Weighted scoringRules: Emphasising Particular Outcomes when Evaluating Probabilistic Forecasts

Sam Allen
ETH Zurich

Abstract

When predicting future events, it is common to issue forecasts that are probabilistic, in the form of probability distributions over the range of possible outcomes. Such forecasts can be evaluated using proper scoring rules. Proper scoring rules condense forecast performance into a single numerical value, allowing competing forecasters to be ranked and compared. To facilitate the use of scoring rules in practical applications, the **scoringRules** package in R provides popular scoring rules for a wide range of forecast distributions. This paper discusses an extension to the **scoringRules** package that additionally permits the implementation of popular weighted scoring rules. Weighted scoring rules allow particular outcomes to be targeted during forecast evaluation, recognising that certain outcomes are often of more interest than others when assessing forecast quality. This introduces the potential for very flexible, user-oriented evaluation of probabilistic forecasts. We discuss the theory underlying weighted scoring rules, and describe how they can readily be implemented in practice using **scoringRules**. Functionality is available for weighted versions of several popular scoring rules, including the logarithmic score, the continuous ranked probability score (CRPS), and the energy score. Two case studies are presented to demonstrate this, whereby weighted scoring rules are applied to univariate and multivariate probabilistic forecasts in the fields of meteorology and economics.

Keywords: forecast evaluation, probabilistic forecasting, proper scoring rules, weighted scoring rules, R.

Preface

This vignette corresponds to an *arXiv* pre-print of the same name (Allen 2023). The two articles are close to identical at the time of writing (August 26$^{\text{th}}$, 2024).

1. Introduction: Weighted scoring rules

When predicting future events, it is common to issue forecasts that are probabilistic. Probabilistic forecasts generally take the form of probability distributions over the range of possible outcomes, comprehensively describing the predictive uncertainty. To assess the quality of a probabilistic forecast, scoring rules are functions $S(F, y)$ that take a forecast F and the corresponding outcome y as inputs, and output a numerical score that quantifies the forecast's accuracy. Scoring rules therefore condense forecast performance into a single value, providing a convenient framework with which to objectively rank and compare competing forecasts. As

such, scoring rules have become a key component of probabilistic forecast evaluation.

To assess probabilistic forecasts in practice, the **scoringRules** package (Jordan *et al.* 2019) in the programming language R has become a widely used resource. The package contains analytical formulae for the two most popular univariate scoring rules — the logarithmic score (LogS; Good 1952) and the continuous ranked probability score (CRPS; Matheson and Winkler 1976) — for forecast distributions belonging to a range of parametric families. These scoring rules are also available when the forecast is a sample from a predictive distribution, which is often the case in practice. The **scoringRules** package additionally allows samples from multivariate forecast distributions to be evaluated using popular multivariate scoring rules, including the energy score (Gneiting and Raftery 2007), variogram score (Scheuerer and Hamill 2015), and a kernel score based on the Gaussian kernel (Gneiting and Raftery 2007).

The scoring rules listed above assess forecasts made for all outcomes. While this is clearly desirable when assessing overall forecast performance, it is often the case that certain outcomes are of more interest than others. For example, one could argue that it is particularly important to issue accurate forecasts for outcomes that have a high impact on the forecast users. To emphasise particular outcomes during forecast evaluation, weighted scoring rules generalise conventional scoring rules by incorporating a weight function into the score. The weight function can be chosen such that a higher weight is assigned to outcomes that are of more interest. Weighted scoring rules therefore allow competing forecast systems to be ranked and compared when predicting particular outcomes, facilitating very flexible, user-oriented forecast evaluation.

Well known examples of weighted scoring rules include the conditional and censored likelihood scores proposed by Diks *et al.* (2011), and the threshold-weighted CRPS introduced by Matheson and Winkler (1976) and Gneiting and Ranjan (2011). However, the theory underlying weighted scoring rules extends beyond these examples: Holzmann and Klar (2017) demonstrate that the conditional and censored likelihood scores can be generalised to construct weighted versions of any proper scoring rule, while Allen *et al.* (2023b) introduce a broad generalisation of the threshold-weighted CRPS that can be applied, for example, to probabilistic forecasts for multivariate outcomes.

In this paper, we describe how the **scoringRules** package has been extended to additionally permit the implementation of popular weighted scoring rules. While several alternative software packages exist to calculate particular scoring rules in certain situations (see Jordan *et al.* 2019, for an overview), the development of weighted scoring rules is more recent. Until recently, for example, efficient application of popular weighted scoring rules was limited by theoretical considerations, leading to ad hoc implementations in practice (Sharpe *et al.* 2018). Hence, to our knowledge, no other packages exist that provide a comprehensive collection of weighted scoring rules. We therefore hope that this extension to **scoringRules** will greatly facilitate the successful implementation of weighted scoring rules in practical applications.

In the following section, we review the existing theory of weighted scoring rules, and introduce examples of weighted versions of several popular scores, such as the LogS, the CRPS, and the energy score. The remainder of the paper then illustrates how these weighted scoring rules can be implemented in practice using the **scoringRules** package. Section 3 outlines the functionality of the package when calculating weighted scoring rules, and discusses implementation options; functionality is currently available for probabilistic forecasts in the form of predictive

samples, for which the weighted scoring rules are easy to implement with arbitrary weight functions. Section 4 then presents two case studies in which these weighted scoring rules are used to target particular outcomes when evaluating probabilistic forecasts in practice. These case studies include applications in weather forecasting and economic forecasting, building on the examples presented in Jordan *et al.* (2019). The paper is summarised in Section 5.

2. Theoretical background

2.1. Proper scoring rules

Suppose we are interested in predicting a random variable Y that takes values in a set Ω, and that our forecasts are in the form of probability distributions over Ω. Let $\mathcal{F}$ denote a set of such forecasts. A scoring rule is a function

$$S : \mathcal{F} \times \Omega \to \mathbb{R} \cup \{-\infty, \infty\},$$

which takes a forecast $F \in \mathcal{F}$ and an observation $y \in \Omega$ as inputs, and outputs a numerical value, or score, that quantifies the forecast accuracy. A lower score is assigned to a more accurate forecast. A scoring rule is proper with respect to $\mathcal{F}$ if, when the observations are drawn from a distribution $G \in \mathcal{F}$, the scoring rule is minimised in expectation by issuing G as the forecast, i.e.

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

for all $F, G \in \mathcal{F}$. If the above inequality is strict, then S is strictly proper with respect to $\mathcal{F}$.

Proper scoring rules allow two forecasters to be compared by the average score assigned to them over a set of forecast cases. Statistical hypothesis tests, such as a t-test or Diebold-Mariano test (Diebold and Mariano 1995), can then be employed to check whether the difference in two mean scores is significantly different from zero, which would suggest that one forecast significantly outperforms the other. However, the results of the comparison may change depending on what scoring rule is used to evaluate forecast performance. Different scoring rules assess different aspects of probabilistic forecast performance, and it is therefore important that the chosen scoring rule(s) reflect the subjective preferences of the forecast users.

When the outcome variable is real-valued ($\Omega \subseteq \mathbb{R}$), probabilistic forecasts are typically evaluated using either the logarithmic score (LogS) or the continuous ranked probability score (CRPS). The LogS is defined as

$$\text{LogS}(F, y) = -\log f(y), \tag{1}$$

where f is the predictive density associated with the cumulative distribution function F (Good 1952). The CRPS is defined as

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

where $\mathbf{1}$ is the indicator function, $X, X' \sim F$ are independent random variables, and it is assumed in the second expression that F has a finite mean (Matheson and Winkler 1976; Gneiting and Raftery 2007).

Generalisations of the LogS and the CRPS are also commonly used to evaluate probabilistic forecasts for multivariate outcomes, i.e. $\Omega \subseteq \mathbb{R}^d$ for $d > 1$. While the LogS in Equation 1 can readily be applied to multivariate predictive densities, it is often the case that only a sample from the multivariate forecast distribution is available, making it difficult to employ the LogS in practice. Instead, alternative scoring rules have been proposed to evaluate multivariate probabilistic forecasts that can readily be applied to samples from a forecast distribution.

Arguably the most well known multivariate scoring rule is the energy score (ES; Gneiting and Raftery 2007), which generalises the CRPS to higher dimensions:

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

where $\|\cdot\|$ is the Euclidean distance in $\mathbb{R}^d$, $\mathbf{y} = (y_1, \ldots, y_d) \in \Omega$, and $\mathbf{X} = (X_1, \ldots, X_d)$, $\mathbf{X}' = (X'_1, \ldots, X'_d) \sim F$ are independent, with F a probability distribution on Ω. It is assumed here and throughout that the expectations are finite where necessary.

An alternative to the energy score is the variogram score (VS; Scheuerer and Hamill 2015). The variogram score aims to explicitly assess the dependence structure of the multivariate forecast distributions by measuring the distance between the variogram of the forecast and that of the observation. The variogram score of order $p > 0$ is defined as

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

where $\mathbf{X} = (X_1, \ldots, X_d) \sim F$, $\mathbf{y} = (y_1, \ldots, y_d) \in \Omega$, and $h_{i,j}$ are non-negative scaling parameters that control how much emphasis is given to a pair of dimensions. Following recommendations from Scheuerer and Hamill (2015), the order of the score, p, is often chosen to be 0.5.

