Decision trees[1]

Virgil Pavlu September 13, 2010

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	europe
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	europe
bad	5	medium	medium	medium	medium	75to78	europe

40 Records

Figure 1: Toy set of records (UCI)

[1]slides thanks to Carlos Guestrin@CMU

2 Univariate trees for classification

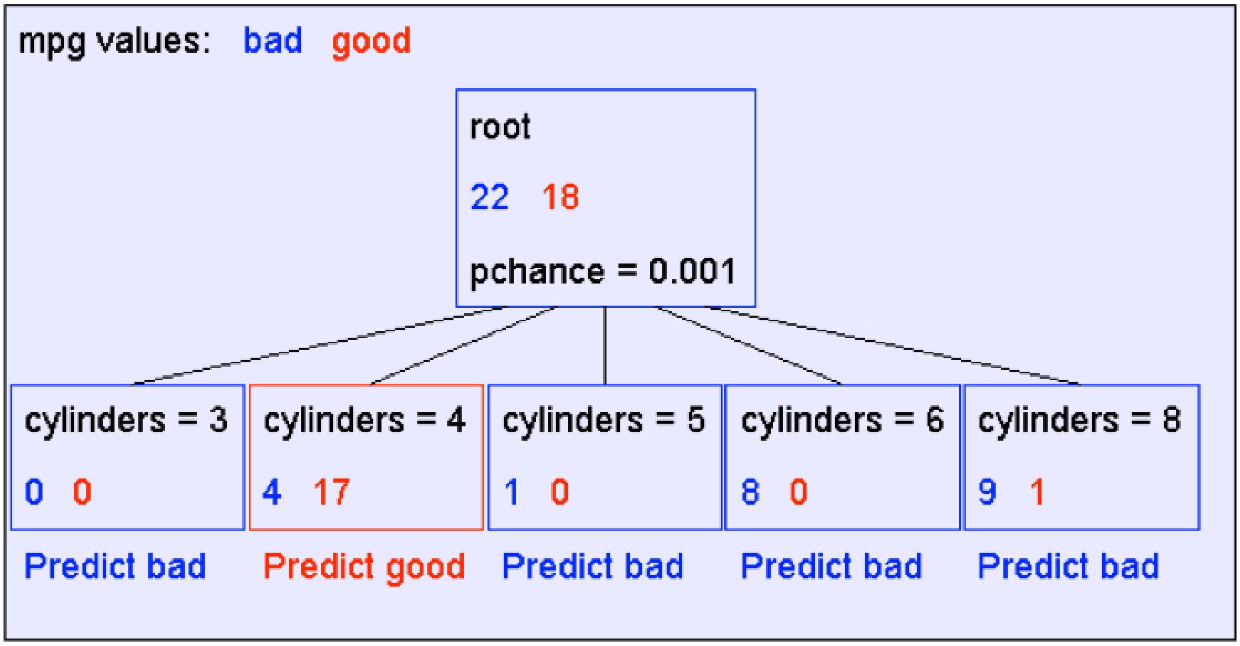

Figure 2: Decision tree, 1 layer.

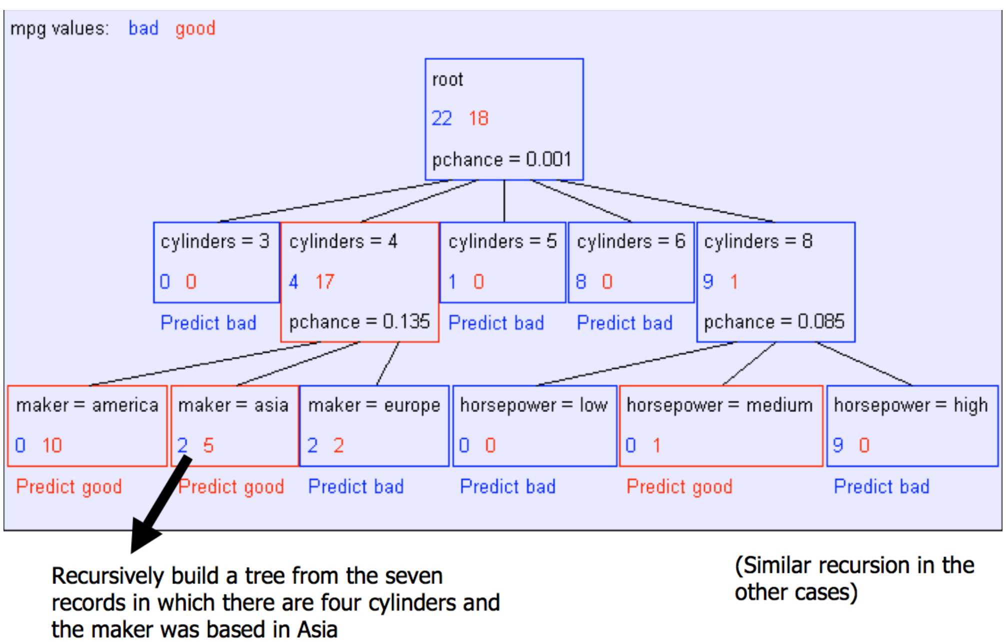

Figure 3: Decision tree, 2 layers.

