Lecture 19

Decision trees and random forests

COURSE MATH 637: Mathematical Techniques in Data Science

WEBSITE https://jacobcalvert.com/teaching/math637fall2026/

INSTRUCTOR Jacob Calvert (calvert@udel.edu)

OFFICE HOURS MW 2:45pm–3:45pm, Ewing Hall 509

Plan for today

• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

Recall: k-NN averages nearby labels

For labels $y_i \in \{0, 1\}$, k-NN estimates $\mathbb{P}(Y = 1 \mid X = x)$ by

$$\hat{\eta}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i,$$

where $N_k(x) =$ indices of the k training points closest to x.

- $\hat{\eta}(x)$ is the fraction of 1s near x
- $N_k(x)$ is chosen without the labels y_i

Recall: k-NN averages nearby labels

For labels $y_i \in \{0, 1\}$, k-NN estimates $\mathbb{P}(Y = 1 \mid X = x)$ by

$$\hat{\eta}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i,$$

where $N_k(x) =$ indices of the k training points closest to x.

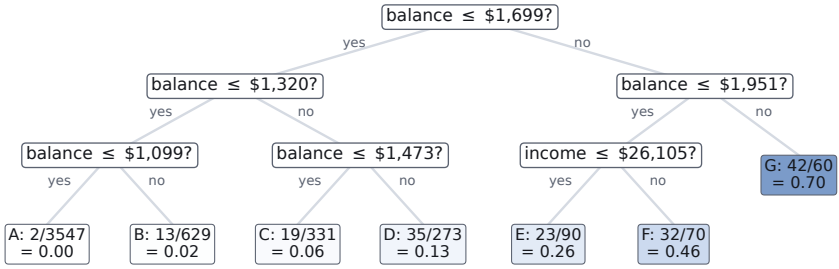
- $\hat{\eta}(x)$ is the fraction of 1s near x
- $N_k(x)$ is chosen without the labels y_i

Question

Can the labels help choose the neighborhoods?

Example: A decision tree for credit default

Recall: credit-default data with $5{,}000$ training accounts.

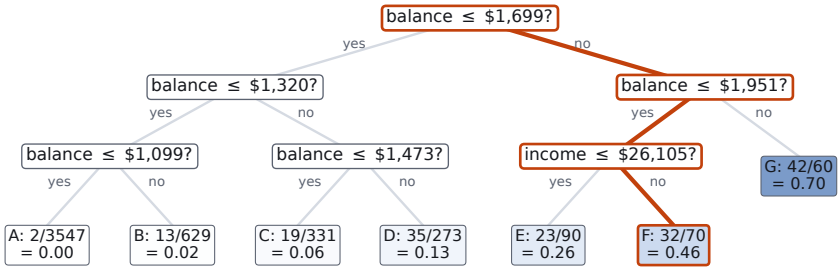


Check

Balance $= \$1{,}850$ and income $= \$40{,}000$ corresponds to which leaf? What is the estimated probability of default?

Example: A decision tree for credit default

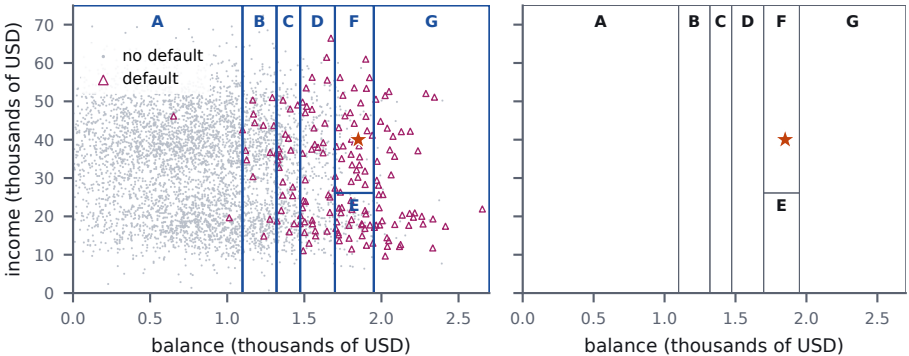
Recall: credit-default data with $5{,}000$ training accounts.



Check

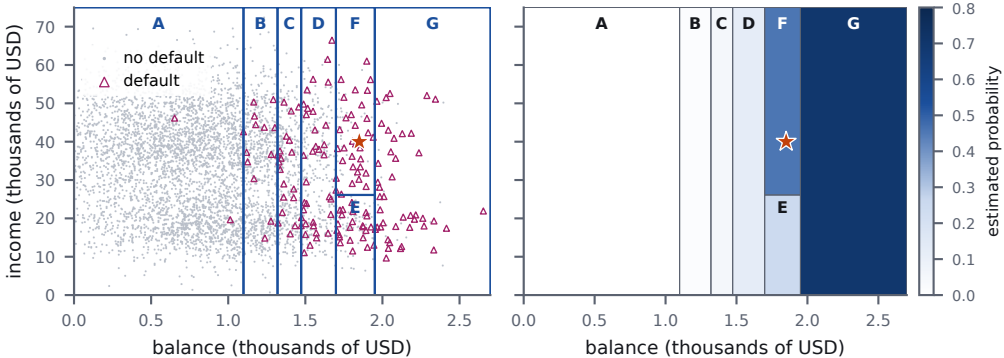
Leaf F: 32 of its 70 training accounts defaulted, so $\hat{\eta} = 32/70 = 0.457$.

