Classification And Regression Trees (CART) & Random Forest

Richard Alligier, David Gianazza & Pascal Lezaud

ENAC

Plan

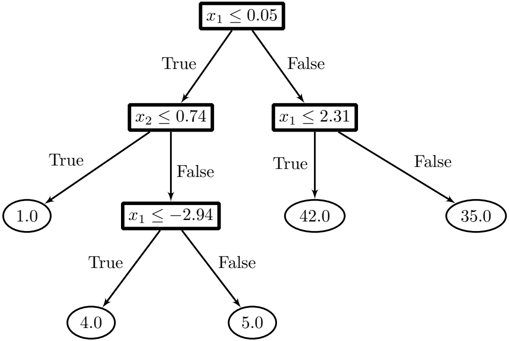
1. Classification And Regression Trees (CART) [Breiman et al., 1984]
2. Bagging [Breiman, 1996]
3. Random Forest [Breiman, 2001]

Plan

1. Classification And Regression Trees (CART) [Breiman et al., 1984]
2. Bagging [Breiman, 1996]
3. Random Forest [Breiman, 2001]

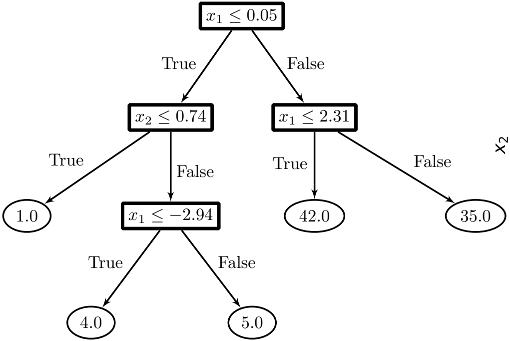
Introduction to Decision Trees

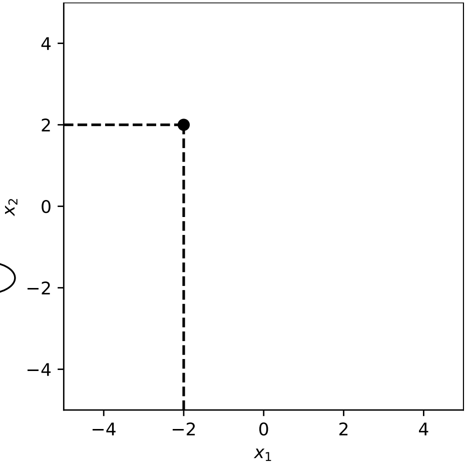
A decision tree is a “cascade” of questions. At the bottom end, there is the predicted value.



Introduction to Decision Trees

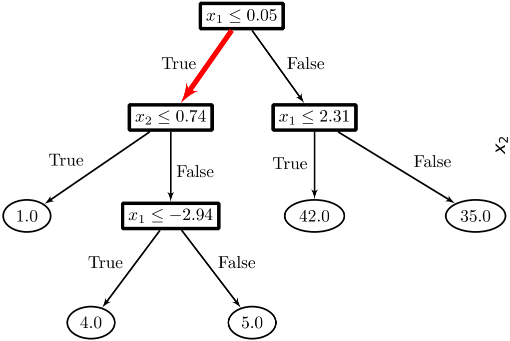
A decision tree is a "cascade" of questions. At the bottom end, there is the predicted value.

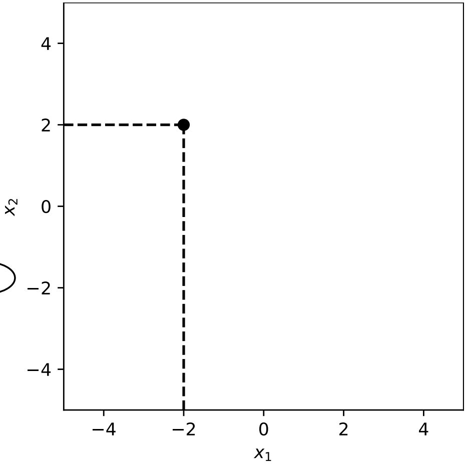




Introduction to Decision Trees

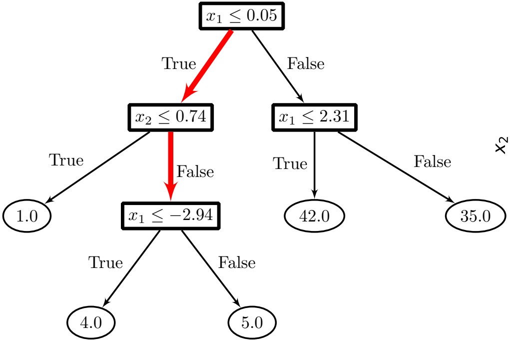
A decision tree is a “cascade” of questions. At the bottom end, there is the predicted value.

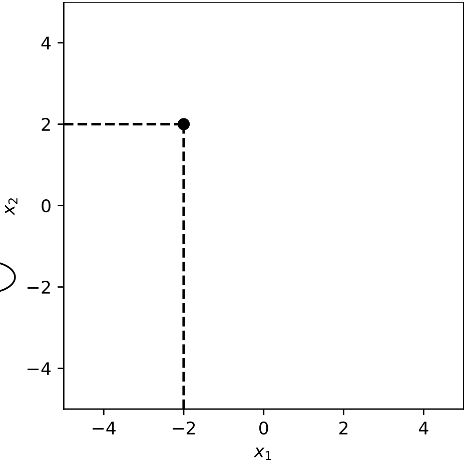




Introduction to Decision Trees

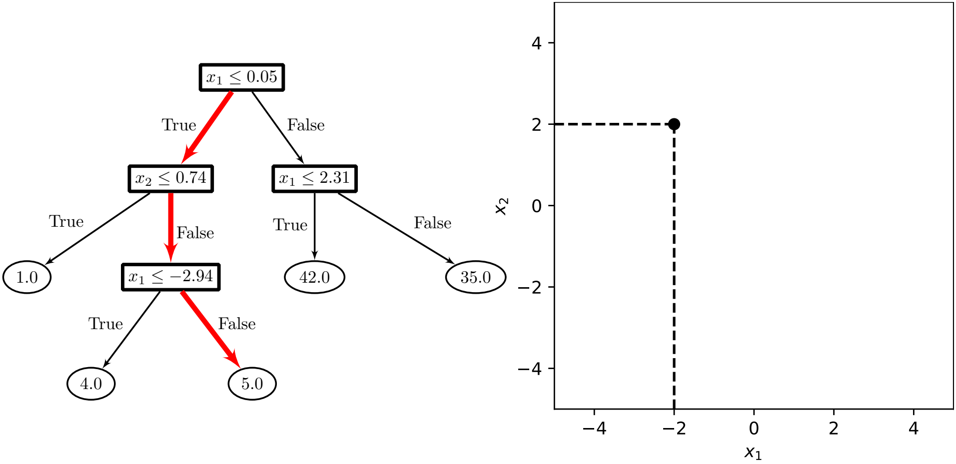
A decision tree is a “cascade” of questions. At the bottom end, there is the predicted value.





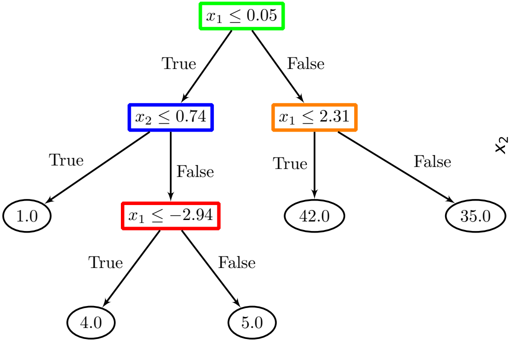
Introduction to Decision Trees

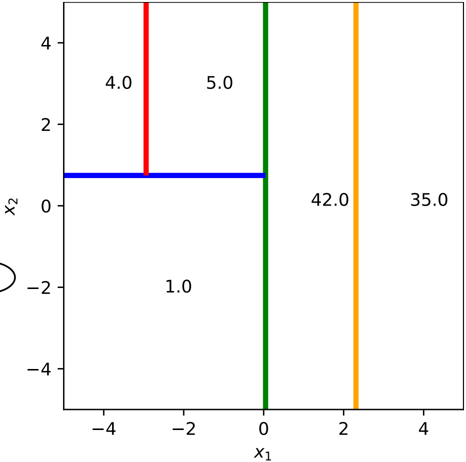
A decision tree is a “cascade” of questions. At the bottom end, there is the predicted value.



Introduction to Decision Trees

A decision tree is a “cascade” of questions. At the bottom end, there is the predicted value.





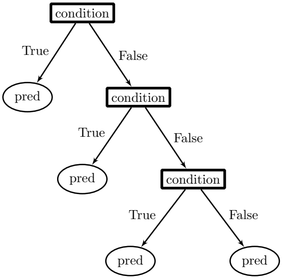
A decision tree encodes a partition of the input space into regions.

How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem

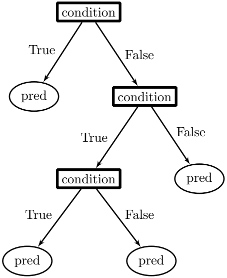
How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem



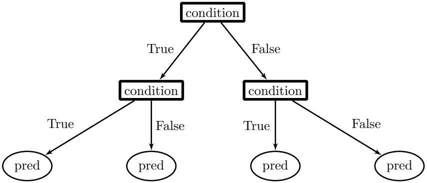
How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem



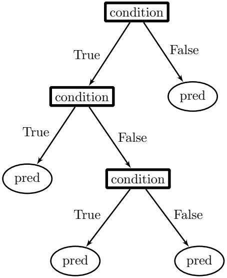
How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem



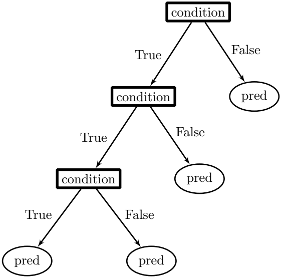
How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem



How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem



How the Tree is Built ?

Building a tree minimizing $\sum_{i=1}^{N} \ell(y_i, h(x_i))$ is a highly combinatorial problem

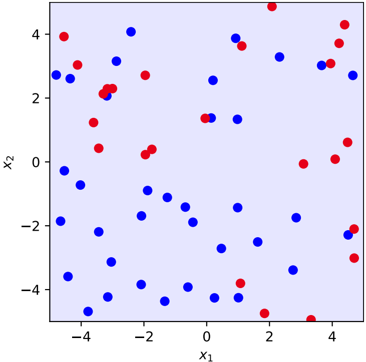
A Greedy Algorithm that Grows the Tree

Computationaly efficient, but do not produce the optimal partitionning

Data: A set of examples $\{(x_i, y_i) | \forall i \in [\![1; N]\!]\}$
Result: Decision tree
initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

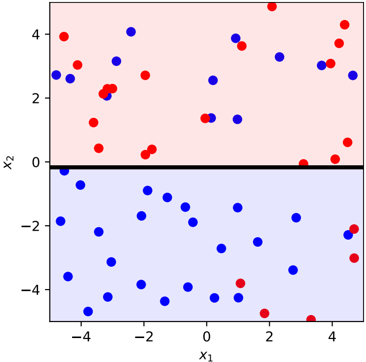
How the Tree is Built ?

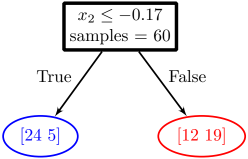
initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region,
  split it: replace the leaf by a node;
end



How the Tree is Built ?

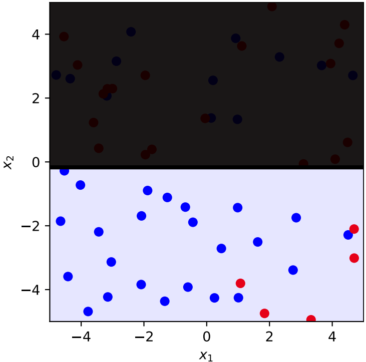
initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

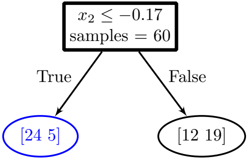




How the Tree is Built ?