REMARK: not capable of classifying data not seen in training

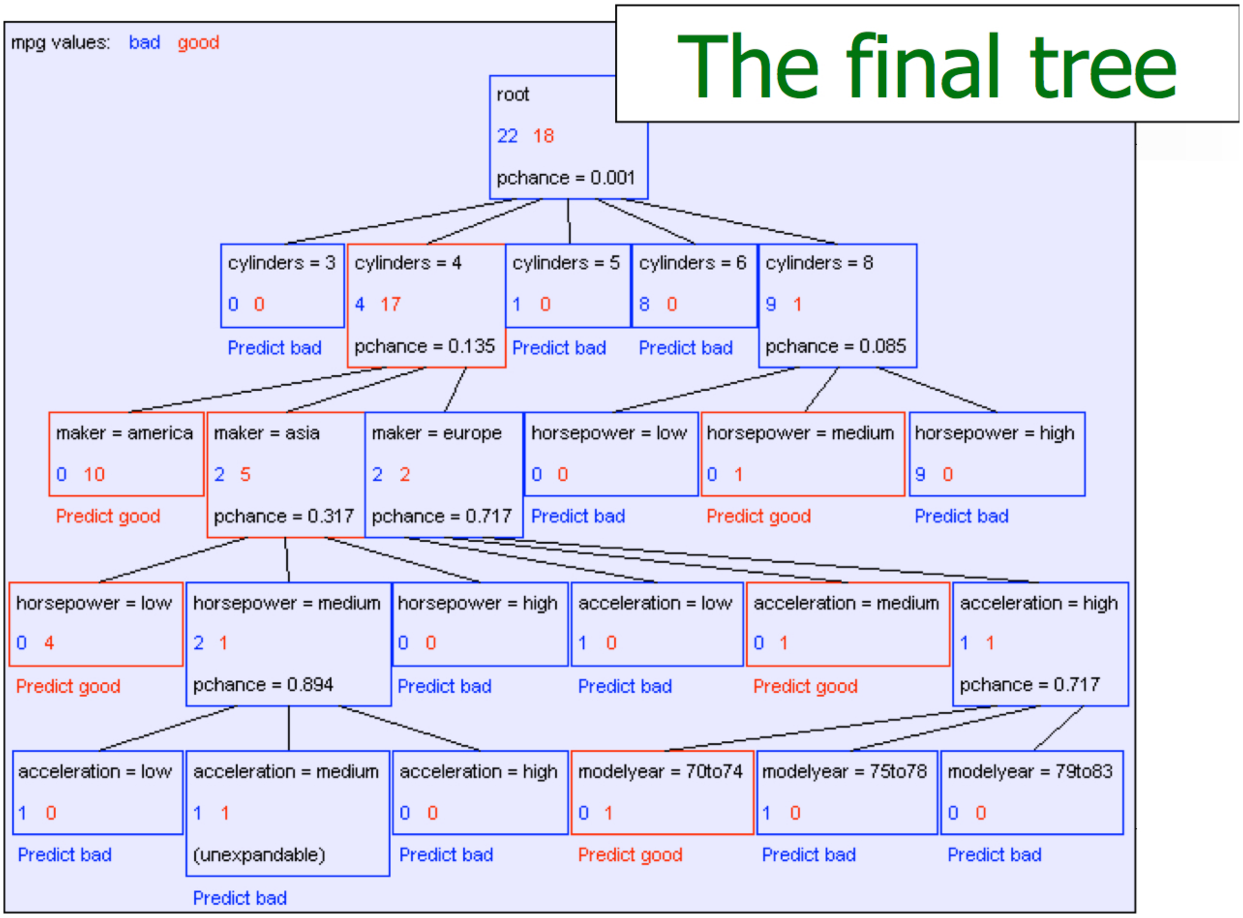

Figure 4: Decision tree, complete

3 Tree splitting

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

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

3.1 Entropy-based gain

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

Entropy after split by X feature

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

Mutual information (or Information Gain).

