TECHNICAL REPORT 3350
June 2024

Softening the Prediction Surface of Decision Tree Regressors

Joshua A. Duclos
Benjamin A. Michlin, Ph.D.
Andrew B. Sabater, Ph.D.
Jamal T. Rorie, Ph.D.
NIWC PACIFIC

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001

This page is intentionally blank.

TECHNICAL REPORT 3350
June 2024

Softening the Prediction Surface of Decision Tree Regressors

Joshua A. Duclos
Benjamin A. Michlin, Ph.D.
Andrew B. Sabater, Ph.D.
Jamal T. Rorie, Ph.D.
NIWC PACIFIC

Administrative Note:

This report was approved through the Release of Scientific and Technical Information (RSTI) process in January 2024 and formally published in the Defense Technical Information Center (DTIC) in June 2024.

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001

NIWC Pacific
San Diego, California 92152-5001

P. M. McKenna, CAPT, USN
Commanding Officer

M. J. McMillan
Executive Director

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Basic and Applied Research Division of the Cyber/Science and Technology Department (717), Naval Information Warfare Center (NIWC) Pacific, San Diego, CA. The NIWC Pacific Naval Innovative Science and Engineering (NISE) Program provided funding for this applied research project.

Released by
John deGrassie, Division Head
Basic and Applied Research Division

Under authority of
Carly Jackson, Department Head
Cyber/Science and Technology Department

ACKNOWLEDGMENTS

Editor: MGK

EXECUTIVE SUMMARY

While decision trees function as base-learners for many machine learning methods and exhibit several useful properties, the compromise between prediction accuracy and generalization limits their utility in non-ensemble applications. We propose a new method to improve the predictive performance of pre-trained decision tree regressors. Using the tree structure and metadata, we derive a set of decision threshold-based weights that modify the leaf prediction values. The weighted values are then aggregated into a final "softened" prediction which more accurately represents the true target distribution. We demonstrate the approach on a variety of benchmark data sets and observe a mean improvement of 11% over the baseline decision tree R^2 values. We further explore the parameters of the approach and characterize their effects.

This page is intentionally blank.

CONTENTS

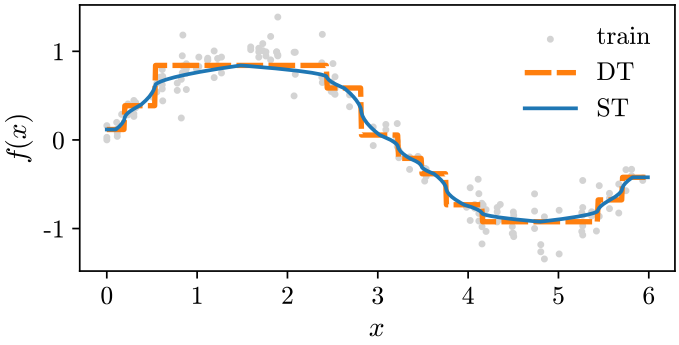
Figures

Tables

This page is intentionally blank.

1. INTRODUCTION

As lightweight function approximators, decision tree regressors form the basis of many model approaches. They are easy to train, are human interpretable, and provide reasonably accurate predictions with low computational overhead. However, decision trees' piecewise prediction function limits their ability to fully represent real-world continuous targets. Further, attempts to increase predictive performance often come at the expense of generalization; increasing tree depth allows for more granular prediction but increases the risk of approximating the training data and not the underlying data-generating process (overfitting). Improving generalization by selectively pruning branches necessarily reduces the expressiveness of the model. Ensembling methods such as random forests [1, 2] and boosted trees (e.g. gradient boosted trees [3], AdaBoost [4, 5] address these limitations using multiple trees. While effective, these methods require additional computational resources and may not be ideal in constrained environments or where prediction explanations are necessary.



Soft tree method applied to a decision tree fit to a 1-dimensional example data set. The stepwise function of a decision tree (DT, orange dashed line) approximates data sampled from a noisy sine process (grey points). The soft tree approach (ST, solid blue line) smooths over the decision thresholds, improving generalized predictive performance.

Figure 1. Soft tree example

Other approaches contend with limitations in decision tree predictive performance by altering the prediction mechanism of the tree. The linear regression tree approach [6] constructs a standard tree with linear regression models at the leaves. This improves over the constant-value prediction of standard trees; however, it remains noncontinuous in the target space. The Bonsai tree approach [7] constructs a sparse, shallow tree where all nodes and leaves become non-linear predictors. Predictions result from the sum of intermediate predictor outcomes along the decision path. Although the decision nodes continue to provide context, prediction explanations under this approach are ultimately obscured by the aggregation of non-linear predictors. Further, while this method works well in the embedded domain, it requires data for training and cannot be applied to a pre-trained decision tree.

A trained decision tree provides a basic information set contained within the nodes and leaves comprising the tree's anatomy. At each decision node, the considered input feature and threshold value identify a division of the feature space. This n-dimensional subdivided space represents the learned structure of the training data with the corresponding leaf prediction values. Additional information is also commonly retained in the probability of each branch derived from the number of samples seen at each node.