Both the energy score and the variogram score belong to the very general class of kernel scores (Gneiting and Raftery 2007). Kernel scores are scoring rules that are constructed using conditionally negative definite kernels, and the kernel score framework has also been leveraged to introduce alternative multivariate scoring rules. Allen *et al.* (2023b), for example, introduced a multivariate scoring rule based on the inverse multiquadric kernel, while the so-called maximum mean discrepancy score (MMDS) is the kernel score corresponding to the Gaussian kernel:

$$\mathrm{MMDS}(F, \mathbf{y}) = \frac{1}{2}\mathsf{E}_F \left[\exp\left\{-\frac{1}{2}\|\mathbf{X} - \mathbf{X}'\|^2\right\}\right] - \mathsf{E}_F \left[\exp\left\{-\frac{1}{2}\|\mathbf{X} - \mathbf{y}\|^2\right\}\right], \tag{5}$$

where $\mathbf{X}, \mathbf{X}' \sim F$ are independent.

To facilitate the implementation of these popular scoring rules in practice, the **scoringRules** package provides analytical expressions of the LogS and CRPS for forecasts that correspond to several familiar parametric distributions. It is also often the case that only a sample from the forecast distribution is available; this is common, for example, when considering ensemble forecasts issued by numerical weather and climate models, or output from Markov chain Monte Carlo (MCMC) algorithms (Krüger *et al.* 2021). The **scoringRules** package therefore

additionally contains versions of the LogS, CRPS, ES, VS, and MMDS that can be used to evaluate forecasts in the form of a predictive sample. This can be achieved by replacing the expectations in Equations 2-5 with sample means (see Appendix A for details). For the LogS, kernel density estimation is used to estimate the predictive density from the sample, prior to calculating the score.

2.2. Weighted scoring rules

The scoring rules introduced in the previous section evaluate the entire forecast distribution. However, one could argue that it is particularly important to issue accurate forecasts for events that have a high impact on the forecast users, and such events should therefore be given more weight during forecast evaluation. Weighted scoring rules achieve this by incorporating a non-negative weight function w into conventional scoring rules, where the weight function determines how much emphasis should be placed on each possible outcome. Different approaches to weight scoring rules exist, and here we focus only on the two most popular frameworks.

Outcome-weighted scoring rules

Diks *et al.* (2011) introduced two weighted versions of the LogS that allow particular outcomes to be emphasised when calculating forecast accuracy. The conditional likelihood score (CoLS) is defined as

$$\text{CoLS}(F, y) = -w(y)\log f(y) + w(y)\log\left(\int_{\mathbb{R}} w(z)f(z)\,\mathrm{d}z\right),$$

while the censored likelihood score (CeLS) is

$$\text{CeLS}(F, y) = -w(y)\log f(y) - (1 - w(y))\log\left(1 - \int_{\mathbb{R}} w(z)f(z)\,\mathrm{d}z\right).$$

To understand how these weighted logarithmic scores behave, consider a weight function of the form $w(z) = \mathbf{1}\{z \in \mathcal{A}\}$, meaning only forecasts for outcomes in the set $\mathcal{A} \subseteq \Omega$ are of interest. In this example, if the observation $y \notin \mathcal{A}$, then the CoLS is equal to zero. If $y \in \mathcal{A}$, then the CoLS is equivalent to the LogS applied to the conditional forecast distribution given that the observation is in $\mathcal{A}$; forecast distributions are therefore assessed only via their restriction to the set $\mathcal{A}$. The CeLS then extends the CoLS by additionally rewarding forecast distributions that can correctly predict when an outcome of interest will or will not occur. Note that if $\mathcal{A} = \Omega$, then the weight function is always one, and both weighted scoring rules revert to the unweighted LogS.

Holzmann and Klar (2017) later generalised the CoLS and CeLS by demonstrating that this framework can readily be applied to any proper scoring rule. The resulting scoring rules, which we call outcome-weighted scoring rules, target particular outcomes by introducing a weighted version of the forecast distribution, and evaluating F via its weighted representation. For the weight function $w(z) = \mathbf{1}\{z \in \mathcal{A}\}$, this weighted representation is simply the conditional distribution given that the outcome is in $\mathcal{A}$, as discussed above for the CoLS and CeLS. Further details can be found in Holzmann and Klar (2017).

An outcome-weighted CRPS can be defined as

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

where $X, X' \sim F$ are independent and $\bar{w}_F = \mathsf{E}_F[w(X)]$. Since this framework applies to any proper scoring rule, outcome-weighted versions of the ES, VS, and MMDS can similarly be introduced to target multivariate outcomes of interest during forecast evaluation:

$$\begin{aligned}
\text{owES}(F, \mathbf{y}) =& \frac{1}{\bar{w}_F} \mathsf{E}_F \left[\|\mathbf{X} - \mathbf{y}\| w(\mathbf{X}) w(\mathbf{y}) \right] - \frac{1}{2\bar{w}_F^2} \mathsf{E}_F \left[\|\mathbf{X} - \mathbf{X}'\| w(\mathbf{X}) w(\mathbf{X}') w(\mathbf{y}) \right]; \\
\text{owVS}^p(F, \mathbf{y}) =& w(\mathbf{y}) \sum_{i=1}^{d} \sum_{j=1}^{d} h_{i,j} \left(\frac{1}{\bar{w}_F} \mathsf{E}_F \left[|X_i - X_j|^p \, w(\mathbf{X}) \right] - |y_i - y_j|^p \right)^2; \\
\text{owMMDS}(F, \mathbf{y}) =& \frac{1}{2\bar{w}_F^2} \mathsf{E}_F \left[\exp\left\{ -\frac{1}{2} \|\mathbf{X} - \mathbf{X}'\|^2 \right\} w(\mathbf{X}) w(\mathbf{X}') w(\mathbf{y}) \right] \\
& - \frac{1}{\bar{w}_F} \mathsf{E}_F \left[\exp\left\{ -\frac{1}{2} \|\mathbf{X} - \mathbf{y}\|^2 \right\} w(\mathbf{X}) w(\mathbf{y}) \right].
\end{aligned}$$

Note that in the multivariate case, the weight function takes a vector as an argument; $\bar{w}_F$ is thus defined as $\bar{w}_F = \mathsf{E}_F[w(\mathbf{X})]$.

The premise behind this class of weighted scoring rules is that, if attention is only on a particular set of outcomes, then the forecasts are only evaluated when these outcomes occur. When these outcomes do occur, the forecast distributions are evaluated using the conditional distribution given that the outcome of interest has occurred. In considering the conditional distribution given that an outcome of interest has occurred, the score does not consider the predicted probability that this outcome will occur. The CeLS extends the CoLS to address this, and suitable adaptations of the larger class of outcome-weighted scoring rules also exist, though these are not considered here (see Holzmann and Klar 2017).

Moreover, these scores are clearly not well-defined if the conditional distribution does not exist. This is equivalent to $\bar{w}_F$ being equal to zero, which could occur, for example, if $w(z) = \mathbf{1}\{z \in \mathcal{A}\}$ and the forecast distribution assigns zero probability to the region $\mathcal{A}$. The use of these outcome-weighted scoring rules is therefore only recommended when the weight function is strictly positive, or when interest is on events that are not rare, such that $\bar{w}_F$ is non-zero (Allen *et al.* 2023a).

Threshold-weighted scoring rules

Arguably the most well known weighted scoring rule is the threshold-weighted CRPS proposed by Matheson and Winkler (1976) and Gneiting and Ranjan (2011). The threshold-weighted CRPS introduces a weight function into the integral defining the CRPS:

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

where v is any function such that $v(z) - v(z') = \int_{z'}^{z} w(z) \, \mathrm{d}z$ for all $z, z' \in \mathbb{R}$ (Taillardat *et al.* 2022; Allen *et al.* 2023b). We follow Allen *et al.* (2023b) and refer to v as a chaining function.