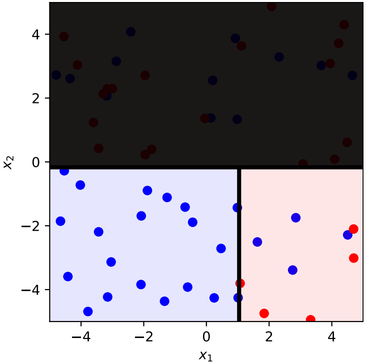
initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

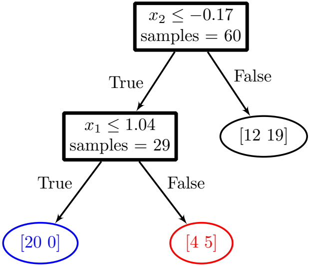




How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

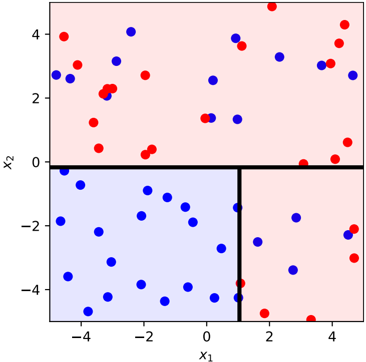


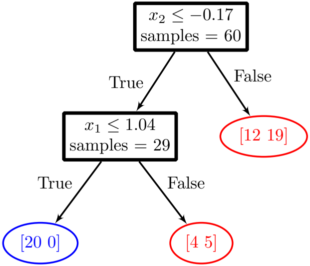


[Breiman et al., 1984]

How the Tree is Built ?

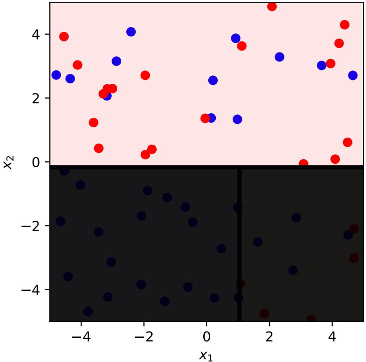
initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

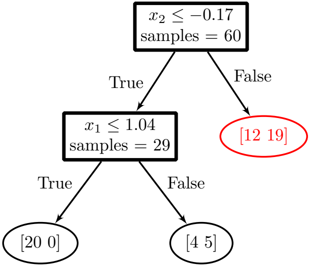




How the Tree is Built ?

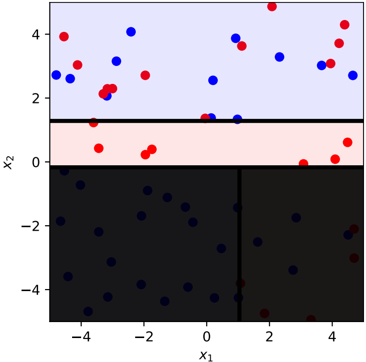
initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

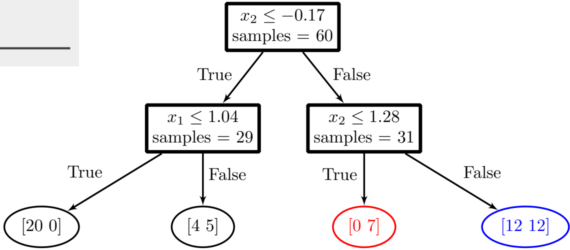




How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

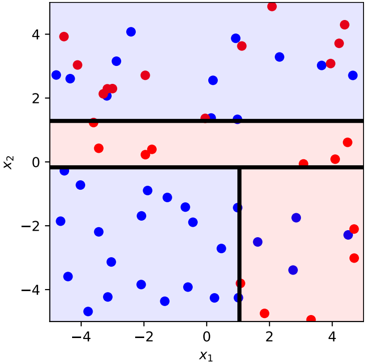


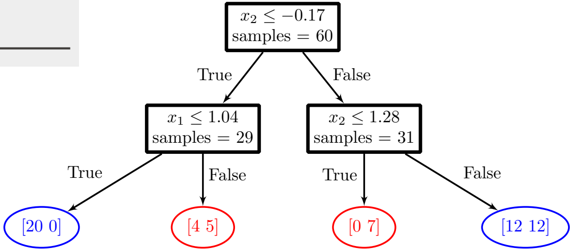


[Breiman et al., 1984]

How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
    Using examples in the region, split it: replace the leaf by a node;
end

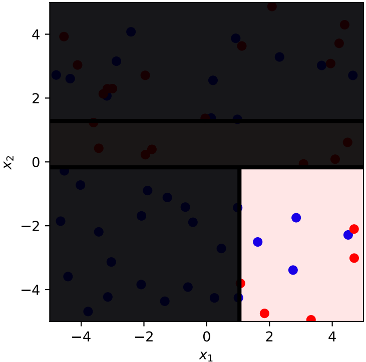


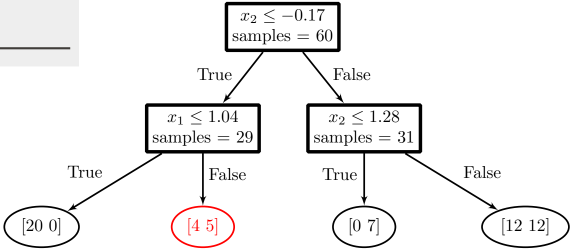


[Breiman et al., 1984]

How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

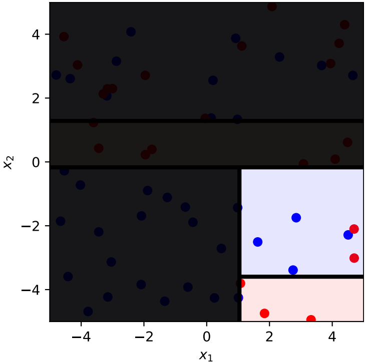


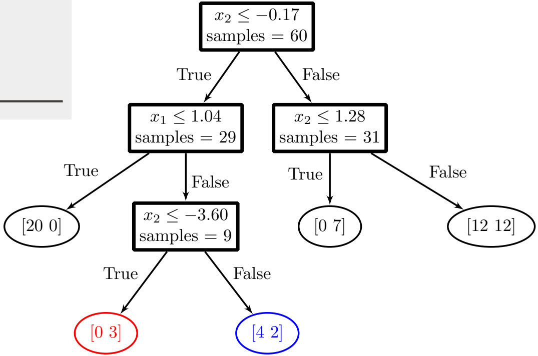


[Breiman et al., 1984]

How the Tree is Built ?

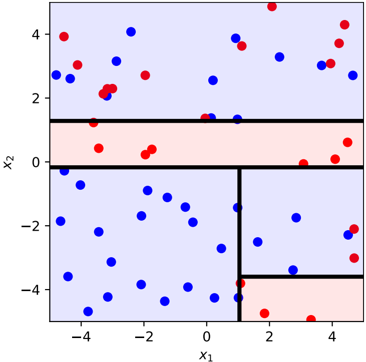
initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

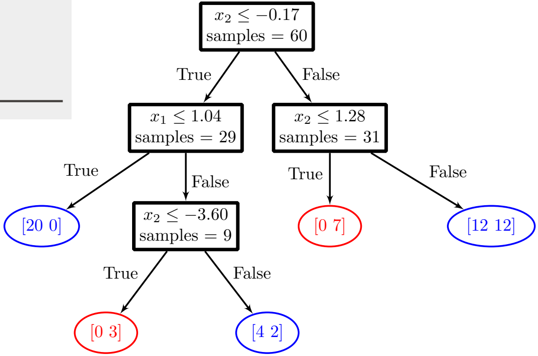




How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

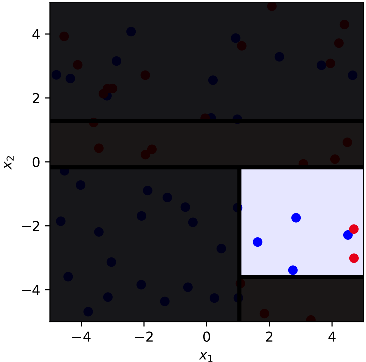


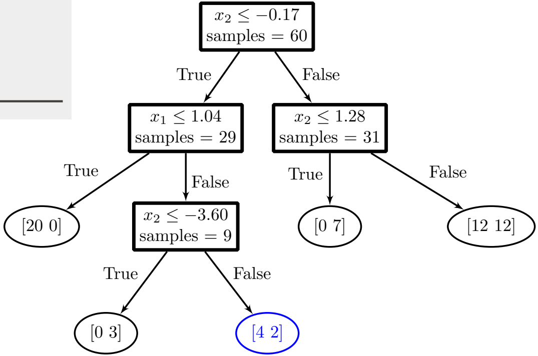


[Breiman et al., 1984]

How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

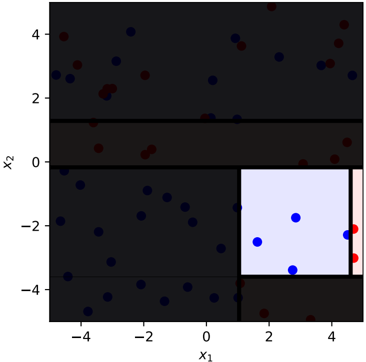


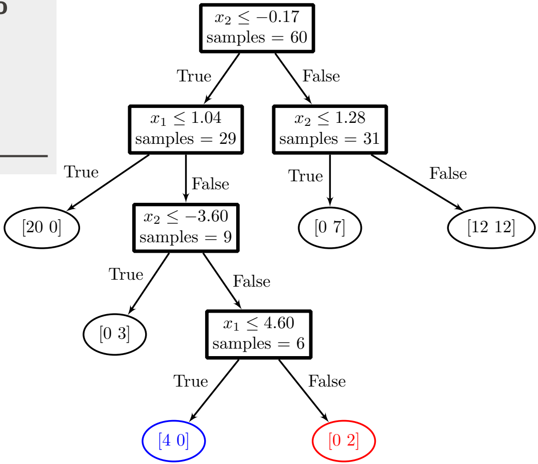


How the Tree is Built ?

---

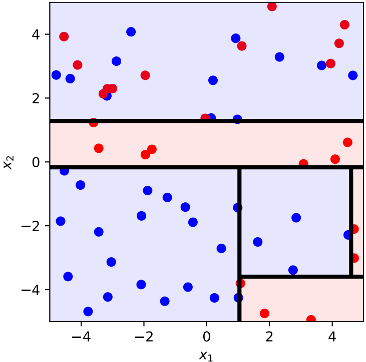
initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end

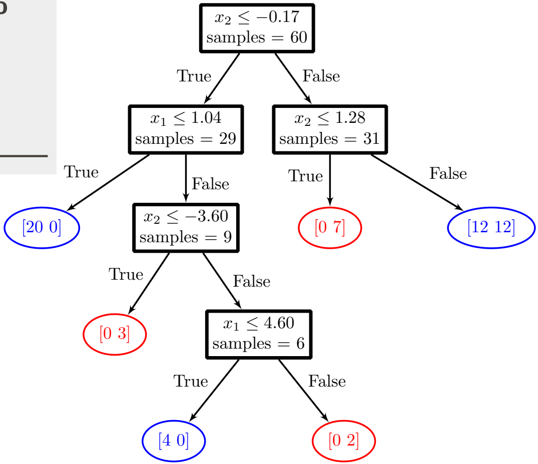




How the Tree is Built ?

initialize a tree as one leaf;
while *there is a splittable region* **do**
  Using examples in the region, split it: replace the leaf by a node;
end





Choose the Condition in the Node Replacing the Leaf

Considered Conditions

The considered conditions use only one variable

- Numerical variable: $X_j \leq t$, where t is a threshold value
- Categorical variable: $X_j = \text{Category}_{j,k}$

How to choose variable j (and threshold t) ?

