Lecture 3: Bagging, Boosting, and Tree-Based Models

Isaiah Hull[1,2]

[1]BI Norwegian Business School
[2]CogniFrame

November 14, 2023

Introduction

Lecture 3: Overview

1. Introduction to Tree-Based Models.
2. Applications.

1. Introduction to Tree-Based Models

Introduction to Tree-Based Models

Overview

- Based on James et al. (2023) and Hull (2021).
  - https://hastie.su.domains/ISLP/ISLP_website.pdf.viewingoogle.html
  - https://link.springer.com/chapter/10.1007/978-1-4842-6373-0_4

Introduction to Tree-Based Models

Tree-Based Methods Overview

- Tree-based methods used for regression and classification.
- Segment the predictor space into simple regions.
- Make predictions based on mean/mode response value for the region.
- Splitting rules summarized in a tree: known as decision tree methods.

Introduction to Tree-Based Models

Advantages and Disadvantages

1. Advantages: Simple and useful for interpretation.

2. Limitations: Not as accurate as some other supervised learning approaches.

3. Solutions for better accuracy: Bagging, random forests, boosting, Bayesian additive regression trees.

Introduction to Tree-Based Models

Basics of Decision Trees

- Applicable to both regression and classification.
- Start with classification example.

Introduction to Tree-Based Models

Understanding Decision Trees

- A decision tree consists of branches and three types of nodes: roots, internal nodes, and leaves.
- The root is where the first sample split occurs.
- Each split is associated with a branch.
- Internal nodes further split the sample.
- The tree terminates at the leaves, which yield predictions or probability distributions.

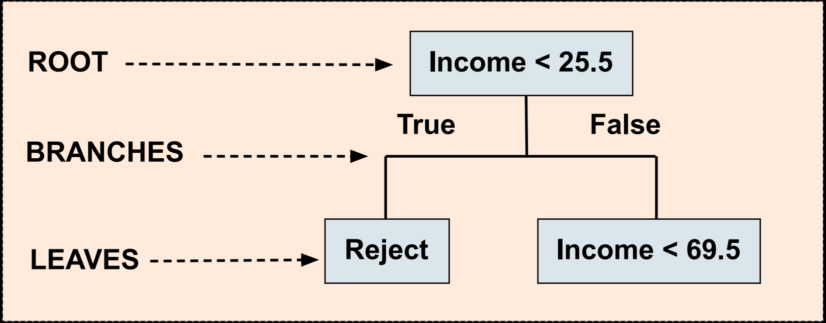
Introduction to Tree-Based Models

HMDA Mortgage Data: An Example

- Objective: Predict mortgage acceptance based on applicant income.
- Simple tree model uses one feature: Applicant income.
- Focus: How the model splits the sample based on income.

Introduction to Tree-Based Models

HMDA Mortgage Data: An Example



Source: Hull (2021)

Introduction to Tree-Based Models

Decision Trees: Depth and Complexity

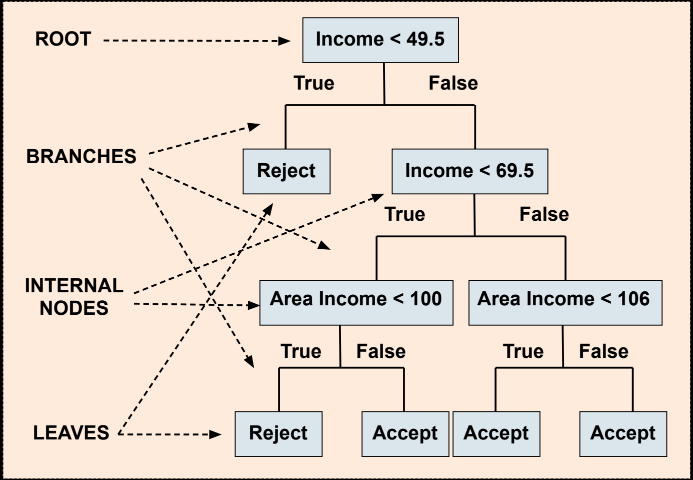
- Depth of a tree is measured by the number of branches from the root to the most distant leaf.
- A depth of one results in a “decision stump."
- Example: applicants with incomes below $25,500 are rejected.

Introduction to Tree-Based Models

Extended Decision Tree Model

- Model extended by increasing tree depth to three.
- Added feature: ratio of census tract income to metropolitan statistical income (multiplied by 100).

Introduction to Tree-Based Models



Source: Hull (2021)

Introduction to Tree-Based Models

Observations from the Model

- Starting from root, the tree first partitions by applicant income.
- Presence of “internal nodes," which are neither roots nor leaves.
- Not all pairs of leaves contain both “accept" and “reject" classes.
- Class of a leaf depends on the empirical distribution associated with the leaf.

