Weighted verification tools to evaluate univariate and multivariate probabilistic forecasts

Sam Allen
University of Bern
Oeschger Centre for Climate Change Research

Abstract

Probabilistic forecasts are often assessed and compared using proper scoring rules. To emphasise particular outcomes when evaluating probabilistic forecasts, various approaches have been proposed to construct weighted scoring rules, both in the univariate and multivariate setting. Allen *et al.* (2023a) discuss and compare these different approaches, and illustrate how the theory underlying weighted scoring rules can also be applied when assessing forecast calibration. This vignette reproduces the results presented in this paper, and demonstrates how the accompanying R package allows these weighted verification tools to be implemented in practice.

Keywords: probabilistic forecast evaluation, proper scoring rules, PIT histograms, R.

1. Introduction

Methods to verify and evaluate forecasts are integral to the development and improvement of forecast systems. Traditionally, the evaluation of probabilistic forecasts focuses on two aspects of forecast performance: forecast accuracy and forecast calibration. Forecast accuracy is a measure of the agreement between a forecast and the corresponding observation, and is quantified using proper scoring rules. Scoring rules summarise forecast performance using a single numerical value, allowing competing forecasters to be ranked and compared objectively; proper scoring rules further encourage the forecaster to issue what they truly believe will occur (Gneiting and Raftery 2007). Forecast calibration, on the other hand, considers to what extent forecasts are reliable, or trustworthy - for example, do the observed outcomes occur with the same probability with which they are predicted? This is typically assessed visually using graphical diagnostic tools, though statistical tests also exist to check calibration more rigorously.

However, it is often the case that certain outcomes are of more interest than others: when predicting events that can have a high impact on the forecast user, for example. Hence, over the past decade or so, several approaches have been developed to target particular outcomes during forecast evaluation. This is typically achieved by incorporating a weight function into conventional methods. Allen *et al.* (2023a) recently reviewed weighted verification tools, and the goal of this vignette is to demonstrate how the accompanying R package allows users to implement these tools in practical applications. The package is not complete, and some potential extensions to the existing functionality are discussed in the final section.

2. Weighted verification tools

2.1. Weighted scoring rules

Scoring rules are functions that take a probabilistic forecast F and an observation y as inputs, and output a real (possibly non-finite) value that quantifies the forecast's accuracy. Scoring rules are typically negatively oriented, such that a lower score indicates a more accurate forecast, and a scoring rule S is called proper with respect to a class of probability distributions $\mathcal{F}$ if

$$\mathsf{E}_{Y\sim G} S(G, Y) \leq \mathsf{E}_{Y\sim G} S(F, Y), \tag{1}$$

for all $F, G \in \mathcal{F}$, where E_Y denotes the expectation with respect to the random variable Y. S is strictly proper with respect to $\mathcal{F}$ if the above inequality is strict.

For example, when the observations are univariate, i.e. $y \in \mathbb{R}$, probabilistic forecasts are often assessed using the *continuous ranked probability score* (CRPS; Matheson and Winkler 1976)

$$\begin{aligned} \mathrm{CRPS}(F, y) &= \int_{-\infty}^{\infty} (F(z) - \mathbb{1}\{y \leq z\})^2 \, \mathrm{d}z \\ &= \mathsf{E}_F|X - y| - \frac{1}{2}\mathsf{E}_F|X - X'|, \end{aligned} \tag{2}$$

where F is the cumulative distribution function associated with the forecast distribution, $X, X' \sim F$ are independent, and $\mathbb{1}$ denotes the indicator function. When the observations are multivariate, i.e. $y \in \mathbb{R}^d$ for $d > 1$, popular scoring rules include the *energy score* (ES; Gneiting and Raftery 2007) and the *variogram score* (VS; Scheuerer and Hamill 2015):

$$\mathrm{ES}(F, y) = \mathsf{E}_F\|X - y\| - \frac{1}{2}\mathsf{E}_F\|X - X'\|, \tag{3}$$

where $\|\cdot\|$ is the Euclidean distance in $\mathbb{R}^d$, and $X, X' \sim F$ are independent;

$$\mathrm{VS}_p(F, y) = \sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j}(\mathsf{E}_F|X_i - X_j|^p - |y_i - y_j|^p)^2, \tag{4}$$

where $y = (y_1, \ldots, y_d) \in \mathbb{R}^d$, $X = (X_1, \ldots, X_d) \sim F$, $h_{i,j}$ are non-negative scaling parameters, and $p > 0$ is the order of the score. Typically, p is chosen to be one half, and the scaling parameters $h_{i,j}$ are all set to one.

These scoring rules all fit into the class of kernel scores, a general class of scoring rules based on conditionally negative definite (c.n.d.) kernels (Gneiting and Raftery 2007). Letting $\mathcal{X}$ denote the set of possible outcomes, a c.n.d. kernel is a symmetric function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ for which

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \rho(x_i, x_j) \leq 0 \tag{5}$$

for all $n \in \mathbb{N}, x_1, \ldots, x_n \in \mathcal{X}$, and $c_1, \ldots, c_n \in \mathbb{R}$ that sum to zero.

Given a c.n.d. kernel ρ, the kernel score corresponding to ρ is the scoring rule

$$S_\rho(F, y) = \mathsf{E}_F[\rho(X, y)] - \frac{1}{2}\mathsf{E}_F[\rho(X, X')] - \frac{1}{2}\rho(y, y), \tag{6}$$

where $X, X' \sim F$ are independent. Clearly, the CRPS is the kernel score corresponding to $\rho(z, z') = |z - z'|$ for $z, z' \in \mathbb{R}$, while the ES is the kernel score corresponding to $\rho(z, z') = \|z - z'\|$ for $z, z' \in \mathbb{R}^d$. Allen *et al.* (2023b) demonstrate that the variogram score additionally fits into this framework.

Scoring rules encountered in practice typically evaluate the overall performance of the forecast distribution. However, it is often the case that certain outcomes are of more interest than others, and these should therefore be emphasised when calculating forecast accuracy. To achieve this, weighted scoring rules have been proposed that introduce a user-specified (non-negative) weight function w into conventional scoring rules. This weight function can then be chosen to emphasise particular outcomes of interest.

Weighted scoring rules have been studied in most detail using the CRPS. For example, Matheson and Winkler (1976) and Gneiting and Ranjan (2011) introduced the *threshold-weighted CRPS* (twCRPS)

$$\begin{aligned} \text{twCRPS}(F, y; w) &= \int_{-\infty}^{\infty} (F(z) - \mathbb{1}\{y \leq z\})^2 w(z) \, \mathrm{d}z \\ &= \mathsf{E}_F|v(X) - v(y)| - \frac{1}{2}\mathsf{E}_F|v(X) - v(X')|, \end{aligned} \tag{7}$$

where $X, X' \sim F$ are independent, $w : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a weight function, and v is any function such that $v(z) - v(z') = \int_{z'}^{z} w(x) \, \mathrm{d}x$, referred to as the chaining function (Allen *et al.* 2023b); Holzmann and Klar (2017) proposed an *outcome-weighted CRPS* (owCRPS)

$$\begin{aligned} \text{owCRPS}(F, y; w) &= w(y) \int_{-\infty}^{\infty} (F_w(z) - \mathbb{1}\{y \leq z\})^2 \, \mathrm{d}z \\ &= w(y)\text{CRPS}(F_w, y), \end{aligned} \tag{8}$$

where

$$F_w(x) = \frac{\mathsf{E}_F\left[\mathbb{1}\{X \leq x\} w(X)\right]}{\mathsf{E}_F[w(X)]}, \tag{9}$$

with $X \sim F$; and Allen *et al.* (2023b) introduced a *vertically re-scaled CRPS* (vrCRPS)

$$\begin{aligned} \text{vrCRPS}(F, y; w, x_0) =& \mathsf{E}_F[|X - y|w(X)w(y)] - \frac{1}{2}\mathsf{E}_F[|X - X'|w(X)w(X')] \\ &+ (\mathsf{E}_F[|X - x_0|w(X)] - |y - x_0|w(y))(\mathsf{E}_F[w(X)] - w(y)), \end{aligned} \tag{10}$$

where $X, X' \sim F$ are independent, and x_0 is an arbitrary real value.

