

Because $N \rightarrow$ is huge
"big data"

Stochastic Gradient Descent

Machine Learning – CSE446

Carlos Guestrin

University of Washington

April 19, 2013

©Carlos Guestrin 2005-2013

1

Logistic Regression

Logistic function
(or Sigmoid): $\frac{1}{1 + \exp(-z)}$

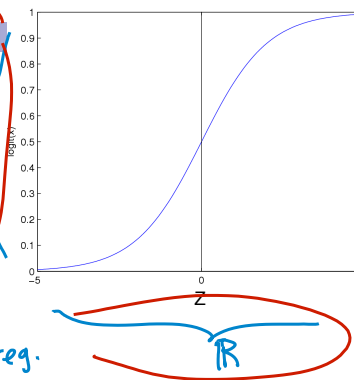
Learn $P(Y|X)$ directly

- Assume a particular functional form for link function
- Sigmoid applied to a linear function of the input features:

$$P(Y = 0|X, W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

choice

z : linear just like in reg.



$w_0 + \sum_i w_i X_i$ ← not bounded, could be neg.

after logistic fcn, output is in $[0, 1]$

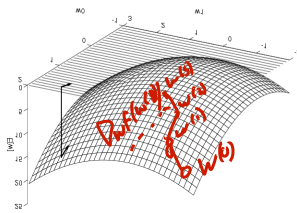
Features can be discrete or continuous!

©Carlos Guestrin 2005-2013

2

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave. Find optimum with gradient ascent



Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = \left[\frac{\partial l(\mathbf{w})}{\partial w_0}, \dots, \frac{\partial l(\mathbf{w})}{\partial w_n} \right]'$

Step size, $\eta > 0$

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w}^{(t)})}{\partial w_i}$$

But Hw? η will be constant

- Gradient ascent is simplest of optimization approaches
 - e.g., Conjugate gradient ascent can be much better

Often, especially in proof, η gets smaller with iterations
e.g. $\eta_t = \frac{\alpha}{t}$ $\alpha \leftarrow \text{constant}$

©Carlos Guestrin 2005-2013

3

Gradient Ascent for LR

Start from some $w^{(0)}$ e.g. \emptyset

revisit soon

Gradient ascent algorithm: iterate until change $< \epsilon$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For $i=1, \dots, k$,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

repeat

©Carlos Guestrin 2005-2013

4

The Cost, The Cost!!! Think about the cost...

$k, N \leftarrow$ in terms of

- What's the cost of a gradient update step for LR???

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_{j=1}^N x_i^j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}$$

for i

$O(k)$

$O(Nk)$

naively $O(Nk^2)$

but cache \hat{P} (same for all i)

$O(Nk) \in \dots$

if N is huge
very slow
per little
gradient step

©Carlos Guestrin 2005-2013

5

Learning Problems as Expectations

- Minimizing loss in training data:
 - Given dataset:
 - Sampled iid from some distribution $p(\mathbf{x})$ on features:
 - Loss function, e.g., hinge loss, logistic loss,...
 - We often minimize loss in training data:

$$\ell_{\mathcal{D}}(\mathbf{w}) = \frac{1}{N} \sum_{j=1}^N \ell(\mathbf{w}, \mathbf{x}^j)$$

- However, we should really minimize expected loss on all data:

$$\ell(\mathbf{w}) = E_{\mathbf{x}} [\ell(\mathbf{w}, \mathbf{x})] = \int p(\mathbf{x}) \ell(\mathbf{w}, \mathbf{x}) d\mathbf{x}$$

- So, we are approximating the integral by the average on the training data

©Carlos Guestrin 2005-2013

6

Gradient ascent in Terms of Expectations

- “True” objective function:

$$\ell(\mathbf{w}) = E_{\mathbf{x}} [\ell(\mathbf{w}, \mathbf{x})] = \int p(\mathbf{x}) \ell(\mathbf{w}, \mathbf{x}) d\mathbf{x}$$

- Taking the gradient:
- “True” gradient ascent rule:
- How do we estimate expected gradient?

©Carlos Guestrin 2005-2013

7

SGD: Stochastic Gradient Ascent (or Descent)

- “True” gradient: $\nabla \ell(\mathbf{w}) = E_{\mathbf{x}} [\nabla \ell(\mathbf{w}, \mathbf{x})]$
- Sample based approximation:
- What if we estimate gradient with just one sample???
 - Unbiased estimate of gradient
 - Very noisy!
 - Called stochastic gradient ascent (or descent)
 - Among many other names
 - VERY useful in practice!!!

©Carlos Guestrin 2005-2013

8

Stochastic Gradient Ascent for Logistic Regression

- Logistic loss as a stochastic function:

$$E_{\mathbf{x}} [\ell(\mathbf{w}, \mathbf{x})] = E_{\mathbf{x}} [\ln P(y|\mathbf{x}, \mathbf{w}) - \lambda \|\mathbf{w}\|_2^2]$$

- Batch gradient ascent updates:

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \frac{1}{N} \sum_{j=1}^N x_i^{(j)} [y^{(j)} - P(Y=1|\mathbf{x}^{(j)}, \mathbf{w}^{(t)})] \right\}$$

- Stochastic gradient ascent updates:

- ☐ Online setting:

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta_t \left\{ -\lambda w_i^{(t)} + x_i^{(t)} [y^{(t)} - P(Y=1|\mathbf{x}^{(t)}, \mathbf{w}^{(t)})] \right\}$$

©Carlos Guestrin 2005-2013

9

Stochastic Gradient Ascent: general case

- Given a stochastic function of parameters:

- ☐ Want to find maximum

- Start from $\mathbf{w}^{(0)}$

- Repeat until convergence:

- ☐ Get a sample data point \mathbf{x}^t
- ☐ Update parameters:

- Works on the online learning setting!

