Econ 2148, fall 2019
Trees, forests, and causal trees

Maximilian Kasy

Department of Economics, Harvard University

Agenda

- Regression trees: Splitting the covariate space.
- Random forests: Many trees.
  Using bootstrap aggregation to improve predictions.
- Causal trees: Predicting heterogeneous causal effects.
  Ground truth not directly observable, for cross-validation.

Takeaways for this part of class

- Trees partition the covariate space and form predictions as local averages.
- Iterative splitting of partitions allows us to be more flexible in regions of the covariate space with more variation of outcomes.
- Bootstrap aggregation (bagging) is a way to get smoother predictions, and leads to random forests when applied to trees.
- Things get more complicated when we want to predict heterogeneous causal effects, rather than observable outcomes.
- This is because we do not directly observe a ground truth that can be used for tuning.

Regression trees

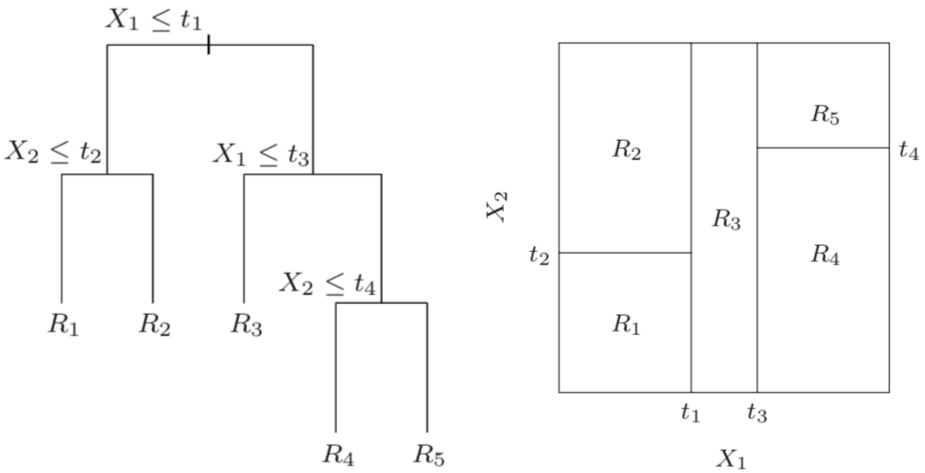
- Suppose we have i.i.d. observations (X_i, Y_i) and want to estimate $g(x) = E[Y|X = x]$.
- Suppose we furthermore have a partition of regressor space into subset $(R_1, \ldots, R_M)$.
- Then we can estimate $g(\cdot)$ by averages in each element of the partition:

$$\hat{g}(x) = \sum_m c_m \cdot \mathbf{1}(x \in R_m)$$

$$c_m = \frac{\sum_i Y_i \cdot \mathbf{1}(X_i \in R_m)}{\sum_i \mathbf{1}(X_i \in R_m)}.$$

- This is a regression analog of a histogram.

Recursive binary partitions



Constructing the partition

- How to choose the partition?
- Start with the trivial partition with one element.
- Greedy algorithm (CART): Iteratively split an element of the partition, such that the in-sample prediction improves as much as possible.
- That is: Given $(R_1, \ldots, R_M)$,
  - For each R_m, $m = 1, \ldots, M$, and
  - for each X_j, $j = 1, \ldots, k$,
  - find the $x_{j,m}$ that minimizes the mean squared error, if we split R_m along variable X_j at $x_{j,m}$.
  - Then pick the (m, j) that minimizes the mean squared error, and construct a new partition with $M + 1$ elements.
  - Iterate.

Tuning and pruning

- Key tuning parameter: Total number of splits M.
- We can optimize this via cross-validation.
- CART can furthermore be improved using "**pruning**."
- Idea:
  - Fit a flexible tree (with large M) using CART.
  - Then iteratively remove (collapse) nodes.
  - To minimize the sum of squared errors,
    plus a penalty for the number of elements in the partition.
- This improves upon greedy search.
  It yields smaller trees for the same mean squared error.

