Decision Trees

Machine Learning – CSE546
Carlos Guestrin (by Sameer Singh)
University of Washington
October 16, 2014

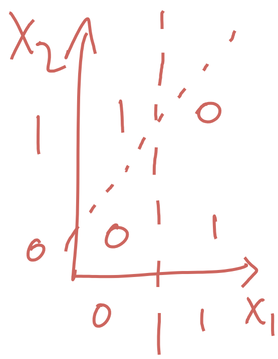
Linear separability

- A dataset is **linearly separable** iff there exists a **separating hyperplane**:
  - Exists **w**, such that:
    - $w_0 + \sum_i w_i x_i > 0$; if $\mathbf{x}=\{x_1,\ldots,x_k\}$ is a positive example
    - $w_0 + \sum_i w_i x_i < 0$; if $\mathbf{x}=\{x_1,\ldots,x_k\}$ is a negative example

$w_0 + w_1 x_1 + w_2 x_2$

Not linearly separable data

- Some datasets are **not linearly separable!**

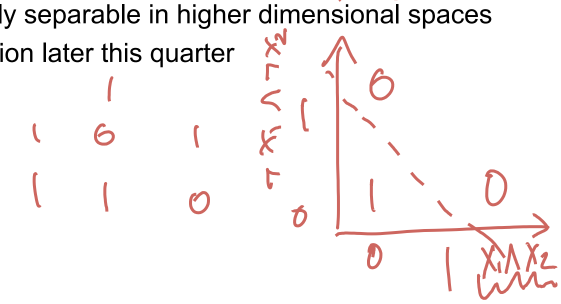


X_1	X_2	$Y = X_1 \text{ XOR } X_2$
0	0	0
0	1	1
1	0	1
1	1	0

Addressing non-linearly separable data – Option 1, non-linear features

- Choose non-linear features, e.g.,
  - Typical linear features: $w_0 + \sum_i w_i x_i$
  - Example of non-linear features:
    - Degree 2 polynomials, $w_0 + \sum_i w_i x_i + \sum_{ij} w_{ij} x_i x_j$
- Classifier $h_{\mathbf{w}}(\mathbf{x})$ still linear in parameters $\mathbf{w}$
  - As easy to learn
  - Data is linearly separable in higher dimensional spaces
  - More discussion later this quarter

$X_1 \wedge X_2$ $\neg X_1 \wedge \neg X_2$



Addressing non-linearly separable data – Option 2, non-linear classifier

- Choose a classifier $h_{\mathbf{w}}(\mathbf{x})$ that is non-linear in parameters **w**, e.g.,
  - Decision trees, boosting, nearest neighbor, neural networks…
- More general than linear classifiers
- But, can often be harder to learn (non-convex/concave optimization required)
- But, but, often very useful
- (BTW. Later this quarter, we'll see that these options are not that different)

A small dataset: Miles Per Gallon

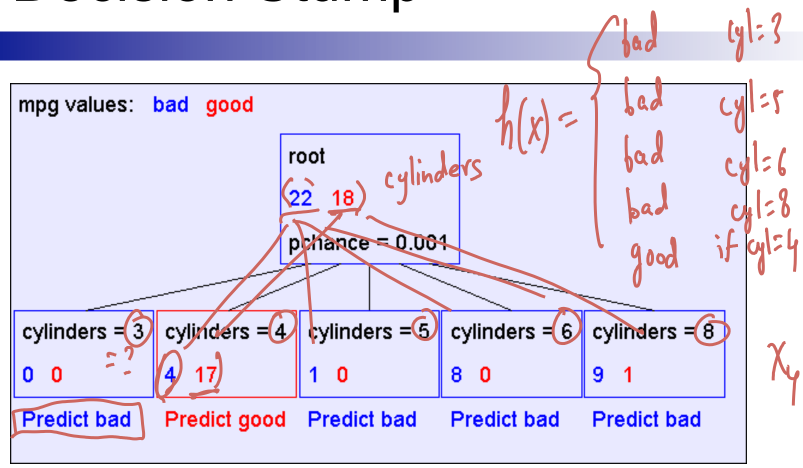
Suppose we want to predict MPG

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 training examples

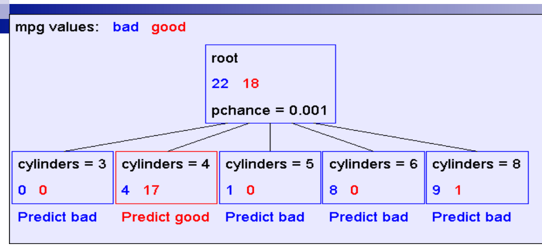
From the UCI repository (thanks to Ross Quinlan)

