Ensemble Methods:
Bagging & Random Forest
Announcements
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

Regression dataset \( \mathcal{D} = \{x_i, y_i\}_{i=1}^n \), \( (x_i, y_i) \sim P \)
Recap on Decision (Regression) Tree

Regression dataset \( \mathcal{D} = \{x_i, y_i\}_{i=1}^n, \quad (x_i, y_i) \sim P \)
Recap on Decision (Regression) Tree

Regression dataset \( \mathcal{D} = \{x_i, y_i\}_{i=1}^{n}, \quad (x_i, y_i) \sim P \)
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$, $(x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n, \ (x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

Regression dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P$
Recap on Decision (Regression) Tree

How to split the note, i.e., what is the impurity measure?
Recap on Decision (Regression) Tree

How to split the note, i.e., what is the impurity measure?

Consider a set of training points \( S = \{x_i, y_i\}_{i=1}^{m} \)
Recap on Decision (Regression) Tree

How to split the note, i.e., what is the impurity measure?

Consider a set of training points $S = \{x_i, y_i\}_{i=1}^m$

Define the sample mean $\hat{y}_S = \sum_{i=1}^m y_i / m$
Recap on Decision (Regression) Tree

How to split the note, i.e., what is the impurity measure?

Consider a set of training points $S = \{x_i, y_i\}_{i=1}^{m}$

Define the sample mean $\hat{y}_S = \frac{1}{m} \sum_{i=1}^{m} y_i$

Impurity: sample variance $\hat{Var}(S) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_S)^2$
Recap on Decision (Regression) Tree

The regression Tree algorithm

Regression_Tree( S ): 

Recap on Decision (Regression) Tree

The regression Tree algorithm

Regression_Tree(S):

• IF $|S| \leq k$:
  Set leaf value to be $\bar{y}_S$
Recap on Decision (Regression) Tree

The regression Tree algorithm

Regression_Tree( S ):

• IF $|S| \leq k$:
  
  Set leaf value to be $\bar{y}_S$

• ELSE:
Recap on Decision (Regression) Tree

The regression Tree algorithm

Regression_Tree( $S$):

- IF $|S| \leq k$:
  
  Set leaf value to be $\bar{y}_S$

- ELSE:

  For all dim and all value, find the split such that minimizes $\frac{|S_L|}{|S|} \text{Var} (S_L) + \frac{|S_R|}{|S|} \text{Var} (S_R)$
Recap on Decision (Regression) Tree

The regression Tree algorithm

Regression_Tree( S ):

- IF |S| ≤ k:
  
  Set leaf value to be $\bar{y}_S$

- ELSE:
  
  For all dim and all value, find the split such that minimizes $\frac{|S_L|}{|S|} \text{Var} (S_L) + \frac{|S_R|}{|S|} \text{Var} (S_R)$

  Call Regression_Tree( S_L ) & Regression_Tree( S_R )
Issues of Decision Trees

Decision Tree can have high variance, i.e., overfilling!
Issues of Decision Trees

Decision Tree can have high variance, i.e., overfitting!
Issues of Decision Trees

Decision Tree can have high variance, i.e., overfilling!
Common regularizations in Decision Trees

1. Minimum number of examples per leaf

   No split if # of examples $< \text{threshold}$
Common regularizations in Decision Trees

1. Minimum number of examples per leaf
   No split if # of examples < threshold

2. Maximum Depth
   No split if it hits depth limit
Common regularizations in Decision Trees

1. Minimum number of examples per leaf
   No split if # of examples < threshold

2. Maximum Depth
   No split if it hits depth limit

3. Maximum number of nodes
   Stop the tree if it hits max # of nodes
Outline of Today

1. Variance Reduction using averaging

2. Bagging: Bootstrap Aggregation

3. Random Forest
Variance Reduction via Averaging

Consider i.i.d random variables $\{x_i\}_{i=1}^n$, $x_i \sim \mathcal{N}(0, \sigma^2)$

$$\text{Var}(x_i) = \sigma^2$$
Variance Reduction via Averaging

Consider i.i.d random variables \( \{x_i\}_{i=1}^n \), \( x_i \sim \mathcal{N}(0, \sigma^2) \)

\[
\text{Var}(x_i) = \sigma^2
\]