- Complexity of each gradient step is constant in number of examples!

- In general, step size changes with iterations

©Carlos Guestrin 2005-2013

10

What you should know...

- Classification: predict discrete classes rather than real values
- Logistic regression model: Linear model
 - Logistic function maps real values to $[0,1]$
- Optimize conditional likelihood
- Gradient computation
- Overfitting
- Regularization
- Regularized optimization
- Cost of gradient step is high, use stochastic gradient descent

©Carlos Guestrin 2005-2013

11

Decision Trees

Machine Learning – CSE446

Carlos Guestrin

University of Washington

April 19, 2013

©Carlos Guestrin 2005-2013

12

Linear separability

- A dataset is **linearly separable** iff there exists a **separating hyperplane**:

- Exists \mathbf{w} , such that:

- $w_0 + \sum_i w_i x_i > 0$; if $\mathbf{x}=\{x_1, \dots, x_k\}$ is a positive example
- $w_0 + \sum_i w_i x_i < 0$; if $\mathbf{x}=\{x_1, \dots, x_k\}$ is a negative example

Not linearly separable data

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

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

©Carlos Guestrin 2005-2013

15

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

- Choose a classifier $h_{\mathbf{w}}(\mathbf{x})$ that is non-linear in parameters \mathbf{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)

©Carlos Guestrin 2005-2013

16

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	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

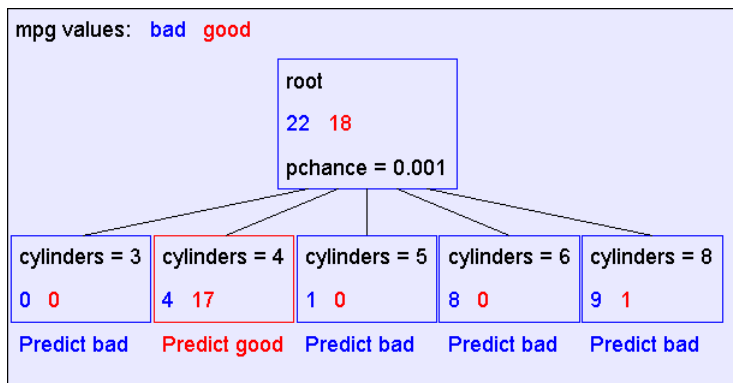
40 training examples

From the UCI repository (thanks to Ross Quinlan)