Allen *et al.* (2023b) then demonstrate that these three approaches to construct weighted versions of the CRPS can be generalised to the arbitrary class of kernel scores, allowing us to introduce, for example, threshold-weighted, outcome-weighted, and vertically re-scaled versions of the energy score and variogram score.

$$\text{twES}(F, y; v) = \mathsf{E}_F\|v(X) - v(y)\| - \frac{1}{2}\mathsf{E}_F\|v(X) - v(X')\|; \tag{11}$$

$$\text{owES}(F, y; w) = \frac{1}{\bar{w}_F}\mathsf{E}_F\left[\|X - y\|w(X)w(y)\right] - \frac{1}{2\bar{w}_F^2}\mathsf{E}_F\left[\|X - X'\|w(X)w(X')w(y)\right]; \tag{12}$$

$$\begin{aligned} \text{vrES}(F, y; w, x_0) =& \mathsf{E}_F[\|X - y\|w(X)w(y)] - \frac{1}{2}\mathsf{E}_F[\|X - X'\|w(X)w(X')] \\ &+ \left(\mathsf{E}_F[\|X - x_0\|w(X)] - \|y - x_0\|w(y)\right)\left(\mathsf{E}_F[w(X)] - w(y)\right); \end{aligned} \tag{13}$$

$$\text{twVS}_p(F, y; v) = \sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j}(\mathsf{E}_F|v(X)_i - v(X)_j|^p - |v(y)_i - v(y)_j|^p)^2; \tag{14}$$

$$\text{owVS}_p(F, y; w) = w(y)\sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j}\left(\frac{1}{\bar{w}_F}\mathsf{E}_F\left[|X_i - X_j|^p\, w(X)\right] - |y_i - y_j|^p\right)^2; \tag{15}$$

$$\begin{aligned}\text{vrVS}_p(F, y; w, x_0) =& \mathsf{E}_F\left[w(X)w(y)\sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j}(|X_i - X_j|^p - |y_i - y_j|^p)^2\right] \\ &- \frac{1}{2}\mathsf{E}_F\left[w(X)w(X')\sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j}(|X_i - X_j|^p - |X'_i - X'_j|^p)^2\right] \\ &+ \left(\mathsf{E}_F\left[w(X)\sum_{j=1}^{d} h_{i,j}(|X_i - X_j|^p - |x_{0,i} - x_{0,j}|^p)^2\right]\right. \\ &\left. - w(y)\sum_{j=1}^{d} h_{i,j}(|y_i - y_j|^p - |x_{0,i} - x_{0,j}|^p)^2\right)(\bar{w}_F - w(y)),\end{aligned} \tag{16}$$

where $w : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is a weight function, $v : \mathbb{R}^d \to \mathbb{R}^d$ is a chaining function, $\bar{w}_F = \mathsf{E}_F[w(X)]$, and $x_0 \in \mathbb{R}^d$.

These weighted multivariate scoring rules allow multivariate outcomes of interest to be emphasised during forecast evaluation. The advantages and disadvantages of the various approaches to weight scoring rules are discussed in Allen *et al.* (2023a).

2.2. Conditional PIT plots

Proper scoring rules condense forecast performance into a single numerical value, allowing competing forecasters to be ranked and compared. However, scoring rules do not readily determine whether a prediction system is trustworthy, in the sense that the observed outcomes are statistically consistent with the forecasts that were issued. If the forecasts do align with the observations, then the prediction system is said to reliable, or calibrated.

For univariate outcomes, forecast calibration is typically assessed using probability integral transform (PIT) values. It is a well-known result that if the outcome Y is a continuous random variable with cumulative distribution function F, then the random variable $F(Y)$ will follow the standard uniform distribution. Hence, if a forecaster issues a sequence of predictive distributions $F_1, \ldots, F_n$, and observes a sequence of observations $y_1, \ldots, y_n$, then the PIT values $F_1(y_1), \ldots, F_n(y_n)$ should resemble a sample from the standard uniform distribution if the forecaster is (probabilistically) calibrated. In this sense, calibration means that the observations can be interpreted as random draws from the forecast distributions. A simple extension exists when the forecast distributions are not continuous, such as in ensemble forecasting; note that PIT values provide a generalisation of the rank of the observation among a finite sample from the forecast distribution.

While statistical tests exist to assess whether or not the PIT values resemble a sample from a standard uniform distribution, calibration is typically assessed graphically, using either a histogram (Gneiting *et al.* 2007) or a P-P plot (also called a reliability diagram; Gneiting and Resin 2021) of the PIT values. If the PIT values are indeed samples from the standard

uniform distribution, then the PIT histogram will be flat, and the PIT reliability diagram will be a straight line along the diagonal (up to sampling variation); if this is not the case, then there is evidence to suggest that the forecasts are miscalibrated, and the behaviour of the deviations can be used to diagnose the nature of the forecast errors.

However, as with conventional scoring rules, PIT histograms assess overall forecast performance. Allen *et al.* (2023a) demonstrate that the theory underlying weighted scoring rules can be applied to checks for forecast calibration by considering conditional PIT values. In particular, if we are interested in values in the range (a, b), for $a, b \in \bar{\mathbb{R}}$, then we can restrict attention to the forecast distributions and the observations within this range. That is, we can calculate the conditional PIT values

$$G(y) = \frac{F(y) - F(a)}{F(b) - F(a)} \tag{17}$$

for all observations y that are between a and b. If interest in on values above some threshold t, then we can set $a = t$ and $b = \infty$.

If the conditional distribution of the outcome variable Y (given that Y is between a and b) is indeed the conditional distribution predicted by the forecasts, then these conditional PIT values should resemble a sample from the standard uniform distribution. This can again be verified by displaying these conditional PIT values in a histogram or reliability diagram. Allen *et al.* (2023a) also discuss extensions of these conditional PIT values to the multivariate setting.

3. Examples

3.1. Weighted scoring rules

In this section, we discuss the implementation of weighted scoring rules and conditional PIT histograms and reliability diagrams using the accompanying R package.