Q: what is the variance of \( \bar{x} = \sum_{i=1}^{n} x_i / n \)
Variance Reduction via Averaging

Consider i.i.d random variables \( \{x_i\}_{i=1}^n \), \( x_i \sim \mathcal{N}(0,\sigma^2) \)

\[
\text{Var}(x_i) = \sigma^2
\]

Q: what is the variance of \( \bar{x} = \sum_{i=1}^{n} x_i/n \)

A: \( \text{Var}(\bar{x}) = \sigma^2/n \)
Variance Reduction via Averaging

Consider i.i.d random variables \( \{x_i\}_{i=1}^n \), \( x_i \sim \mathcal{N}(0,\sigma^2) \)

\[
\text{Var}(x_i) = \sigma^2
\]

Q: what is the variance of \( \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \)?

A: \( \text{Var}(\bar{x}) = \frac{\sigma^2}{n} \)

Avg significantly reduced variance!
Variance Reduction via Averaging

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n, \quad x_i \sim \mathcal{N}(0, \sigma^2) \)
Variance Reduction via Averaging

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n \), \( x_i \sim \mathcal{N}(0,\sigma^2) \)

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
\end{bmatrix}
\sim \mathcal{N}
\left(
\begin{bmatrix}
  0 \\
  \sigma^2 \\
  \sigma^2 \\
\end{bmatrix}
\right)
\]
Variance Reduction via Averaging

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n, \quad x_i \sim \mathcal{N}(0, \sigma^2) \)

\[
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \sigma^2 & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{2,1} & \sigma^2 & \sigma_{2,3} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma^2 \end{bmatrix} \right)
\]

\[
\sigma_{i,j} = \mathbb{E}[x_i x_j]
\]
Variance Reduction via Averaging

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n, \quad x_i \sim \mathcal{N}(0, \sigma^2) \)

\[
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \sigma^2 & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{2,1} & \sigma^2 & \sigma_{2,3} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma^2 \end{bmatrix} \right)
\]

\[
\sigma_{i,j} = \mathbb{E}[x_i x_j]
\]

Q: what is the variance of \( \bar{x} = \sum_{i=1}^{3} x_i / 3 \)
Variance Reduction via Averaging

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n, \quad x_i \sim \mathcal{N}(0, \sigma^2) \)

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 
\end{bmatrix} \sim \mathcal{N}
\left(0, \begin{bmatrix}
  \sigma^2 & \sigma_{1,2} & \sigma_{1,3} \\
  \sigma_{2,1} & \sigma^2 & \sigma_{2,3} \\
  \sigma_{3,1} & \sigma_{3,2} & \sigma^2 
\end{bmatrix}\right)
\]

Q: what is the variance of \( \bar{x} = \sum_{i=1}^{3} x_i/3 \)

A: \( \text{Var}(\bar{x}) = \sigma^2/3 + \sum_{i \neq j} \sigma_{i,j}/9 \)

\( \sigma_{i,j} = \mathbb{E}[x_i x_j] \)


**Variance Reduction via Averaging**

Consider (possibly correlated) random variables \( \{x_i\}_{i=1}^n, \quad x_i \sim \mathcal{N}(0, \sigma^2) \)

\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix} \sim \mathcal{N}
\begin{pmatrix}
0, \\
\begin{bmatrix}
\sigma^2 & \sigma_{1,2} & \sigma_{1,3} \\
\sigma_{2,1} & \sigma^2 & \sigma_{2,3} \\
\sigma_{3,1} & \sigma_{3,2} & \sigma^2
\end{bmatrix}
\end{pmatrix}
\]

\( \sigma_{i,j} = \mathbb{E}[x_i x_j] \)