A Decision Stump

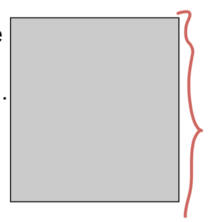


$h(x) = 1 \cdot X_4$

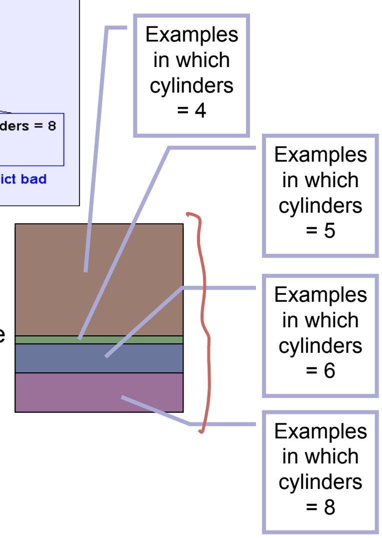
Recursion Step



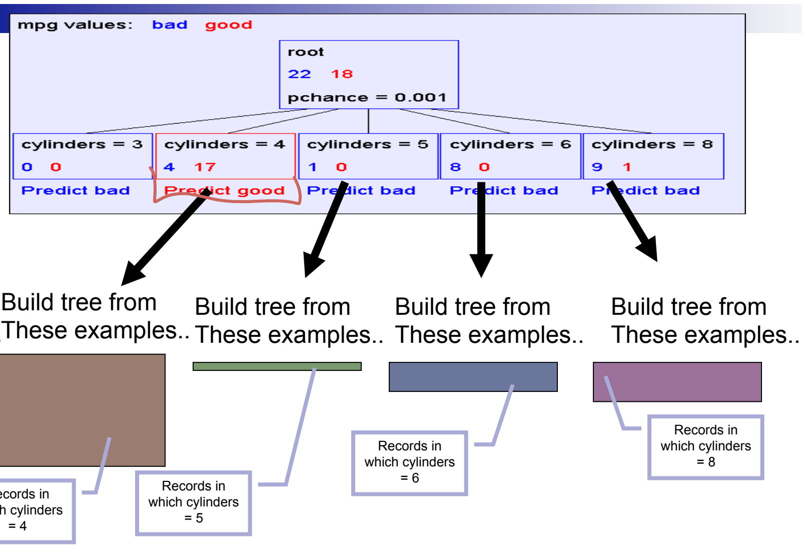
Take the Original Dataset..



