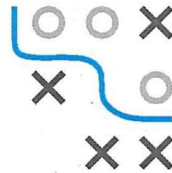
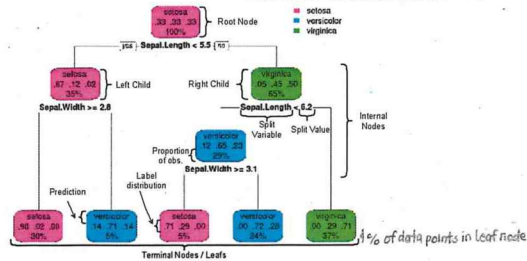


CLASSIFICATION TREES [CART]

Regression Trees work the same way, but with constant numerical prediction (mean)

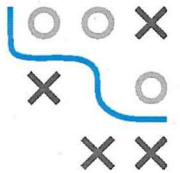


TREE AS AN ADDITIVE MODEL

Trees divide the feature space \mathcal{X} into rectangular regions:

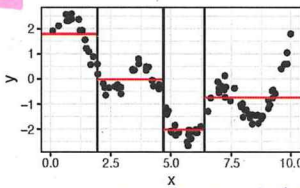
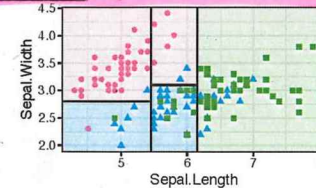
$$f(x) = \sum_{m=1}^M c_m \mathbb{I}(x \in Q_m),$$

defines Hypothesis Space



where a tree with M leaf nodes defines M "rectangles" Q_m .
 c_m is the predicted numerical response, class label or class distribution in the respective leaf node.

Classif Trees: Not great if linear decision boundary is best



flexible with non-linearities, more stable vs. outliers than normal LM
 Regression Trees with L2 loss:
 - piecewise constant LM (same as kNN) (bagging improves this)
 - can model sloped curve or smooth f.d. in several
 - linear dependencies can only be modelled over several splits
 - constant pred. at edges!
 - terrible extrapolation to feature value outside train range and often bad performance on test data

- Classification trees use the structure of a binary tree
- Binary splits are constructed top-down in a data optimal way
- Each split is a threshold decision for a single feature
- Each node contains the training points which follow its path
- Each leaf contains a constant prediction
- Predicting new data: Use learned split points & pass observation through tree

recursive greedy optimization at each node N
 Ideally all possible splits of N on all possible points t across all features X_j are considered w.r.t. $R_{emp}()$
 Each split has intermediate result, where train data are "distributed" to new (two) child nodes
 Procedure repeats in child nodes
 May want to implement Early Stopping

Hyperparameter: Stopping Criteria and its concrete values

Without Stopping Criteria CART could run until every leaf contains single train obs.!

Very complex Trees will massively overfit the training data

Possible Steps (no more splitting) at node: If certain leaf number reached, to few obs. in node or its children, risk improvement too small, or maxdepth (maxsplits)

Leaf nodes (sometimes 4/ down) identical feature values etc.

Classif: $\mathbb{I}_{k \neq k'} = 1$ (all others 0)
 \Rightarrow Gini & entropy impurity = 0

CATEGORICAL FEATURES

- A split on a categorical feature partitions the feature levels:
 $x_j \in \{a, b, c\} \leftarrow N \rightarrow x_j \in \{d, e\}$
- For a feature with m levels, there are about 2^m different possible partitions of the m values into two groups ($2^m - 1$ because of symmetry and empty groups).
- Searching over all these becomes prohibitive for large values of m .
- For regression with L2 loss and for binary classification, we can define clever shortcuts.

For 0 - 1 responses, in each node:

- Calculate the proportion of 1-outcomes for each category of the feature in N .
- Sort the categories according to these proportions.
- The feature can then be treated as if it was ordinal, so we only have to investigate at most $m - 1$ splits.

For continuous responses, in each node:

- Calculate the mean of the outcome in each category
- Sort the categories by increasing mean of the outcome

Brier Score: $\frac{1}{M} \sum_{k=1}^M (\bar{\pi}_k - \sigma_k)^2$
 Log Loss: $-\sum_{k=1}^K \sigma_k \log(\bar{\pi}_k)$
 insert $\frac{1}{M} \sum_{k=1}^M \dots$ for impurity

FINDING THE BEST SPLIT

Empirical risk $t \in \{ \text{midpoint of two unique } (x_j) \text{ vals} \}$

- Splitting feature x_j at split point t divides a parent node N into two child nodes:

$$N_1 = \{(x, y) \in N : x_j \leq t\} \text{ and } N_2 = \{(x, y) \in N : x_j > t\}$$