Threshold-weighted and outcome-weighted versions of the CRPS, ES, and VS (and a kernel score based on the Gaussian kernel) are now available in the **scoringRules** package in R, for forecasts in the form of a predictive sample or ensemble (Jordan *et al.* 2019; Allen 2023). Here, functions are available to compute weighted versions of the CRPS when the forecast distribution belongs to a few parametric families. For example, when the forecast is a normal distribution, the threshold-weighted, outcome-weighted, and vertically re-scaled CRPS can be calculated using

```
twcrps_norm(y, mean = 0, sd = 1, a = -Inf, b = Inf)
owcrps_norm(y, mean = 0, sd = 1, a = -Inf, b = Inf, BS = T)
vrcrps_norm(y, mean = 0, sd = 1, a = -Inf, b = -Inf)
```

As in **scoringRules**, the `mean` and `sd` arguments refer to the mean and standard deviation of the normal predictive distributions, while `a` and `b` refer to the upper and lower bounds in the weight function $w(z) = \mathbb{1}\{a < z < b\}$, which emphasises observations between `a` and `b`. In the default case, `a = -Inf` and `b = Inf`, and the weighted scoring rules all revert to the unweighted CRPS.

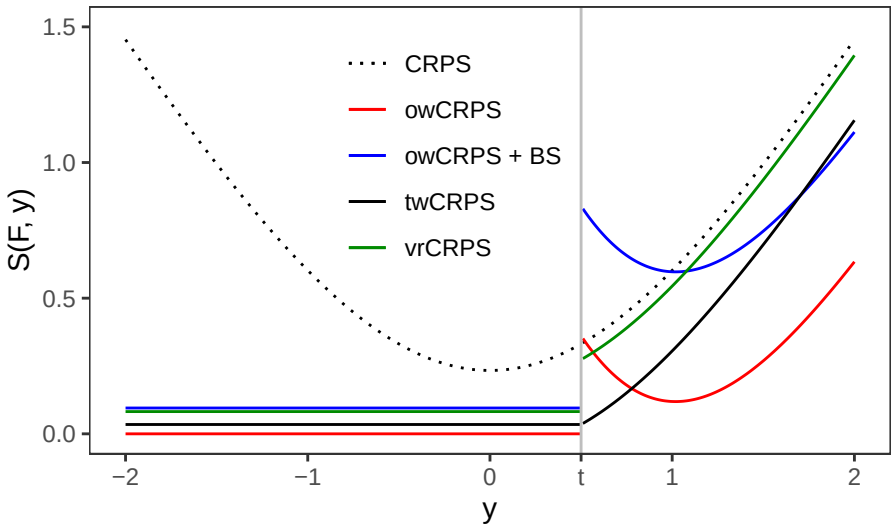

Figure 1: Weighted CRPS as a function of the observation y when the forecast is a standard normal distribution.

The `BS` argument specifies whether or not the outcome-weighted CRPS should be complemented by adding the Brier score (see Holzmann and Klar 2017, for details). In doing so, the scoring rule will not only assess the shape of the conditional distribution, but also the probability it assigns to the region (a, b).

While **scoringRules** allows the weighted scoring rules to be implemented with arbitrary, user-specified weight functions, this is difficult to achieve for parametric forecast distributions. Hence, functionality only currently exists for the weight function $w(z) = \mathbb{1}\{a < z < b\}$, which is most commonly employed in practice. If other weight functions are desired, then the user can first draw a large sample from the forecast distribution, which can then be assessed using the `twcrps_sample()` and `owcrps_sample()` functions in **scoringRules**.

The following code can be used to obtain weighted (and unweighted) CRPS values corresponding to a standard normal predictive distribution, for a range of possible observations. These scores are displayed in Figure 1 as a function of the observation y.

```
R> y <- seq(-2, 2, 0.01)
R> mu <- 0
R> sigma <- 1
R> t <- 0.5
R>
R> s <- crps_norm(y, mu, sigma)
R> tws <- twcrps_norm(y, mu, sigma, a = t)
R> ows <- owcrps_norm(y, mu, sigma, a = t, BS = F)
R> ows_bs <- owcrps_norm(y, mu, sigma, a = t)
R> vrs <- vrcrps_norm(y, mu, sigma, a = t)
```

Weighted scoring rules can be implemented analogously for logistic and Student's t distributions, e.g.

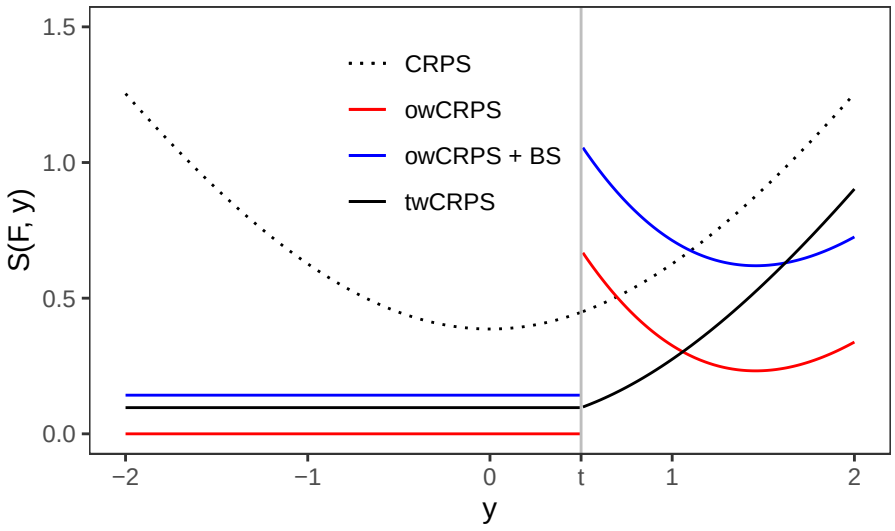

Figure 2: As in Figure 1 but for a standard logistic forecast distribution.

```
twcrps_logis(y, location = 0, scale = 1, a = -Inf, b = Inf)
owcrps_logis(y, location = 0, scale = 1, a = -Inf, b = Inf, BS = T)
```

and weighted versions of the CRPS are similarly displayed for a standard logistic forecast distribution in Figure 2.

Functionality is not currently available to compute the vertically re-scaled CRPS for parametric distributions other than the normal distribution. We note, however, that it should be straightforward to obtain an analytical formula for the vertically re-scaled CRPS for several familiar distributions.

3.2. Conditional PIT plots

The package additionally contains functions to check forecast calibration using PIT histograms and reliability diagrams

```
pit_hist(z, bins = NULL, ranks = TRUE, title = NULL, ymax = NULL,
         ylab = "Rel. Freq.", xlab = "Rank")
pit_reldiag(z, resampling = TRUE, n_resamples = 1000, region_level = 0.9,
            title = NULL)
```

The `pit_hist()` function takes a vector `z` as input, containing ranks (`ranks = TRUE`) or PIT values (`ranks = FALSE`), and plots a histogram of these values. The function additionally takes several arguments related to the design of the plot, including the number of bins in the histogram (`bins`), the x- and y-axes labels (`xlab` and `ylab`), the title (`title`), and the maximum of the y-axis (`ymax`).

The `pit_reldiag()` function, on the other hand, displays a reliability diagram (i.e. P-P plot) containing the empirical distribution function of the PIT values in the vector `z`. The `resampling` argument specifies whether consistency intervals are to be generated around the

diagonal line, indicating how much sampling uncertainty would be present if the forecasts were calibrated, while `n_resamples` and `region_level` specify the number of resamples to be calculated and the nominal coverage of the consistency interval to be displayed.