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

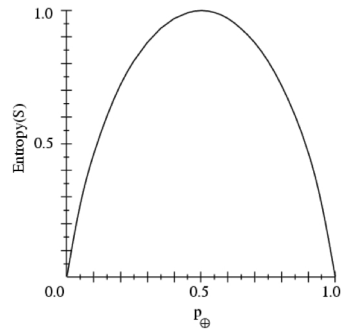

Figure 5: Entropy for 2-valued distribution

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

$$H(Y|X^1) = \frac{1}{2}H(Y|X^1 = T) + \frac{1}{2}H(Y|X^1 = F) = 0 + \frac{1}{2}(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}) \approx .405$$
$$IG(X^1) = H(Y) - H(Y|X^1) = .954 - .405 = .549$$

$$H(Y|X^2) = \frac{1}{2}H(Y|X^2 = T) + \frac{1}{2}H(Y|X^2 = F) = \frac{1}{2}(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}) + \frac{1}{2}(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}) \approx .905$$
$$IG(X^2) = H(Y) - H(Y|X^2) = .954 - .905 = .049$$

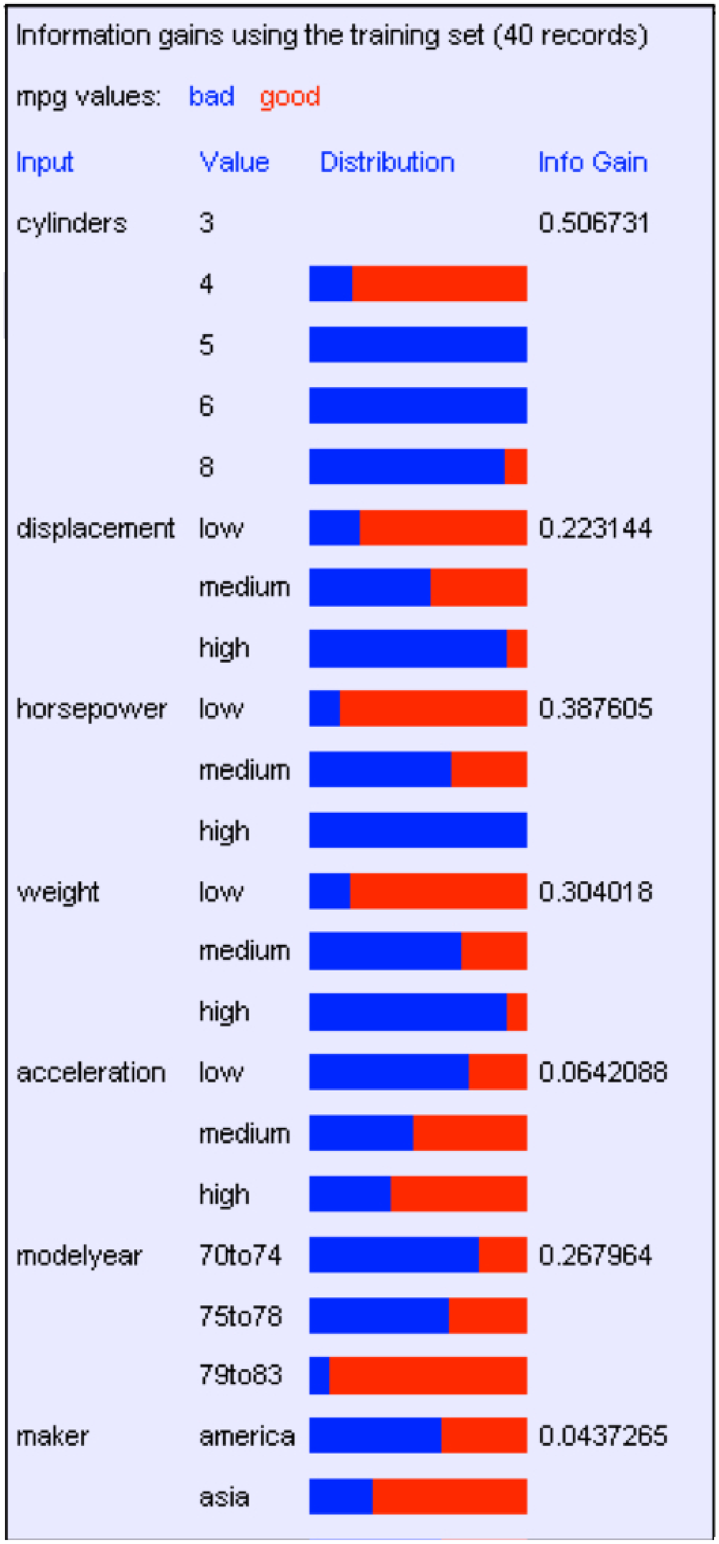

Figure 7: Information gain

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

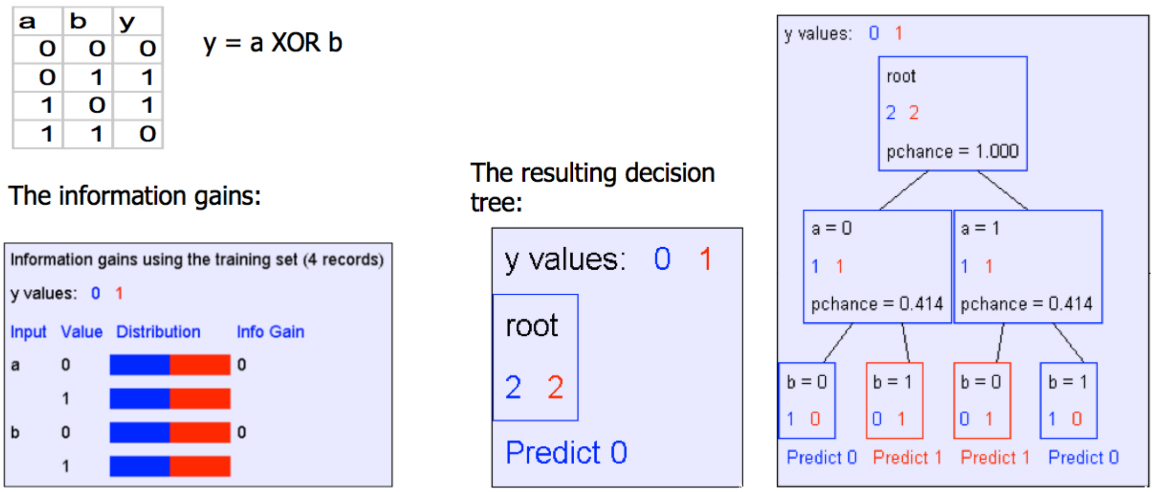

Figure 8: "Do not split" VS "split" when Information gain is 0

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 $x_1, x_2, ..., x_n$ are the values of feature X in the training set.

