Decision trees

Can we collect data to automatically create a decision tree, without domain experts?

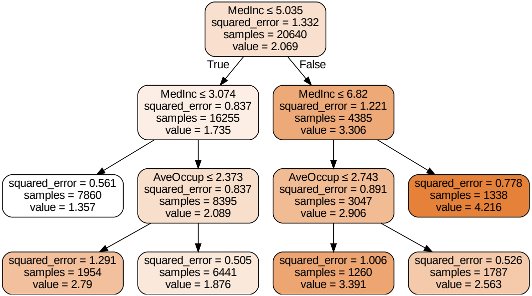
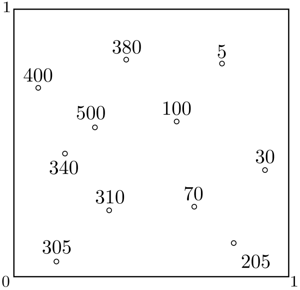


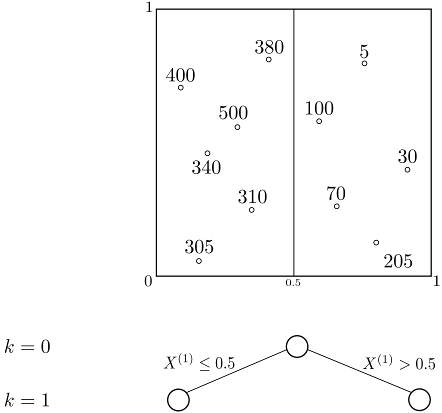
Figure: Output of a decision tree trained on a real-estate data set (1990 California housing data set).

Construction of Decision trees - regression

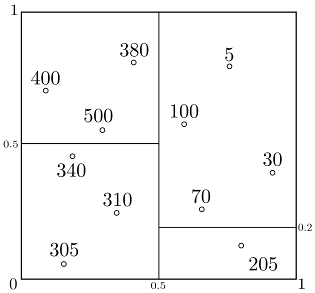


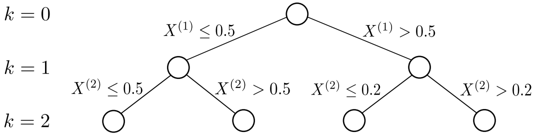
$k = 0$

Construction of Decision trees - regression

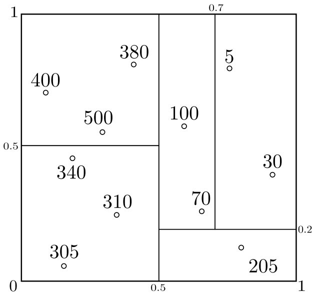


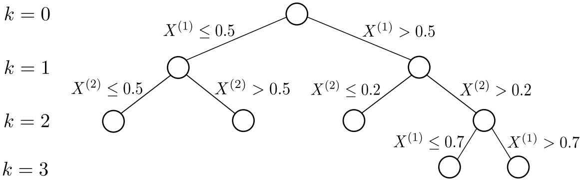
Construction of Decision trees - regression



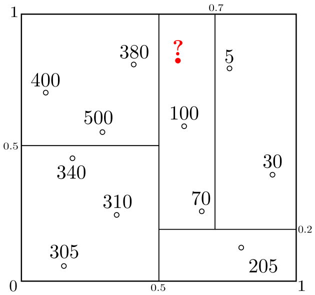


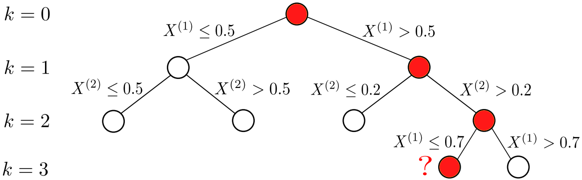
Construction of Decision trees - regression



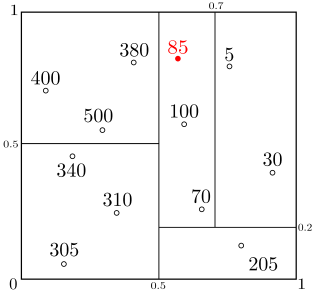


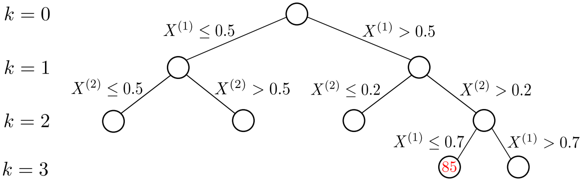
Construction of Decision trees - regression



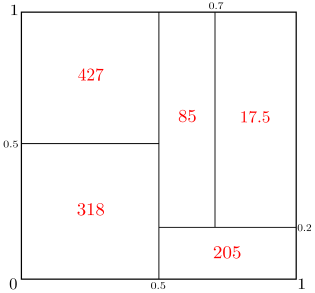


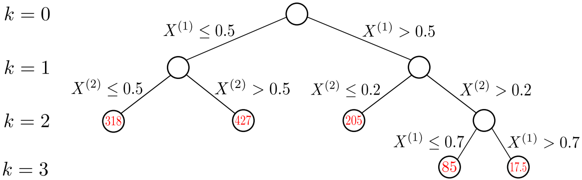
Construction of Decision trees - regression