Example: Same decision tree, viewed as a *partition*



- leaf = rectangle: every x in rectangle F gets $\hat{\eta}(x) = 0.457$

Example: Same decision tree, viewed as a *partition*



- leaf $=$ rectangle: every x in rectangle F gets $\hat{\eta}(x) = 0.457$
- each boundary $=$ one "split" in the decision tree

Definition: Decision tree

Definition (Decision tree)

Splitting the feature space repeatedly, one feature and one threshold at a time, gives disjoint rectangles $R_1, \dots, R_M$ (the leaves). The tree predicts

$$f(x) = \sum_{m=1}^{M} c_m \, \mathbf{1}\{x \in R_m\}, \qquad c_m = \text{mean of the } y_i \text{ with } x_i \in R_m.$$

- binary y_i: c_m = fraction of 1s in R_m, an estimate of $\mathbb{P}(Y = 1 \mid X \in R_m)$

Definition: Decision tree

Definition (Decision tree)

Splitting the feature space repeatedly, one feature and one threshold at a time, gives disjoint rectangles $R_1, \dots, R_M$ (the leaves). The tree predicts

$$f(x) = \sum_{m=1}^{M} c_m \, \mathbf{1}\{x \in R_m\}, \qquad c_m = \text{mean of the } y_i \text{ with } x_i \in R_m.$$

- binary y_i: c_m = fraction of 1s in R_m, an estimate of $\mathbb{P}(Y = 1 \mid X \in R_m)$
- K-nary y_i: each leaf stores class fractions $(\hat{\eta}_1, \dots, \hat{\eta}_K)$

Definition: Decision tree

Definition (Decision tree)

Splitting the feature space repeatedly, one feature and one threshold at a time, gives disjoint rectangles $R_1, \dots, R_M$ (the leaves). The tree predicts

$$f(x) = \sum_{m=1}^{M} c_m \, \mathbf{1}\{x \in R_m\}, \qquad c_m = \text{mean of the } y_i \text{ with } x_i \in R_m.$$

- binary y_i: c_m = fraction of 1s in R_m, an estimate of $\mathbb{P}(Y = 1 \mid X \in R_m)$
- K-nary y_i: each leaf stores class fractions $(\hat{\eta}_1, \dots, \hat{\eta}_K)$
- regression: c_m = average response in the leaf

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

- each x_i lies in one R_m, so sum is $\sum_m \sum_{i:\, x_i \in R_m} (y_i - f(x_i))^2$

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

- each x_i lies in one R_m, so sum is $\sum_m \sum_{i:\, x_i \in R_m} (y_i - f(x_i))^2$
- but $f(x_i) = c_m$ in R_m, giving $\sum_m \sum_{i:\, x_i \in R_m} (y_i - c_m)^2$

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

- each x_i lies in one R_m, so sum is $\sum_m \sum_{i:\, x_i \in R_m} (y_i - f(x_i))^2$
- but $f(x_i) = c_m$ in R_m, giving $\sum_m \sum_{i:\, x_i \in R_m} (y_i - c_m)^2$
- inner sum is minimized when $c_m =$ mean response in R_m

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

- each x_i lies in one R_m, so sum is $\sum_m \sum_{i:\, x_i \in R_m} (y_i - f(x_i))^2$
- but $f(x_i) = c_m$ in R_m, giving $\sum_m \sum_{i:\, x_i \in R_m} (y_i - c_m)^2$
- inner sum is minimized when $c_m =$ mean response in R_m

Surprise! Leaf values c_m are least squares

Problem

Fix the rectangles. Show that $\sum_i (y_i - f(x_i))^2$ is smallest when each c_m is the mean of the y_i in R_m.

- each x_i lies in one R_m, so sum is $\sum_m \sum_{i:\, x_i \in R_m} (y_i - f(x_i))^2$
- but $f(x_i) = c_m$ in R_m, giving $\sum_m \sum_{i:\, x_i \in R_m} (y_i - c_m)^2$
- inner sum is minimized when $c_m =$ mean response in R_m

Note: Equivalently, define $Z_{im} = \mathbf{1}\{x_i \in R_m\}$ and solve $Z^\mathsf{T} Z\, c = Z^\mathsf{T} y$ to find $n_m c_m = \sum_{x_i \in R_m} y_i$ for every row m.

Plan for today

• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

How should we choose the rectangles?

Problem

Among trees with at most M leaves, find one with smallest residual sum of squares

$$\mathrm{RSS} = \sum_i (y_i - f(x_i))^2.$$

- if no limit M on leaves: just give each training point its own! $\mathrm{RSS} = 0$

How should we choose the rectangles?

Problem

Among trees with at most M leaves, find one with smallest residual sum of squares

$$\mathrm{RSS} = \sum_i (y_i - f(x_i))^2.$$

- if no limit M on leaves: just give each training point its own! $\mathrm{RSS} = 0$
- with limit: all splits must be chosen jointly, a combinatorial search that is generally impractical

How should we choose the rectangles?

Problem

Among trees with at most M leaves, find one with smallest residual sum of squares

$$\mathrm{RSS} = \sum_i (y_i - f(x_i))^2.$$

- if no limit M on leaves: just give each training point its own! $\mathrm{RSS} = 0$
- with limit: all splits must be chosen jointly, a combinatorial search that is generally impractical
- in practice: greedily choose one split at a time

Definition: Greedy recursive splitting

Definition (Greedy recursive splitting)

At a node "holding" example indices I:

1. for each feature j and each threshold s between consecutive values of x_{ij}, let $I_L = \{i \in I : x_{ij} \leq s\}$ and $I_R = I \setminus I_L$,

$$\mathrm{RSS}(j, s) = \sum_{i \in I_L} (y_i - \bar{y}_L)^2 + \sum_{i \in I_R} (y_i - \bar{y}_R)^2,$$

   where $\bar{y}_L$, $\bar{y}_R$ are the mean responses of the children nodes;
2. split on the (j, s) with the smallest $\mathrm{RSS}(j, s)$;
3. repeat with I_L, and separately with I_R, until a stopping rule is met.