Introduction to Tree-Based Models

Why Use Trees?

1. Advantages over other models:
   - Easier to interpret.
   - Graphically intuitive.

Introduction to Tree-Based Models

Prediction via Stratification of the Feature Space

- Divide predictor space into J distinct and non-overlapping regions, $R_1, R_2, ..., R_J$.
- Predict with the mean response value of training observations in each region.

Introduction to Tree-Based Models

How to Construct the Regions

- Regions can theoretically have any shape.
- Opt for high-dimensional rectangles for simplicity.
- Aim: Minimize the RSS, given by:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y - \hat{y})^2$$

Introduction to Tree-Based Models

Recursive Binary Splitting

- A top-down, greedy approach.
- Starts at top (all observations in a single region).
- Best split is made at each step without looking ahead.
- Split predictor space into regions: $X|X_j < s$ and $X|X_j \geq s$.

Introduction to Tree-Based Models

Splitting Process

- For any j and s, define:

$$R_1(j, s) = \{X | X_j < s\}$$

$$R_2(j, s) = \{X | X_j \geq s\}$$

- Minimize equation:

$$\sum_{i: x_i \in R_1(j,s)} (y - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y - \hat{y}_{R_2})^2$$

Introduction to Tree-Based Models

Further Splitting

- Repeat the process to minimize RSS in resulting regions.
- Instead of entire predictor space, split one of the previous regions.
- Continue until a stopping criterion is met.

Introduction to Tree-Based Models

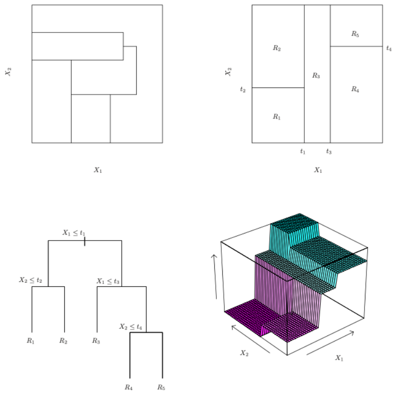


FIGURE 8.3. Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Tree Pruning

- Large trees may overfit data.
- Aim for a smaller tree to reduce variance and increase interpretability.
- One approach: grow a large tree and then prune it back.
- Use cost complexity pruning to achieve this.

Introduction to Tree-Based Models

Cost Complexity Pruning

- Also known as weakest link pruning.
- Consider a sequence of trees indexed by tuning parameter α.
- α controls a trade-off between the subtree's complexity and its fit.
- Use validation or cross-validation to select α.

Introduction to Tree-Based Models

Subtree Selection with α

Given a value of α:

$$\sum_{m=1}^{|T|} \sum_{i \in R_m} (y_i - \hat{y}_{Rm})^2 + \alpha |T| \tag{1}$$

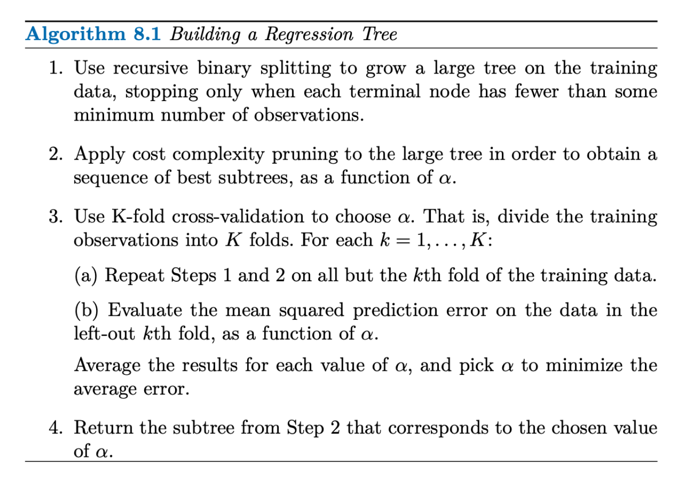
- $|T|$: number of terminal nodes in tree T.
- R_m: rectangle of the m-th terminal node.
- $\hat{y}_{Rm}$: mean of training observations in R_m.
- $\alpha = 0$ implies $T = T_0$. As α increases, smaller subtrees are preferred.

Introduction to Tree-Based Models

Analogy with the Lasso

- Akin to the lasso in linear models.
- Controls the complexity of a model.
- As α increases from 0, branches get pruned predictably.
- Obtain entire sequence of subtrees for varying α.
- Select α through validation or cross-validation.

Introduction to Tree-Based Models



Algorithm 8.1 *Building a Regression Tree*

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.

2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.

3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each $k = 1, \ldots, K$:

   (a) Repeat Steps 1 and 2 on all but the kth fold of the training data.

