Technometrics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/utch20

Building Trees for Probabilistic Prediction via Scoring Rules

Sara Shashaani, Özge Sürer, Matthew Plumlee & Seth Guikema

To cite this article: Sara Shashaani, Özge Sürer, Matthew Plumlee & Seth Guikema (2024) Building Trees for Probabilistic Prediction via Scoring Rules, Technometrics, 66:4, 625-637, DOI: 10.1080/00401706.2024.2343062

To link to this article: https://doi.org/10.1080/00401706.2024.2343062

TECHNOMETRICS
2024, VOL. 66, NO. 4, 625–637
https://doi.org/10.1080/00401706.2024.2343062

Taylor & Francis
Taylor & Francis Group

Building Trees for Probabilistic Prediction via Scoring Rules

Sara Shashaani[a], Özge Sürer[b], Matthew Plumlee[c], and Seth Guikema[d]

[a]Department of Industrial and Systems Engineering North Carolina State University, Raleigh, NC; [b]Department of Information Systems and Analytics, Miami University, Oxford, OH; [c]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL; [d]Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI

ABSTRACT

Decision trees built with data remain in widespread use for nonparametric prediction. Predicting probability distributions is preferred over point predictions when uncertainty plays a prominent role in analysis and decision-making. We study modifying a tree to produce nonparametric predictive distributions. We find the standard method for building trees may not result in good predictive distributions and propose changing the splitting criteria for trees to one based on proper scoring rules. Analysis of both simulated data and several real datasets demonstrates that using these new splitting criteria results in trees with improved predictive properties considering the entire predictive distribution.

ARTICLE HISTORY
Received July 2023
Accepted April 2024

KEYWORDS
Forecast evaluation; Interval scores; Machine learning; Proper scores; Uncertainty

1. Introduction

Binary trees that partition continuous response variables based on predictor variables have been proven useful for nonparametric regression (Breiman et al. 1984). Nonparametric regression is a general class of regression models that does not assume a parametric form for the relationship between predictors and dependent variables; binary trees can be considered an instance of them. After the tree is statistically learned via training data, any new data point maps to a leaf (terminal node) in the tree based on the predictors' values. The resulting output for prediction is typically a statistic measuring the center of responses (in the training data) that belong to the same node (Hastie, Tibshirani, and Friedman 2009). However, this statistic yields a decidedly deterministic forecast. In contrast, for many applications, such as weather and finance, it makes sense to predict probabilistically to communicate the stochastic nature of the system.

Goal of prediction. A probabilistic prediction has two major goals: (i) to have the observations be consistent with the predictive distribution, and (ii) to concentrate (sharpen) the prediction as much as possible given the predictor variables (Gneiting, Balabdaoui, and Raftery 2007). Thus, a reliable predictive distribution communicates both the magnitude of the prediction and the amount of uncertainty. The user could then convert the predictive distribution into a prediction. The best prediction might be a measure of center (mean), but it might be another feature of the predictive distribution based on the use case. For example, when predicting the number of power outages in a region after a storm, a 95% upper bound would provide a picture of high-risk areas (quantile). In another scenario, one might wish to find the probability that an online article does not meet a view target (tail probability). In yet another scenario, predicting the variance of power consumption in a neighborhood can be critical to understanding potential load imbalance risks (second moment). In all of these examples, the nature of the prediction cannot be gleaned from the sample mean.

1.1. Probabilistic Predictions with Trees

Given a tree, one can generate a nonparametric predictive distribution for each terminal node using that node's empirical cumulative distribution function (ECDF). This suggestion by Meinshausen (2006) is an input for the popular *quantile regression forests*. But there is no guarantee that standard trees learned from data will have good predictive properties.

The "standard" tree is built through a recursion where at each terminal node, potential splits of the tree are considered, and the split that most reduces the sum of squared errors (SSE) is chosen. Section 2 of this article will demonstrate that even in simple conditions, trees built by splitting based on the SSE criteria do not necessarily possess good predictive properties. There have been other criteria designed for splitting rules beyond this typical approach. Splitting rules for *classification* were well-dissected by Taylor and Silverman (1993) and Breiman (1996), which covered the Gini criteria and entropy. There appears to be less extensive literature on splitting rules for continuous prediction in nonparametric regression. Other splitting criteria such as log-rank (LeBlanc and Crowley 1993), likelihoods (Su, Wang, and Fan 2004; Zeileis, Hothorn, and Hornik 2008), and treatment difference models (Su et al. 2009; Athey and Imbens 2016) are specialized and/or rely on parametric frameworks. Athey, Tibshirani, and Wager (2019) offer a fully nonparametric method for predicting a quantity, not a predictive distribution.

CONTACT Sara Shashaani sshasha2@ncsu.edu Department of Industrial and Systems Engineering North Carolina State University, Raleigh, NC.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

1.2. Summary of Contributions and Insights

This article offers novelty by suggesting splitting criteria for trees based on *scoring rules*. Scoring rules assess predictions and have a lengthy history in statistics, information theory, and convex analysis (Gneiting and Raftery 2007). A scoring rule $S(F, y)$ takes a predictive distribution F (throughout the article this means the cumulative distribution function) and a realized quantity y and converts it into a scalar score. We will consider negatively oriented scoring rules, where the smaller the score is, the better we have done. A *proper* scoring rule encourages the predictor to provide the true distribution, which means one makes careful assessments and is honest about uncertainty (Garthwaite, Kadane, and O'Hagan 2005). Scoring rules tend to reward both goals of probabilistic prediction, though the respective importance is often hidden from the user.

The novel splitting criteria are as follows. Consider splitting a current terminal node into two smaller terminal nodes with data $\{y_1, \ldots, y_l\}$ and $\{y_{l+1}, \ldots, y_{l+r}\}$, to the left and right subsets, respectively. Then we choose the split that minimizes

$$\sum_{i=1}^{l} S\left(\widehat{F}_{\mathcal{L}}, y_i\right) + \sum_{i=1}^{r} S\left(\widehat{F}_{\mathcal{R}}, y_{l+i}\right), \tag{1}$$

where $\widehat{F}_{\mathcal{L}}$ and $\widehat{F}_{\mathcal{R}}$ are the predictive ECDFs of y relating to the left and right side of the split. As discussed earlier, finding an optimal split in a standard tree is through minimizing SSE, which is itself a scoring rule. Despite the relative simplicity of this formulation of splitting rules, the authors have found no reference to this mechanism with respect to building trees. The closest attempts in this direction have been studies on quantile-based loss functions (Bhat, Kumar, and Vaz 2015), density forecasts (Iacopini, Ravazzolo, and Rossini 2022), or gradient forests (Athey, Tibshirani, and Wager 2019).

By considering scoring rules other than SSE, we aim to improve the predictability of trees. But perhaps the most promising advantage of our proposed method is the application-dependent choice of a scoring rule. In different applications, probabilistic properties other than the mean behavior can be of importance. For example, interval scores that encourage narrow and consistent predictive intervals can be beneficial if a user routinely uses only predictive intervals from the predictive distribution (Christoffersen 1998), or in reliability applications and prediction of high-risk (extreme) events. Two-moment scores are more useful when mean and variance are both of importance, or with datasets that possess significant heteroscedasticity. While the aforementioned scoring rules might need to be justified for the context, continuously ranked probabilistic scores (CRPS) are strictly proper scoring rules that are already understood as a better fit in weather forecasting (Taillardat et al. 2016; Vogel et al. 2018). See Section 3 for a modest background on these scoring rules and their computational costs.

We describe the algorithmic structure for building score-based trees and pruning them in Section 4 along with an important structural property of proper scoring rules: monotonic improvement. Asymptotic analysis of splits in Section 5 provides a more general insight into scoring rules' necessary conditions for consistency. By examining synthetic and real datasets in Section 6, we show that different scoring rules return substantially different trees in real, practical examples. Our experiments confirm that when data is not completely summarized by the mean value, trees built with non-SSE scoring rules provide better predictions. Additionally, non-SSE trees can improve the SSE performance beyond traditional trees, and interval scores and CRPS achieve good prediction no matter what the goal of probabilistic prediction is. Section 7 closes the article with some remarks on extensions to this approach.