©Carlos Guestrin 2005-2013

17

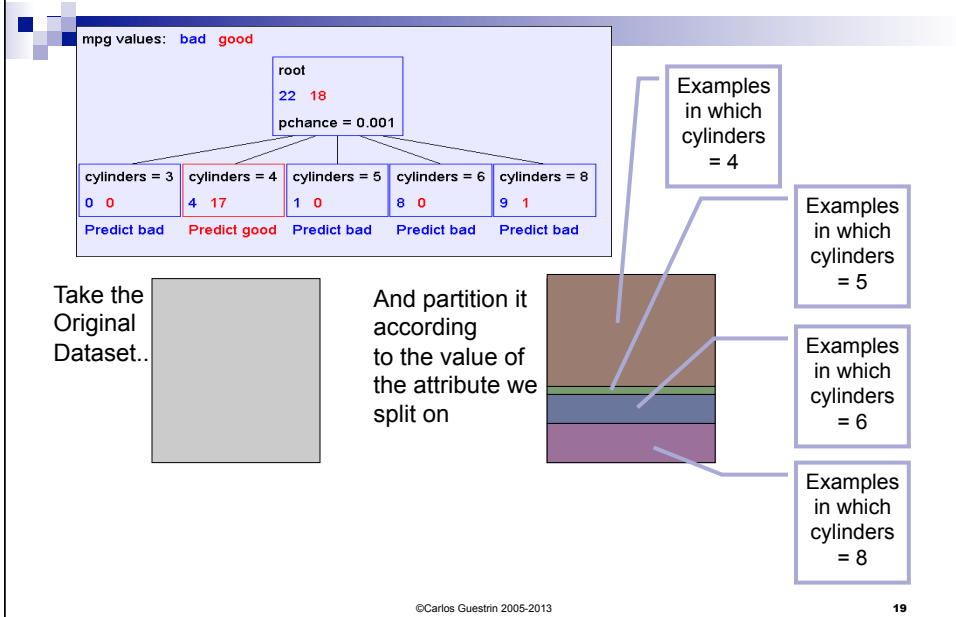
A Decision Stump



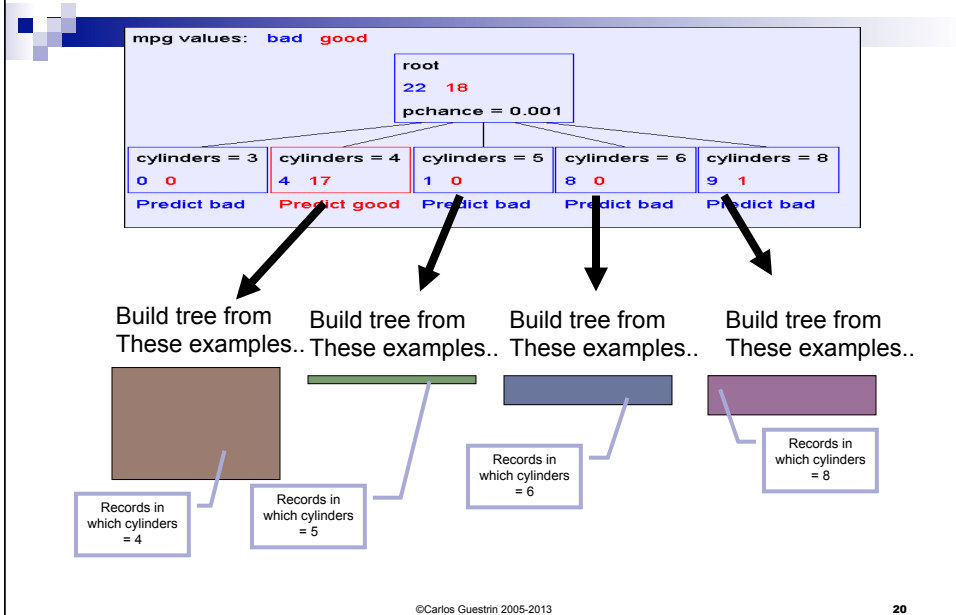
©Carlos Guestrin 2005-2013

18

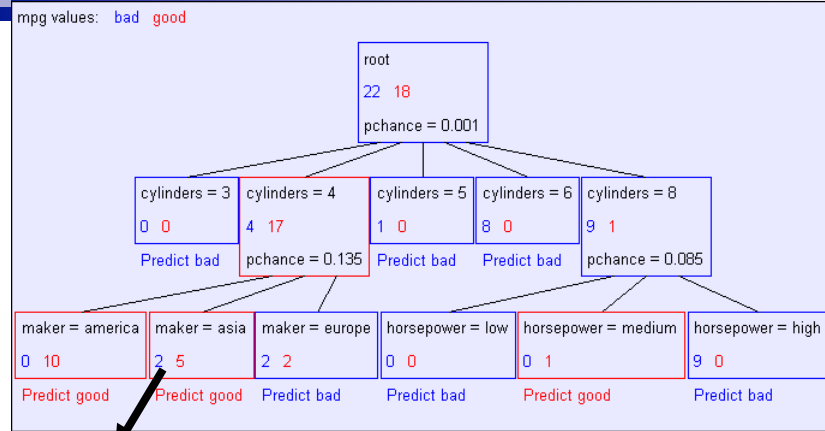
Recursion Step



Recursion Step



Second level of tree



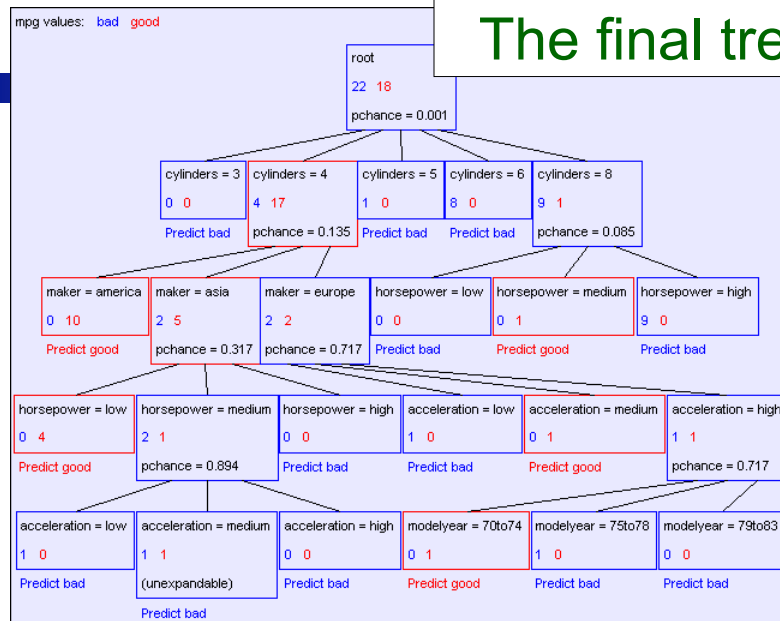
Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)

©Carlos Guestrin 2005-2013

21

The final tree

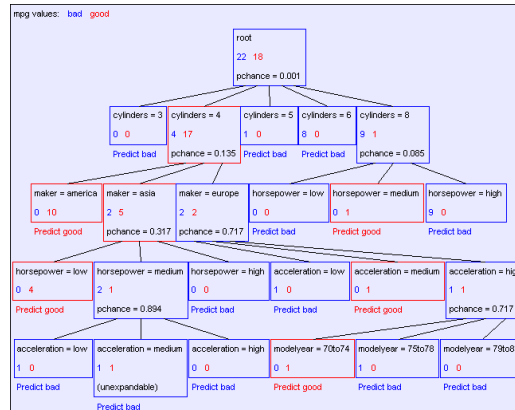


©Carlos Guestrin 2005-2013

22

Classification of a new example

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



©Carlos Guestrin 2005-2013

23

Are all decision trees equal?

