DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 27 2020

Announcements

- HW 3 is due on Thu, Oct. 29
- Project proposal
  - Due on Monday, Nov. 2
  - Team of 2
  - Resources and example projects on Piazza

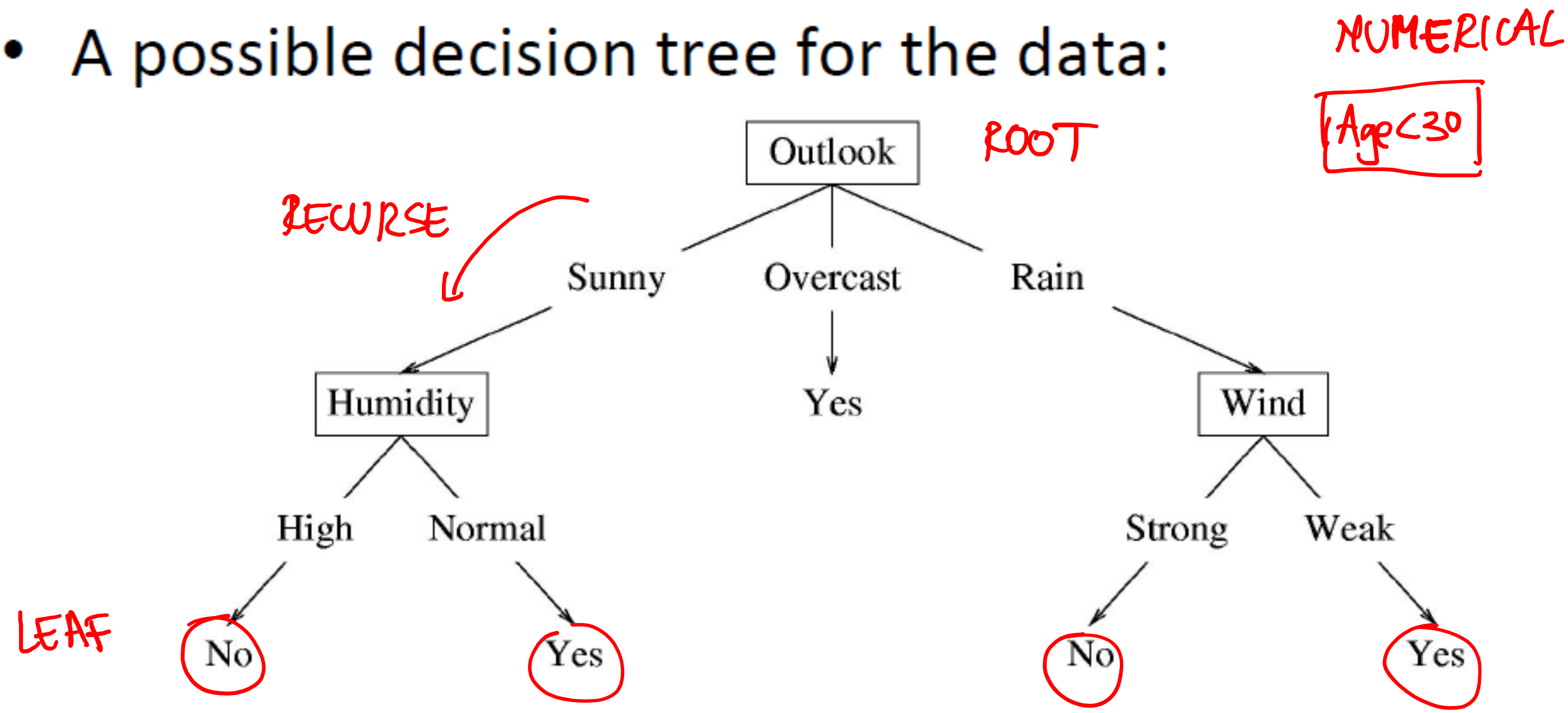
- EXAM- NOV. 19

Outline

- Decision trees
  - Information gain
  - Learning decision trees
- Ensemble learning
  - Combine multiple classifiers to reduce model variance and improve accuracy
- Bagging
  - Bootstrap samples
  - Random Forests

Decision Tree

- A possible decision tree for the data:



- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid x \in \text{leaf})$)

Information Gain

SHANNON

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Information Gain:

$IG(Y|X)$ = **I must transmit** Y. **How many bits on average would it save me if both ends of the line knew** X**?**

$$IG(Y|X) = \underbrace{H(Y)}_{\text{ENTROPY}} - \underbrace{H(Y|X)}_{\text{COND. ENTROPY}}$$

Example:

X&Y INDEP $\Rightarrow IG(Y|X) = 0$

$X = Y \Rightarrow IG(Y|X) = H(Y)$

- **H(Y) = 1**
- **H(Y|X) = 0.5**
- **Thus IG(Y|X) = 1 – 0.5 = 0.5**

$$H(Y|X) = \sum_{v \text{ values}} P[X=v] \cdot H(Y \mid X=v)$$

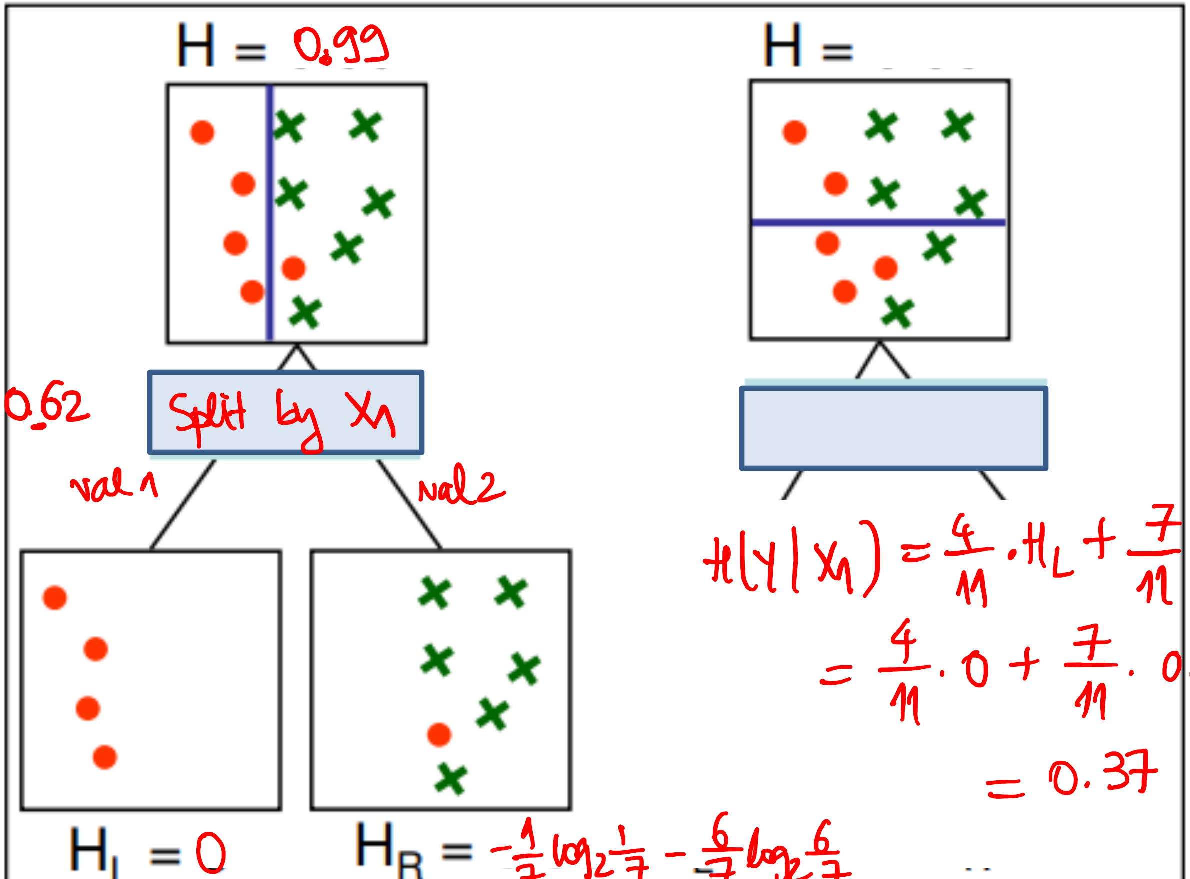
Example Information Gain

$$Y: \begin{pmatrix} \text{Red} & \text{Green} \\ 5/11 & 6/11 \end{pmatrix}$$

$$H(Y) = -\frac{5}{11}\log_2\frac{5}{11} - \frac{6}{11}\log_2\frac{6}{11}$$

IG = 0.62

PURE NODES



$$H(Y|X_1) = \frac{4}{11}\cdot H_L + \frac{7}{11}\cdot H_R$$

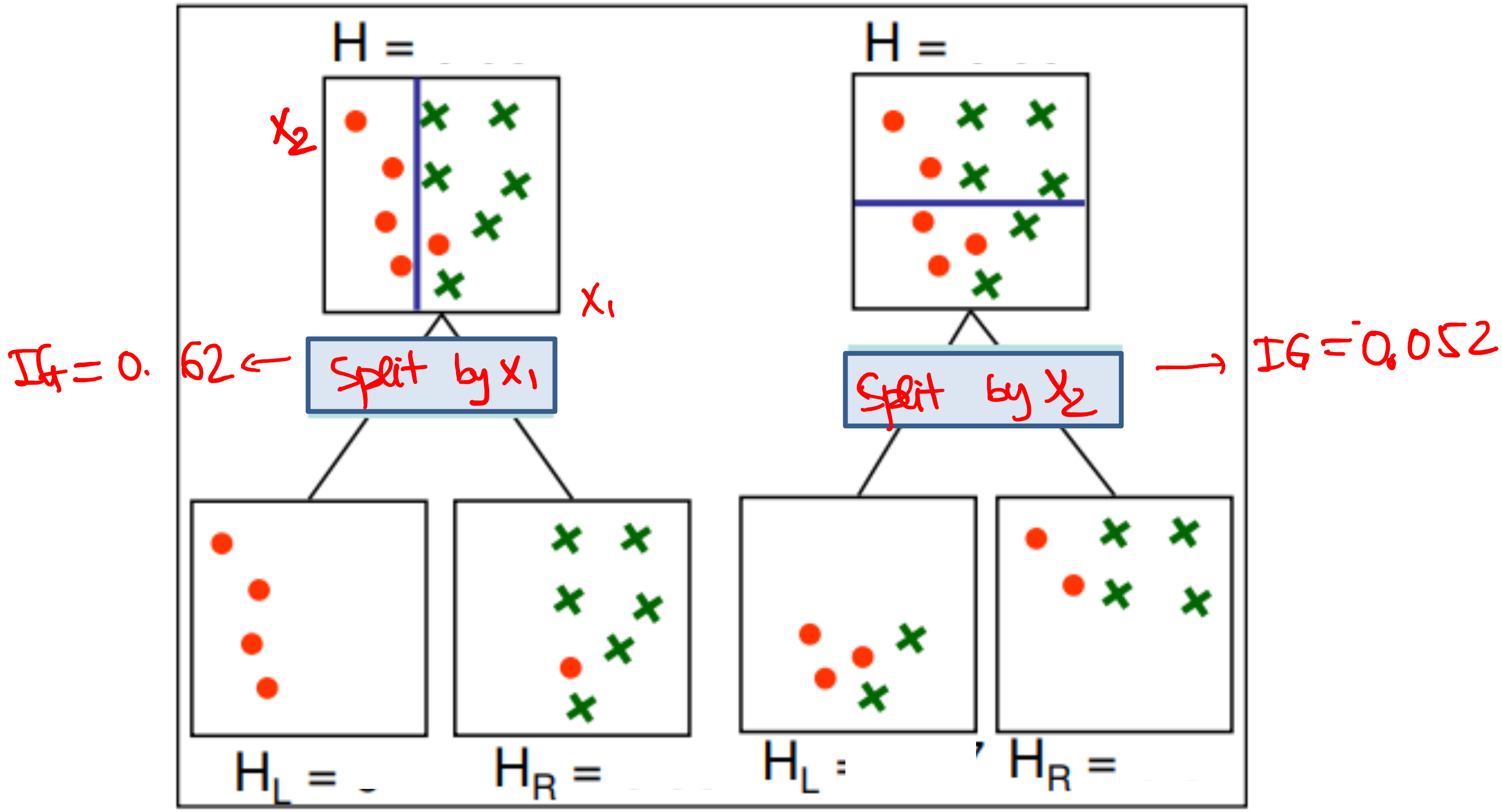
$$= \frac{4}{11}\cdot 0 + \frac{7}{11}\cdot 0.58$$

$$= 0.37$$

0.58

$$IG(Y|X_1) = H(Y) - H(Y|X_1) = 0.99 - 0.37 = 0.62$$

Example Information Gain



PICK FEATURE max IG

SELECT X_1 – BETTER SPLIT

Learning Decision Trees

ROOT: ALL FEATURES ARE AVAILABLE

- Start from empty decision tree
- Split on **next best attribute (feature)**
  - Use, for example, information gain to select attribute: GINI INDEX

