

Decision trees¹

Virgil Pavlu September 27, 2008

1 Supervised learning

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europa
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europa
bad	5	medium	medium	medium	medium	75to78	europa

40 Records

Figure 1: Toy set of records (UCI)

¹slides thanks to Carlos Guestrin@CMU

2 Univariate trees for classification

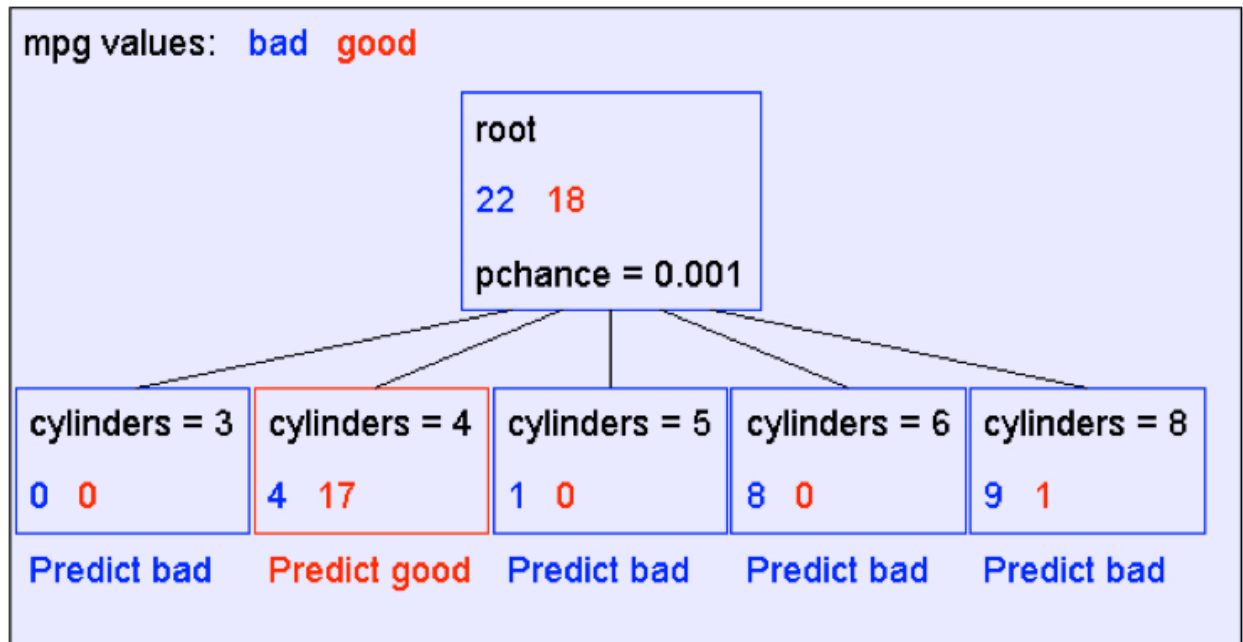
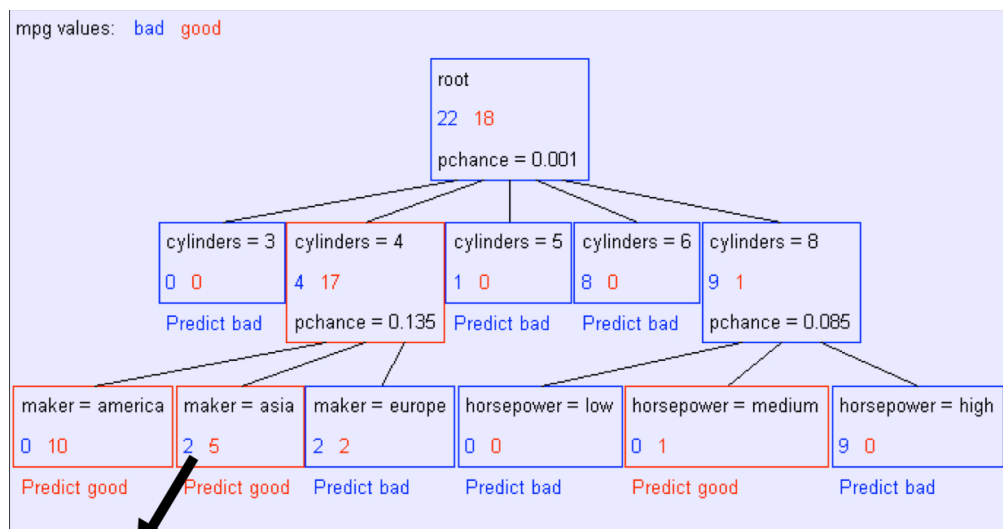


Figure 2: Decision tree, 1 layer.