And partition it according to the value of the attribute we split on

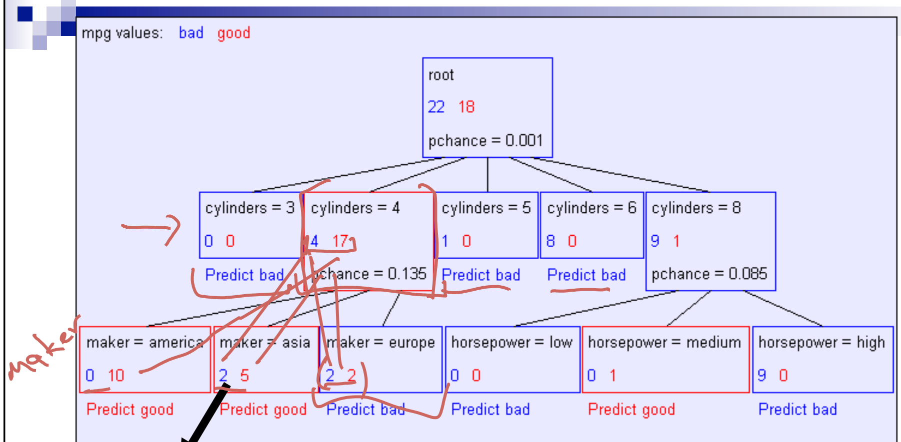


Recursion Step



©Carlos Guestrin 2005-2013 9

Second level of tree

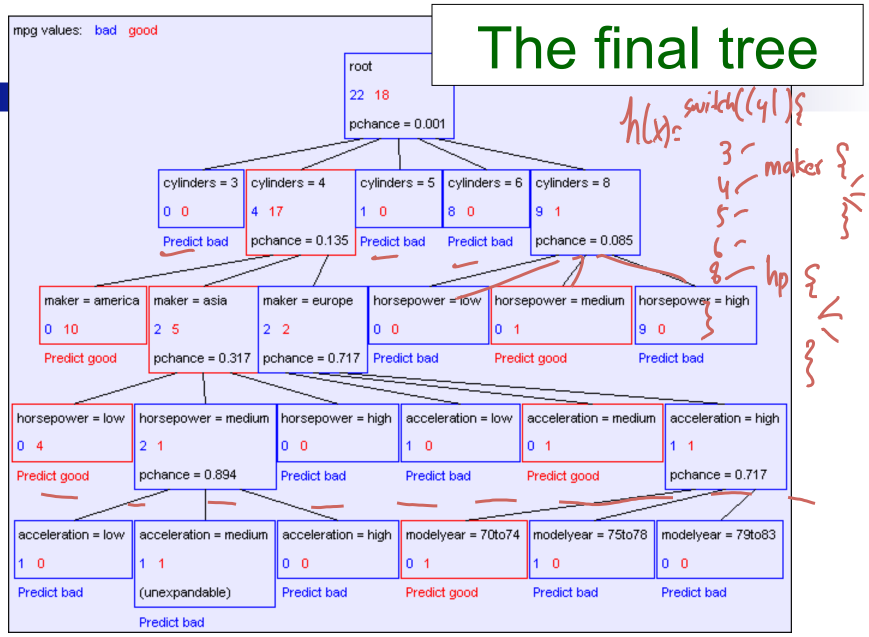


Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

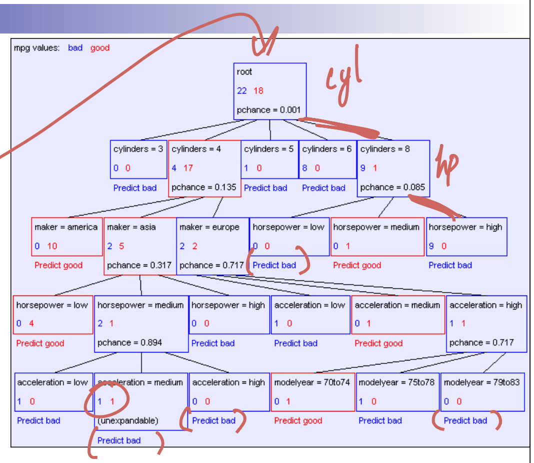
©Carlos Guestrin 2005-2013 10

The final tree



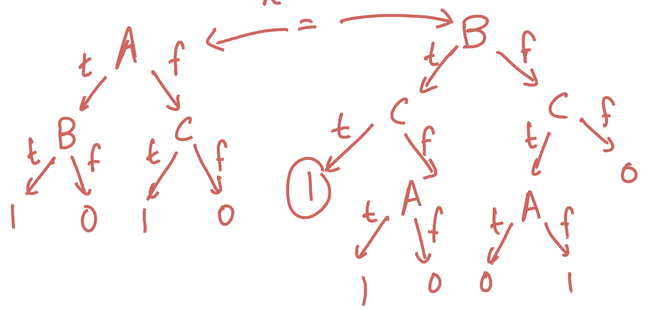
Classification of a new example

- Classifying a test example – traverse tree and report leaf label



Are all decision trees equal?