We want to split the region R, we define:

$$R^{(l)}(j,t) = \{y_i \mid \forall i \in [\![1;N]\!] \ / \ x_i \in R \text{ and } x_{i,j} \leq t\}$$
$$R^{(r)}(j,t) = \{y_i \mid \forall i \in [\![1;N]\!] \ / \ x_i \in R \text{ and } x_{i,j} > t\}$$

We use a function H measuring the “heterogeneity”
Choose j and t minimizing the “heterogeneity” inside the new regions:

$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

Choose the Condition in the Node Replacing the Leaf

$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

Choice of H quantifying the "heterogeneity"

- For regression, we use $H(Y) = \min_c \frac{1}{|Y|} \sum_{y \in Y} \ell(y, c)$
  - L2-loss: $H(Y) = \frac{1}{|Y|} \sum_{y \in Y} (y - \mathrm{mean}(Y))^2$
- For classification, we note $p_k = \frac{1}{|Y|} \sum_{y \in Y} \mathbb{1}(y = k)$
  - Cross-entropy: $H(Y) = -\sum_{k=1}^{K} p_k \log p_k$
  - Gini impurity: $H(Y) = \sum_{k=1}^{K} p_k (1 - p_k)$

Choose the Condition in the Node Replacing the Leaf

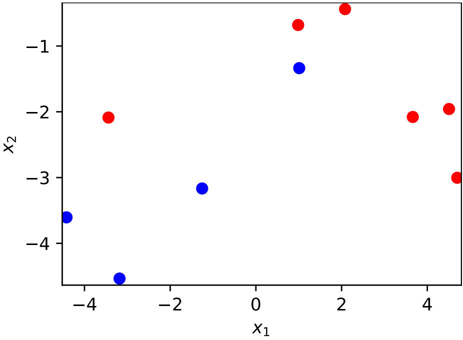
$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

How to find (j,t) minimizing G ?

$$j,t = \underset{j,t}{\operatorname{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1;p]\!]$, with p input features
- $t \in \mathbb{R} \leftarrow$ hard to enumerate



Choose the Condition in the Node Replacing the Leaf

$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

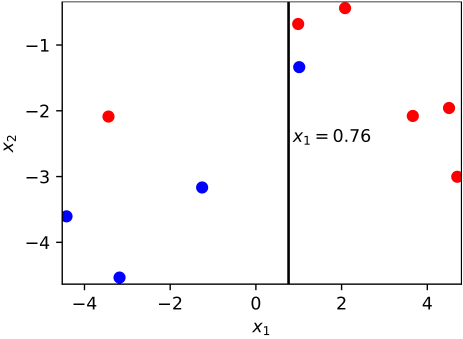
How to find (j,t) minimizing G ?

$$j,t = \underset{j,t}{\operatorname{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1;p]\!]$, with p input features
- $t \in \mathbb{R} \leftarrow$ hard to enumerate

Let us test $j = 1$ and $t = 0.76$



Choose the Condition in the Node Replacing the Leaf

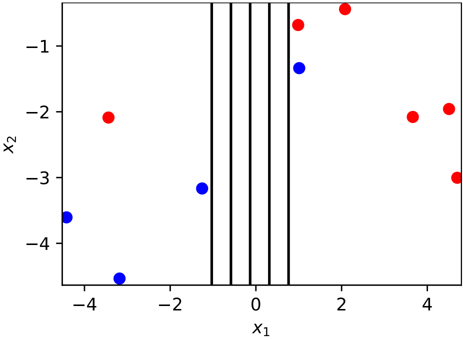
$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

How to find (j,t) minimizing G ?

$$j,t = \underset{j,t}{\text{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1;p]\!]$, with p input features
- $t \in \mathbb{R} \leftarrow$ hard to enumerate



Choose the Condition in the Node Replacing the Leaf

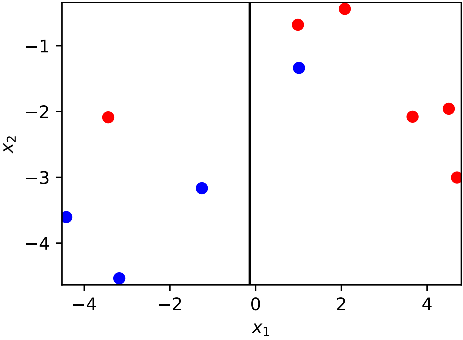
$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

How to find (j,t) minimizing G ?

$$j, t = \underset{j,t}{\operatorname{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1; p]\!]$, with p input features
- $t \in \mathbb{R} \leftarrow$ hard to enumerate



Choose the Condition in the Node Replacing the Leaf

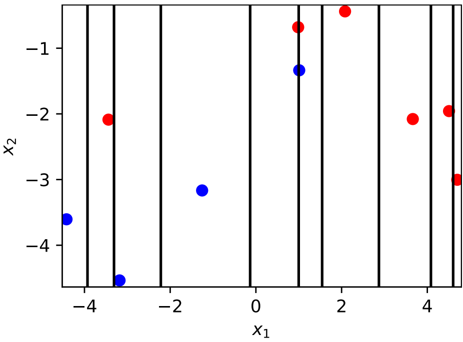
$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

How to find (j,t) minimizing G ?

$$j,t = \underset{j,t}{\operatorname{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1; p]\!]$, with p input features
- For a fixed j, we sort the values of the feature j: $x_{(1),j} \leq \ldots \leq x_{(|R|),j}$ Then, we only have to test: $t \in \{\frac{x_{(i-1),j} + x_{(i),j}}{2} \mid \forall i \in [\![2; |R|]\!]\}$



Choose the Condition in the Node Replacing the Leaf

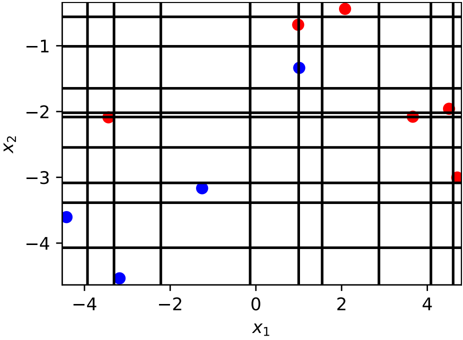
$$G(j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

How to find (j,t) minimizing G ?

$$j,t = \underset{j,t}{\operatorname{argmin}}\ G(j,t)$$

Generate and test all the possibilities !

- $j \in [\![1;p]\!]$, with p input features
- For a fixed j, we sort the values of the feature j: $x_{(1),j} \leq \ldots \leq x_{(|R|),j}$
  Then, we only have to test:
  $t \in \{\frac{x_{(i-1),j}+x_{(i),j}}{2} \mid \forall i \in [\![2;|R|]\!]\}$



$(|R|-1)p$ possibilities for (j,t)

Which Value is Predicted inside Each Region ?

Our data consists of N observations $\{(x_i, y_i) | \forall i \in [\![1; N]\!]\}$
In region $R^{(\text{new})}$ we predict:

$$c = \underset{c}{\operatorname{argmin}} \sum_{i / x_i \in R^{(\text{new})}} \ell(y_i, c)$$

Classification

- misclassification loss: $\ell(y, \hat{y}) = 0$ if $y = \hat{y}$ else 1
  $c = \text{Majority}(\{y_i | \forall i \in [\![1; N]\!] / x_i \in R^{(\text{new})}\})$

Regression

- quadratic loss: $\ell(y, \hat{y}) = (y - \hat{y})^2$
  $c = \text{Avg}(\{y_i | \forall i \in [\![1; N]\!] / x_i \in R^{(\text{new})}\})$
- L1-loss: $\ell(y, \hat{y}) = |y - \hat{y}|$
  $c = \text{Median}(\{y_i | \forall i \in [\![1; N]\!] / x_i \in R^{(\text{new})}\})$

When a Region is Splittable ?

If we consider that a region is splittable till it contains only 1 example then we obtain a very a large tree with a null training error
$\Rightarrow$ Setting a splittability criteria is a way to control the model complexity

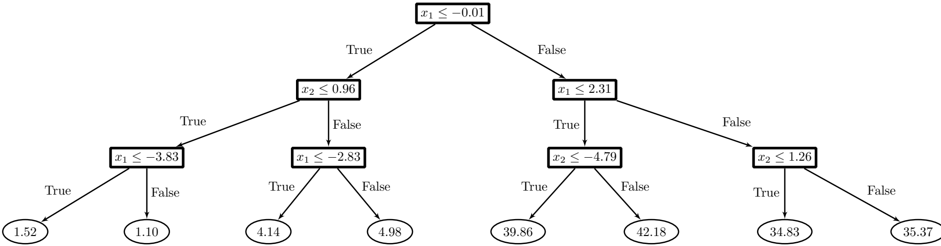
Strategies to control the complexity of the model

- Set $n_{\min}$, the minimum number of examples in each leaf
  $\Rightarrow R$ must contain at least $2n_{\min}$
- Set a threshold $\mathrm{mindecrease}$, split is allowed iff it reduces "heterogeneity" by at least $\mathrm{mindecrease}$
- CART (Classification and Regression Trees): a two steps strategy [Breiman et al., 1984]
  1. Grow a large tree T_0
  2. Prune T_0 using *weakest link pruning*

Pruning a Tree

Pruning

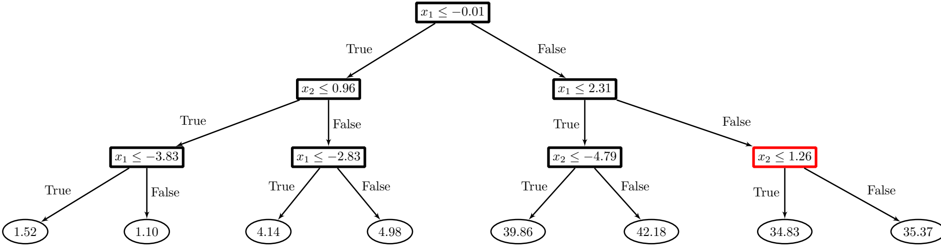
Replace some nodes by leaves



Pruning a Tree

Pruning

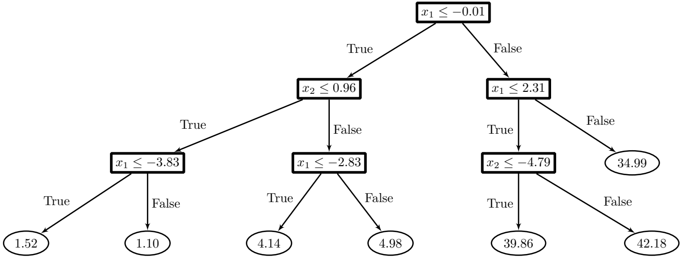
Replace some nodes by leaves



Pruning a Tree

Pruning

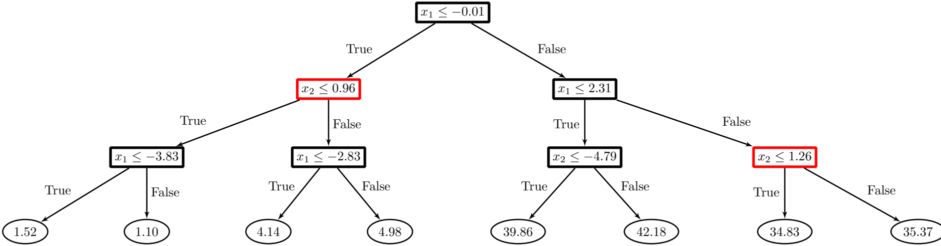
Replace some nodes by leaves



Pruning a Tree

Pruning

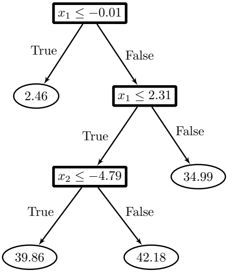
Replace some nodes by leaves



Pruning a Tree

Pruning

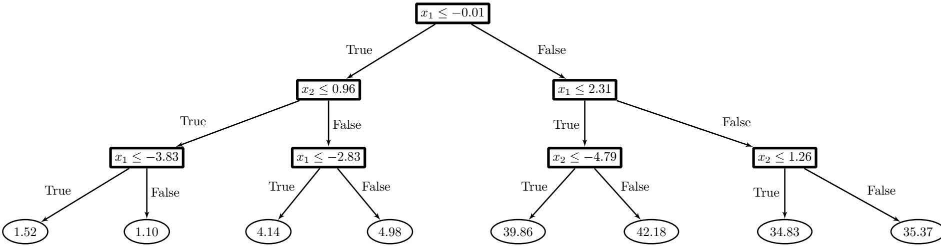
Replace some nodes by leaves



Pruning a Tree

Pruning

Replace some nodes by leaves



A Lot of Possibilities !

