Right sized trees
(when to stop splitting)

Dilemma:

- If tree (# of regions) is too small, then piecewise constant approx too crude (bias) causing increased prediction error

- If tree is too large, then it fits the training data too closely (overfitting) causing increased variance \( \Rightarrow \) prediction error.

\[
\text{risk} = E[y - \hat{F}_M(x)]^2
\]

\( M^* \) = right sized tree

\( M = \text{tree size} \)
risk \sim (bias)^2 + (variance)

as tree gets larger (more regions)
(bias)^2 goes down
(variance) goes up

therefore, for any particular
problem, there is a right sized
tree (# of regions - terminal nodes)

How to find it?
Possible stopping rules:

A. \#(R_m) too small (bounds variance) too crude

B. Next split not worthwhile

Let \[ \hat{E}_m = \frac{1}{N} \sum_{x_i \in R_m} (y_i - \bar{y}_m)^2 \]

= contribution of \( R_m \) to MSE

\[ \frac{1}{N} \sum_{i=1}^{N} \left[ y_i - \hat{F}(x_i) \right]^2 = \sum_{m=1}^{M} \hat{E}_m \]
B. Next split not worthwhile

\[ \hat{I}_m = \hat{e}_m - \hat{e}_{mL} - \hat{e}_{mR} \]

\[ \hat{I}_m > 0 \text{ even if } I_m < 0 \]

\[ \hat{I}_m \text{ optimistically biased estimate of } I_m \]

Introduce improvement threshold \( k \):

If \( \hat{I}_m \geq k \) accept, split
ELSE make \( m \) terminal

Not sufficient condition!
- Not a sufficient condition

It often happens that a single split is not worthwhile by itself but it opens possibilities for further splitting.

\[
\begin{array}{cc|cc}
 & & 1 & 2 \\ \hline
1 & & & \\
2 & & & \\
\end{array}
\]

symmetric interaction effect
\[ \hat{I}_m = \hat{e}_m - \hat{e}_{m+1} - \hat{e}_{m+3} - \hat{e}_{m+4} \]

two splits

If \( \hat{I}_m > 2 \kappa \) accept both splits
ELSE make \( m \) terminal

\[ \hat{I}_m \geq 2 \kappa \iff \hat{e}_m + \kappa \geq \hat{e}_{m+1} + \kappa \]
\[ + \hat{e}_{m+3} + \kappa \]
\[ + \hat{e}_{m+4} + \kappa \]

cost of node \( m \)

sum of the costs of its terminal descendents
Let $\hat{\lambda}_m \equiv \hat{e}_m$
Define \textit{cost} of node $m$

If $m$ is terminal
\[ C_m = \hat{\lambda}_m + \kappa \]

If $m$ is nonterminal
\[ C_m = \sum_{m' \in \mathcal{m}} C_{m'} \]
\[ \uparrow \text{terminal descendents} \]

Make choice to minimize $C_m$. 
Note:

$$\sum_{m' \in E_m} C_{m'} = \sum_{m' \in E_m^L} C_{m'} + \sum_{m' \in E_m^R} C_{m'}$$

↑

Cost of $m$

↑

Cost of $m_L$

↑

Cost of $m_R$

Termination Rule:

If $\hat{C}_m + \kappa \leq C_{m_L} + C_{m_R}$
then make $m$ terminal
and set $C_m = \hat{C}_m + \kappa$

ELSE
accept split
and set $C_m = C_{m_L} + C_{m_R}$

Apply in inverse order of depth.
Recursive partitioning Procedure:

A. Apply recursive splitting top down as far as possible

B. Apply termination rule to each nonterminal node (in turn) in inverse order of depth (bottom up)

Question: how to choose \( k \)?
Termination rule (8) is an algorithm for solving the following optimization problem:

\[ \mathcal{T} = \text{set of all possible trees arbitrarily terminated} \]

\[ T \in \mathcal{T} \text{ and } |T| = \text{number of terminal nodes (regions)} = M \]

\[ \hat{\mu}(T) = \sum_{m=1}^{M} \hat{\mu}_m \]

\[ \text{terminal nodes (regions)} \]

\[ = \text{empirical risk for tree } T \]
Problem: find

\[ T^*_k = \min \left[ \hat{\lambda}(T) + k \mid T \mid \right] \]

\( k \) = complexity parameter
(penalty for each terminal region)

\( T^*_k \) = optimally terminated tree
for complexity value \( k \).

