Chapter 6: Decision Trees

Dr. Xudong Liu
Assistant Professor
School of Computing
University of North Florida

Monday, 2/7/2022

(Some from R&N's AI: a Modern Approach)

Notations

- $A = \{a_1, \ldots, a_d\}$: a set of **attributes**, where each attribute can be real or categorical.
- $\mathcal{X}$: the d dimensional space given A of all **examples** (or samples, instances) $\boldsymbol{x_i} = (x_{i1}, \ldots, x_{id})$.
  - $\boldsymbol{x_i}$ essentially is a vector in a d dimensional space.
- $\mathcal{Y}$: a space of **labels** y_i that is a scalar (or numerical) value.
  - $\mathcal{Y} = \{Yes, No\}$ for binary classification problems
  - $\mathcal{Y} = \{\text{all reals between 0 and 1}\}$ for regression problems
- $D = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_m}, y_m)\}$: a dataset of m examples in $\mathcal{X}$ labeled by the labels in $\mathcal{Y}$.

Restaurant Data

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

Figure: Situations where I will/won't wait for a table

A Decision Tree for the Restaurant Data

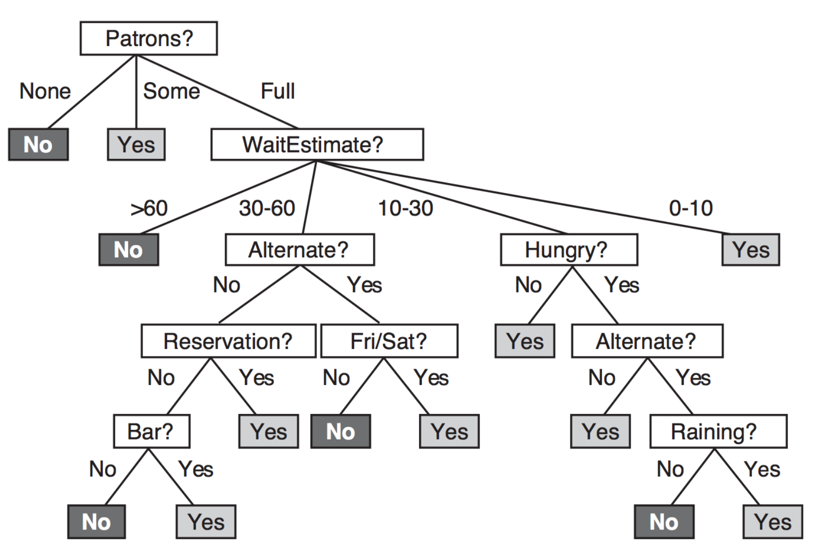


Figure: A decision tree that can predict whether wait or not

Decision Tree Learning

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

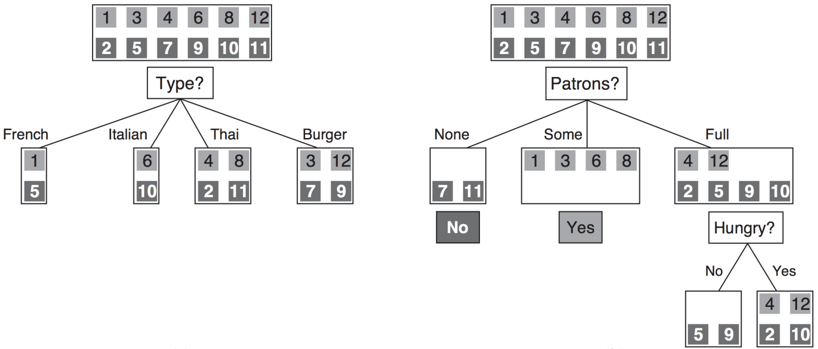
  if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
  else if all *examples* have the same classification **then return** the classification
  else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
  else
    $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
    tree ← a new decision tree with root test A
    for each value v_k of A **do**
      $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
      subtree ← DECISION-TREE-LEARNING(*exs*, *attributes* − A, *examples*)
      add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
    return *tree*

Figure: A decision tree learning template algorithm

- Goal: learn/build a decision tree using the attributes and examples
- Key: recursively choosing “most important” attribute as root of (sub)tree
  - Options: information gain (Quinlan’s ID3), gain ratio (Quinlan’s C4.5), Gini index (Breiman et al.’s CART)

Information Gain

- Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative.”
- Shown below, Patrons is better than Type at the root.



(a) (b)

Information Gain

- Information entropy: a way to measure certainty or purity.

$$Ent(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \cdot \log_2 p_k,$$

  where p_k is the percent of examples in D that are labeled by y_k.
- We convention that, if $p = 0$, $p \cdot \log_2 p = 0$.
- $Ent(D) \in [0, \log_2 |\mathcal{Y}|]$
  - $Ent(D) = 0$, when all examples in D are labeled by the same label.
  - $Ent(D) = \log_2 |\mathcal{Y}|$, when $p_k = \frac{1}{k}$ for all $y_k \in \mathcal{Y}$.
  - The bigger the entropy, the more uncertain and the more impure.
  - The smaller the entropy, the more certain and the more pure.
- Information gain: a way to measure the increase of purity by picking an attribute.

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v),$$

  where V is the number of values in attribute a's domain $\{a^1, \ldots, a^V\}$, and D^v is the set of the examples in D that have a^v as the value of attribute a.