We propose a method for improving decision tree regression performance by leveraging the latent information of the tree structure. $Soft\ trees$ improve model predictions by smoothing the piecewise prediction surface. This approach modifies the distance of the input vector from the threshold at each decision node with a smoothing function to produce a weight. The branch probabilities are balanced against these distance weights to provide overall branch contributions to the final value. The predicted output of the soft tree then becomes a recursive weighted combination of all the leaves in a selected subtree. The smoothing shape, the branch probability balance, and the aggregation depth are all tunable parameters.

Conceptual similarities to soft trees may be found compared to spline regression [8]. The knots bounding the splines are analogous to the midpoints between thresholds of the softened decision tree. Traditionally, the number and position of these knots can present a challenge for tuning, although more recent work [9] has demonstrated methods for automatic selection. In contrast, soft trees rely on the thresholds learned by the tree from the data and are able to incorporate arbitrary elements within the tree into a final prediction.

Notably, the soft tree method improves a trained decision tree without access to training data[1]. This approach capability is desirable where existing decision trees are trained on lost or restricted data, collecting new training data is impractical (e.g., data generated monthly), or where retraining is otherwise infeasible.

The soft tree approach is detailed in this paper and organized as follows: Section 2 describes the soft tree methodology and presents a discussion of the parameters; Section 3 identifies data sets and details experiment execution; Section 4 explores the results of the experimentation and considers findings; Section 5 discusses conclusions and identifies future work.

[1]While training data are unnecessary, the approach requires metadata representing the expected value range of the features.

2. METHODOLOGY

Assume a set of data (X, Y) such that the independent variable $x_i \in \mathbb{R}^n$ is a vector $x : (x_1, x_2, \ldots, x_n)$ of length $n > 0$ within X and the dependent variable $y_i \in \mathbb{R}$ is an associated scalar in Y. A decision tree regressor, f, is constructed such that $\hat{y} = f(x; k)$ minimizes the expected squared error, $\mathbb{E}\left[(\hat{y} - y)^2\right]$, for all $(x, y) \in (X, Y)$ where k is the depth of the tree.

The resulting trained tree contains at least a minimal set of metadata sequences, each describing the tree's structure or descriptive elements of the training data. The sequences $\mathcal{C}_\mathrm{L}$ and $\mathcal{C}_\mathrm{R}$ represent each node's left and right children, respectively. Each decision node has elements identifying the selected feature, $\mathcal{H}$, and the decision threshold, $\mathcal{T}$. Finally, $\mathcal{S}$ represents the number of training samples arriving at each node and $\mathcal{V}$ contains the prediction values of the leaves[2]. The soft tree approach exploits this set of meta-features to create a new predictor, $\hat{f}$, such that $\mathbb{E}[(\hat{f}(x) - y)^2] \leq \mathbb{E}[(f(x; k) - y)^2]$.

At the core of the method, decision nodes establish a weighting for each child branch. This continues recursively down each subtree until the leaf nodes are reached. The derived weights control the relative contribution of the leaf values as they are propagated up the tree during inference. The relationship between branch weights at the i^{th} node is given by

$$1 = w_{c_i^\mathrm{L}} + w_{c_i^\mathrm{R}} \tag{1}$$

where $w_{c_i^\mathrm{L}}$ is the weighting applied to the left branch child's result, and $w_{c_i^\mathrm{R}}$ is applied to the right. The weights are each restricted to the interval $[0, 1] \in \mathbb{R}$, in accordance with (1). The branch weights are composed of three components: a contribution from threshold distance, the branch probability, and a probability adjustment.

2.1 Threshold distance

Intuitively, input feature values near corresponding decision thresholds suggest an inherent uncertainty in the tested inequality. For samples very close to the threshold, even small perturbations—for example, sampling error—could dramatically alter the decision path and eventual predicted value. Accordingly, a threshold distance, d, is a proxy for this uncertainty and facilitates a correlated weight. Formally, the threshold distance at the i^{th} decision node is the normalized difference of the input value and the feature threshold as defined by

$$d_i = \frac{|t_i - x_{h_i}|}{v_i^{\max} - v_i^{\min}} \tag{2}$$

where $t \in \mathcal{T}$ is the threshold value; x_h is the input value at decision feature $h \in \mathcal{H}$; and $v^{\max}$ and $v^{\min}$ are the maximum and minimum values of feature h at node i. The resulting normalized distance, d_i, is constrained to the interval $[0, 1]$.

