictive performance compared to a nonhomogeneous regression model with only ensemble mean and standard deviation of the predictand variable as input. The regularization of boosting showed to have only a positive effect for short training data lengths.

In our case study we employed a large set of different ensemble predictions from ECMWF (approximately 100) at surface and several pressure levels. We aggregated these predictions over the regarded time windows and computed ensemble means and log standard deviations. Additionally, we also used the last available observations as potential predictor variables. Clearly there are many more potential input variables that we have not included [e.g., current observations of other variables or from neighboring weather stations, deterministic predictions or ensemble predictions from other centers, transformations of all of these variables (e.g., logarithm, roots, or powers), etc.]. Including some of these would probably further improve the forecasts.

In this paper, we assumed minimum and maximum temperatures to follow normal distributions. However, the PIT histograms indicate that the conditional distribution of maximum and minimum temperatures given the ensemble forecast is not perfectly symmetric so that using a different asymmetric distribution could improve the forecast performance. Other distributions might also be required for predictions of other nonnormally distributed variables such as precipitation or wind speed. Although we presented boosting for normally distributed predictive distributions, most concepts can easily be transferred to other distributions as well. Similarly, also other differentiable loss functions, such as the CRPS, could be employed instead of the negative log-likelihood.

Variable selection is clearly not new in the statistical postprocessing literature. Glahn and Lowry (1972) already recognized the importance of variable selection for deterministic model output statistics and proposed the use of stepwise selection. However, except Bröcker (2010) who proposed lasso regularization for logistic regression and Wahl (2015) who used lasso penalization for quantile regression, automatic variable selection has rarely been used in the ensemble postprocessing literature so far.

Simple variable selection approaches such as stepwise selection (e.g., Glahn and Lowry 1972; Wilks 2011) could also be adapted to nonhomogeneous regression. However, these require a high number of model fittings and quickly become computationally infeasible for a higher number of potential predictor variables.

Nonhomogeneous boosting is an easily implementable extension of the popular nonhomogeneous regression to automatically select the most relevant input variables of possibly very large sets of candidates. To facilitate the implementation and adaption to other problems, we provide all our algorithms in the software package crch (Messner et al. 2016) for the open source software R (R Core Team 2015).

Acknowledgments. We thank two anonymous reviewers for their valuable comments to improve this manuscript. This study was supported by the Austrian Science Fund (FWF) TRP 290-N26. Data from the ECMWF forecasting system were obtained from the ECMWF Data Server.

<table>
<thead>
<tr>
<th>50 days</th>
<th>100 days</th>
<th>200 days</th>
<th>400 days</th>
<th>800 days</th>
<th>1500 days</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>10</td>
<td>13</td>
<td>26</td>
<td>36</td>
<td>51</td>
</tr>
</tbody>
</table>

**Fig. 8.** Continuous ranked probability skill score (CRPSS) relative to NGR for 42-h maximum temperature forecasts and (left to right) different training data lengths aggregated over the five stations studied. The respective median numbers of selected input variables are shown at the top of each panel. Circles, boxes, and whiskers have the same meaning as in Fig. 5.
APPENDIX

Partial Derivatives of Log-Likelihood

This appendix provides the partial derivatives ∂L(μ, σ)/∂μ and ∂L(μ, σ)/∂σ of the log-likelihood L [Eq. (4)] that are used in the boosting algorithm [Eq. (6)]:

\[
\frac{\partial L(\mu, \sigma)}{\partial \mu} = \frac{y - \mu}{\sigma^2}, \quad (A1)
\]

\[
\frac{\partial L(\mu, \sigma)}{\partial \sigma} = \frac{1}{\sigma} + \frac{(y - \mu)^2}{\sigma^3}. \quad (A2)
\]

REFERENCES


