Lecture III: Classification and Decision Trees (CART)

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

January 27, 2026

Classification and regression tree(s) (CART)

Learnin a CART

Predicting with a CART

Some issues with CART

Reading HTF Ch.: 9.2 CART, Murphy Ch.: 16.2.1–4 CART, Bach Ch.:

Classification and regression trees (CART)

- A **classification tree** or (**decision tree**) is built recursively by splitting the data with hyperplanes parallel to the coordinate axes.
  - At each split, try to separate $+$ examples from $-$ examples as well as possible.
  - Eventually, all the regions will be "pure", i.e. will contain examples from one class only.
- Classification trees can be used in multiway classification as well (there we try to create a pure region on at least one side of the split)
- A **regression tree** uses the same principle for regression
  here we try to obtain regions where the outputs are nearly the same

Classification Tree (Decision Tree)

Regression Tree

Hierarchical partitions

- a **hierarchical partition** $\mathcal{T}$ of $\mathbb{R}^d$ is a set of regions $\{R_q\}$, so that $\mathbb{R}^d = \bigcup_q R_q$ and between any two $R_q, R_{q'}$ we have either

$$R_q \cap R_{q'} = \emptyset, \text{ or } R_q \subset R_{q'} \text{ or } R_{q'} \subset R_q. \quad (1)$$

(we include the boundariy between 2 regions $R_q, R_{q'}$ arbitrarily in a single one of them)

- In a CART, the partitions are usually chosen to be axis-aligned, i.e. $R_q = \{x \,|\, \pm x_{j_1} ">" \tau_1, \pm x_{j_2} ">" \tau_2, \ldots \pm x_{j_l} ">" \tau_l\}$ where $">"$ stands for one of $>$ or $\geq$, so that the union of all regions covers $\mathbb{R}^d$.
- The number of inequalities l defining the region is called the *level* of the region.
- R_q is a **leaf** of $\mathcal{T}$ iff there is no other $R_{q'}$ included in it.

Example (**A hierarchical partition with 3 levels over $\mathbb{R}^2$**)

Level 1:	$R_1 = \{x \,|\, x_2 > 3\}$,
	$R_2 = \{x \,|\, x_2 \leq 3\}$
Level 2:	$R_3 = \{x \,|\, x_2 > 3,\ x_1 \geq -2\}$,
	$R_4 = \{x \,|\, x_2 > 3,\ x_1 < -2\}$,
	$R_5 = \{x \,|\, x_2 \leq 3, x_1 > 0\}$,
	$R_6 = \{x \,|\, x_2 \leq 3, x_1 \leq 0\}$
Level 3:	$R_7 = \{x \,|\, x_2 > 3,\ x_1 \geq -2, x_1 < 4\}$,
	$R_8 = \{x \,|\, x_2 > 3,\ x_1 \geq 4\}$,
	$R_9 = \{x \,|\, x_2 < 3, x_1 \geq 1\}$
	$R_{10} = \{x \,|\, x_2 \leq 3, x_1 \leq 0, x_2 > -1\}$,
	$R_{11} = \{x \,|\, x_2 \leq -1, x_1 \leq 0\}$,
	$R_{12} = \{x \,|\, x_2 < 3, x_1 > 0, x_1 < 1\}$

The leaves are $R_4, R_7, \ldots R_{12}$. Not all leaves are at the same level; for example R_4 is at level 2.

Some advantages of CART

- Natural and easy to interpret (if small)
- Can approximate any function (with enough leaves)

"Learning" a CART

A standard algorithm for building a decision tree works recursively in top-down fashion.

Input Training set $\mathcal{D}$ of size n
Initialize with a tree with only one region, that contains all the data
Repeat until all leaves are pure (or until desired purity is attained in all leaves)

1.
2. Find the "optimal" split over all leaves R_q and all possible splits of R_q.
   "Optimal" is defined in terms on purity (e.g split the least pure leaf, find the split that makes one of the new leaves pure)
3. Perform the "optimal" split and add the two new leaves to the tree

This is a greedy algorithm. Sometimes, trees obtained this way are pruned back to smaller sizes.

Purity

- Natural ways to set y_q based on the data, once the partition $\mathcal{T}$ has been fixed:
  - denote $Y_q = \{y^i \,|\, x^i \in R_q,\ i = 1:N\}$ the set of labels at a leaf R_q
  - Regression y_q = average of Y_q
  - Classification y_q = majority label of Y_q
- a leaf R_q is **pure** if all labels are the same, i.e. if $|Y_q| = 1$
- criteria for the **(im)purity** of a leaf R_q
  - Regression impurity = sample variance of Y_q
  - Classification let p_q be the frequency of y_q in Y_q

$$\text{impurity} \;=\; \begin{cases} \text{Misclassification error} & 1 - p_q \\ \text{Gini} & p_q(1 - p_q) \\ \text{Entropy} & p_q \ln p_q + (1 - p_q)\ln(1 - p_q) \end{cases} \tag{2}$$

These measures generalize naturally to the multiclass setting.

Predicting with a CART

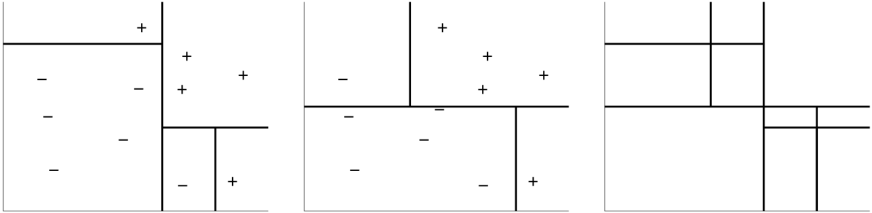
Given new x

1. Find the unique leaf $R(x)$ so that $x \in R(x)$
2. Predict $\hat{y}$ based on the data in this leaf
   - **Regression**
     Predict $\hat{y}(x) = \mathrm{average}\{y^i \text{ with } x^i \in R(x)\}$
   - **Classification**
     Predict $\hat{y}(x) = \mathrm{majority}\{y^i \text{ with } x^i \in R(x)\}$

A decision tree over $\mathcal{D}$ is not unique

Same dataset $\mathcal{D}$, two different trees. Both classify the sample $\mathcal{D}$ perfectly.

$\mathcal{T}_1$ $\mathcal{T}_2$ differences of $\mathcal{T}_1$, $\mathcal{T}_2$



But they produce different decision regions.