Lecture
–
Decision Trees

Histogram rules - Local averaging

- K-NN limitations: a nearest neighbor may be very far from X!
- Consider a **partition** of the feature space:

$$C_1 \bigcup \cdots \bigcup C_K = \mathcal{X}$$

- Apply the **majority rule**: suppose that X lies in C_k,
  1. Count the number of training examples with positive label lying in C_k
  2. If $\sum_{i:\ X_i \in C_k} \mathbb{I}\{Y_i = +1\} > \sum_{i:\ X_i \in C_k} \mathbb{I}\{Y_i = -1\}$, predict $Y = +1$. Otherwise predict $Y = -1$.
- This corresponds to the "plug-in" classifier $2\mathbb{I}\{\widehat{\eta}(x)\} - 1$, where

$$\widehat{\eta}(x) = \sum_{k=1}^{K} \mathbb{I}\{x \in C_k\} \frac{\sum_{i=1}^{n} \mathbb{I}\{Y_i = +1,\ X_i \in C_k\}}{\sum_{i=1}^{n} \mathbb{I}\{X_i \in C_k\}}$$

is the **Nadaraya-Watson estimator** of the posterior probability.

Kernel rules - Local averaging

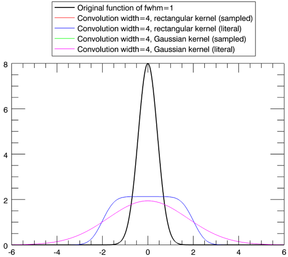
- Smooth the estimator/boundary decision!
- Replace the indicator function by a **convolution kernel**:

$$K : \mathbb{R}^d \to \mathbb{R}_+, \quad K \geq 0, \text{ symmetric and } \int K(x)dx = 1$$

- Bandwidth $h > 0$ and **rescaling**

$$K_h(x) = \frac{1}{h} K(x/h)$$

- Examples: Gaussian kernel, Novikov, Haar, *etc.*



Kernel rules - Local averaging

- If $\sum_{i=1}^{n} \mathbb{I}\{Y_i = +1\} K_h(x - X_i) > \sum_{i=1}^{n} \mathbb{I}\{Y_i = -1\} K_h(x - X_i)$, predict $Y = +1$. Otherwise predict $Y = -1$.

- This corresponds to the "plug-in" classifier $2\mathbb{I}\{\widetilde{\eta}(x)\} - 1$, where

$$\widetilde{\eta}(x) = \frac{\sum_{i=1}^{n} \mathbb{I}\{Y_i = +1\} K_h(x - X_i)}{\sum_{i=1}^{n} K_h(x - X_i)}$$

  is the **Nadaraya-Watson estimator** of the posterior probability.
- **Statistical argument:** if η is a "smooth" function, $\widetilde{\eta}$ may be a better estimate than $\widehat{\eta}$ (smaller variance but... biased)

Decision Trees: the CART Algorithm

- If the partition is picked in advance (before observing the data)...

Decision Trees: the CART Algorithm

- If the partition is picked in advance (before observing the data)... many cells may be empty!

Decision Trees: the CART Algorithm

- If the partition is picked in advance (before observing the data)... many cells may be empty!
- Choose the partition **depending on the training data!**

Decision Trees: the CART Algorithm

- If the partition is picked in advance (before observing the data)... many cells may be empty!
- Choose the partition **depending on the training data!**
- The CART Book - Breiman, Friedman, Olshen & Stone (1986)
- **Greedy** Recursive Dyadic Partitioning: $X = (X^{(1)}, \ldots, X^{(d)}) \in \mathbb{R}^d$