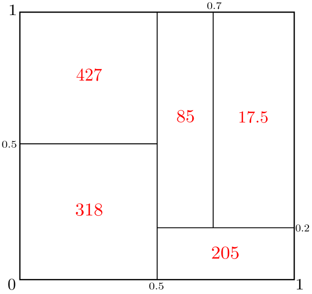


Construction of Decision trees - regression





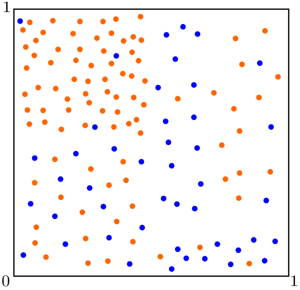
Construction of Decision trees - regression



Decision tree building

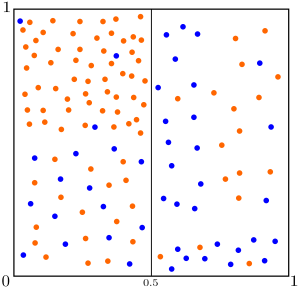
- Requires a splitting rule
- Requires a stopping rule
- Requires a prediction rule
  $\rightarrow$ Average per leaf

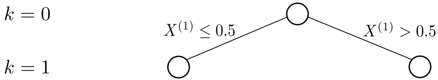
Construction of Decision trees - classification



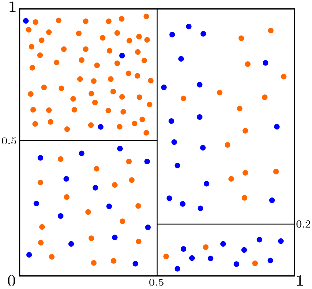
$k = 0$

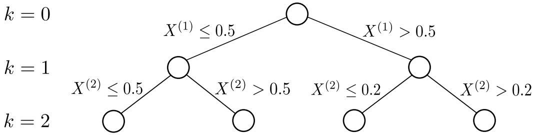
Construction of Decision trees - classification



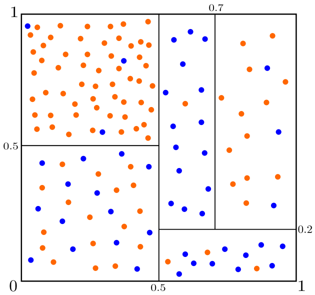


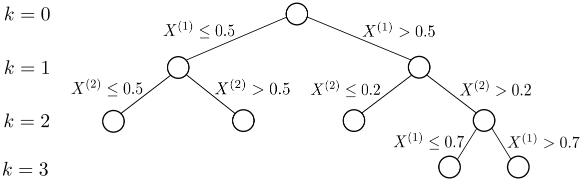
Construction of Decision trees - classification



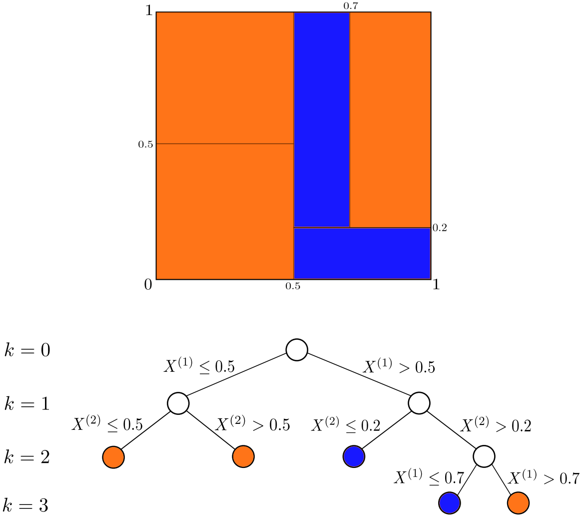


Construction of Decision trees - classification

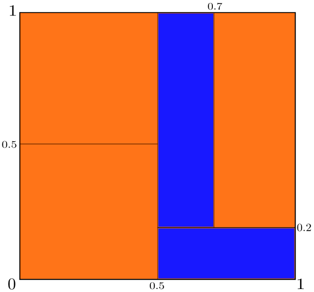




Construction of Decision trees - classification



Construction of Decision trees - classification



Decision tree building

- Requires a splitting rule
- Requires a stopping rule
- Requires a prediction rule
  $\rightarrow$ Majority vote per leaf

Outline

Outline

Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j, s; A) \\ =& Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

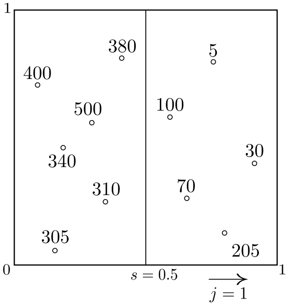
where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).

Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j = 1, s = 0.5; A) \\ =&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).

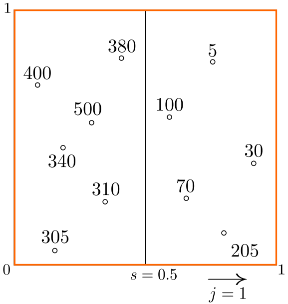


Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j = 1, s = 0.5; A) \\ =&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).

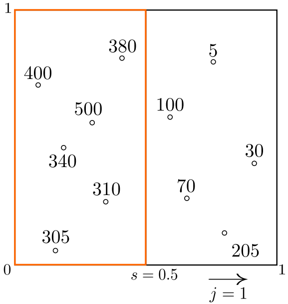


Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j = 1, s = 0.5; A) \\ =&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).

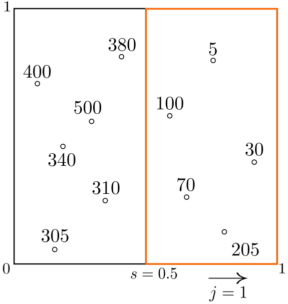


Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j = 1, s = 0.5; A) \\ =&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).