$$\boxed{\arg\max_i IG(X_i)} = \arg\max_i H(Y) - H(Y \mid X_i)$$

- Recurse : SUBSET OF FEATURES

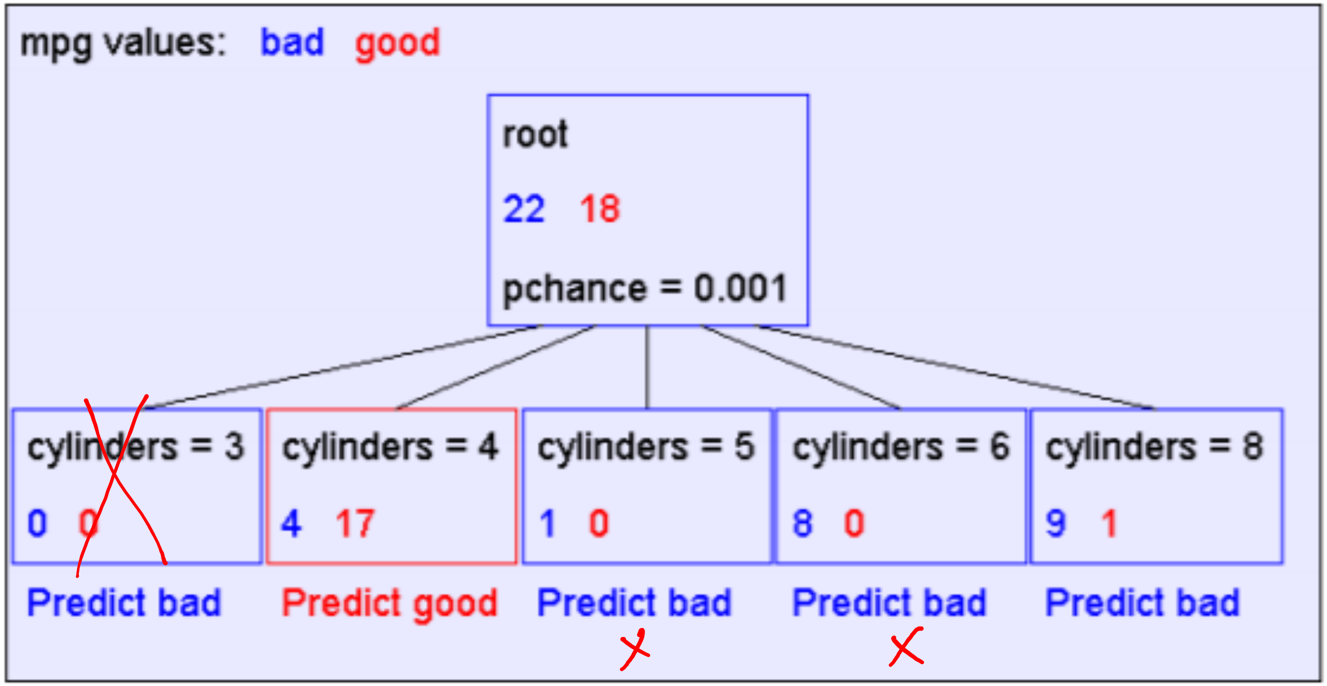
MAX. HEIGHT OF DECISION TREE : NUMBER OF FEATURES

ID3 algorithm uses Information Gain

Information Gain reduces uncertainty on Y

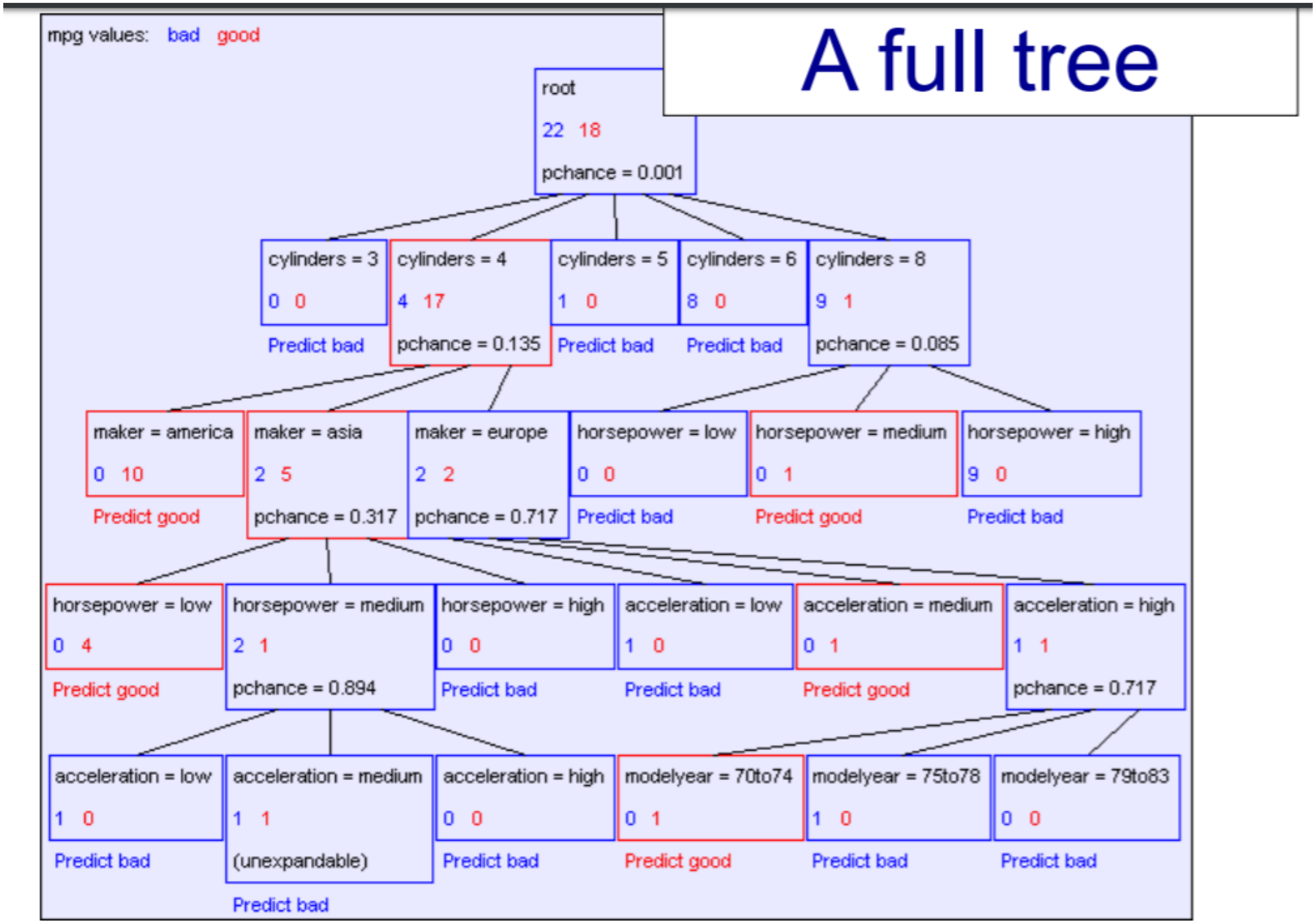
OVERFITTING

When to stop?