These functions can also be applied to conditional PIT values. Conditional PIT values corresponding to parametric forecast distributions can be calculated using e.g.

```
cpit_norm(y, mean = 0, sd = 1, a = -Inf, b = Inf)
```

for the normal distribution. Here, `y` is a vector of observations, and the remaining arguments are as described above. Functionality exists to calculate conditional PIT values for a wide range of families of distributions, all distributions for which the distribution function is readily available in R, e.g. using `pnorm()`.

Conditional PIT values can also be calculated based on predictive samples, or ensemble forecasts, using the `cpit_sample()` function

```
cpit_sample(y, dat, a = -Inf, b = Inf, bw = NULL)
```

Here, `dat` is a matrix with each row containing a sample or ensemble member. To calculate the conditional PIT values, the sample is first smoothed using kernel density estimation; note that this can lead to unreliable results if interest is on outcomes that exceed all sample members, since the conditional PIT values will rely on how the density estimation extrapolates beyond the observed sample. The `bw` argument specifies the bandwidth parameter to use within kernel density estimation; if this is not provided, then it is selected automatically based on the data.

For example, consider the case where the observations are drawn from a $\mathcal{N}(\mu, \sigma^2)$, with $\mu \sim \mathcal{N}(0, 1-\sigma^2)$ and $\sigma^2 = 1/3$. The ideal forecaster issues the distribution $\mathcal{N}(\mu, \sigma^2)$ as their forecast, which, of course, is calibrated. To verify this, Figure 3 displays the corresponding PIT histogram and reliability diagram.

The conditional PIT values can similarly be visualised using these functions, and Figure 3 additionally displays these values, when $a = 1$ and $b = \infty$, i.e. when interest is on outcomes greater than one. For comparison, Figure 3 also contains a histogram of the PIT values (*not* the conditional PIT values) when attention is restricted to observations that exceed one.

These so-called restricted PIT values are not uniformly distributed, highlighting why alternative approaches are required to focus on particular outcomes when evaluating forecast calibration. The PIT values and conditional PIT values, on the other hand, appear to resemble a standard uniform distribution.

Now suppose the observations are drawn from a logistic distribution with mean $\mu \sim \mathcal{N}(0, 1-\sigma^2)$ and standard deviation equal to σ, with $\sigma^2 = 1/3$. Consider two alternative forecast distributions, based on the normal distribution and Student's t distribution with five degrees of freedom. PIT histograms and PIT reliability diagrams corresponding to these three forecasters are displayed in Figure 4. Conditional PIT values are also displayed for when $a = 2$ and $b = \infty$.

Clearly, the ideal forecasts, constructed using the logistic distribution, are probabilistically and conditionally calibrated, whereas the alternative forecast strategies lead to forecasts with alternative biases when forecasting high outcomes.

```
R> n <- 10000
R> t <- 1
R>
R> sig <- sqrt(1/3)
R> mu <- rnorm(n, 0, sqrt(1 - sig^2))
R> y <- rnorm(n, mu, sig)                              # observations
R>
R> F_y <- pnorm(y, mu, sig)                            # PIT values
R> F_y_res <- pnorm(y[y > t], mu[y > t], sig)          # restricted PIT values
R> F_y_con <- cpit_norm(y, mu, sig, a = t)             # conditional PIT values
R>
R> bins <- 10
R> p_s <- pit_hist(F_y, ranks = F, bins, ymax = 0.35, title = "PIT")
R> p_r <- pit_hist(F_y_res, ranks = F, bins, ymax = 0.35, title = "Restricted PIT")
R> p_c <- pit_hist(F_y_con, ranks = F, bins, ymax = 0.35, title = "Conditional PIT")
R>
R> p_s_rd <- pit_reldiag(F_y, title = "PIT")
R> p_r_rd <- pit_reldiag(F_y_res, title = "Restricted PIT")
R> p_c_rd <- pit_reldiag(F_y_con, title = "Conditional PIT")
R>
R> gridExtra::grid.arrange(p_s, p_r, p_c, p_s_rd, p_r_rd, p_c_rd, nrow = 2)
```

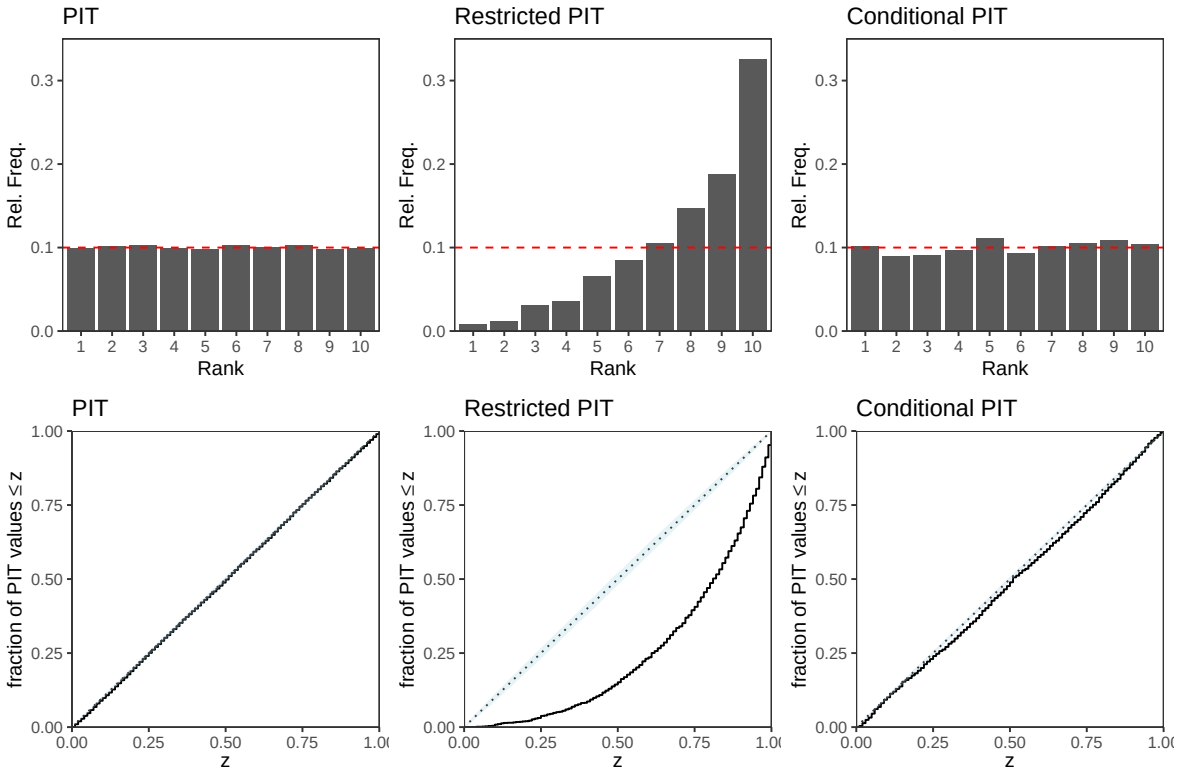

Figure 3: PIT histograms and PIT reliability diagrams for the ideal forecaster when the observations are drawn from the distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu \sim \mathcal{N}(0, 1 - \sigma^2)$ and $\sigma^2 = 1/3$. Histograms and reliability diagrams are also displayed for PIT values when attention is restricted to observations that exceed 1, as well as conditional PIT values in this case.