Just as we can generate outcome-weighted versions of any proper scoring rule, Allen *et al.* (2023b) demonstrate that the theory underlying the threshold-weighted CRPS can readily be extended to any kernel score. As discussed, the ES, VS, and MMDS are all kernel scores,

allowing threshold-weighted versions of these scores to be introduced:

$$\begin{aligned}
\text{twES}(F, \mathbf{y}) =& \mathsf{E}_F \|v(\mathbf{X}) - v(\mathbf{y})\| - \frac{1}{2}\mathsf{E}_F\|v(\mathbf{X}) - v(\mathbf{X}')\|; \\
\text{twVS}^p(F, \mathbf{y}) =& \sum_{i=1}^{d}\sum_{j=1}^{d} h_{i,j} \left(\mathsf{E}_F \left|v(\mathbf{X})_i - v(\mathbf{X})_j\right|^p - \left|v(\mathbf{y})_i - v(\mathbf{y})_j\right|^p\right)^2; \\
\text{twMMDS}(F, \mathbf{y}) =& \frac{1}{2}\mathsf{E}_F\left[\exp\left\{-\frac{1}{2}\|v(\mathbf{X}) - v(\mathbf{X}')\|^2\right\}\right] - \mathsf{E}_F\left[\exp\left\{-\frac{1}{2}\|v(\mathbf{X}) - v(\mathbf{y})\|^2\right\}\right],
\end{aligned} \tag{6}$$

where $\mathbf{X}, \mathbf{X}' \sim F$ are independent random variables taking values on $\Omega \subseteq \mathbb{R}^d$, and $v : \mathbb{R}^d \to \mathbb{R}^d$ is a chaining function, so that $v(\mathbf{y}) = (v(\mathbf{y})_1, \dots, v(\mathbf{y})_d)$ and likewise for $v(\mathbf{X})$ and $v(\mathbf{X}')$.

In contrast to the outcome-weighted scoring rules, threshold-weighted scoring rules transform the forecasts and observations according to a chaining function v prior to employing the unweighted version of the scores. The chaining function can therefore be chosen to focus the scoring rules on particular outcomes. While there exists a canonical way to obtain a chaining function from a given weight function in the univariate case, no such relationship exists when evaluating multivariate forecasts. This is discussed further in the following section.

2.3. Weight and chaining functions

These weighted scoring rules provide attractive ways to target particular outcomes of interest when evaluating forecast performance, both in the univariate and multivariate case. In this section, we discuss possible weight and chaining functions that can be used within these weighted scoring rules. Certain choices can result in weighted scoring rules that are not proper, and these weight and chaining functions must therefore be chosen with care, to ensure that forecasters are not evaluated using an improper scoring rule.

If both a weighted and unweighted version of a scoring rule are proper, then they will both be minimised on average by the same forecast distribution: the true distribution of the outcome. However, for two imperfect forecasts, the ranking of these forecasts may change depending on whether a weighted or unweighted scoring rule is employed. Weighted scoring rules may be less powerful than conventional scoring rules when discriminating between two forecast distributions, but they should be more discriminative when comparing forecasts made for particular outcomes. Put differently, if weighted scoring rules detect a difference between two forecast systems, then it is generally easier to interpret this difference than if it were detected using an unweighted scoring rule.

Readers are referred to Gneiting and Ranjan (2011), Lerch *et al.* (2017), and Allen *et al.* (2023b) for further details regarding what weight and chaining functions preserve the (strict) propriety of scoring rules. The weight and chaining functions that we consider here all result in weighted scoring rules that are themselves proper (though not necessarily strictly proper).

Weight functions

The choice of weight and chaining function is case-specific, and should depend on what information is to be extracted from the forecasts. Most commonly, interest is on outcomes within a certain range, or above or below a predefined threshold; this range or threshold may correspond to relevant quantiles of the previously observed outcomes, for example. A univariate

weight function that restricts attention to these events is

$$w(z) = \mathbf{1}\{a < z < b\} \quad \text{for some} \quad -\infty \leq a < b \leq \infty, \tag{7}$$

which is one if z is between a and b, and zero otherwise. To emphasise values above (below) some threshold t, we can set $a = t$ and $b = \infty$ ($a = -\infty$ and $b = t$).

Alternatively, certain events could be emphasised using a smoother weight function, which assigns a positive weight to all outcomes, but a higher weight to the events of interest. Popular weight functions to emphasise rare events include a Gaussian or logistic distribution function, e.g.

$$w(z) = \Phi_{\mu,\sigma}(z), \tag{8}$$

where $\Phi_{\mu,\sigma}$ is the Gaussian distribution function with mean μ and standard deviation σ, with these parameters controlling the location of the weight function and the rate at which it tends to zero and one (Gneiting and Ranjan 2011). The Gaussian survival function $1-\Phi_{\mu,\sigma}(z)$ could analogously emphasise low values of z.

Gaussian and logistic density functions could additionally be used to target outcomes that are not rare. For example, the weight function

$$w(z) = \phi_{\mu,\sigma}(z),$$

where $\phi_{\mu,\sigma}$ is the Gaussian density function with mean μ and standard deviation σ. This weight function will emphasise values around the location parameter μ, with σ determining the concentration of the weight around μ.

Similar weight functions can also be used in the multivariate case. For example, it is common to define rare multivariate events as threshold exceedances that occur simultaneously in multiple dimensions, in which case a canonical weight function is

$$w(\mathbf{z}) = \mathbf{1}\{a_1 < z_1 < b_1, \ldots, a_d < z_d < b_d\} \quad \text{for} \quad -\infty \leq a_i < b_i \leq \infty, \quad i = 1, \ldots, d. \tag{9}$$

As in the univariate case, some values of the vectors $\mathbf{a} = (a_1, \ldots, a_d)$ and $\mathbf{b} = (b_1, \ldots, b_d)$ can be set to $\pm\infty$ in order to focus on threshold exceedances. Multivariate Gaussian distribution and density functions could then again be used to target particular regions of multivariate space in a smoother way (Allen *et al.* 2023a). These weight functions are listed in Table 1.

Chaining functions

While the outcome-weighted scoring rules depend on a weight function, the threshold-weighted scoring rules depend on a chaining function. It is arguably less intuitive to choose a chaining function to emphasise certain outcomes of interest than a weight function. In the univariate case, the chaining function can be derived easily from a given weight function: we can take any function v that satisfies

$$v(z) - v(z') = \int_{z'}^{z} w(z)\, \mathrm{d}z \quad \text{for all} \quad z, z' \in \mathbb{R}. \tag{10}$$

That is, v is an anti-derivative of the chosen weight function. Table 1 lists examples of chaining functions that correspond to the univariate weight functions given above.

In the multivariate case, however, there is no canonical approach to derive a chaining function from a given weight function. Allen *et al.* (2023b) discuss possible chaining functions that

Weight function	Chaining function
$w(z) = 1$	$v(z) = z$
$w(z) = \mathbf{1}\{a < z < b\}$	$v(z) = \min(\max(z, a), b)$
$w(z) = \Phi_{\mu,\sigma}(z)$	$v(z) = (z-\mu)\Phi_{\mu,\sigma}(z) + \sigma^2\phi_{\mu,\sigma}(z)$
$w(z) = 1 - \Phi_{\mu,\sigma}(z)$	$v(z) = z - (z-\mu)\Phi_{\mu,\sigma}(z) - \sigma^2\phi_{\mu,\sigma}(z)$
$w(z) = \phi_{\mu,\sigma}(z)$	$v(z) = \Phi_{\mu,\sigma}(z)$
$w(z) = (1 + \exp(-\frac{z-\mu}{\sigma}))^{-1}$	$v(z) = \sigma \log(1 + \exp(\frac{z-\mu}{\sigma}))$
$w(z) = 1 - (1 + \exp(-\frac{z-\mu}{\sigma}))^{-1}$	$v(z) = z - \sigma \log(1 + \exp(\frac{z-\mu}{\sigma}))$
$w(z) = \frac{1}{\sigma}\exp(-\frac{z-\mu}{\sigma})(1 + \exp(-\frac{z-\mu}{\sigma}))^{-2}$	$v(z) = (1 + \exp(-\frac{z-\mu}{\sigma}))^{-1}$
$w(\boldsymbol{z}) = \mathbf{1}\{a_1 < z_1 < b_1, \ldots, a_d < z_d < b_d\}$	$v(\boldsymbol{z})_i = \min(\max(z_i, a_i), b_i)$
$w(\boldsymbol{z}) = \boldsymbol{\Phi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{z})$	$v(\boldsymbol{z})_i = (z_i - \mu_i)\Phi_{\mu_i,\sigma_i}(z_i) + \sigma_i^2\phi_{\mu_i,\sigma_i}(z_i)$
$w(\boldsymbol{z}) = 1 - \boldsymbol{\Phi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{z})$	$v(\boldsymbol{z})_i = z_i - (z_i - \mu_i)\Phi_{\mu_i,\sigma_i}(z_i) - \sigma_i^2\phi_{\mu_i,\sigma_i}(z_i)$
$w(\boldsymbol{z}) = \boldsymbol{\phi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{z})$	$v(\boldsymbol{z})_i = \Phi_{\mu_i,\sigma_i}(z_i)$

Table 1: Examples of weight functions and chaining functions that could be used in weighted scoring rules. $\boldsymbol{\Phi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ and $\boldsymbol{\phi}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$ denote the multivariate Gaussian distribution and density functions with mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ and covariance matrix Σ with diagonal entries $\sigma_1, \ldots, \sigma_d$. The multivariate chaining functions are component-wise extensions of the univariate chaining functions; hence, for concision, only the i-th component (for $i \in \{1, \ldots, d\}$) of the multivariate chaining function is shown.