\( k' > k \Rightarrow T^*_{k'} \leq T^*_k \)

for all \( k' > k \) there are at most \( |T^*_k| \) distinct trees - all of which can be found without resplitting
Setting \( k=0 \) constructs the largest tree – split until all nodes are pure.

From \( T_0^* \) we can find all distinct trees for all \( k > 0 \). There are at most \( |T_0^*| \) of them.

This gives a (nested) sequence of trees indexed by a single parameter, \( k \).

\( k \) is the only parameter of the procedure.

It penalizes for the increased variance associated with more complex model linear regression.

\[
\text{Var} \hat{\gamma} \sim 10^2 / N
\]
Choose the value of $k$, $k^*$, (and thus $T_{k^*}$) to minimize future, $\mathcal{R}(T_{k^*})$, risk.

1) resubstitution estimate (always gives $k^* = 0$, largest possible tree)

2) "test set" estimate

3) v-fold cross validation

Use the last two "honest" estimates—depending on sample size—to get estimate of best (right sized) tree.
Bias - variance trade-off

![Graph showing bias and variance trade-off with MSE risk and variance curves.]
**Classification**

(Trees most used)

\( y \in \{ c_1, c_2, \ldots, c_K \} \) unordered categorical "class label"

\( \hat{c}(x) = \text{prediction } y \mid x \)

**Tree structural model:**

\( \hat{c}(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m) \)

\( \{ R_m \}_1^M \) same type as regression trees

if \( x \in R_m \) \( \Rightarrow \hat{c}(x) = \hat{c}_m \)

\( \{ \hat{c}_m \in \{ c_1, c_2, \ldots, c_K \} \}_1^M \)

\( \Rightarrow \hat{c}(x) \in \{ c_1, c_2, \ldots, c_K \} \)

**Score criterion:**

\[ \text{population: } \mathbb{E}_{y \in \mathcal{X}} L(y, \hat{c}(x)) \]

= expected misclassification loss ("risk")

\( L(y, c) = \text{loss predicting } c \) when truth is \( y \)
often \( L(y, c(x)) = I(y \neq c(x)) \)

**misclassification risk**

\( \hat{r} = E_{y,x} \, L(y, c(x)) = E_{y,x} \, I(y \neq c(x)) \)

= error rate

(assumes all misclassification errors have *same* cost)

**Data score criterion:**

natural candidate:

\[ \hat{r} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, c(x_i)) \]

empirical risk on training data.

**Search strategy:**

Given \( \mathcal{L}_{RM}^{3,M} \)

\[ \hat{c}_m = \arg\min_c \sum_{x_i \in \mathcal{R}_m} L(y_i, c) \]
Need to find good $\{ \hat{R}_m \}_{m=1}^M$

Recursive partitioning

$R_1 = S' = $ entire predictor variable space.

Splitting rule:

$$
\hat{R}_m
\begin{array}{c}
\downarrow \\
\hat{j}
\end{array}
\begin{array}{c}
\hat{R}_m^{(i)} \\
\hat{R}_m^{(ii)}
\end{array}
$$

$x_j \in \hat{S}_m$

$x_j \in \hat{S}_m^c$

Types of splits same as regression:

- numeric $x_j \rightarrow$ split point
- categorical $x_j \rightarrow$ subset

Need to find optimal split for entire tree:

$$
\hat{R}_M = \sum_{m=1}^M \hat{p}_m \hat{R}_m
$$
where: (empirical)

\[ \hat{p}_m = \frac{N_m}{N} = \text{prob. of } x \in R_m \]

\[ \hat{r}_m = \frac{1}{N_m} \sum_{x \in R_m} L(y_i, \hat{c}_m) \quad \text{risk given } x \in R_m \]

\[ N_m = \#(R_m) = \# \text{ in } R_m \]

Improvement as a result of split:

\[ I(j_m, \delta_{jm}) = \]

\[ \hat{p}_m \hat{r}_m - \hat{p}_m \hat{r}_m - \hat{p}_m \hat{r}_m \]

\[ (A) \quad (B) \quad (C) \]

A = contribution to model risk of parent
B = " " " " left daughter
C = " " " " right "

Optimal split:

\[ (m^*, j^*, \delta_{jm}^*) = \text{argmax } I(j_m, \delta_{jm}) \]

\[ m, j, \delta_{jm} \]
A problem!

The sequence of cuts

$(1, S_a)$

$(2, S_b)$

$(2, S_c)$

gives zero estimated risk

But, recursive partitioning by sequentially minimizing 1-dimensional estimated risk's leads to totally ineffective partitioning

$(1, S_d)$

$(2, S_b)$
What happened?