**Q:** what is the variance of \( \bar{x} = \frac{1}{3} \sum_{i=1}^{3} x_i / 3 \)?

**A:**

\[
\text{Var}(\bar{x}) = \sigma^2 / 3 + \sum_{i \neq j} \sigma_{i,j} / 9
\]

**Worst case:** when these RVs are positively correlated, averaging may not reduce variance
Outline of Today

1. Variance Reduction using averaging

2. Bagging: Bootstrap Aggregation

3. Random Forest
Why Bagging

Imaging train Decision Tree, i.e., \( \hat{h} = \text{ID3}(\mathcal{D}) \)
Why Bagging

Imaging train Decision Tree, i.e., $\hat{h} = \text{ID3}(\mathcal{D})$

$\hat{h}$ is a random quantity + it has high variance
Why Bagging

Imaging train Decision Tree, i.e., $\hat{h} = \text{ID3}(\mathcal{D})$

$\hat{h}$ is a random quantity + it has high variance

Q: can we learn multiple $\hat{h}$ and perform averaging to reduce variance?
Why Bagging

Imaging train Decision Tree, i.e., $\hat{h} = \text{ID3}(\mathcal{D})$

$\hat{h}$ is a random quantity + it has high variance

Q: can we learn multiple $\hat{h}$ and perform averaging to reduce variance?

Yes, we do this via Bootstrap
Detour: Bootstrapping

Consider dataset $\mathcal{D} = \{z_i\}_{i=1}^n, z_i \sim P$
Detour: Bootstrapping

Consider dataset $\mathcal{D} = \{z_i\}_{i=1}^n$, $z_i \sim P$

Let us approximate $P$ with the following discrete distribution:

$$\hat{P}(z_i) = 1/n, \forall i \in [n]$$
Detour: Bootstrapping

Consider dataset $\mathcal{D} = \{z_i\}_{i=1}^n$, $z_i \sim P$

Let us approximate $P$ with the following discrete distribution:

$$\hat{P}(z_i) = \frac{1}{n}, \forall i \in [n]$$
Bootstrapping

\[ \hat{P}(z_i) = 1/n, \forall i \in [n] \]

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?
Bootstrapping

\[ \hat{P}(z_i) = 1/n, \forall i \in [n] \]

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,
Bootstrapping

\( \hat{P}(z_i) = 1/n, \forall i \in [n] \)

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,

\[
\mathbb{E}_{z \sim \hat{P}}[z] = \sum_{i=1}^{n} \frac{z_i}{n}
\]
Bootstrapping

\[ \hat{P}(z_i) = 1/n, \forall i \in [n] \]

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,

\[
\mathbb{E}_{z \sim \hat{p}}[z] = \sum_{i=1}^{n} \frac{z_i}{n} \rightarrow \mathbb{E}_{z \sim p}[z]
\]
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,

\[
\mathbb{E}_{z \sim \hat{P}}[z] = \sum_{i=1}^{n} \frac{z_i}{n} \to \mathbb{E}_{z \sim P}[z] \\
\mathbb{E}_{z \sim \hat{P}}[z^2] = \sum_{i=1}^{n} \frac{z_i^2}{n}
\]
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,

\[
\mathbb{E}_{z \sim \hat{P}}[z] = \frac{1}{n} \sum_{i=1}^{n} z_i \rightarrow \mathbb{E}_{z \sim P}[z] \\
\mathbb{E}_{z \sim \hat{P}}[z^2] = \frac{1}{n} \sum_{i=1}^{n} z_i^2 \rightarrow \mathbb{E}_{z \sim P}[z^2]
\]
Bootstrapping

\( \hat{P}(z_i) = 1/n, \forall i \in [n] \)

Why \( \hat{P} \) can be regarded as an approximation of \( P \)?