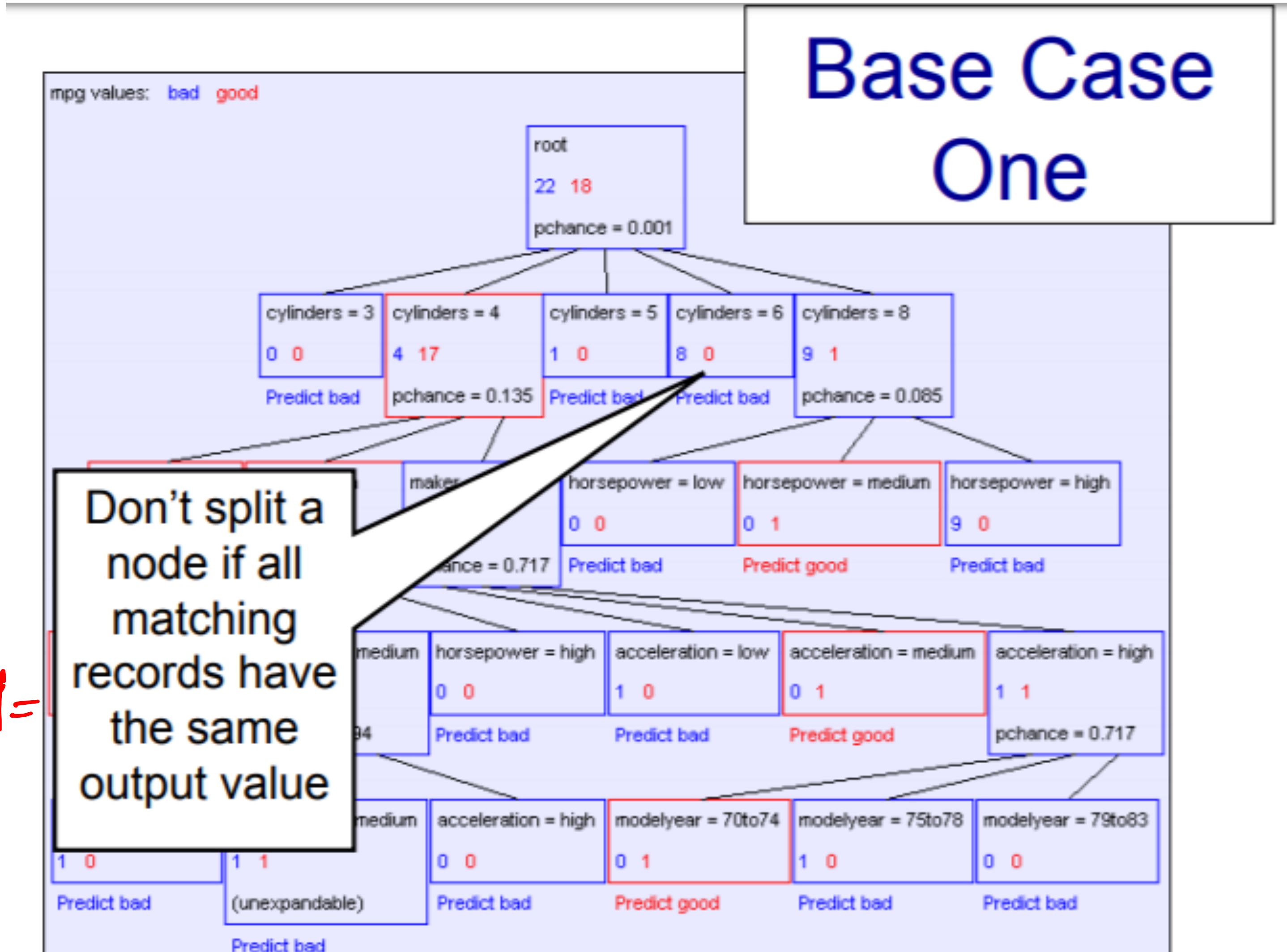


First split looks good! But, when do we stop?