   (b) Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.

   Average the results for each value of α, and pick α to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of α.

Source: James et al. (2023)

Introduction to Tree-Based Models

Classification Trees

Distinction from Regression Trees:

- Predicts a qualitative response, not quantitative.
- Predicted response: most commonly occurring class in a region.
- Also, interested in class proportions for each region.

Introduction to Tree-Based Models

Growing a Classification Tree

Similarities and Differences:

- Growth similar to regression trees.
- Recursive binary splitting is used.
- RSS not applicable.

Introduction to Tree-Based Models

Classification Error Rate

- Error rate definition:

$$E = 1 - \max_{k}(\hat{p}_{mk}) \tag{2}$$

- Where $\hat{p}_{mk}$ is the proportion of training observations in the m^{th} region from the k^{th} class.
- Classification error isn't sufficiently sensitive for tree-growing.

Introduction to Tree-Based Models

Gini Index

- Defined as:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{3}$$

- Measures total variance across classes.
- Small if all $\hat{p}_{mk}$'s are near 0 or 1.
- Indicates node purity.

Introduction to Tree-Based Models

Entropy as an Alternative

- Entropy is given by:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \tag{4}$$

- Used as an alternative to the Gini Index.

Introduction to Tree-Based Models

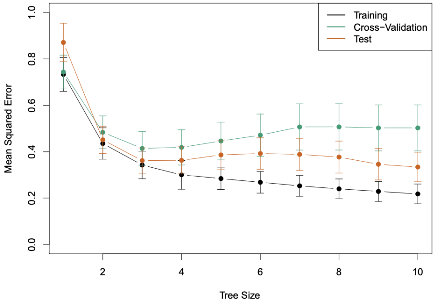


FIGURE 8.5. *Regression tree analysis for the* Hitters *data. The training, cross-validation, and test MSE are shown as a function of the number of terminal nodes in the pruned tree. Standard error bands are displayed. The minimum cross-validation error occurs at a tree size of three.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Understanding Entropy

- $0 \leq \hat{p}_{mk} \leq 1$ implies $0 \leq -\hat{p}_{mk} \log(\hat{p}_{mk})$.
- Value near zero if $\hat{p}_{mk}$'s are near 0 or 1.
- Indicates node purity, similar to Gini index.

Introduction to Tree-Based Models

Choosing Between Metrics

- Gini index and entropy evaluate split quality for classification trees.
- Preferable over classification error rate for building.
- For pruning: classification error rate prioritized for prediction accuracy.

Introduction to Tree-Based Models

The Heart Dataset Example

- Binary outcome HD for 303 patients.
- Yes/No represents presence/absence of heart disease.
- 13 predictors (e.g., Age, Sex, Chol).
- Resultant tree has six terminal nodes.

Introduction to Tree-Based Models

Qualitative Predictor Variables

- Decision trees can handle qualitative predictors.
- Splits assign certain qualitative values to branches.
- Example: Splits on “Thal" and “ChestPain."

Introduction to Tree-Based Models

Interesting Tree Characteristics

- Some splits yield terminal nodes with identical predictions.
- Example: Split “xwRestECG<1."
- The objective is to increase node purity.
- Node purity improves Gini index and entropy, enhancing certainty.

Introduction to Tree-Based Models

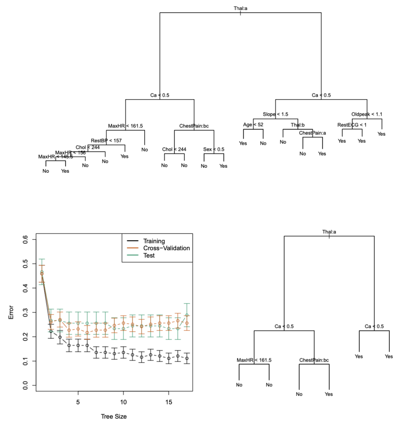


FIGURE 8.6. `Heart` *data.* Top: *The unpruned tree.* Bottom Left: *Cross-validation error, training, and test error, for different sizes of the pruned tree.* Bottom Right: *The pruned tree corresponding to the minimal cross-validation error.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Trees Versus Linear Models

- Linear regression assumes:$f(X) = X\beta$
- Regression trees assume: $f(X) = \sum_{m=1}^{M} c_m \cdot 1(X \in R_m)$
- $R_1, \ldots, R_M$ partition the feature space.

Introduction to Tree-Based Models

Choosing the Right Model

- Linear model works best if relationship is linear.
- Regression trees work for non-linear, complex relationships.
- Performance can be assessed using test error.
- Methods: cross-validation or validation set approach.

Introduction to Tree-Based Models

Other Considerations

- Beyond test error, interpretability and visualization are vital.
- Tree-based models offer clear visualization paths.
- Choice depends on problem specifics and desired outcomes.