2. An Illustration

This section will motivate the use of scoring rules to guide the splitting of tree models from a strictly statistical perspective. Throughout, we use script fonts for sets, the notation $[a] := \{1, 2, \ldots, a\}$ for some positive integer a, and $y_n \xrightarrow{\text{p}} y$ for convergence in probability of a random sequence y_n to a random variable y. Let $\{(x_i, y_i)\}_{i=1}^n$ be the available data with $x_i = (x_i^1, x_i^2, \ldots, x_i^p)$ representing the p independent variables as potential predictors (features), and y_i representing the (real-valued) response that we wish to predict for unseen data. We denote $\mathcal{J} = \{1, 2, \ldots, n\}$ as the index set of the whole data. We also use the notations $\widehat{F}_{\mathcal{A}}$ and $\bar{y}_{\mathcal{A}}$ for the ECDF and sample mean of y values whose indices are in the set $\mathcal{A}$ (of cadinality $|\mathcal{A}|$), that is,

$$\widehat{F}_{\mathcal{A}}(z) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathbb{I}(y_i \le z), \quad \text{and} \quad \bar{y}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} y_i. \tag{2}$$

The SSE criterion cannot distinguish splits if the predictor variable x impacts the distribution of the response variable y but leaves the mean of y unperturbed. Conversely, other scoring rules, which consider the entire distribution, can easily find these splits. To show this effect, we use an obvious shift of behavior in a small toy example where predictors $x_1, \ldots, x_n \sim \text{Unif}(-1, 1)$ and the response variable is distributed as

$$y_i \sim \begin{cases} \text{Normal}(\mu = 1, \sigma = 2) & \text{if } x_i \in [-1, 0], \\ \text{Exponential}(\lambda = 1) & \text{if } x_i \in (0, 1]. \end{cases} \tag{3}$$

Say we do not know the split occurs at 0 but wish to build a tree of depth 1 to give good predictions. For any split (k, s), the traditional SSE score after split is

$$\begin{aligned} \text{SSE}(k, s) &= \sum_{i \in \mathcal{L}(k,s)} \underbrace{\left(y_i - \bar{y}_{\mathcal{L}(k,s)}\right)^2}_{S(\widehat{F}_{\mathcal{L}(k,s)}, y_i)} + \sum_{i \in \mathcal{R}(k,s)} \underbrace{\left(y_i - \bar{y}_{\mathcal{R}(k,s)}\right)^2}_{S(\widehat{F}_{\mathcal{R}(k,s)}, y_i)} \\ &= \sum_{i=1}^{n} \left(y_i - \bar{y}_{\mathcal{J}}\right)^2 - |\mathcal{L}(k,s)| \left(\bar{y}_{\mathcal{J}} - \bar{y}_{\mathcal{L}(k,s)}\right)^2 \\ &\qquad - |\mathcal{R}(k,s)| \left(\bar{y}_{\mathcal{J}} - \bar{y}_{\mathcal{R}(k,s)}\right)^2 \\ &= \sum_{i=1}^{n} \underbrace{\left(y_i - \bar{y}_{\mathcal{J}}\right)^2}_{S(\widehat{F}_{\mathcal{J}}, y_i)} - \underbrace{\frac{|\mathcal{L}(k,s)||\mathcal{R}(k,s)|}{n} \left(\bar{y}_{\mathcal{L}(k,s)} - \bar{y}_{\mathcal{R}(k,s)}\right)^2}_{\text{reduction in SSE after split}}, \end{aligned} \tag{4}$$

which shows the SSE score will always reduce as a result of the split. In (4), (k, s) denotes the split using the kth predictor based on its values less or greater than s. $\mathcal{L}(k, s) = \{i \in \mathcal{J} : x_i^k \le s\}$ and $\mathcal{R}(k, s) = \{i \in \mathcal{J} : x_i^k > s\}$ denote the index subsets to the left and right of the (k, s) split. The optimal split is then given by $(k^{\text{SSE}}, s^{\text{SSE}}) = \arg\min_{k \in [p], s \in \mathbb{R}} \text{SSE}(k, s)$. Clearly, this criteria depends only on the sample means on each side of the split. If

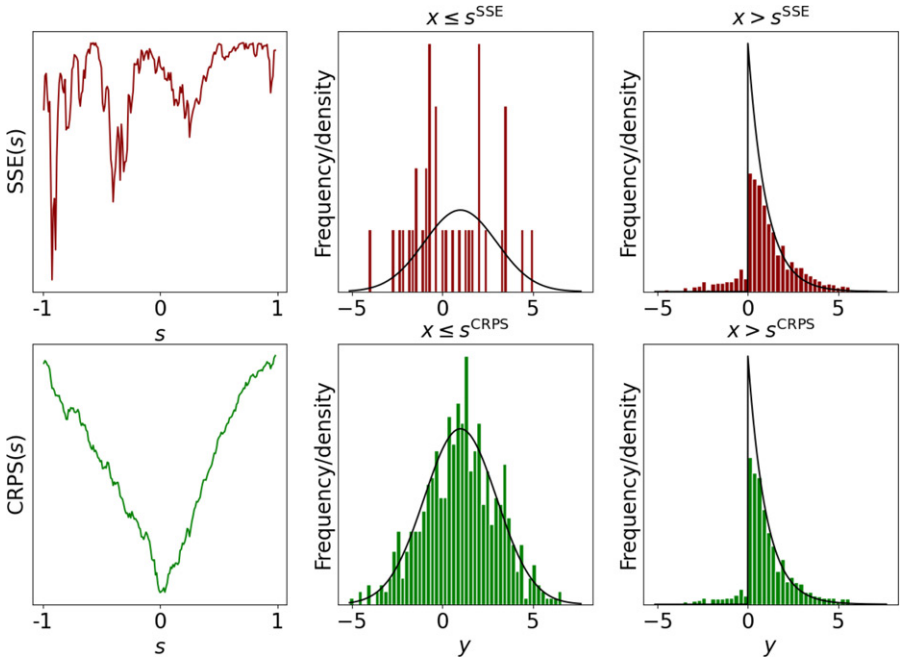


Figure 1. Splits resulting from the simulated experiment (3). In this one-dimensional example, k is always 1. The left panels show the criteria versus the split point. The center (right) panels show the histogram from the data below (above) the selected split overlaying the true density when $x \in [-1, 0]$ ($x \in (0, 1]$).

the sample means on each side are relatively close, SSE gives no information to guide the split; this criterion would likely be poor for splitting in our setting.

We can alternatively use scoring rules such as CRPS with a simple implementation of

$$\mathrm{CRPS}(k,s) = \underbrace{\frac{1}{2|\mathcal{L}(k,s)|}\sum_{i\in\mathcal{L}(k,s)}\sum_{j\in\mathcal{L}(k,s)}|y_i - y_j|}_{\sum_{i\in\mathcal{L}(k,s)} S(\widehat{F}_{\mathcal{L}(k,s)}, y_i)} + \underbrace{\frac{1}{2|\mathcal{R}(k,s)|}\sum_{i\in\mathcal{R}(k,s)}\sum_{j\in\mathcal{R}(k,s)}|y_i - y_j|}_{\sum_{i\in\mathcal{R}(k,s)} S(\widehat{F}_{\mathcal{R}(k,s)}, y_i)}, \quad (5)$$

to guide the split, yielding $(k^{\mathrm{CRPS}}, s^{\mathrm{CRPS}}) = \arg\min_{k\in[p],s\in\mathbb{R}} \mathrm{CRPS}(k,s)$. Compared to SSE, this criterion analyzes the difference between all of the values as opposed to just the sample means. We will discuss CRPS and its properties further in Section 3. See Gneiting and Raftery (2007) for a thorough description of CRPS.

Returning to our toy example, by changing s in the $[-1, 1]$ range, one can investigate where each of these criteria suggests the splitting must occur. Figure 1 illustrates the best split corresponding to each criterion and the density of data below and above that split value (overlaid by the known density) on a random dataset generated from (3) with $n = 1000$. The SSE criterion is noisy, with no particular behavior where the true split is known. The CRPS criterion also has some noise, but the trend focuses on the minimizer near 0. In this experiment, constructing the tree by minimizing a proper score is superior to using the standard splitting criteria.

With the promise of this experiment, we propose the idea of constructing the tree by minimizing a criterion based on proper scoring rules. These criteria are used to find the split that minimizes the *total score* $C(k, s)$:

$$C(k,s) := \sum_{i\in\mathcal{L}(k,s)} S\left(\widehat{F}_{\mathcal{L}(k,s)}, y_i\right) + \sum_{i\in\mathcal{R}(k,s)} S\left(\widehat{F}_{\mathcal{R}(k,s)}, y_i\right). \quad (6)$$

While the proposed framework is broad, we work with several famous scores. Of course, the use of each score depends on the application, but the main point is to investigate whether contracting the trees with proper scores results in more reliable trees.

Remark 1. The training data is used for (a) constructing the ECDF and (b) calculating the scores. However, both of these steps are at the service of fitting a model to the data at hand, such that the fitted model can best mimic that data. Using separate sets of data or cross-validation for fitting, that is, a training dataset to decide the splits and a separate validation dataset to calculate $\hat{F}$ for each split, will fail to minimize the score (loss) in the training dataset. We note double purposes with the training data does not lead to overfit; in a traditional tree, too, the same data that provides a predictive distribution, is used to compute the score at the fitting step. To avoid overfitting, we control its root cause, that is, model complexity (Hastie, Tibshirani, and Friedman 2009, sec. 2.9), by carefully choosing the tree parameters and by a pruning mechanism via cross-validation; see Sections 4.2 and 4.3.

3. Background on Scoring Rules

A scoring rule takes a distribution and an observed value and returns a score, often used to assess the closeness between the predictive distribution and reality. Gneiting and Raftery (2007), Dawid (2007), and Carvalho (2016) provide reviews, summaries, and applications of scoring rules. Here we employ them

to compare two or more alternative predictive distributions. However, there are other usages of scoring rules such as elicitation of distributions (Garthwaite, Kadane, and O'Hagan 2005), missing value imputation (Hasan et al. 2021), and Bayesian utility theory (Bernardo and Smith 2006).

Let $S(G, y)$ represent the score when distribution G is used and y is an observed continuous quantity. Moreover, let

$$\mathrm{ES}(G, F) := \mathbb{E}_{y\sim F} S(G, y), \tag{7}$$

denote the expected score of any distribution G for data that are randomly distributed following F. Scoring rules are negatively oriented, where smaller is better, and their most important property is propriety. A scoring rule is *proper* if for any distribution G,

$$\mathrm{ES}(F, F) \le \mathrm{ES}(G, F), \tag{8}$$

and a scoring rule is strictly proper if for all $G \neq F$ the inequality is strict. One interpretation of propriety is that if we choose a distribution to predict a quantity, the long-run average score is best minimized by selecting the true distribution. The function $\mathrm{ES}(F, F)$ is sometimes referred to as the information measure of S (Grünwald and Philip Dawid 2004). For us, it measures the ability of a set of data to predict other elements in that particular set of data, or self-similarity.