Decision Trees: the CART Algorithm

- Training data $(X_1, Y_1), \ldots, ; (X_n, Y_n)$
- For any subset $R \subset \mathcal{X}$, consider the **majority label**: $\bar{Y}_R$ where

$$\bar{Y}_R = +1 \text{ if } \sum_{i=1}^{n} \mathbb{I}\{Y_i = +1,\ X_i \in R\} > \frac{1}{2} \sum_{i=1}^{n} \mathbb{I}\{X_i \in R\}$$

and $\bar{Y}_R = -1$ otherwise

- One starts from the root node $R = \mathcal{X} = C_{0,0}$ and the (constant classifier) $\bar{Y}_{C_{0,0}}$. The goal pursued is to split the cell $C_{0,0}$

$$C_{0,0} = C_{1,0} \bigcup C_{1,1}$$

so as to refine the classifier and produce

$$g_1(x) = \bar{Y}_{C_{1,0}} \mathbb{I}\{x \in C_{1,0}\} + \bar{Y}_{C_{1,1}} \mathbb{I}\{x \in C_{1,1}\}.$$

"Growing the Tree"

- The partition of the cell $C_{0,0} = \mathcal{X}$ is selected in order to minimize $\widehat{L}_N(g_1)$, or equivalently the *impurity measure*

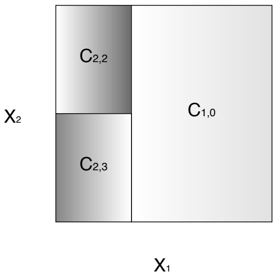
$$\sum_{i=1}^{N} \mathbb{I}\{X_i \in C_{1,0},\ Y_i \neq \bar{Y}_{C_{1,0}}\} + \mathbb{I}\{X_i \in C_{1,1},\ Y_i \neq \bar{Y}_{C_{1,1}}\}$$

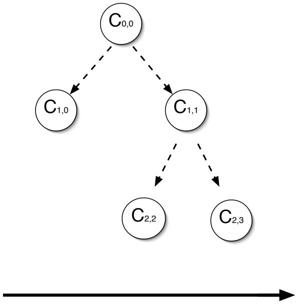
- Consider subsets of the form

$$\begin{aligned} C_{1,0} &= C_{0,0} \cap \{X^{(j)} \leq s\}, \\ C_{1,1} &= C_{0,0} \cap \{X^{(j)} > s\}. \end{aligned}$$

- It is sufficient to choose the best split values among the $X_i^{(j)}$'s!

Decision Trees: the CART Algorithm





"Growing the Tree"

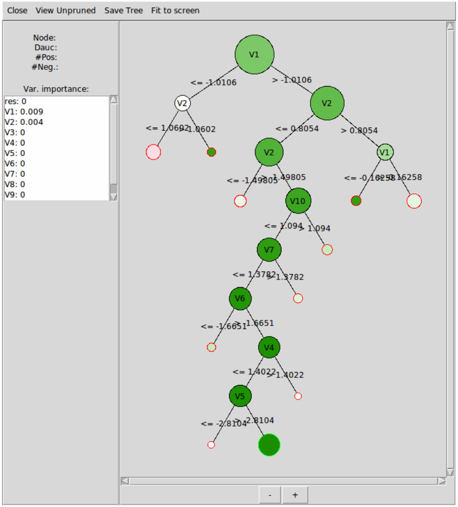
- "Growing the Tree": iterate in order to split $C_{j,k}$ if it is not pure and contains at least $n_{\min}$ training observations
  1. For $j = 1$ to d, find s (best split value) so as to minimize the impurity of the regions

$$C_{j,k} \cap \{X_j > s\} \;\; \textit{and} \;\; C_{j,k} \cap \{X_j \leq s\}$$

  2. Find the best split variable X_j

- Measuring **impurity**:
  - misclassification error
  - Gini index

Decision Trees: the CART Algorithm



The CART algorithm

- Qualitative variables
- Incomplete data
- Relative Importance
- Randomization
- Diagonal splits
- Asymmetric cost
- Multiclass, regression
- Best subtrees, "pruning" the tree
- Alternative tree learning algorithm: C4.5 (Ross Quinlan)