6 Multiway splits

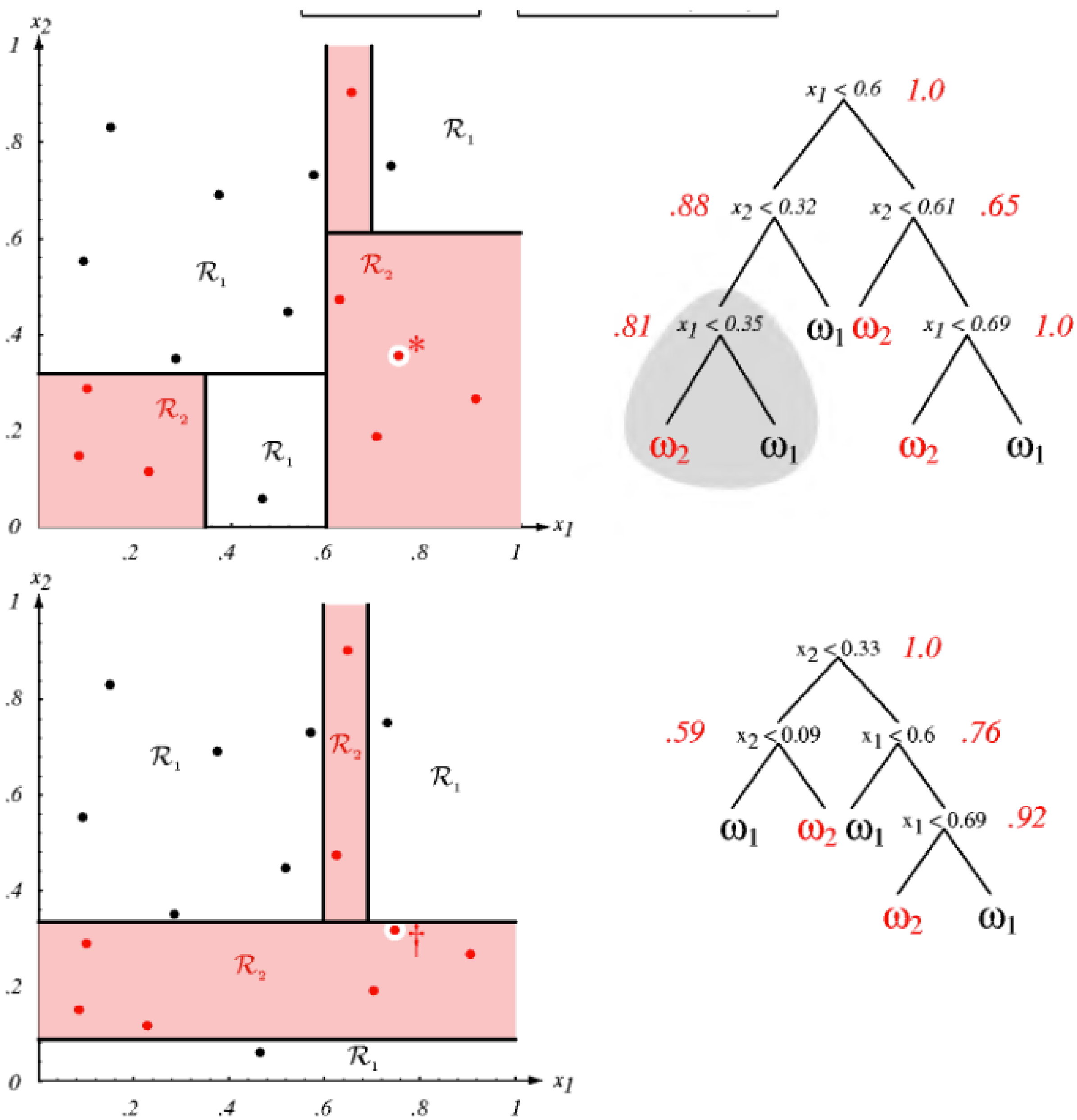

Figure 9: Real-valued information gain and decision tree

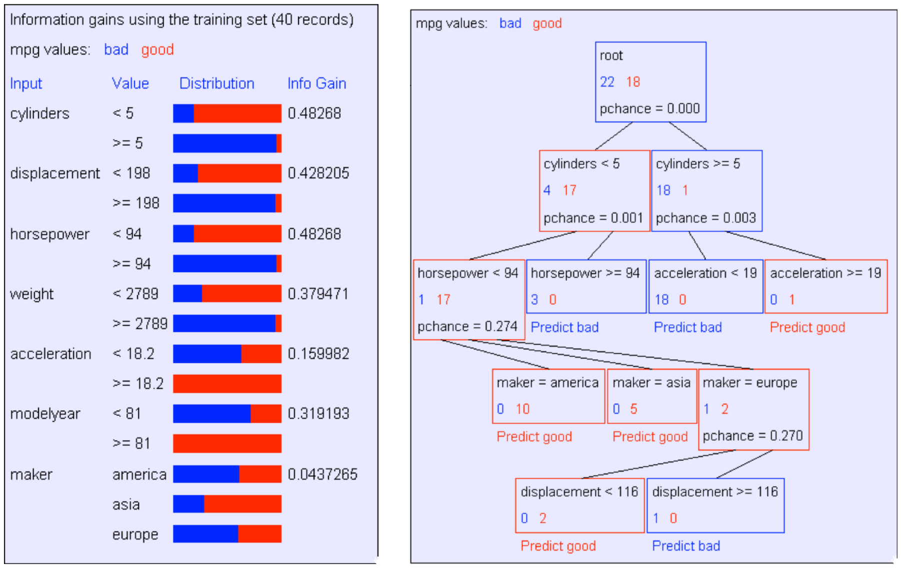

Figure 10: Real-valued information gain and decision tree

7 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 $x_1, x_2, ..., x_k$, and lets call χ_{mj} the subset of χ_m for which $X = x_j$.

