Online Appendix to:
Random Forests for Benefit Transfer

October 28, 2024

Abstract

The material contained herein is supplementary to the paper named in the title.

The mechanics of a simple regression tree

A stylized example of a regression tree is given in figure A1, generated with `R` packages `rpart` and `tree_diagram`. Assume the outcome of interest is WTP (in dollars), and the available features are context-specific variables, as used in our empirical application. As can be seen in the top, or "root" node, the initial mean WTP equals $141, and, of course, applies to 100% of the sample at this initial stage. The tree then decides that the largest gain in predictive accuracy, as measured by MSE, is achieved by splitting the sample along the binary indicator "fishing affected," which takes a value of one if the proposed water quality change enhances recreational fishing opportunities. Observations for which this condition holds are sent down the right, or "yes" branch of the tree. As indicated by the corresponding "child" node, mean WTP equals $294 for these cases, comprising 19% of the initial sample. Furthermore, the tree finds no other feature and split point that could bring further gains in fit for this segment. Thus, the node becomes a terminal "leaf," as is evident from the tree diagram. However, for the opposite segment of "fishing affected = 0" cases (81% of the full sample), fit can be further improved by splitting along the binary indicator "payment vehicle = tax." In this case the right or "yes" branch refers to the "0" cases, as indicated by the splitting label, and the left branch to the "1" cases. For the latter, this leads to another terminal leaf, with average WTP of $72 and including 39% of the initial sample. For the "yes" group (41% of the sample), two additional splits are implemented involving features "log of population income" (with split point of 11), and a binary indicator for the location of a source study in the southern U.S. census region. Once no further accuracy gains can be obtained by more splitting, or leaf sizes reach a predetermined minimum (usually in the five to 10 range), the tree is fully grown, and every single observation is allocated to a terminal leaf.

The tree can now be used to generate predictions for a new sample point, by sending the new observation along the branches corresponding to its feature values until it drops into one of the terminal leaves. For example, a new point with "fishing affected" = 0, "payment vehicle" = access fee (not a tax), "log income" = 10.2, and "south census" = 0 will end in leaf 20, which also hosts 26% of the original or "training" sample. The predicted WTP for the new point is then simply the mean WTP for that leaf, in this case $86.

Interpretation of RF predictions as the solution to a weighted regression

The result in the last line of equation (7) in the main text can also be interpreted as the solution to a weighted OLS regression of $\tilde{\mathbf{y}}$ on a constant, i.e. the sought prediction $\tilde{y}_p$. Specifically, letting $\tilde{\mathbf{A}}$ be a diagonal matrix featuring $\sqrt{\alpha_i\left(\mathbf{x}_p, \mathbf{X}\right)}$ as the i^{th} diagonal element, and $\mathbf{A} = \tilde{\mathbf{A}} * \tilde{\mathbf{A}}$ as the corresponding diagonal matrix with $\alpha_i\left(\mathbf{x}_p, \mathbf{X}\right)$ as the i^{th} diagonal element, we can specify the regression model as

$$\tilde{\mathbf{A}}\tilde{\mathbf{y}} = \tilde{\mathbf{A}}\mathbf{i}\tilde{y}_p + \boldsymbol{\epsilon},$$

where $\mathbf{i}$ is an n by 1 vector of ones. The OLS solution then emerges as

$$\begin{aligned}
&\hat{\tilde{y}}_p = \left(\mathbf{i}'\mathbf{A}\mathbf{i}\right)^{-1}\mathbf{i}'\mathbf{A}\tilde{\mathbf{y}} = \\
&\frac{1}{\sum_{i=1}^{n}\alpha_i\left(.\right)}\sum_{i=1}^{n}\alpha_i\left(.\right)\tilde{y}_i = \sum_{i=1}^{n}\alpha_i\left(.\right)\tilde{y}_i \quad \text{since} \\
&\sum_{i=1}^{n}\alpha_i\left(.\right) = 1
\end{aligned}$$

The last condition can be easily derived as (dropping the functional relationship of α_i with the underlying data for ease of exposition):

$$\begin{aligned}
&\sum_{i=1}^{n}\alpha_i = \\
&\sum_{i=1}^{n}\frac{1}{B}\sum_{b=1}^{B}\frac{I\left(\mathbf{x}_i \in L_b\left(\mathbf{x}_p\right)\right)}{|L_b\left(\mathbf{x}_p\right)|} = \\
&\frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n}\frac{I\left(\mathbf{x}_i \in L_b\left(\mathbf{x}_p\right)\right)}{|L_b\left(\mathbf{x}_p\right)|} = \\
&\frac{1}{B}\sum_{b=1}^{B}\frac{n_b}{|L_b\left(\mathbf{x}_p\right)|} = \frac{1}{B}B = 1,
\end{aligned}$$

where n_b denotes the number of observations that share a leaf with the policy point in tree b. By definition, this value must be equal to the number of elements in that leaf, i.e. $|L_b\left(\mathbf{x}_p\right)|$.