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

Figure 3: Decision tree, 2 layers.

REMARK: not capable of classifying data not seen in training

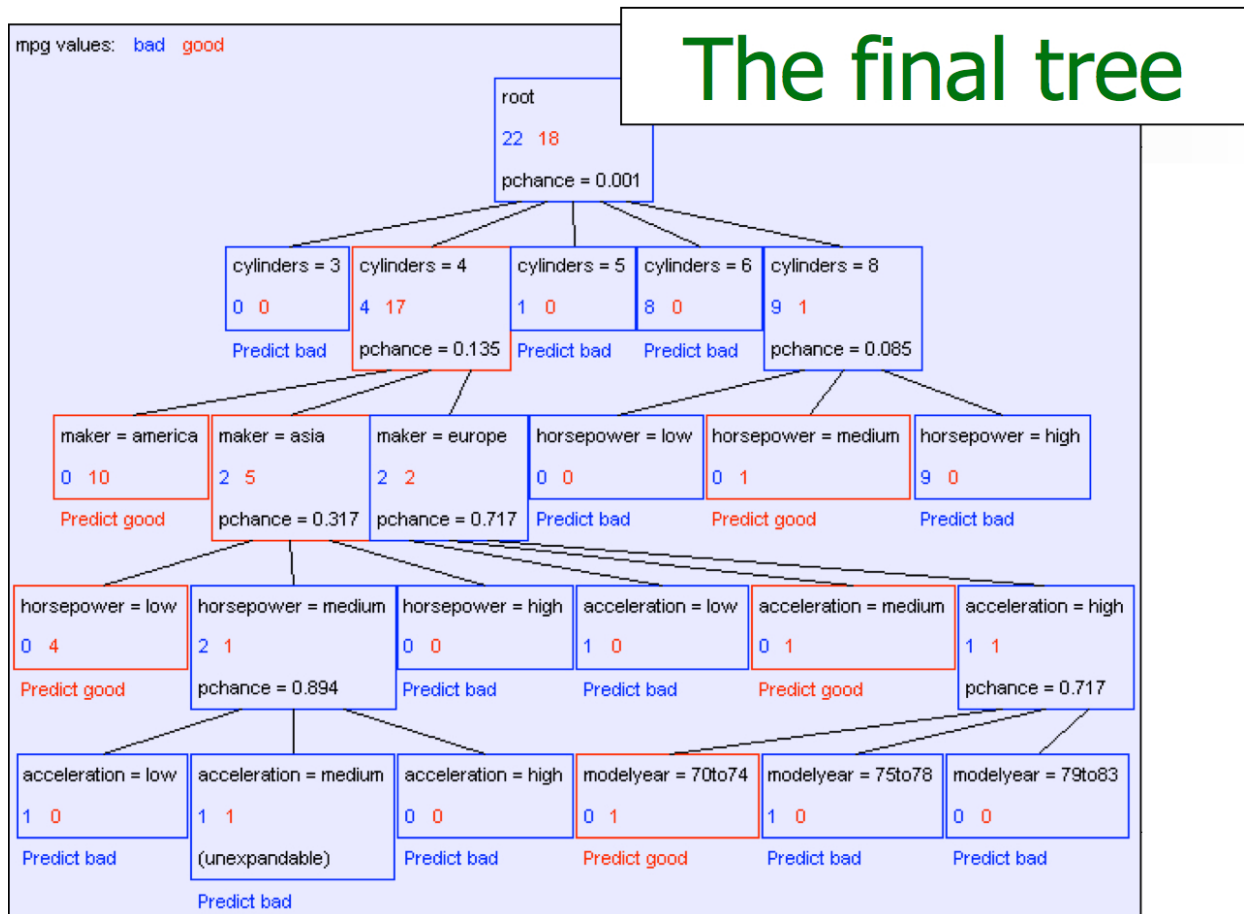


Figure 4: Decision tree, complete

3 Tree splitting

Finding the smallest decision tree is NP-complete. Use a heuristic:

- start with an empty decision tree
- split on the best feature.
- recurse

3.1 Entropy-based gain

$$H(Y) = \sum_j P(y_j) \log_2\left(\frac{1}{P(y_j)}\right)$$

Entropy after split by X feature

$$H(Y|X) = \sum_i P(x_i) \sum_j P(y_j|x_i) \log_2\left(\frac{1}{P(y_j|x_i)}\right)$$

Mutual information (or Information Gain).

$$IG(X) = H(Y) - H(Y|X)$$

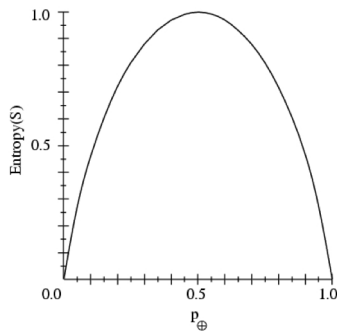


Figure 5: Entropy for 2-valued distribution

At each split we are going to choose the feature that gives the highest information gain.

x^1	x^2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Figure 6: 2 possible features to split by

$$H(Y|X^1) = \frac{1}{2}H(Y|X^1 = T) + \frac{1}{2}H(Y|X^1 = F) = 0 + \frac{1}{2}\left(\frac{1}{4}\log_2 \frac{1}{4} + \frac{3}{4}\log_2 \frac{3}{4}\right) \approx .405$$

$$IG(X^1) = H(Y) - H(Y|X^1) = .954 - .405 = .549$$

$$H(Y|X^2) = \frac{1}{2}H(Y|X^2 = T) + \frac{1}{2}H(Y|X^2 = F) = \frac{1}{2}\left(\frac{1}{4}\log_2 \frac{1}{4} + \frac{3}{4}\log_2 \frac{3}{4}\right) + \frac{1}{2}\left(\frac{1}{2}\log_2 \frac{1}{2} + \frac{1}{2}\log_2 \frac{1}{2}\right) \approx .905$$