1. We can use \( \hat{P} \) to approximate \( P \)'s mean and variance, i.e.,

\[
\mathbb{E}_{z \sim \hat{P}}[z] = \sum_{i=1}^{n} \frac{z_i}{n} \rightarrow \mathbb{E}_{z \sim P}[z]
\]

\[
\mathbb{E}_{z \sim \hat{P}}[z^2] = \sum_{i=1}^{n} \frac{z_i^2}{n} \rightarrow \mathbb{E}_{z \sim P}[z^2]
\]

2. In fact for any \( f : Z \rightarrow \mathbb{R} \)

\[
\mathbb{E}_{z \sim \hat{P}}[f(z)] = \sum_{i=1}^{n} \frac{f(z_i)}{n} \rightarrow \mathbb{E}_{z \sim P}[f(z)]
\]
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Booststrap: treat \( \hat{P} \) as if it were the ground truth distribution \( P \)!
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Bootstrap: treat \( \hat{P} \) as if it were the ground truth distribution \( P \)!

Now we can draw as many samples as we want from \( \hat{P} \)!
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Bootstrap: treat \( \hat{P} \) as if it were the ground truth distribution \( P \)!

Now we can draw as many samples as we want from \( \hat{P} \)!

Q: What’s the procedure of drawing \( n \) i.i.d samples from \( \hat{P} \)?
Bootstrapping

\[ \hat{P}(z_i) = 1/n, \forall i \in [n] \]

Bootstrap: treat \( \hat{P} \) as if it were the ground truth distribution \( P \)!

Now we can draw as many samples as we want from \( \hat{P} \)!

Q: What’s the procedure of drawing \( n \) i.i.d samples from \( \hat{P} \)?

A: sample uniform randomly from \( \hat{P} \) \( n \) times \textit{w/ replacement}
Bootstrapping

\[ \hat{P}(z_i) = \frac{1}{n}, \forall i \in [n] \]

Bootstrap: treat \( \hat{P} \) as if it were the ground truth distribution \( P \)!

Now we can draw as many samples as we want from \( \hat{P} \)!

Q: What’s the procedure of drawing \( n \) i.i.d samples from \( \hat{P} \)?

A: sample uniform randomly from \( \hat{P} \) \( n \) times \textbf{w/ replacement}

Q: after \( n \) samples, what’s the probability that \( z_1 \) never being sampled?
**Bootstrapping**

\[
\hat{P}(z_i) = 1/n, \forall i \in [n]
\]

Bootstrap: treat \(\hat{P}\) as if it were the ground truth distribution \(P\)!

Now we can draw as many samples as we want from \(\hat{P}\)!

Q: What’s the procedure of drawing \(n\) i.i.d samples from \(\hat{P}\)?

A: sample uniform randomly from \(\hat{P}\) \(n\) times w/ replacement

Q: after \(n\) samples, what’s the probability that \(z_1\) never being sampled?

A: \((1 - 1/n)^n \to 1/e, n \to \infty\)
Bagging: Bootstrap Aggregation

Consider dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n, (x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$
Bagging: Bootstrap Aggregation

Consider dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$

1. Construct $\hat{P}$, s.t., $\hat{P}(x_i, y_i) = 1/n, \forall i \in [n]$
Bagging: Bootstrap Aggregation

Consider dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$

1. Construct $\hat{P}$, s.t., $\hat{P}(x_i, y_i) = 1/n, \forall i \in [n]$

2. Treat $\hat{P}$ as the ground truth, draw $k$ datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ from $\hat{P}$
Bagging: Bootstrap Aggregation

Consider dataset \( \mathcal{D} = \{ x_i, y_i \}_{i=1}^n \), \( (x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\} \)

1. Construct \( \hat{P} \), s.t., \( \hat{P}(x_i, y_i) = 1/n, \forall i \in [n] \)

2. Treat \( \hat{P} \) as the ground truth, draw \( k \) datasets \( \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k \) from \( \hat{P} \)

Bootstrapped samples
Bagging: Bootstrap Aggregation

Consider dataset \( \mathcal{D} = \{x_i, y_i\}_{i=1}^n \), \((x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}\)

1. Construct \( \hat{P} \), s.t., \( \hat{P}(x_i, y_i) = 1/n, \forall i \in [n] \)

2. Treat \( \hat{P} \) as the ground truth, draw \( k \) datasets \( \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k \) from \( \hat{P} \)