We now discuss a few options for scoring rules. The CRPS is given by

$$\begin{aligned} S^{\mathrm{CRPS}}(F, y) &= \int_{-\infty}^{\infty} \big(F(z) - \mathbb{I}(y \le z)\big)^2 \, \mathrm{d}z \\ &= \mathbb{E}_{z\sim F}|z - y| - \frac{1}{2}\mathbb{E}_{z\sim F, z'\sim F}|z - z'|. \end{aligned}$$

This is a strictly proper scoring rule and the one used in Section 2. Sometimes probabilistic predictions are summarized with their mean and variance. Scores that are only based on the mean and variance then can be used to evaluate the goodness of the predictions. The Dawid-Sebastiani score (Dawid and Sebastiani 1999) (DSS) is one example, given by

$$S^{\mathrm{DSS}}(F, y) = \frac{(\mu_F - y)^2}{\sigma_F^2} + \ln(\sigma_F^2),$$

where μ_F is the expected value corresponding to F and σ_F^2 is the variance corresponding to F. This has evident connections to the log-likelihood of a normal distribution, that is, $\ln\left(\exp\left(\frac{-(\mu_F - y)^2}{2\sigma_F^2}\right)/\sqrt{2\pi\sigma_F^2}\right)$, but normality is not required to employ this scoring rule. DSS is a proper scoring rule, but it is not a strictly proper scoring rule. This score can be used, for example, if the only important aspects of the distribution can be distilled down to the mean and variance. A further reduction would simply be the SSE scoring rule:

$$S^{\mathrm{SSE}}(F, y) = (\mu_F - y)^2,$$

(standard trees) which ignores the variance and is especially limiting when the variance is heterogeneous across different subregions of data. Lastly, we consider two scoring rules related to two-sided and one-sided intervals. Suppose that we are interested in $(1-\alpha)\times 100\%$ prediction intervals for some $0 \le \alpha \le 1$. The two-sided $1-\alpha$ interval score is defined as

$$\begin{aligned} S^{\mathrm{IS2}}(F, y) = {} & q_F\left(1 - \frac{\alpha}{2}\right) - q_F\left(\frac{\alpha}{2}\right) \\ & + \begin{cases} \frac{2}{\alpha}\left(q_F\left(\frac{\alpha}{2}\right) - y\right) & \text{if } y < q_F\left(\frac{\alpha}{2}\right) \\ \frac{2}{\alpha}\left(y - q_F\left(1 - \frac{\alpha}{2}\right)\right) & \text{if } y > q_F\left(1 - \frac{\alpha}{2}\right) \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

where $q_F(1-\alpha) = \inf\{z \in \mathbb{R} : 1 - \alpha \le F(z)\}$ is the $(1-\alpha)$th quantile of F. The definition of the quantile is important to maintaining the propriety of the scoring rule. While two-sided intervals are reported for many estimates, risk analysis often focuses on a single upper bound. One-sided intervals are also useful for positive data when the lower bound for a two-sided interval is close to zero. An upper bound interval score (IS1) has the form

$$S^{\mathrm{IS1}}(F, y) = q_F(1-\alpha) + \begin{cases} \frac{1}{\alpha}(y - q_F(1-\alpha)), & \text{if } y > q_F(1-\alpha) \\ 0 & \text{otherwise.} \end{cases}$$

A use case of IS1 is when forecasting potential crop yield where we want to find upper bounds to locate high-risk areas. This list of scoring rules is purposefully not exhaustive but presents various circumstances where each can be used. We will use each of these scoring rules to illustrate ideas throughout this article.

4. Building a Tree via Scoring Rules

We now formalize the proposed methodology to build a prediction tree based on data consisting of p predictors and a response for each of n observations.

Trees are typically built recursively (Breiman et al. 1984). Thus, the process used to find the first split, that is, node $t = 0$, is mirrored for all subsequent splits. We let $\mathcal{J}_t$ be the set of indices of data points that lie in node t (i.e., satisfy the intersection of splitting rules of node t's parent and grandparents recursively until reaching the root note). A split (s, k) creates two index sets, namely $\mathcal{L}_t(k, s) = \{i \in \mathcal{J}_t : x_i^k \le s\}$ and $\mathcal{R}_t(k, s) = \{i \in \mathcal{J}_t : x_i^k > s\}$. We propose to choose (k, s) by evaluating the predictive distributions resulting from the split via a scoring rule of interest, that is, the total score similar to (6), which can be rewritten as

$$\begin{aligned} C_t(k, s) = {} & |\mathcal{L}_t(k, s)|\mathrm{ES}\left(\widehat{F}_{\mathcal{L}_t(k,s)}, \widehat{F}_{\mathcal{L}_t(k,s)}\right) \\ & + |\mathcal{R}_t(k, s)|\mathrm{ES}\left(\widehat{F}_{\mathcal{R}_t(k,s)}, \widehat{F}_{\mathcal{R}_t(k,s)}\right). \end{aligned} \tag{9}$$

In line with the definition of expected score in (7), here we assume that the y data in node t follows its ECDF. Our splitting rule is selecting a predictor k and split value s that minimize $C_t(k, s)$ for a chosen scoring rule; we denote this rule for node t by (k_t, s_t).

There is an important property of scoring rules that makes our splitting criteria particularly attractive over alternatives. A tree recursively grown with SSE has a key feature of *monotonicity*. This means, as computed in (4), the SSE is nonincreasing after splitting:

$$\sum_{i\in\mathcal{L}_t(k,s)} (y_i - \bar{y}_{\mathcal{L}_t(k,s)})^2 + \sum_{i\in\mathcal{R}_t(k,s)} (y_i - \bar{y}_{\mathcal{R}_t(k,s)})^2 \le \sum_{i\in\mathcal{J}_t} (y_i - \bar{y}_{\mathcal{J}_t})^2$$

$$\text{for all } k \in [p], s \in \mathcal{S}_t^k,$$

Algorithm 1 PredictiveTree ($\{(x_i, y_i)\}_{i=1}^n$, max tree depth D, min node size N)

Create a terminal node with indices in $\mathcal{J}$ containing all data and set depth $d = 1$.
while $d \leq D$ **do**
  for nodes $t \in \{2^d - 1, \ldots, 2^{d+1} - 2\}$ labeled terminal **do**
    if terminal node has at most N data points **then**
      Label node as leaf and go to next terminal node.
    else
      Find $(k_t, s_t) = \arg\min_{k \in [p], s \in \mathcal{S}_t^k} C_t(k, s)$, where $C_t(k, s)$ is defined in (9).
      Create two terminal nodes whose sets of indices are $\mathcal{L}_t(k_t, s_t)$ and $\mathcal{R}_t(k_t, s_t)$.
      Index the two new nodes $2t + 1$ and $2t + 2$ and set $t = t + 1$.
    end if
  end for
  Set $d = d + 1$.
end while

where $\mathcal{S}_t^k$ is the set of all values that the kth predictor takes while being in node t. Arbitrary splitting rules will not always have this monotonicity property. However, Theorem 1 proves that our proposed splitting criteria have a monotonic feature analogous to SSE.

Theorem 1. Let $\mathcal{J}_t$ contain a subset of indices in the tth node of the tree, with y values to the left and right of any split (k, s) assumed to follow the corresponding ECDF, that is, $\widehat{F}_{\mathcal{L}_t(k,s)}$ and $\widehat{F}_{\mathcal{R}_t(k,s)}$. If S is a proper scoring rule, then

$$\sum_{i \in \mathcal{L}_t(k,s)} S\left(\widehat{F}_{\mathcal{L}_t(k,s)}, y_i\right) + \sum_{i \in \mathcal{R}_t(k,s)} S\left(\widehat{F}_{\mathcal{R}_t(k,s)}, y_i\right) \leq \sum_{i \in \mathcal{J}_t} S\left(\widehat{F}_{\mathcal{J}_t}, y_i\right)$$

for all possible (k, s).

That is, any splitting of the data will either reduce the total score or keep it unchanged.

We note, Theorem 1 states that the tree improves the score on the training data after every split. See supplemental material B for the proof. For a recursive algorithm, such a guarantee to improve the objective by considering more splits prevents the algorithm from getting stuck without finding the best possible tree.

4.1. Score-based Trees

The regression tree via scoring rules, as listed in Algorithm 1 is constructed starting at the root node with $t = 0$, containing the whole data. At each level d of the tree, all the nodes in that level that were labeled *terminal* are considered to be further split using the splitting criteria $C_t(k, s)$, unless they contain fewer than N (pre-specified parameter) data points, at which point those nodes are labeled as *leaves* and excluded from having offsprings. Ultimately, the leaves will provide the probabilistic predictions for data points that satisfy the same recursive criteria that form them. This process repeats up to a pre-specified depth of D in the tree. N and D are hyperparameters that classically control the tree-based models' complexity. Each node t that is split will generate two new nodes $2t + 1$ and $2t + 2$ with index sets $\mathcal{J}_{2t+1} := \mathcal{L}_t(k_t, s_t)$ and $\mathcal{J}_{2t+2} := \mathcal{R}_t(k_t, s_t)$.