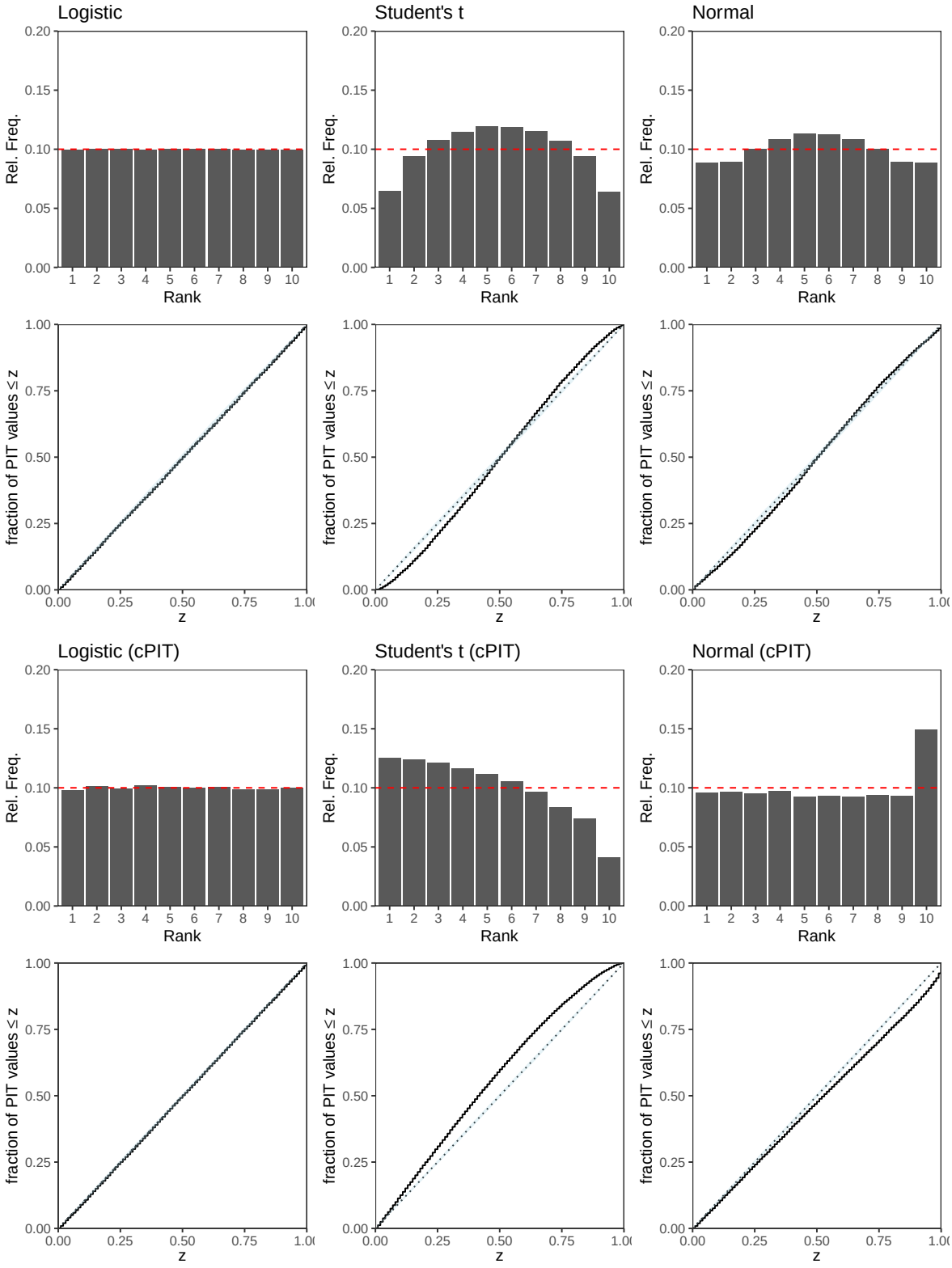

Figure 4: PIT histograms and reliability diagrams for logistic, Student's t, and normal forecast distributions, when the observations are drawn from a logistic distribution with mean $\mu \sim \mathcal{N}(0, 1 - \sigma^2)$ and variance equal to $\sigma^2 = 1/3$. The bottom two rows display the conditional PIT values when interest is on outcomes that are greater than two.

Heat level	Criterion
1	T < 25°C on all three days
2	T ≥ 25°C on one or two days
3	T ≥ 25°C on all three days, T < 27°C on at least one day
4	T ≥ 27°C on all three days

Table 1: MeteoSwiss heat warning levels given daily mean temperatures (T) over a three day period.

4. Application

4.1. Data

In this section, we reproduce the results presented in Allen *et al.* (2023a), wherein the weighted verification tools are used to evaluate forecasts obtained from an operational ensemble prediction system (COSMO-E) at the Swiss Federal Office of Meteorology and Climatology (MeteoSwiss). We consider COSMO-E forecasts for the daily mean temperature, and use the weighted verification tools to put emphasis on extreme heat events. The COSMO-E prediction system generates ensemble forecasts comprised of $M = 21$ members, and forecasts are available at 149 synoptic weather stations across Switzerland, for the seven summer seasons (May–September) from 2014 to 2020.

Extreme heat events are defined in terms of MeteoSwiss' heat warning levels, which are used to inform the public when dangerously high temperatures are expected to occur. There are four heat levels, defined in Table 1, which depend on the daily mean temperature over a three day period. We therefore consider daily mean temperature forecasts over the coming three days. The forecasts are evaluated at each lead time separately using univariate verification techniques, while multivariate tools are used to assess the forecasts over the entire three day period.

The COSMO-E ensemble forecasts are compared to two alternative forecast strategies: a climatological forecast, which always issues the local climatological temperature distribution as the prediction, and a statistically post-processed forecast, designed to remove systematic errors that occur in the COSMO-E forecasts. The post-processing method is based on an approach employed at MeteoSwiss, and further details regarding these methods (and the COSMO-E forecasts) can be found in Allen *et al.* (2023a).

Note that the exact data used in Allen *et al.* (2023a) is owned by MeteoSwiss and therefore not publicly available. To circumvent this, random noise has been added to the data, and a subset of this noisy data is available in this package. The results that we obtain here are therefore not identical to those presented in Allen *et al.* (2023a), though the implementation of the weighted verification tools is the same. The data can be accessed using

```
R> data("noisy_data", package = "WeightedForecastVerification")
R> list2env(noisy_data, globalenv())
R> rm(noisy_data)
```

The result is a list of matrices, which are converted to independent variables using the `list2env()` function. These matrices include the observations (`obs_dat`), the COSMO-

E ensemble forecasts (`ens_raw`), and ensemble forecasts obtained from the climatological (`ens_clim`), and statistical post-processed (`ens_pp`) forecast distributions. Both the climatological and post-processed ensemble forecasts correspond to quantiles from a normal distribution, which are then reordered according to some multivariate dependence template. The means and standard deviations of these normal distributions are also available (`clim_mean`, `clim_sd`, `pp_mean`, `pp_sd`). The subset of data contains three lead times, 2000 forecast cases, and 10 ensemble members.