3. For each \( i \in [k] \), train classifier, e.g., \( \hat{h}_k = \text{ID3}(\mathcal{D}_k) \)

Bootstrapped samples
Bagging: Bootstrap Aggregation

Consider dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$

1. Construct $\hat{P}$, s.t., $\hat{P}(x_i, y_i) = 1/n, \forall i \in [n]$

2. Treat $\hat{P}$ as the ground truth, draw $k$ datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ from $\hat{P}$

3. For each $i \in [k]$, train classifier, e.g., $\hat{h}_k = \text{ID3}(\mathcal{D}_k)$

4. Averaging / Aggregation, i.e., $\overline{h} = \sum_{i=1}^k \hat{h}_i/k$
Bagging: Bootstrap Aggregation

Consider dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, $(x_i, y_i) \sim P, x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$

1. Construct $\hat{P}$, s.t., $\hat{P}(x_i, y_i) = 1/n, \forall i \in [n]$

2. Treat $\hat{P}$ as the ground truth, draw $k$ datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ from $\hat{P}$

3. For each $i \in [k]$, train classifier, e.g., $\hat{h}_k = \text{ID3}(\mathcal{D}_k)$

4. Averaging / Aggregation, i.e., $\bar{h} = \sum_{i=1}^k \hat{h}_i / k$

The step that reduces Var!

Bootstrapped samples
Bagging in Test Time

Given a test example $x_{test}$

We can use $\{\hat{h}_i\}_{i=1}^k$ to form a distribution over labels:

$$\hat{y} = \begin{bmatrix} p \\ 1 - p \end{bmatrix}$$
Bagging in Test Time

Given a test example $x_{test}$

We can use $\{\hat{h}_i\}_{i=1}^k$ to form a distribution over labels:

$$\hat{y} = \begin{bmatrix} p \\ 1 - p \end{bmatrix}$$

where:

$$p = \frac{\text{# of trees predicting } -1}{k}$$
Bagging reduces variance

$$\bar{h} = \frac{1}{k} \sum_{i=1}^{k} \hat{h}_i$$

What happens when $k \to \infty$?
Bagging reduces variance

\[ \bar{h} = \frac{1}{k} \sum_{i=1}^{k} \hat{h}_i \]

What happens when \( k \to \infty \)?

\[ \bar{h} \to \mathbb{E}_{\mathcal{D} \sim \hat{p}} [\text{ID3}(\mathcal{D})] \]
Bagging reduces variance

\[ \bar{h} = \frac{1}{k} \sum_{i=1}^{k} \hat{h}_i \]

What happens when \( k \to \infty \)?

\[ \bar{h} \to \mathbb{E}_{\mathcal{D} \sim \hat{P}} \left[ \text{ID3} (\mathcal{D}) \right] \]

\[ \hat{P} \to P, \text{ when } n \to \infty \]
Bagging reduces variance

$$\bar{h} = \frac{1}{k} \sum_{i=1}^{k} \hat{h}_i$$

What happens when $k \to \infty$?

$$\bar{h} \to \mathbb{E}_{\mathcal{D} \sim \hat{P}} [\text{ID3}(\mathcal{D})]$$

$$\hat{P} \to P, \text{ when } n \to \infty$$

$$\mathbb{E}_{\mathcal{D} \sim P} [\text{ID3}(\mathcal{D})]$$
Bagging reduces variance

\[ \bar{h} = \frac{1}{k} \sum_{i=1}^{k} \hat{h}_i \]

What happens when \( k \to \infty \)?

\[ \bar{h} \to \mathbb{E}_{\hat{D} \sim \hat{P}} \left[ \text{ID3}(\mathcal{D}) \right] \]

\[ \hat{P} \to P, \text{ when } n \to \infty \]

\[ \mathbb{E}_{\mathcal{D} \sim P} \left[ \text{ID3}(\mathcal{D}) \right] \]

The expected decision tree (under true \( P \))
Bagging reduces variance

\[ \overline{h} = \sum_{i=1}^{k} \frac{\hat{h}_i}{k} \]