4.2. Parameters and Implementation Specifics

Through standard mechanisms (Hastie, Tibshirani, and Friedman 2009, p. 308) in trees, the maximum depth D ensures terminal nodes will not be split when they have a certain number of parents. With abundant data, deeper trees could make the defining halfspaces in the leaves more complicated and in some sense, less interpretable. D best scales logarithmically with n (Klusowski 2020), specifically $D \approx \frac{p}{p+2} \log n$, to allow more splits if we have more data.

Besides the choice of D, because our tree will use ECDFs of y as predictive distributions, it is important to ensure that we have at least. say, N data points in each node. One rule of thumb for N is the Dvoretzky-Kiefer-Wolfowitz inequality (Dvoretzky, Kiefer, and Wolfowitz 1956; Massart 1990). This inequality suggests that $N \geq \frac{\log(2/\alpha)}{2\varepsilon^2}$ can guarantee at least ε-accurate ECDF with $1 - \alpha$ confidence. For example, 95% confidence at an accuracy of 10% gives at least 66 samples. In SSE-based trees, however, N is often chosen to be smaller ($\sim$ 10 (Bertsimas, Dunn, and Mundru 2019)). This can be explained by non-SSE-based trees tend to successfully assess the distributional behavior of the data at the cost of forcing larger terminal nodes. But larger terminal nodes could mean smaller trees, which may be advantageous for generalization (Athey, Tibshirani, and Wager 2019). Importantly, N is not a termination criterion for the entire tree; it prevents a certain branch of the tree from growing more than what we have data for. In all classical tree building literature, both N and D are used to mitigate risks of overfitting. If the tree is too deep, it will tightly track the training data. On the other hand, if a node is too small, it yields too crude ECDF and error-prone statistical information. Controlling the node size with maximum depth D is not guaranteed because while deeper trees ultimately result in smaller nodes, with a shallow tree one can still have terminal nodes that do not have enough data points in them. Hence, ensuring at least N data points in leaves becomes necessary.

A different parameter to set is related to searching among the split values of each variable for the best split. Given that this algorithm is likely to be used on tall datasets with potentially sizeable $\mathcal{S}_t^k$ sets in a terminal node k, cycling through all unique values of $\mathcal{S}_t^k$ (to consider them as a potential split value) leads to a slowdown in the algorithm. Thus, for each predictor, one can opt for a search through a set of $1/\ell$ quantiles $\mathcal{Q}_t^k(\ell) =$

$\{q_t^k(\ell), q_t^k(2\ell), \ldots, q_t^k(1-\ell)\}$ of each predictor k in node t instead. For example, when $\ell = 0.05$, then for each predictor only 20 split values will become candidates to identify the split. For discrete predictors with 10 unique values or less, as well as the categorical predictors, all the possibilities will be considered in the search for best splits. In the experiments, DSS and IS1 have computational time comparable with SSE but CRPS is computationally more expensive.

As the last practical consideration, given that CRPS requires $\mathcal{O}(n^2)$ operations in (5) and expensive for larger datasets, it is more appropriate for implementation of CRPS-based trees to use an alternative computation of CRPS with $\mathcal{O}(n \log n)$ complexity with the approximation $S^{\text{CRPS}}(\widehat{F}_{\mathcal{J}}, y) \approx \frac{2}{n^2} \sum_{i=1}^{n} (y_{(i)} - y)(n\mathbb{I}(y < y_{(i)}) - i + \frac{1}{2})$ that uses the order statistics $y_{(i)}$'s (sorted samples) for computation (Zamo and Naveau 2018).

4.3. Pruning Probabilistic Trees

The tree in Algorithm 1 is grown to depth D symmetrically. However, given the greediness of optimal splits, the best tree structure that divides the data into partitions may not be symmetric depending on the identified first optimal split. Trees tend to overfit, and the tree size (i.e., the number of terminal nodes in the tree with depth D) is controlled by a complexity (regularization) parameter κ. Smaller trees are understood to provide better accuracy and interpretability power. Pruning is done after growing a full tree (post-pruning) or simultaneously (pre-pruning), which implies stopping the growth at a node. Pre-pruning is more cost-effective, and its common approaches are listed in the supplementary material Section A for the reader's reference.

Unlike the common approach, which is growing the tree to its full size and then cutting back subtrees to combine some of the predictions, we explore stopping the tree growth at the nodes whose split does not dramatically improve the prediction quality. There have been setbacks about this approach for potentially missing a very good split that follows a seemingly weak split in the tree (James et al. 2013). However, we adopt this pruning approach to avoid unnecessary computation and obtain smaller trees, albeit with varying sensitivity levels across different scoring rules, which we will explore.

For each terminal node t with more than N data points, the optimal split leads to two new terminal nodes that by the monotonicity property satisfy

$$|\mathcal{J}_{2t+1}|\text{ES}(\widehat{F}_{\mathcal{J}_{2t+1}}, \widehat{F}_{\mathcal{J}_{2t+1}}) + |\mathcal{J}_{2t+2}|\text{ES}(\widehat{F}_{\mathcal{J}_{2t+2}}, \widehat{F}_{\mathcal{J}_{2t+2}}) \leq |\mathcal{J}_t|\text{ES}(\widehat{F}_{\mathcal{J}_t}, \widehat{F}_{\mathcal{J}_t}).$$

Let $\Delta_t := |\mathcal{J}_t|\text{ES}(\widehat{F}_{\mathcal{J}_t}, \widehat{F}_{\mathcal{J}_t}) - \left(|\mathcal{J}_{2t+1}|\text{ES}(\widehat{F}_{\mathcal{J}_{2t+1}}, \widehat{F}_{\mathcal{J}_{2t+1}}) + |\mathcal{J}_{2t+2}|\text{ES}(\widehat{F}_{\mathcal{J}_{2t+2}}, \widehat{F}_{\mathcal{J}_{2t+2}})\right)$ be the reduction is score after splitting in node t.

By expecting that Δ_t gradually decreases as the tree becomes deeper, we propose a heuristic to accept the split on node t if the point-average reduction in the score as a result of it is at least $\kappa \in [0, 1]$ factor of the point-average reduction in the score as a result of the split in the root node (the first optimal split), that is, $\Delta_t/n_t > \kappa \Delta_0/n$ where $n_t = |\mathcal{J}_t|$. Equivalently, we accept the best split at node t if

$$\text{ES}(\widehat{F}_{\mathcal{J}_t}, \widehat{F}_{\mathcal{J}_t}) - \left(\frac{n_{2t+1}}{n_t}\text{ES}(\widehat{F}_{\mathcal{J}_{2t+1}}, \widehat{F}_{\mathcal{J}_{2t+1}}) + \frac{n_{2t+2}}{n_t}\text{ES}(\widehat{F}_{\mathcal{J}_{2t+2}}, \widehat{F}_{\mathcal{J}_{2t+2}})\right) > \kappa\left(\text{ES}(\widehat{F}_{\mathcal{J}}, \widehat{F}_{\mathcal{J}}) - \left(\frac{n_1}{n}\text{ES}(\widehat{F}_{\mathcal{J}_1}, \widehat{F}_{\mathcal{J}_1}) + \frac{n_2}{n}\text{ES}(\widehat{F}_{\mathcal{J}_2}, \widehat{F}_{\mathcal{J}_2})\right)\right). \tag{10}$$

Note, with $\kappa = 0$, Algorithm 1 remains the same. As κ increases, the size of the tree becomes smaller. If $\kappa = 1$, we only have a root node in the tree.

5. Near-Optimality of the Empirical Split

This section explains some of the theoretical behavior of our trees learned from finite data. Our treatment will be decidedly less general than comparative work on the asymptotic behavior of trees (Gordon and Olshen 1980; Toth and Eltinge 2011; Scornet et al. 2015). This section's goal is to explain the impact of finite data on the new splitting criteria based on scoring rules. With some loss of generality, this section will only consider the behavior of a single split and keeps the available dataset used for splitting fixed (not random). Here we answer the following question in a general setting: given that our split is based on finite data, how does this compare to the prediction if one chooses the split optimally?

Say that we have a collection of realizations $(x_1, y_1), \ldots, (x_n, y_n)$ which are assumed to be from some joint distribution. Throughout the analysis, we fix this dataset that has an optimal split (yielding lowest total score when used to predict unseen targets y). Denote the potential splits by regions $A_1, \ldots, A_t, \ldots, A_T$; these are a collection of *half-spaces* of the form $\{x : x^k \leq s\}$. The potential splits are considered to be nonrandom for simplicity. In this section, we replace (k, s) splits with A regions to ease the exposure, and use $\mathcal{L}(A; n)$ and $\mathcal{R}(A; n)$ to reflect the dependence on n. Our chosen split is dictated by

$$\hat{A}_n = \underset{A \in \{A_1, \ldots, A_T\}}{\arg\min} \sum_{i:\, x_i \in A} S\left(\widehat{F}_{\mathcal{L}(A;n)}, y_i\right) + \sum_{i:\, x_i \in A^c} S\left(\widehat{F}_{\mathcal{R}(A;n)}, y_i\right),$$

where $\mathcal{L}(A; n)$ and $\mathcal{R}(A; n)$ are the subsets of n data points with their predictors lying on either side of the split that defines sub-region A. Let $F_{\mathcal{L}(A;\infty)}$ and $F_{\mathcal{R}(A;\infty)}$ represent the *true* conditional distributions of y for data whose predictors lie on either side of the split that defines sub-region A. It makes sense to judge a split A via the following criteria

$$g(A) := \text{ES}\left(\widehat{F}_{\mathcal{L}(A;n)}, F_{\mathcal{L}(A;\infty)}\right)\Pr(x \in A) + \text{ES}\left(\widehat{F}_{\mathcal{R}(A;n)}, F_{\mathcal{R}(A;\infty)}\right)\Pr\left(x \in A^c\right).$$

This represents the expected score for a new prediction of unobserved data after the split is finished. An oracle would choose the split such that

$$g^* := g(A_n^*) = \min_{A \in \{A_1, \ldots, A_T\}} g(A),$$

where the *oracle split* choice that yields g^* is denoted by A_n^*. Clearly, we would like $g\left(\hat{A}_n\right)$ to be as close as possible to g^*. In the spirit of the generality of this article, we now state a condition for general scoring rules.