The minimum and maximum values of the features at each decision node are discovered by interrogating the tree structure[3]. For a given node i considering feature h, the upper and lower bounds in the feature dimension are defined as

$$v_i^{\max} = \begin{cases} \min\{t \mid t > t_i \text{ and } t \in T_h\}, & \text{if } x > t_i \\ t_i, & \text{else} \end{cases} \tag{3}$$

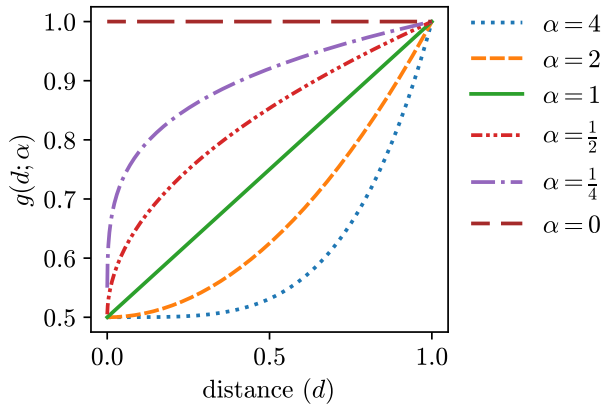
[2]Sequences have an upper size bound related to the maximum number of decision nodes $(2^k - 1)$ and leaves (2^k) in a full tree.

[3]Assumes consistency in the direction of inequalities at all decision nodes (e.g., nodes always test for *less-than-or-equal*).

and

$$v_i^{\min} = \begin{cases} \max\{t \mid t < t_i \text{ and } t \in T_h\}, & \text{if } x \leq t_i \\ t_i, & \text{else} \end{cases} \tag{4}$$

for $t_i, T_h \in \mathcal{T}$ with t_i as the node threshold value and T_h as the subset of all thresholds in $\mathcal{T}$ associated with feature h. Where the bounding values are not defined, for example, when a feature is used in exactly one decision node (i.e. $\{t|t \neq t_i \text{ and } t \in T_h\} = \emptyset$), the maximum or minimum values of the feature are calculated from available data. Without data, the anticipated feature bounds can be provided as *a priori* estimates.



Smoothing function g maps threshold distance to weight contribution. Parameter α governs the impact of distance on the resulting weight: large values (blue dotted line) apply weight at larger distances while smaller values (purple dash-dot line) apply weight at even short distances.

Figure 2. Smoothing function distance to weight mapping

2.2 Smoothing Function

Applying the normalized threshold distance as a raw weight results in a linear interpolation over the threshold. While this is reminiscent of linear regression trees, it is unlikely to be an optimal representation of the unseen distribution. Instead, a smoothing function is introduced to allow nonlinear transitions of the prediction surface across the decision thresholds. The function defines a mapping from distance, d, to a weight on the interval $[0, 1] \in \mathbb{R}$. The smoothing function in this work was selected empirically to allow for control of the distance-weight relationship through a single parameter. Accordingly, the function is defined as

$$g(d; \alpha) = \frac{d^\alpha + 1}{2} \tag{5}$$

where g is parameterized by a value $\alpha \geq 0$.

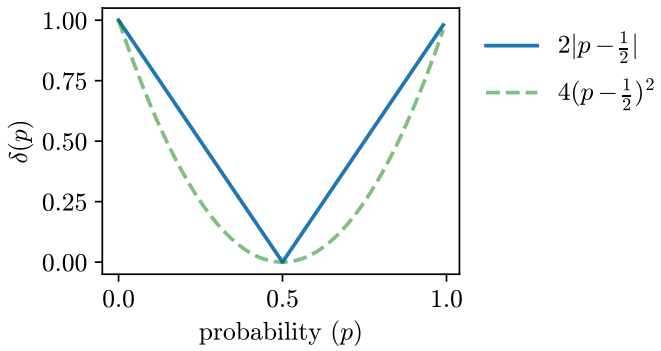
As illustrated in Figure 2, the parameter α influences the impact of distance on the final weighting. At a distance of zero, the function produces a weight of 0.5, reflecting a balanced consideration between branches as defined in (1). For an α value of one, the resulting weights scale linearly with the distance. As the value is increased, larger distances are required to receive the full weight value. Conversely, as α approaches zero, even small distances receive considerable weight. When α is zero, the full weight is applied regardless of threshold distance, mirroring a standard decision tree.

2.3 Branch probability

It is well known that decision trees are prone to overfit training data as tree depth increases [10]. Assuming the training data are representative of the true distribution[4], overfit subtrees are identified within the structure of the tree as branches with comparatively few training samples[5].

For a given decision node, assume $s_{\mathrm{L}}, s_{\mathrm{R}} \in \mathcal{S}$ are the counts of training samples arriving at the left and right child branches, respectively. The unconditional probability of selecting the left branch is then $p_{\mathrm{L}} = s_{\mathrm{L}}(s_{\mathrm{L}} + s_{\mathrm{R}})^{-1}$ and the probability of the right branch is $p_{\mathrm{R}} = 1 - p_{\mathrm{L}}$. When deriving an aggregate result (Section 2.4), these probabilities modify the branch weight to amplify or attenuate the threshold distance component.