Check for optimal number of trees

For all three forest models (RF,LLF, LLF.cov) we examine how the out-of-bag Root MSE (RMSE) changes with forest size, i.e. the number of underlying trees. Specifically, we evaluate all forests for 250 to 4000 trees, in increments of 250. The standard forest (RF) is run without any additional tuning (as is the default for LLF and LLF.cov), to allow for a direct focus on the effect of forest size. Figure A2 shows that the RMSE stabilizes around 1000-1500 trees for all specifications. We are thus confident that our chosen forest size of 2000 is sufficient for all components of our analysis.

Robustness check for tuner settings in linear forests

As mentioned in the main text, standard RFs allow for automatic tuning of key parameters, such as the fraction of the data to be used to build a given tree (`sample.fraction`), the number of explanatory variables to be considered at each split (`mtry`), and the minimum node size (`min.node.size`). This automatic tuning feature is not (yet) available for linear forests with a penalized splitting rule, as used in our analysis. This implies that the analyst needs to chose these tuning parameters "manually." In our estimation, we adopt the suggested default settings of `sample.fraction` = 0.5, `mtry` = k (number of available explanatory variables) = 22, and min.node.size = 5.[1] Table A1 shows model fit results, akin to those presented in Table 2 of the main text, for different settings for these three primary tuners. Specifically, we examine MSE and MAPE statistics for the full permutation of: `sample.fraction` $\in \{, 0.4, 0.45, 0.5\}$, `mtry` $\in \{10, 15, 22\}$, and `min.node.size` $\in \{3, 5, 10\}$. These settings are chosen to balance algorithmic restrictions (`sample.fraction` must not exceed 0.5 for LLF-type forests), and typical bias-variance tradeoffs (smaller bias but larger variance for smaller `mtry` and `min.node.size` settings).

Table A1 gives the result of these robustness checks. The second-to-last row, highlighted in grey, captures the default setting used in our main analysis, and thus reproduces the entries in Table 2 for LLF and LLF.cov in the main text. As is clear from the table, the different tuner combinations lead to only minor changes in model fit, with default settings producing fit statistics that are located towards the lower (= better) end of the spectrum. We are thus confident that proceeding with default settings does not imply sacrificing measurable gains in predictive accuracy.

Adding-Up examination for larger quality steps

We implement three additional AU scenarios featuring large and / or uneven quality steps. Scenario one starts with a small step of 1.5 points from 41 to 42.5, followed by a large, 18.5-point step from 42.5 to 61. Scenario two reverses this order, and scenario three proceeds in two equal steps of ten points, starting again from a baseline of 41. Table A2 captures corresponding BT predictions and AU errors. As can be seen from the table, the GL-MRM, LWR, and LLF continue to exhibit negligible AU errors at or under one percent, while errors for the LLF.cov increase from approximately six percent for the large-large scenario to 17% for the small-large scenario. While a 17% error may still be acceptable in certain policy contexts, this highlights the importance of examining AU-related performance of forest models on a case-by-case basis in a given empirical application.

References for Online Appendix

Tibshirani, J., Athey, S., Sverdrup, E., Wager, S., 2024. The GRF Algorithm. Web site: `https://grf-labs.github.io/grf/REFERENCE.html`, last accessed 2024-10-11.

[1] As explained on in the online reference guide to the GRF package, the actual number of variables chosen for a given split is drawn from a Poisson distribution with parameter `mtry` (Tibshirani et al., 2024)