Let us note $P(h)$ the number of trees we can obtain by pruning one full binary tree of height h:
$P(h) = P(h-1)^2 + 1$ with $P(0) = 0$; $\rightarrow P(10) \simeq 3.8 \times 10^{90}$

Weakest Link Pruning

- Let us note $E(T) = \sum_{m=1}^{|T|} \sum_{y \in R_m} \ell(y, c_m)$
- we define the cost complexity criterion with $\alpha \geq 0$,

$$C_\alpha(T) = E(T) + \alpha|T|.$$

$$T_\alpha = \underset{T \subset T_0}{\operatorname{argmin}}\ C_\alpha(T)$$

- C_α expresses a compromise, set by the hyperparameter α, between the tree cost $E(T)$ and its complexity $|T|$ (number of leaves)

T_α is Easy to Compute !

Data: A full grown tree T_0 and α
Result: A pruned tree T_α
$T = T_0$;
while $\min_{u \in \mathrm{node}(T)} g(u) \leq \alpha$ **do**
  $u_{\min} = \operatorname{argmin}_{u \in \mathrm{node}(T)} g(u)$; `// is called the weakest link`
  modify T: replace node $u_{\min}$ by a leaf;
end
return T

The choice of the node to replace is based on this criteria:

$$g(u) = \frac{E(f_u) - E(T_u)}{|T_u| - 1}$$

Why this criteria:

$$g(u) \leq \alpha \Leftrightarrow C_\alpha(f_u) \leq C_\alpha(T_u) \Leftrightarrow C_\alpha(\text{“}T - T_u + f_u\text{”}) \leq C_\alpha(T)$$

[Breiman et al., 1984]

T_α is Easy to Compute !

$T = T_0$;

while $\min_{u \in T} g(u) \leq \alpha$ **do**

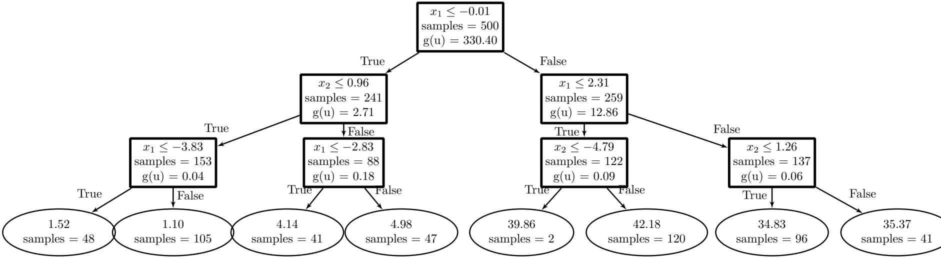
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;

  replace node $u_{\min}$ by leaf;

end

return T

$\alpha = 0.1$



[Breiman et al., 1984]

T_α is Easy to Compute !

$T = T_0$;

while $\min_{u \in T} g(u) \leq \alpha$ **do**

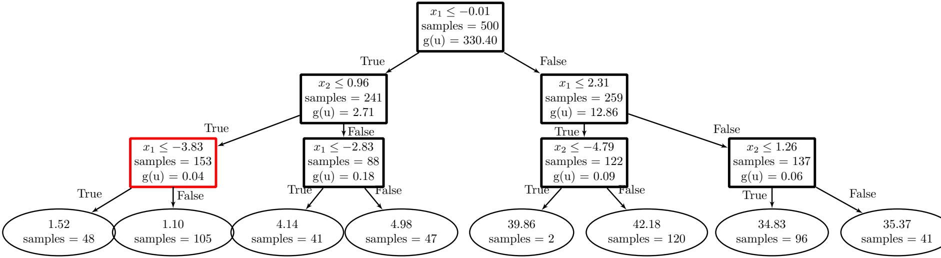
    $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;

    replace node $u_{\min}$ by leaf;

end

return T

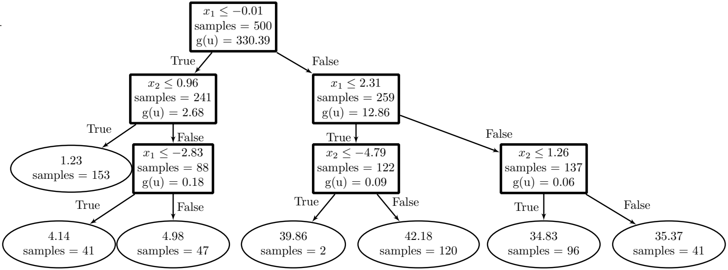
$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;
while $\min_{u \in T} g(u) \leq \alpha$ **do**
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;
  replace node $u_{\min}$ by leaf;
end
return T

$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;

while $\min_{u \in T} g(u) \leq \alpha$ **do**

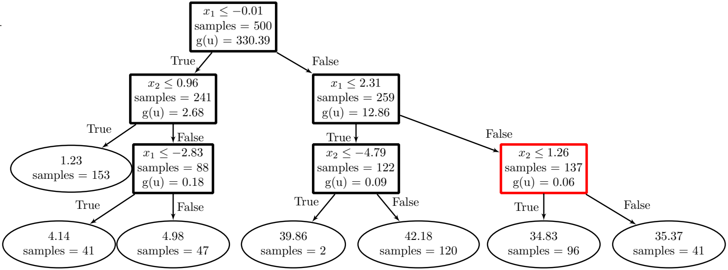
$u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;

replace node $u_{\min}$ by leaf;

end

return T

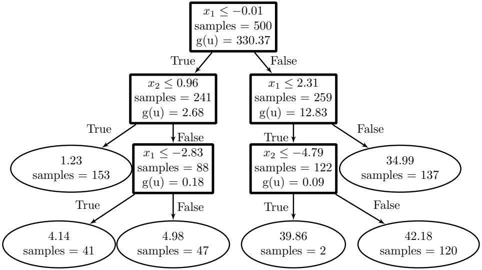
$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;
while $\min_{u \in T} g(u) \leq \alpha$ **do**
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;
  replace node $u_{\min}$ by leaf;
end
return T

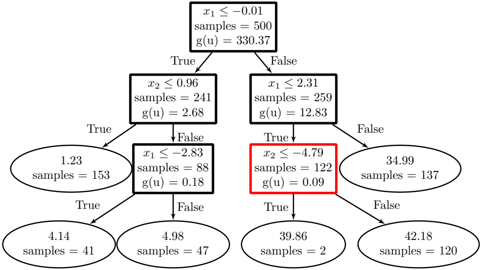
$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;
while $\min_{u \in T} g(u) \leq \alpha$ **do**
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;
  replace node $u_{\min}$ by leaf;
end
return T

$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;

while $\min_{u \in T} g(u) \leq \alpha$ **do**

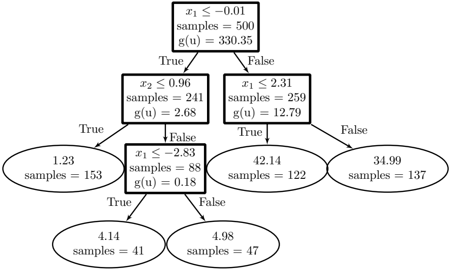
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;

  replace node $u_{\min}$ by leaf;

end

return T

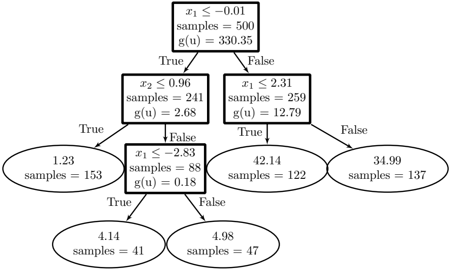
$\alpha = 0.1$



T_α is Easy to Compute !

$T = T_0$;
while $\min_{u \in T} g(u) \leq \alpha$ **do**
  $u_{\min} = \operatorname{argmin}_{u \in T} g(u)$;
  replace node $u_{\min}$ by leaf;
end
return T

$\alpha = 0.1$



$T_0 \supset T_{\alpha_1} \supset \ldots \supset T_{\alpha_k}$, with $0 < \alpha_1 < \ldots < \alpha_k$

Advantages and Disadvantages of Decision Trees

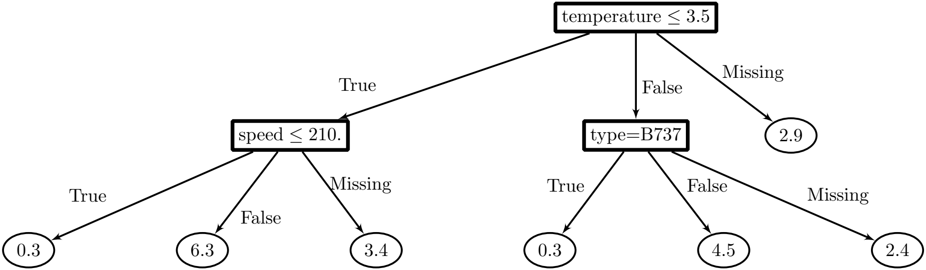
Advantages

- No variable scaling/normalization required
- Can handle numerical and categorical variable without pre-processing
- Can easily manage missing variable
- Relatively undisturbed by outliers (they are isolated in small nodes)
- Embeds a feature selection
- Interpretability: the feature space partition is fully described by a single tree

Advantages and Disadvantages of Decision Trees

Advantages

- No variable scaling/normalization required
- Can handle numerical and categorical variable without pre-processing
- Can easily manage missing variable
- Relatively undisturbed by outliers (they are isolated in small nodes)
- Embeds a feature selection
- Interpretability: the feature space partition is fully described by a single tree



Advantages and Disadvantages of Decision Trees

Advantages

- No variable scaling/normalization required
- Can handle numerical and categorical variable without pre-processing
- Can easily manage missing variable
- Relatively undisturbed by outliers (they are isolated in small nodes)
- Embeds a feature selection
- Interpretability: the feature space partition is fully described by a single tree

Disadvantages

- Lack of smoothness (rectangular regions) with a constant prediction
- There are concepts that are hard to learn because decision trees do not express them easily
- Instability of Trees: a small change in the data can result in a very different series of splits.

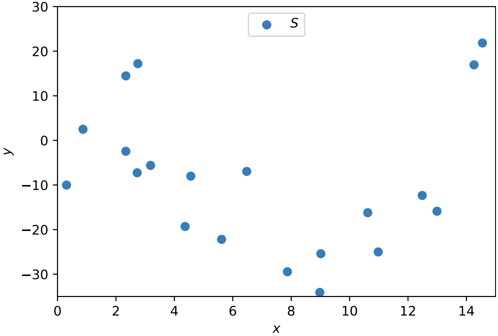
Plan

1. Classification And Regression Trees (CART) [Breiman et al., 1984]
2. Bagging [Breiman, 1996]
3. Random Forest [Breiman, 2001]

Introduction to Bagging

Supervised Machine Learning

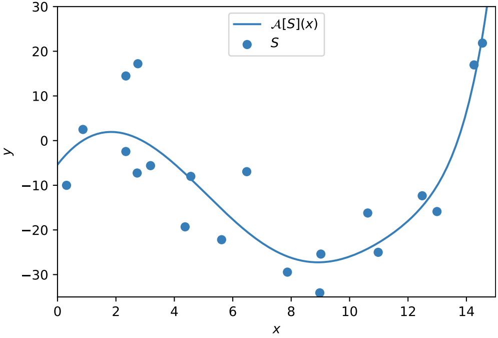
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

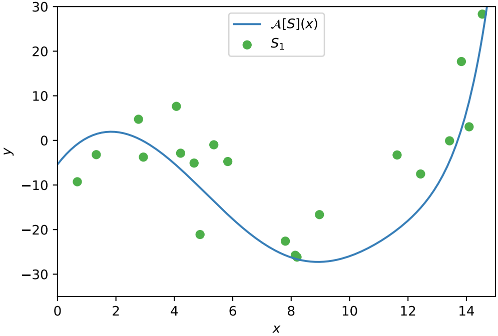
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

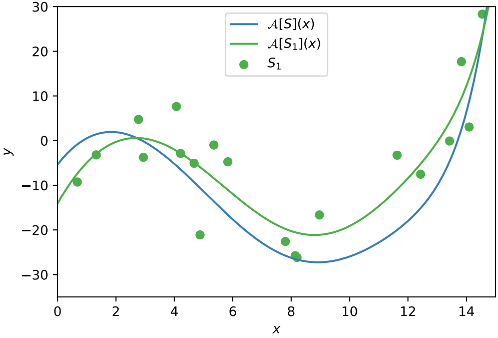
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