4.2. Results

Firstly, consider the unweighted scores for the three forecast strategies. These scores can be obtained using the **scoringRules** functionality. The following code returns the average CRPS for the three forecast strategies as a function of the forecast lead time.

```
R> n_lt <- 3
R>
R> s_clim <- sapply(1:n_lt, function(lt)
+    crps_norm(obs_dat[lt, ], clim_mean[lt, ], clim_sd[lt, ]))
R> s_raw <- sapply(1:n_lt, function(lt)
+    crps_sample(obs_dat[lt, ], ens_raw[lt, , ]))
R> s_pp <- sapply(1:n_lt, function(lt)
+    crps_norm(obs_dat[lt, ], pp_mean[lt, ], pp_sd[lt, ]))
R>
R> mean_score <- c(colMeans(s_clim), colMeans(s_raw), colMeans(s_pp))
R>
R> score_mat <- matrix(mean_score, nrow = 3, byrow = T)
R> rownames(score_mat) <- c("Clim.", "COSMO", "PP")
R> colnames(score_mat) <- 1:n_lt
R> print(score_mat)

         1     2     3
Clim. 3.39 3.452 3.389
COSMO 1.15 1.050 1.155
PP    0.95 0.902 0.962
```

The accuracy of the forecast trajectories over the three days can similary be assessed using the energy score and variogram score.

```
        ES   VS
Clim. 6.33 2.73
COSMO 2.17 1.61
PP    1.91 1.54
```

Post-processing appears beneficial here, both in the univariate and multivariate case, while the raw COSMO-E forecasts significantly outperform the climatological forecasts.

The calibration of the forecasts can be assessed using PIT histograms and PIT reliability diagrams. Having calculated the ranks and PIT values, Figure 5 displays the resulting rank

histogram (for the COSMO-E ensemble), PIT histograms (for the climatological and post-processed forecast distributions), and PIT reliability diagrams, at a lead time of three days.

The COSMO-E forecasts, despite achieving better scores than the climatological forecasts, are clearly under-dispersed and thus miscalibrated. The PIT histogram and reliability diagram of the climatological forecasts indicate that the normal distribution is perhaps not the most appropriate distribution to use when modelling the daily mean temperatures in this study. The post-processed forecast distributions, on the other hand, appear well-calibrated.

The deviation of the PIT histograms from a flat histogram can be quantified using reliability indices. These can be calculated using the function

```
rel_index(z, bins = NULL, ranks = TRUE, method = "absolute")
```

The argument `z` again represents a vector of ranks or PIT values (depending on `ranks`), while `bins` is the number of bins to be used in the histogram. Three methods are then available to calculate reliability indices: `method = 'absolute'` (default) measures the sum of absolute distances between the observed relative frequencies and the optimal frequency, 1/`bins`; `method = 'squared'` measures the sum of squared distances between the observed relative frequencies and 1/`bins`, which follows a chi-squared distribution when appropriately scaled; and `method = 'entropy'` measures the entropy of the relative frequencies, which will be equal to 1 for a uniform rank histogram, and between 0 and 1 otherwise. These methods are discussed in detail in Wilks (2019).

We can calculate reliability indices for the three forecast strategies considered here as a function of lead time. The reliability index is closest to zero for the post-processed forecasts, and is largest for the raw COSMO-E forecasts, which (for `method = 'absolute'`) suggests that the post-processed forecasts are well-calibrated, whereas the COSMO-E forecasts are the least calibrated. These results align with the PIT histograms and reliability diagrams in Figure 5.

```
           1      2      3
Clim. 0.1401 0.1411 0.1543
COSMO 0.4975 0.4655 0.4685
PP    0.0639 0.0613 0.0409
```

While these methods assess overall forecast accuracy, consider now forecasts made for extreme heat events. We can evaluate the accuracy of these forecasts using weighted scoring rules. For example, to calculate the threshold-weighted CRPS for the weight function $w(z) = \mathbb{1}\{z > t\}$, for some threshold t, we can use

```
R> t_vec <- -5:30              # select range of thresholds
R> lead <- 3                   # select lead time
R>
R> wcrps_raw <- sapply(t_vec, function(t)
+    twcrps_sample(obs_dat[lead, ], ens_raw[lead, , ], a = t))
R> wcrps_raw <- colMeans(wcrps_raw)
```

This can be repeated for the alternative forecast strategies, and for the outcome-weighted and vertically re-scaled CRPS, and these weighted scores are displayed as a function of t in Figure 6

```
R> bins <- 11
R> p_c <- pit_hist(pit_clim, bins, ranks = F, title = "Climatology", ymax = 0.32)
R> p_r <- pit_hist(pit_raw, title = "COSMO", ymax = 0.32)
R> p_p <- pit_hist(pit_pp, bins, ranks = F, title = "Post-processed", ymax = 0.32)
R>
R> p_c_rd <- pit_reldiag(pit_clim)
R> p_r_rd <- pit_reldiag(pit_raw, ranks = T)
R> p_p_rd <- pit_reldiag(pit_pp)
R>
R> gridExtra::grid.arrange(p_c, p_r, p_p, p_c_rd, p_r_rd, p_p_rd, nrow = 2)
```