Full Tree

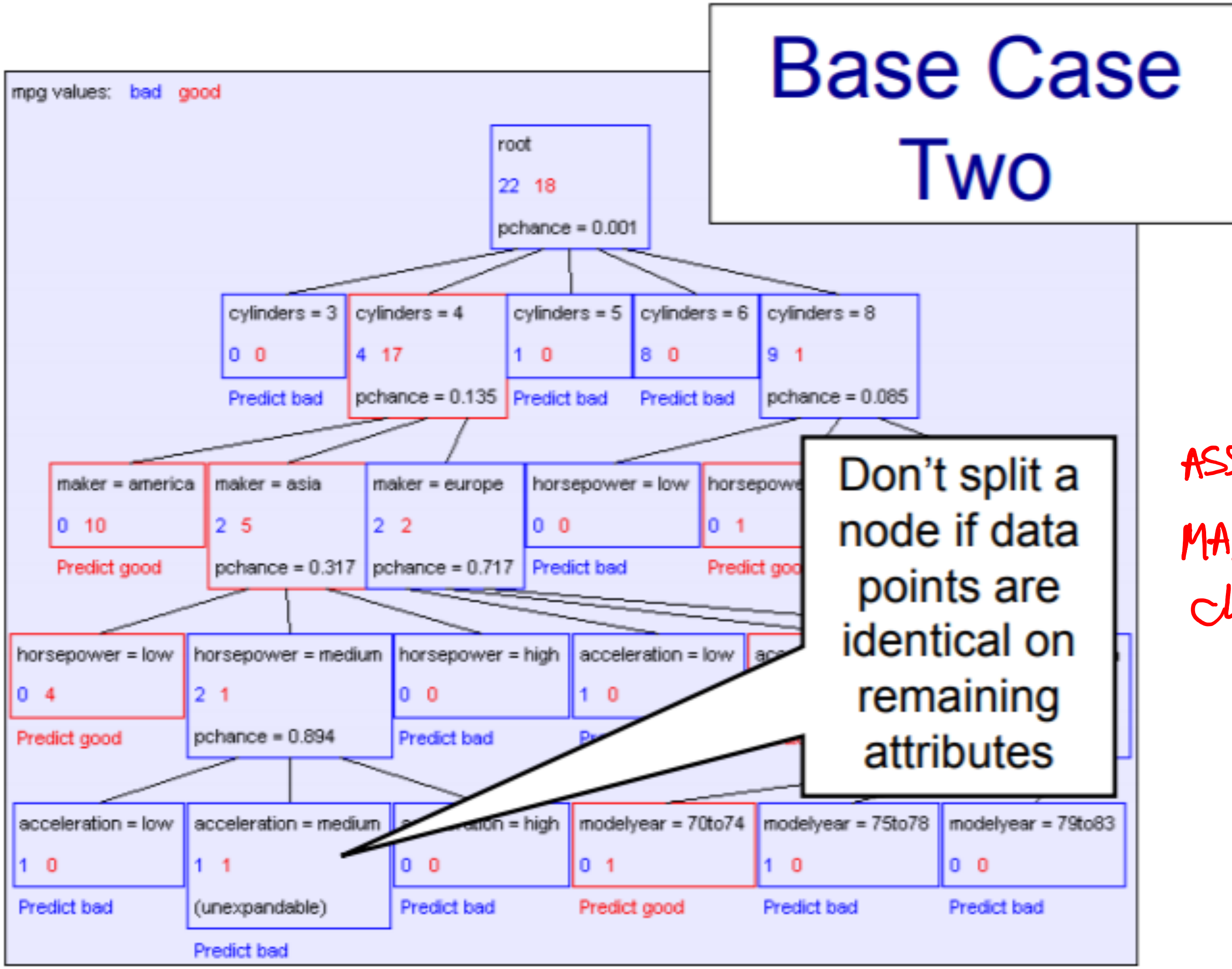

A full tree
mpg values: bad good
root
22 18
pchance = 0.001
cylinders = 3
0 0
Predict bad
cylinders = 4
4 17
pchance = 0.135
cylinders = 5
1 0
Predict bad
cylinders = 6
8 0
Predict bad
cylinders = 8
9 1
pchance = 0.085
maker = america
0 10
Predict good
maker = asia
2 5
pchance = 0.317
maker = europe
2 2
pchance = 0.717
horsepower = low
0 0
Predict bad
horsepower = medium
0 1
Predict good
horsepower = high
9 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
0 1
Predict good
acceleration = high
1 1
pchance = 0.717
acceleration = low
1 0
Predict bad
acceleration = medium
1 1
(unexpandable)
Predict bad
acceleration = high
0 0
Predict bad
modelyear = 70to74
0 1
Predict good
modelyear = 75to78
1 0
Predict bad
modelyear = 79to83
0 0
Predict bad

Case 1



ENTROPY = 0

Case 2



ASSIGN MAJORITY CLASS

Decision Trees

TRAINING SET

LABELS

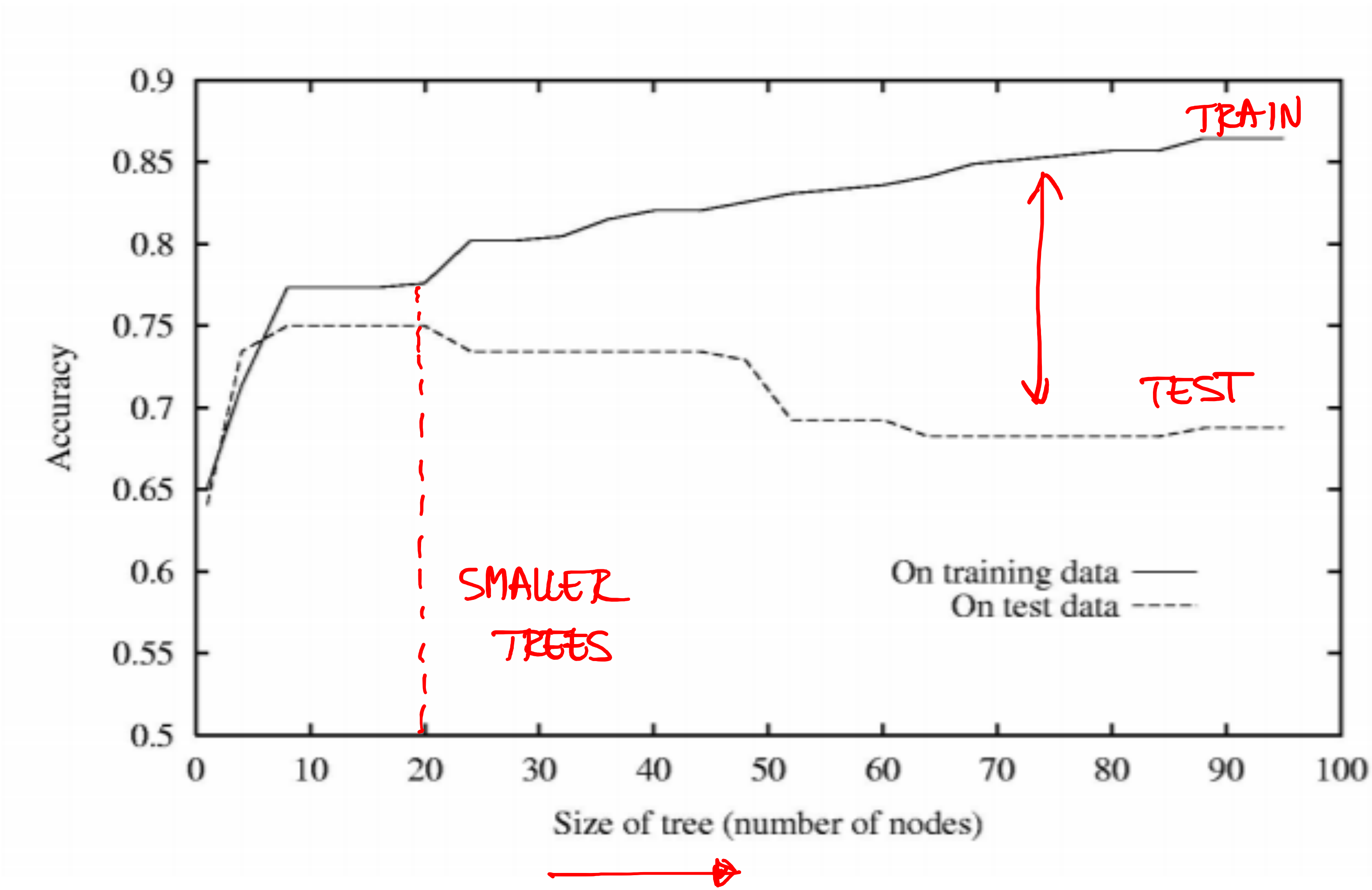
BuildTree(*DataSet*,*Output*)

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

    BuildTree(DS_i,*Output*)

  Where DS_i contains the records in DataSet where X = *i*th value of X.