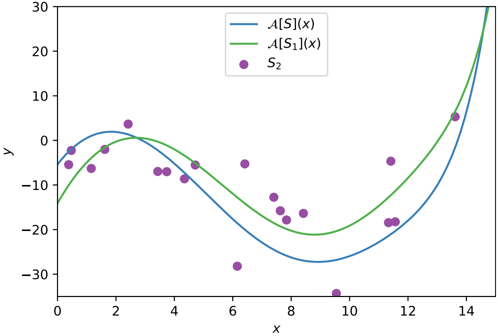
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

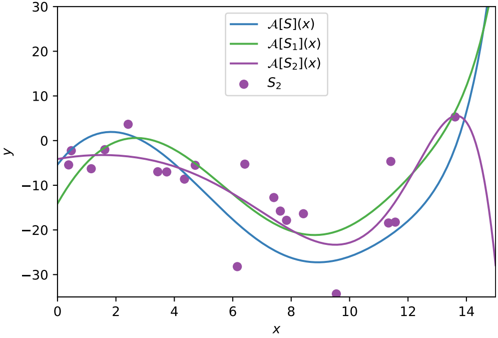
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

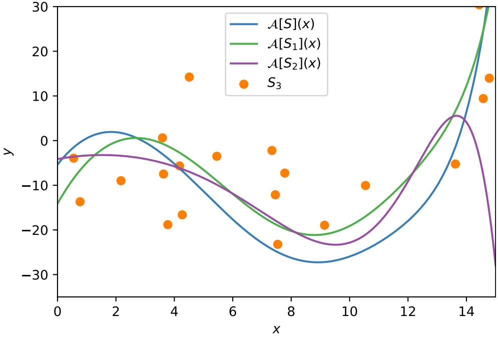
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

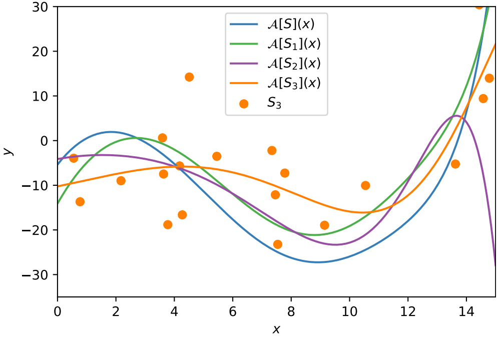
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

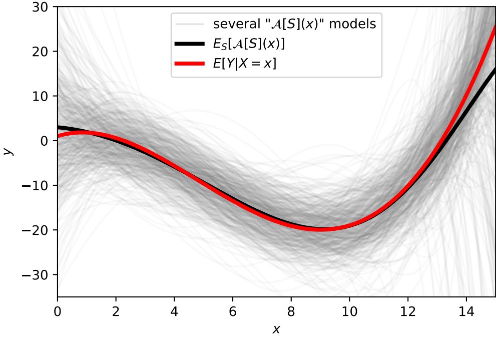
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

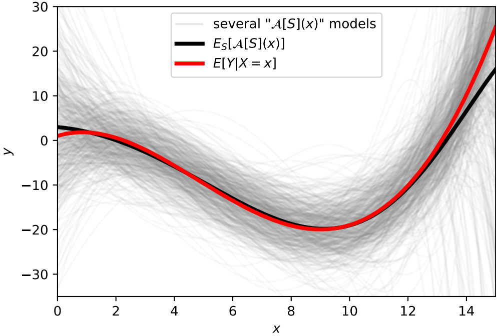
Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Introduction to Bagging

Supervised Machine Learning

Use a training set S drawn from an unknown law (X, Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



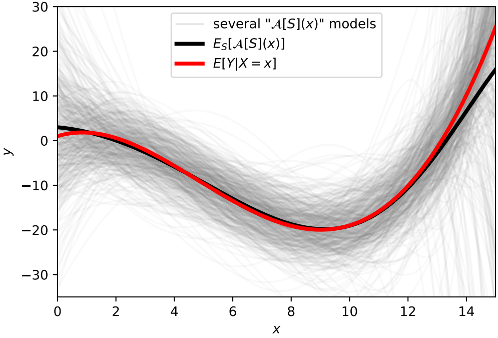
Remember bias-variance decomposition at a given point x_0:

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}\left[(Y-\mathcal{A}[S](x_0))^2\right]\right] = \mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[\mathcal{A}[S](x_0)])^2] + \mathrm{Var}_S\left(\mathcal{A}[S](x_0)\right)$$

Introduction to Bagging

Supervised Machine Learning

Use a training set S drawn from an unknown law (X,Y) and a learning algorithm $\mathcal{A}$ to build a model $\mathcal{A}[S]$ that predicts y from x



Remember bias-variance decomposition at a given point x_0:

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}\left[(Y-\mathcal{A}[S](x_0))^2\right]\right]=\mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[\mathcal{A}[S](x_0)])^2]+\mathrm{Var}_S\left(\mathcal{A}[S](x_0)\right)$$

$$\Rightarrow \mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}\left[(Y-\mathcal{A}[S](x_0))^2\right]\right]\geq \mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[\mathcal{A}[S](x_0)])^2]$$

Bagging (**B**ootstrap **Agg**regat**ing**)[Breiman, 1996]

Idea

Try to approximate $\mathbb{E}_S[\mathcal{A}[S]]$ by averaging several models trained by $\mathcal{A}$

If y is numeric: $f_S(x) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{A}[S_b](x)$

If y is a class: $f_S(x) = \text{Majority}\left(\{\mathcal{A}[S_1](x), \ldots, \mathcal{A}[S_B](x)\}\right)$

Bagging (**B**ootstrap **Agg**regat**ing**)[Breiman, 1996]

Idea

Try to approximate $\mathbb{E}_S[\mathcal{A}[S]]$ by averaging several models trained by $\mathcal{A}$

If y is numeric: $f_S(x) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}[S_b](x)$

If y is a class: $f_S(x) = \text{Majority}\left(\{\mathcal{A}[S_1](x), \ldots, \mathcal{A}[S_B](x)\}\right)$

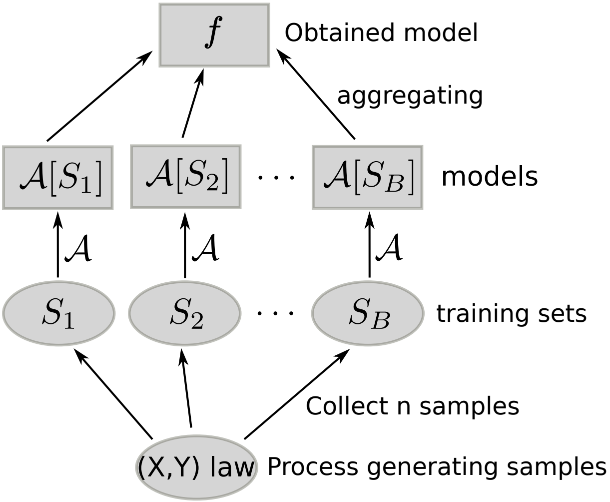
Intuition About the Choice of the Learning Algorithm $\mathcal{A}$

If we succeed then $f_S(x_0) \simeq \mathbb{E}_S[\mathcal{A}[S](x_0)]$

$\Rightarrow \mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2] \simeq \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}[S](x_0)])^2]$

$\Rightarrow$ Better choose a low bias (and high variance) learning algorithm $\mathcal{A}$

The Ideal Case: We Can Freely Draw From (X, Y) Law



Why it is not a practical solution ?

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z
To do this, we have a **fixed set** $z_1, \ldots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

Mean of Z^*:

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z
To do this, we have a **fixed set** $z_1, \dots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n} \sum_{i=1}^{n} z_i$

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z

To do this, we have a **fixed set** $z_1, \dots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n} \sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n \to +\infty]{\text{p.s.}} \mathbb{E}[Z]$

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z

To do this, we have a **fixed set** $z_1, \ldots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n}\sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n \to +\infty]{\text{p.s.}} \mathbb{E}[Z]$

CDF of Z^*:

$F_{Z^*}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{]-\infty; z_i]}(z)$

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z

To do this, we have a **fixed set** $z_1, \ldots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n}\sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n\to+\infty]{\text{p.s.}} \mathbb{E}[Z]$

CDF of Z^*:

$F_{Z^*}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{]-\infty;z_i]}(z)$

One can show that:

$\sup_z |F_{Z^*}(z) - F_Z(z)| \xrightarrow[n\to+\infty]{\text{p.s.}} 0$

Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z
To do this, we have a **fixed set** $z_1, \ldots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

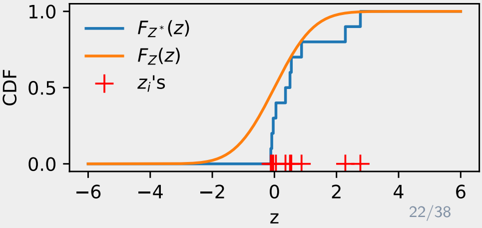
Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n}\sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n\to+\infty]{\text{p.s.}} \mathbb{E}[Z]$

CDF of Z^*:

$F_{Z^*}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{]-\infty;z_i]}(z)$

One can show that:

$\sup_z |F_{Z^*}(z) - F_Z(z)| \xrightarrow[n\to+\infty]{\text{p.s.}} 0$



Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z
To do this, we have a **fixed set** $z_1, \dots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

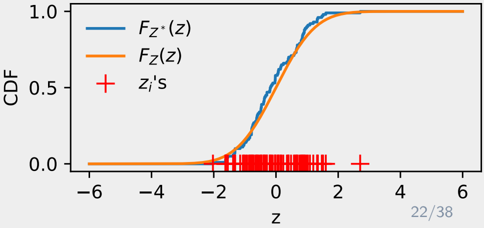
Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n}\sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n\to+\infty]{\text{p.s.}} \mathbb{E}[Z]$

CDF of Z^*:

$F_{Z^*}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{]-\infty;z_i]}(z)$

One can show that:

$\sup_z |F_{Z^*}(z) - F_Z(z)| \xrightarrow[n\to+\infty]{\text{p.s.}} 0$



Bootstrap Come to the Rescue !

Problem

We want to build a surrogate of the distribution generating Z

To do this, we have a **fixed set** $z_1, \ldots, z_n$ *i.i.d.* samples of Z.

Solution: Empirical Distribution

To generate one sample of Z^*, we draw equiprobably one index i and z_i will be the sample drawn

Properties

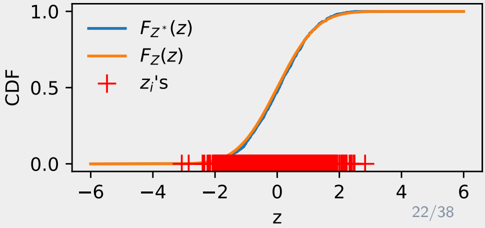
Mean of Z^*:$\mathbb{E}[Z^*] = \sum_{i=1}^{n} z_i \mathbb{P}(I = i) = \frac{1}{n}\sum_{i=1}^{n} z_i \Rightarrow \mathbb{E}[Z^*] \xrightarrow[n\to+\infty]{\text{p.s.}} \mathbb{E}[Z]$

CDF of Z^*:

$F_{Z^*}(z) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{]-\infty;z_i]}(z)$

One can show that:

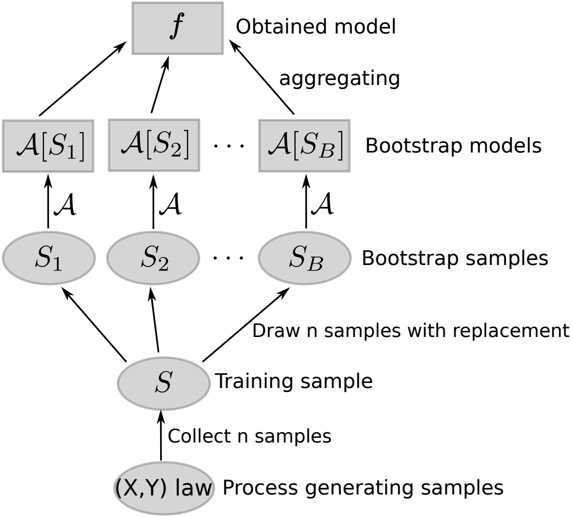
$\sup_z |F_{Z^*}(z) - F_Z(z)| \xrightarrow[n\to+\infty]{\text{p.s.}} 0$



Bagging (I)

Bootstrap Come to the Rescue !