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

©Carlos Guestrin 2005-2013

24

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)**
 - Recurse

©Carlos Guestrin 2005-2013

25

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

©Carlos Guestrin 2005-2013

26

Measuring uncertainty

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

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

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

©Carlos Guestrin 2005-2013

27

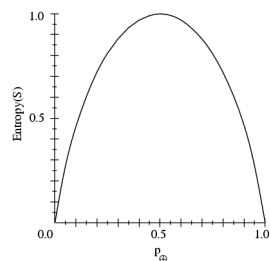
Entropy

Entropy $H(X)$ of a random variable Y

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

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)



©Carlos Guestrin 2005-2013

28

Andrew Moore's Entropy in a nutshell



Low Entropy



High Entropy

©Carlos Guestrin 2005-2013

29

Andrew Moore's Entropy in a nutshell



Low Entropy



High Entropy

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

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

©Carlos Guestrin 2005-2013

30

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

- Entropy after split

- Weight by probability of following each branch, i.e., normalized number of records

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

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

©Carlos Guestrin 2005-2013

31

Learning decision trees

- Start from empty decision tree

- Split on **next best attribute (feature)**

- Use, for example, information gain to select attribute

- Split on $\arg \max_i IG(X_i) = \arg \max_i H(Y) - H(Y | X_i)$

- Recurse

©Carlos Guestrin 2005-2013

32

Suppose we want
to predict MPG

Look at all the
information
gains...

Information gains using the training set (40 records)

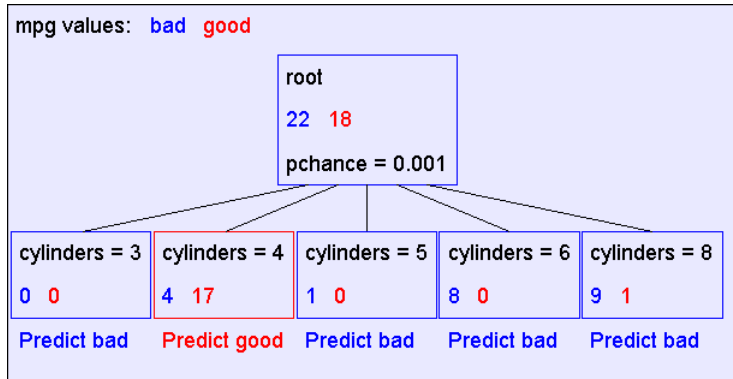
mpg values: bad good

Input	Value	Distribution	Info Gain
cylinders	3		0.506731
	4		
	5		
	6		
	8		
displacement	low		0.223144
	medium		
	high		
horsepower	low		0.387605
	medium		
	high		
weight	low		0.304018
	medium		
	high		
acceleration	low		0.0642088
	medium		
	high		
modelyear	70to74		0.267964
	75to78		
	79to83		
maker	america		0.0437265
	asia		