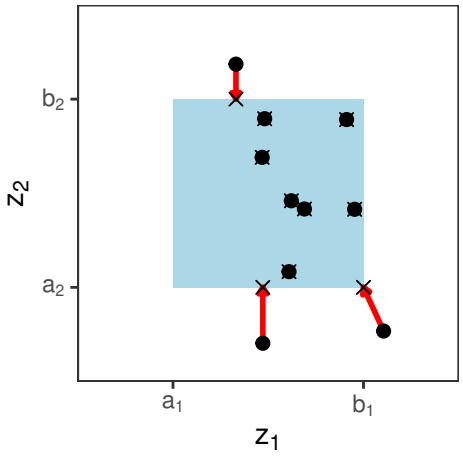

Figure 1: Example of the weight function in Equation 9 and the chaining function in Equation 11. Ten two-dimensional observations are shown before (points) and after (crosses) applying the chaining function. The shaded region is the area specified by the weight function. Points in this region are unchanged after applying the chaining function, whereas points outside of this region are mapped onto the region's border, as indicated by the red arrows.

could be used to target certain multivariate outcomes when interest is on high-impact events. For the multivariate weight function in Equation 9, one possible chaining function is

$$v(z) = (\min(\max(z_1, a_1), b_1), \ldots, \min(\max(z_d, a_d), b_d)) \,, \tag{11}$$

which is essentially a component-wise extension of the chaining function for the univariate weight in Equation 7 (see Table 1). In this case, the weight function represents an orthant, or a box, in $\mathbb{R}^d$, and the chaining function projects points not in the orthant onto its perimeter; the points inside the orthant, i.e. for which the weight function is equal to one, remain unchanged. A two-dimensional example of this is given in Figure 1.

Similarly, for the smooth weight functions based on multivariate Gaussian distribution and density functions, a chaining function can be derived from a component-wise extension of the chaining functions corresponding to univariate Gaussian weight functions. Examples of such chaining functions are presented in Table 1. Note, however, that these component-wise extensions implicitly assume that the covariance matrix in the multivariate Gaussian weight function is diagonal. Readers are referred to Allen *et al.* (2023b) for a more detailed discussion on multivariate chaining functions.

Although the weight and chaining functions presented in this section are simple examples that are frequently used in practice, the weighted scoring rules discussed herein can be employed with arbitrary such functions, permitting very flexible, user-oriented forecast evaluation. Several weight functions can be employed to evaluate forecasts with respect to different regions of the outcome space. However, it is not advisable to employ a weight function that changes adaptively on the forecast setting. Different weight functions lead to scores on different scales, and if the chosen weight function depends on the outcome, then the resulting scoring rule will generally be improper. As a result, the functions in the following sections do not allow for adaptive weight functions.

3. Package functionality

In the remainder of this paper, we will discuss how the weighted scoring rules introduced in the previous section have been integrated into the **scoringRules** package, facilitating their use in practical applications.

3.1. Univariate weighted scoring rules

The weighted scoring rules discussed in the previous section can all be implemented using the **scoringRules** package. Functionality is currently available for probabilistic forecasts that take the form of a predictive sample. In this case, it is straightforward to calculate the weighted scoring rules with arbitrary, user-specified weight functions, which is generally not the case for parametric families of distributions. Expressions for the weighted scoring rules discussed in the previous section when the forecast is a predictive sample are given in Appendix A.

The **scoringRules** package already contains functions to calculate the LogS, CRPS, ES, VS, and MMDS for forecasts in the form of predictive samples. Suppose the sample is comprised of m members. As explained in Jordan *et al.* (2019), the naming convention of these functions is `[score]_sample()`, where `[score]` refers to the scoring rule to be calculated. These functions take the observed value(s) and the forecast samples as inputs, and output the desired score value. For example, to calculate the CRPS corresponding to a vector of n observations `y` and a $n \times m$ matrix `dat` whose rows contain the m forecast samples corresponding to each observation, one could use

```
crps_sample(y, dat)
```

The output is a numeric vector containing the score for each of the n forecast cases.

The same convention is adopted for the weighted scoring rules. In the univariate case, the following functions calculate the outcome-weighted and threshold-weighted CRPS, and the conditional or censored likelihood scores:

```
owcrps_sample(y, dat, a = -Inf, b = Inf, weight_func = NULL,
              w = NULL, show_messages = TRUE)
twcrps_sample(y, dat, a = -Inf, b = Inf, chain_func = NULL,
              w = NULL, show_messages = TRUE)
clogs_sample(y, dat, a = -Inf, b = Inf, bw = NULL,
             show_messages = FALSE, cens = TRUE)
```

The `cens` argument in `clogs_sample()` specifies whether the conditional likelihood score or the censored likelihood score should be returned; the default is `cens = TRUE`, in which case the CeLS is calculated.

As discussed in Section 2.1, the LogS takes a predictive density as input, and hence cannot readily be applied to predictive samples. To circumvent this, `logs_sample()` employs kernel density estimation to estimate a predictive density from the sample, and then calculates the LogS from the estimated density function. However, Krüger *et al.* (2021) demonstrate that the resulting score is sensitive to the bandwidth parameter `bw` of the kernel density estimation, and the authors therefore recommended using the CRPS instead of the LogS, particularly when the sample size m is small. Similarly, the conditional and censored likelihood scores

also require a predictive density as inputs, and kernel density estimation is used to estimate this from the predictive sample prior to calculating the weighted scores. We anticipate that these weighted scores will be yet more sensitive to the kernel density estimation parameters, especially when a weight function is used that targets more extreme outcomes. As such, when the forecast is in the form of a predictive sample, we similarly recommend employing weighted versions of the CRPS, rather than the conditional or censored likelihood score.

In addition to observations and forecast samples, the functions listed above have arguments that allow particular outcomes to be targeted when calculating the weighted scores. By default, the weighted scoring rules employ the weight function $w(z) = \mathbf{1}\{a < z < b\}$, which, as discussed in the previous section, is most commonly applied in practice. The arguments `a` and `b` are single numeric values representing the lower and upper bounds in this weight function, respectively. If these arguments are not specified, then their default values are `a = -Inf` and `b = Inf`, resulting in a weight function that is always one, and thus recovering the unweighted scoring rules.

```
R> obs <- rnorm(5)
R> sample_m <- matrix(rnorm(5e4), nrow = 5)
R> score_df <- data.frame(crps = crps_sample(obs, sample_m),
+                         owcrps = owcrps_sample(obs, sample_m),
+                         twcrps = twcrps_sample(obs, sample_m))
R> print(score_df)

   crps owcrps twcrps
1 0.275  0.275  0.275
2 1.230  1.230  1.230
3 0.246  0.246  0.246
4 0.764  0.764  0.764
5 1.355  1.355  1.355
```

On the other hand, if we want to emphasise outcomes above a threshold `t`, then we can set the lower bound in the weight function to `a = t`, and the upper bound to `b = Inf`.

```
R> t <- 0
R> score_df <- data.frame(crps = crps_sample(obs, sample_m),
+                         owcrps = owcrps_sample(obs, sample_m, a = t),
+                         twcrps = twcrps_sample(obs, sample_m, a = t))
R> print(score_df)

   crps owcrps twcrps
1 0.275  0.000  0.120
2 1.230  0.000  0.115
3 0.246  0.000  0.119
4 0.764  0.306  0.645
5 1.355  0.809  1.235
```

Similarly, if we want to emphasise values below the threshold, then we can set `a = -Inf` and `b = t`. To avoid misuse, an error is returned if `a` is not smaller than `b`.

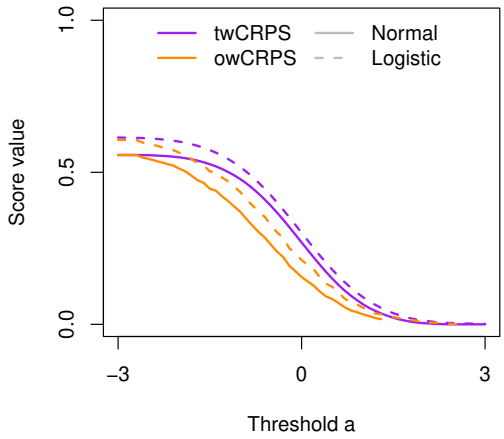

Figure 2: Average twCRPS (purple) and owCRPS (orange) as a function of the threshold a in the weight function $w(z) = \mathbf{1}\{z > a\}$. The observations are drawn from a standard normal distribution, and scores are shown for predictive samples from a standard normal (solid) and standard logistic (dashed) distribution. Note that for high thresholds, the owCRPS is not always well-defined.

A useful diagnostic tool is to plot the average score as a function of the threshold. In this case, as the lower bound in the weight function `a` becomes smaller (or the upper bound `b` becomes larger), the weighted score tends to the unweighted score, allowing the user to simultaneously visualise overall forecast performance, as well as performance when predicting particular outcomes (Gneiting and Ranjan 2011). An example of this is presented in Figure 2, where the outcome-weighted CRPS and threshold-weighted CRPS for two forecasts distributions are displayed as a function of `a` in the default weight function, with `b = Inf`.