Splitting criterion

Finding the best split in a cell A requires an impurity criterion Imp. Based on this criterion, one can define the impurity reduction associated to a split (j, s) as

$$\begin{aligned} &\Delta Imp(j, s; A) \\ =& Imp(A) - p_L Imp(A_L) - p_R Imp(A_R), \end{aligned} \tag{1}$$

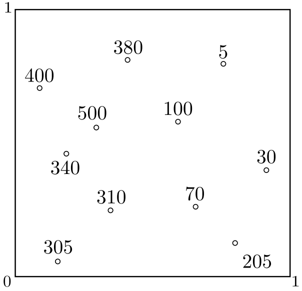
where p_L (resp. p_R) is the fraction of observations in A that fall into A_L (resp. A_R).

The best split $(j^\star, s^\star)$ is then chosen as

$$(j^\star, s^\star) \in \underset{j,s}{\operatorname{argmax}} \Delta Imp(j, s; A). \tag{2}$$

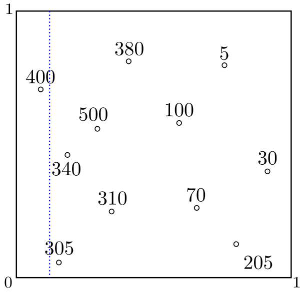
An instance of $Imp(A)$ in regression: the empirical variance of the Y_is in A.

Finding the best split - an example



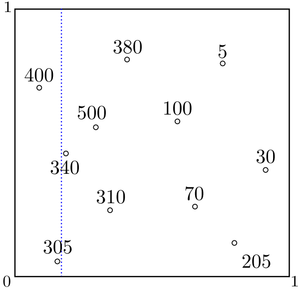
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



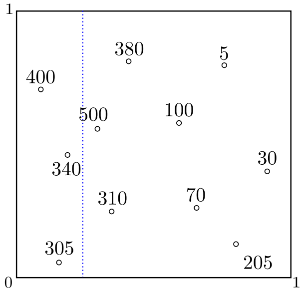
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



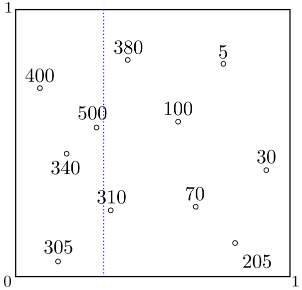
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



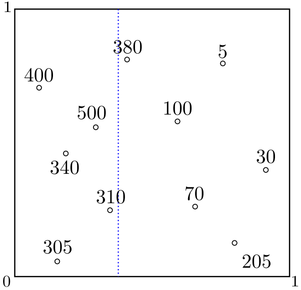
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



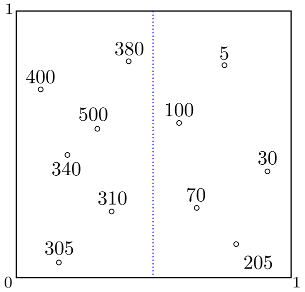
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



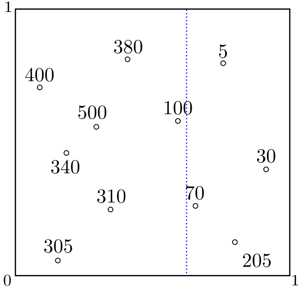
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



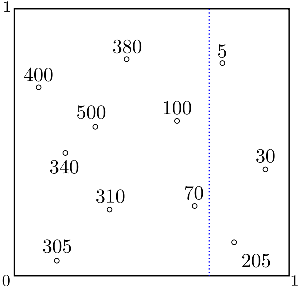
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



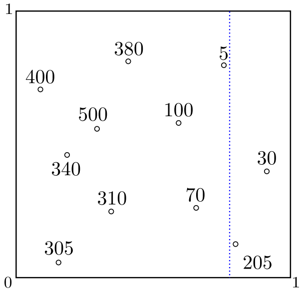
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



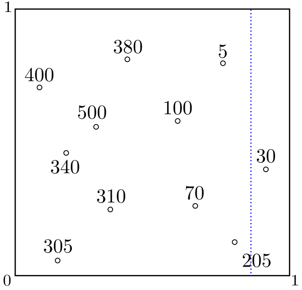
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



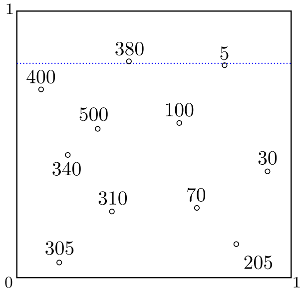
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



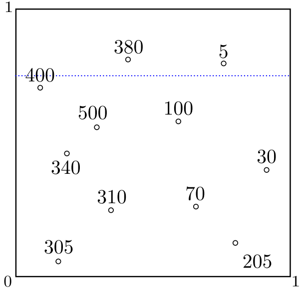
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



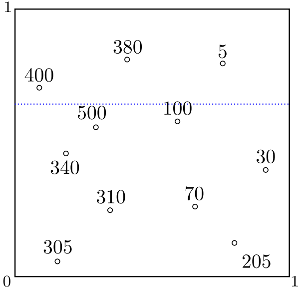
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



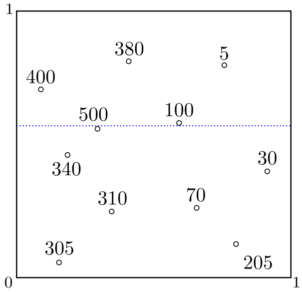
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



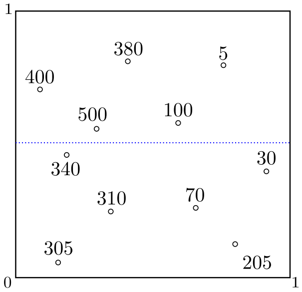
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