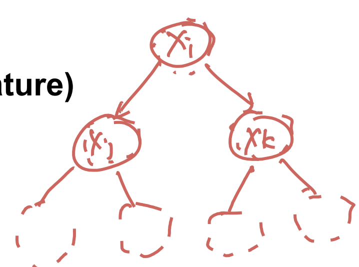
- Many trees can represent the same concept h
- But, not all trees will have the same size!
  - e.g., $\phi = \underbrace{A \wedge B \vee \neg A \wedge C}_{h}$ ((A and B) or (not A and C))



 13

Learning decision trees is hard!!!

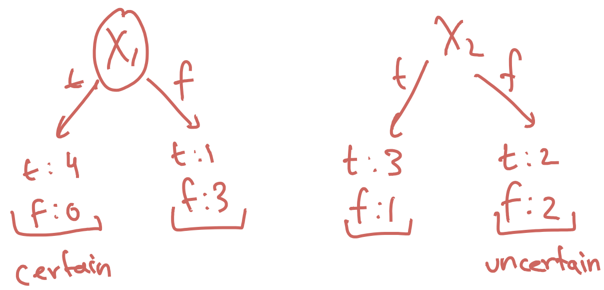
- Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest '76]
- Resort to a greedy heuristic:
  - Start from empty decision tree
  - Split on **next best attribute (feature)** X_i
  - Recurse



 14

Choosing a good attribute

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



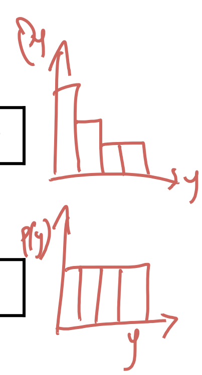
"certainty" is good!

Measuring uncertainty

- Good split if we are more certain about classification after split
  - Deterministic good (all true or all false)
  - Uniform distribution bad

P(Y=A) = 1/2	P(Y=B) = 1/4	P(Y=C) = 1/8	P(Y=D) = 1/8

P(Y=A) = 1/4	P(Y=B) = 1/4	P(Y=C) = 1/4	P(Y=D) = 1/4



Entropy

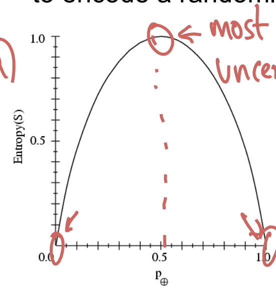
Entropy $H(Y)$ of a random variable $Y = \{y_1, y_2 \ldots y_k\}$

$$\boxed{\underline{H(Y)} = -\sum_{i=1}^{k} \underline{P(Y = y_i)} \log_2 \underline{P(Y = y_i)}}$$

More uncertainty, more entropy!

Information Theory interpretation: H(Y) is the expected number of bits needed to encode a randomly drawn value of Y (under most efficient code)

$H(Y)$



← most uncertain

$Y = \{0, 1\}$

P_0 $\quad 1 - P_0$

$P_0(Y)$

$$\frac{P_0 = 0}{P_0 = 1}$$

$P_0 = 0.5$

0 1

0 1

0 1

Andrew Moore's Entropy in a nutshell

Low Entropy

High Entropy

Andrew Moore's Entropy in a nutshell

Low Entropy

..the values (locations of soup) sampled entirely from within the soup bowl

High Entropy

..the values (locations of soup) unpredictable... almost uniformly sampled throughout our dining room

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

- Advantage of attribute – decrease in uncertainty
  - Entropy of Y before you split

    $H(Y) = \frac{5}{6}\log\frac{5}{6} + \frac{1}{6}\log\frac{1}{6} = 0.65$

    $P(y=t) = 5/6$

    $P(y=f) = 1/6$
  - Entropy after split
    - Weight by probability of following each branch, i.e., normalized number of records