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

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

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

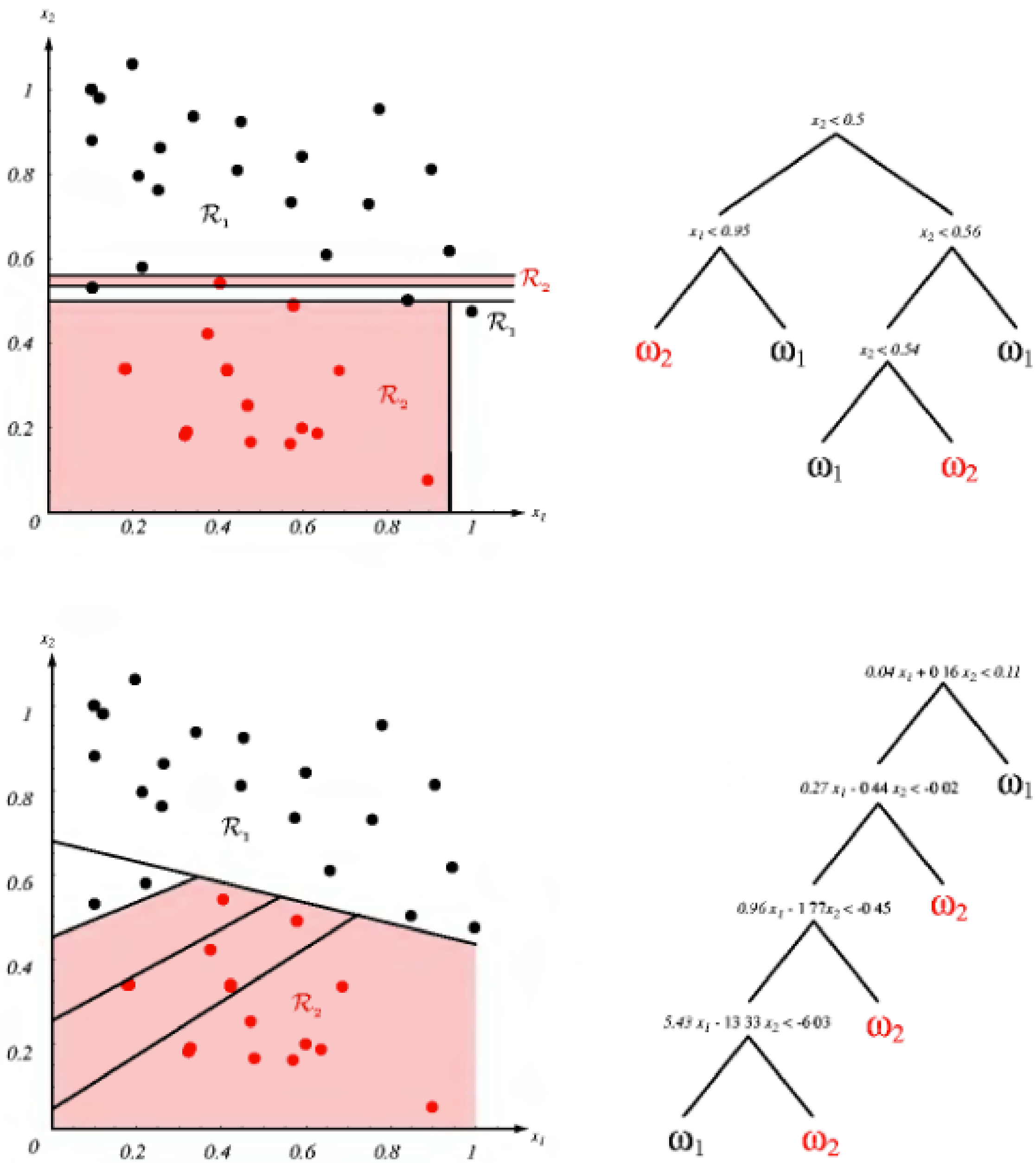

Figure 11: Multiway splits

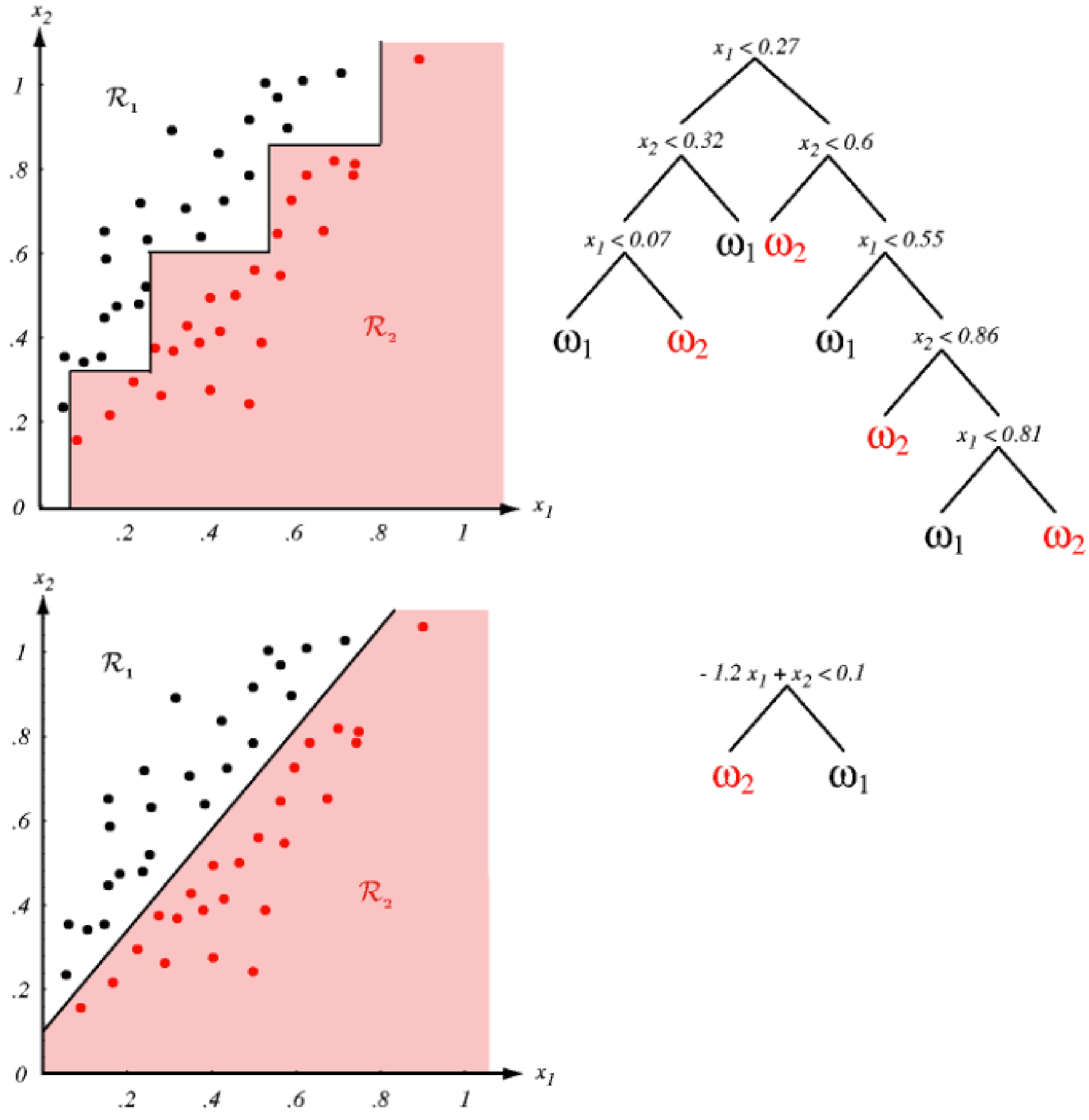

Figure 12: Multiway splits

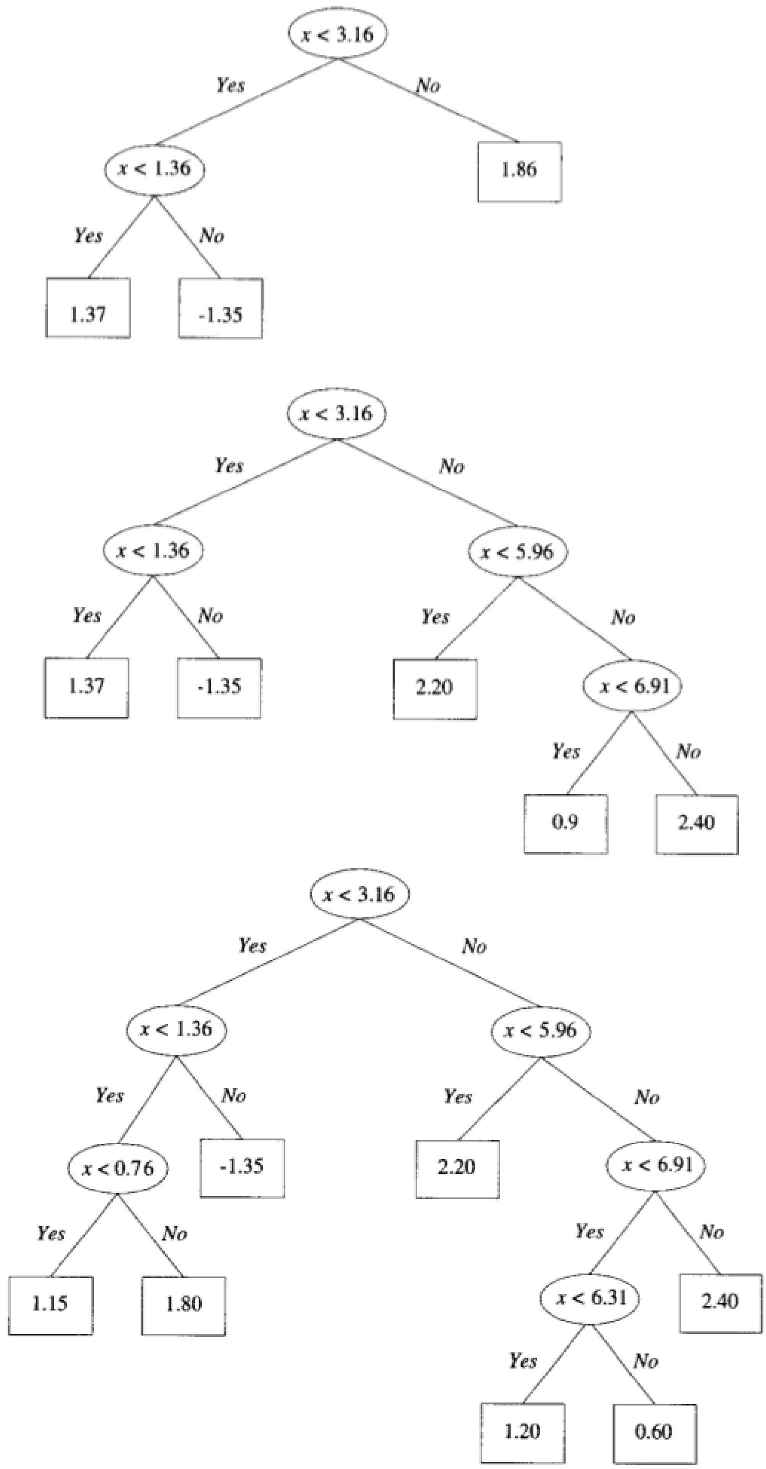