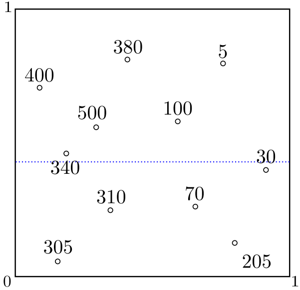
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



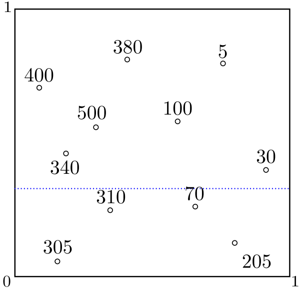
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



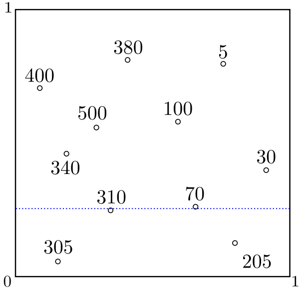
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



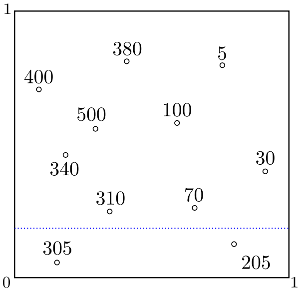
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



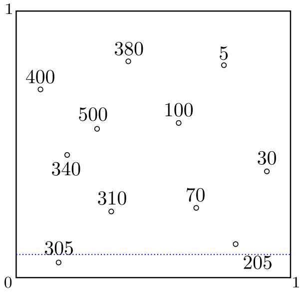
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



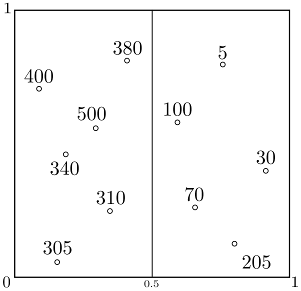
- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.

Finding the best split - an example



- Consider splits at the middle of two consecutive observations
- For each split, compute the decrease in impurity between the parent node and the two resulting nodes.
- Select the split maximizing the decrease in impurity

Impurity criteria

For **regression**, letting $N_n(A)$ the number of observations in the cell A and $\bar{Y}_A$ the mean of the Y_is in A:

- Variance
$$Imp_V(A) = \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2, \tag{3}$$
- Mean absolute deviation around the median
$$Imp_{L_1}(A) = \frac{1}{N_n(A)} \sum_{i, X_i \in A} |Y_i - \mathrm{Med}(Y_i : X_i \in A)|. \tag{4}$$

Impurity criteria

For **classification**, letting $p_{k,n}(A)$ the proportion of observations in A such that $Y = k$:

- Misclassification error rate
$$Imp_{err}(A) = 1 - \max_{1 \leq k \leq K} p_{k,n}(A) \tag{3}$$
- Gini
$$Imp_G(A) = \sum_{k=1}^{K} p_{k,n}(A)(1 - p_{k,n}(A)). \tag{4}$$
- Entropy
$$Imp_H(A) = -\sum_{k=1}^{K} p_{k,n}(A) \log_2(p_{k,n}(A)). \tag{5}$$

Splitting criterion and risk of the method

Consider the variance as impurity measure:

$$Imp(A) = \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2. \qquad (6)$$

Splitting criterion and risk of the method

Consider the variance as impurity measure:

$$Imp(A) = \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2. \tag{6}$$

For any split (j, s) in any cell A resulting in cells A_L and A_R, the impurity reduction takes the form

$$\begin{aligned}
&\Delta Imp(j, s; A) \\
=& Imp(A) - p_L Imp(A_L) - p_R Imp(A_R) \\
=& \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2 \\
&- \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2.
\end{aligned} \tag{7}$$

Splitting criterion and risk of the method

For any split (j, s) in any cell A resulting in cells A_L and A_R, the impurity reduction takes the form

$$\begin{aligned}
&\Delta Imp(j, s; A) \\
=&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R) \\
=&\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2 \\
&- \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2.
\end{aligned} \tag{6}$$

Thus finding the best split is equivalent to minimizing

$$\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2 \tag{7}$$

Splitting criterion and risk of the method

For any split (j, s) in any cell A resulting in cells A_L and A_R, the impurity reduction takes the form

$$\begin{aligned}
&\Delta Imp(j, s; A) \\
=&Imp(A) - p_L Imp(A_L) - p_R Imp(A_R) \\
=&\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_A)^2 \\
&- \frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2.
\end{aligned} \tag{6}$$

Thus finding the best split is equivalent to minimizing

$$\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2 \tag{7}$$

This corresponds to the square loss of a predictor, which is piecewise constant on A_L and A_R, whose values equal the mean of Y_i's in each cell.

Splitting criterion and risk of the method

Thus finding the best split is equivalent to minimizing

$$\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2 \quad (6)$$

This corresponds to the square loss of a predictor, which is piecewise constant on A_L and A_R, whose values equal the mean of Y_i's in each cell.

Optimal partition. Finding the tree partition with the minimal quadratic risk on the training set.

- Statistically sound
- Computationally infeasible

Splitting criterion and risk of the method

Thus finding the best split is equivalent to minimizing

$$\frac{1}{N_n(A)} \sum_{i, X_i \in A} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R})^2 \quad (6)$$

This corresponds to the square loss of a predictor, which is piecewise constant on A_L and A_R, whose values equal the mean of Y_i's in each cell.