Theorem 2. Let $\mathcal{P}$ be a class of every distribution of a random y given $x \in A$ for all subsets A of the predictor space, $\{y_1, \ldots, y_n\}$

be independent draws from a mixture of two distributions $F, G \in \mathcal{P}$ defined on nonoverlapping support sets A^F and A^G, with n_F as the number drawn from F and n_G as the number drawn from G where $n = n_F + n_G$. Denote the corresponding empirical predictive distributions by $\widehat{F}_n$ and $\widehat{G}_n$. Setting the mixture distribution $M := \frac{n_F}{n}\widehat{F}_n + \frac{n_G}{n}\widehat{G}_n$ given n_F and n_G, it holds true that $g(\hat{A}_n) \xrightarrow{\text{p}} g^*$ as $n \to \infty$ if as $n \to \infty$ for all $F, G \in \mathcal{P}, F \neq G$,

$$\text{ES}\left(M, \widehat{F}_n\right) \xrightarrow{\text{p}} \text{ES}\left(M, F\right). \tag{11}$$

Theorem 2 states that the predictive distributions (ECDFs in subregions, given a fixed dataset) of a score-based tree approach the highest accuracy (smallest score) when predicting increasingly large sets of unseen data. The implication of (11) is that the score must obey consistency (in the second argument) for the target variable. For the special cases of scoring rules used in this article, the next corollary shows this requirement is met in some reasonably well-behaved probability space $\mathcal{P}$. The tricky part of showing this result for a given scoring rule is that $\widehat{F}_n$ appears on both sides of the score. Thus, we cannot directly invoke the law of large numbers. See supplemental materials B for the proofs.

Besides providing the result in full generality, we next offer specific conditions for the scoring rules introduced in Section 3.

Corollary 1. If S is chosen to be CRPS or DSS, then assuming that for all subregions of predictor space A, the distribution of a random y conditioned on $x \in A$ is such that $\mathbb{E}(y^2)$ is finite, we get $g(\hat{A}_n) \xrightarrow{\text{p}} g^*$ as $n \to \infty$. If S is chosen to be IS1 or IS2, then assuming that for all subsets of predictor space, A, the distribution of a random y conditioned on $x \in A$ is such that the CDF for y is strictly increasing near $\alpha/2$ and $1-\alpha/2$ for IS2 and $1-\alpha$ for IS1, we get $g(\hat{A}_n) \xrightarrow{\text{p}} g^*$ as $n \to \infty$.

The moment condition of Corollary 1 gives guarantees that the CRPS/DSS score is well-behaved. For the interval and upper bound score, the condition shifts from a moment-based condition to one that guarantees convergence of the sample quantile. This condition can be modified for discrete data.

These results are intended to verify the intuition that these scores based on ECDFs lead to splits that, even though we have no proof for them to be the correct optimal splits, their resulting tree scores will be close enough to the scores in the optimal trees with high probability. Thus, the scoring rule choice will impact the ultimate tree that is constructed, no matter how much data is present. The choice of scoring rule thus, cannot be ignored and can have a large impact on the resulting prediction. One example of this was in Section 2, but our analysis of real data in Section 6 confirms this result. Table 2 in supplemental material Section D also shows, using synthetic datasets, that certain scoring rules fall short of finding the boundaries in the data where the probabilistic behavior changes especially if the change happens less obviously and beyond mean values.

6. Numerical Experiments

In this section, we examine the new tree construction methods using different scoring rules with experiments on synthetic datasets and real public datasets. As a baseline for comparison, we use standard trees with SSE criteria. All approaches are implemented under our own Python package `scoreTree`, publicly available at *https://github.com/sshashaa/scoreTree*. The code is also provided as an online supplementary material and the `README` file provides instructions to replicate examples from the article.

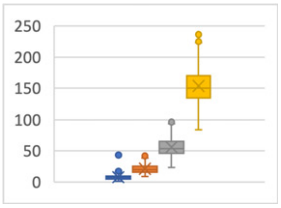
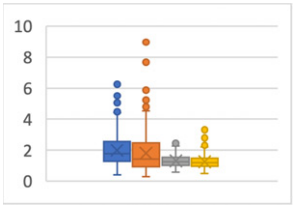
6.1. Synthetic Datasets

Two synthetic datasets for one-dimensional continuous feature space in $[0, 1]$ are designed with response behavior in four regions described in Table 1. The *easy* dataset exhibits easy-to-distinguish behavior of the response in each subregion, evidenced by significant differences in the first and higher central moments. The conjecture is that SSE should easily separate these regions using the first moment. On the other hand, the *hard* dataset entails more similarly behaving responses in the first two moments everywhere, making it harder for SSE-based trees to predict when there is difference in behavior. Although real-world data may not be in a tree structure, the synthetic datasets mimic the heteroscedasticity and responses that follow a mixture of distributions.

We construct trees with several scoring rules (Build $\in$ {SSE, CRPS, DSS, IS1}) and evaluate their performance under a varied number of training data sizes n and different choices of the pruning parameter κ introduced in (10). The benchmark procedure is summarized in Algorithm 2. For each dataset presented in Table 1, samples of size $n \in \{200, 400, 800, 1600\}$ are generated as training datasets and thresholds $\kappa \in \{0.0, 0.1, 0.3, 0.5, 0.8\}$ are implemented with each tree. For the comparisons, for each experiment (i.e., for each combination of Build, n, and κ), we generated one test set of 1,000 observations (to evaluate its performance). Each tree is evaluated with both in-sample (I) and out-of-sample (O) errors via different scoring rules denoted by Eval $\in$ {SSE, CRPS, DSS, IS1} using training and test sets, respectively. Data is repeatedly ($r = 30$ independent times) divided into an equal-sized training set for all experiments with common random numbers (CRN). CRN helps us see the effect of different trees and their performances on the same sets of data for training and testing, reducing the variability for comparison. Consequently, the predictive distributions are approximated by the data points that lie in the terminal node $t(j; \text{Build}, \kappa, b)$ as indicated in Line 5 of Algorithm 2 for the bth data replicate, the trees constructed with Build score, and pruned with threshold κ. In-sample and out-of-sample errors, represented by $\{\text{I}_b^{\text{Eval}}(\text{Build}, \kappa)\}_{b=1,2,\ldots,r}$ and $\{\text{O}_b^{\text{Eval}}(\text{Build}, \kappa)\}_{b=1,2,\ldots,r}$, are evaluated with Eval score to summarize the results (Lines 7–8 in Algorithm 2). Since the responses are nonnegative, we only use the upper interval score IS1 with $\alpha = 0.2$ (implying that, when fitting a tree we penalize a prediction that is worse than the 0.8-quantile of the predictive distribution). For all trees, $D = 4$, $N = 50$, $\ell = 0.05$ following the rules of thumb described in Section 4.2.

Our first comparison validates whether the tree built with a scoring rule of interest (Eval) yields better probabilistic predictions (lower scores) on out-of-sample data than trees constructed with the same data but with different scoring rules. We

Table 1. Synthetic datasets with `logNormal(lgN)` distributions of y on x subregions.

	Regions	$-1 < x < -0.5$	$-0.5 < x < 0$	$0 < x < 0.5$	$0.5 < x < 1$	Sub-region boxplots
Easy dataset	y Dist.	`lgN(2,1/2)`	`lgN(3,1/3)`	`lgN(4,1/4)`	`lgN(5,1/5)`	
	$\mathbb{E}[y]$	7.9	21.1	56.1	153.4	
	$\mathbb{E}[y^2]$	82.4	494.0	3359.4	24372.0	
	$\mathbb{E}[y^3]$	1210.3	12894.8	213981.9	4012557.9	
Hard dataset	y Dist.	`lgN(1/2,0.5)`	`lgN(1/3,0.6)`	`lgN(1/4,0.3)`	`lgN(1/5,0.3)`	
	$\mathbb{E}[y]$	1.99	1.80	1.31	1.26	
	$\mathbb{E}[y^2]$	5.04	4.73	1.88	1.76	
	$\mathbb{E}[y^3]$	15.66	17.50	2.90	2.75	

Algorithm 2 ScoreTreeExperiment(bootstrapped datasets $\ell_b,\ b = 1, 2, \ldots, r$)

for Bootstrap $\ell_b,\ b = 1, 2, \ldots, r$ **do**
  for Pruning parameter $\kappa \in \{0, 0.1, 0.3, 0.5, 0.8\}$ **do**
    for Scoring rule Build $\in$ {SSE, CRPS, DSS, IS1} **do**
      Train a tree with the Build score, ℓ_b data, and pruning parameter κ.
      Return $t(j; \text{Build}, \kappa, b)$, terminal node containing jth data point $\forall j \in \ell_b$.
      for Scoring rule Eval $\in$ {SSE, CRPS, DSS, IS1} **do**
        Compute $\text{I}_b^{\text{Eval}}(\text{Build}, \kappa) := \sum_{j \in \ell_b} S^{\text{Eval}}\left(\widehat{F}_{\mathcal{J}_{t(j;\text{Build},\kappa,b)}}, y_j\right)$.
        Compute $\text{O}_b^{\text{Eval}}(\text{Build}, \kappa) := \sum_{j \notin \ell_b} S^{\text{Eval}}\left(\widehat{F}_{\mathcal{J}_{t(j;\text{Build},\kappa,b)}}, y_j\right)$.
      end for
    end for
  end for
end for

evaluate the paired difference of scores for out-of-sample scores:

$$\text{OD}_b(\text{Eval}, \text{Build}, \kappa) := \text{O}_b^{\text{Eval}}(\text{Eval}, \kappa) - \text{O}_b^{\text{Eval}}(\text{Build}, \kappa), \quad \forall b = 1, 2, \cdots, r \tag{12}$$

between trees constructed with Eval and Build scores using the Eval score, where negative values validate that trees yield better predictions if trained with the same scoring rule that evaluates them (based on the goal of prediction).