- Compute empirical risks in child nodes and minimize their sum to find best split (impurity reduction):

$$\arg \min_{j,t} R(N, j, t) = \arg \min_{j,t} R(N_1) + R(N_2)$$

Automatic feature selection: favours numerical / categ. x_j with high m

Note: If R is the average instead of the sum of loss functions, we need to rewrite:

g-way classification: $\hat{\pi}_k^{(N)} = \frac{1}{|N|} \sum_{(x,y) \in N} \mathbb{I}(y=k)$
 Brier score \rightarrow Gini impurity
 Bernoulli loss \rightarrow entropy impurity
 Regression (quadratic loss): $R(N) = \sum_{(x,y) \in N} (y - c)^2$ with $c = \frac{1}{|N|} \sum_{(x,y) \in N} y$

Optimization in i -th child node: $c_{N_i} = \arg \min_c \sum_{(x,y) \in N_i} L(y, c)$ optimal const. prediction, L doesn't have to be different. In loop-based tree opt.

- Exhaustive search over all split candidates, choice of risk-minimal split ("greedy" search)
- In practice: reduce number of split candidates (e.g., using quantiles instead of all observed values)

Split evaluation only possible for observations where used feature has no NA \Rightarrow can bias splits towards features with many NAs but bigger problem is NAs during prediction time (new obs.)

SURROGATE SPLITS

- created during training as alternative split (yes) using other features at that node, mimicking outcome of optimal x_j -split (really only ONE split)
 e.g. by learning $(x_j < c) \Rightarrow \text{TRUE or FALSE}$ as target column
- typically create several surrogates (like 5) for every inner node \Rightarrow order of surrogates to use when better surrogate-feature also NA

Node becomes Leaf Node if $R(N) \leq R(N_1) + R(N_2)$ for all possible splits

In Classif-CART usually until Node is "pure" (exception: Multiple obs. with identical values for all x_j but different classes \Rightarrow Leaf Node not pure)

In Regr-CART splitting typically until almost every obs. in its own leaf node

Exercise: CART Trees and Bagging/Random Forests

Regression Trees

We derive optimal const. predictor for node N (mean) by minimizing R_{emp} under L2 Loss

$$\arg \min_{c \in \mathbb{R}} \sum_{i=1}^M (y^{(i)} - c)^2 = \sum_{i=1}^M (y^{(i)2} - 2cy^{(i)} + c^2) = Mc^2 - 2c \sum_{i=1}^M y^{(i)} + \sum_{i=1}^M y^{(i)2}$$

$$\Rightarrow \frac{\partial}{\partial c} \left(\sum_{i=1}^M (y^{(i)} - c)^2 \right) = 2Mc - 2 \sum_{i=1}^M y^{(i)} \stackrel{!}{=} 0 \Leftrightarrow Mc = \sum_{i=1}^M y^{(i)} \Leftrightarrow \hat{c} = \frac{1}{M} \sum_{i=1}^M y^{(i)} = \bar{y}_{Node}$$

same result for average risk $\frac{1}{M} R_{emp}$

\bar{y} minimizes summed L2-Loss in Node. Plugging in \bar{y} for c we get for average risk: $\frac{1}{M} \sum_{i=1}^M (y^{(i)} - \bar{y})^2$

This is identical to the (biased) sample variance \Rightarrow Predicting sample mean minimizes both L2 risk and variance impurity

Quite an intuitive look at Prediction variance via L2 minimization, also connects to intercept LM trained with L2 Loss within Node (piecewise for entire Tree)

That nicely explains why regression trees (without Early Stopping) grow so deep: Variance of Subsets cannot possibly be worse than variance of parent node. It's impossible to make the Regr. Tree worse by introducing a further split. Most likely it gets a lot better w.r.t. train error!

Classification Trees Develop intuition for Brier Score / Log Loss and their resulting Node Evaluation Metric (Gini / Entropy / Impurity) for Splitting.

Average / cross entropy loss
a) Multiclass Log Loss in Node: $\phi_{LL} = \left(\frac{1}{M} \right) \sum_{i=1}^M \sum_{k=1}^g \mathbb{I}[y^{(i)}=k] \log(\hat{\pi}_k) = - \sum_{k=1}^g \left(\frac{1}{M} \sum_{i=1}^M \mathbb{I}[y^{(i)}=k] \right) \log(\hat{\pi}_k)$ CART: $\hat{\pi}_k$ indep. of i , const. node pred.