Introduction to Tree-Based Models

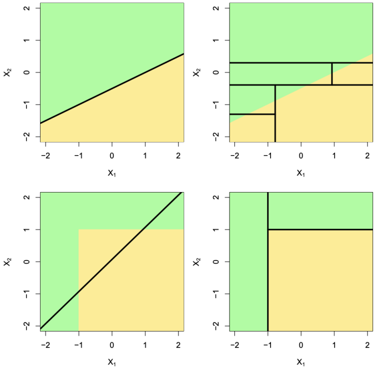


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

Source: James et al. (2023)

Introduction to Tree-Based Models

Advantages and Disadvantages of Trees

- Trees are easy to explain, often simpler than linear regression.
- Trees mirror human decision-making.
- Graphically displayed and easily interpreted.
- Handle qualitative predictors without dummy variables.

Introduction to Tree-Based Models

Challenges with Trees

- Lower predictive accuracy compared to other models.
- Non-robustness: small data changes can affect outcomes.

Introduction to Tree-Based Models

Improving Tree Predictions

- Aggregate many decision trees for better performance.
- Methods: bagging, random forests, boosting.

Introduction to Tree-Based Models

Ensemble Methods and Their Importance

- An ensemble method combines many “building block” models to create a powerful singular model.
- Building blocks often referred to as *weak learners*.
- Focus: Bagging, random forests, boosting, and Bayesian additive regression trees.
- Building blocks for these methods are regression or classification trees.

Introduction to Tree-Based Models

Bagging: Introduction and Motivation

- Bootstrap: A powerful concept, often used to compute hard-to-determine standard deviations.
- Decision trees have high variance.
- Bagging (Bootstrap Aggregation): Reduces variance of statistical methods, especially useful for decision trees.

Introduction to Tree-Based Models

Bagging: Conceptual Overview

- High variance means different results for different data splits.
- Averaging observations reduces variance.
- Goal: Build prediction models on many training sets and average predictions.
- Challenge: Generally, only one training set is available.
- Solution: Bootstrap – take repeated samples from the single training data set.

Introduction to Tree-Based Models

Bagging: Mathematical Insight

- Averaging B models:

$$f_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x) \tag{5}$$

- Using bootstrap for B training sets, the bagged estimator is:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*}_{b}(x) \tag{6}$$

Introduction to Tree-Based Models

Bagging: An Overview

- Bagging improves predictions for many regression methods, especially for decision trees.
- Procedure: Construct B regression trees using B bootstrapped training sets and average the predictions.
- Individual trees: High variance, low bias.
- Averaging B trees helps in reducing the variance.

Introduction to Tree-Based Models

Bagging in Regression vs Classification

- Described procedure suits regression (quantitative outcome).
- For classification (qualitative outcome):
  - Predict class using each of the B trees.
  - Take a majority vote to determine the overall prediction.

Introduction to Tree-Based Models

Bagging: Performance Insights

- Bagging test error rate is often lower than the error rate from a single tree.
- Number of trees, B, isn't critical; very large B doesn't lead to overfitting.
- Practical tip: Choose B large enough for error stabilization (e.g., B = 100 in the example).

Introduction to Tree-Based Models

Out-of-Bag Error Estimation

- Bagging uses bootstrapped subsets of observations. On average, each tree uses about two-thirds of the observations.
- The remaining one-third are the out-of-bag (OOB) observations.
- For each observation, around $B/3$ OOB predictions are made.
- OOB error is a valid estimate of the test error.
- OOB error is roughly equivalent to leave-one-out cross-validation error with sufficiently large B.

Introduction to Tree-Based Models

Variable Importance Measures

- Bagging improves accuracy but reduces interpretability.
- One advantage of trees is their easily interpreted diagram.
- Measure importance with RSS (regression trees) or Gini index (classification trees).
- For regression, track total RSS decrease due to splits over a predictor, averaged over all B trees.
- For classification, sum the total decrease in Gini index by splits over a predictor, averaged over all B trees.

Introduction to Tree-Based Models

Graphical Representation of Variable Importances

- The variable importances can be visually represented.
- It shows the mean decrease in Gini index for each variable.
- Variables with the most significant mean decrease in Gini index are Thal, Ca, and ChestPain.