Draw from S with a uniform probability approximates drawing from (X, Y)



Bagging (II)

Algorithm 1: Bagging

Data: Dataset $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
Result: A set of models $A[S_b]$
$\text{BaggedModels} = \{\}$;
for $b = 1$ **to** B **do**
  Draw a bootstrap set S_b of size n from the training data;
  Train the model using the bootstrap training set $\mathcal{A}[S_b]$;
  Add $\mathcal{A}[S_b]$ to BaggedModels;
end
return BaggedModels *containing* $\{\mathcal{A}[S_1], \ldots, \mathcal{A}[S_B]\}$

Prediction Using $\{\mathcal{A}[S_1], \ldots, \mathcal{A}[S_B]\}$

- For regression: $f_S(x) = \frac{1}{B} \sum_{b=1}^{B} \mathcal{A}[S_B](x)$
- For classification: $f_S(x) = \text{Majority}(\{\mathcal{A}[S_1](x)\})$

Analysis of the Average Risk of f_S

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

Analysis of the Average Risk of f_S

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

1. Bias term:
   $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}[S_1](x_0)])^2]$

Analysis of the Average Risk of f_S

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{A}[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

1. Bias term:
   $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}[S_1](x_0)])^2]$
2. Variance term:
   - Let us assume that $\mathrm{Var}_S[\mathcal{A}[S_b](x_0)] = \sigma^2$
   - And $\forall i \neq j$, $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$, then:

$$\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Analysis of the Average Risk of f_S

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

1. Bias term:
   $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}[S_1](x_0)])^2]$
2. Variance term:
   - Let us assume that $\mathrm{Var}_S[\mathcal{A}[S_b](x_0)] = \sigma^2$
   - And $\forall i \neq j$, $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$, then:

$$\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

If $S_1, \ldots, S_B$ were actual draws from (X, Y), then $\rho = 0$

Analysis of the Average Risk of f_S

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

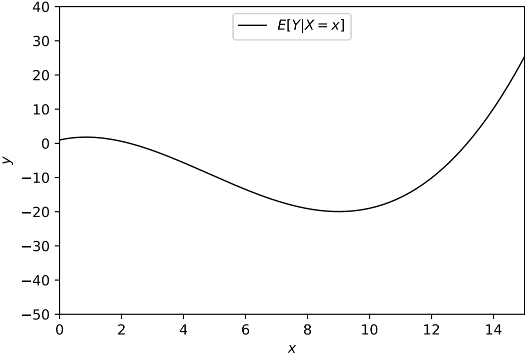
1. Bias term:
   $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}[S_1](x_0)])^2]$
2. Variance term:
   - Let us assume that $\mathsf{Var}_S[\mathcal{A}[S_b](x_0)] = \sigma^2$
   - And $\forall i \neq j$, $\mathsf{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$, then:

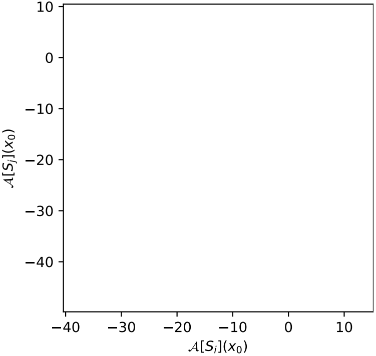
$$\mathsf{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

If $S_1, \ldots, S_B$ were actual draws from (X, Y), then $\rho = 0$
But they are boostrap sets, so is $\rho = 0$?

Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

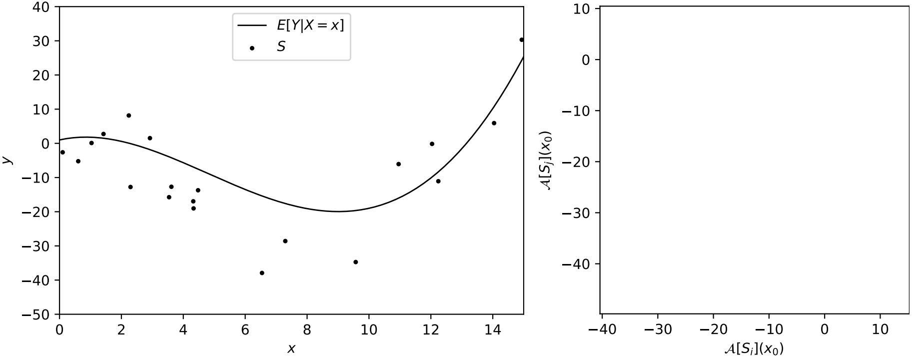
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !





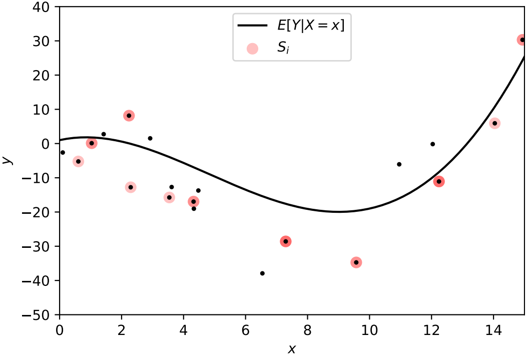
Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

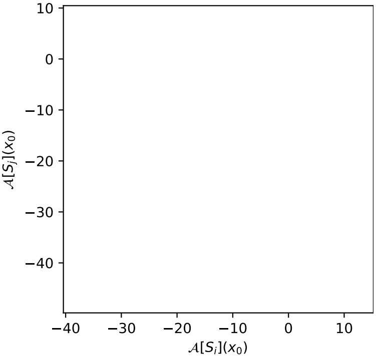
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

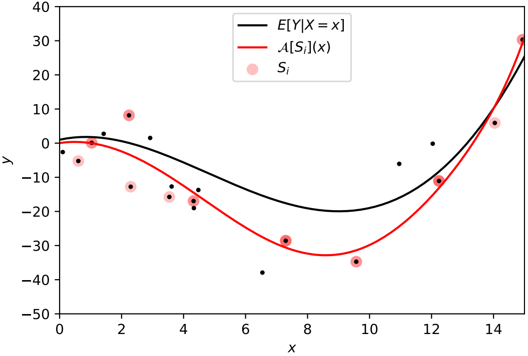
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

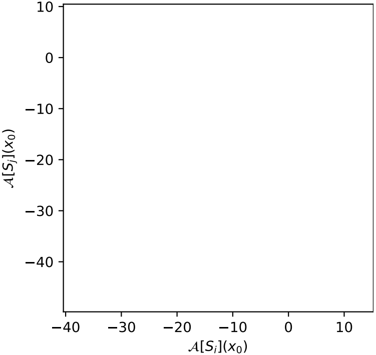




Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

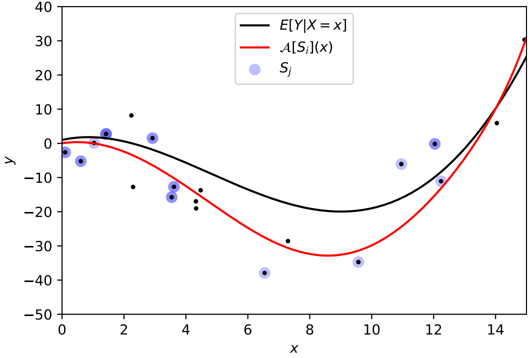
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

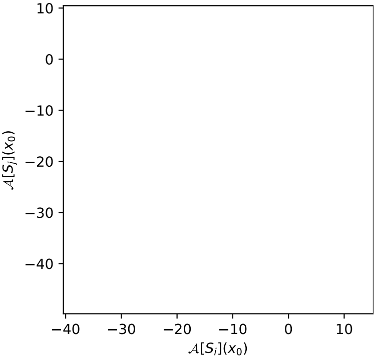




Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

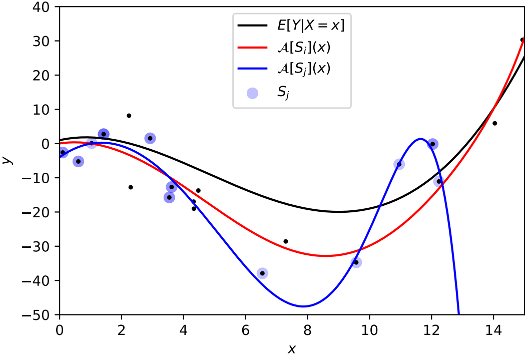
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

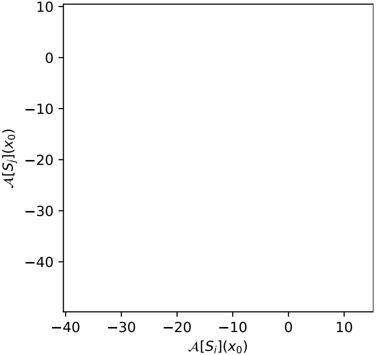




Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

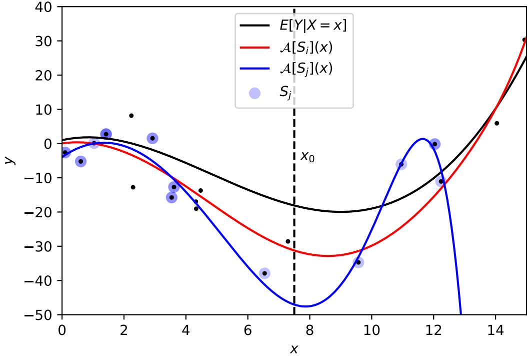
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

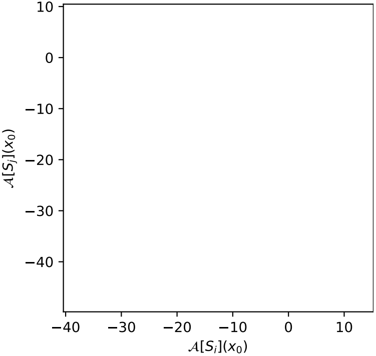




Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

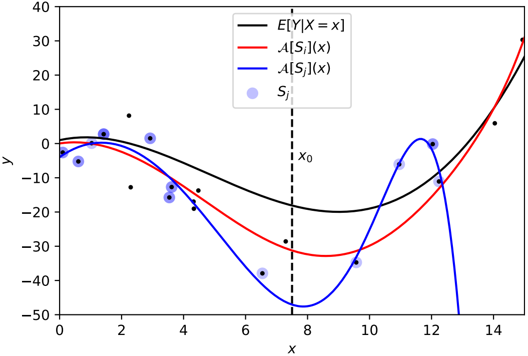
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

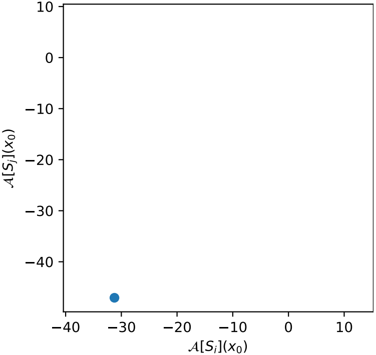




Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

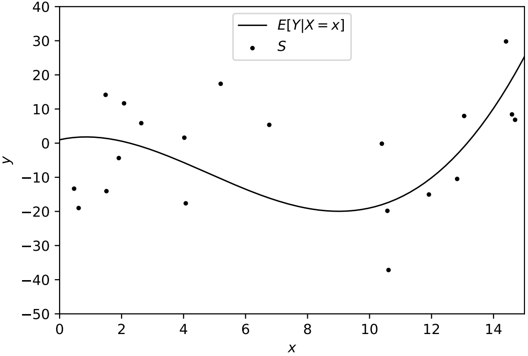
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

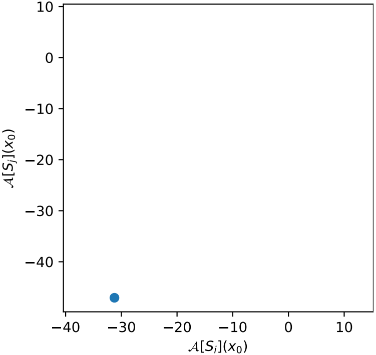




Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

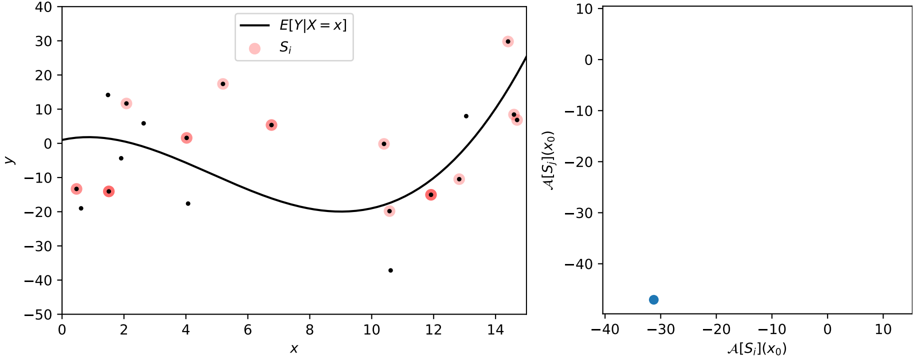
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !





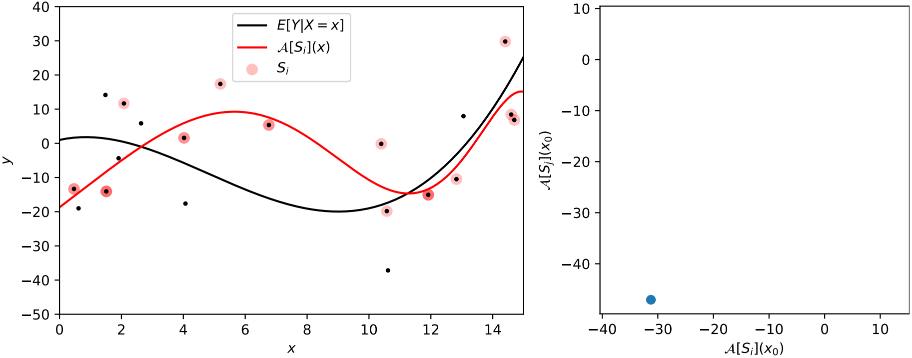
Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



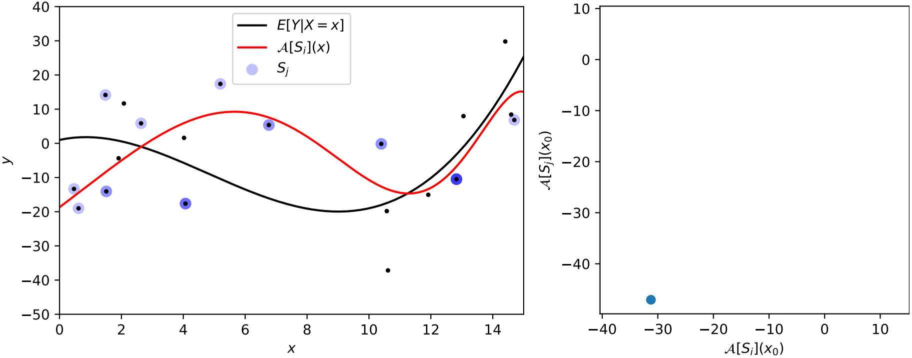
Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



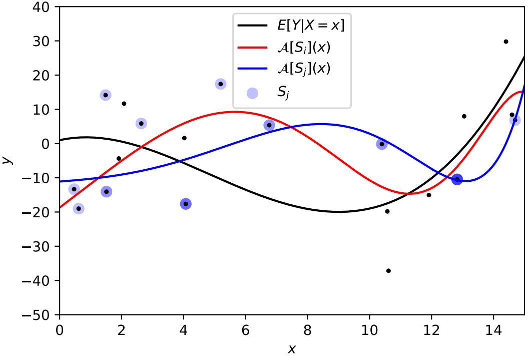
Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

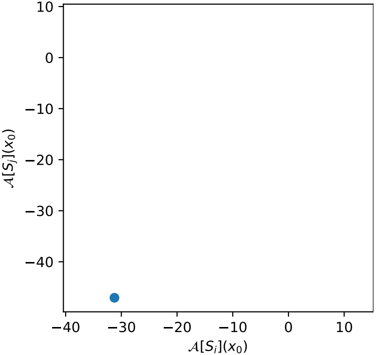
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

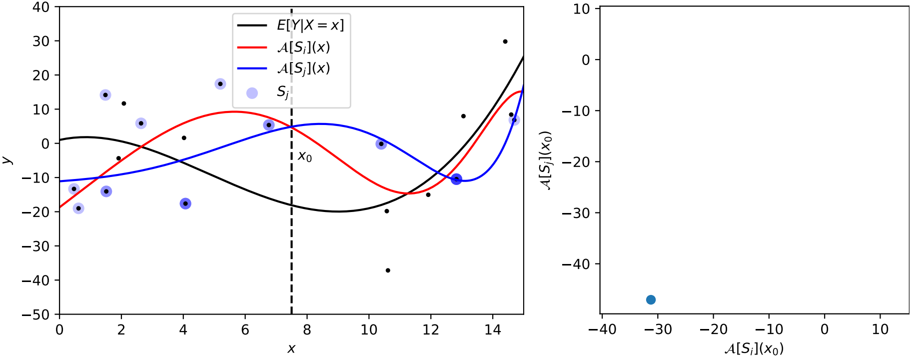
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !





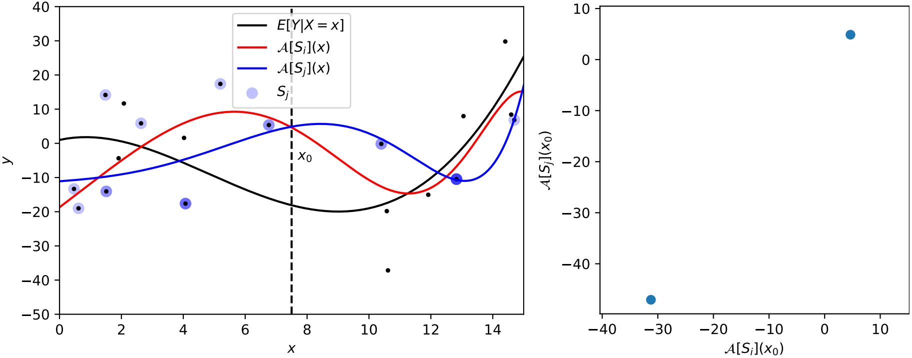
Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



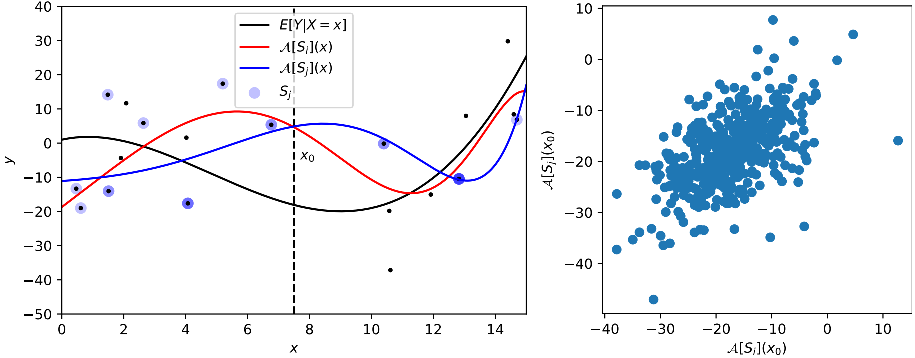
Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



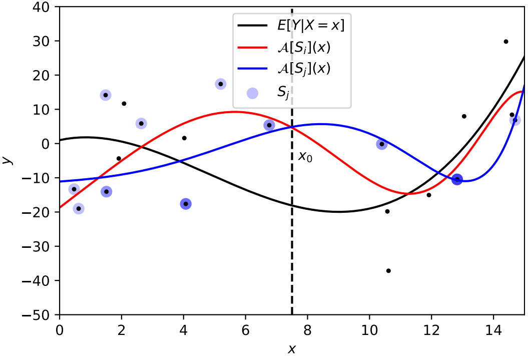
Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

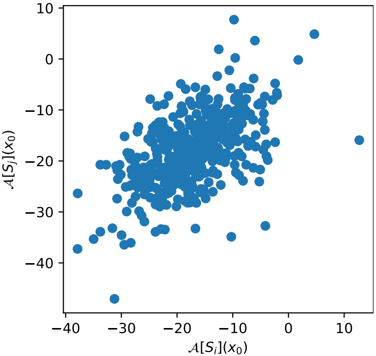
Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !



Analysis of Correlation $\mathrm{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !

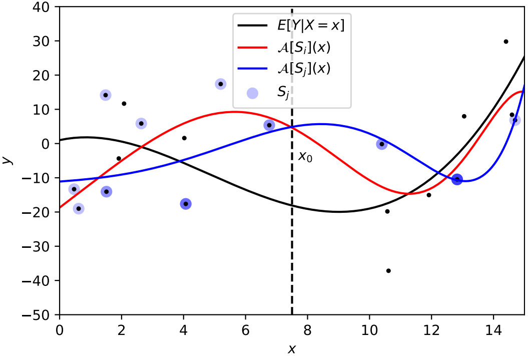


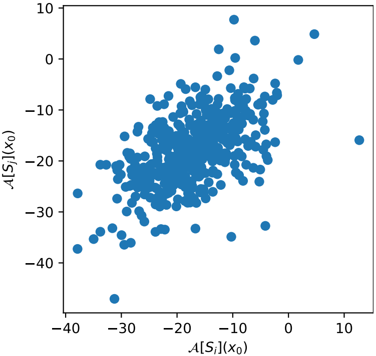


With this simulation $\rho \simeq 0.56$

Analysis of Correlation $\text{corr}_S\left(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0)\right) = \rho$

Simulate draws of $(\mathcal{A}[S_i](x_0), \mathcal{A}[S_j](x_0))$, and compute correlation !





With this simulation $\rho \simeq 0.56$

$$\text{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

As B increases, the 2nd term decreases, but the 1st term remains, and hence the correlation of pairs limits the benefits of averaging

Conclusion on Bagging

Advantage

- No hypothesis on the learning algorithm $\mathcal{A}$
- Especially useful when $\mathcal{A}$ has a low bias and large variance

Disadvantage

- Needs to compute several models
- The variance term reduction is limited by the correlation caused by bootstrap

Plan

1. Classification And Regression Trees (CART) [Breiman et al., 1984]
2. Bagging [Breiman, 1996]
3. Random Forest [Breiman, 2001]

Random Forest [Breiman, 2001]

Idea

Use Bagging with $\mathcal{A} =$ "modified CART" to reduce correlation between $\mathcal{A}[S_i](x_0)$ and $\mathcal{A}[S_j](x_0)$.
This correlation reduction is here to further reduce the variance of f_S

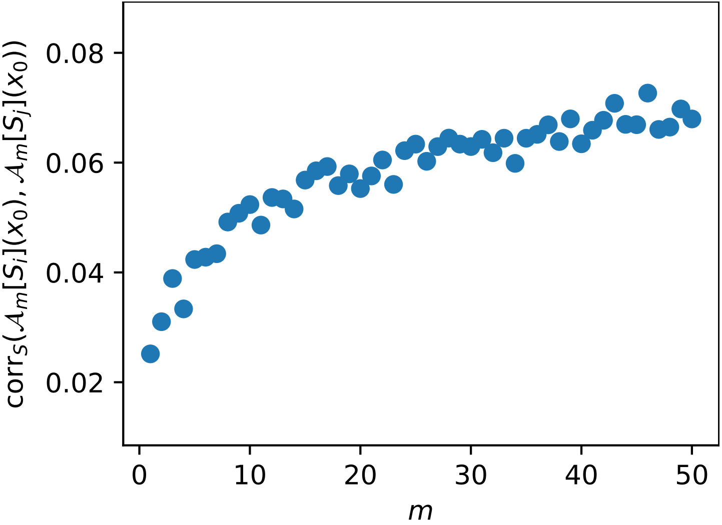
Tree Growth Modified

Data: A set of examples $\{(x_i, y_i) | \forall i \in [\![1; N]\!]\}$; $m \in [\![1, \ldots, p]\!]$
Result: Decision tree
initialize a tree as one leaf;
while *there is a splittable region* **do**
  Choose randomly a set of m variables among the p input variables;
  Using examples in the region, considering the m variables, split it: replace the leaf by the "best" node;
end

$m = p \ \rightarrow$ "vanilla" CART; $m = 1 \ \rightarrow$ random split variable j

Does it Actually Reduce the Correlation Between Trees ?