Misclassification risk is not a continuous function of model parameters!

Problem for all classification procedures.

In this case, each split minimizes estimated risk assuming that this is the final split. It does not take into account possibility of further splitting.

Greedy strategy gives bad results.

What to do?

1. Complete combinatorial optimization (look ahead)

2. Substitute surrogate criterion that is continuous in parameters to be used with greedy strategy (recursive partitioning) that has the same (population) solution
How?

$$\text{loss } (y = c_x, \ c(x) = \ell_k) = \mathcal{L}(c_x, c_k) = \sum_{k=1}^{K} \mathcal{L}(c_x, c_k) \mathbb{I}(y = c_k | x)$$

risk in predicting $c(x) = c_k$ when truth is $y = c_x$

$$\lambda_k(x) = \mathbb{E} \mathcal{L}(y = c_x, c(x) = c_k)$$

$$= \sum_{k=1}^{K} \mathcal{L}(c_x, c_k) \mathbb{E} \mathbb{I}(y = c_k | x)$$

$$= \sum_{k=1}^{K} \mathcal{L}(c_x, c_k) \mathbb{P}(y = c_k | x)$$

"Bayes" optimal prediction rule

$$\mathcal{R}^*(x) = \arg \min_{1 \leq k \leq K} \lambda_k(x); \ c^*(x) = c_{\mathcal{R}^*(x)}$$

$$= \arg \min_{1 \leq k \leq K} \sum_{k=1}^{K} \mathcal{L}(c_x, c_k) \mathbb{P}(y = c_k | x)$$

$$= \mathbb{P}(y = c_k | x)$$
Regression paradigm for classification
(used by most classification methods)

Obtain estimates \( \{ \hat{\beta}^*_k(x) \}_{k=1}^K \)

\( \hat{c}(x) = \arg\min_{1 \leq k \leq K} \sum_{i=1}^n L_{ik} \hat{\rho}^*_k(x) \); \( \hat{c}(x) = c_{\hat{\rho}^*(x)} \)

Note: \( \{ \hat{\rho}^*_k(x) \}_{k=1}^K \Rightarrow \hat{\rho}^*(x) = \rho^*(x) \)

Bayes optimal rule minimizes misclassification risk

Equivalent population score criterion

\( E[y \mid L(y, c(x))] < E[y \mid J(y, \{ \hat{\rho}^*_{\rho}(x) \}_{k=1}^K)] \)

where \( \{ \hat{\rho}^*_{\rho}(x) \}_{k=1}^K = \arg\min_{\{ \hat{\rho}^*_{\rho}(x) \}_{k=1}^K} E[J(y, \{ \hat{\rho}^*_{\rho}(x) \}_{k=1}^K) \mid x] \)

some solution with above rule

Need (surrogate) criterion \( J(\cdot, \cdot) \)
that is a smooth function of the parameters of \( \hat{\rho}^*_k(x) \)

Trees: split subsets on variables
One candidate: **squared-error loss**

\[ J(y, \{ \hat{f}_k^2 \}_1^K) = \sum_{m=1}^{K} \left[ I(y = c_m | x) - \hat{f}_m(x) \right]^2 \]

**Surrogate score criterion:**

**Population:**

\[ \left\{ f^*_m(x) \right\}_1^K = \arg\min_{\left\{ f_m(x) \right\}_1^K} \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{K} \left[ I(y_i = c_m | x) - f_m(x_i) \right]^2 \]

- \( K \)-numeric target functions
- \( \hat{f}_m(x) \) \( \Rightarrow \) \( K \)-response least-squares regression

**Note:** \( 0 \leq f_m(x) \leq 1 \) \( \forall i \) \( \sum_{m=1}^{K} f_m(x) = 1 \) for all \( x \)

**Data:**

\[ \left\{ \hat{f}_m(x) \right\}_1^K = \arg\min_{\left\{ f_m(x) \right\}_1^K} \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{K} \left[ I(y_i = c_m | x) - \hat{f}_m(x_i) \right]^2 \]