©Carlos Guestrin 2005-2013

33

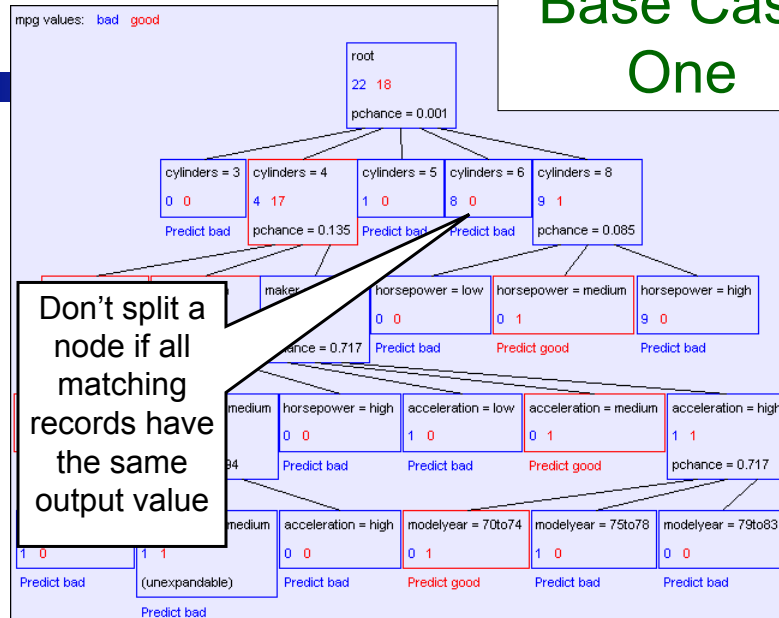
A Decision Stump



©Carlos Guestrin 2005-2013

34

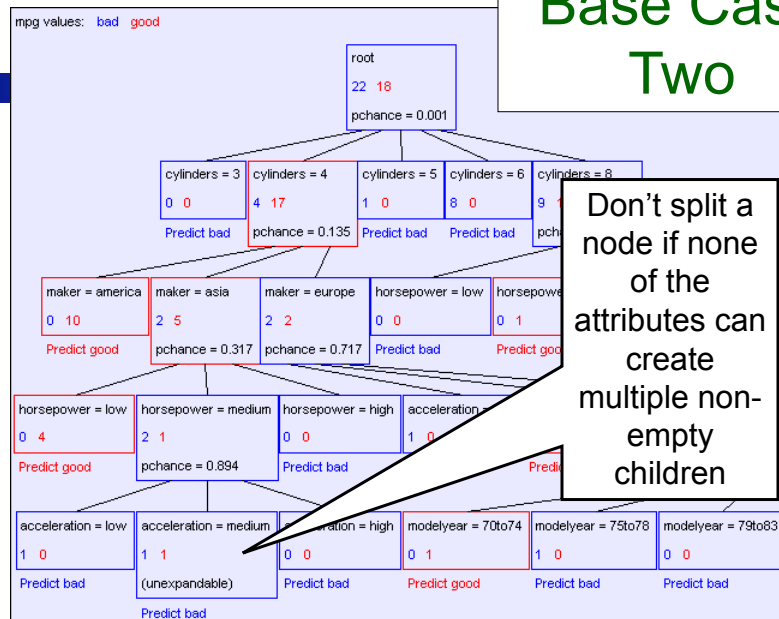
Base Case One



©Carlos Guestrin 2005-2013

35

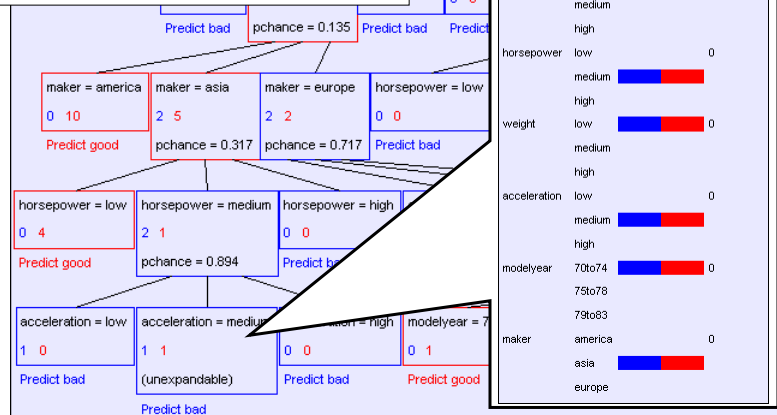
Base Case Two



©Carlos Guestrin 2005-2013

36

Base Case Two: No attributes can distinguish



©Carlos Guestrin 2005-2013

37

Base Cases

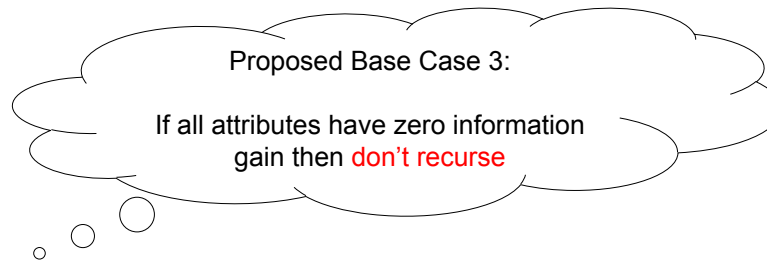
- Base Case One: If all records in current data subset have the same output then **don't recurse**
- Base Case Two: If all records have exactly the same set of input attributes then **don't recurse**

©Carlos Guestrin 2005-2013

38

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**



•Is this a good idea?

©Carlos Guestrin 2005-2013

39

The problem with Base Case 3

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

$$Y = A \text{ XOR } B$$

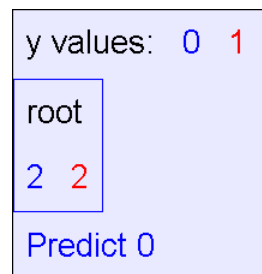
The information gains:

Information gains using the training set (4 records)

y values: 0 1

Input	Value	Distribution	Info Gain
a	0		0
a	1		0
b	0		0
b	1		0

The resulting bad decision tree:



©Carlos Guestrin 2005-2013

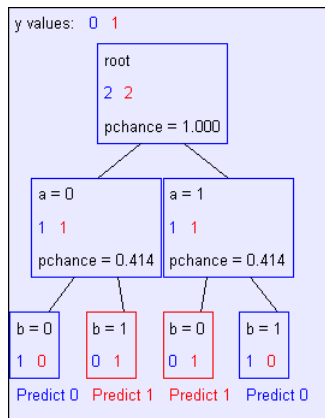
40

If we omit Base Case 3:

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

$$y = a \text{ XOR } b$$

The resulting decision tree:



©Carlos Guestrin 2005-2013

41

Basic Decision Tree Building Summarized

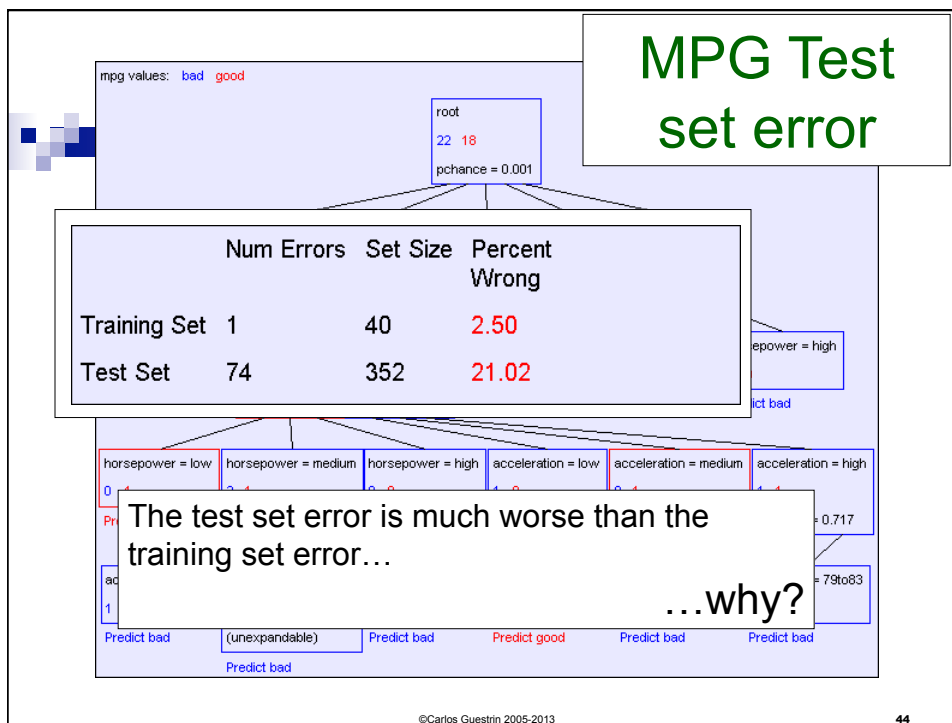
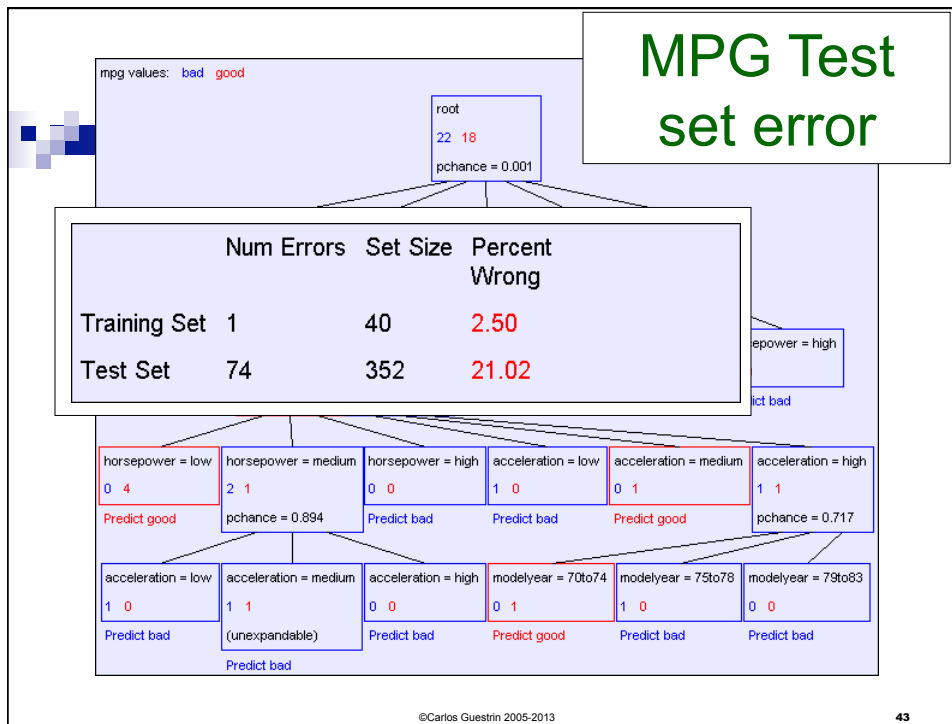
BuildTree(DataSet, Output)

- If all output values are the same in *DataSet*, return a leaf node that says “predict this unique output”
- 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*.