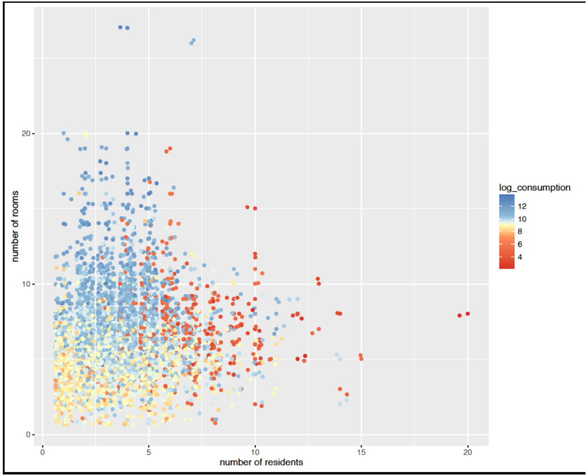
From trees to forests

- Trees are intuitive and do OK, but they are not amazing for prediction.
- We can improve performance a lot using either bootstrap aggregation (bagging) or boosting.
- **Bagging:**
  - Repeatedly draw bootstrap samples $(X_i^b, Y_i^b)_{i=1}^n$ from the observed sample.
  - For each bootstrap sample, fit a regression tree $\hat{g}^b(\cdot)$.
  - Average across bootstrap samples to get the predictor

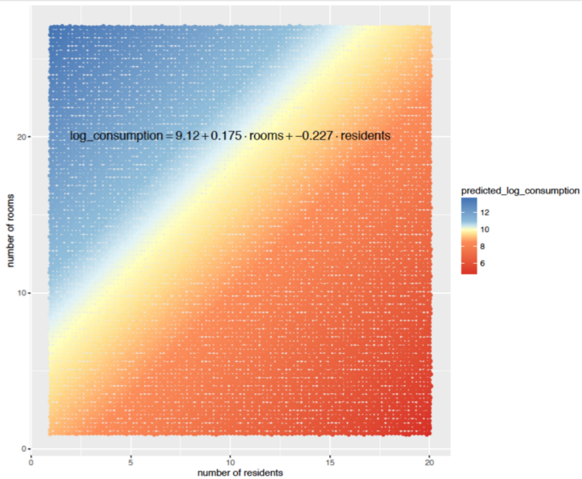
$$\hat{g}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{g}^b(x).$$

- This is a technique for smoothing predictions.
  The resulting predictor is called a "random forest."
- Possible modification:
  Restrict candidate splits to random subset of predictors in each tree-fitting step.

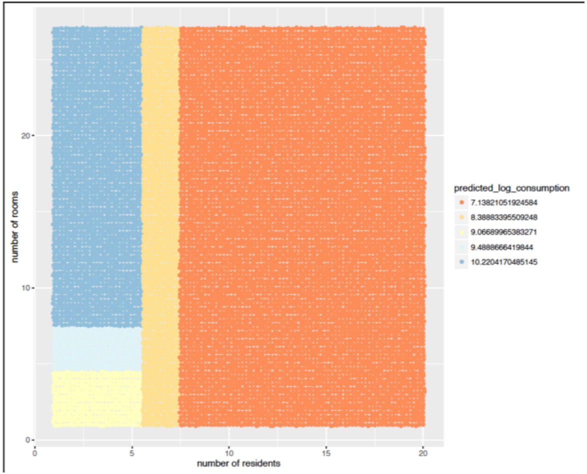
An empirical example (courtesy of Jann Spiess)