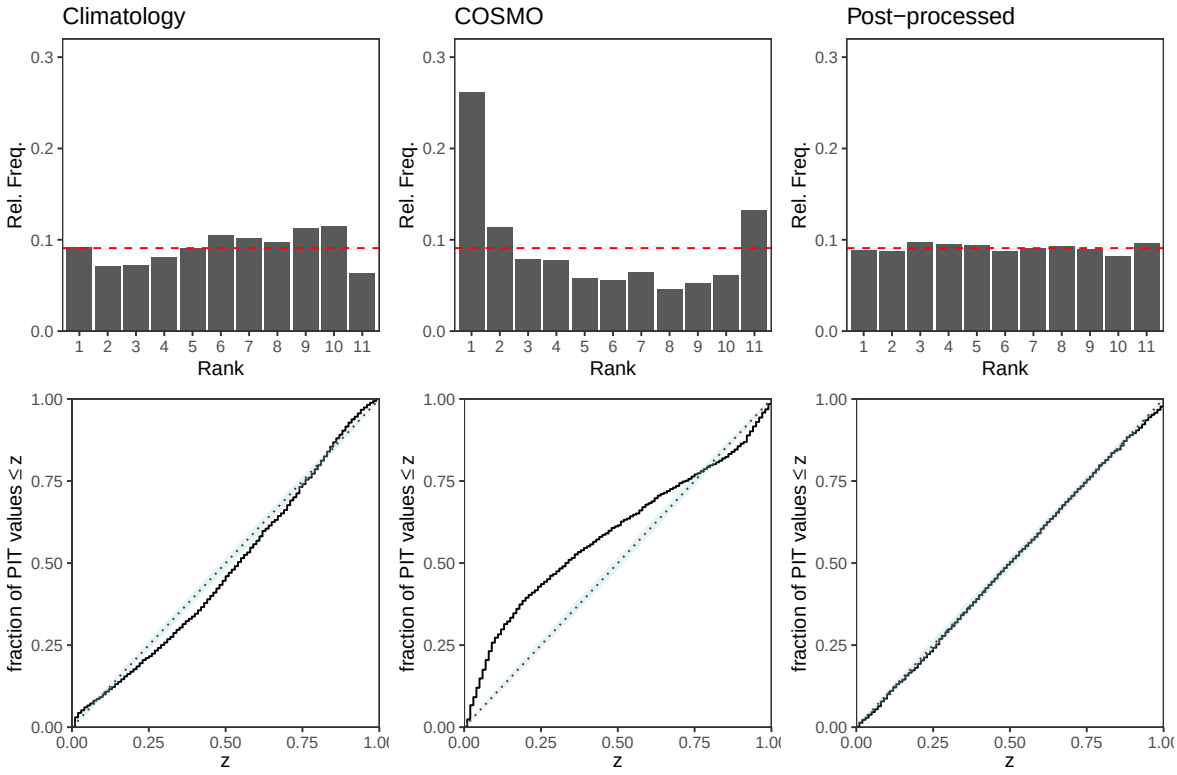

Figure 5: PIT histograms and reliability diagrams for the three forecast strategies, at a lead time of three days.

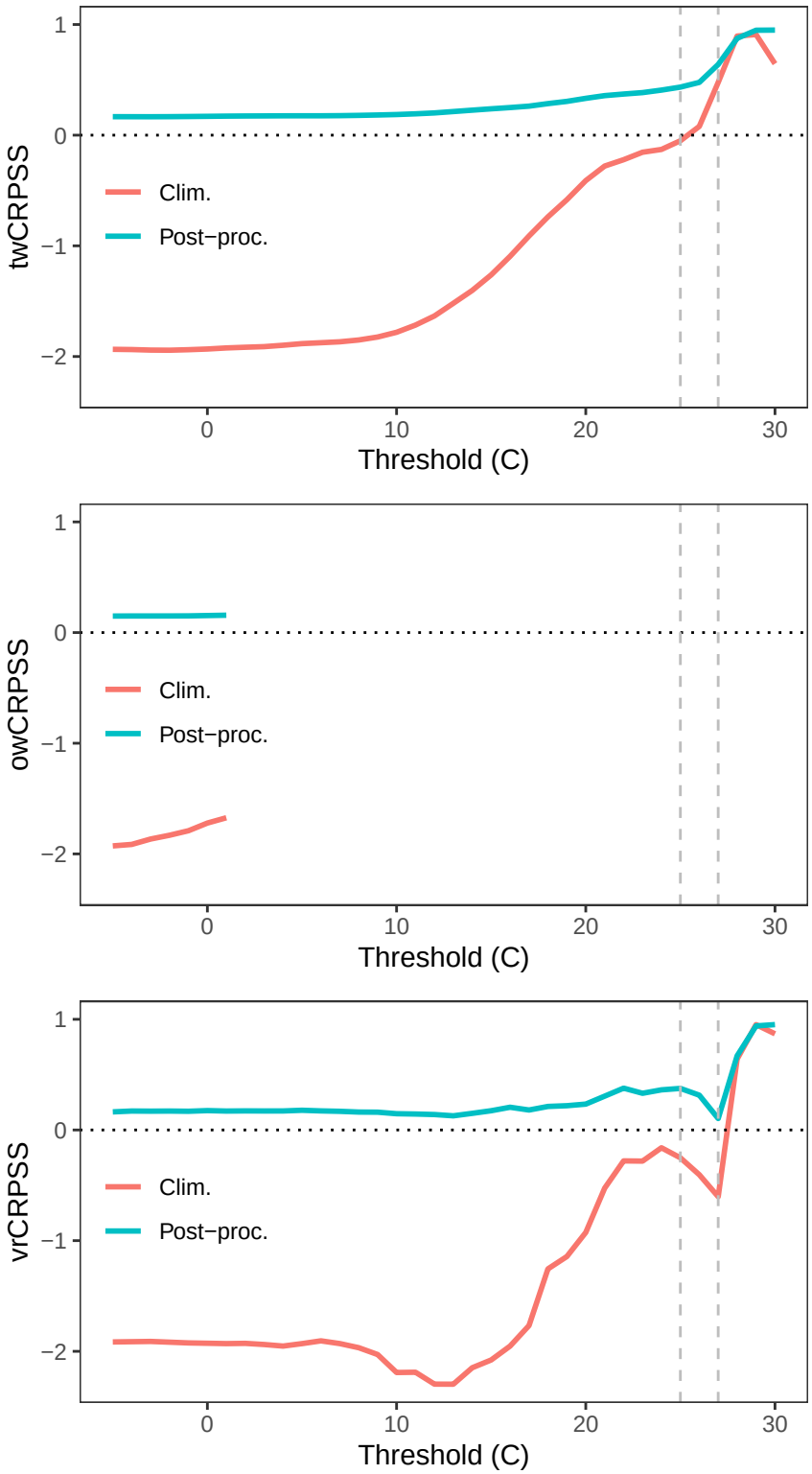

Figure 6: Skill scores for the twCRPS, owCRPS, and vrCRPS as a function of the threshold used in the weight function $w(z) = \mathbb{1}\{z > t\}$ at a lead time of three days. The skill scores are shown for the climatological and post-processed forecast distributions, with the COSMO-E forecasts as the reference approach. Vertical dashed lines are drawn at 25 and 27 degrees.

While the threshold-weighted CRPS can be applied at each lead time separately, if we want to emphasise the heat events defined in Table 1, then we need to consider the multivariate forecast performance. This can be achieved using threshold-weighted versions of the energy and variogram score. In this case, the multivariate forecast distributions are in the form of a finite sample, and hence the `twes_sample()` and `twvs_sample()` functions from **scoringRules** can be employed.

The resulting threshold-weighted energy scores corresponding to each heat level are

```
       All Lev. 1 Lev. 2 Lev. 3 Lev. 4
Clim. 6.33   6.34  0.970 0.1027 0.0847
COSMO 2.17   2.16  1.095 0.1692 0.4604
PP    1.91   1.90  0.829 0.0839 0.1136
```

while the threshold-weighted variogram scores are

```
       All Lev. 1 Lev. 2 Lev. 3 Lev. 4
Clim. 2.73   2.91   4.13  0.248  0.145
COSMO 1.61   1.82   4.28  0.385  0.648
PP    1.54   1.65   2.92  0.198  0.194
```

In both cases, the weight function is chosen to emphasise the heat events defined in Table 1; further details can be found in Allen *et al.* (2023a). These results suggest that although the climatological forecasts perform worst when interest is on all outcomes, these forecasts result in lower scores when interest is on more extreme heat events. The COSMO-E forecasts tend to perform worst with respect to these outcomes.

While weighted scoring rules allow the competing forecast strategies to be compared when predicting particular outcomes, they cannot be used to identify *why* particular forecasts outperform others. To this end, conditional PIT histograms and reliability diagrams assess whether or not the forecast distributions are calibrated when predicting these outcomes. Figure 7 suggests that the COSMO-E forecasts severely over-predict the severity of daily mean temperatures, whereas this is corrected by the statistical post-processing model.