Inserting $\hat{\pi}_k = \frac{1}{M} \sum_{i=1}^M \mathbb{I}[y^{(i)}=k]$ we get $-\sum_{k=1}^g \hat{\pi}_k \log(\hat{\pi}_k) \triangleq$ Entropy impurity that we use to quantify empirical risk of single obs./pred. in Node

Average
Multiclass Brier Score in Node: $\phi_{BS} = \left(\frac{1}{M} \right) \sum_{i=1}^M \sum_{k=1}^g (\mathbb{I}[y^{(i)}=k] - \hat{\pi}_k)^2 = \sum_{k=1}^g \frac{1}{M} \sum_{i=1}^M (\mathbb{I}[y^{(i)}=k] - \hat{\pi}_k)^2$

Inserting $\hat{\pi}_k = \frac{1}{M} \sum_{i=1}^M \mathbb{I}[y^{(i)}=k]$ yields that we can write the inner part as the variance of the binomial random variable $\mathbb{I}[y=k] \in \{0,1\}$ success probability π_k "estimated" by $\hat{\pi}_k$

(Biased) empirical Variance of $\mathbb{I}[y^{(i)}=k]$: $\hat{\pi}_k (1 - \hat{\pi}_k)$

\Rightarrow Overall we get the Gini impurity $\sum_{k=1}^g \hat{\pi}_k (1 - \hat{\pi}_k) = \sum_{k=1}^g \hat{\pi}_k - \sum_{k=1}^g (\hat{\pi}_k)^2 = 1 - \sum_{k=1}^g (\hat{\pi}_k)^2$

b) Fractions of the classes $k=1, \dots, g$ in Node N of decision tree are $\hat{\pi}_k^{(N)} = \frac{1}{M} \sum_{(x^{(i)}, y^{(i)}) \in N} \mathbb{I}[y^{(i)}=k]$

Deterministic prediction rule for hard label: $\hat{k}|N = \arg \max_k \hat{\pi}_k^{(N)}$ pick class with highest empirical frequency

Randomizing rule: $\hat{k} \sim \text{Cat}(\hat{\pi}_1^{(N)}, \dots, \hat{\pi}_g^{(N)})$ draw from categorical distribution parameterized by these same empirical frequencies

We assume train data for target $y^{(i)}$ are i.i.d. drawn from this categ. distr., $P(y^{(i)}=k|N) = \hat{\pi}_k^{(N)} \xrightarrow{\text{stochastic pred. rule}} P(\hat{y}^{(i)}=k|N) = \hat{\pi}_k^{(N)}$ by design identical categ. distr.
most frequent class is drawn with prob. $\max_k \hat{\pi}_k^{(N)}$, so in expectation we predict max class most often
 $y^{(i)}$ randomness obvious here, $\hat{y}^{(i)}$ randomness from training sample!

Define MCE for node with $n=M$ when using randomizing pred. rule: $\phi_{MCE}(N) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[y^{(i)} \neq \hat{y}^{(i)}]$

We will show that the expectation of the Misclassification Error in this setting is exactly the Gini impurity \triangleq expected freq. with which training sample (obs.) will be misclassified in given node $N \Rightarrow$ minimize!

$$\begin{aligned} \mathbb{E}_{y^{(i)}, \hat{y}^{(i)}}(\phi_{MCE}(N)) &= \mathbb{E}_{y^{(i)}, \hat{y}^{(i)}} \left(\frac{1}{n} \sum_{i=1}^n \mathbb{I}[y^{(i)} \neq \hat{y}^{(i)}] \right) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{y^{(i)}, \hat{y}^{(i)}} (\mathbb{I}[y^{(i)} \neq \hat{y}^{(i)}]) \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{y^{(i)}} \left(\sum_{k=1}^g \hat{\pi}_k^{(N)} \cdot \mathbb{I}[y^{(i)} \neq k] \right) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{y^{(i)}} (1 - \hat{\pi}_{k=y^{(i)}}^{(N)}) = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^g \hat{\pi}_k^{(N)} \cdot (1 - \hat{\pi}_k^{(N)}) \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^g \hat{\pi}_k^{(N)} \cdot \mathbb{I}[y^{(i)} \neq k] = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^g \hat{\pi}_k^{(N)} \cdot P(y^{(i)} \neq k) = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^g \hat{\pi}_k^{(N)} \cdot (1 - \hat{\pi}_k^{(N)}) \end{aligned}$$

Random Forests

Let's show that the prob. for an observation to be OOB in arbitram bootstrap sample converges to $\frac{1}{e} \approx 0.37$

Prob. of not being drawn = $1 - \frac{1}{n} \Rightarrow$ sample with replacement: $P(i \in \text{OOB}) = \left(1 - \frac{1}{n}\right)^n \xrightarrow{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = e^{-1} = \frac{1}{e} \approx 0.37$

When we use OOB error for GE-estimation then we do kind of a similar thing as a simple holdout split with $|D_{test}| = 0.37$; just more advanced with \hat{y} averaging effect & smoothing component
*sufficiently large n , $P(i \in \text{OOB}) < 0.37$ for small datasets!