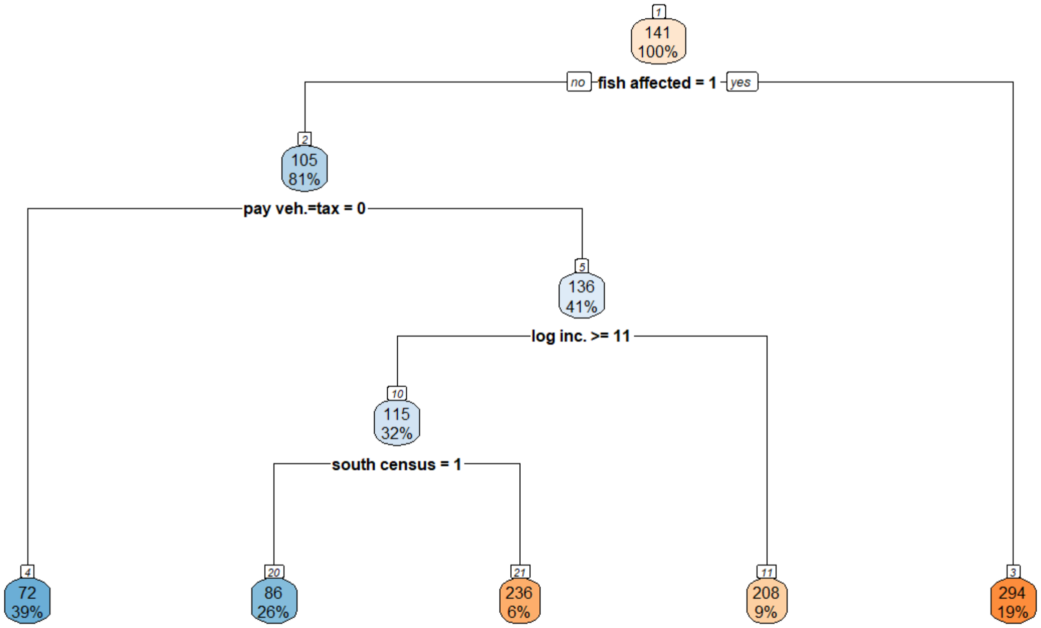
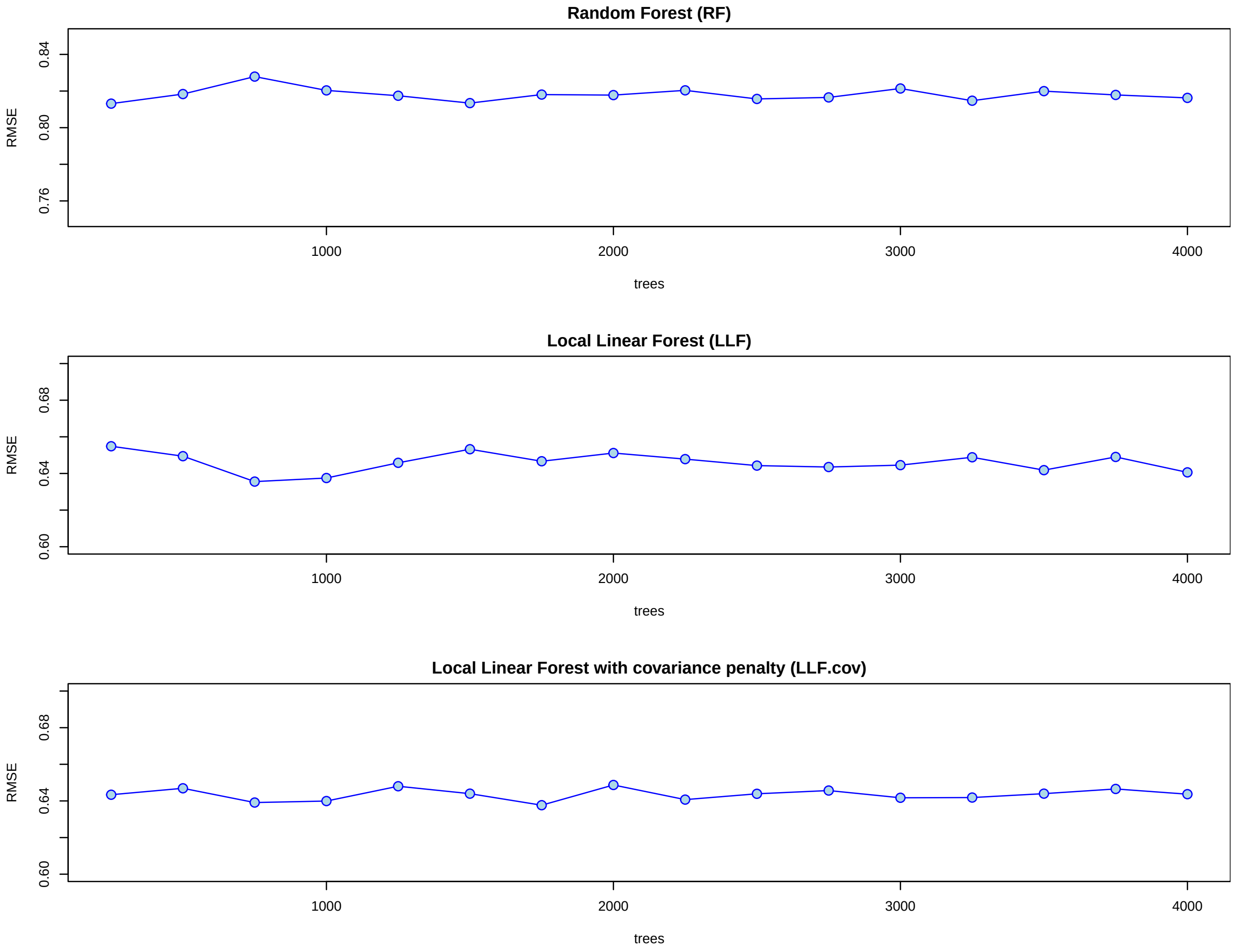


Figure A1: Example of a regression tree

The top entry in each node and leaf shows the mean of the outcome variable, the bottom entry the corresponding share of the sample.