X_1

$$H(Y \mid X_1) = -\sum_{j=1}^{v} P(X_1 = x_j) \sum_{i=1}^{k} P(Y = y_i \mid X_1 = x_j) \log_2 P(Y = y_i \mid X_1 = x_j)$$

over values of X_1 $X_1 = t$ over values of Y $X_1 = f$

$$H(Y|X_1) = \frac{2}{3}(0) + \frac{1}{3}(1) = \frac{1}{3}$$

- Information gain is difference $IG(X) = H(Y) - H(Y \mid X)$

$$IG(X_1) = 0.65 - \frac{1}{3} = \boxed{0.32}$$

Learning decision trees

- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute
  - Split on $\arg\max_i IG(X_i) = \arg\max_i H(Y) - H(Y \mid X_i)$
- Recurse

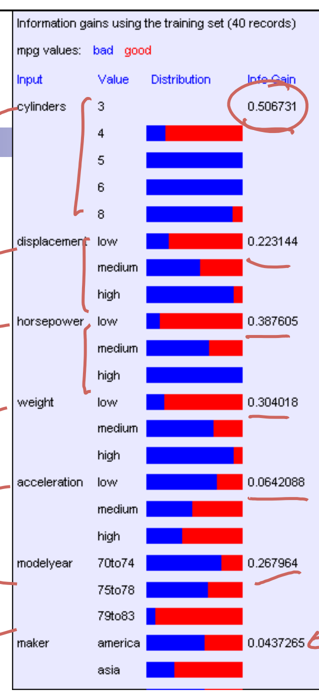
When do you stop?