\( \hat{f} = \) class of functions, here = trees

\[ \hat{f}_m(x) = \sum_{m=1}^{M} \hat{f}_{m,m} \ I(x \in R_m) \]

piecewise constant approximation
Given \( \{ R_m \}_{m=1}^M \):

\[
\{ \theta_m \}_{m=1}^k = \text{argmin} \sum_{m=1}^k \sum_{x_i \in R_m} \left[ I(y_i = c_m) - \theta_m \right]^2
\]

minimize separately
for each \( m \)

\[
\hat{\theta}_m = \text{argmin} \sum_{x_i \in R_m} \left[ I(y_i = c_m) - \theta_m \right]^2
\]

\[
= \frac{\text{mean} \ I(y_i = c_m)_{x_i \in R_m}}{\sqrt{\#(R_m)}}
\]

\[
= \frac{N_{c_m \cap R_m}}{N_m}
\]

\[
\uparrow \quad \#(y_i = c_m \cap x_i \in R_m)
\]

\[
= \hat{P}_c (y = c_m \mid x_i \in R_m)
\]
Squared error risk (ave. loss) on learning sample for entire tree

\[ \hat{e}_M = \sum_{m=1}^{M} \hat{e}_m \]

\[ \hat{e}_m = \frac{1}{N} \sum_{k=1}^{K} \sum_{x_i \in R_m} \left[ I(y_i = c_k) - \hat{\beta}_{km} \right]^2 \]

\[ \frac{\hat{\beta}_{km}}{N_m / N} \]

Contribution of \( R_m \) to overall risk of entire tree

\[ = \frac{N_m}{N} \sum_{k=1}^{K} \frac{1}{N_m} \sum_{x_i \in R_m} \left[ I(y_i = c_k) - \hat{\beta}_{km} \right]^2 \]

\[ \text{mean } I(y_i = c_k) \]

\[ = \frac{N_m}{N} \sum_{k=1}^{K} \hat{\beta}_{km} \left( 1 - \hat{\beta}_{km} \right) \text{ binary variables} \]

\[ = \Pr(x \in R_m) \sum_{k=1}^{K} \Pr(y = c_k | x \in R_m) \]

\[ \cdot \left( 1 - \Pr(y = c_k | x \in R_m) \right) \]

\[ = \Pr(x \in R_m) \times \text{variance ("diversity") of sample labels in } R_m \]
"Gimi" index of diversity

Given a collection of objects (Ball in urn) each with a label \( c_u \): let \( P_{c_u} = \frac{N_{c_u}}{N} \) be the probability of obj having label \( c_u \)

Gimi index of diversity of the objects

\[
G(P_1, P_2, \ldots, P_K) = \sum_{k=1}^{K} P_{c_k} (1 - P_{c_k}) = 1 - \sum_{k=1}^{K} P_{c_k}^2
\]

\[
G(1/K, \ldots, 1/K) = 1 - \frac{1}{K} = \text{max diversity} = \text{min purity}
\]

\[
G(0, 0, \ldots, 1, 0, \ldots, 0) = 0 = \text{min diversity} = \text{max purity}
\]
Consider split $R_m \rightarrow R_m^{(e)} \cup R_m^{(a)}$

**Before split:**

$$\hat{e}_M = \sum_{m=1}^{M} \hat{e}_m$$

- contribution of $R_m$

**After split:**

$$\hat{e}_{M+1} = \sum_{m' \neq m} \hat{e}_{m'} + \hat{e}_{ml} + \hat{e}_{mr}$$

- left
- right
- daughter regions

**Improvement due to split:**

$$\hat{I}_m(j, s_{jm}) = \hat{e}_m - \hat{e}_{M+1}$$

$$= \hat{e}_m - \hat{e}_{ml} - \hat{e}_{mr}$$

Maximize over all $m, j, s_{jm}$ to get next split.

$$\hat{I}_m(j, s_{jm}) = \hat{P}_l(x \in R_m) \text{ Gini}(R_m)$$

- $\hat{P}_l(x \in R_m^{(e)}) \text{ Gini}(R_m^{(e)})$

- $\hat{P}_l(x \in R_m^{(a)}) \text{ Gini}(R_m^{(a)})$
For each (potential) parent R_m, find a split that minimizes weighted diversity (or maximizes weighted purity) of daughters.

\[ \hat{I}_m = \max_{\text{split}} \hat{I}_m = \max_{\text{split}} \hat{\lambda} = \max_{\text{split}} \]

\[ \text{first split } \quad \text{(next split)} \quad \text{(no next split)} \]

Splitting based on diversity (purity) index attempts to purify class labels.