Figure 2 shows one instance of these comparisons with Build = SSE and Eval = CRPS for the hard dataset and two choices $\kappa \in \{0, 0.5\}$ for pruning. We observe that as the training data size increases, SSE-based trees fail to provide good predictions when the goal is to have a good CRPS performance, since the paired difference confidence interval is negative. This weakness of SSE-based trees is statistically significant with pruning. See Table 1 in the supplemental material for a complete statistical test for all pairs of Eval and Build scores. This complete statistical test suggests that we can generally validate that Eval trees are better than Build trees when compared in Eval score. However, for the hard dataset, some scores struggle more than others. An interesting observation is the effect of pruning in helping the fit when using different scores on both datasets. For example, for the hard dataset, we observe that even a small pruning of $\kappa = 0.1$ can impact the validation of DSS- and IS1 trees.

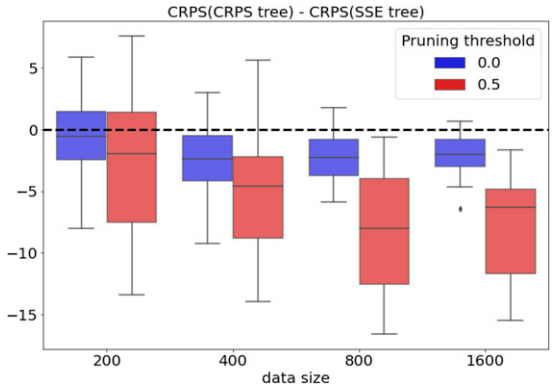


Figure 2. Boxplots of the paired difference of CRPS scores between CRPS trees and SSE trees on out-of-sample predictions for the hard dataset suggest that with the growing size of training data, CRPS trees provide better predictions than SSE trees.

Different scoring rules will best function under varying intensities of pruning. Figure 3 shows a CRPS tree trained with different pruning parameters for one instance of the hard dataset. As expected, the higher pruning values lead to a smaller tree (solid lines); yet the same pruning parameter may lead to different tree sizes when used with different scores. The best pruning value for SSE may not be the same as that for CRPS. To

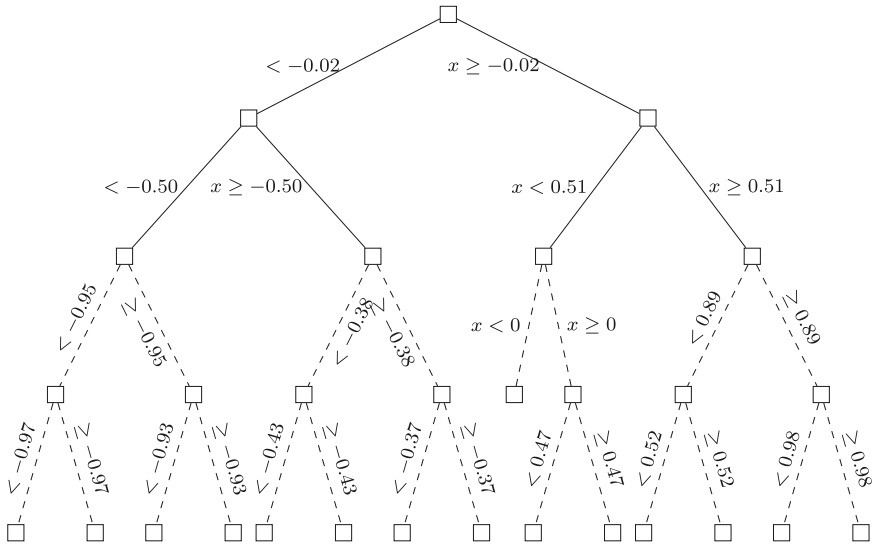


Figure 3. A CRPS tree at $D = 4$ with $\kappa = 0$ (dashed lines) and $\kappa = 0.5$ (solid lines).

Table 2. The optimal pruning κ^*(Score) for each scoring rule and each data size.

	Easy dataset				Hard dataset			
n : Data size	200	400	800	1600	200	400	800	1600
Score								
SSE	0.0	0.0	0.0	0.0	0.8	0.8	0.8	0.3
CRPS	0.1	0.1	0.1	0.0	0.8	0.8	0.8	0.5
DSS	0.1	0.1	0.1	0.1	0.8	0.3	0.1	0.1
IS1	0.1	0.1	0.1	0.0	0.8	0.8	0.8	0.3

compare each tree with its counterparts built via other scores, we first find the best pruning value for each score via cross-validation (using out-of-sample results):

$$\kappa^*(\text{Score}) := \arg\min_{\kappa} \frac{1}{r} \sum_{b=1}^{r} \text{O}_b^{\text{Score}}(\text{Score}, \kappa),$$

given a data size. These values are summarized in Table 2. These values suggest that for the easy dataset, $\kappa^* = 0.1$ is generally a good value across training data sizes and scoring rules, except SSE which does not appear to benefit from pruning (aligned with evidence from the hypothesis test results in Table 1 of the supplementary material).

There are more irregularities in the hard dataset. All scoring rules favor pruning, some less than others when sufficient training data is available. However, for the small data size, all scoring rules provide their best performance with the smallest tree that is pruned with $\kappa = 0.8$. DSS shows different behavior than the other scores for the hard dataset. Besides these observations, while not visible in Table 2, the variance of the optimal κ performance for IS1 is noticeably larger than the other scores. Another noteworthy point is that for the in-sample results, the $\kappa^* = 0.0$ for all scores and all data sizes implies that without pruning, the trees are subject to overfit, especially for the hard dataset.

To alleviate the interactive effect of pruning and scores, we compare the best version of each score-based tree using their corresponding optimal pruning value. We construct a confidence interval for the paired difference of optimal scores

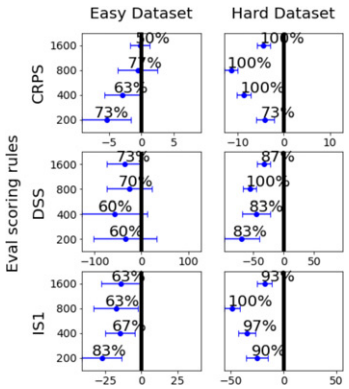


Figure 4. Confidence intervals of OD*(Eval, SSE) and corresponding estimated success probabilities EPS(Eval, SSE) (labels on each interval) from 30 replications with CRN, for varying training data sizes (y axis) of the easy and hard dataset.

$$\text{OD}_b^*(\text{Eval}, \text{Build}) := \text{O}_b^{\text{Eval}}(\text{Eval}, \kappa^*(\text{Eval})) - \text{O}_b^{\text{Eval}}(\text{Build}, \kappa^*(\text{Build})),\ \forall b = 1, 2, \ldots, r \quad (13)$$

defined similar to (12). We also estimate the *estimated probability of success*, EPS(Eval,Build), defined as the fraction of replications with the Eval-based tree leading to better Eval scores than Build-based trees, that is,

$$\text{EPS}(\text{Eval}, \text{Build}) := \frac{1}{r} \sum_{b=1}^{r} \mathbb{I}\{\text{OD}_b^*(\text{Eval}, \text{Build}) \leq 0\}. \quad (14)$$

Figure 4 summarizes paired difference of optimal Eval scores for Eval- and SSE-based treesand corresponding estimated success probabilities of the Eval-based trees.