3.2. Multivariate weighted scoring rules

Similarly to the LogS and CRPS, **scoringRules** contains functions to calculate the ES, VS, and MMDS for multivariate forecast distributions in the form of predictive samples.

```
es_sample(y, dat, w = NULL)
vs_sample(y, dat, w = NULL, w_vs = NULL, p = 0.5)
mmds_sample(y, dat, w = NULL)
```

These multivariate scoring rule functions can only evaluate a single multivariate forecast at a time. Hence, the observation argument `y` is a vector of length d, representing an element in $\mathbb{R}^d$, the forecast argument `dat` is a $d \times m$ matrix, with the columns representing the simulated samples (or ensemble members) from the multivariate forecast distribution, and the output is a single value. These functions can then be sequentially applied to multiple forecast cases using the `apply()` functions or `for` loops (see Appendix B of Jordan *et al.* 2019).

Similarly, outcome-weighted and threshold-weighted versions of these multivariate scoring rules are calculated using

```
owes_sample(y, dat, a = -Inf, b = Inf, weight_func = NULL, w = NULL)
owvs_sample(y, dat, a = -Inf, b = Inf, weight_func = NULL,
```

```
            w = NULL, w_vs = NULL, p = 0.5)
owmmds_sample(y, dat, a = -Inf, b = Inf, weight_func = NULL, w = NULL)
twes_sample(y, dat, a = -Inf, b = Inf, chain_func = NULL, w = NULL)
twvs_sample(y, dat, a = -Inf, b = Inf, chain_func = NULL,
            w = NULL, w_vs = NULL, p = 0.5)
twmmds_sample(y, dat, a = -Inf, b = Inf, chain_func = NULL, w = NULL)
```

As in the univariate case, the default weight function corresponds to Equation 9, where interest is on a range of values in each dimension. The default chaining function used to calculate the threshold-weighted scores is Equation 11. Arguments `a` and `b` are again used to define the lower and upper bounds of the default weight function. In contrast to the univariate case, however, `a` and `b` are numeric vectors of length d, rather than single values.

If the input value of `a` or `b` is a single value, then it is automatically converted into a vector of length d, all containing the same element. The default values are `a = -Inf` and `b = Inf`, which again returns the unweighted scoring rule. Hence, if we want to emphasise values above the same threshold t in all dimensions, then we could either use `a = c(t, t, ...)` and `b = c(Inf, Inf, ...)`, or we could use `a = t` and `b = Inf`. For example, for the threshold-weighted energy score, we have

```
R> d <- length(obs)
R> twes_sample(obs, sample_m, a = t)

[1] 1.34

R> twes_sample(obs, sample_m, a = rep(t, d))

[1] 1.34
```

Finally, note that the functions to calculate the multivariate weighted scores also include optional weight arguments that cannot be used to target particular outcomes of interest. The argument `w` is a vector of length m that allows more weight to be given to particular elements of the sample in the forecast distribution. This argument is also available when calculating the unweighted scoring rules, and the univariate weighted scores. The variogram score functions additionally have an argument `w_vs`, which is a $d \times d$ matrix containing the scaling parameters $h_{i,j}$ in Equation 4. These scaling parameters put more emphasis on combinations of dimensions of the multivariate variables, rather than targeting particular outcomes.

3.3. Custom weight and chaining functions

The functions to calculate the weighted scoring rules use a default weight function that assumes emphasis is to be placed on a particular region of the outcome space. Although this weight function is frequently applied in practice, it may be the case that another weight function is desired. As discussed, the motivation for considering only forecasts in the form of a predictive sample is that it is straightforward to calculate the resulting scores for arbitrary weight and chaining functions. The weighted scoring rule functions in **scoringRules** therefore

additionally contain an argument that allows for a custom weight or chaining function to be used.

The `weight_func` argument can be used to incorporate a custom weight function into the outcome-weighted scoring rules. This argument must be a function that takes a vector as an input, and outputs either a vector of the same length as the input (if a univariate scoring rule is being used), or a single numeric value (if a multivariate scoring rule is used). An error is returned if the weight function is found to return negative weights, or if the output is not of the correct format.

For example, consider the Gaussian distribution function in Equation 8, with location parameter `mu` and scale parameter `sigma`. To use this as the weight function when calculating the outcome-weighted CRPS, one could use

```
R> mu <- 0; sigma <- 1
R> weight_func <- function(x) pnorm(x, mean = mu, sd = sigma)
R> owcrps_sample(obs, sample_m, weight_func = weight_func)

[1] 0.2002 0.0703 0.1868 0.3524 0.8788
```

Similarly, a multivariate Gaussian distribution could be used as a multivariate weight function. Let `mu` be the mean vector of this distribution, and assume the covariance matrix is diagonal with entries $\sigma_1, \ldots, \sigma_d$. Then, the outcome-weighted ES with this weight function can be calculated using

```
R> mu <- rnorm(d, 0, 0.5); sigma <- runif(d, 0.5, 1.5)
R> weight_func <- function(x) prod(pnorm(x, mean = mu, sd = sigma))
R> owes_sample(obs, sample_m, weight_func = weight_func)

[1] 0.0418
```

Since weight functions based on Gaussian and logistic distributions are also commonly employed in practice, **scoringRules** additionally exports a function `get_weight_func()` that can be used to obtain R functions corresponding to the weight and chaining functions listed in Table 1.

```
get_weight_func(name = "norm_cdf", mu = 0, sigma = 1, weight = TRUE)
```

The `name` argument specifies the desired weight or chaining function. This must be one of `'norm_cdf'`, `'norm_pdf'`, `'norm_surv'`, `'logis_cdf'`, `'logis_pdf'` and `'logis_surv'`, corresponding to the cumulative distribution function, probability density function, and survival function of the Gaussian and logistic distribution, respectively. `mu` and `sigma` correspond to the location and scale parameters of the Gaussian or logistic distribution, which are single numeric values in the univariate case, and numeric vectors (of the same length) in the multivariate case. In the multivariate setting, `mu` represents the mean vector of the multivariate distribution, and `sigma` the diagonal elements of the covariance matrix; functionality is currently only available for multivariate weight and chaining functions corresponding to the multivariate normal distribution with a diagonal covariance matrix. This also means `name` must be one of `'norm_cdf'`, `'norm_pdf'` and `'norm_surv'` in the multivariate case.

The above examples can be simplified using `get_weight_func()`.

```
R> weight_func <- get_weight_func(name = "norm_cdf", mu = 0, sigma = 1)
R> owcrps_sample(obs, sample_m, weight_func = weight_func)

[1] 0.2002 0.0703 0.1868 0.3524 0.8788
```

and

```
R> weight_func <- get_weight_func(name = "norm_cdf", mu = mu, sigma = sigma)
R> owes_sample(obs, sample_m, weight_func = weight_func)

[1] 0.0418
```

Whereas the outcome-weighted scores depend on a weight function, the threshold-weighted scores rely on a chaining function. For the threshold-weighted CRPS, a chaining function corresponds directly to a weight function via Equation 10. However, computation of the threshold-weighted CRPS for a sample forecast requires the chaining function rather than a weight function, and hence functionality is not currently available to take a weight function as an argument. In this case, it is necessary to derive the chaining function corresponding to the weight. For the simple weight functions commonly used in practice, this is typically straightforward to achieve (see Table 1 for popular choices).

The `chain_func` argument can be used to incorporate a custom chaining function into the threshold-weighted scoring rules. In contrast to `weight_func`, the `chain_func` argument should be a function whose inputs and outputs are the same length as the observation input `y`. For example, in the multivariate case, this function should both input and output a vector of length d.

In the univariate case, if the chaining function satisfies Equation 10 for some non-negative weight function w, then it will be a non-decreasing function; that is, if $z > z'$, then $v(z) \geq v(z')$ for all $z, z' \in \mathbb{R}$. While a decreasing chaining function could also be used within Equation 2, this does not correspond to the original definition of the twCRPS presented in Gneiting and Ranjan (2011), and is therefore not recommended: a warning message is returned if `chain_func` is found to be decreasing.

Table 1 contains possible chaining functions corresponding to the Gaussian weight functions employed above. These chaining functions can be implemented within `twcrps_sample` and `twes_sample` as follows

```
R> chain_func <- function(x) (x - mu)*pnorm(x, mu, sigma) +
+    (sigma^2)*dnorm(x, mu, sigma)
R> mu <- 0; sigma <- 1
R> twcrps_sample(obs, sample_m, chain_func = chain_func)

[1] 0.135 0.263 0.123 0.528 1.082

R> mu <- rnorm(d, 0, 0.5); sigma <- runif(d, 0.5, 1.5)
R> twes_sample(obs, sample_m, chain_func = chain_func)

[1] 1.48
```

Weighted versions of the other scoring rules discussed herein can be calculated analogously, and these examples can again be simplified using `get_weight_func()`.

```
R> chain_func <- get_weight_func("norm_cdf", mu = 0, sigma = 1,
+                                weight = FALSE)
R> twcrps_sample(obs, sample_m, chain_func = chain_func)

[1] 0.135 0.263 0.123 0.528 1.082

R> chain_func <- get_weight_func("norm_cdf", mu = mu, sigma = sigma,
+                                weight = FALSE)
R> twes_sample(obs, sample_m, chain_func = chain_func)

[1] 1.48
```

The argument `weight = FALSE` specifies that a chaining function should be returned instead of a weight function; the default (`weight = TRUE`) is to return the weight function.