1) entropy is 0

2) Cannot split

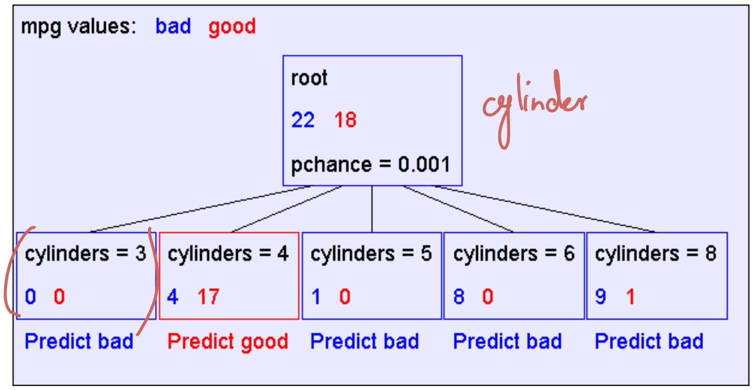
3) when IG is 0

Suppose we want to predict MPG

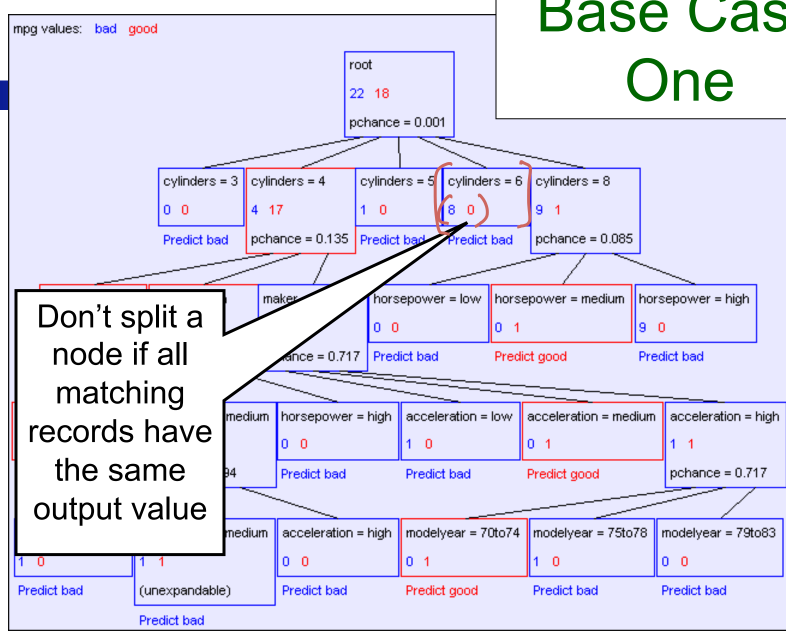


Look at all the information gains…

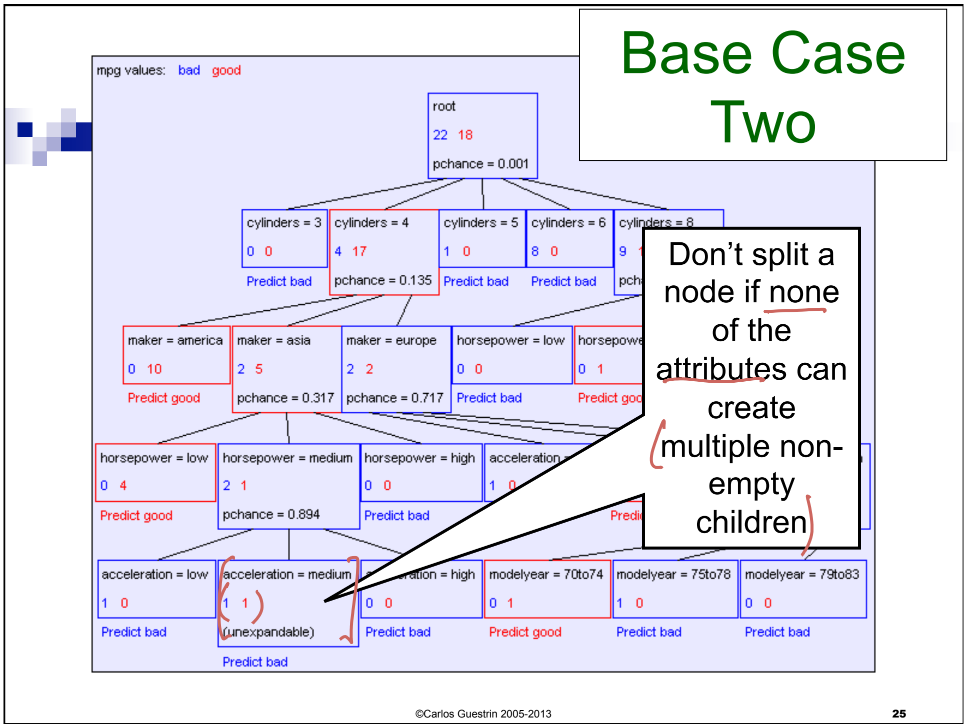
A Decision Stump

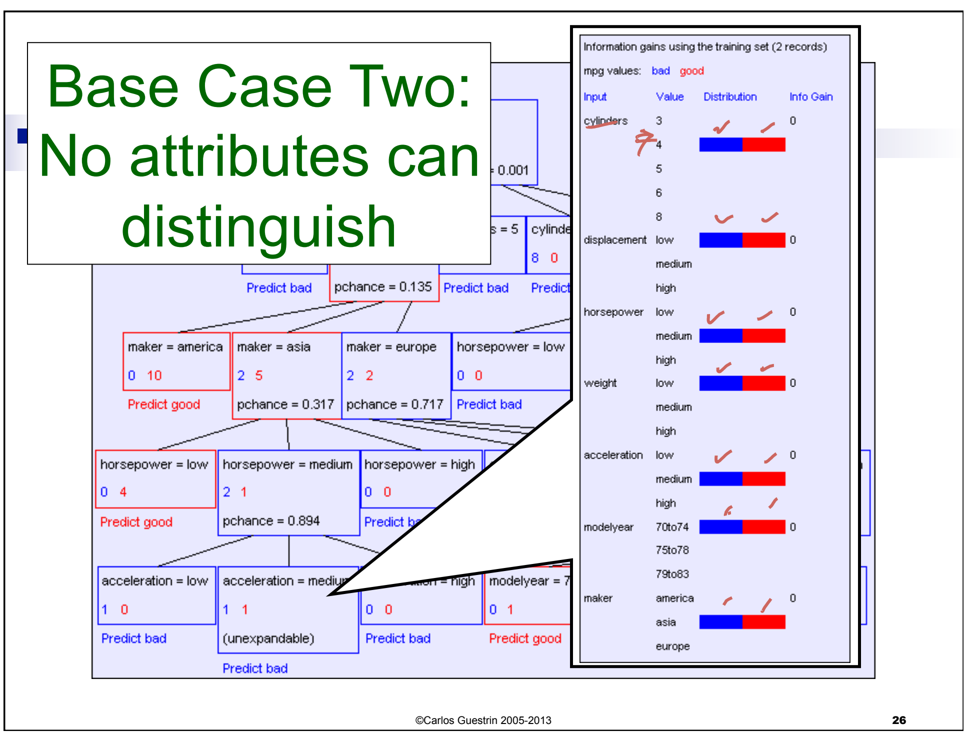


Base Case One



Don't split a node if all matching records have the same output value


Base Case Two
Don't split a node if none of the attributes can create multiple non-empty children
mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
Predict bad
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad
©Carlos Guestrin 2005-2013
25


Base Case Two: No attributes can distinguish
Information gains using the training set (2 records)
mpg values: bad good
Input Value Distribution Info Gain
cylinders 3 4 5 6 8 0
displacement low medium high 0
horsepower low medium high 0
weight low medium high 0
acceleration low medium high 0
modelyear 70to74 75to78 79to83 0
maker america asia europe 0
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = low
0 4
Predict good
horsepower = medium
2 1
pchance = 0.894
horsepower = high
0 0
acceleration = low
1 0
Predict bad
1 1
(unexpandable)
Predict bad
0 0
Predict bad
0 1
Predict good
©Carlos Guestrin 2005-2013
26

Base Cases

- Base Case One: If all records in current data subset have the same output then don't recurse
- Base Case Two: If all records have exactly the same set of input attributes then don't recurse

3rd case: 0 Info gain