In most cases, especially for the hard dataset, the out-performance of CRPS-, DSS-, and IS1-based trees over SSE-based trees is statistically significant. The percentage of times that an SSE-based tree is worse than its counterparts is also notably high across cases. This result confirms that non-SSE-based trees can achieve better probabilistic predictions when the data is not completely summarized by mean values (a property synthesized in the hard dataset). We also observe that the length

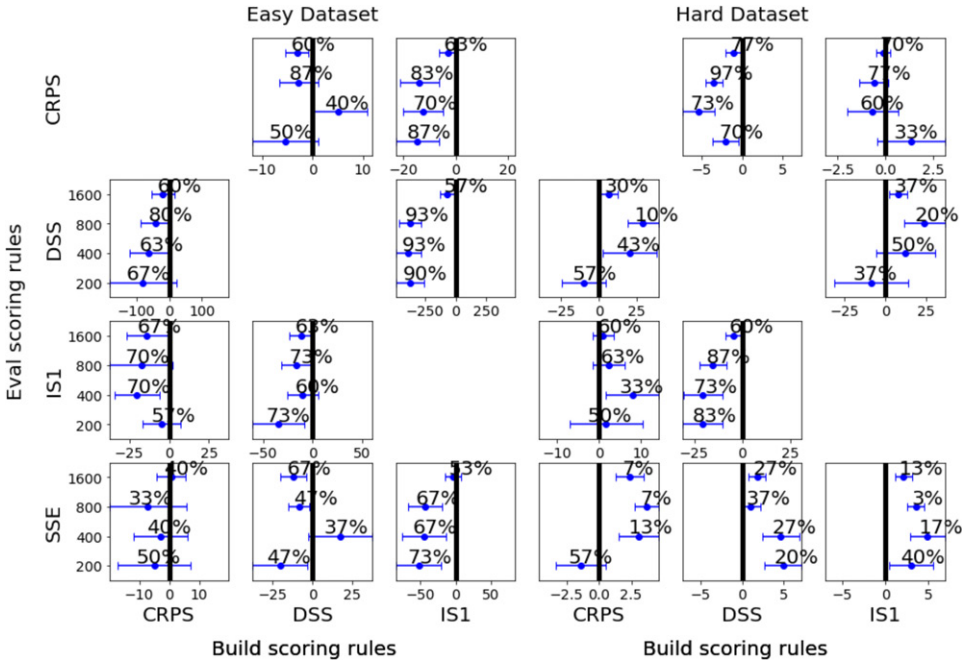


Figure 5. Confidence intervals of OD*(Eval, Build) and corresponding estimated success probabilities EPS(Eval, Build) (labels on each interval) of non-SSE Build scoring rules for varying training data sizes of the easy and hard dataset.

of the confidence intervals often decreases with sample size. This can be explained by the fact that estimates of mean, quantiles, and variance for the purpose of fitting trees is noisier with smaller training data.

In a follow-up experiment, to see whether there is a score that unanimously outperforms other scores, we investigated OD*(Eval, Build) confidence intervals and success probabilities for building trees with CRPS, DSS, and IS1 scores. Figure 5 summarizes these results.

A number of observations from Figure 5 are noteworthy:

(a) In the easy dataset, all scoring rules provide relatively similar probabilistic predictions; while IS1 barely ever improves trees regardless of the goal of prediction (negative CI's and large probabilities in IS1 column), there is less evidence to say the same for CRPS and DSS.
(b) In the hard dataset, CRPS- and IS1-based trees provide similar performance to one another. But compared to DSS- and SSE-based trees, they are more likely to provide better predictions and their improved performance is statistically significant as the data size increases. DSS-based tree only show better performance compared to SSE-based trees, but do so with statistical significance invariably across data sizes. Non-SSE trees almost always lead to better SSE scores (positive CI's and small probabilities in SSE row). The same holds for DSS score when the data size is not too small. Good CRPS and IS1 scores are not achievable with SSE- and DSS-based trees.

In our final investigation of this section, we compare the trees' ability to find the correct splits. While the main purpose of score-based trees is to produce better probabilistic predictions, identifying the correct subregions will render their suitability more convincing. As expected, the non-SSE trees more successful discern the subregions; Figure 3 shows, for example, that

Table 3. The optimal pruning κ^*(Score) for each scoring rule and each data size.

	Yield dataset		Divvy dataset	
n : Data size	5000	10,000	5000	10,000
Score				
SSE	0.0	0.1	0.8	0.8
CRPS	0.0	0.0	0.0	0.0
DSS	0.0	0.0	0.8	0.5
IS1	0.0	0.0	0.8	0.3

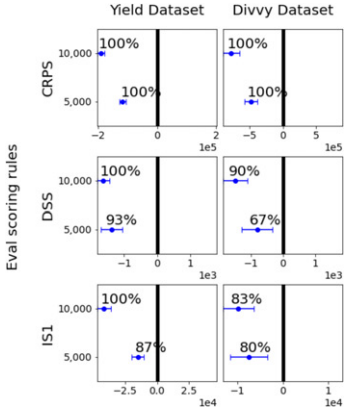


Figure 6. Confidence intervals of OD*(Eval, SSE) and corresponding estimated success probabilities EPS(Eval, SSE) (labels on each interval) from 30 replications with CRN, for varying training data sizes (y axis) of the Yield and Divvy dataset.

the CRPS-based tree is able to identify correct splits (within a ± 0.02 margin of error). If the tree is not sufficiently pruned, many incorrect splits will be contained in the tree structure (dash lines). But even in a sufficiently small tree, the split values can be incorrect if other scoring rules are used for splitting. For a more comprehensive comparison in this regard, see Table 2 in supplementary material Section D. Table 2 shows that (i) all true split points are more likely to be recovered by non-SSE trees,

(ii) SSE- and DSS-based trees find several inaccurate splits on average, while CRPS-based trees find the fewest inaccurate splits of all, and (iii) among non-SSE trees, the most difficult subregion are more likely identified by IS1 than DSS, and most likely identified by CRPS. A direct implication of correct identification of split values is the improved interpretability of data. In many applications such as in health outcome predictions, these correct split values lead to correct clustering of patients with distributionally similar outcomes and more accurate personalized predictions (Mao et al. 2022).

6.2. Real Datasets

We next investigate the score-based trees on two real datasets; see supplementary material Section E for their detailed descriptions. The first is the Yield data from the Ethiopian Annual Agricultural Surveys with 174,028 rows × 5 predictors, and ∼ $94K$ unique response values. The second is the Divvy bikeshare data from the city of Chicago with $1.3M$ rows × 9 predictors and ∼ $3.4K$ unique response values. Our analysis again entails $r = 30$ replications, with CRN and training data of sizes $n = 5000$ and $n = 10{,}000$, and computed κ^* for each score (see Table 3). For all trees, we set $D = 4$, $N = 250$, $\ell = 0.05$.

We note that Table 3 indicates that the best results are obtained without pruning the trees in almost all cases for the Yield dataset. A possible explanation for the superior prediction performance on the Yield dataset with larger trees could be attributed to its heterogeneous behavior, particularly evident in the tails, as illustrated in Figure E.1 of supplementary material.

We compare the Eval-score of optimal Eval-based trees with those of the optimal SSE-based trees in Figure 6. Similar to the synthetic data, we make our comparisons with (i) probability of success computed by (14) and (ii) paired difference confidence intervals computed using (13). Figure 6 illustrates that non-SSE trees provide statistically better predictions than SSE trees. In all cases, non-SSE trees outperform the SSE trees more resoundingly with larger training data.

In a subsequent experiment, we examine the confidence intervals and success probabilities of OD*(Eval, Build) when constructing trees using CRPS, DSS, and IS1 scores to determine whether one of these scoring rules consistently outperforms others. Figure 7 provides the summary of the results. In almost all cases, non-SSE trees lead to enhanced SSE scores (SSE rows in Figure 7). In the Divvy dataset, CRPS-based trees demonstrate superior performance compared to other tree types, while those based on DSS and IS1 exhibit similar performance. In the Yield dataset, while CRPS-based trees are not very sensitive to the sample size, the performance of DSS- and IS1-based trees improves with larger sample sizes. We note that if the prediction goal is to obtain good CRPS scores, then CRPS-based trees are unbeatable (CRPS rows in Figure 7).

In the supplementary material Section E, we provide additional experiment results with the minimum node size $N = 100$ (Figures E.2 and E.3). These additional results aim to deepen our understanding of the sensitivity exhibited by each score-based tree. While the results are robust for changing values of N for the Divvy dataset, the performance of the SSE-based trees is affected by decreasing N, as illustrated in Figure E.3 (fully negative confidence intervals and high success probability in the SSE row of the Yield dataset comparing SSE-based trees with CRPS- and DSS-based trees). This experiment underscores that smaller terminal nodes in SSE-based trees can yield competitive