Figure 13: Regression tree

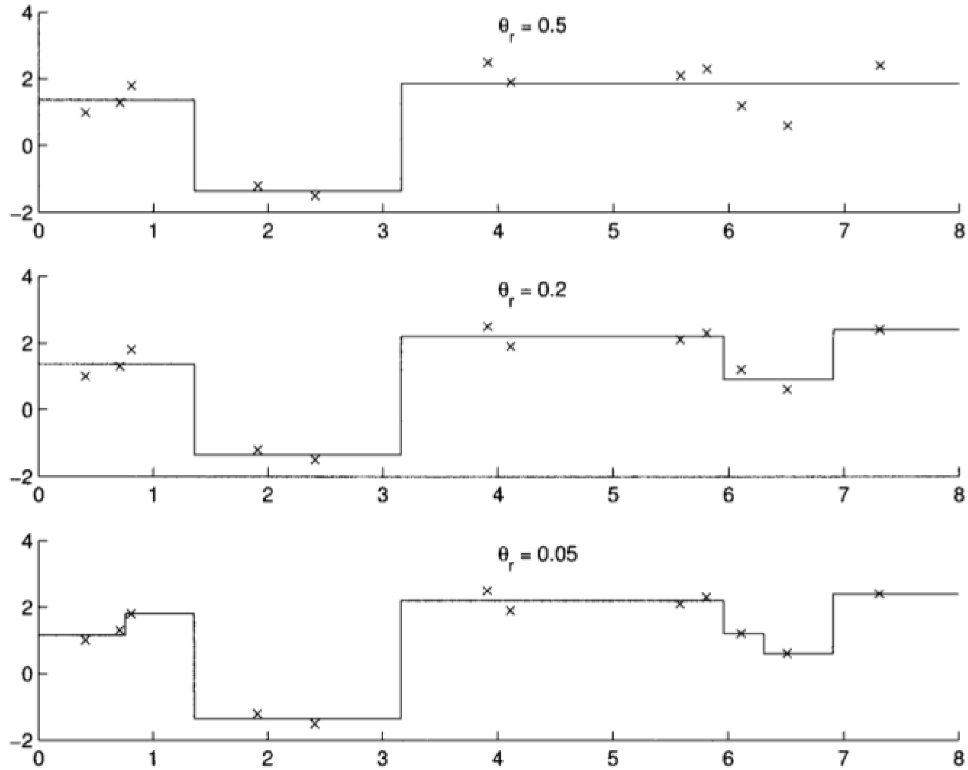

Figure 14: Regression fit

8 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 that, 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.

9 Rules extraction

Go over the branches of the tree and write down the splits. For example, for the tree in figure 10, 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.