Greedy partition. At each step, finding the split with the minimal quadratic risk on the training set.

- Not the best predictive performances
- Computationally cheap

Splitting criterion and risk of the method

Thus finding the best split is equivalent to minimizing

$$\frac{1}{N_n(A)} \sum_{i, X_i \in A} \left(Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i \in A_L} - \bar{Y}_{A_R} \mathbb{1}_{X_i \in A_R}\right)^2 \quad (6)$$

This corresponds to the square loss of a predictor, which is piecewise constant on A_L and A_R, whose values equal the mean of Y_i's in each cell.

Greedy partition. At each step, finding the split with the minimal quadratic risk on the training set.

- Not the best predictive performances
- Computationally cheap

General rule. Choose the splitting criterion corresponding to the risk you want to minimize.

Splitting criterion and risk of the method

General rule. Choose the splitting criterion corresponding to the risk you want to minimize.

Regression

- The variance corresponds to the L_2 risk.
- The mean absolute deviation around the median is close to the L_1 risk

Splitting criterion and risk of the method

General rule. Choose the splitting criterion corresponding to the risk you want to minimize.

Regression

- The variance corresponds to the L_2 risk.
- The mean absolute deviation around the median is close to the L_1 risk

Classification

- The entropy impurity is related to the cross-entropy loss
- The Gini impurity is not related to any loss, as it does not correspond to a majority vote but rather a random one
- The misclassification error rate is related to $0-1$ loss, which should not be used, as detailed hereafter.

Classification - which impurity to use?

We can choose between

- Misclassification rate

$$Imp_{err}(A) = 1 - \max_{1 \le k \le K} p_{k,n}(A) \tag{6}$$

- Gini

$$Imp_G(A) = \sum_{k=1}^{K} p_{k,n}(A)(1 - p_{k,n}(A)). \tag{7}$$

- Entropy

$$Imp_H(A) = -\sum_{k=1}^{K} p_{k,n}(A) \log_2(p_{k,n}(A)). \tag{8}$$

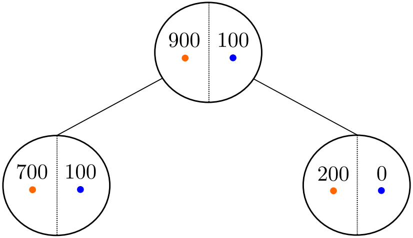
Classification - which impurity to use?

In a binary classification setting, impurities can be rewritten as

- Misclassification rate
$$Imp_{err}(A) = 1 - \max_{k \in \{0,1\}} p_{k,n}(A) \tag{6}$$
- Gini
$$Imp_G(A) = 2p_{0,n}(A)(1 - p_{0,n}(A)) \tag{7}$$
- Entropy
$$\begin{aligned} Imp_H(A) = &- p_{0,n}(A) \log_2(p_{0,n}(A)) \\ &- (1 - p_{0,n}(A)) \log_2(1 - p_{0,n}(A)) \end{aligned} \tag{8}$$

Classification - which impurity to use?

Let us take an example:

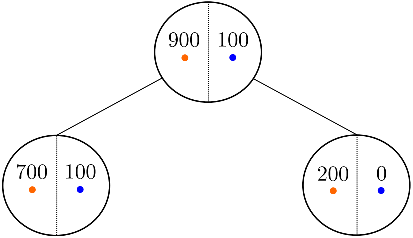


For such a split of the parent cell A, we have

$$Imp_{err}(A) = Imp_{err}(A_L) = Imp_{err}(A_R) = 0.1, \qquad (6)$$

Classification - which impurity to use?

Let us take an example:



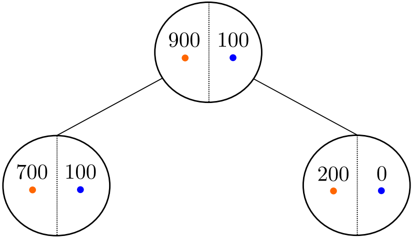
For such a split of the parent cell A, we have

$$Imp_{err}(A) = Imp_{err}(A_L) = Imp_{err}(A_R) = 0.1, \quad (6)$$

- Since $\Delta Imp_{err} = 0$, the split appears to be non-informative.

Classification - which impurity to use?

Let us take an example:

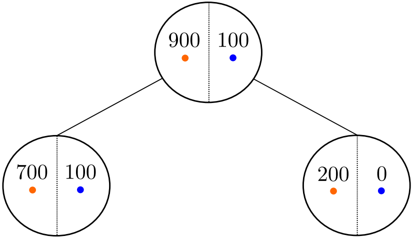


- Since $\Delta Imp_{err} = 0$, the split appears to be non-informative.
- But the right node is pure! The decrease in impurity for the two other criterion is

$$\begin{aligned} & \Delta Imp_G(A) = 0.005 \\ \text{and} \quad & \Delta Imp_H(A) = 0.01. \end{aligned} \tag{6}$$

Classification - which impurity to use?

Let us take an example:



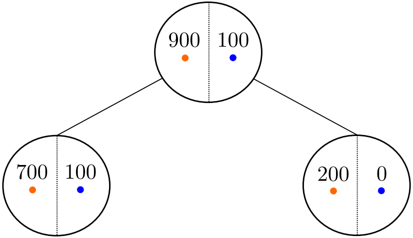
- Since $\Delta Imp_{err} = 0$, the split appears to be non-informative.
- But the right node is pure!

This phenomenon results from the fact that the misclassification rate in the binary setting is not strictly concave, contrary to the Entrope/Gini criterion. More explanation here[a]

[a] https://tushaargvs.github.io/assets/teaching/dt-notes-2020.pdf

Classification - which impurity to use?