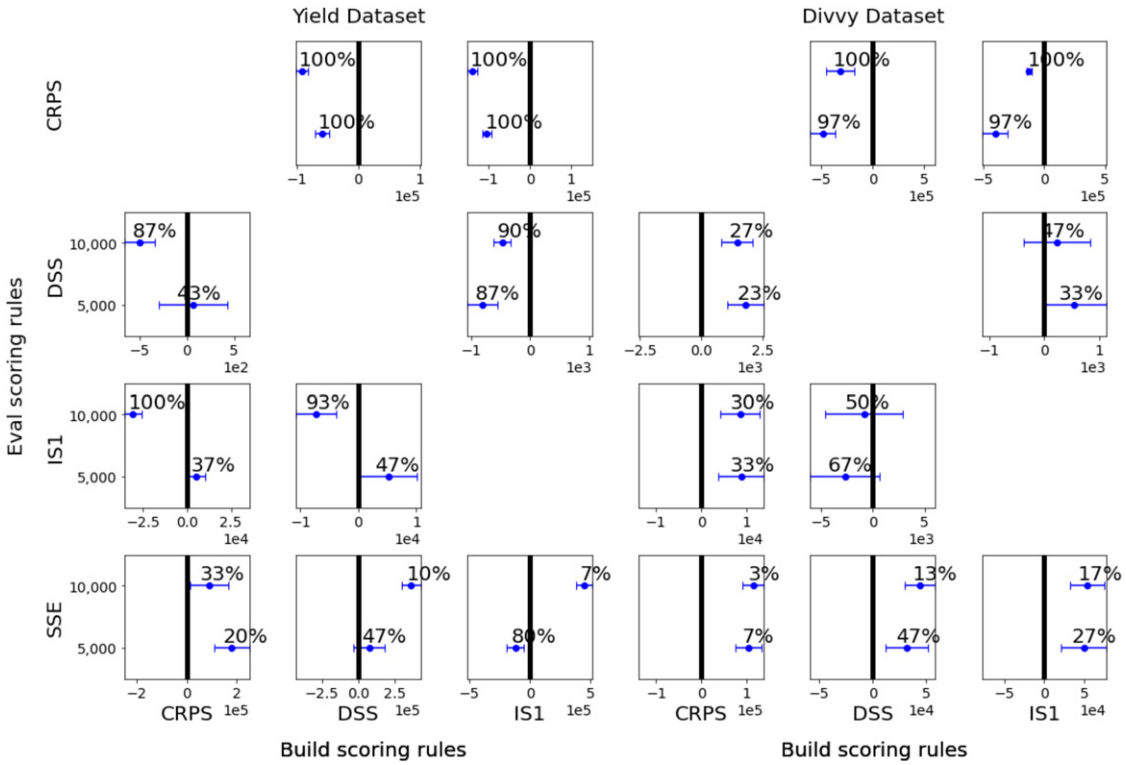


Figure 7. Confidence intervals of OD*(Eval, Build) and corresponding estimated success probabilities EPS(Eval, Build) (labels on each interval) of non-SSE Build scoring rules for varying training data sizes of the Yield and Divvy datasets.

SSE scores comparable to those derived from the remaining trees. This is not surprising because more data improves the empirical estimates of quantiles and higher moments and subsequently enhances the quality of trees trained with non-SSE scoring rules.

7. Concluding Remarks

In this article, we discuss that standard mechanisms for regression trees are not designed to grow a tree with the goal of creating good nonparametric predictive distributions. We aim to build a tree with generally good predictive distribution and conclude that fitting regression trees to training data by using proper scoring rules other than SSE as the split criteria can improve predictive properties. This is because, unlike SSE that summarizes the predictive distribution with its mean value, other proper scoring rules will focus on various other summary statistics (quantiles, higher moments, etc) that are of importance depending on the application and heterogeneity in the data. Since the recursive partitioning of the proposed trees is dictated by the scoring rule, when the scoring rule is chosen to align with the goal of prediction or based on some knowledge about the data, the resulting tree produces improvements over SSE-based trees. The type of score can also affect the additional computation for computing other summary statistics in the predictive distributions, but if chosen well, it can lead to not only better predictions, but potentially also better interpretation on the partitions created for the data (finding the correct split points). Our near-optimal analysis and numerical results conclusively show unanimous gain in using scoring-based trees. By extension, trees with proper scoring rules can provide a significant improvement when used as based learners and in ensemble settings such as forests. We leave these important extensions for future research.

Supplementary Materials

The supplemental materials include A) List of the existing pre-pruning algorithms; B) Proofs of theorems; C) Statistical tests based on different scores; D) Additional plots with synthetic data for tree comparisons; E) Finding the true splits; and F) Real data descriptions and additional experiments.

Acknowledgments

The authors gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory. The authors are also thankful to Dr. Timothy Williams for their help in acquiring and understanding one of the Yield dataset used in this study.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

Sara Shashaani is grateful for support from the National Science Foundation (NSF) grant CMMI-2226347. Özge Sürer's work was supported by the NSF grant OAC 2004601.

References

Athey, S., and Imbens, G. (2016), "Recursive Partitioning for Heterogeneous Causal Effects," *Proceedings of the National Academy of Sciences*, 113, 7353–7360. [625]

Athey, S., Tibshirani, J., and Wager, S. (2019), "Generalized Random Forests," *The Annals of Statistics*, 47, 1148–1178. [625,626,629]

Bernardo, J., and Smith, A. (2006), *Bayesian Theory*, Toronto: Wiley. [628]

Bertsimas, D., Dunn, J., and Mundru, N. (2019), "Optimal Prescriptive Trees," *INFORMS Journal on Optimization*, 1, 164–183. [629]

Bhat, H. S., Kumar, N., and Vaz G. J. (2015), "Towards Scalable Quantile Regression Trees," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 53–60. IEEE. [626]

Breiman, L. (1996), "Some Properties of Splitting Criteria," *Machine Learning*, 24, 41–47. [625]

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, London: Chapman and Hall/CRC. [625,628]

Carvalho, A. (2016), "An Overview of Applications of Proper Scoring Rules," *Decision Analysis*, 13, 223–242. [627]

Christoffersen, P. F. (1998), "Evaluating Interval Forecasts," *International Economic Review*, 39, 841–862. [626]

Dawid, A. P. (2007), "The Geometry of Proper Scoring Rules," *Annals of the Institute of Statistical Mathematics*, 59, 77–93. [627]

Dawid, A. P., and Sebastiani, P. (1999), "Coherent Dispersion Criteria for Optimal Experimental Design," *Annals of Statistics*, 27, 65–81. [628]

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. (1956), "Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator," *The Annals of Mathematical Statistics*, 27, 642–669. [629]

Garthwaite, P. H., Kadane, J. B., and O'Hagan, A. (2005), "Statistical Methods for Eliciting Probability Distributions," *Journal of the American Statistical Association*, 100, 680–701. [626,628]

Gneiting, T., and Raftery, A. E. (2007), "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, 102, 359–378. [626,627]

Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007), "Probabilistic Forecasts, Calibration and Sharpness," *Journal of the Royal Statistical Society*, Series B, 69, 243–268. [625]

Gordon, L., and Olshen, R. A. (1980), "Consistent Nonparametric Regression from Recursive Partitioning Schemes," *Journal of Multivariate Analysis*, 10, 611–627. [630]

Grünwald, P. D., and Philip Dawid, A. (2004), "Game Theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian Decision Theory," *The Annals of Statistics*, 32, 1367–1433. [628]

Hasan, M. K., Alam, M. A., Roy, S., Dutta, A., Jawad, M. T., and Das, S. (2021), "Missing Value Imputation Affects the Performance of Machine Learning: A Review and Analysis of the Literature (2010–2021)," *Informatics in Medicine Unlocked*, 27, 100799. [628]

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York, NY: Springer. [625,627,629]

Iacopini, M., Ravazzolo, F., and Rossini, L. (2022), "Proper Scoring Rules for Evaluating Density Forecasts with Asymmetric Loss Functions," *Journal of Business & Economic Statistics*, 41, 482–496. [626]

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013), *An Introduction to Statistical Learning* (Vol. 112), New York: Springer. [630]

Klusowski, J. (2020), "Sparse Learning with Cart," *Advances in Neural Information Processing Systems*, 33, 11612–11622. [629]

LeBlanc, M., and Crowley, J. (1993), "Survival Trees by Goodness of Split," *Journal of the American Statistical Association*, 88, 457–467. [625]

Mao, L., Vahdat, K., Shashaani, S., and Swann, J. L. (2022), "Personalized Predictions for Unplanned Urinary Tract Infection Hospitalizations with Hierarchical Clustering," in *AI and Analytics for Public Health: Proceedings of the 2020 INFORMS International Conference on Service Science*, pp. 453–465, Springer. [635]

Massart, P. (1990), "The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality," *The annals of Probability*, 18, 1269–1283. [629]

Meinshausen, N. (2006), "Quantile Regression Forests," *Journal of Machine Learning Research*, 7, 983–999. [625]

Scornet, E., Biau, G., Vert, J.-P., et al. (2015), "Consistency of Random Forests," *The Annals of Statistics*, 43, 1716–1741. [630]

Su, X., Wang, M., and Fan, J. (2004), "Maximum Likelihood Regression Trees," *Journal of Computational and Graphical Statistics*, 13, 586–598. [625]

Su, X., Tsai, C.-L., Wang, H., Nickerson, D. M., and Li, B. (2009), "Subgroup Analysis via Recursive Partitioning," *Journal of Machine Learning Research*, 10, 141–158. [625]

Taillardat, M., Mestre, O., Zamo, M., and Naveau, P. (2016), "Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics," *Monthly Weather Review*, 144, 2375–2393. [626]

Taylor, P. C., and Silverman, B. W. (1993), "Block Diagrams and Splitting Criteria for Classification Trees," *Statistics and Computing*, 3, 147–161. [625]

Toth, D., and Eltinge, J. L. (2011), "Building Consistent Regression Trees from Complex Sample Data," *Journal of the American Statistical Association*, 106, 1626–1636. [630]

Vogel, P., Knippertz, P., Fink, A. H., Schlueter, A., and Gneiting, T. (2018), "Skill of Global Raw and Postprocessed Ensemble Predictions of Rainfall over Northern Tropical Africa," *Weather and Forecasting*, 33, 369–388. [626]

Zamo, M., and Naveau, P. (2018), "Estimation of the Continuous Ranked Probability Score with Limited Information and Applications to Ensemble Weather Forecasts," *Mathematical Geosciences*, 50, 209–234. [630]

Zeileis, A., Hothorn, T., and Hornik, K. (2008), "Model-based Recursive Partitioning," *Journal of Computational and Graphical Statistics*, 17, 492–514. [625]