Overfitting

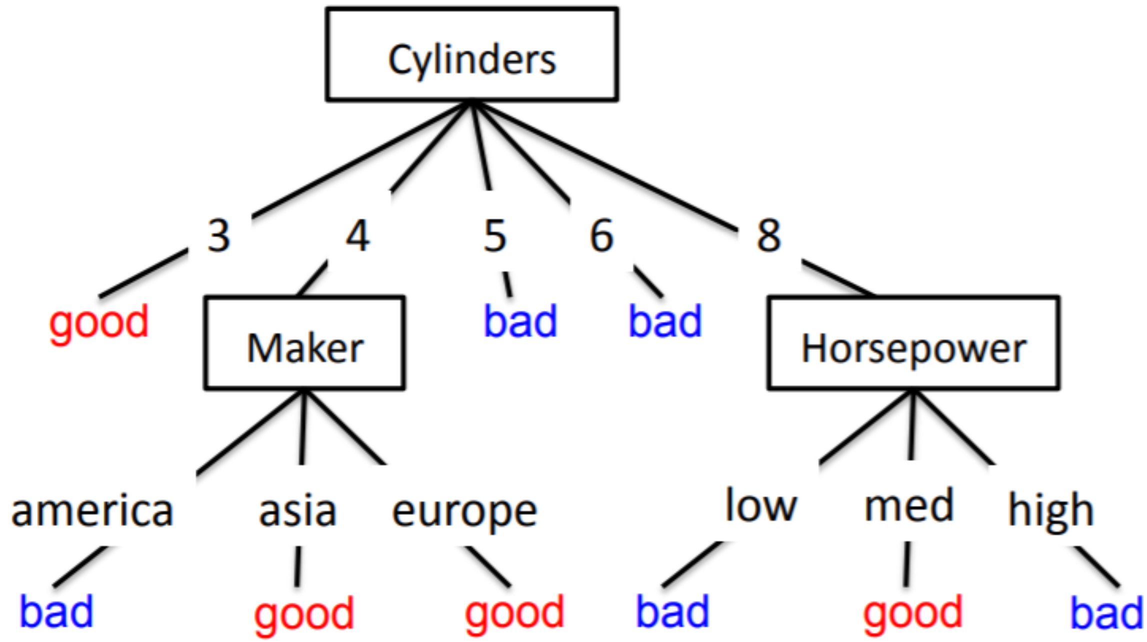


Solutions against Overfitting

- Standard decision trees have no learning bias
  - Training set error is always zero!
    - (If there is no label noise)
  - Lots of variance
  - Must introduce some bias towards simpler trees
- Many strategies for picking simpler trees
  - Fixed depth
  - Minimum number of samples per leaf
- Pruning
  - Remove branches of the tree that increase error using cross-validation

Interpretability

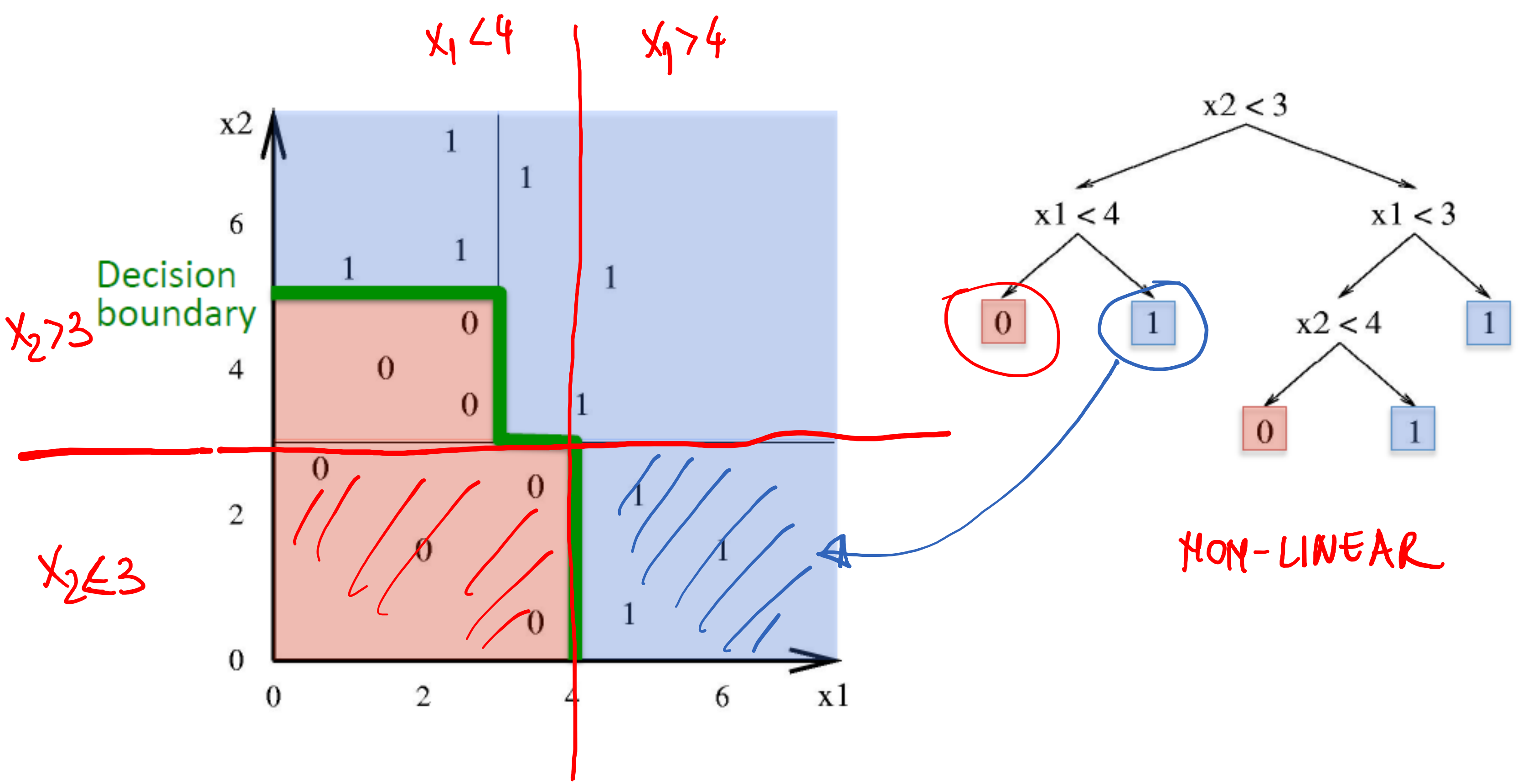
- Each internal node tests an attribute x_i
- One branch for each possible attribute value $x_i=v$
- Each leaf assigns a class y
- To classify input x: traverse the tree from root to leaf, output the labeled y