©Carlos Guestrin 2005-2013

42



Decision trees & Learning Bias

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	78to78	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	6	medium	medium	medium	medium	75to78	europa

©Carlos Guestrin 2005-2013

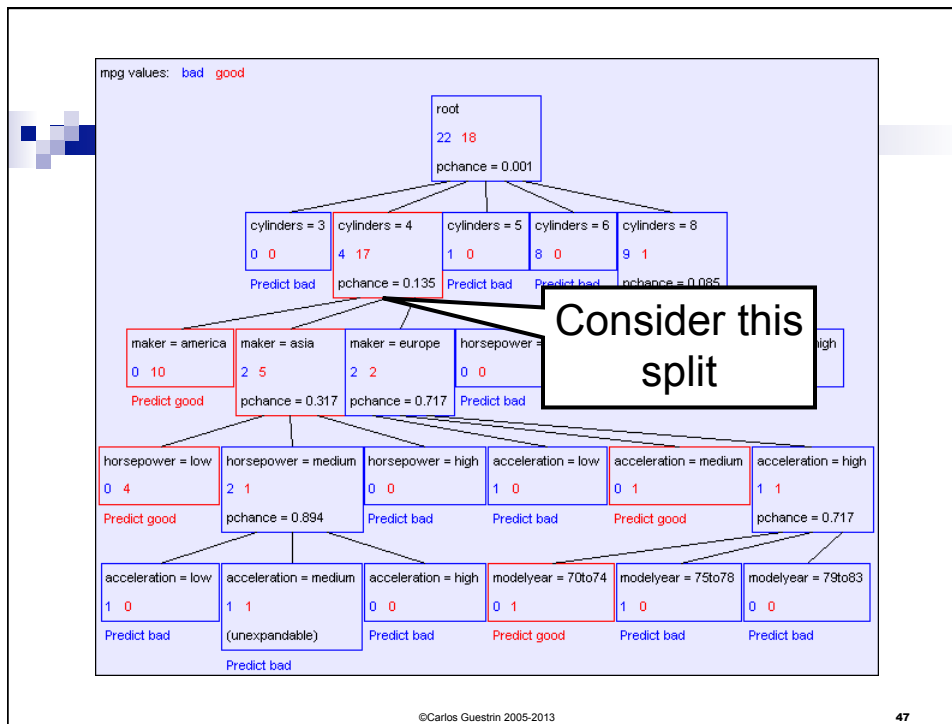
45

Decision trees will overfit

- Standard decision trees are have no learning bias
 - Training set error is always zero!
 - (If there is no label noise)
 - Lots of variance
 - Will definitely overfit!!!
 - Must bias towards simpler trees
- Many strategies for picking simpler trees:
 - Fixed depth
 - Fixed number of leaves
 - Or something smarter...

©Carlos Guestrin 2005-2013

46



A chi-square test

mpg values: bad good







maker	bad	good	$H(\text{mpg} \text{maker})$
america	0	10	0
asia	2	5	0.863121
europe	2	2	1

$H(\text{mpg}) = 0.702467$ $H(\text{mpg} | \text{maker}) = 0.478183$
 $IG(\text{mpg} | \text{maker}) = 0.224284$

- Suppose that MPG was completely uncorrelated with maker.
- What is the chance we'd have seen data of at least this apparent level of association anyway?

A chi-square test

mpg values: bad good

maker	america	0	10			$H(\text{mpg} \mid \text{maker} = \text{america}) = 0$
	asia	2	5			$H(\text{mpg} \mid \text{maker} = \text{asia}) = 0.863121$
	europa	2	2			$H(\text{mpg} \mid \text{maker} = \text{europa}) = 1$

$H(\text{mpg}) = 0.702467$ $H(\text{mpg} \mid \text{maker}) = 0.478183$
 $IG(\text{mpg} \mid \text{maker}) = 0.224284$

- Suppose that mpg was completely uncorrelated with maker.
- What is the chance we'd have seen data of at least this apparent level of association anyway?

By using a particular kind of chi-square test, the answer is 7.2%

(Such simple hypothesis tests are very easy to compute, unfortunately, not enough time to cover in the lecture, but see readings...)

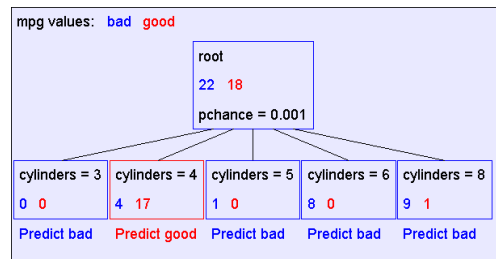
Using Chi-squared to avoid overfitting

- Build the full decision tree as before
- But when you can grow it no more, start to prune:
 - Beginning at the bottom of the tree, delete splits in which $p_{\text{chance}} > \text{MaxPchance}$
 - Continue working your way up until there are no more prunable nodes

MaxPchance is a magic parameter you must specify to the decision tree, indicating your willingness to risk fitting noise

Pruning example

- With MaxPchance = 0.1, you will see the following MPG decision tree:



Note the improved test set accuracy compared with the unpruned tree

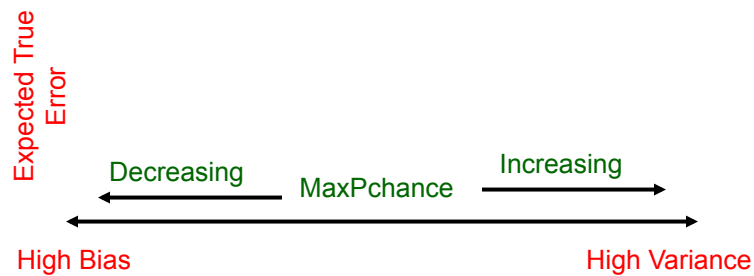
	Num Errors	Set Size	Percent Wrong
Training Set	5	40	12.50
Test Set	56	352	15.91