Does this work?

Base Cases: An idea

- Base Case One: If all records in current data subset have the same output then don't recurse
- Base Case Two: If all records have exactly the same set of input attributes then don't recurse

Proposed Base Case 3:

If all attributes have zero information gain then don't recurse

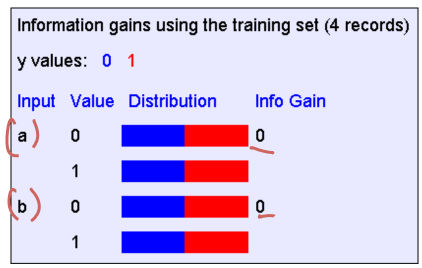
• *Is this a good idea?*

The problem with Base Case 3

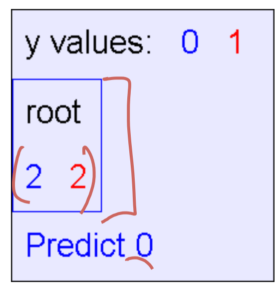
a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

Y = A XOR B

The information gains:



The resulting bad decision tree:



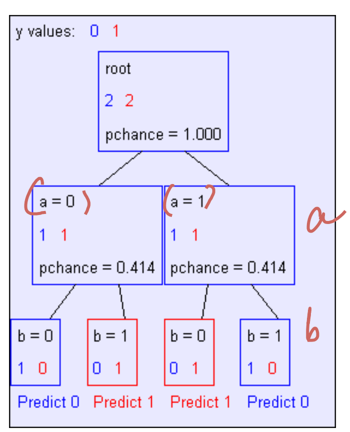
 29

If we omit Base Case 3:

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

y = a XOR b

The resulting decision tree:



 30

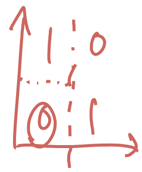
Basic Decision Tree Building Summarized

BuildTree(*DataSet*,*Output*)

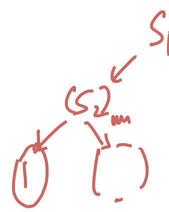
- bc 1 If all output values are the same in *DataSet*, return a leaf node that says "predict this unique output"
- bc 2 If all input values are the same, return a leaf node that says "predict the majority output"
- Else find attribute *X* with highest Info Gain
- Suppose *X* has n_X distinct values (i.e. X has arity n_X).
  - Create and return a non-leaf node with n_X children.
  - The *i*'th child should be built by calling

    BuildTree(DS_i,*Output*)

    Where DS_i built consists of all those records in DataSet for which X = *i*th distinct value of X.