Information Gain

- The bigger $Gain(D, a)$, the more increment of purity by picking a.
- Therefore, the IMPORTANCE method will select the attribute a^* that maximize information gain; that is,
$$a^* = \arg\max Gain(D, a).$$
- This is the ID3 algorithm by Quinlan, 1986.
- For the previous dataset, we compute $Gain(D, Patrons)$:
$$1 - (\tfrac{2}{12} Ent(D^{Patrons=None}) + \tfrac{4}{12} Ent(D^{Patrons=Some}) + \tfrac{6}{12} Ent(D^{Patrons=Full})) = 0.541.$$
- We then compute $Gain(D, Type)$:
$$1 - (\tfrac{2}{12} Ent(D^{Type=French}) + \tfrac{2}{12} Ent(D^{Type=Italian}) + \tfrac{4}{12} Ent(D^{Type=Burger}) + \tfrac{4}{12} Ent(D^{Type=Thai})) = 0.$$
- In fact, attribute Patrons has the highest information gain; thus, Patrons is selected as the root attribute.

Gain Ratio

- Information gain often favors attributes with more values, which could lead to bad generalization.
- Algorithm C4.5 by Quinlan, 1993 proposes to use *gain ratio* instead to balance attributes with fewer values with attributes with more values:
$$Gain\_ratio(D, a) = \frac{Gain(D,a)}{IV(D,a)},$$
where $IV(D, a)$ is the *intrinsic value* that is defined:
$$IV(D, a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \cdot \log_2 \frac{|D^v|}{|D|}.$$
- The more values an attribute has, often the bigger its intrinsic value.
- In C4.5, the IMPORTANCE method will select the attribute a^* that maximize gain ratio; that is,
$$a^* = \arg\max Gain\_ratio(D, a).$$

Gini Index

- Algorithm CART by Breiman et al., 1984 proposes to use *Gini index*.
- Purity of dataset also can be quantified by the Gini value:

$$Gini(D) = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2,$$

  where p_k is the percent of examples in D that are labeled by y_k.
- Intuitively, $Gini(D)$ is the probability of two randomly chosen examples from D being labeled differently.
- Therefore, the smaller $Gini(D)$, the more purity in D.
- Now we define the Gini index of an attribute in dataset D:

$$Gini\_index(D, a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|} Gini(D^v),$$

- In CART, the IMPORTANCE method will select the attribute a^* that minimize Gini index; that is,

$$a^* = \arg\min Gini\_index(D, a).$$

Continuous Attributes

- Attributes with continuous values cannot be directly selected for a node in the decision tree.
- Ways to handle them: discretization, and bi-partition.
- We focus on bi-partition, used in C4.5 by Quinlan, 1993.
- Given D and continuous attribute a, suppose a's values in D are $a^1, \ldots, a^n$, in order.
- A *split point* t separates D to D_t^- (set of examples where attribute a's value is no bigger than t) and D_t^+ (set of examples where a's value is bigger than t).
- We now create a candidate set of possible split points:

$$T_a = \{\frac{a^i + a^{i+1}}{2} | 1 \leq i \leq n-1\}.$$

Continuous Attributes in ID3

- We first define information gain of attribute a for a split point $t \in T_a$:

$$Gain(D, a, t) = Ent(D) - \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda).$$

- All we need now is to adjust the information gain formula in ID3 for the continuous attribute a:

$$Gain(D, a) = \arg\max_{t \in T_a} Gain(D, a, t).$$

- Unlike categorical attributes, continuous attributes can appear multiple times along a path in the decision tree.

Continuous Attributes in C4.5

- All we need now is to adjust the gain ratio formula in C4.5 for the continuous attribute a:

$$\begin{aligned} Gain\_ratio(D, a) &= \arg\max_{t \in T_a} Gain\_ratio(D, a, t) \\ &= \arg\max_{t \in T_a} \frac{Gain(D, a, t)}{IV(D, a, t)}, \end{aligned}$$

where $IV(D, a, t) = -\sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} \cdot \log_2 \frac{|D_t^\lambda|}{|D|}$

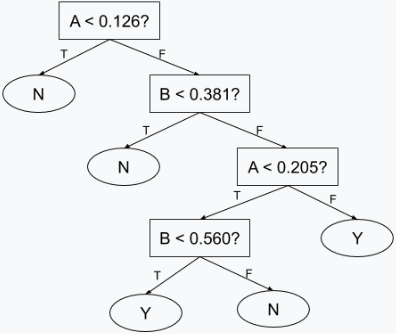
Continuous Attributes in CART

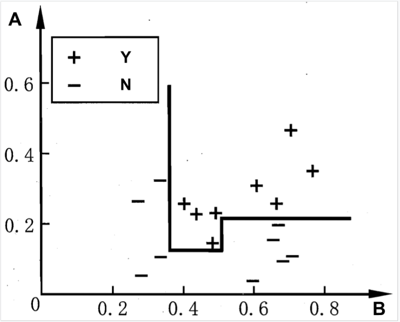
- All we need now is to adjust the Gini index formula in CART for the continuous attribute a:

$$
\begin{aligned}
Gini\_index(D,a) &= \underset{t \in T_a}{\arg\min}\ Gini\_index(D,a,t) \\
&= \underset{t \in T_a}{\arg\min} \sum_{\lambda \in \{-,+\}} \frac{|D_t^\lambda|}{|D|} Gini(D_t^\lambda)
\end{aligned}
$$

Decision Trees' Decision Boundaries

- Decision boundaries produced by decision trees are *axis-parallel*; that is, they are segments parallel to the axes.
- This makes decision trees very explainable, but requires many many segments when the learning task needs complex decision boundaries.





Multi-Variate Decision Trees

- Every non-leaf node in a multi-variate tree has a linear combination of multiple attributes, instead of a single attribute:

$$\sum_{i=1}^{d} w_i a_i < t.$$

- Decision boundaries produced by multi-variate decision trees are *axis-oblique*; that is, they possibly are not parallel to the axes.