Example: One split by hand

Problem

Find the best split of $x = 1, 2, 3, 4, 5, 6$ with $y = 1, 2, 3, 7, 8, 9$.

split	left mean	right mean	RSS
$x \leq 1.5$	1	5.8	
$x \leq 2.5$	1.5	6.75	
$x \leq 3.5$	2	8	
$x \leq 4.5$	3.25	8.5	
$x \leq 5.5$	4.2	9	

Example: One split by hand

Problem

Find the best split of $x = 1, 2, 3, 4, 5, 6$ with $y = 1, 2, 3, 7, 8, 9$.

split	left mean	right mean	RSS
$x \leq 1.5$	1	5.8	$0 + 38.8 = 38.8$
$x \leq 2.5$	1.5	6.75	$0.5 + 20.75 = 21.25$
$x \leq 3.5$	2	8	$2 + 2 = 4$
$x \leq 4.5$	3.25	8.5	$20.75 + 0.5 = 21.25$
$x \leq 5.5$	4.2	9	$38.8 + 0 = 38.8$

Example: One split by hand

Problem

Find the best split of $x = 1, 2, 3, 4, 5, 6$ with $y = 1, 2, 3, 7, 8, 9$.

split	left mean	right mean	RSS
$x \leq 1.5$	1	5.8	$0 + 38.8 = 38.8$
$x \leq 2.5$	1.5	6.75	$0.5 + 20.75 = 21.25$
$x \leq 3.5$	2	8	$2 + 2 = 4$
$x \leq 4.5$	3.25	8.5	$20.75 + 0.5 = 21.25$
$x \leq 5.5$	4.2	9	$38.8 + 0 = 38.8$

Note: Before splitting, $\mathrm{RSS} = 58$. Next, each child $\{1, 2, 3\}$ and $\{4, 5, 6\}$ is split the same way, on its own points.

Various measures of split quality in GRS

Idea: Give each node an "impurity" Q, which is small when the node's labels/responses y_i are alike and large otherwise.

A split of n points into n_L and n_R is scored by $n_L Q(\text{left}) + n_R Q(\text{right})$.

	impurity Q of a node	**with class-1 fraction $\bar{y}$**
regression	mean squared deviation	
classification	misclassification rate	$\min(\bar{y}, 1-\bar{y})$
	Gini impurity	$2\bar{y}(1-\bar{y})$
	entropy	$-\bar{y}\log\bar{y} - (1-\bar{y})\log(1-\bar{y})$

- for regression, the score is the RSS from before

Various measures of split quality in GRS

Idea: Give each node an "impurity" Q, which is small when the node's labels/responses y_i are alike and large otherwise.

A split of n points into n_L and n_R is scored by $n_L Q(\text{left}) + n_R Q(\text{right})$.

	impurity Q of a node	**with class-1 fraction $\bar{y}$**
regression	mean squared deviation	
classification	misclassification rate	$\min(\bar{y}, 1-\bar{y})$
	Gini impurity	$2\bar{y}(1-\bar{y})$
	entropy	$-\bar{y}\log\bar{y} - (1-\bar{y})\log(1-\bar{y})$

- for regression, the score is the RSS from before
- for classification there are several choices

First: misclassification rate is a poor criterion

Example: Default data, split on balance. Each node predicts majority label:

node	accounts	default rate	majority label	mistakes
before the split	$5{,}000$	3.3%	no default	166
balance $\leq \$1{,}699$	$4{,}780$	1.4%	no default	69
balance $> \$1{,}699$	220	44.1%	no default	97

- very different groups, yet $69 + 97 = 166$ mistakes, as before; the same for 97% of the thresholds on balance (why?)

First: misclassification rate is a poor criterion

Example: Default data, split on balance. Each node predicts majority label:

node	accounts	default rate	majority label	mistakes
before the split	5,000	3.3%	no default	166
balance $\leq$ \$1,699	4,780	1.4%	no default	69
balance $>$ \$1,699	220	44.1%	no default	97

- very different groups, yet $69 + 97 = 166$ mistakes, as before; the same for 97% of the thresholds on balance (why?)

Watch out: If both children keep parent's majority label, the number of mistakes cannot change. A criterion should respond to the fractions themselves.

Second: Gini impurity is RSS for labels

Check

For $y_i \in \{0, 1\}$ in a node with n points and class-1 fraction $\bar{y}$, show that $\text{RSS} = \sum_i (y_i - \bar{y})^2 = n\bar{y}(1 - \bar{y})$.

- Note that $y_i^2 = y_i$, so

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i^2 - 2y_i\bar{y} + \bar{y}^2) = n\bar{y} - n\bar{y}^2 = n\bar{y}(1 - \bar{y})$$

Second: Gini impurity is RSS for labels

Check

For $y_i \in \{0, 1\}$ in a node with n points and class-1 fraction $\bar{y}$, show that $\mathrm{RSS} = \sum_i (y_i - \bar{y})^2 = n\bar{y}(1 - \bar{y})$.

- Note that $y_i^2 = y_i$, so

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i^2 - 2y_i\bar{y} + \bar{y}^2) = n\bar{y} - n\bar{y}^2 = n\bar{y}(1 - \bar{y})$$

- Since $G = 2\bar{y}(1 - \bar{y})$, impurity of a new split is $n_L G_L + n_R G_R = 2\,\mathrm{RSS}$

Second: Gini impurity is RSS for labels

Check

For $y_i \in \{0, 1\}$ in a node with n points and class-1 fraction $\bar{y}$, show that $\mathrm{RSS} = \sum_i (y_i - \bar{y})^2 = n\bar{y}(1 - \bar{y})$.