Introduction to Tree-Based Models

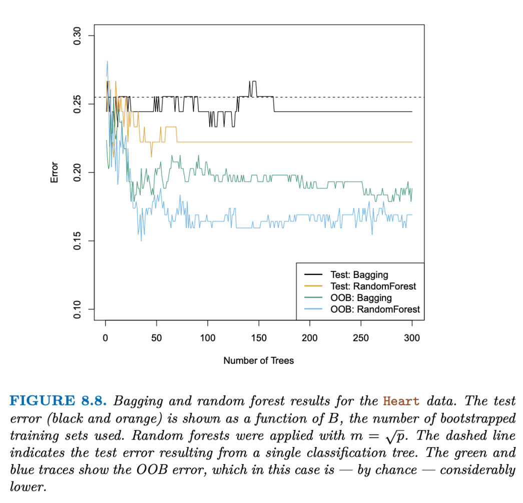


FIGURE 8.8. *Bagging and random forest results for the* Heart *data. The test error (black and orange) is shown as a function of B, the number of bootstrapped training sets used. Random forests were applied with $m = \sqrt{p}$. The dashed line indicates the test error resulting from a single classification tree. The green and blue traces show the OOB error, which in this case is — by chance — considerably lower.*

Source: James et al. (2023)

Introduction to Tree-Based Models

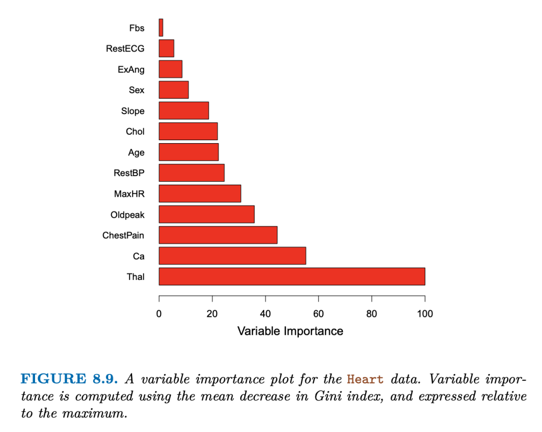


FIGURE 8.9. *A variable importance plot for the* `Heart` *data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Random Forests

- Improvement over bagged trees by decorrelating trees.
- Build decision trees on bootstrapped training samples.
- At each tree split, a random sample of m predictors is chosen out of p.
- Typically, $m \approx \sqrt{p}$.

Introduction to Tree-Based Models

Rationale Behind Random Forests

- If one strong predictor dominates, all bagged trees will look similar.
- Predictions from similar trees are highly correlated.
- Averaging many correlated predictions doesn’t reduce variance much.

Introduction to Tree-Based Models

Rationale Behind Random Forests

- Random forests limit the predictors considered at each split.
- Decorrelates the trees, making their average more reliable.

Introduction to Tree-Based Models

Differences: Bagging vs. Random Forests

- Main difference is the choice of predictor subset size m.
- For $m = p$, random forests are equivalent to bagging.
- Random forests reduce both test error and OOB error.
- Using small m values helps when predictors are correlated.

Introduction to Tree-Based Models

Application on Biological Data

- Data: Expression measurements of 4,718 genes from 349 patients.
- Goal: Predict cancer type based on 500 genes with highest variance.
- Randomly split data into training and test sets.

Introduction to Tree-Based Models

Application on Biological Data

- Applied random forests with different values of m.
- Using 400 trees is sufficient for good performance.
- Choice of $m = \sqrt{p}$ gave a small improvement over bagging.

Introduction to Tree-Based Models

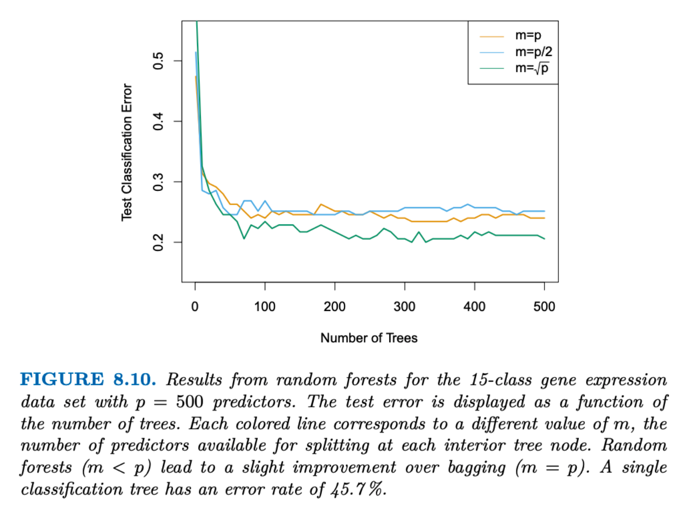


FIGURE 8.10. *Results from random forests for the 15-class gene expression data set with $p = 500$ predictors. The test error is displayed as a function of the number of trees. Each colored line corresponds to a different value of m, the number of predictors available for splitting at each interior tree node. Random forests ($m < p$) lead to a slight improvement over bagging ($m = p$). A single classification tree has an error rate of 45.7%.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Boosting Overview