Intuitively, with this approach, low probability branches correspond to reduced contribution to the final result thereby mitigating the impact of outliers and deep trees on prediction generalization. To avoid unnecessary probability weighting, the influence of the probability is adjusted such that it becomes strongest as p approaches 0 or 1. The probability effect coefficient is defined as $\delta = 2|p - \frac{1}{2}|$.



The shape of the function (blue solid line) ensures an adjustment coefficient, $\delta(p)$, of zero when the branches are balanced. An alternative function (green dashed line) offers a lower adjustment coefficient in the probability region around the balance point ($p \approx 0.5$).

Figure 3. Probability adjustment functions

In this formulation, as the value of p nears 0.5, the coefficient (and resulting adjusted impact of probability) reaches a minimum, $\delta = 0$, due to the equivalency between branches. The relationship between the adjustment coefficient and the probability is illustrated in Figure 3, along with an alternative, $4(p - \frac{1}{2})^2$, which requires more distance from the equilibrium point to apply an effect.

2.4 Aggregate prediction

While branch probabilities introduce a defense to overfitting, it is not always desirable to incorporate these values into an aggregate output[6]. To facilitate probability-free results, a parameter $\beta \in \mathbb{R}$, on the interval $[0, 1]$, is introduced. The β parameter controls a branch weight's relative contributions from the distance threshold and the probability. For a β value of one, aggregate predictions ignore the weight contribution from distance. When β is zero, the contribution of branch probability is ignored. The final weight assigned to a branch at a given node is then

[4]Trees trained with out-of-distribution data may see limited or no improvement from the soft tree approach.

[5]Relative probability comparison is ineffective in fully-overfit trees (e.g., trees containing exactly one training sample per leaf).

[6]In particular, it is preferable to eschew probability weights when it is expected or known that the decision tree is not overfit.

$$w = \beta\delta p + [1 - \beta\delta]\, g(d) \tag{6}$$

where p and $g(d)$ are evaluated from the perspective of the same branch (e.g. $x_i \leq t_i$). The alternate branch weight at the decision node is derived from the symmetry of (1).

The depth at which the soft tree method is applied to the decision tree has a significant influence on the consequent aggregate value. While softening a prediction using the entire tree is possible, this may not result in an optimal prediction. In order to specify the aggregation depth, a parameter λ is introduced residing on the interval $[0, k] \in \mathbb{N}$ where k is the tree depth. The λ parameter is indexed from the bottom of the decision path upward (leaf nodes reside at level zero) and specifies the root of the subtree considered. At $\lambda = k$, the entire tree contributes to the prediction.

Given an aggregation depth $\lambda > 0$ and input vector x, the final softened prediction is the result of a weighted combination of the child branches applied recursively[7] as given by

$$\hat{f}(x; \lambda) = w \cdot \hat{f}_{\mathrm{L}}(x; \lambda - 1) + (1 - w) \cdot \hat{f}_{\mathrm{R}}(x; \lambda - 1) \tag{7}$$

where $\hat{f}_{\mathrm{L}}$ and $\hat{f}_{\mathrm{R}}$ are the applications of the soft trees algorithm to the subtree of the left and right child branches with weight w as defined in (6). At the leaf nodes, the soft tree produces the same value as the underlying decision tree; $\hat{f}_j(x; \lambda = 0) = v_j$ where $v_j \in \mathcal{V}$ is the predicted value of the j^{th} leaf in the decision tree.

[7] Traversing the subtree results in a computational complexity of $O(n2^{\lambda})$ for predicting n samples at aggregation depth λ.

3. EXPERIMENTATION

In order to explore applications of the soft trees method, we designed and executed a set of comparative experiments. The experiments target common benchmark data sets using standard decision trees, soft trees, and random forests. The standard decision trees provide both an expected floor of performance and the basis for softening. Similarly, random forests yield tree-based ensemble results which represent an expected upper performance bound for the soft tree method. In addition to exercising the softening approach, the experiments serve as a baseline evaluating future enhancements.

We selected open-source benchmark data sets to represent a diverse set of regression tasks. These sets comprise a variety of sample sizes, numbers of features, data types, and application domains to facilitate analysis of comparative approach characteristics. The details of each data set are summarized in Table 1. Additional detail about the data and preparation is found in Appendix A.

Table 1. Benchmark data sets summary.

Data set	Feature types	Number of features	Number of samples
Accelerometer	Real, Categ.	5	153,000
Air Quality	Real	15	9,358
Diabetes	Real, Categ.	10	442
Friedman 1	Real	5	100,000
Friedman 2	Real	4	100,000
Friedman 3	Real	4	100,000
Red Wine	Real	12	4,898
Tetouan Power	Real	9	52,416
Titanic	Real, Categ.	10	331

While the value of the experimentation lies in comparing relative performance, reasonably performant models more accurately reflect real-world application. Parameterization of the algorithms has a dramatically effects the performance and generalization of the resulting models. To produce locally-optimal models, we subjected each algorithm to a grid search over a subset of the parameter space for each data set. The selected parameters identified in this search are detailed in Appendix B.