Random Forest (RF)

Local Linear Forest (LLF)

Local Linear Forest with covariance penalty (LLF.cov)

Figure A2: Check for sufficient forest size (number of underlying trees)

Table A1: Robustness check for tuners in the linear forest models

tuning parameter			LLF		LLF.cov	
s.frac	mtry	min.node	mse	mape	mse	mape
0.4	10	3	8.385	57.273	6.785	55.909
0.4	10	5	8.354	58.598	6.630	56.369
0.4	10	10	8.939	61.786	6.763	56.666
0.4	15	3	8.379	58.422	6.449	55.483
0.4	15	5	8.219	58.846	6.365	55.517
0.4	15	10	9.199	61.642	6.476	57.005
0.4	22	3	6.973	55.789	6.303	54.835
0.4	22	5	7.599	57.342	6.870	56.534
0.4	22	10	8.439	60.731	6.293	55.887
0.45	10	3	9.034	58.569	6.590	54.952
0.45	10	5	8.878	59.226	6.556	55.697
0.45	10	10	9.066	61.020	6.517	56.269
0.45	15	3	7.138	55.707	6.524	55.401
0.45	15	5	7.408	56.728	6.684	56.182
0.45	15	10	8.528	60.049	6.394	56.436
0.45	22	3	7.713	56.265	6.329	54.929
0.45	22	5	7.729	56.507	6.486	55.719
0.45	22	10	8.593	60.425	6.068	55.348
0.5	10	3	7.998	58.252	7.056	56.137
0.5	10	5	8.603	60.235	6.730	56.670
0.5	10	10	9.758	62.121	6.606	56.700
0.5	15	3	8.028	56.549	6.460	54.356
0.5	15	5	8.192	57.658	6.495	55.194
0.5	15	10	8.950	61.298	6.223	55.807
0.5	22	3	7.181	55.035	6.583	55.198
0.5	22	5	7.707	56.642	6.379	55.230
0.5	22	10	8.144	59.557	6.133	55.886

s.frac = fraction of data used to build each tree
mtry = number of variables considered for each split
min.node = minimum node size
mse = Mean Squared Error (in 1000’s)
mape = Mean Absolute Percentage Error
LLF = Local Linear Forest (with residual splitting)
LLF.cov = LLF with covariance-adjusted ridge penalty

Table A2: Adding-up examination for larger quality steps

	GL-MRM			LWR			LLF			LLF.cov		
scenario (WQI)	low	mean	high	low	mean	high	low	mean	high	low	mean	high
41 to 42.5	0.286	1.657	6.048	0.375	1.353	3.496	0.404	0.538	0.671	1.342	1.531	1.720
42.5 to 61	3.154	18.009	65.581	3.948	13.942	36.737	3.918	5.756	7.594	9.905	11.357	12.809
41 to 61	3.548	19.654	69.898	4.251	15.408	41.117	4.255	6.229	8.203	13.528	15.527	17.526
adding-up error		0.06%			-0.73%			1.04%			-17.00%	
41 to 59.5	3.177	18.354	65.950	4.045	14.327	36.654	3.973	5.799	7.625	13.538	15.938	18.339
59.5 to 61	0.229	1.311	4.722	0.271	0.957	2.530	0.271	0.394	0.518	0.550	0.657	0.765
41 to 61	3.548	19.654	69.898	4.251	15.408	41.117	4.255	6.229	8.203	13.528	15.527	17.526
adding-up error		0.05%			-0.81%			-0.57%			6.88%	
41 to 51	1.820	10.470	37.860	2.255	8.301	21.918	2.333	3.325	4.317	8.497	9.747	10.996
51 to 61	1.611	9.224	33.426	1.927	6.951	18.147	2.011	2.924	3.838	4.016	4.859	5.703
41 to 61	3.548	19.654	69.898	4.251	15.408	41.117	4.255	6.229	8.203	13.528	15.527	17.526
adding-up error		0.20%			-1.01%			0.32%			-5.93%	

GL-MRM = Globally-Linear MRM
LWR = Locally-Weighted MRM
RF = Random Forest
LLF = Local Linear Forest (with residual splitting)
LLF.cov = LLF with covariance-adjusted ridge penalty