Let us take an example:



- Since $\Delta Imp_{err} = 0$, the split appears to be non-informative.
- But the right node is pure!

Misclassification criterion is not precise enough to be used for building trees.

Outline

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

- All samples have the same label (classification)

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

- All samples have the same label (classification)
- No reduction of the impurity criterion

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

- All samples have the same label (classification)
- No reduction of the impurity criterion
- The next split will produce cells with less than `min-samples-leaf` observations (1, by default)

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

- All samples have the same label (classification)
- No reduction of the impurity criterion
- The next split will produce cells with less than `min-samples-leaf` observations (1, by default)
- The cell contains less than `min-samples-split` observations (2, by default)

Decision tree

Now that we have defined a splitting rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule

Stopping rule for splitting a cell:

- All samples have the same label (classification)
- No reduction of the impurity criterion
- The next split will produce cells with less than `min-samples-leaf` observations (1, by default)
- The cell contains less than `min-samples-split` observations (2, by default)
- The cell has already been split `max-depth` times (∞, by default)

Decision tree

Now that we have defined a splitting and a stopping rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule (by default, one observation per leaf)

Decision tree

Now that we have defined a splitting and a stopping rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule (by default, one observation per leaf)
- Prediction rule

Prediction rule:

Decision tree

Now that we have defined a splitting and a stopping rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule (by default, one observation per leaf)
- Prediction rule

Prediction rule:

- Regression - Average of labels per leaf

$$\hat{t}_n(x) = \sum_{i=1}^{n} Y_i \frac{\mathbb{1}_{X_i \in A_n(x)}}{N_n(A_n(x))} \tag{6}$$

Decision tree

Now that we have defined a splitting and a stopping rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule (by default, one observation per leaf)
- Prediction rule

Prediction rule:

- Regression - Average of labels per leaf

$$\hat{t}_n(x) = \sum_{i=1}^{n} Y_i \frac{\mathbb{1}_{X_i \in A_n(x)}}{N_n(A_n(x))} \tag{6}$$

- Classification - Majority vote per leaf

$$\hat{t}_n(x) = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \sum_{i=1}^{n} \frac{\mathbb{1}_{Y_i = k} \mathbb{1}_{X_i \in A_n(x)}}{N_n(A_n(x))} \tag{7}$$

Decision tree

Now that we have defined a splitting and a stopping rule, let us see the rest of the tree construction.

Decision tree building

- Splitting rule (Variance in regression, Gini or Entropy in classification)
- Stopping rule (by default, one observation per leaf)
- Prediction rule (average or majority vote per leaf)

Outline

Handling different types of features

There exist three main types of features:

- Continuous (blood pressure)
- Ordinal (Glasgow score)
- Nominal (Medical treatments)

Handling different types of features

There exist three main types of features:

- Continuous (blood pressure)
- Ordinal (Glasgow score)
- Nominal (Medical treatments)

Continuous features. The tree above was built on continuous features: splits are of the form $X^{(j)} \leq s$.

Handling different types of features

There exist three main types of features:

- Continuous (blood pressure)
- Ordinal (Glasgow score)
- Nominal (Medical treatments)

Continuous features. The tree above was built on continuous features: splits are of the form $X^{(j)} \leq s$.

Ordinal features. Construction can be directly extended to ordinal features: splits are exactly of the same form $X^{(j)} \leq s$.

Handling different types of features

There exist three main types of features:

- Continuous (blood pressure)
- Ordinal (Glasgow score)
- Nominal (Medical treatments)

Continuous features. The tree above was built on continuous features: splits are of the form $X^{(j)} \leq s$.

Ordinal features. Construction can be directly extended to ordinal features: splits are exactly of the same form $X^{(j)} \leq s$.

Nominal features. For nominal feature, it makes no sense to consider such splits: there is no natural order on treatments.

Nominal features

A **nominal features** $X^{(j)}$ can take different discrete values that are not ordered. For example, $X^{(j)}$ can be the type of treatment, which is surgical, chemical, or nothing (three different modalities).

Nominal features

A **nominal features** $X^{(j)}$ can take different discrete values that are not ordered. For example, $X^{(j)}$ can be the type of treatment, which is surgical, chemical, or nothing (three different modalities).

Exhaustive search Letting $\mathcal{C}$ the set of all modalities of a variable, any split along this variable is of the form C versus C^c for any $C \subset \mathcal{C}$.

- All partitions of modalities in two groups is admissible
- Computationally costly / infeasible to evaluate all these splits for variables with high cardinality (number of modalities)

Nominal features

A **nominal features** $X^{(j)}$ can take different discrete values that are not ordered. For example, $X^{(j)}$ can be the type of treatment, which is surgical, chemical, or nothing (three different modalities).

Common practice - One-hot encoding Creating as many new (dummy) variables as modalities. In our example, our treatment variable would become

$$(1,0,0) \text{ for surgical treatments, } (0,1,0) \text{ for chemical treatments,}$$
$$(0,0,1) \text{ for no treatment}$$

- A split is the of the type "one modality" VS "all other modalities".
- More limited number of splits, computationally appealing but decreases the model predictivity.

Nominal features

A **nominal features** $X^{(j)}$ can take different discrete values that are not ordered. For example, $X^{(j)}$ can be the type of treatment, which is surgical, chemical, or nothing (three different modalities).

Common practice - One-hot encoding Creating as many new (dummy) variables as modalities. In our example, our treatment variable would become

$(1, 0, 0)$ for surgical treatments, $(0, 1, 0)$ for chemical treatments,
$(0, 0, 1)$ for no treatment

- A split is the of the type "one modality" VS "all other modalities".
- More limited number of splits, computationally appealing but decreases the model predictivity.