For the standard decision tree and random forest algorithms, we fit a model to a subset of each data set using the identified locally-optimal parameters, then subsequently evaluated those models on holdout subsets. In order to provide a measure of uncertainty, we repeated the process over 10 folds. The sample composition for the individual folds is held constant for each model's experiments to maintain parity. We then "fit" soft trees[8] to each trained decision tree and evaluate the predictions on the corresponding holdout sets. To avoid the confounding influence of divergent scales in the dependent variables, we selected the coefficient of determination (R^2) as the predictive performance evaluation criterion.

[8]Soft trees are not *trained* on data in the traditional sense, but rather modify predictions of pre-trained trees.

This page is intentionally blank.

4. RESULTS

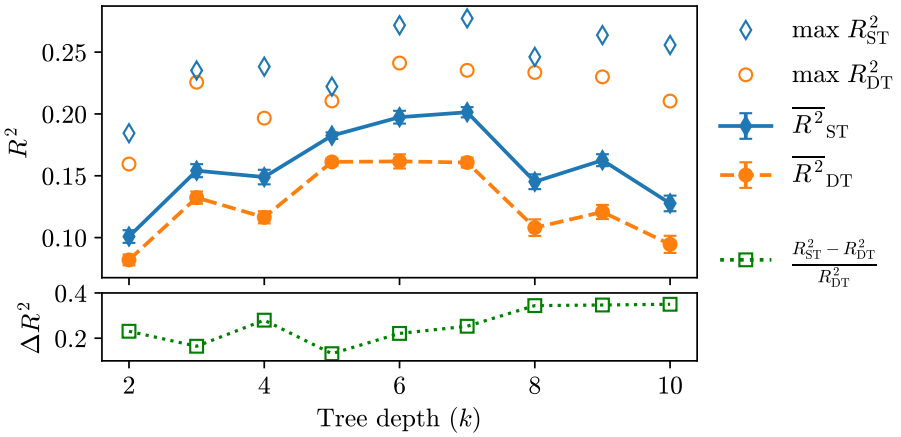
The results of the soft trees comparative experimentation are shown in Table 2. The identified uncertainties in the mean R^2 values are derived from the corresponding standard deviations over 10 folds of cross validation. As expected, the random forest models outperform both decision and soft tree approaches, occasionally by a wide margin. Soft trees meet or exceed the performance of standard decision trees for all data sets, and we observe a general increase in mean performance ranging up to $+10\%$, excluding one extreme outlier. The variances in the soft trees' R^2 performance are approximately equivalent to those of the associated decision trees, excepting the Titanic data set. This data set also realizes the largest improvement in mean performance at $+70\%$; however, the large variance places this result within the uncertainty. The unstable performance in this data set is due to the small number of samples per fold and large variability in the per-sample elements predictive of the target.

Table 2. Predictive performance results from comparative experimentation

Data set	Decision tree	Random forest	Soft tree	Pct. Improvement
Accelerometer	0.803± 0.002	0.839± 0.001	0.805± 0.002	0.20%
Air Quality	0.770± 0.025	0.880± 0.004	0.804± 0.034	4.44%
Diabetes	0.306± 0.072	0.433± 0.065	0.336± 0.056	9.98%
Friedman 1	0.353± 0.003	0.464± 0.003	0.367± 0.004	4.00%
Friedman 2	0.458± 0.004	0.471± 0.003	0.463± 0.004	1.10%
Friedman 3	0.080± 0.002	0.088± 0.002	0.080± 0.010	0.53%
Red Wine	0.269± 0.040	0.427± 0.028	0.284± 0.032	5.74%
Tetouan Power	0.380± 0.006	0.560± 0.005	0.388± 0.006	2.18%
Titanic	0.191± 0.379	0.423± 0.057	0.326± 0.137	70.38%

The mean R^2 performance is provided for each evaluated methodology, for all benchmark data sets, aggregated over 10 folds of cross-validation. The improvement of mean soft tree R^2 performance over decision tree performance is given as a percentage. Soft trees improve over decision trees for all data sets and maintain similar variance. The uncertainty shown is one standard deviation of the observed values.

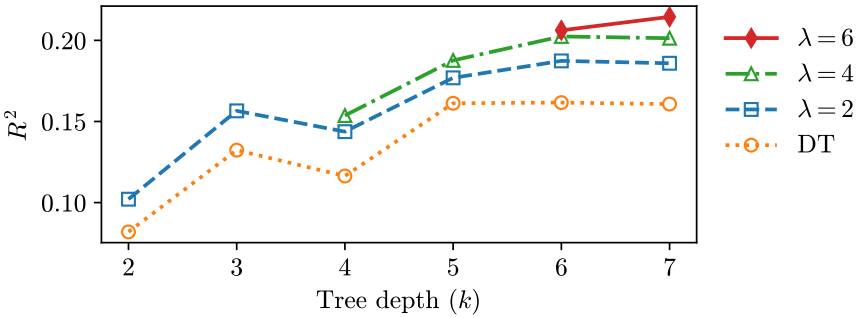
The data sets where the soft tree approach achieves the lowest improvement—Accelerometer and Friedman 3—also exhibit similar mean R^2 values between decision trees and random forests. Given the performance similarity, this indicates the decision tree is near-optimal leaving little room for improvement by the soft tree method. Likewise, as the complexity of the data-generating process increases, the decision tree performance decreases, highlighting the limits of single-tree approximation. This constraint in model expressiveness necessarily impacts the performance improvement capability of the soft trees method, as illustrated by the R^2 value progression in the Friedman data sets; as the complexity of the Friedman functions increase (Appendix A), the soft tree improvement percentage decreases.