It is challenging to construct general analytical formulae for weighted scoring rules corresponding to parametric forecast distributions and arbitrary weight functions. Similarly, since the CoLS and CeLS require kernel density estimation to estimate the predictive density given the sample, these scores cannot be readily implemented with arbitrary weight functions. Hence, `clogs_samples` does not take custom weight or chaining functions as arguments, so that only the default weight function is available for these scores.

4. Usage examples

Jordan *et al.* (2019) present two practical applications in which the **scoringRules** functionality is used to evaluate probabilistic forecasts. In this section, we revisit these applications, and illustrate how the weighted scoring rules available in **scoringRules** allow particular outcomes to be targeted during forecast evaluation. In both examples, the data and probabilistic models are as described in Jordan *et al.* (2019), and further details can be found therein.

4.1. Probabilistic weather forecasting via ensemble post-processing

Firstly, consider forecasts of precipitation accumulation in Innsbruck, Austria. The `RainIbk` data set in the **crch** R package (Messner *et al.* 2016) contains three-day precipitation accumulations recorded in Innsbruck from January 2000 to September 2013. Forecasts for these precipitation accumulations can be obtained from numerical weather prediction (NWP) models, which use physical laws to emulate the evolution of the atmosphere through time. These models are typically run several times, using different initial conditions and possibly different model configurations, yielding an ensemble of predictions that characterises the forecast uncertainty. The `RainIbk` data set contains 11-member ensemble forecasts corresponding to the three-day precipitation accumulations between five and eight days in advance.

However, operational ensemble forecasts tend to exhibit systematic errors when predicting surface weather variables such as precipitation. Hence, it is common for the ensemble forecasts to undergo some form of statistical post-processing. Post-processing methods try to

learn the systematic errors that manifest in the NWP models, and then remove them from the forecasts. While a number of statistical post-processing methods have been proposed, the methods considered here assume that the square root of the precipitation accumulation follows a parametric distribution. The location and scale parameters of this distribution are assumed to depend linearly on the mean and the log-transformed standard deviation of the ensemble members, respectively. This general framework for post-processing ensemble weather forecasts is typically known as non-homogeneous regression or ensemble model output statistics (EMOS; Gneiting *et al.* 2005).

Three alternative parametric distributions are then compared within this framework: a logistic, Gaussian, and Student's t distribution. These three parametric distributions are censored below at zero, resulting in a forecast distribution that assigns zero probability to negative precipitation accumulations, and a non-negligible probability to zero precipitation. Further details about the statistical post-processing methods are available in Jordan *et al.* (2019) and references therein. In the following code chunks, the logical variable `use_crch` indicates whether the **crch** package has been installed, which is a precondition for running the code.

Data from January 2000 to November 2004 is used to train the statistical post-processing models, and the resulting forecasts are then evaluated out-of-sample using the data from January 2005 to September 2013. The models are fit to the training data using maximum likelihood estimation via the **crch** package. Applying these models to the ensemble forecasts in the test data set returns predictive location and scale parameters for each model and each forecast case. For concision, we only show the code used to evaluate the Gaussian forecast distributions; this can easily be extended to the other models. In this case, the vectors `gauss_mu` and `gauss_sc` contain the estimated location and scale parameters that characterise the predictive distributions in the test data, while `obs` represents the time series of the corresponding observed precipitation accumulations.

These three post-processing models can then be evaluated and compared using scoring rules. Jordan *et al.* (2019) demonstrate how the CRPS can be used for this purpose.

```
R> if (use_crch){
+    gauss_crps <- crps_cnorm(y = obs, location = gauss_mu, scale = gauss_sc,
+                             lower = 0, upper = Inf)
+  }
```

The result is a vector of scores corresponding to each forecast case in the test data set, and the competing forecast strategies can then be compared using their average scores.

```
R> if (use_crch){
+    scores <- data.frame(Logistic = logis_crps, Gaussian = gauss_crps,
+                         Students_t = stud_crps, Ensemble = ens_crps)
+    sapply(scores, mean)
+  }

  Logistic   Gaussian Students_t   Ensemble
     0.875      0.876      0.875      1.321
```

The mean CRPS values indicate that all post-processing models substantially improve upon the raw ensemble forecasts, and there are only small differences between the post-processing

models. Of course, in a formal study, we should accompany these scores with measures of uncertainty, or perform statistical tests that clarify whether the differences between the scores are significant.

While the CRPS assesses overall forecast performance, the three post-processing models could also be compared with respect to their predictions of particular outcomes. This can be achieved here using weighted versions of the CRPS. As discussed, weighted scoring rules are only available in **scoringRules** for forecasts in the form of a predictive sample. Hence, to evaluate the post-processed forecast distributions, we must first sample from the predictive distributions, and use this sample to estimate the score for the parametric forecasts. Here, we sample 1000 observations from the predictive distributions, which should provide a reasonable approximation of the score for the continuous forecast distribution (Jordan *et al.* 2019, Figure 2).

```
R> if (use_crch){
+    ens_size <- 1000
+    n <- length(obs)
+    gauss_sample <- replicate(ens_size, rnorm(n, gauss_mu, gauss_sc))
+    gauss_sample[gauss_sample < 0] <- 0
+  }
```

An obvious question concerns which weight function to employ within the weighted scores. Weight and chaining functions should be chosen such that the outcomes that are of most interest to the practitioners are emphasised. As discussed, this will likely change on a case-by-case basis, and more than one weight function could be employed to gain a more complete understanding of the forecast performance; summarising plots such as Figure 2 are particularly useful.

In a weather forecasting context, the relationship between weather conditions and socio-economical impacts is relatively well-understood. Hence, an obvious choice of the weight function is one that reflects the costs associated with each possible outcome. Outcomes that correspond to higher impacts would therefore be emphasised in the weighted scores, and forecasters would be encouraged to issue more accurate forecasts for these high-impact events. Alternatively, to reduce the impact of these events, national weather centres issue warnings to the general public. These warnings typically correspond to relevant thresholds of the outcome variable, determined from climatological records and thorough analyses of the risks of weather to infrastructure and public health. A simple alternative weight function would thus use an indicator weight function (e.g. Equation 7), with the parameters in this weight function defined by the warning thresholds.

For the rainfall forecasts considered here, we firstly employ this latter weight function to emphasise values above a threshold of interest t, i.e. $w(z) = \mathbf{1}\{z > t\}$. This weight function can be employed in **scoringRules** by setting the arguments `a = t` and `b = Inf` in the weighted scoring rule functions. In doing so, the weighted scoring rules will assess the forecasts in their ability to predict high precipitation accumulations, which are of particular relevance since they often lead to flooding. As a threshold, we choose $t = \sqrt{30}$mm. This choice was made since 30mm is commonly chosen as a threshold in rainfall warning systems in central Europe, and $\sqrt{30}$ is roughly equal to the 95th percentile of the square root-transformed precipitation values in the training data considered here.

```
R> if (use_crch){
+    t <- sqrt(30)
+    gauss_twcrps <- twcrps_sample(y = obs, dat = gauss_sample, a = t)
+  }
```

Having repeated this for the logistic and Student's t forecast distributions, we can then compare the three post-processing models using their average threshold-weighted CRPS.

```
R> if (use_crch){
+    scores <- data.frame(Logistic = logis_twcrps, Gaussian = gauss_twcrps,
+                         Students_t = stud_twcrps, Ensemble = ens_twcrps)
+    sapply(scores, mean)
+  }

  Logistic   Gaussian Students_t   Ensemble
    0.0491     0.0490     0.0489     0.0774
```

These threshold-weighted CRPS values illustrate that, while the three post-processing methods are again almost indistinguishable, the relative improvement of the post-processed forecasts upon the raw ensemble forecasts has increased. This suggests that post-processing is particularly beneficial when predicting more extreme precipitation accumulations.

We could also calculate the outcome-weighted CRPS for these forecasts in a similar way. However, when interest is on extreme events, there is a chance that neither the observation nor any members of the predictive sample exceed the threshold of interest, resulting in an undefined outcome-weighted score.

Alternatively, the Gaussian distribution function could also be used to emphasise larger precipitation values without restricting attention only to values above a threshold. The threshold-weighted CRPS values corresponding to this weight function are given below. The results largely agree with those observed for the previous weight function.

```
R> if (use_crch){
+    sigma <- 1
+    weight_func <- get_weight_func("norm_cdf", mu = t, sigma = sigma)
+    chain_func <- get_weight_func("norm_cdf", mu = t, sigma = sigma,
+                                  weight = FALSE)
+    gauss_twcrps <- twcrps_sample(obs, gauss_sample, chain_func = chain_func)
+  }

R> if (use_crch){
+    scores <- data.frame(CRCHlogis = logis_twcrps, CRCHgauss = gauss_twcrps,
+                         CRCHstud = stud_twcrps, Ensemble = ens_twcrps)
+    sapply(scores, mean)
+  }

CRCHlogis CRCHgauss  CRCHstud  Ensemble
   0.0676    0.0676    0.0674    0.1079
```

The choice of `sigma` above is somewhat arbitrary, and depends on how quickly the weight function should tend to zero or one. If `sigma` is equal to zero, then, in theory, we should recover the indicator weight function employed above.