- Note that $y_i^2 = y_i$, so

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i^2 - 2y_i\bar{y} + \bar{y}^2) = n\bar{y} - n\bar{y}^2 = n\bar{y}(1 - \bar{y})$$

- Since $G = 2\bar{y}(1 - \bar{y})$, impurity of a new split is $n_L G_L + n_R G_R = 2\,\mathrm{RSS}$
- K classes: $G = 1 - \sum_k \hat{\eta}_k^2$

Second: Gini impurity is RSS for labels

Check

For $y_i \in \{0, 1\}$ in a node with n points and class-1 fraction $\bar{y}$, show that $\mathrm{RSS} = \sum_i (y_i - \bar{y})^2 = n\bar{y}(1 - \bar{y})$.

- Note that $y_i^2 = y_i$, so

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i^2 - 2y_i\bar{y} + \bar{y}^2) = n\bar{y} - n\bar{y}^2 = n\bar{y}(1 - \bar{y})$$

- Since $G = 2\bar{y}(1 - \bar{y})$, impurity of a new split is $n_L G_L + n_R G_R = 2\,\mathrm{RSS}$
- K classes: $G = 1 - \sum_k \hat{\eta}_k^2$

Second: Gini impurity is RSS for labels

Check

For $y_i \in \{0, 1\}$ in a node with n points and class-1 fraction $\bar{y}$, show that $\mathrm{RSS} = \sum_i (y_i - \bar{y})^2 = n\bar{y}(1 - \bar{y})$.

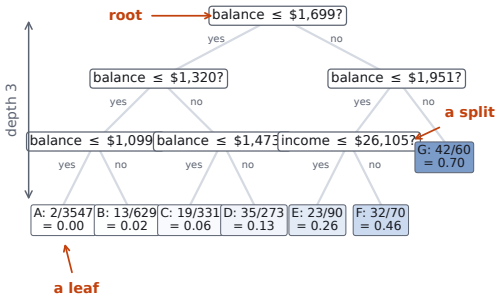
- Note that $y_i^2 = y_i$, so

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i^2 - 2y_i\bar{y} + \bar{y}^2) = n\bar{y} - n\bar{y}^2 = n\bar{y}(1 - \bar{y})$$

- Since $G = 2\bar{y}(1 - \bar{y})$, impurity of a new split is $n_L G_L + n_R G_R = 2\,\mathrm{RSS}$
- K classes: $G = 1 - \sum_k \hat{\eta}_k^2$

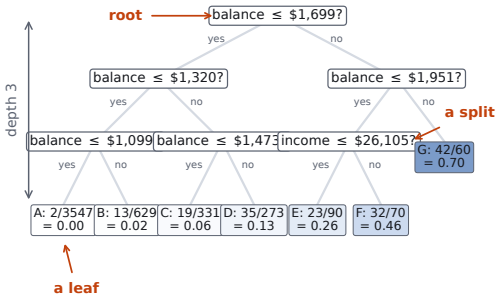
Key idea: A tree grown with Gini impurity *is* the regression tree fitted to the $0/1$ labels.

Aside: Terminology



- node: a set of training points
- split: a feature j and a threshold s; points with $x_j \leq s$ go left
- leaf: a node that is not split; it stores c_m
- depth: number of splits from the root to the deepest leaf
- leaf size: number of training points in a leaf

Aside: Terminology



- node: a set of training points
- split: a feature j and a threshold s; points with $x_j \leq s$ go left
- leaf: a node that is not split; it stores c_m
- depth: number of splits from the root to the deepest leaf
- leaf size: number of training points in a leaf

Note: A split compares one feature with a threshold, so changing units does not change the tree. No standardization needed, unlike k-NN!

Python: A classification tree

```
from sklearn.tree import (DecisionTreeClassifier,
                          export_text)
X2 = X_train[["balance", "income"]]
tree = DecisionTreeClassifier(max_depth=3,
           min_samples_leaf=50, random_state=0)
tree.fit(X2, y_train)          # criterion="gini"
print(export_text(tree, feature_names=list(X2)))
X2_test = X_test[["balance", "income"]]
p_hat = tree.predict_proba(X2_test)[:, 1]  # fractions
```

- `predict_proba` returns the class fractions of each account's leaf
- `max_depth` and `min_samples_leaf` are stopping rules

Plan for today

• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

Greedy splitting needs a stopping rule

Greedy splitting says how to split a node, not when to stop splitting!

Common stopping rules:

- a maximum depth

Greedy splitting needs a stopping rule

Greedy splitting says how to split a node, not when to stop splitting!

Common stopping rules:

- a maximum depth
- a minimum leaf size: no split may leave $< L$ points in a child node

Greedy splitting needs a stopping rule

Greedy splitting says how to split a node, not when to stop splitting!

Common stopping rules:

- a maximum depth
- a minimum leaf size: no split may leave $< L$ points in a child node
- a minimum improvement in the score

Greedy splitting needs a stopping rule

Greedy splitting says how to split a node, not when to stop splitting!

Common stopping rules:

- a maximum depth
- a minimum leaf size: no split may leave $< L$ points in a child node
- a minimum improvement in the score

Greedy splitting needs a stopping rule

Greedy splitting says how to split a node, not when to stop splitting!

Common stopping rules:

- a maximum depth
- a minimum leaf size: no split may leave $< L$ points in a child node
- a minimum improvement in the score

Question

What happens with no stopping rule at all?

No stopping: A tree grown until every leaf is pure

Example: Credit data again. Use balance, income, and student indicator. With no stopping rule: 190 leaves, every training account correct.