$$IG(X^2) = H(Y) - H(Y|X^2) = .954 - .905 = .049$$

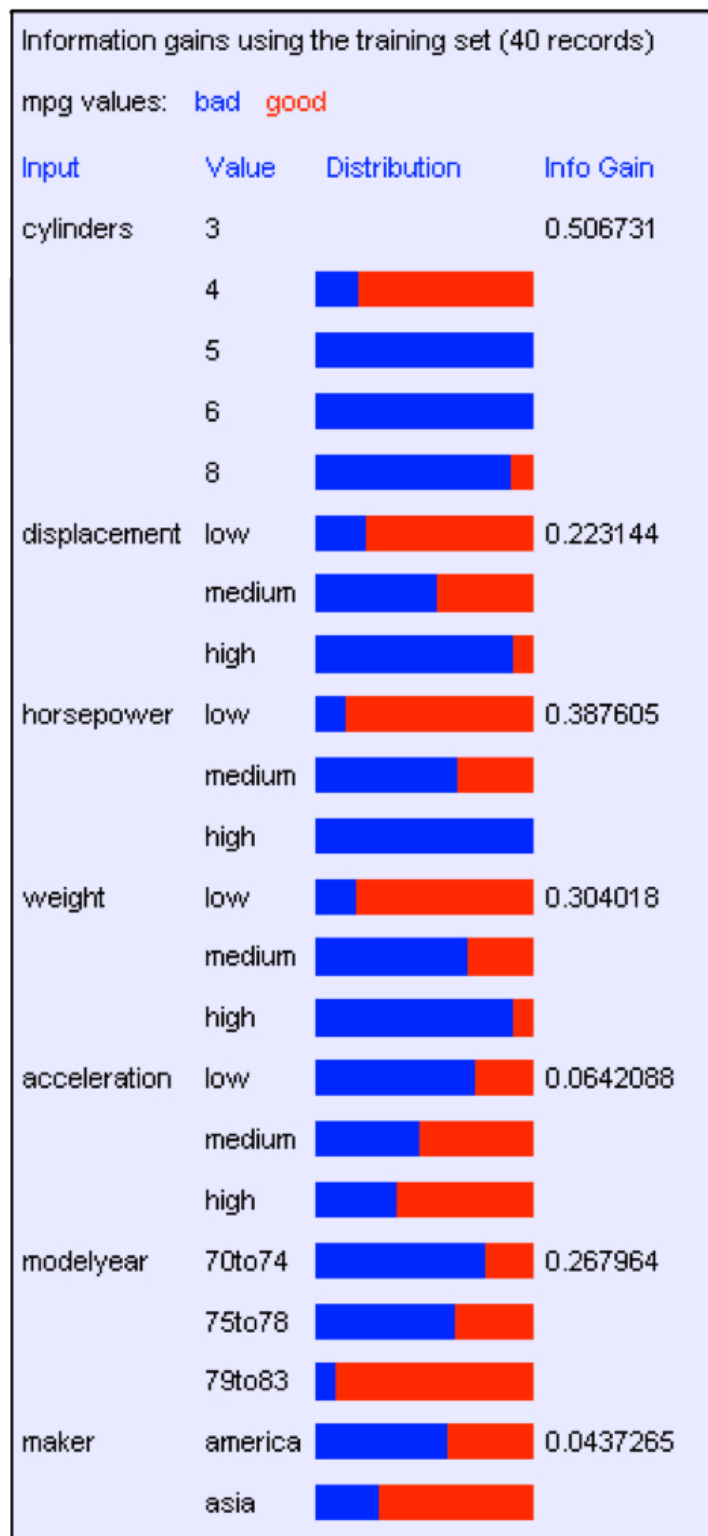


Figure 7: Information gain

4 When to stop splitting

- matching records have the same attribute value. REMARK: $H(Y) = 0$
- No attributes can further distinguish records. REMARK : $H(Y|X) = H(Y)$ for any feature X

4.0.1 0 information gain case in general

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

$y = a \text{ XOR } b$

The information gains:

Information gains using the training set (4 records)			
y values: 0 1			
Input	Value	Distribution	Info Gain
a	0		0
	1		0
b	0		0
	1		0

The resulting decision tree:

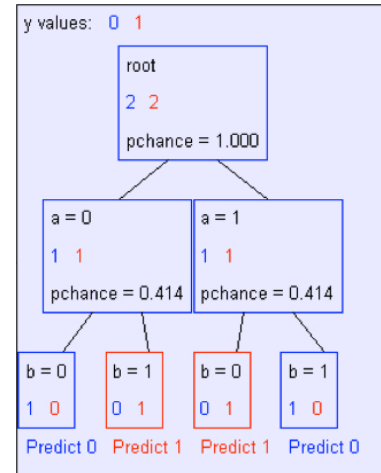
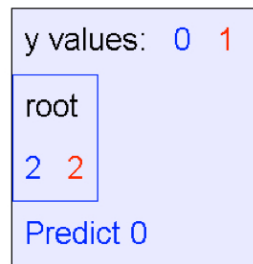


Figure 8: "Do not split" VS "split" when Information gain is 0

5 Real-valued inputs

If the input values of X are real (and/or continuous), then splitting branches by feature-values are not feasible. Instead find the best threshold t for the feature X .

$$H(Y|X : t) = P(X < t)H(Y|X < t) + P(X \geq t)H(Y|X \geq t)$$

$$IG(X : t) = H(Y) - H(Y|X : t)$$

Find t that maximizes $IG(X : t)$. To do so, a possibility is to consider all t of the form $(x_i + x_{i+1})/2$, where x_1, x_2, \dots, x_n are the values of feature X in the training set.