One-hot encoding is the most common encoding method.

A clever encoding: Target encoding

Nominal variables. A classic way to handle them is via one-hot encoding. Sadly, it limits the predictivity of the model.

In **binary classification**, we can do better.

A clever encoding: Target encoding

Nominal variables. A classic way to handle them is via one-hot encoding. Sadly, it limits the predictivity of the model.

In **binary classification**, we can do better.

- Choose an impurity (misclassification rate, entropy or Gini)

A clever encoding: Target encoding

Nominal variables. A classic way to handle them is via one-hot encoding. Sadly, it limits the predictivity of the model.

In **binary classification**, we can do better.

- Choose an impurity (misclassification rate, entropy or Gini)
- Consider a nominal variable X_j that can take L modalities. Reorder it so that the empirical probabilities in a given cell A satisfy

$$\begin{aligned} & \mathbb{P}_n[Y = 1 | X_j = C_1, X \in A] \\ \leq & \mathbb{P}_n[Y = 1 | X_j = C_2, X \in A] \\ \leq & \ \dots \\ \leq & \mathbb{P}_n[Y = 1 | X_j = C_L, X \in A]. \end{aligned} \tag{6}$$

A clever encoding: Target encoding

Nominal variables. A classic way to handle them is via one-hot encoding. Sadly, it limits the predictivity of the model.

In **binary classification**, we can do better.

- Choose an impurity (misclassification rate, entropy or Gini)
- Consider a nominal variable X_j that can take L modalities. Reorder it so that the empirical probabilities in a given cell A satisfy
$$\begin{aligned} & \mathbb{P}_n[Y=1|X_j=C_1, X\in A] \\ \leq & \mathbb{P}_n[Y=1|X_j=C_2, X\in A] \\ \leq & \dots \\ \leq & \mathbb{P}_n[Y=1|X_j=C_L, X\in A]. \end{aligned} \tag{6}$$
- Then the best split (that maximizes the decrease in impurity) is of the form
$$X_j \in \{C_1, \dots, C_\ell\} \quad \text{vs} \quad X_j \in \{C_{\ell+1}, \dots, C_L\}. \tag{7}$$

This is a result from Fisher 1958

A clever encoding: Target encoding

- Choose an impurity (misclassification rate, entropy or Gini)
- Consider a nominal variable X_j that can take L modalities. Reorder it so that the empirical probabilities in a given cell A satisfy

$$\begin{aligned} &\mathbb{P}_n[Y = 1 | X_j = C_1, X \in A] \\ \leq &\mathbb{P}_n[Y = 1 | X_j = C_2, X \in A] \\ \leq &\ \dots \\ \leq &\mathbb{P}_n[Y = 1 | X_j = C_L, X \in A]. \end{aligned} \tag{6}$$

- Then the best split (that maximizes the decrease in impurity) is of the form

$$X_j \in \{C_1, \dots, C_\ell\} \quad \text{vs} \quad X_j \in \{C_{\ell+1}, \dots, C_L\}. \tag{7}$$

Summary. Finding the optimal split by reordering and then evaluating $L-1$ splits instead of $2^{L-1}-1$ splits for exhaustive search (and L splits with suboptimal decision for one-hot encoding).

A clever encoding: Target encoding

- Choose an impurity (misclassification rate, entropy or Gini)
- Consider a nominal variable X_j that can take L modalities. Reorder it so that the empirical probabilities in a given cell A satisfy
$$\begin{aligned} &\mathbb{P}_n[Y = 1 | X_j = C_1, X \in A] \\ \leq &\mathbb{P}_n[Y = 1 | X_j = C_2, X \in A] \\ \leq &\ \dots \\ \leq &\mathbb{P}_n[Y = 1 | X_j = C_L, X \in A]. \end{aligned} \tag{6}$$
- Then the best split (that maximizes the decrease in impurity) is of the form
$$X_j \in \{C_1, \dots, C_\ell\} \quad \text{vs} \quad X_j \in \{C_{\ell+1}, \dots, C_L\}. \tag{7}$$

Extension to regression. The same procedure holds in regression by considering the average values of Y for each modality (instead of the probabilities).

Outline

Generalization / Overfitting

By default, a tree is fully grown, i.e., there is only one observation per leaf.

What is the training error of such trees?

Generalization / Overfitting

By default, a tree is fully grown, i.e., there is only one observation per leaf.

What is the training error of such trees?

The training error is exactly zero. The tree makes perfect prediction on the training set!

Generalization / Overfitting

By default, a tree is fully grown, i.e., there is only one observation per leaf.

What is the training error of such trees?

The training error is exactly zero. The tree makes perfect prediction on the training set!

Overfitting. Unfortunately, it is unlikely to have the same level of performances on new data. If the test error is very large compared to the training error, we say that our method **overfits** the data.

Generalization / Overfitting

Overfitting. Unfortunately, it is unlikely to have the same level of performances on new data. If the test error is very large compared to the training error, we say that our method **overfits** the data.

Fighting overfitting. To prevent this phenomenon from happening, we can limit the complexity of the method. In decision trees, this means:

- setting parameters to limit the depth of the tree (`min-samples-leaf`, `min-samples-split`, `max-depth`)
- using pruning strategies, that is building a fully-grown tree and remove/prune some branches of the tree to obtain a simpler tree that generalizes better.

Generalization / Overfitting

Overfitting. Unfortunately, it is unlikely to have the same level of performances on new data. If the test error is very large compared to the training error, we say that our method **overfits** the data.