On the $5{,}000$ test accounts:

- 109 defaulters get $\hat{\eta} = 0$, 105 non-defaulters get $\hat{\eta} = 1$

No stopping: A tree grown until every leaf is pure

Example: Credit data again. Use balance, income, and student indicator. With no stopping rule: 190 leaves, every training account correct.

On the $5{,}000$ test accounts:

- 109 defaulters get $\hat{\eta} = 0$, 105 non-defaulters get $\hat{\eta} = 1$
- log loss $-\frac{1}{n}\sum_i [y_i \log \hat{\eta}(x_i) + (1 - y_i) \log(1 - \hat{\eta}(x_i))]$ from L17 is ∞

No stopping: A tree grown until every leaf is pure

Example: Credit data again. Use balance, income, and student indicator. With no stopping rule: 190 leaves, every training account correct.

On the $5{,}000$ test accounts:

- 109 defaulters get $\hat{\eta} = 0$, 105 non-defaulters get $\hat{\eta} = 1$
- log loss $-\frac{1}{n}\sum_i [y_i \log \hat{\eta}(x_i) + (1 - y_i) \log(1 - \hat{\eta}(x_i))]$ from L17 is ∞
- the Brier score $\frac{1}{n}\sum_i (\hat{\eta}(x_i) - y_i)^2$ (squared error of probabilities) $= 0.043$, but trivially predicting $\hat{\eta}(x) = 0.033$ for everyone gives 0.032

No stopping: A tree grown until every leaf is pure

Example: Credit data again. Use balance, income, and student indicator. With no stopping rule: 190 leaves, every training account correct.

On the $5{,}000$ test accounts:

- 109 defaulters get $\hat{\eta} = 0$, 105 non-defaulters get $\hat{\eta} = 1$
- log loss $-\frac{1}{n}\sum_i [y_i \log \hat{\eta}(x_i) + (1 - y_i) \log(1 - \hat{\eta}(x_i))]$ from L17 is ∞
- the Brier score $\frac{1}{n}\sum_i (\hat{\eta}(x_i) - y_i)^2$ (squared error of probabilities) $= 0.043$, but trivially predicting $\hat{\eta}(x) = 0.033$ for everyone gives 0.032

No stopping: A tree grown until every leaf is pure

Example: Credit data again. Use balance, income, and student indicator. With no stopping rule: 190 leaves, every training account correct.

On the $5{,}000$ test accounts:

- 109 defaulters get $\hat{\eta} = 0$, 105 non-defaulters get $\hat{\eta} = 1$
- log loss $-\frac{1}{n}\sum_i [y_i \log \hat{\eta}(x_i) + (1 - y_i) \log(1 - \hat{\eta}(x_i))]$ from L17 is ∞
- the Brier score $\frac{1}{n}\sum_i (\hat{\eta}(x_i) - y_i)^2$ (squared error of probabilities) $= 0.043$, but trivially predicting $\hat{\eta}(x) = 0.033$ for everyone gives 0.032

Key idea: At fixed x, with $\eta = \mathbb{P}(Y = 1 \mid X = x)$:

$$\mathbb{E}\big[(\hat{\eta}(x) - Y)^2\big] = \eta(1 - \eta) + \big(\mathbb{E}\hat{\eta}(x) - \eta\big)^2 + \operatorname{Var} \hat{\eta}(x).$$

In example above, small bias, very large variance.

Instead: Choose min leaf size on a validation set

Example: Fit on $4{,}000$ training accounts; compute the squared error on the other $1{,}000$.

smallest leaf allowed	1	20	100	200	400
leaves	153	39	14	9	7
validation error	0.0430	0.0274	0.0252	0.0239	0.0248

Refit with leaves of at least 200 on all $5{,}000$, test once.

Gives Brier score of 0.023 (trivial prediction had 0.032) and an AUC of 0.935

Plan for today

• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$. Grow a tree on each half with no stopping rule. Both halves come from the same population, so *ideally* the two trees would give similar estimates.

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts
- for 242 test accounts, one tree says 0 and the other says 1

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts
- for 242 test accounts, one tree says 0 and the other says 1
- among the 456 test accounts with balance above $\$1{,}500$: 167, or 36.6%

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts
- for 242 test accounts, one tree says 0 and the other says 1
- among the 456 test accounts with balance above $\$1{,}500$: 167, or 36.6%
- the estimates therefore depend heavily on which accounts happen to be in A and B

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts
- for 242 test accounts, one tree says 0 and the other says 1
- among the 456 test accounts with balance above $\$1{,}500$: 167, or 36.6%
- the estimates therefore depend heavily on which accounts happen to be in A and B

Key idea: Deep trees are unstable

Example: Split $5{,}000$ training accounts into two sets of $2{,}500$, called A and B. Grow a tree on each with no stopping rule. Both come from the same population, so *ideally* the A and B trees would give similar estimates.

- compare estimates $\hat{\eta}^{(A)}(x)$ and $\hat{\eta}^{(B)}(x)$ on the test accounts
- for 242 test accounts, one tree says 0 and the other says 1
- among the 456 test accounts with balance above $\$1{,}500$: 167, or 36.6%
- the estimates therefore depend heavily on which accounts happen to be in A and B

Idea: Suppress variance by average many such trees, each grown on different subsets of the training data.

Aside: The bootstrap

Definition (Bootstrap sample (Efron, 1979))

Draw n points with replacement from the n training points.

- training points $\{1, 2, 3, 4, 5\}$; bootstrap samples, e.g., $\{2, 2, 3, 5, 5\}$ or $\{1, 1, 1, 3, 4\}$

Aside: The bootstrap

Definition (Bootstrap sample (Efron, 1979))

Draw n points with replacement from the n training points.