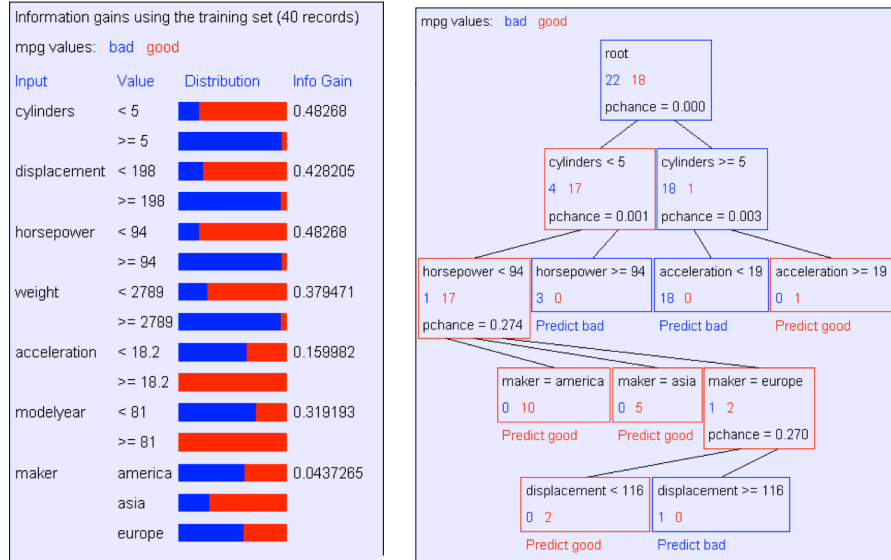


Figure 9: Real-valued information gain and decision tree

6 Regression trees

Lets say that fro each node m , χ_m is the set of datapoints reaching that node.

Estimate a predicted value per tree node

$$g_m = \frac{\sum_{t \in \chi_m} y_t}{|\chi_m|}$$

Calculate mean square error

$$E_m = \frac{\sum_{t \in \chi_m} (y_t - g_m)^2}{|\chi_m|}$$

How to choose the next split. If $E_m < \theta$, then stop splitting. Otherwise choose the split that realizes the maximum drop in error for all all brances. Say we are considering feature X with branches x_1, x_2, \dots, x_k , and lets call χ_{m_j} the subset of χ_m for which $X = x_j$.

$$g_{m_j} = \frac{\sum_{t \in \chi_{m_j}} y_t}{|\chi_{m_j}|}$$

$$E'_m(X) = \frac{\sum_j \sum_{t \in \chi_{m_j}} (y_t - g_{m_j})^2}{|\chi_m|}$$

We shall choose X such that $E'_m(X)$ is minimized, or the drop in error is maximized.