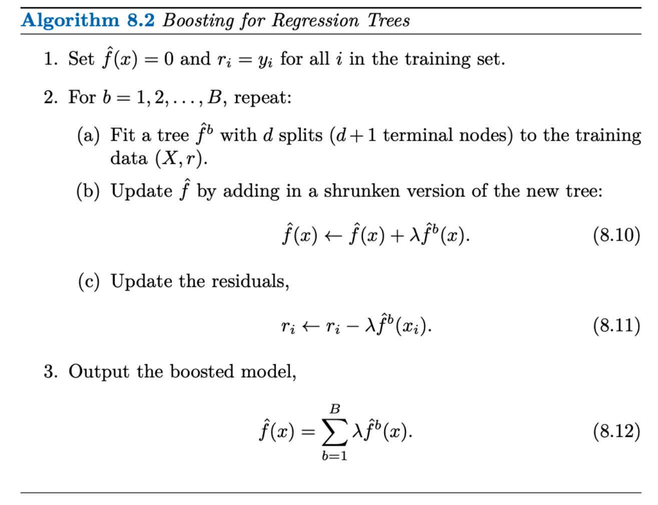
- Boosting improves predictions from decision trees.
- Can be applied to many learning methods.
- Unlike bagging, trees in boosting are grown sequentially.
- Each tree improves on previous trees.
- Does not use bootstrap sampling.

Introduction to Tree-Based Models

Boosting Mechanics

- Each tree is fit on a modified version of the original dataset.
- Boosting learns slowly: fits small trees to residuals.
- Shrinkage parameter λ slows the learning.
- Each tree depends on prior trees (unlike bagging).

Introduction to Tree-Based Models



Algorithm 8.2 *Boosting for Regression Trees*

```
1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all $i$ in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
    (a) Fit a tree $\hat{f}^b$ with $d$ splits ($d + 1$ terminal nodes) to the training data $(X, r)$.
    (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
            $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$    (8.10)
    (c) Update the residuals,
            $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$    (8.11)
3. Output the boosted model,
            $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$    (8.12)
```

Source: James et al. (2023)

Introduction to Tree-Based Models

Parameters in Boosting

- Number of trees B: Boosting can overfit with a large B.
- Shrinkage parameter λ: Controls learning rate.
- Number of splits d: Controls tree complexity.
- $d = 1$ (a stump) often works well.
- d is the interaction depth of the boosted model.

Introduction to Tree-Based Models

Boosting vs Random Forests

- Boosting considers other trees while growing a tree.
- Often, smaller trees are sufficient in boosting.
- Using stumps leads to an additive model.
- Boosted stumps can outperform more complex models.

Introduction to Tree-Based Models

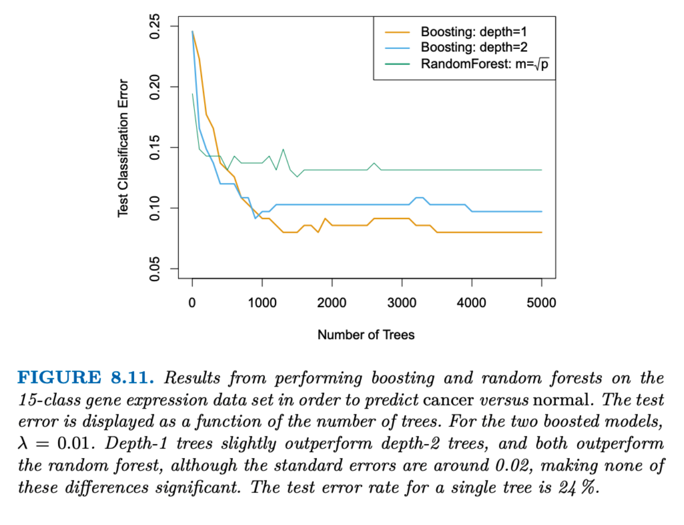
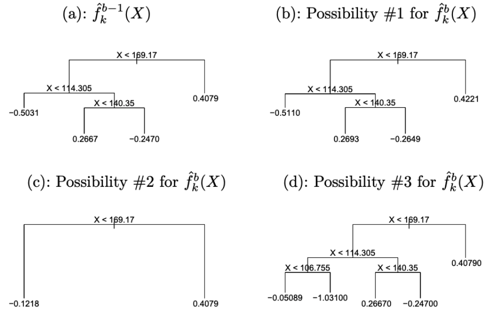


FIGURE 8.11. *Results from performing boosting and random forests on the 15-class gene expression data set in order to predict* cancer *versus* normal. *The test error is displayed as a function of the number of trees. For the two boosted models,* $\lambda = 0.01$. *Depth-1 trees slightly outperform depth-2 trees, and both outperform the random forest, although the standard errors are around 0.02, making none of these differences significant. The test error rate for a single tree is 24 %.*