Comparison of R^2 values for soft trees (ST) and decision trees (DT) as a function of tree depth. The mean soft tree R^2 performance (solid blue line) is greater than decision trees (dashed orange line) at all depths, however the improvement ratio of soft trees over decision trees (lower plot, green dotted line) continues to grow with depth. The uncertainty is given by the standard error of the mean.

Figure 4. Soft tree vs. decision tree R^2 performance

The depth of the decision tree plays a large role in determining the performance of the tree and subsequent efficacy of the smoothing; under- and over-fit trees will inevitably limit the overall performance capability of soft trees. Figure 4 plots the impact of depth on soft trees using the Red Wine data set. The soft tree approach improves over the baseline decision tree at all depths, but performs optimally where the decision tree performs optimally (recall that soft trees *modify* decision tree prediction). However, the percent improvement relative to the standard tree actually continues to *increase* with the depth, even as overall performance decreases.

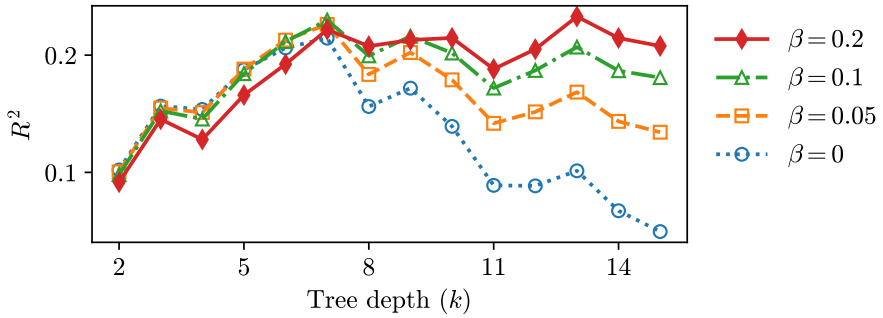


Soft tree mean R^2 performance by tree depth for various aggregation depths (λ). As tree depth increases, additional soft tree performance is gained relative to the baseline decision tree (DT, orange dotted line) by aggregating larger subtrees (i.e. where $\lambda \to k$).

Figure 5. Soft tree mean R^2 performance for various λ

The performance impact of aggregation depth, λ, is similarly dependent on the depth of the decision tree. We demonstrate the effect of λ at various tree depths in Figure 5. While all values of λ improve the baseline decision tree performance, the improvement is greater when larger subtrees are considered in the prediction. Incorporating the entire tree (or most of the tree) in a prediction may seem counterintuitive,

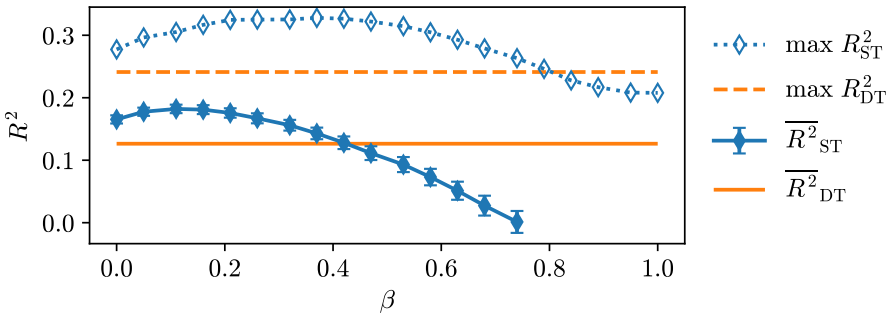
however recall that node contributions are weighted by threshold distance, so branches far from the decision path impart minimal influence.



Soft tree mean R^2 performance by tree depth for various probability balance parameter values (β). As tree depth increases, more benefit is realized from the incorporation of probability. When probability is not considered (blue circles), performance decreases as the underlying decision tree overfits ($k > 7$).

Figure 6. Soft tree mean R^2 performance for various β

The results also indicate that incorporating probability into the prediction is generally unnecessary. Only two data sets—Diabetes and Titanic—use the β parameter to improve performance. Deeper trees may experience greater performance benefit from probability application, as shown in Figure 6. In this example, larger β values reduce mean performance until the tree depth reaches a tipping point at $k = 7$. As tree depth increases, larger values of β are required to avoid a performance decline coincident with the increasingly overfit decision tree. Conversely, for depths under the tipping point, the increased integration of branch probabilities at $\beta = 0.2$ diminishes the expected R^2 value as compared to lower values of β.