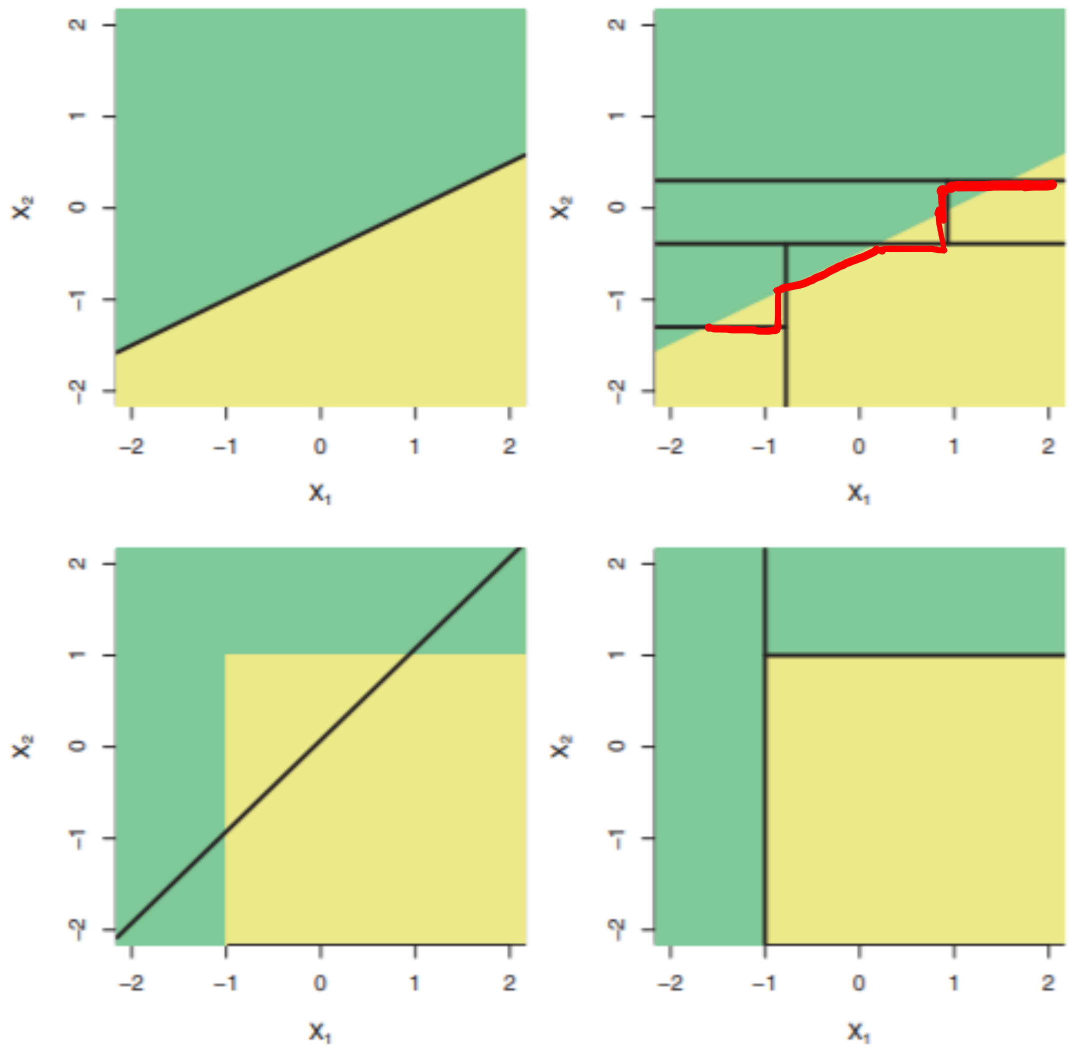
Human interpretable!

Decision Boundary

- Decision trees divide the feature space into axis-parallel (hyper-)rectangles
- Each rectangular region is labeled with one label



