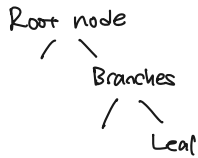


Chapter 9 Trees



- CART: Classification and Regression Trees

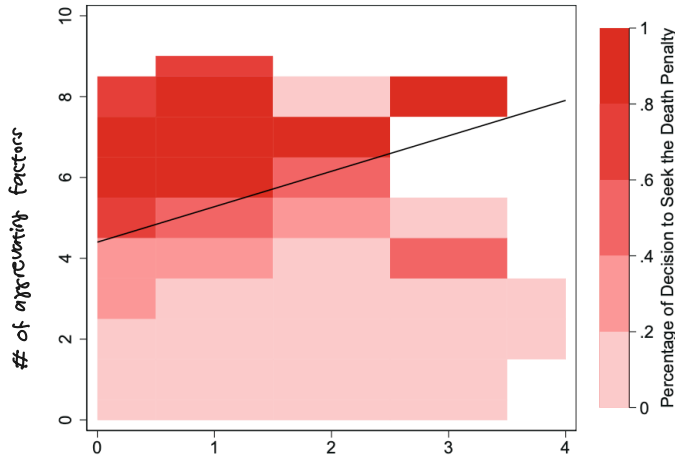


Fig. 9.3 Death penalty raw data: The percentage of seek-decisions as a function of the number of aggravating and mitigating factors. The diagonal line corresponds to the classification boundary based on logistic regression

of mitigating factors

← white spaces = without data.

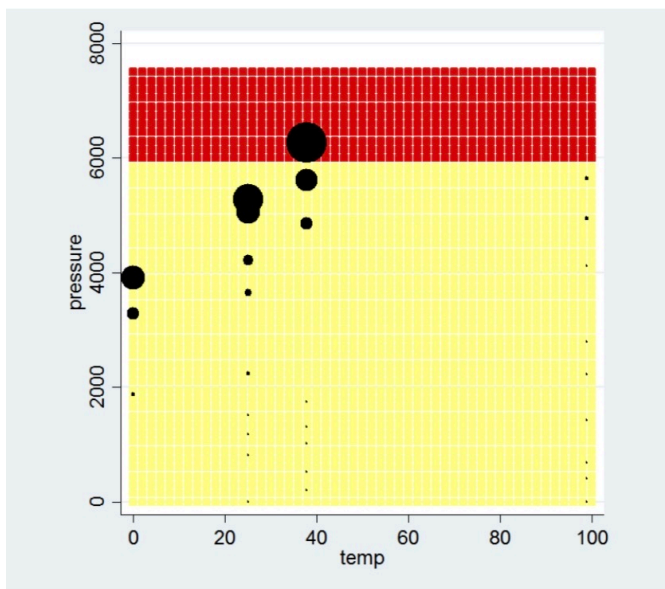
Diagonal line = classification or decision boundary ($p=0.5$)

based on logistic regression

Represented as a line since logistic

regression is a GLM model.

- In regression trees, best split is chosen across all variables and all values for each variable such that the residual sums of square is minimized



← Same predicted value by colors

Divided on: pressure < 5957

Model and Fitting Strategies

M = # of leaves (hyper-rectangles)

C_m = constant predicted value in leaf m (or, hyper-rectangle R_m)

$$\hat{y} = \sum_{m=1}^M C_m I(x \in R_m)$$

Indicator function to identify which hyper-rectangle applies

↳ whether or not a test observation falls into that leaf

- At each interior node, decision rule creates two child nodes.

x = p -dimensional vector

- At each step, split an existing region into two parts.

↳ Recursive binary splitting

Regression Trees

Finding the optimal splits for regression trees.

- Prediction: (Average value of all observations in that leaf)

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$$

- Find the split that minimizes the residual sum of squares (RSS)

$$RSS = \sum_{i=1}^n (y_i - f(x_i))^2$$

↳ $f(x_i) = \text{Constant within each leaf or region}$

We try to minimize the squared error loss

Proposed split $x_j \leq s$, x_j is j th variable and s is the split value divides the existing region into 2

$$\text{Maximize RSS reduction in any one split: } \arg\max_{m,j,s} \left[\sum_{i: x_i \in R} (y_i - \hat{c})^2 - \left(\sum_{i: x_i \in R_{m1}} (y_i - \hat{c}_1)^2 + \sum_{i: x_i \in R_{m2}} (y_i - \hat{c}_2)^2 \right) \right]$$

Each region, R_m , $m \in \{1 \dots T\}$ is split into two regions: R_{m1} and R_{m2}

$$\text{Split} \rightarrow R_1(j, s) = \{x | x_j \leq s\}$$

$$R_2(j, s) = \{x | x_j > s\}$$

ex

y	x
0	3
-2	4
4	4
3	5
3	6

If $s = 3.5$ (threshold) and x_j is a feature.