Mean and maximum R^2 performance for decision (DT) and soft trees (ST) as a function of the probability balance parameter, β. Diminishing returns are seen in mean performance (solid blue diamonds) for $\beta > 0.15$ although maximum performance (empty blue diamonds) remains elevated for larger β values. The uncertainty is given by the standard error of the mean.

Figure 7. Soft tree mean and max performance by β

Well-fit decision trees (i.e., those with depth equal to the tipping point), are also susceptible to the influence of β. In such cases, preferring probability to threshold distance results in diminishing returns in performance as illustrated in Figure 7. Larger β values begin to reduce the maximum performance improvement and eventually *degrade* performance as compared to standard decision trees. The mean soft

tree performance deteriorates even faster, with $R^2_{\mathrm{ST}} \leq R^2_{\mathrm{DT}}$ when $\beta \geq 0.4$. Such behavior is not unexpected; well-fit trees have no outlier leaves, by definition. Accordingly, any smoothing achieved by incorporating larger β probability in these trees results from discarding more useful threshold distance information, thereby reducing predictive performance.

5. CONCLUSIONS

The soft tree method introduced in this work represents a novel approach to improving the performance of decision tree regressors. Given a pre-trained decision tree, a soft tree leverages meta-features of the model to smooth prediction values across decision thresholds, without requiring training data. The weighted predictions are further influenced by parameters controlling the aggregation depth, probability impact, and smoothing function shape.

We demonstrated the soft tree approach on a varied collection of common benchmark data sets and observed an average of 11% improvement in the R^2 value over the underlying decision trees. We show that the magnitude of the expected improvements offered by the soft trees depend on the complexity of the modeled process and the accuracy of the decision tree approximation. Additionally, we show how the aggregation depth improves performance as tree depth increases and how the incorporation of probability mitigates the effects of overfitting in deep trees.

Future work broadly resides in two categories: extension and interpretability. Work continues in extending the soft trees methodology to decision tree classification and ensemble methods. These enhancements, along with missing feature tolerance and per-node parameterization, represent promising avenues of exploration. Finally, as an inherent feature of decision trees, it is important to recover interpretability via the soft tree method. While approximate prediction explanations can be extracted from the decision paths, weighted aggregation often obscures full explicability. Continuing efforts seek to identify a robust, model-native explainability framework under the soft tree approach.

This page is intentionally blank.

A. DATA SETS

We retrieved all data sets from their associated sources with the exception of sets Friedman 1, Friedman 2, and Friedman 3 [11]. These data are generated from noisy processes defined in (8), (9), and (10), respectively. Each Friedman set was created with $100{,}000$ random samples.

Friedman 1 produces data from a polynomial and sine transform with noise $\epsilon = 5$.

$$y(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon \cdot \mathcal{N}(0,1) \tag{8}$$

Friedman 2 produces data from a multiplicative and reciprocal transform with noise $\epsilon = 400$.

$$y(x) = \left[x_1^2 + (x_2 x_3 - \frac{1}{x_2 x_4})^2\right]^{\frac{1}{2}} + \epsilon \cdot \mathcal{N}(0,1) \tag{9}$$

Friedman 3 produces data from an arctangent, multiplicative, and reciprocal transform with noise $\epsilon = 1$.

$$y(x) = \tan^{-1}\left[\frac{x_2 x_3 - (x_2 x_4)^{-1}}{x_1}\right] + \epsilon \cdot \mathcal{N}(0,1) \tag{10}$$

Within each data set, we identified a real-valued regression target. The Accelerometer[9] data set [12] targets the cooler fan RPM speed percent value. The hourly averaged sensor response for tin oxide is the target for the Air Quality[9] data set [13]. The Diabetes[9] set [14] targets a quantitative measure of disease progression after one year. The wine quality measure is the target for the Red Wine[9] data set [15]. The Tetouan Power[9] data set [16] targets zone 1 power consumption. The passenger fare is selected as the target of the Titanic[10] data set [17]. The targets of the Friedman sets are specified in their respective equations. We performed a basic set of preparation steps on each data set; we encoded categorical features in a set of dummy variables ("one-hot encoding") and applied a standard scaler to real-valued features.

[9] Creative Commons Attribution 4.0 International
[10] Public domain

This page is intentionally blank.

B. MODEL PARAMETERS

The locally-optimal parameters for each algorithm under test are identified using a 5-fold cross-validated grid search. The decision tree, random forest, and parameter grid search use implementations from Scikit-learn [18]. The parameters discovered in the search are outlined in Table 3.