Source: James et al. (2023)

Introduction to Tree-Based Models



(a): $\hat{f}_k^{b-1}(X)$

(b): Possibility #1 for $\hat{f}_k^{b}(X)$

(c): Possibility #2 for $\hat{f}_k^{b}(X)$

(d): Possibility #3 for $\hat{f}_k^{b}(X)$

FIGURE 8.12. *A schematic of perturbed trees from the BART algorithm.* (a): *The kth tree at the* $(b-1)$*st iteration,* $\hat{f}_k^{b-1}(X)$*, is displayed. Panels (b)–(d) display three of many possibilities for* $\hat{f}_k^{b}(X)$*, given the form of* $\hat{f}_k^{b-1}(X)$*.* (b): *One possibility is that* $\hat{f}_k^{b}(X)$ *has the same structure as* $\hat{f}_k^{b-1}(X)$*, but with different predictions at the terminal nodes.* (c): *Another possibility is that* $\hat{f}_k^{b}(X)$ *results from pruning* $\hat{f}_k^{b-1}(X)$*.* (d): *Alternatively,* $\hat{f}_k^{b}(X)$ *may have more terminal nodes than* $\hat{f}_k^{b-1}(X)$*.*

Source: James et al. (2023)

Introduction to Bayesian Methods

Bayesian Additive Regression Trees (BART)

- Ensemble method using decision trees.
- Related to bagging, random forests, and boosting.
- Each tree captures unaccounted signal.
- Novelty lies in tree generation.

Introduction to Tree-Based Models

Notations in BART

- K: Number of regression trees.
- B: Number of iterations in the BART algorithm.
- $f^b(x)$: Prediction at x for k^{th} tree in b^{th} iteration.
- After each iteration: $\hat{f}^b(x) = \sum_{k=1}^{K} \hat{f}^b(x)$ for $b = 1, ..., B$.

Introduction to Tree-Based Models

BART Algorithm

- Initially: $\hat{f}^1(x) = \frac{1}{K} \sum_{i=1}^{n} y_i$.
- Each tree is updated in subsequent iterations.
- Partial residuals help decide perturbations.
- Perturbations can change tree structure or node predictions.

Introduction to Tree-Based Models

Perturbations in BART

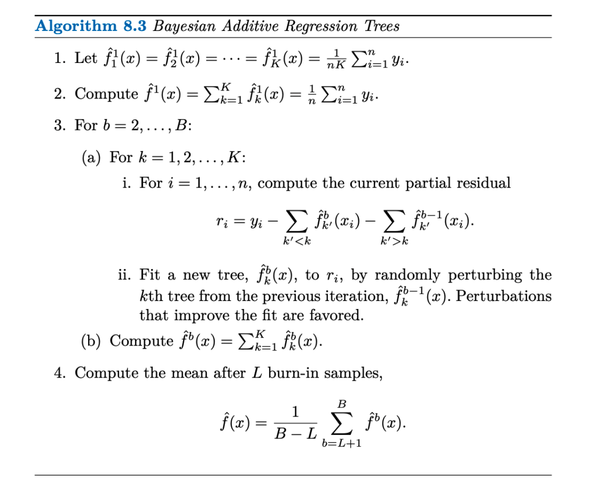
- Perturbations are chosen based on fit to the partial residual.
- Tree structure can be altered by adding/pruning branches.
- Terminal node predictions can be modified.

Introduction to Tree-Based Models

BART Output

- Output: Collection of prediction models $\hat{f}(x)$.
- For every observation: $\hat{f}(x) = \hat{f}b(x),$ for $b = 1, 2, ..., B$.

Introduction to Tree-Based Models



Algorithm 8.3 *Bayesian Additive Regression Trees*

```
1. Let $\hat{f}_1^1(x) = \hat{f}_2^1(x) = \cdots = \hat{f}_K^1(x) = \frac{1}{nK}\sum_{i=1}^n y_i$.
2. Compute $\hat{f}^1(x) = \sum_{k=1}^K \hat{f}_k^1(x) = \frac{1}{n}\sum_{i=1}^n y_i$.
3. For $b = 2, \ldots, B$:
   (a) For $k = 1, 2, \ldots, K$:
       i. For $i = 1, \ldots, n$, compute the current partial residual
            $r_i = y_i - \sum_{k'<k} \hat{f}_{k'}^b(x_i) - \sum_{k'>k} \hat{f}_{k'}^{b-1}(x_i).$
       ii. Fit a new tree, $\hat{f}_k^b(x)$, to $r_i$, by randomly perturbing the
           $k$th tree from the previous iteration, $\hat{f}_k^{b-1}(x)$. Perturbations
           that improve the fit are favored.
   (b) Compute $\hat{f}^b(x) = \sum_{k=1}^K \hat{f}_k^b(x)$.
4. Compute the mean after $L$ burn-in samples,
       $\hat{f}(x) = \frac{1}{B-L}\sum_{b=L+1}^{B} \hat{f}^b(x).$
```