In general, the performance of the forecasts will change depending on the weight function. The weight functions discussed herein are only examples, and the choice of weight function will depend on the application. Other weight functions could also readily be employed.

4.2. Bayesian forecasts of US GDP growth

The second case study considers an example from economics. It is standard for national banks to issue forecasts for the country's gross domestic product (GDP) growth in the coming quarters. In this example, probabilistic forecasts of US GDP growth (in %) are obtained using a Markov switching autoregressive model, with exact details given in Krüger *et al.* (2021). This model is used to derive forecasts of quarterly US GDP growth for the following year, i.e. the next four quarters.

The data used in this example is available from the data set `gdp` in **scoringRules**, which contains observed US GDP growth for 271 quarters between 1947 and 2014. We use the data prior to 2014 as training data to estimate the Markov switching autoregressive model, which is then used to forecast the GDP growth in the four quarters of 2014.

The model implemented here is Bayesian, and is estimated using Markov chain Monte Carlo (MCMC) methods. As is common for Bayesian models that employ MCMC, the analytical form of the predictive distribution is not known. The forecast distributions considered here are therefore predictive samples obtained from the MCMC algorithm. The resulting forecast distributions are displayed in Figure 4 of Jordan *et al.* (2019).

These forecast distributions for each quarter can be evaluated univariately using both the CRPS and the LogS. In the following, `obs` denotes a vector containing the four observed GDP growths in 2014, while `X` is a matrix containing the MCMC predictions.

```
R> scores_crps <- crps_sample(obs, X)
R> scores_logs <- logs_sample(obs, X)
R> print(cbind(scores_crps, scores_logs))
```

```
       scores_crps scores_logs
2014Q1        3.48        4.06
2014Q2        1.33        2.28
2014Q3        1.73        2.56
2014Q4        0.72        1.97
```

As in the previous example, weighted versions of these scoring rules can again be used to target particular outcomes when quantifying forecast performance. The choice of weight function will again depend on what information is most relevant for practitioners. When forecasting GDP growth, it is particularly important to accurately predict when growth rates will be negative, since this suggests a decline in the country's economy. To emphasise negative growth rates within weighted scoring rules, a canonical weight function is $w(z) = \mathbf{1}\{z < 0\}$. Outcome-weighted and threshold-weighted CRPS values, as well as conditional and censored likelihood scores, are shown below for the Markov switching autoregressive forecasts considered here.

```
R> t <- 0
R> scores_owcrps <- owcrps_sample(obs, X, b = t)
R> scores_twcrps <- twcrps_sample(obs, X, b = t)
R> scores_cols <- clogs_sample(obs, X, b = t, cens = FALSE)
R> scores_cels <- clogs_sample(obs, X, b = t)
R> print(cbind(scores_owcrps, scores_twcrps, scores_cols, scores_cels))

       scores_owcrps scores_twcrps scores_cols scores_cels
2014Q1         0.636        1.8781        1.96       4.061
2014Q2         0.000        0.0300        0.00       0.211
2014Q3         0.000        0.0495        0.00       0.263
2014Q4         0.000        0.0605        0.00       0.280
```

The outcome-weighted CRPS and the conditional likelihood scores are zero for the last three quarters, since the observed GDP growths are greater than zero in these cases. The threshold-weighted CRPS and censored likelihood score, on the other hand, additionally assess the forecast probability that is assigned to a positive GDP growth occurring.

One could argue that economists also receive considerable attention when exceptionally high growth is forecast. To address this, a weight function could be used that simultaneously emphasises low and high GDP growth: e.g. $w(z) = \mathbf{1}\{z < 0\} + \mathbf{1}\{z > t\}$ for some reasonably high threshold $t > 0$. Although this does not align with the default weight function used within **scoringRules**, the `weight_func` and `chain_func` arguments allow this custom weight function to be employed within the weighted versions of the CRPS. The resulting scores are presented below. In this case, t is chosen to be 9%, which again roughly corresponds to the 95th percentile of the previously observed GDP growths.

```
R> a <- 0
R> b <- 9
R> weight_func <- function(x) as.numeric((x < a) | (x > b))
R> chain_func <- function(x) (x < a)*(x - a) + (x > b)*(x - b) + a

R> scores_owcrps <- owcrps_sample(obs, X, weight_func = weight_func)
R> scores_twcrps <- twcrps_sample(obs, X, chain_func = chain_func)
R> print(cbind(scores_owcrps, scores_twcrps))

       scores_owcrps scores_twcrps
2014Q1         0.766        1.8782
2014Q2         0.000        0.0302
2014Q3         0.000        0.0498
2014Q4         0.000        0.0612
```

Yet more relevant than forecasting negative GDP growth in an individual quarter is predicting a decline in GDP growth in successive quarters; this is commonly used by analysts as an indicator for a recession (U.S. Bureau of Economic Analysis (BEA) 2018). Hence, utilising this definition, evaluating forecasts for recessions becomes a multivariate problem.

The weighted multivariate scoring rules discussed herein allow us to emphasise successive quarters with negative GDP growth when evaluating forecast accuracy. If we consider two

consecutive declines in GDP growth, then a suitable weight function to employ in the weighted multivariate scores is $w(\mathbf{z}) = \mathbf{1}\{z_1 < 0, z_2 < 0\}$. This weight function can readily be extended to consider further quarters of negative GDP growth.

The threshold-weighted energy score, variogram score, and maximum mean discrepancy score are all shown below for the Markov switching autoregressive forecasts when predicting negative GDP growth in the two following quarters.

```
R> d <- 2
R> scores_twes <- twes_sample(obs[1:d], X[1:d, ], b = 0)
R> scores_twvs <- twvs_sample(obs[1:d], X[1:d, ], b = 0)
R> scores_twmmds <- twmmds_sample(obs[1:d], X[1:d, ], b = 0)
R> print(cbind(scores_twes, scores_twvs, scores_twmmds))

     scores_twes scores_twvs scores_twmmds
[1,]        1.78        2.61         0.243
```

Note that these weighted scoring rules consider not only the probability that a recession will occur, but also the severity of the recession.

5. Summary and discussion

Scoring rules are well-established when evaluating and comparing probabilistic forecasts, and the **scoringRules** package in R has become well-established when implementing popular scoring rules in practice. In this paper, we discuss how the functionality of the **scoringRules** package has been extended such that particular outcomes can be emphasised when using scoring rules to assess forecast performance.

Two approaches to target particular outcomes are available, which can be applied to probabilistic forecasts for both univariate and multivariate outcomes. These approaches are available for popular scoring rules including the LogS, CRPS, ES, VS, and MMDS, facilitating very flexible, user-oriented evaluation of probabilistic forecasts in a wide range of practical applications. In particular, functionality is available to calculate the conditional and censored likelihood scores proposed by Diks *et al.* (2011); outcome-weighted versions of the CRPS, ES, VS, and MMDS, which can be constructed from the general framework outlined by Holzmann and Klar (2017); and threshold-weighted versions of the CRPS, ES, VS, and MMDS (Matheson and Winkler 1976; Gneiting and Ranjan 2011; Allen *et al.* 2023b).

While the **scoringRules** package contains analytical expressions for the LogS and CRPS for several parametric distributions, the weighted scoring rules discussed herein are only available for forecast distributions in the form of a simulated sample, or an ensemble. In this case, the weighted scoring rules can readily be calculated for arbitrary weight functions, which is generally not the case for forecasts in the form of parametric distributions. While this permits very flexible forecast evaluation, the **scoringRules** package could be extended further by incorporating weighted scoring rules for certain families of parametric distributions.

Acknowledgements

This work was funded by the Swiss Federal Office for Meteorology and Climatology (Me-

teoSwiss) and the Oeschger Centre for Climate Change Research. I am very grateful to Fabian Krüger, Sebastian Lerch, and Alexander Jordan for their contribution to this work. Jonas Bhend and José Carlos Araujo Acuña are thanked for their many fruitful suggestions, while comments from two anonymous reviewers have also improved the paper.

References

Allen S (2023). "Weighted scoringRules: Emphasising particular outcomes when evaluating probabilistic forecasts." *arXiv preprint arXiv:2305.07312*. doi:10.48550/arXiv.2305.07312.

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**(3), 499–516. doi:10.1175/WAF-D-22-0161.1.

Allen S, Ginsbourger D, Ziegel J (2023b). "Evaluating forecasts for high-impact events using transformed kernel scores." *SIAM/ASA Journal on Uncertainty Quantification*, **11**(3), 906–940. doi:10.1137/22M1532184.