$\hat{c}_1$ is the best constant prediction (average of y values where $c_1 = \text{mean}(y \text{ in } R_1)$).

Algorithm happens in a way: ① Try every possible split (j, s)

② Compute RSS for left + right

③ Choose the split with the smallest RSS

Data points: $(3, 0)$, $(4, -2)$, $(4, 4)$, $(5, 3)$, $(6, 3)$. Sorted by x already

For each split s : ① Left region $R_1: x \leq s$

② Right region $R_2: x > s$

③ Prediction in each region = mean of y

$$\textcircled{4} RSS = \sum (y - \text{mean})^2$$

(2b) Split at $s = 3.5$

Left = [0]
Right = [-2 + 4 + 3 + 3]

$C_1 = \frac{0}{4} = 0$
 $C_2 = \frac{8}{4} = 2$

$$RSS = \text{left} \rightarrow (0-0)^2 = 0$$

$$\text{right} \rightarrow (-2-2)^2 + (4-2)^2 + (3-2)^2 + (3-2)^2 = 16 + 4 + 1 + 1 = 22$$

Thus, total RSS = 22

Split at 9.5 Left $\rightarrow [0, -2, 4]$ $C_1 = \frac{2}{3}$
Right $\rightarrow [3, 8]$ $C_2 = \frac{6}{2} = 3$

$$\begin{aligned} \text{RSS} \rightarrow \text{left: } & (0 - \frac{2}{3})^2 + (-2 - \frac{2}{3})^2 + (4 - \frac{2}{3})^2 \\ & = \frac{4}{9} + (\frac{8}{3})^2 + (\frac{10}{3})^2 = \frac{4}{9} + \frac{64}{9} + \frac{100}{9} = \frac{168}{9} = 18.666... \end{aligned}$$

Right: $(3-3)^2 + (3-3)^2 = 0$

S with RSS of 18.66 has the smallest RSS

Reduction in SS is: $25.2 - 18.66 = 6.54$

We need reductions to compute the reductions in sums of squares (SS)

Thus, $\frac{0-2+4+3+3}{5} = \frac{8}{5}$.

And $(0 - \frac{8}{5})^2 + (-2 - \frac{8}{5})^2 + (4 - \frac{1}{5})^2 + (3 - \frac{8}{5})^2 + (3 - \frac{1}{5})^2 = 25.2$

This split removes $25.22 - 18.66 = 6.54$ units of squared error.

Stopping Criterion

- When to stop growing to avoid overfitting
- Minimum observations: Do not split if a tree produces a leaf with less than fewer observations

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + a|T| \quad ; \quad a \geq 0, \quad M = \text{indexer leaves}, \quad i = \text{indexes obs within a given leaf}$$

$|T|$ = # of leaves, $\rightarrow \lambda$ is tuned with CV
 $\alpha|T|$ = penalty for having too many leaves

↳ This is regularization, controls bias/variance trade-off

Larger tree has less bias but more variance

smaller tree has more bias but less variance

Ideally, we want them both low

Bias - error from model being too simple. Underfits

Variance - Model being too sensitive to training data

- Prediction: Predicted value of a given leaf is the average outcome for all observations in the leaf.

$$\hookrightarrow \hat{c}_m = \text{ave}(y_i \mid x_i \in R_m)$$

$\hookrightarrow$ By convention, average between any two points is used as a split value even though any value between the two points would result in the same split.

$$\hookrightarrow \text{RSS} = \sum_{i=1}^n (y_i - f(x_i))^2$$

$\nwarrow$ avg. in general

$$\text{argmax}_{m,j,s} \left[\sum_{i: x \in R_m} (y_i - \hat{c})^2 - \left(\sum_{i: x \in R_{m1}} (y_i - \hat{c}_1)^2 + \sum_{i: x \in R_{m2}} (y_i - \hat{c}_2)^2 \right) \right]$$

$\uparrow$ Region 1 $\uparrow$ Region 2 due to split

Splits with Categorical x-variables and Missing x-values

- Splitting on continuous x-variables only exploits their order of values and not their values

$\hookrightarrow$ ex ["high school", "some college", "college", "postgraduate"], $\rightarrow$ {1, 2, 3, 4}

As long as the order is maintained, doesn't matter what value is actually getting used.

- Indicator variables: Variables that only take 0 or 1.

$\hookrightarrow$ This is a special case of ordered variables, only allowing a single split

$\hookrightarrow$ The split point is always 0.5 given the indicator variable is coded as 0 and 1.

- Unordered categorical values: Since they are unordered, simple x:SS does not suffice.