- training points $\{1, 2, 3, 4, 5\}$; bootstrap samples, e.g., $\{2, 2, 3, 5, 5\}$ or $\{1, 1, 1, 3, 4\}$
- some points appear more than once, others not at all

Aside: The bootstrap

Definition (Bootstrap sample (Efron, 1979))

Draw n points with replacement from the n training points.

- training points $\{1, 2, 3, 4, 5\}$; bootstrap samples, e.g., $\{2, 2, 3, 5, 5\}$ or $\{1, 1, 1, 3, 4\}$
- some points appear more than once, others not at all
- the training points stand in for the population: drawing from them imitates drawing a new training set

Aside: The bootstrap

Definition (Bootstrap sample (Efron, 1979))

Draw n points with replacement from the n training points.

- training points $\{1, 2, 3, 4, 5\}$; bootstrap samples, e.g., $\{2, 2, 3, 5, 5\}$ or $\{1, 1, 1, 3, 4\}$
- some points appear more than once, others not at all
- the training points stand in for the population: drawing from them imitates drawing a new training set
- can recompute any statistic/fitted model on each bootstrap sample

Aside: The bootstrap

Definition (Bootstrap sample (Efron, 1979))

Draw n points with replacement from the n training points.

- training points $\{1, 2, 3, 4, 5\}$; bootstrap samples, e.g., $\{2, 2, 3, 5, 5\}$ or $\{1, 1, 1, 3, 4\}$
- some points appear more than once, others not at all
- the training points stand in for the population: drawing from them imitates drawing a new training set
- can recompute any statistic/fitted model on each bootstrap sample
- the spread across samples shows how much it depends on data

Definition: Bagging

Definition (Bootstrap aggregation (Breiman, 1996))

For $b = 1, \ldots, B$: draw a bootstrap sample and grow a tree $\hat{\eta}^{(b)}$ on it. Estimate

$$\hat{\eta}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{\eta}^{(b)}(x).$$

- we average the trees' estimated probabilities, as scikit-learn does (another version takes a majority vote of their labels)

Definition: Bagging

Definition (Bootstrap aggregation (Breiman, 1996))

For $b = 1, \dots, B$: draw a bootstrap sample and grow a tree $\hat{\eta}^{(b)}$ on it. Estimate

$$\hat{\eta}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{\eta}^{(b)}(x).$$

- we average the trees' estimated probabilities, as scikit-learn does (another version takes a majority vote of their labels)
- the trees are grown without a stopping rule: each has small bias and large variance

Averaging lowers the variance

Proposition

Let $T_1, \dots, T_B$ be identically distributed, with mean μ, variance σ^2, and correlation ρ between any two. Then

$$\mathbb{E}\left[\frac{1}{B}\sum_b T_b\right] = \mu, \qquad \mathrm{Var}\left(\frac{1}{B}\sum_b T_b\right) = \frac{1-\rho}{B}\,\sigma^2 + \rho\,\sigma^2.$$

- $T_b = \hat{\eta}^{(b)}(x)$ at a fixed x; random through the training data and the bootstrap draw

Averaging lowers the variance

Proposition

Let $T_1, \ldots, T_B$ be identically distributed, with mean μ, variance σ^2, and correlation ρ between any two. Then

$$\mathbb{E}\left[\frac{1}{B}\sum_b T_b\right] = \mu, \qquad \mathrm{Var}\left(\frac{1}{B}\sum_b T_b\right) = \frac{1-\rho}{B}\,\sigma^2 + \rho\,\sigma^2.$$

- $T_b = \hat{\eta}^{(b)}(x)$ at a fixed x; random through the training data and the bootstrap draw
- $\mathrm{Var}(\sum_b T_b) = B\sigma^2 + B(B-1)\rho\sigma^2$ (Check this.)

Averaging lowers the variance

Proposition

Let $T_1, \dots, T_B$ be identically distributed, with mean μ, variance σ^2, and correlation ρ between any two. Then

$$\mathbb{E}\left[\frac{1}{B}\sum_b T_b\right] = \mu, \qquad \mathrm{Var}\left(\frac{1}{B}\sum_b T_b\right) = \frac{1-\rho}{B}\,\sigma^2 + \rho\,\sigma^2.$$

- $T_b = \hat{\eta}^{(b)}(x)$ at a fixed x; random through the training data and the bootstrap draw
- $\mathrm{Var}(\sum_b T_b) = B\sigma^2 + B(B-1)\rho\sigma^2$ (Check this.)
- more trees shrink the first term, but the trees share one training set, so $\rho > 0$ and $\rho\sigma^2$ remain

Example: Averaging in a simulation

Choose a truth: L17's fitted logistic regression. Keep the $5{,}000$ training accounts; draw their labels from it 200 times; each time, grow bootstrap trees and average B of them.

- at one test account x, the estimate $\hat{\eta}_{\text{bag}}(x)$ changes from one simulated training set to the next: compute its variance over the 200

	$B = 1$	$B = 10$	$B = 100$	limit $\rho\sigma^2$
average variance of $\hat{\eta}_{\text{bag}}(x)$	0.0241	0.0066	0.0049	0.0047

Example: Averaging in a simulation

Choose a truth: L17's fitted logistic regression. Keep the $5{,}000$ training accounts; draw their labels from it 200 times; each time, grow bootstrap trees and average B of them.