Fighting overfitting. To prevent this phenomenon from happening, we can limit the complexity of the method. In decision trees, this means:

- setting parameters to limit the depth of the tree (`min-samples-leaf`, `min-samples-split`, `max-depth`)
- using pruning strategies, that is building a fully-grown tree and remove/prune some branches of the tree to obtain a simpler tree that generalizes better.

Pruning strategies are always/often preferred!

Fighting overfitting. To prevent this phenomenon from happening, we can limit the complexity of the method. In decision trees, this means:

- setting parameters to limit the depth of the tree (`min-samples-leaf`, `min-samples-split`, `max-depth`)
- using pruning strategies, that is building a fully-grown tree and remove/prune some branches of the tree to obtain a simpler tree that generalizes better.

Pruning strategies are always/often preferred!

$\rightarrow$ Stopping the tree construction when the splitting criterion is low is not a valid strategy.

Pruning strategies

Two types of pruning strategies exist:

- Reducing Error, consists in removing branches of the fully-grown tree, based on the error computed on an extra data set (validation set). Simple but implies that less data are used for the training of the tree (first step).
- Cost-complexity pruning (CART) is based on a penalization of the decision tree error via the number of leaves.

Pruning strategies

Cost-complexity pruning. Let T_0 be the trained fully-grown tree. We denote by $R(T)$ the risk of any tree T, defined as either the misclassification rate (1 - accuracy) or the weighted impurity of each one of its leaves:

$$R(T) = \sum_{A \in \mathrm{Leaf}(T)} p_A Imp(A), \tag{8}$$

where p_A is the proportion of observations falling into A (usually $1/n$).

Cost-complexity pruning. Let T_0 be the trained fully-grown tree. We denote by $R(T)$ the risk of any tree T, defined as either the misclassification rate (1 - accuracy) or the weighted impurity of each one of its leaves:

$$R(T) = \sum_{A \in \mathrm{Leaf}(T)} p_A Imp(A), \tag{8}$$

where p_A is the proportion of observations falling into A (usually $1/n$).

As mentioned before, for a fully-grown tree T_0, $R(T_0) = 0$ and then does not give a good measure of predictive performances of T_0.

Pruning strategies

Cost-complexity pruning. Let T_0 be the trained fully-grown tree. We denote by $R(T)$ the risk of any tree T, defined as either the misclassification rate (1 - accuracy) or the weighted impurity of each one of its leaves:

$$R(T) = \sum_{A \in \mathrm{Leaf}(T)} p_A Imp(A), \tag{8}$$

where p_A is the proportion of observations falling into A (usually $1/n$).

For all $\alpha > 0$, we define the cost-complexity measure $R_\alpha(T)$ as

$$R_\alpha(T) = R(T) + \alpha|\mathrm{Leaf}(T)|, \tag{9}$$

where $|\mathrm{Leaf}(T)|$ is the number of leaves in T.

A cross-validation procedure can then be used to select the best value for α, therefore producing an shallower tree than T_0.

Outline

Final algorithm

Tree construction

- Input: a dataset, an impurity measure.
- At each node A, select the best split via
$$(j^\star, s^\star) \in \underset{j\in\{1,\dots,d\}, s\in \mathrm{range}(X^{(j)})}{\text{argmax}} \Delta Imp(j, s; A).$$
- Repeat for each cell until the leaf contains one observation
- Output: a fully-grown decision tree.

Final algorithm

Tree construction

- Input: a dataset, an impurity measure.
- At each node A, select the best split via
$$(j^\star, s^\star) \in \underset{j\in\{1,\dots,d\}, s\in\mathrm{range}(X^{(j)})}{\mathrm{argmax}} \Delta Imp(j, s; A).$$
- Repeat for each cell until the leaf contains one observation
- Output: a fully-grown decision tree.

Tree pruning

- Input: A fully-grown decision tree, a data set, an impurity measure.
- Choose one of the two pruning strategies:
  - Reduction Error pruning (RE, C4.5)
  - Cost complexity pruning (CART)
- Output: a pruned decision tree.

Final algorithm

Tree construction

- Input: a dataset, an impurity measure.
- At each node A, select the best split via
  $$(j^\star, s^\star) \in \underset{j\in\{1,\ldots,d\}, s\in \mathrm{range}(X^{(j)})}{\operatorname{argmax}} \Delta Imp(j, s; A).$$
- Repeat for each cell until the leaf contains one observation
- Output: a fully-grown decision tree.

Tree pruning

- Input: A fully-grown decision tree, a data set, an impurity measure.
- Choose one of the two pruning strategies:
  - Reduction Error pruning (RE, C4.5)
  - Cost complexity pruning (CART)
- Output: a pruned decision tree.

Tree prediction Tthe tree prediction at x_{new} is given by the average / majority vote among the training observations falling into the same leaf as x_{new}.

Pro/cons

Benefits

- Work in classification and regression
- Can handle categorical and continuous features
- Interpretable
- Invariant by monotonic transformation of the data
- Missing values
- Numerical complexity : $nd \log n$
- Feature selection / good in high-dimensional settings

Pro/cons

Benefits

- Work in classification and regression
- Can handle categorical and continuous features
- Interpretable
- Invariant by monotonic transformation of the data
- Missing values
- Numerical complexity : $nd \log n$
- Feature selection / good in high-dimensional settings

Drawbacks

- Non-robust to small changes in data
- Limited approximation capacity (thresholded nature)

[Fis58] Walter D Fisher. “On grouping for maximum homogeneity”. In: *Journal of the American statistical Association* 53.284 (1958), pp. 789–798.