Diebold FX, Mariano RS (1995). "Comparing predictive accuracy." *Journal of Business & Economic Statistics*, **13**(3), 253–263. doi:10.1080/07350015.1995.10524599.

Diks C, Panchenko V, Van Dijk D (2011). "Likelihood-based scoring rules for comparing density forecasts in tails." *Journal of Econometrics*, **163**(2), 215–230. doi:10.1016/j.jeconom.2011.04.001.

Gneiting T, Raftery AE (2007). "Strictly Proper Scoring Rules, Prediction, and Estimation." *Journal of the American Statistical Association*, **102**(477), 359–378. doi:10.1198/016214506000001437.

Gneiting T, Raftery AE, Westveld III AH, Goldman T (2005). "Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation." *Monthly Weather Review*, **133**(5), 1098–1118. doi:10.1175/MWR2904.1.

Gneiting T, Ranjan R (2011). "Comparing density forecasts using threshold-and quantile-weighted scoring rules." *Journal of Business & Economic Statistics*, **29**(3), 411–422. doi:10.1198/jbes.2010.08110.

Good IJ (1952). "Rational Decisions." *Journal of the Royal Statistical Society B*, **14**(1), 107–114. doi:10.1111/j.2517-6161.1952.tb00104.x.

Holzmann H, Klar B (2017). "Focusing on regions of interest in forecast evaluation." *The Annals of Applied Statistics*, **11**(4), 2404–2431. doi:10.1214/17-AOAS1088.

Jordan A, Krüger F, Lerch S (2019). "Evaluating Probabilistic Forecasts with scoringRules." *Journal of Statistical Software*, **90**(12), 1–37. doi:10.18637/jss.v090.i12.

Krüger F, Lerch S, Thorarinsdottir TL, Gneiting T (2021). "Predictive Inference Based on Markov Chain Monte Carlo Output." *International Statistical Review*, **89**(2), 274–301. doi:10.1111/insr.12405.

Lerch S, Thorarinsdottir TL, Ravazzolo F, Gneiting T (2017). "Forecaster's dilemma: Extreme events and forecast evaluation." *Statistical Science*, **32**(1), 106–127. doi:10.1214/16-STS588.

Matheson JE, Winkler RL (1976). "Scoring rules for continuous probability distributions." *Management Science*, **22**(10), 1087–1096. doi:10.1287/mnsc.22.10.1087.

Messner JW, Mayr GJ, Zeileis A (2016). "Heteroscedastic Censored and Truncated Regression with crch." *The R Journal*, **8**(1), 173–181. doi:10.32614/RJ-2016-012.

Scheuerer M, Hamill TM (2015). "Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities." *Monthly Weather Review*, **143**(4), 1321–1334. doi:10.1175/MWR-D-14-00269.1.

Sharpe MA, Bysouth CE, Stretton RL (2018). "How well do Met Office post-processed site-specific probabilistic forecasts predict relative-extreme events?" *Meteorological Applications*, **25**(1), 23–32. doi:10.1002/met.1665.

Taillardat M, Fougères AL, Naveau P, de Fondeville R (2022). "Evaluating probabilistic forecasts of extremes using continuous ranked probability score distributions." *International Journal of Forecasting*. doi:10.1016/j.ijforecast.2022.07.003.

US Bureau of Economic Analysis (BEA) (2018). "Recession." https://www.bea.gov/help/glossary/recession. Accessed: 2023-01-01.

A. Scores for simulated predictive distributions

Consider $\Omega \subseteq \mathbb{R}$, and suppose the forecast distribution is only available via a simulated sample $x_1, \ldots, x_m \in \Omega$. To evaluate the empirical distribution function defined by this sample,

$$\hat{F}_m(z) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{x_i \leq z\},$$

the CRPS simplifies to

$$\text{CRPS}(\hat{F}_m, y) = \frac{1}{m} \sum_{i=1}^{m} |x_i - y| - \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |x_i - x_j|.$$

The outcome-weighted CRPS and threshold-weighted CRPS are defined analogously: given a univariate weight function w, the outcome-weighted CRPS can be written as

$$\text{owCRPS}(\hat{F}_m, y) = \frac{1}{m\bar{w}} \sum_{i=1}^{m} |x_i - y| w(x_i) w(y) - \frac{1}{2m^2\bar{w}^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |x_i - x_j| w(x_i) w(x_j) w(y),$$

where $\bar{w} = \sum_{i=1}^{m} w(x_i)/m$. Letting v denote the corresponding chaining function, the threshold-weighted CRPS is

$$\text{twCRPS}(\hat{F}_m, y) = \frac{1}{m} \sum_{i=1}^{m} |v(x_i) - v(y)| - \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |v(x_i) - v(x_j)|.$$

Similarly, let F be a forecast distribution on $\Omega \subseteq \mathbb{R}^d$ for $d > 1$, and suppose that only a sample $\mathbf{x}_1, \ldots, \mathbf{x}_m$ from F is available, with $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d}) \in \Omega$ for $i = 1, \ldots, m$. In this case, the energy score for the corresponding empirical multivariate distribution $\hat{F}_m$ can be written as

$$\text{ES}(\hat{F}_m, \mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{y}\| - \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|\mathbf{x}_i - \mathbf{x}_j\|,$$

the variogram score of order p becomes

$$\text{VS}^p(\hat{F}_m, \mathbf{y}) = \sum_{i=1}^{d} \sum_{j=1}^{d} h_{i,j} \left(\frac{1}{m} \sum_{k=1}^{m} |x_{k,i} - x_{k,j}|^p - |y_i - y_j|^p \right)^2,$$

and the maximum mean discrepancy score is

$$\text{MMDS}(\hat{F}_m, \mathbf{y}) = \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \exp\left\{ -\frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right\} - \frac{1}{m} \sum_{i=1}^{m} \exp\left\{ -\frac{1}{2} \|\mathbf{x}_i - \mathbf{y}\|^2 \right\}.$$

Given a multivariate weight function w and chaining function v, the outcome-weighted and threshold-weighted versions of these scores can be calculated as follows. In this case, $\bar{w} = \sum_{i=1}^{m} w(\mathbf{x}_i)/m$, and $v(\mathbf{x}_i) = (v(\mathbf{x}_i)_1, \ldots, v(\mathbf{x}_i)_d) \in \Omega$ for $i = 1, \ldots, m$.

$$\text{owES}(\hat{F}_m, \mathbf{y}) = \frac{1}{m\bar{w}} \sum_{i=1}^{m} \|\mathbf{x}_i - \mathbf{y}\| w(\mathbf{x}_i) w(\mathbf{y}) - \frac{1}{2m^2\bar{w}^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|\mathbf{x}_i - \mathbf{x}_j\| w(\mathbf{x}_i) w(\mathbf{x}_j) w(\mathbf{y});$$

$$\text{owVS}^p(\hat{F}_m, \mathbf{y}) = w(\mathbf{y}) \sum_{i=1}^{d} \sum_{j=1}^{d} h_{i,j} \left(\frac{1}{m\bar{w}} \sum_{k=1}^{m} |x_{k,i} - x_{k,j}|^p \, w(\mathbf{x}_k) - |y_i - y_j|^p \right)^2;$$

$$\begin{aligned}\text{owMMDS}(\hat{F}_m, \mathbf{y}) =& \frac{1}{2m^2\bar{w}^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \exp\left\{ -\frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 w(\mathbf{x}_i) w(\mathbf{x}_j) w(\mathbf{y}) \right\} \\ & - \frac{1}{m\bar{w}} \sum_{i=1}^{m} \exp\left\{ -\frac{1}{2} \|\mathbf{x}_i - \mathbf{y}\|^2 w(\mathbf{x}_i) w(\mathbf{y}) \right\};\end{aligned}$$

$$\text{twES}(\hat{F}_m, \mathbf{y}) = \frac{1}{m} \sum_{i=1}^{m} \|v(\mathbf{x}_i) - v(\mathbf{y})\| - \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|v(\mathbf{x}_i) - v(\mathbf{x}_j)\|;$$

$$\text{twVS}^p(\hat{F}_m, \mathbf{y}) = \sum_{i=1}^{d} \sum_{j=1}^{d} h_{i,j} \left(\frac{1}{m} \sum_{k=1}^{m} |v(\mathbf{x}_k)_i - v(\mathbf{x}_k)_j|^p - |v(\mathbf{y})_i - v(\mathbf{y})_j|^p \right)^2;$$

$$\text{twMMDS}(\hat{F}_m, \mathbf{y}) = \frac{1}{2m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \exp\left\{ -\frac{1}{2} \|v(\mathbf{x}_i) - v(\mathbf{x}_j)\|^2 \right\} - \frac{1}{m} \sum_{i=1}^{m} \exp\left\{ -\frac{1}{2} \|v(\mathbf{x}_i) - v(\mathbf{y})\|^2 \right\}.$$

Affiliation:

Sam Allen
ETH Zurich
Seminar for Statistics
Ramistrasse 101
8092 Zurich, Switzerland
E-Mail: sam.allen@stat.math.ethz.ch