- at one test account x, the estimate $\hat{\eta}_{\text{bag}}(x)$ changes from one simulated training set to the next: compute its variance over the 200
- do this at $1{,}000$ test accounts, and average the $1{,}000$ variances

	$B = 1$	$B = 10$	$B = 100$	limit $\rho\sigma^2$
average variance of $\hat{\eta}_{\text{bag}}(x)$	0.0241	0.0066	0.0049	0.0047

Out-of-bag estimates

- each bootstrap sample leaves out some training points

Out-of-bag estimates

- each bootstrap sample leaves out some training points
- so every training point was not used by some of the trees

Out-of-bag estimates

- each bootstrap sample leaves out some training points
- so every training point was not used by some of the trees
- those trees can predict it as if it were new data

Out-of-bag estimates

- each bootstrap sample leaves out some training points
- so every training point was not used by some of the trees
- those trees can predict it as if it were new data

Out-of-bag estimates

- each bootstrap sample leaves out some training points
- so every training point was not used by some of the trees
- those trees can predict it as if it were new data

Definition (Out-of-bag estimate)

For training point i, average $\widehat{\eta}^{(b)}(x_i)$ over only the trees b whose bootstrap sample did not contain i. Comparing these estimates with the y_i estimates the error on new data, with no validation set.

Out-of-bag estimates

- each bootstrap sample leaves out some training points
- so every training point was not used by some of the trees
- those trees can predict it as if it were new data

Definition (Out-of-bag estimate)

For training point i, average $\hat{\eta}^{(b)}(x_i)$ over only the trees b whose bootstrap sample did not contain i. Comparing these estimates with the y_i estimates the error on new data, with no validation set.

Note: Point i is left out with probability $(1 - 1/n)^n \to e^{-1} \approx 0.37$ (why?): about a third of the trees. Used to choose settings, it is part of model selection, not a final test.

Plan for today

• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

Definition: Random forest

Definition (Random forest (Breiman, 2001))

Bagging, except that each split chooses among only m of the p features, drawn at random for that split.

- aim: lower the correlation ρ between trees, so $\rho\sigma^2$ is smaller

Definition: Random forest

Definition (Random forest (Breiman, 2001))

Bagging, except that each split chooses among only m of the p features, drawn at random for that split.

- aim: lower the correlation ρ between trees, so $\rho\sigma^2$ is smaller
- cost: a split may be forced onto a weak feature, so each tree may be worse

Definition: Random forest

Definition (Random forest (Breiman, 2001))

Bagging, except that each split chooses among only m of the p features, drawn at random for that split.

- aim: lower the correlation ρ between trees, so $\rho\sigma^2$ is smaller
- cost: a split may be forced onto a weak feature, so each tree may be worse
- $m = p$ is bagging; Breiman suggests $m \approx \sqrt{p}$ for classification

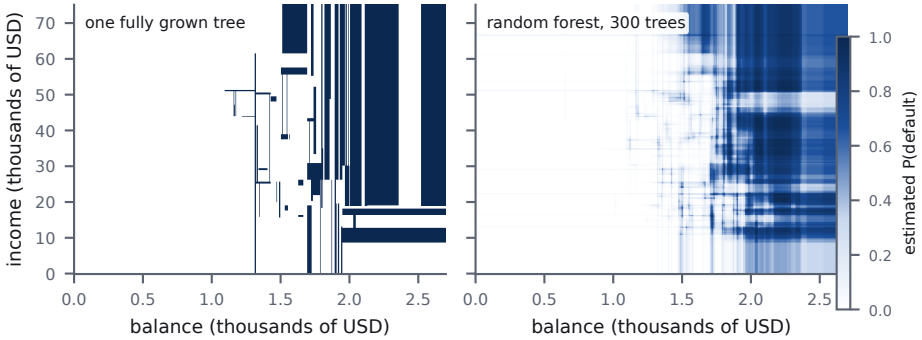
Definition: Random forest

Definition (Random forest (Breiman, 2001))

Bagging, except that each split chooses among only m of the p features, drawn at random for that split.

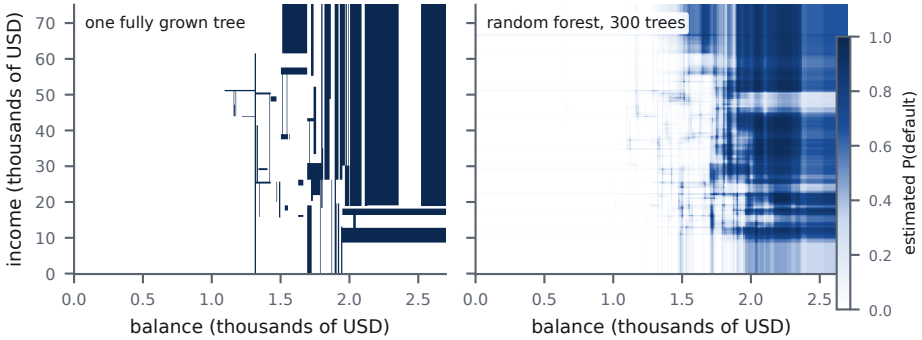
- aim: lower the correlation ρ between trees, so $\rho\sigma^2$ is smaller
- cost: a split may be forced onto a weak feature, so each tree may be worse
- $m = p$ is bagging; Breiman suggests $m \approx \sqrt{p}$ for classification
- Default has $p = 3$ features; we use $m = 1$

Example: One tree, and a forest



- left, no stopping rule: 91 leaves at 1 hold the 166 defaulters, 62 alone; typically $\$6$ of balance wide, and tall where income was never split

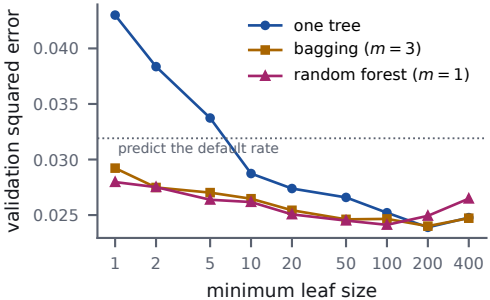
Example: One tree, and a forest



- left, no stopping rule: 91 leaves at 1 hold the 166 defaulters, 62 alone; typically $\$6$ of balance wide, and tall where income was never split
- right, 300 trees ($m = 1$): averages of such estimates

Example: Leaf size, and averaging

Each point: grow one tree, or 100 trees by bagging ($m = 3$) or as a random forest ($m = 1$), on the $4{,}000$ fitting accounts, with the minimum leaf size on the axis; then compute the squared error of the estimates on the $1{,}000$ validation accounts.