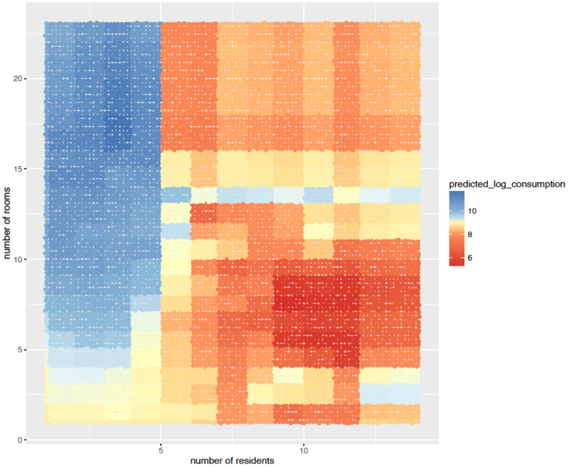
OLS


log_consumption = 9.12 + 0.175 · rooms + −0.227 · residents
number of rooms
number of residents
0
10
20
0
5
10
15
20
predicted_log_consumption
12
10
8
6

Regression tree



Random forest



Causal trees

- Suppose we observe i.i.d. draws of (Y_i, D_i, X_i), and wish to estimate
$$\tau(x) = E[Y|D = 1, X = x] - E[Y|D = 0, X = x].$$
- Motivation: This is the conditional average treatment effect under an unconfoundedness assumption on potential outcomes,
$$(Y^0, Y^1) \perp D|X.$$
- This is relevant, in particular, for targeted treatment assignment.
- We might, for a given partition $\mathscr{R} = (R_1, \ldots, R_M)$, use the estimator
$$\hat{\tau}(x) = \sum_m \left(c_m^1 - c_m^0\right) \cdot \mathbf{1}(x \in R_m)$$
$$c_m^d = \frac{\sum_i Y_i \cdot \mathbf{1}(X_i \in R_m, D_i = d)}{\sum_i \mathbf{1}(X_i \in R_m, D_i = d)}.$$

Targets for splitting and cross-validation

- Recall that CART uses greedy splitting.
  It aims to minimize in-sample mean squared error.
- For tuning, we proposed to use the out-of-sample mean squared error in order to choose the tree depth.
- Analog for estimation of $\tau(\cdot)$: Sum of squared errors,

$$SSE(\mathscr{S}) = \sum_{i \in \mathscr{S}} \left((\tau_i - \hat{\tau}(X_i))^2 - \tau_i^2 \right),$$

  where $\mathscr{S}$ is either the estimation sample, or a hold-out sample for cross-validation.
  (The term τ_i^2 is added as a convenient normalization.)
- Problem: τ_i is not observed.

Targets continued

- Solution: We can rewrite $SSE(\mathscr{S})$,

$$SSE(\mathscr{S}) = \sum_{i \in \mathscr{S}} \left(\hat{\tau}(X_i, \mathscr{R}) \cdot \left(\hat{\tau}(X_i, \mathscr{R})^2 - 2\tau_i \right) \right).$$

- Suppose we split our sample into $(\mathscr{S}^1, \mathscr{S}^2)$, use $\mathscr{S}^1$ for estimation, and $\mathscr{S}^2$ for tuning. Let $\hat{\tau}_j(X, \mathscr{R})$ be the estimator based on sample $\mathscr{S}^j$.
- An unbiased estimator of $SSE(\mathscr{S}^2)$ (for tuning) is then given by

$$\widehat{SSE}(\mathscr{S}^2) = \sum_{i \in \mathscr{S}} \left(\hat{\tau}_1(X_i, \mathscr{R}) \cdot \left(\hat{\tau}_1(X_i, \mathscr{R})^2 - 2\tau_2(X_i, \mathscr{R}) \right) \right).$$

- An analog to the in-sample sum of squared errors (for CART splitting) is given by

$$\widehat{SSE}(\mathscr{S}^1) = \sum_{i \in \mathscr{S}} \left(-\hat{\tau}_1(X_i, \mathscr{R})^2 \right).$$

References

- *Friedman, J., Hastie, T., and Tibshirani, R. (2001).* The elements of statistical learning, *volume 1. Springer series in statistics Springer, Berlin, chapters 8 and 9.*
- *Athey, S. and Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects.* Proceedings of the National Academy of Sciences, *113(27):7353–7360.*