Define: \( H^{(z)} = -\sum_{k=1}^{K} \hat{p}_k \log \hat{p}_k \quad (y = c_k \mid x \in R) \)

"Second order entropy"

\[ \hat{I}(d, a) = \hat{p}_ctr (x \in R_d) H_m^{(z)} - \hat{p}_ct (x \in R_{d'}^{(z)}) H_m^{(z)} - \hat{p}_ct (x \in R_{d''}^{(z)}) H_m^{(z)} \]

Maximize improvement \( \Rightarrow \) minimize weighted entropy in daughters.
Gini index ($H^{(2)}$) not the only diversity measure

Ordinary entropy:

$$H = - \sum_{e=1}^{K} p_e \log p_e$$

Any superlinear function of $\sum_{e=1}^{K} p_e$, will work as an impurity measure.

Remember $\sum_{e=1}^{K} p_e = 1$

**Splitting strategy:**

1. Define purity measure for $R_m$
2. Define goodness-of-split as difference between purity of parent and weighted sum of purity of daughters
3. Find split that maximizes goodness-of-split over all $(m, j, \Delta j)$

**Problem: Categorical variables**

- Exhaustive search $\# \text{values} \leq 13$
- Heuristic (clustering) $\# > 13$

Sorting trick doesn't work except for $K=2$. 
CART Classification

(1) Top down splitting using criterion to purify regions

(2) terminal nodes $\{R_m\}_{m=1}^M$ have

$$\hat{P}_m = \hat{P}_m (y=c|x \in R_m) = N_{c|m} / N_{m} \quad (k=1, \ldots, K)$$

(3) Prediction rule:

$$x \in R_m \Rightarrow \hat{c} (x) = \arg \min_{1 \leq k \leq K} \sum_{l=1}^K L_{kl} \hat{P}_{lm}$$

(4) bottom-up tree pruning same as regression, but use misclassification risk

$$\hat{L}_m = \sum_{l=1}^K L_{kl} \hat{c}_m (x) \hat{P}_{lm}$$

to define node cost

$$C_m = \hat{L}_m + \lambda \quad \text{cost + complexity parameter}$$
TREE: ADVANTAGES

(1) Relatively fast: \( \text{comp} \lesssim nN \log N \) (worst \( N^2 \))

(2) All types of variables:
   numeric, binary, categorical, missing values

(3) Invariant under monotone xforms: \( x_j \leftarrow g_j(x_j) \)
   no issue of selecting xforms
   immunity to \( x_j \) outliers
   variable scales irrelevant
TREE: ADVANTAGES (cont.)

(4) Resistance to many irrelevant variables

(5) Few tunable parameters

"off-the-shelf" procedure

(6) Interpretable model representation

binary tree graphic
Tree disadvantage: inaccuracy

Problems with decision trees:

1. Discontinuous piecewise constant model

\[ F(x) \]

Big error (bias) near region boundaries

2. \( \hat{F}(x) \) must involve high order interactions

\[ \hat{F}(x) = \sum_{\ell=1}^{K} \hat{c}_{m} \hat{f}_{\mu}(x_{\mu}) \]

\( f_{\mu}(x_{\mu}) = I(x_{\mu} \in \text{stem}) \)

To enter \( X_j \) into model, must split on it \( \Rightarrow \) enter multiplicatively

Not good for low interaction \( F^*(x) \)

\[ F^*(x) = a_0 + \sum_{j=1}^{m} a_j x_j \quad \text{linear} \]

\[ = \sum_{j=1}^{m} f_j^*(x_j) \quad \text{additive} \]
(3) hyper-rectangular regions

decision boundary
(smooth)

(4) data fragmentation (partition search strategy)

Each split reduces training data for subsequent splits.

Introduce dependence on \( x_j \) \( \Rightarrow \) split (usually several \( \rightarrow \) many times)

So cannot model local dep. on more than a few vars. (run out of data)

Not good for \( F(x) \) that has dep. on many variables.

Above problems cause bias in tree based estimates.
(5) **HIGH VARIANCE** (recently recognised)

Caused by greedy search strategy (local optima)

Each split conditioned on all of its ancestor splits =>

Errors (variance) in each upper split are propagated down to affect all splits below it (progeny)

\[
\hat{F}(x) = \prod_{l \in \text{path from root to } R_m} I(x_{j,e} \in \mathcal{A}_{j,e})
\]

Errors in estimating \((j,e), \mathcal{A}_{j,e}\) multiply

Small changes in data (sampling) => **big change in tree**

=> trees very unstable

=> **high variance** (sampling fluctuations)

=> **big error + interpretation questionable**
What to do?

(1) live with problems
(2) use other methods (when possible)
(3) fix-up trees
   (a) maintain advantages
   (b) dramatically improve accuracy

Recent research (next few lectures)
- Bagging
- Boosting
- MARS (multivariate adaptive regression splines)