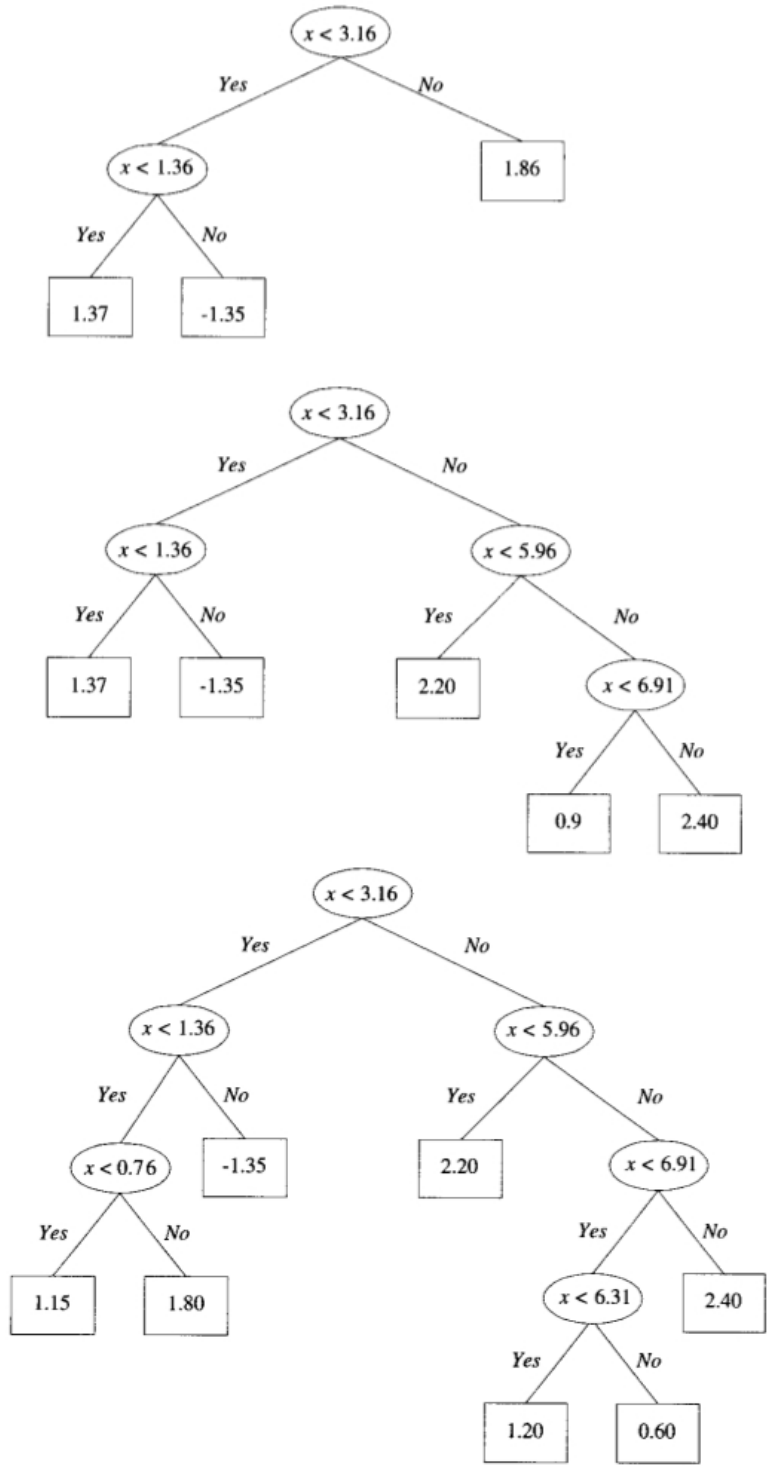


Figure 10: Regression tree

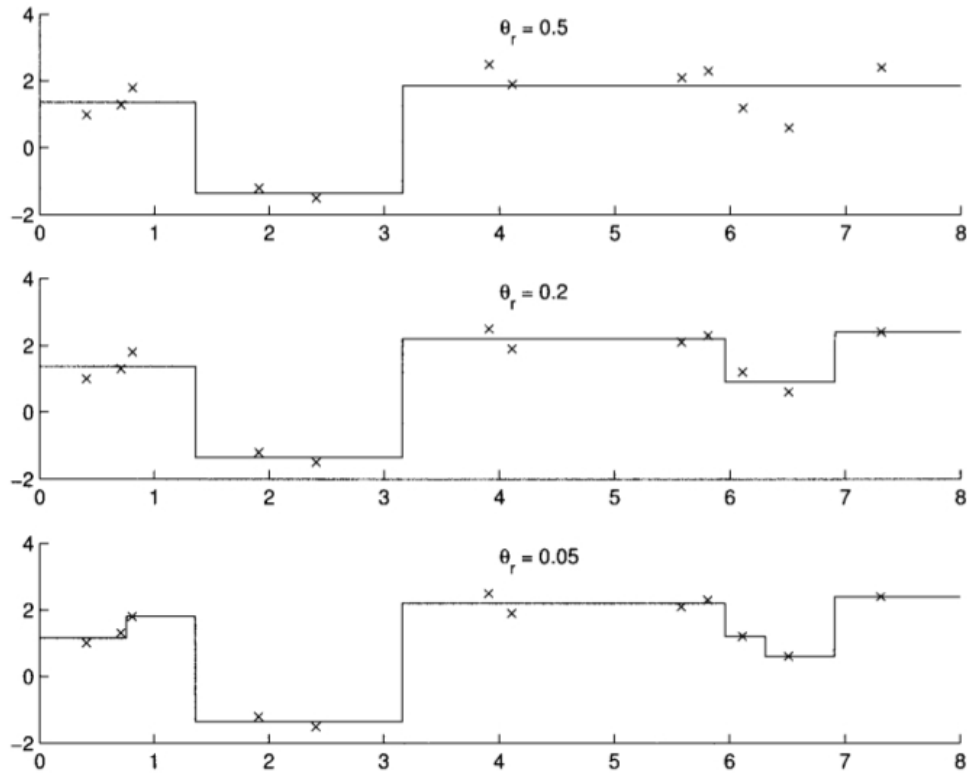


Figure 11: Regression fit

7 Pruning

If a tree is "too small", the model does not capture all structure of data, or it *underfits*. If the tree is too big, it captures structure that is too local and it cannot be generalized (*overfits*). Pruning helps heuristically to find the appropriate tree size.

Pre-pruning If a tree node contains less than, say, 5% of the training set, stop splitting (even if there are features with positive information gain).

Post-pruning Grow the tree until all positive information gains are used for splitting; then find the overfitting subtrees and merge them together. To do so, we need a pruning set (separate from testing or validation sets): if merging subtrees does not increase the classification error on the pruning set (by more than ϵ), then we merge the subtrees.

8 Rules extraction

Go over the branches of the tree and write down the splits. For example, for the tree in figure 9, some rules are:

IF (Cylinders<5) AND (horsepower<94) AND (maker= asia) THEN "predict good"