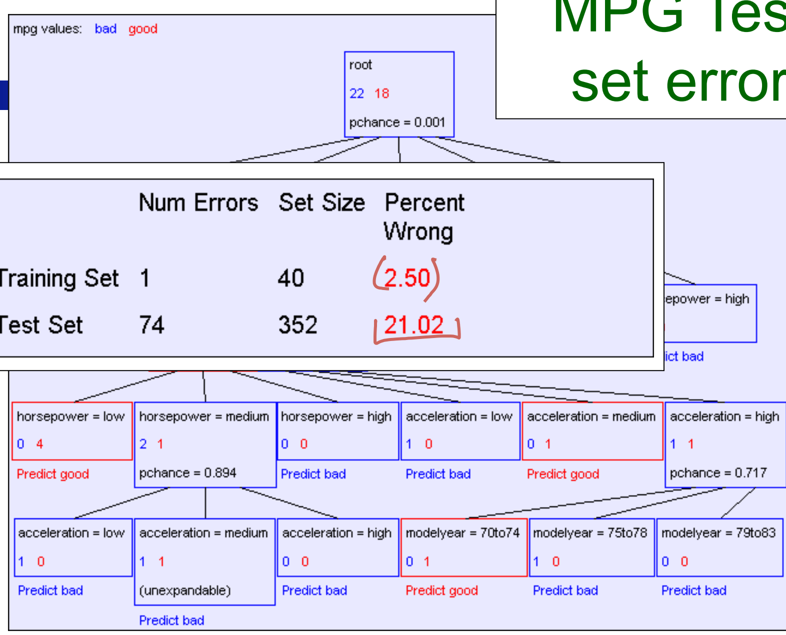


$DS_i \subseteq DS$



 31

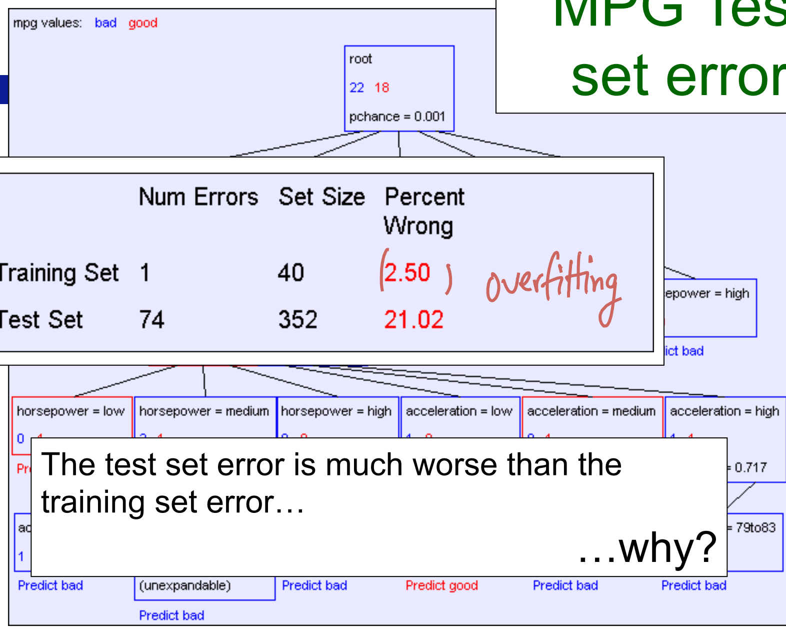
MPG Test set error



	Num Errors	Set Size	Percent Wrong
Training Set	1	40	2.50
Test Set	74	352	21.02

 32

MPG Test set error



	Num Errors	Set Size	Percent Wrong
Training Set	1	40	(2.50)
Test Set	74	352	21.02

overfitting

The test set error is much worse than the training set error…

…why?

Decision trees & (Learning Bias)

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

$h(x)$	linear	decision
bias	high	low
variance	low	high