©Carlos Guestrin 2005-2013

51

MaxPchance

- Technical note MaxPchance is a regularization parameter that helps us bias towards simpler models



©Carlos Guestrin 2005-2013

52

Real-Valued inputs

- What should we do if some of the inputs are real-valued?

mpg	cylinders	displacemen	horsepower	weight	acceleration	modelyear	maker
good	4	97	75	2265	18.2	77	asia
bad	6	199	90	2648	15	70	america
bad	4	121	110	2600	12.8	77	europa
bad	8	350	175	4100	13	73	america
bad	6	198	95	3102	16.5	74	america
bad	4	108	94	2379	16.5	73	asia
bad	4	113	95	2228	14	71	asia
bad	8	302	139	3570	12.8	78	america
.
.
.
.
good	4	120	79	2625	18.6	82	america
bad	8	455	225	4425	10	70	america
good	4	107	86	2464	15.5	76	europa
bad	5	131	103	2830	15.9	78	europa

Infinite number of possible split values!!!

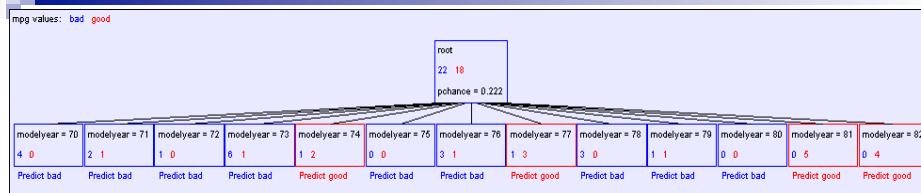
Finite dataset, only finite number of relevant splits!

Idea One: Branch on each possible real value

©Carlos Guestrin 2005-2013

53

“One branch for each numeric value” idea:



Hopeless: with such high branching factor will shatter the dataset and overfit

©Carlos Guestrin 2005-2013

54

Threshold splits

- Binary tree, split on attribute X
 - One branch: $X < t$
 - Other branch: $X \geq t$

Choosing threshold split

- Binary tree, split on attribute X
 - One branch: $X < t$
 - Other branch: $X \geq t$
- Search through possible values of t
 - Seems hard!!!
- But only finite number of t 's are important
 - Sort data according to X into $\{x_1, \dots, x_m\}$
 - Consider split points of the form $x_i + (x_{i+1} - x_i)/2$

A better idea: thresholded splits

- Suppose X is real valued
- Define $IG(Y|X:t)$ as $H(Y) - H(Y|X:t)$
- Define $H(Y|X:t) = H(Y|X < t) P(X < t) + H(Y|X \geq t) P(X \geq t)$
 - $IG(Y|X:t)$ is the information gain for predicting Y if all you know is whether X is greater than or less than t
- Then define $IG^*(Y|X) = \max_t IG(Y|X:t)$
- For each real-valued attribute, use $IG^*(Y|X)$ for assessing its suitability as a split
- Note, may split on an attribute multiple times, with different thresholds

©Carlos Guestrin 2005-2013

57

Information gains using the training set (40 records)

mpg values: bad good

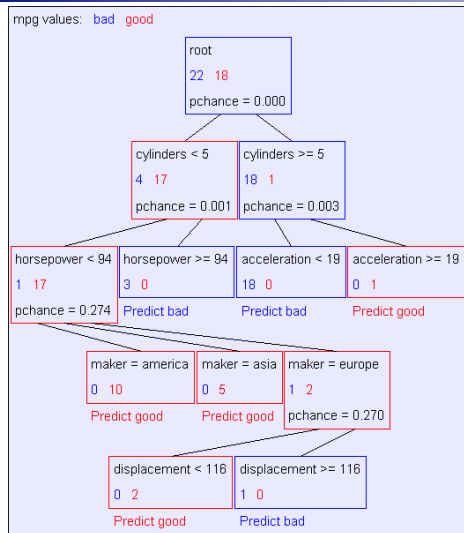
Input	Value	Distribution	Info Gain
cylinders	< 5		0.48268
	>= 5		
displacement	< 198		0.428205
	>= 198		
horsepower	< 94		0.48268
	>= 94		
weight	< 2789		0.379471
	>= 2789		
acceleration	< 18.2		0.159982
	>= 18.2		
modelyear	< 81		0.319193
	>= 81		
maker	america		0.0437265
	asia		
	europe		

©Carlos Guestrin 2005-2013

58

Example with MPG

Example tree using reals



©Carlos Guestrin 2005-2013

59

What you need to know about decision trees

- Decision trees are one of the most popular data mining tools
 - Easy to understand
 - Easy to implement
 - Easy to use
 - Computationally cheap (to solve heuristically)
- Information gain to select attributes (ID3, C4.5,...)
- Presented for classification, can be used for regression and density estimation too
- Decision trees will overfit!!!
 - Zero bias classifier ! Lots of variance
 - Must use tricks to find "simple trees", e.g.,
 - Fixed depth/Early stopping
 - Pruning
 - Hypothesis testing

©Carlos Guestrin 2005-2013

60

Acknowledgements

- Some of the material in the decision trees presentation is courtesy of Andrew Moore, from his excellent collection of ML tutorials:

- <http://www.cs.cmu.edu/~awm/tutorials>