IF (cylinders>=5) AND (acceleration<19) THEN "predict bad"

....

Rules extraction directly from data. Also based on Information gain, but it traverses the data DFS instead of BFS.

9 Multivariate tree

In a multivariate tree, the splitting criteria can be a functional of more than one feature. For example, at the root we can have the following split:

$$\text{cylinders} * 20 + \text{horsepower} < 180$$

More generally, a binary linear multivariate node m split can look like

$$w^1x^1 + w^2x^2 + \dots w^dx^d + w^0 > 0$$

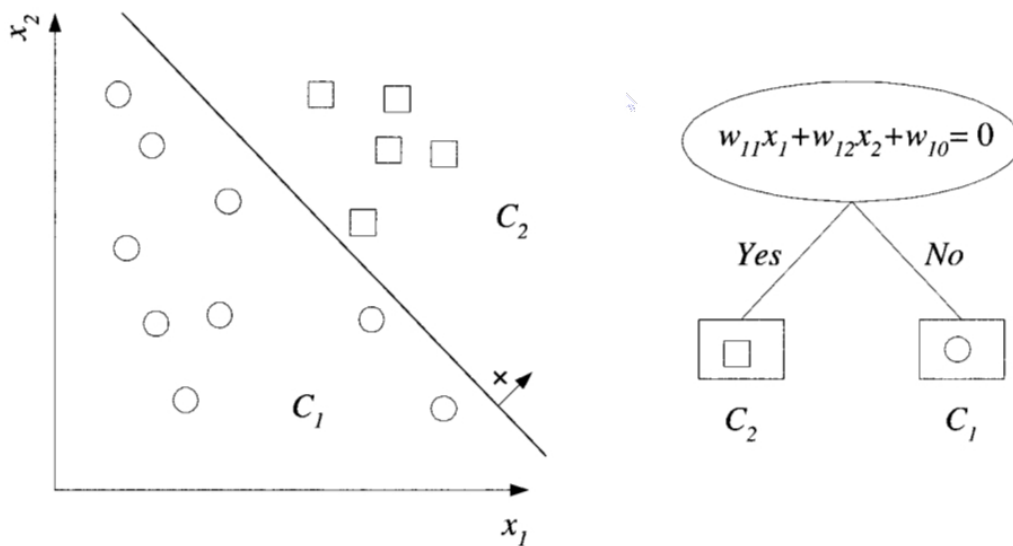


Figure 12: Multivariate split

Such splits can be extremely powerful (if data is linearly separable, a single split at root can create a perfect classification); even more complex splits can be obtained using nonlinear split functions.

However, finding a good multivariate split is not anymore a matter of brute force: there are $2^d \binom{N}{d}$ possible splits (or hyperplanes). Later on in the course we will discuss linear classification and how good hyperplanes can be obtained without an exhaustive search.