Let us perform simulations on a regression problem with 50 variables, and compute correlation for each m using draws of $(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$



Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

- Variance $\mathsf{Var}_S[f_S(x_0)]$:
  $\mathsf{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \mathrm{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \mathrm{Var}_S[\mathcal{A}_m[S_b](x_0)]$

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

- Variance $\mathrm{Var}_S[f_S(x_0)]$:
  $\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \mathrm{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \mathrm{Var}_S[\mathcal{A}_m[S_b](x_0)]$
  When $m \searrow$: $\rho \searrow$,

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathsf{Var}_S[f_S(x_0)]$$

- Variance $\mathsf{Var}_S[f_S(x_0)]$:
  $\mathsf{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \operatorname{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \operatorname{Var}_S[\mathcal{A}_m[S_b](x_0)]$
  When $m \searrow$: $\rho \searrow$, $\sigma^2 \rightsquigarrow \Rightarrow \rho\sigma^2 \searrow$

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

- Variance $\mathrm{Var}_S[f_S(x_0)]$:
  $\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \mathrm{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \mathrm{Var}_S[\mathcal{A}_m[S_b](x_0)]$
  When $m \searrow$: $\rho \searrow$, $\sigma^2 \rightsquigarrow \Rightarrow \rho\sigma^2 \searrow$
- Sq. Bias $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2]$:
  $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] =$
  $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}_m[S_b](x_0)])^2]$

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

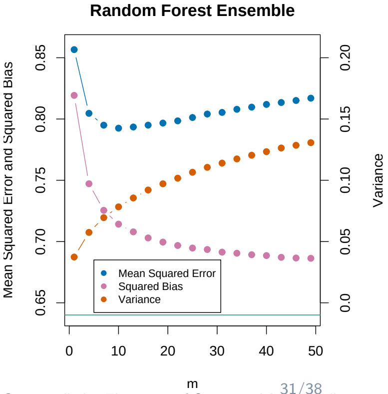
- Variance $\mathrm{Var}_S[f_S(x_0)]$:
  $\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \mathrm{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \mathrm{Var}_S[\mathcal{A}_m[S_b](x_0)]$
  When $m \searrow$: $\rho \searrow$, $\sigma^2 \rightsquigarrow \Rightarrow \rho\sigma^2 \searrow$
- Sq. Bias $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2]$:
  $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[f_S(x_0)])^2] =$
  $\mathbb{E}_{Y|X=x_0}[(Y - \mathbb{E}_S[\mathcal{A}_m[S_b](x_0)])^2]$
  When $m \searrow$: Squared Bias $\nearrow$

Does it Actually Reduce the Risk of the Random Forest ?

$$f_S(x_0) = \frac{1}{B}\sum_{b=1}^{B}\mathcal{A}_m[S_b](x_0)$$

$$\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y-f_S(x_0))^2]\right] = \mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[f_S(x_0)])^2] + \mathrm{Var}_S[f_S(x_0)]$$

- Variance $\mathrm{Var}_S[f_S(x_0)]$:
  $\mathrm{Var}_S[f_S(x_0)] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
  with
  $\rho = \mathrm{corr}_S(\mathcal{A}_m[S_i](x_0), \mathcal{A}_m[S_j](x_0))$
  and $\sigma^2 = \mathrm{Var}_S[\mathcal{A}_m[S_b](x_0)]$
  When $m \searrow$: $\rho \searrow$, $\sigma^2 \rightsquigarrow \Rightarrow \rho\sigma^2 \searrow$
- Sq. Bias $\mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[f_S(x_0)])^2]$:
  $\mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[f_S(x_0)])^2] =$
  $\mathbb{E}_{Y|X=x_0}[(Y-\mathbb{E}_S[\mathcal{A}_m[S_b](x_0)])^2]$
  When $m \searrow$: Squared Bias $\nearrow$



Source: "The Elements of Statistical Learning"

Choice for the $\mathcal{A}_m$'s hyper-parameters and B

Choice for the $\mathcal{A}_m$'s hyper-parameters

- The inventors of the algorithm make the following recommendations:
  - For classification, the default value for m is $\lfloor\sqrt{p}\rfloor$, and the minimum number examples in leaf is 1 (cf HTF),
  - For regression, the default value for $m = \lfloor p/3 \rfloor$, and the minimum number examples in leaf is 5 (cf HTF).
- In practice the best values for these parameters will depend on the problem and they should be tuned

Choice of B

- The expected risk $\mathbb{E}_S\left[\mathbb{E}_{Y|X=x_0}[(Y - f_S(x_0))^2]\right]$ decreases as $B \nearrow$
- As B increases, the computational cost increases

Be careful, Random Forest can overfit even with a large B:

- $f_S(x_0) \underset{B\to+\infty}{\longrightarrow} \mathbb{E}_{S_b|S}[\mathcal{A}_m[S_b](x_0)]$
- $\mathrm{Var}_S[\mathbb{E}_{S_b|S}[\mathcal{A}_m[S_b](x_0)]] = \rho\sigma^2 > 0$

Out-Of-Bag (OOB) Samples/Error

Out-Of-Bag (OOB) Samples

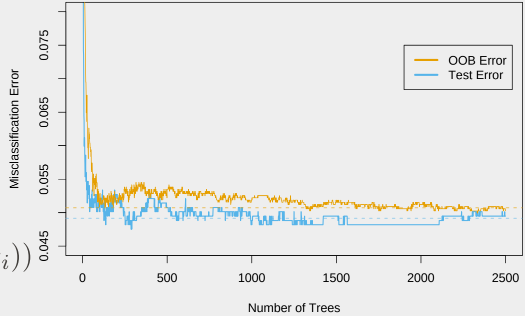
- We note $S_b^{\text{OOB}} = S \setminus S_b$, these examples were not used to train $\mathcal{A}[S_b]$
- S_b^{OOB} is called the *Out-Of-Bag (OOB) samples*
- S_b^{OOB} contains on average $(1 - \frac{1}{n})^n \underset{n \to +\infty}{\longrightarrow} \simeq 37\%$ of the samples of S

Out-Of-Bag Error

The Out-Of-Bag Error is defined as: $\text{OOB}_{\text{error}} = \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, h^{(-i)}(x_i)\right)$

Let $D_i = \{k | (x_i, y_i) \in S_k^{\text{OOB}}\}$

- For regression:
  $h^{(-i)}(x_i) = \frac{1}{|D_i|} \sum_{k \in D_i} \mathcal{A}[S_k](x_i)$
- For classification:
  $h^{(-i)}(x_i) = \underset{k \in D_i}{\text{Majority}} \left(\mathcal{A}[S_k](x_i)\right)$



Source: "The Elements of Statistical Learning" (HTF)

Variable Importance For One Tree

Remember How we Choose the Node Condition

We want to split the region R, we define:

$$R^{(l)}(j,t) = \{y_i \mid \forall i \in [\![1;N]\!] \ / \ x_i \in R \text{ and } x_{i,j} \leq t\}$$
$$R^{(r)}(j,t) = \{y_i \mid \forall i \in [\![1;N]\!] \ / \ x_i \in R \text{ and } x_{i,j} > t\}$$

We use a function H measuring the “heterogeneity” (*impurity*)
Choose j and t minimizing the “heterogeneity” inside the new regions:

$$G(R,j,t) = \frac{|R^{(l)}(j,t)|}{|R|} H\left(R^{(l)}(j,t)\right) + \frac{|R^{(r)}(j,t)|}{|R|} H\left(R^{(r)}(j,t)\right)$$

Variable Importance for One Tree [Breiman et al., 1984]

We can compute the *impurity* decrease: $\Delta H(R,j,t) = H(R) - G(R,j,t)$
For one tree T, we use:

$$\mathrm{VI}(j,T) = \sum_{u \in \mathrm{Node}(T) / \mathrm{var}(u) = j} \frac{|\mathrm{Region}(u)|}{N} \Delta H(\mathrm{Region}(u), \mathrm{var}(u), \mathrm{thresh}(u))$$

Variable Importance: Mean Decrease Impurity

Mean Decrease Impurity [Breiman, 2001]

For an ensemble of trees $T_1, \ldots, T_b$ in a Random Forest, we use:

$$\mathrm{VI}^{\mathrm{MDI}}(j) = \frac{1}{B} \sum_{b=1}^{B} \mathrm{VI}(j, T_b)$$

Advantages/Drawbacks

- Computationally cheap as it is computed along the training process
- Biased towards high cardinality features
- It quantifies the usefulness of a feature to reduce the training error, not the usefulness to make an actual prediction

Variable Importance: Permutation Importance and Mean Decrease Accuracy

Permutation Importance [Breiman, 2001]

Idea: Quantify the impact of the permutation of variable j on predictive performance

$$\text{Shuffle}(S, j) = \begin{pmatrix} x_{1,1} & \dots & x_{1,j-1} & x_{\pi(1),j} & x_{1,j+1} & \dots & x_{1,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & \dots & x_{i,j-1} & x_{\pi(i),j} & x_{i,j+1} & \dots & x_{i,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & \dots & x_{n,j-1} & x_{\pi(n),j} & x_{n,j+1} & \dots & x_{n,p} \end{pmatrix}$$

Importance of variable j in a model h is measured on a test set test_{set}:
"$\text{VI}^{\text{PI}}(h, \text{test}_{\text{set}}, j) = \text{error}(h, \text{Shuffle}(\text{test}_{\text{set}}, j)) - \text{error}(h, \text{test}_{\text{set}})$"

Variable Importance: Permutation Importance and Mean Decrease Accuracy

Permutation Importance [Breiman, 2001]

Idea: Quantify the impact of the permutation of variable j on predictive performance

$$\mathrm{Shuffle}(S, j) = \begin{pmatrix} x_{1,1} & \dots & x_{1,j-1} & x_{\pi(1),j} & x_{1,j+1} & \dots & x_{1,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & \dots & x_{i,j-1} & x_{\pi(i),j} & x_{i,j+1} & \dots & x_{i,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & \dots & x_{n,j-1} & x_{\pi(n),j} & x_{n,j+1} & \dots & x_{n,p} \end{pmatrix}$$

Importance of variable j in a model h is measured on a test set $\mathrm{test}_{\mathrm{set}}$:
"$\mathrm{VI}^{\mathrm{PI}}(h, \mathrm{test}_{\mathrm{set}}, j) = \mathrm{error}(h, \mathrm{Shuffle}(\mathrm{test}_{\mathrm{set}}, j)) - \mathrm{error}(h, \mathrm{test}_{\mathrm{set}})$"

Mean Decrease Accuracy [Breiman, 2001]

Idea: Compute the average $\mathrm{VI}^{\mathrm{PI}}$ over the trees $T_1, \dots, T_b$ in the Random Forest using Out-Of-Bag samples $S_1^{\mathrm{OOB}}, \dots, S_B^{\mathrm{OOB}}$

$\Rightarrow$ Importance of variable j: $\mathrm{VI}^{\mathrm{MDA}}(j) = \frac{1}{B}\sum_{b=1}^{B} \mathrm{VI}^{\mathrm{PI}}(T_b, S_b^{\mathrm{OOB}}, j)$

Advantages and Disadvantages of Random Forest

Advantages

- No variable scaling/normalization required
- Can handle numerical and categorical variable without pre-processing
- Can easily manage missing variable
- Relatively undisturbed by outliers (they are isolated in small nodes)
- Trees can be trained in parallel
- Easy to tune and powerful
- Can be used for feature selection

Disadvantages

- Interpretability
- Uses deep trees, if a large number of trees is used then prediction can be slow

[Breiman, 1996] Breiman, L. (1996).
Bagging predictors.
Machine learning, 24(2):123–140.

[Breiman, 2001] Breiman, L. (2001).
Random forests.
Machine learning, 45(1):5–32.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984).
Classification and regression trees.
CRC press.