With leaves of 200 or more, each tree already averages many accounts, and averaging trees changes little.

Python: A random forest

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=300,
            max_features=1, oob_score=True,
            random_state=0)
forest.fit(X_train, y_train)  # all three features
p_hat = forest.predict_proba(X_test)[:, 1]  # average
p_oob = forest.oob_decision_function_[:, 1]
np.mean((p_oob - y_train) ** 2)  # OOB squared error
```

- `max_features` is m; the classifier's default is $\sqrt{p}$
- by default `oob_score_` is the OOB accuracy

A question

Question

In a forest of fully grown trees, each tree fits its own bootstrap sample almost exactly.

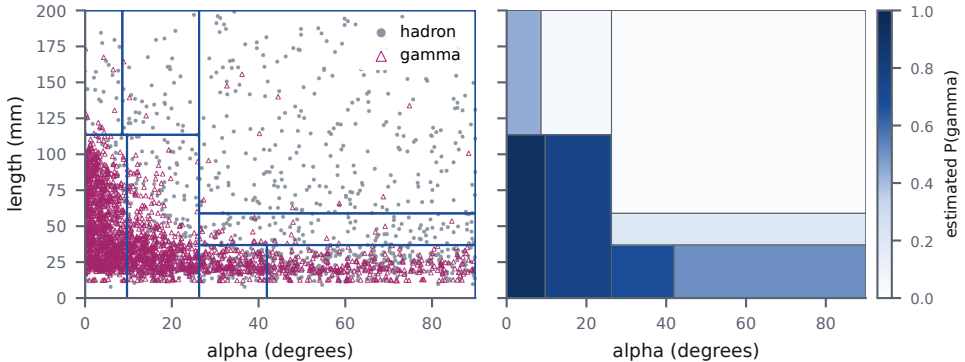
(a) Why can the forest estimate new accounts better than any one of its trees?

(b) Going from 100 to $1{,}000$ trees: what changes, and can it make the forest overfit?

Plan for today

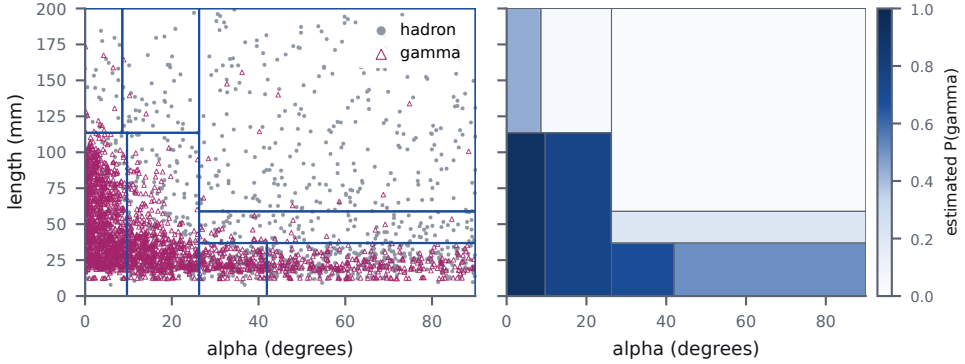
• Trees: predictions and partitions	9 min
• Growing a tree	14 min
• Overfitting and leaf size	6 min
• Bagging	11 min
• Random forests	6 min
• An application	4 min

Application: Gamma rays or background?



- MAGIC telescopes image light from particle showers in the air: gamma ray (signal) or hadron (background)?

Application: Gamma rays or background?



- MAGIC telescopes image light from particle showers in the air: gamma ray (signal) or hadron (background)?
- a depth-3 tree on 2 of 10 numbers; first split: alpha (ellipse's angle to the camera's center) $\leq 26^\circ$

Example: Ten numbers per event

19,020 simulated events (UCI): fit 8,000; choose settings on 2,000; refit on the 10,000; test once on 9,020.

	accuracy	AUC	gammas kept*
logistic regression	79.0%	0.842	33%
one tree (leaves ≥ 20)	84.8%	0.886	—
random forest (300 trees)	87.6%	0.932	60%

*fraction of gammas flagged at the threshold that lets through 5% of hadrons

- the depth-3 tree on two numbers: 78.9%, like logistic regression on ten

Example: Ten numbers per event

19,020 simulated events (UCI): fit 8,000; choose settings on 2,000; refit on the 10,000; test once on 9,020.

	accuracy	AUC	gammas kept*
logistic regression	79.0%	0.842	33%
one tree (leaves ≥ 20)	84.8%	0.886	—
random forest (300 trees)	87.6%	0.932	60%

*fraction of gammas flagged at the threshold that lets through 5% of hadrons

- the depth-3 tree on two numbers: 78.9%, like logistic regression on ten
- on the credit data logistic regression was the better fit; here the forest is the best of these fits

Summary

- trees split the feature space into rectangles and predict the mean response y in each; for $0/1$ labels, this is a class fraction or estimated probability
- grow trees greedily, e.g., using Gini impurity which $=$ least squares on $0/1$ labels
- deep trees have small bias and large variance; choose the leaf size on a validation set
- averaging identically distributed trees keeps their mean and lowers their variance toward $\rho\sigma^2$; random feature choice aims to lower ρ