Table 3. Model hyperparameters for each data set

	Decision Tree			Soft Tree			Random Forest			
Data set	Max depth	Max features	Max samp. split	α	β	λ	Max depth	Max features	Max samp. split	Num. estimators
Accelerometer	10	1	5	1	0	10	18	0.5	7	160
Air Quality	3	0.3	5	1	0	3	6	0.5	7	130
Diabetes	3	1	3	0.5	0.5	2	11	0.3	7	70
Friedman 1	5	1	2	0.5	0	5	13	0.4	7	170
Friedman 2	5	1	2	4	0	4	7	0.8	4	90
Friedman 3	5	1	2	0.5	0	5	7	0.7	7	90
Red Wine	5	0.6	3	0.5	0	4	7	0.5	5	130
Tetouan Power	10	1	3	4	0	8	25	0.9	2	170
Titanic	3	0.6	5	0.5	0.9	3	3	1	5	110

Model parameters selected for experimentation for each evaluated algorithm, for all benchmark data sets.

This page is intentionally blank.

REFERENCES

1. Breiman, L. 2001. "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32.

2. Biau, G. and Scornet, E. 2016. "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227.

3. Friedman, J. H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232.

4. Freund, Y. and Schapire, R. E. 1997. "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139.

5. Cao, Y., Miao, Q.-G., Liu, J.-C., and Gao, L. 2013. "Advance and Prospects of AdaBoost Algorithm," *Acta Automatica Sinica*, vol. 39, no. 6, pp. 745–758.

6. Quinlan, J. R. 1992. "Learning with continuous classes," *5th Australian Joint Conference on Artificial Intelligence*, vol. 92 (pp. 343–348). World Scientific.

7. Kumar, A., Goyal, S., and Varma, M. 2017. "Resource-efficient Machine Learning in 2 KB RAM for the Internet of Things," D. Precup and Y. W. Teh, eds., *Proceedings of the 34th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 70 (pp. 1935–1944), PMLR.

8. Wahba, G. 1990. *Spline models for observational data*, Society for Industrial and Aapplied Mathematics.

9. Goepp, V., Bouaziz, O., and Nuel, G. 2018. "Spline Regression with Automatic Knot Selection," Doi: 10.48550/arXiv.1808.01770.

10. Mitchell, T. M. 1997. *Machine Learning*, McGraw-Hill, Inc., USA, 1 ed.

11. Friedman, J. H. 1991. "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67.

12. Scalabrini Sampaio, G., Rabello de Aguiar Vallim Filho, A., Santos de Silva, L., and Augusto da Silva, L. 2023. "Accelerometer," UCI Machine Learning Repository, doi: 10.24432/C5Q61V.

13. Vito, S. 2016. "Air Quality," UCI Machine Learning Repository, doi: 10.24432/C59K5F.

14. Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. 2004. "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407 – 499.

15. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. 2009. "Wine Quality," UCI Machine Learning Repository, doi: 10.24432/C56S3T.

16. Salam, A. and El Hibaoui, A. 2018. "Comparison of Machine Learning Algorithms for the Power Consumption Prediction: Case Study of Tetouan City," *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)* (pp. 1–5). IEEE.

17. Cukierski, W. 2012. "Titanic - Machine Learning from Disaster," URL `https://kaggle.com/competitions/titanic`.

18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. 2011. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.

This page is intentionally blank.

INITIAL DISTRIBUTION

84310	Technical Library/Archives	(1)
71740	Joshua A. Duclos	(1)
53629	Benjamin A. Michlin, Ph.D.	(1)
71780	Andrew B. Sabater, Ph.D.	(1)
53629	Jamal T. Rorie, Ph.D.	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

This page is intentionally blank.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10-06-2024	Final	

4. TITLE AND SUBTITLE

Softening the Prediction Surface of Decision Tree Regressors

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Joshua A. Duclos
Benjamin A. Michlin, Ph.D.
Andrew B. Sabater, Ph.D.
Jamal T. Rorie, Ph.D.
NIWC Pacific

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NIWC Pacific
53560 Hull Street
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-3350

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

The NIWC Pacific Naval Innovative Science and Engineering (NISE) Program
53560 Hull Street
San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

NISE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This work was funded under a NISE Applied Research project.

14. ABSTRACT

While decision trees function as base-learners for many machine learning methods and exhibit several useful properties, the compromise between prediction accuracy and generalization limits their utility in non-ensemble applications. We propose a new method to improve the predictive performance of pre-trained decision tree regressors. Using the tree structure and metadata, we derive a set of decision threshold-based weights that modify the leaf prediction values. The weighted values are then aggregated into a final "softened" prediction which more accurately represents the true target distribution. We demonstrate the approach on a variety of benchmark data sets and observe a mean improvement of 11% over the baseline decision tree R2 values. We further explore the parameters of the approach and characterize their effects.

15. SUBJECT TERMS

decision tree regression, prediction generalization, soft tree

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Josh Duclos
U	U	U	SAR	34	19b. TELEPHONE NUMBER *(Include area code)* 619-767-4989

This page is intentionally blank.

This page is intentionally blank.

Naval Information Warfare Center (NIWC) Pacific
San Diego, CA 92152-5001