Conditional PIT values focus on forecast performance given that an outcome of interest has occurred. However, they do not assess how well the forecast distributions can predict whether or not these outcomes will occur. This is a binary forecasting problem, and forecasts for the event occurrence can therefore be evaluated using reliability diagrams.

5. Discussion

This vignette describes the weighted forecast verification tools discussed by Allen *et al.* (2023a), and reproduces the results therein. Code to achieve this is available in the accompanying **R** package. This package includes weighted scoring rules, particularly for parametric forecast distributions, as well as functions to plot PIT values and conditional PIT values using both histograms and reliability diagrams. The package is still in development, and there are several possible extensions that could be made.

Firstly, the package currently only contains functionality to apply the vertically re-scaled CRPS to the normal distribution, whereas it should be straightforward to calculate this

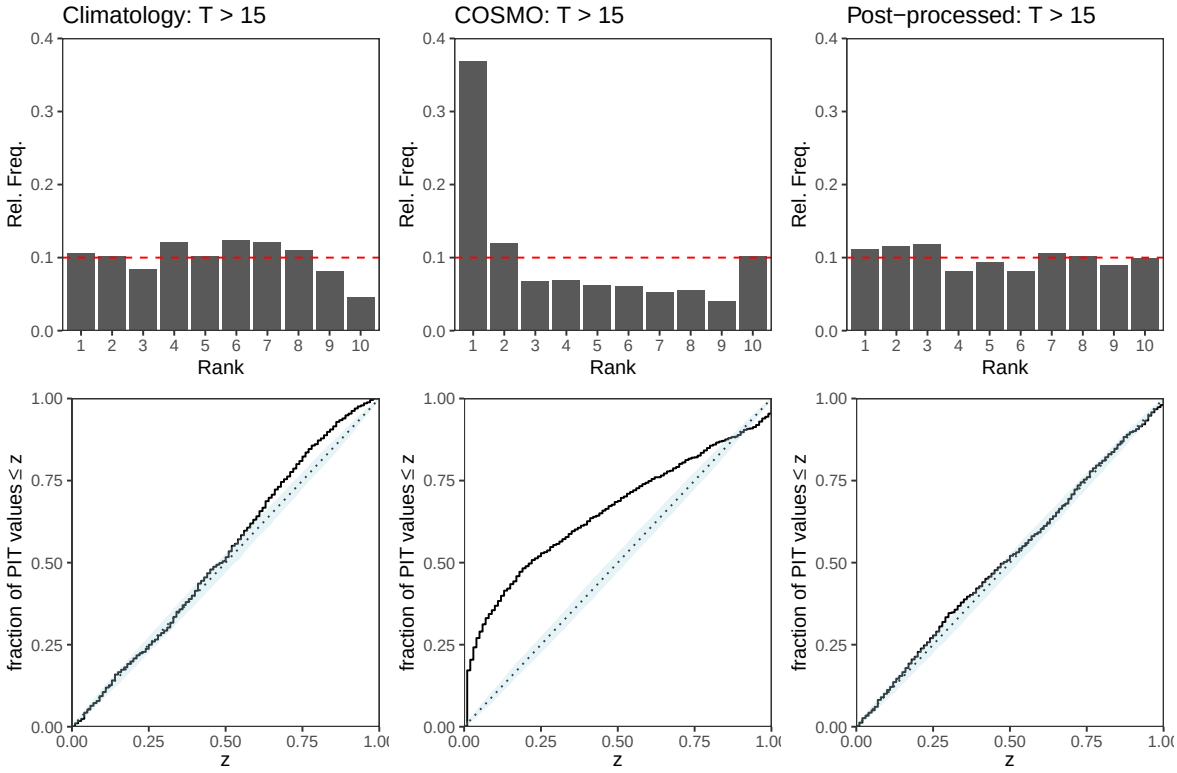

Figure 7: Conditional PIT histograms and reliability diagrams for the three forecast strategies when interest is on daily mean temperatures that exceed 15 degrees Celcius.

score for alternative parametric families, such as the logistic and Student's t distributions. Moreover, while **scoringRules** allows the threshold- and outcome-weighted CRPS to be applied to forecasts in the form of a predictive sample, this is currently not available for the vertically re-scaled CRPS (though code does exist to achieve this at `https://github.com/sallen12/weighted_mv_scores`).

This package also contains several reliability indices to quantify the deviation between a sample of ranks and the uniform distribution over these ranks; these provide a single measure of forecast miscalibration. Currently, however, these are based on ranks, and cannot be applied to PIT values. Measures of deviation between PIT values and the standard uniform distribution, e.g. based on entropy or maximum mean discrepancies, would also be useful in practice, and should be straightforward to incorporate here.

Finally, Allen *et al.* (2023a) additionally discuss how conditional multivariate rank and PIT histograms could be constructed. There is currently no software that contains the functionality to implement checks for multivariate calibration, and this could relatively easily be incorporated into this package, with the ultimate goal of providing a complete collection of methods to assess probabilistic forecast calibration. This could include, for example: reliability diagrams, rank histograms, (conditional) PIT histograms, (conditional) PIT reliability diagrams, (conditional) multivariate rank histograms, reliability indices, tests for calibration, sequential measures of calibration, functional-calibration reliability diagrams, and proper score decompositions, among other things.

References

Allen S (2023). "Weighted scoringRules: Emphasising Particular Outcomes when Evaluating Probabilistic Forecasts."

Allen S, Bhend J, Martius O, Ziegel J (2023a). "Weighted Verification Tools to Evaluate Univariate and Multivariate Probabilistic Forecasts for High-impact Weather Events." *Weather and Forecasting*, **38**, 499–516.

Allen S, Ginsbourger D, Ziegel J (2023b). "Evaluating forecasts for high-impact events using transformed kernel scores." *Journal of Uncertainty Quantification*.

Gneiting T, Balabdaoui F, Raftery AE (2007). "Probabilistic forecasts, calibration and sharpness." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **69**, 243–268.

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

Gneiting T, Ranjan R (2011). "Comparing density forecasts using threshold-and quantile-weighted scoring rules." *Journal of Business & Economic Statistics*, **29**, 411–422.

Gneiting T, Resin J (2021). "Regression diagnostics meets forecast evaluation: Conditional calibration, reliability diagrams, and coefficient of determination." *arXiv preprint arXiv:2108.03210*.

Holzmann H, Klar B (2017). "Focusing on regions of interest in forecast evaluation." *The Annals of Applied Statistics*, **11**, 2404–2431.

Jordan A, Krüger F, Lerch S (2019). "Evaluating Probabilistic Forecasts with scoringRules." *Journal of Statistical Software*, **90**, 1–37.

Matheson JE, Winkler RL (1976). "Scoring rules for continuous probability distributions." *Management Science*, **22**, 1087–1096.

Scheuerer M, Hamill TM (2015). "Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities." *Monthly Weather Review*, **143**, 1321–1334.

Wilks D (2019). "Indices of rank histogram flatness and their sampling properties." *Monthly Weather Review*, **147**, 763–769.

Affiliation:

Sam Allen
University of Bern
Institute of Mathematical Statistics and Actuarial Science
Alpeneggstrasse 22
3012 Bern, Switzerland
E-Mail: sam.allen@unibe.ch