What happens when \( k \to \infty \)?

\[ \overline{h} \to \mathbb{E}_{\mathcal{D} \sim \hat{P}}[\text{ID3}(\mathcal{D})] \]

\( \hat{P} \to P \), when \( n \to \infty \)

\[ \mathbb{E}_{\mathcal{D} \sim P}[\text{ID3}(\mathcal{D})] \]

The expected decision tree (under true \( P \))

Deterministic, i.e., zero variance
Outline of Today

1. Variance Reduction using averaging

2. Bagging: Bootstrap Aggregation

3. Random Forest
Motivation of Random Forest

Consider any two hypothesis $\hat{h}_i, \hat{h}_j, i \neq j$ in Bagging
Motivation of Random Forest

Consider any two hypothesis $\hat{h}_i, \hat{h}_j, i \neq j$ in Bagging

$\hat{h}_j, \hat{h}_i$ are not independent under true distribution $P$
Motivation of Random Forest

Consider any two hypothesis \( \hat{h}_i, \hat{h}_j, i \neq j \) in Bagging

\( \hat{h}_j, \hat{h}_i \) are not independent under true distribution \( P \)

\[ \text{e.g., } D_i, D_j \text{ have overlap samples} \]
Motivation of Random Forest

Consider any two hypothesis $\hat{h}_i, \hat{h}_j, i \neq j$ in Bagging

$\hat{h}_j, \hat{h}_i$ are not independent under true distribution $P$

e.g., $\mathcal{D}_i, \mathcal{D}_j$ have overlap samples

Recall that: $\text{Var}(\bar{x}) = \frac{\sigma^2}{3} + \sum_{i \neq j} \frac{\sigma_{i,j}}{9}$
Motivation of Random Forest

Consider any two hypothesis $\hat{h}_i, \hat{h}_j, i \neq j$ in Bagging

$\hat{h}_j, \hat{h}_i$ are not independent under true distribution $P$

e.g., $\mathcal{D}_i, \mathcal{D}_j$ have overlap samples

Recall that: $\text{Var}(\bar{x}) = \sigma^2/3 + \sum_{i \neq j} \sigma_{i,j}/9$

To avoid positive correlation, we want to make $\hat{h}_i, \hat{h}_j$ as independent as possible
Random Forest

Key idea:
In ID3, for every split, randomly select $k$ ($k < d$) many features, find the split only using these $k$ features
Random Forest

Key idea:
In ID3, for every split, randomly select $k$ ($k < d$) many features, find the split only using these $k$ features
Random Forest

Key idea:
In ID3, for every split, randomly select $k$ ($k < d$) many features, find the split only using these $k$ features

Regular ID3: looking for split in all $d$ dimensions
Random Forest

Key idea:
In ID3, for every split, **randomly select** \( k \) \((k < d)\) many features, find the split **only using these \( k \) features**

Regular ID3: looking for split in all \( d \) dimensions
ID3 in RF: looking for split in \( k \) randomly picked dimensions
Benefit of Random Forest

By always randomly selecting subset of features for every tree, and every split:

We further reduce the correlation between $\hat{h}_i$ & $\hat{h}_j$. 
Demo of Random Forest

DT w/ Depth 10
Demo of Random Forest

DT w/ Depth 10

RF w/ 2 trees
Demo of Random Forest

- DT w/ Depth 10
- RF w/ 2 trees
- RF w/ 5 trees
Demo of Random Forest

DT w/ Depth 10

RF w/ 2 trees

RF w/ 5 trees

RF w/ 10 trees

RF w/ 20 trees

RF w/ 50 trees
Summary for today

An approach to reduce the variance of our classifier:
Summary for today

An approach to reduce the variance of our classifier:

1. Create datasets via bootstrapping + train classifiers on them + averaging
Summary for today

An approach to reduce the variance of our classifier:

1. Create datasets via bootstrapping + train classifiers on them + averaging

2. To further reduce correlation between classifiers, RF randomly selects subset of dimensions for every split.