Source: James et al. (2023)

Introduction to Tree-Based Models

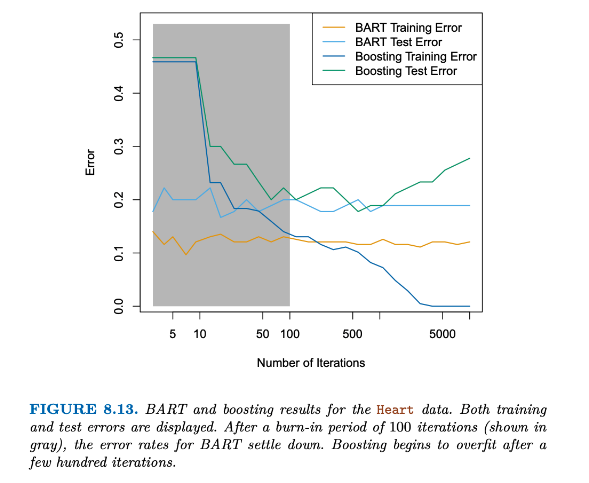


FIGURE 8.13. *BART and boosting results for the* `Heart` *data. Both training and test errors are displayed. After a burn-in period of* 100 *iterations (shown in gray), the error rates for BART settle down. Boosting begins to overfit after a few hundred iterations.*

Source: James et al. (2023)

Introduction to Tree-Based Models

Burn-in Period in BART

- The first few models, called burn-in period, are discarded.
- Denoted by L, e.g., $L = 200$.
- Averages taken after burn-in: $f(x) = \frac{1}{B-L} \sum_{b=L+1}^{B} f^b(x)$.
- Beyond averages, percentiles can show prediction uncertainty.

Introduction to Tree-Based Models

BART Tree Modifications

- Trees aren’t entirely refitted but slightly modified.
- Limits overfitting.
- Individual trees are generally small to avoid overfitting.

Introduction to Tree-Based Models

BART Application on Heart Data

- Used $K = 200$ trees and increased iterations to 10,000.
- Initial iterations showed varying errors.
- Small difference between training and test errors.
- Indicates effective prevention of overfitting.

Introduction to Tree-Based Models

Comparison with Boosting

- Boosting test error approaches BART, but then rises.
- Boosting overfits with increased iterations.
- BART seen as a Bayesian approach to ensemble trees.
- Applies a Markov chain Monte Carlo method.

Introduction to Tree-Based Models

Parameter Choices in BART

- Must select: K (trees), B (iterations), L (burn-in iterations).
- Typical choices: $K = 200$, $B = 1,000$, $L = 100$.
- BART is known for impressive out-of-box performance.

Introduction to Tree-Based Models

Importance of Trees in Ensemble Methods

- Flexible and can handle various predictor types.
- Suitable for both qualitative and quantitative predictors.

Introduction to Tree-Based Models

Overview of Four Tree Ensemble Approaches

- Bagging
- Random Forests
- Boosting
- BART

Introduction to Tree-Based Models

Bagging

- Trees grown independently on random observation samples.
- Trees tend to be very similar.
- Risks getting caught in local optima.

Introduction to Tree-Based Models

Random Forests

- Like bagging, trees grown independently on random samples.
- Uses random subset of features for each split.
- Trees are decorrelated for better model space exploration.

Introduction to Tree-Based Models

Boosting

- Utilizes original data without random samples.
- Trees grown successively and use a "slow" learning approach.
- Each tree fits the residual signal and is then shrunk.

Introduction to Tree-Based Models

BART

- Uses original data and trees are grown successively.
- Each tree is perturbed for a more thorough model exploration.
- Avoids local minima challenges.

2. Applications

Introduction to TensorFlow

Colab Tutorial

- Tree-Based Models in Sklearn
- Tree-Based Models in TensorFlow
- Official CatBoost Tutorial

References I

Hull, Isaiah (2021) *Machine Learning for Economics and Finance in TensorFlow 2*: Apress, 10.1007/978-1-4842-6373-0.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor (2023) *An Introduction to Statistical Learning: with Applications in Python*, Springer Texts in Statistics: Springer Cham, 1st edition, XV, 60, https://doi.org/10.1007/978-3-031-38747-0, eBook ISBN: 978-3-031-38747-0; Softcover ISBN: 978-3-031-39189-7; Series ISSN: 1431-875X; Series E-ISSN: 2197-4136.