$\hookrightarrow$ Requires splitting the set of variables into two subsets... there could be many ways to create unordered levels

- Missing x-values:

① Find a surrogate variable to split on

② Only pass the observation down the tree until a splitting rule requires a value that is missing and does not pass down further

③ May be passed to both child nodes and later assigned to a class with greater probability

④ Make the missingness as an indicator variable

Stopping and Pruning

- When to stop splitting trees into sub-trees

① Only when a split contains minimum set # of observations

② Split only if the split improves RSS by at least some threshold value

- Pruning: Build an overly large tree, and prune the tree later.

Reduce an existing tree by removing leaves that do not substantially improve classification or regression.

↳ Introduce penalty for large trees as a part of the criterion.

Modified RSS is: $\sum_{m=1}^T \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha T$

- T is the # of leaves, $\alpha \geq 0$ is the tuning parameter for regularization.

- αT : penalty for having too many leaves

In general, smaller trees are assumed to be more variable than large trees.

' α ' regulates this bias-variance tradeoff.

- Other models that use tree as a base learner: RF, gradient boosting, XG Boost

↳ also uses pruning

For Classification

- Estimated predicted value is no longer an average

- Criterion for splitting is no longer the residual sums of squares

- Prediction: Most frequently occurring class.

↳ Now estimate the probabilities for a given leaf

$$\text{ex) } P(\text{history} \mid \text{leaf}_m) = \frac{1}{2}$$

$$P(\text{Comedy} \mid \text{leaf}_m) = \frac{1}{3} \quad \text{leaf}_m = \text{Leaf in question.}$$

$$P(\text{Tragedy} \mid \text{leaf}_m) = \frac{1}{6} \quad \text{Largest frequency} = \text{Largest probability}$$

↳ Thus, $\hat{c}_m = \arg\max_k (\hat{p}_{mk})$, where $\hat{p}_{mk}$ = Estimated probability for category k in leaf m .

- Criterion for splitting:

$$\text{① Classification Error} \Rightarrow E = 1 - \max_k (\hat{p}_{mk})$$

$$\text{② Gini Index} \Rightarrow G = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk}) \quad \begin{array}{l} k = \text{indexes outcomes} \\ m = \text{indexes leaves} \end{array}$$

$$\text{③ Entropy (Deviance)} \Rightarrow D = - \sum_{k=1}^K \hat{p}_{mk} \log(\hat{p}_{mk})$$

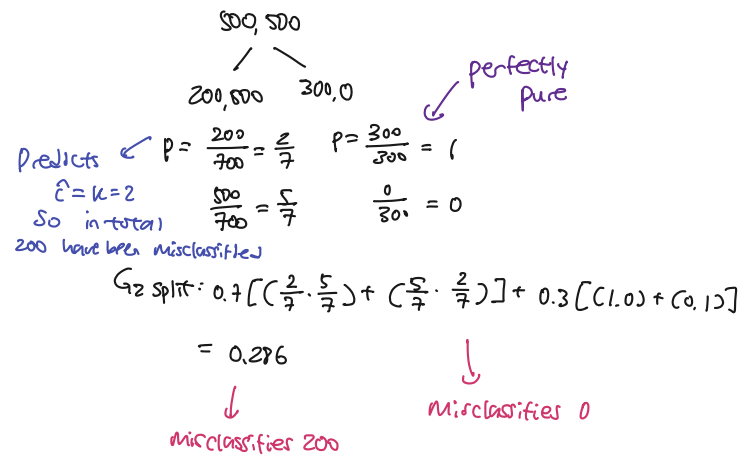
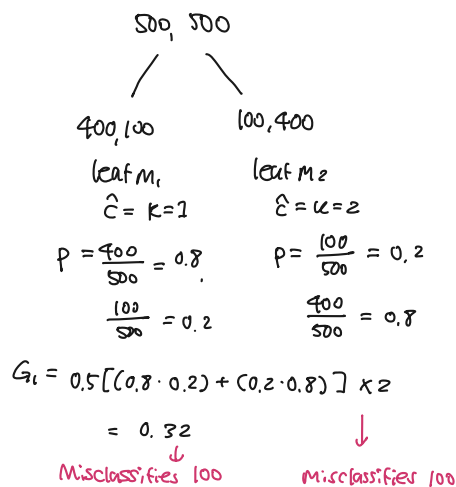
① Classification error = fraction of the training data incorrectly classified in that leaf.

② Gini criterion = measures the variance across K -classes

③ Entropy / Deviance = Measure of impurity

- Choose a split that minimizes the criterion

②



- Gini / Deviance indexes require classification over all classes, not just the predicted class

- Both Gini / Deviance take smaller values if the leaf is more pure