10 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:

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

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

$$w^1x^1 + w^2x^2 + ...w^dx^d + w^0 > 0$$

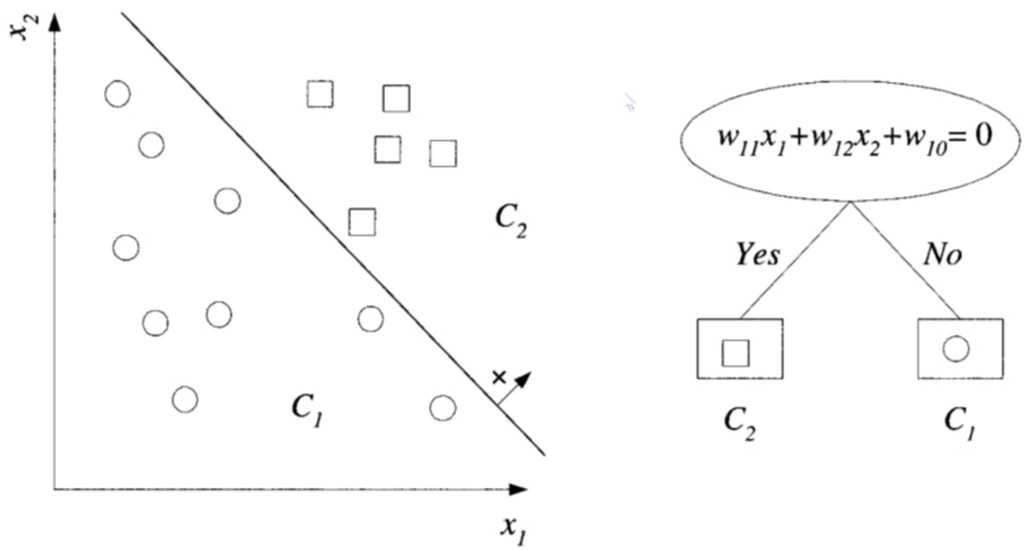

Figure 15: Multivariate split

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.