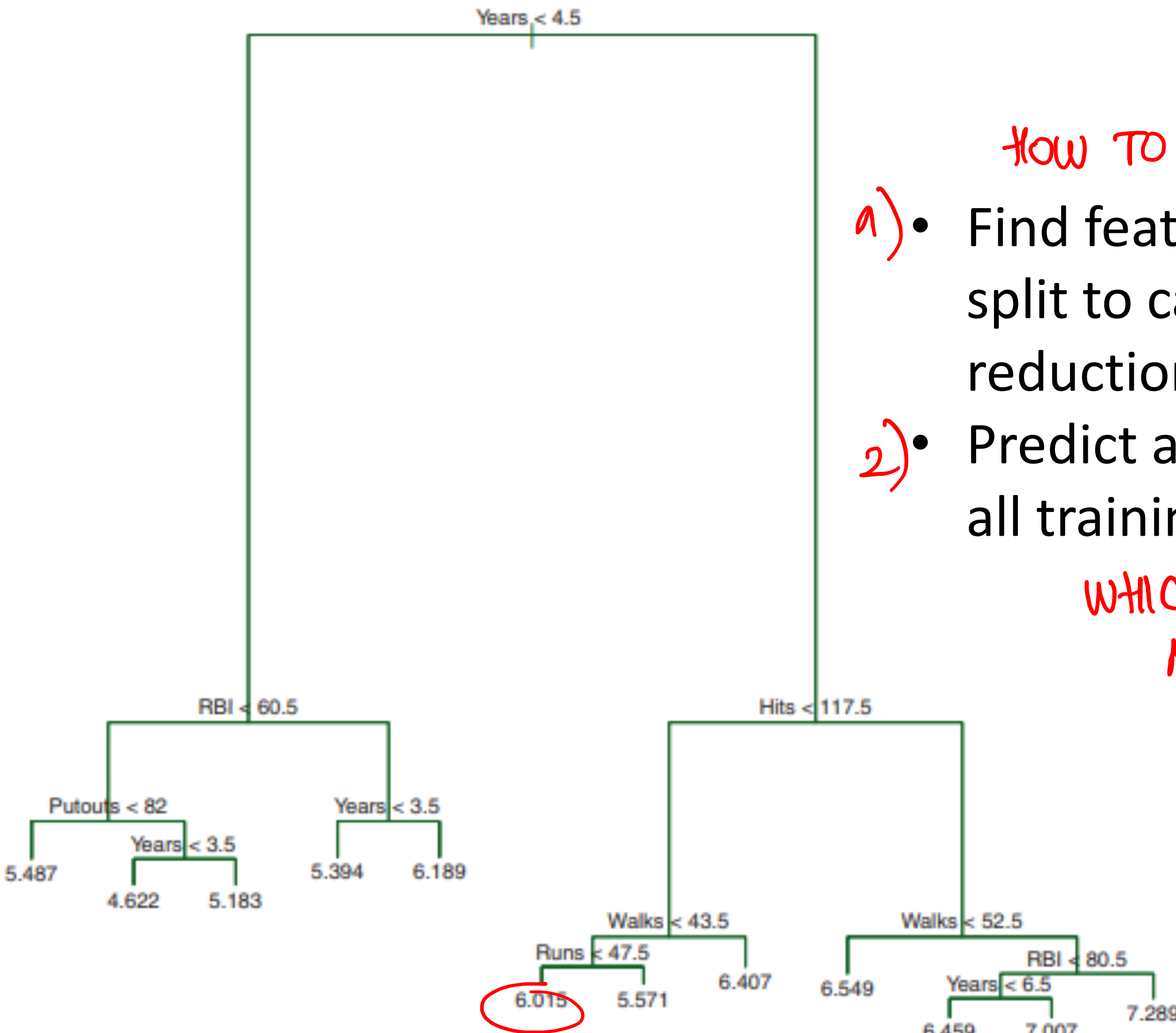
Decision Trees vs Linear Models



Linear model   Decision tree

LINEARLY SEPARABLE

Regression Trees



10 TRAINING EXAMPLE

HOW TO SPLIT

1) • Find feature and value to split to cause the maximum reduction in MSE

2) • Predict average response of all training data at each leaf

WHICH PREDICTION TO MAKE AT EACH LEAF

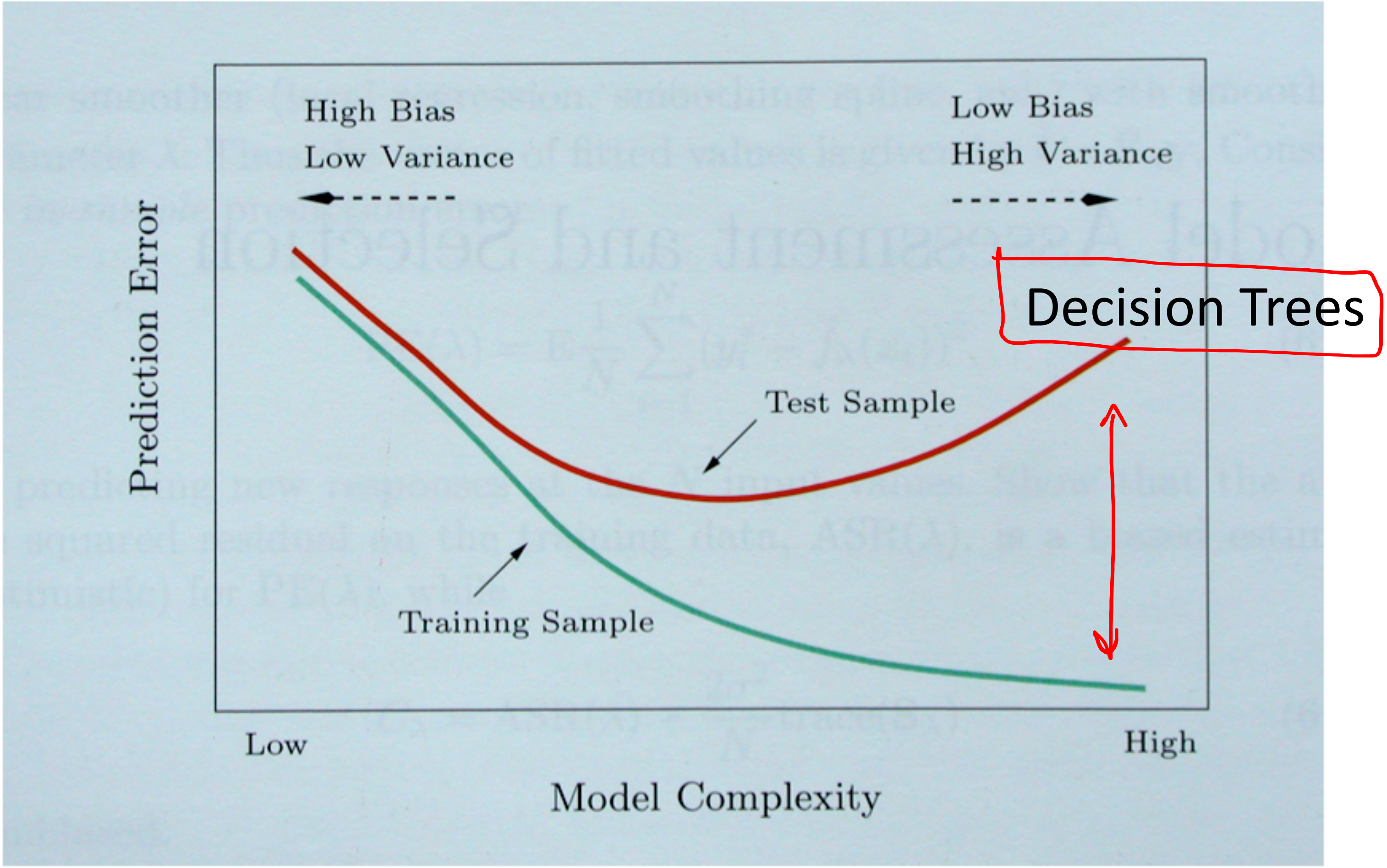
Summary Decision Trees

- Representation: decision trees
- Bias: prefer small decision trees
- Search algorithm: greedy
- Heuristic function: information gain or information content or others
- Overfitting / pruning

Strengths
- Fast to evaluate
- Interpretable
- Generate rules
- Supports categorical and numerical data

Weaknesses
- Overfitting
- Splitting method might not be optimal
- Accuracy is not always high
- Batch learning

Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001

How to reduce variance of single decision tree?

Ensemble Learning

Consider a set of classifiers $h_1, \ldots, \mathrm{h}_\mathrm{L}$

Idea: construct a classifier $H(\mathbf{x})$ that combines the individual decisions of $h_1, \ldots, \mathrm{h}_\mathrm{L}$

- e.g., could have the member classifiers vote, or
- e.g., could use different members for different regions of the instance space

Successful ensembles require **diversity**

- Classifiers should make different mistakes
- Can have different types of base learners

Build Ensemble Classifiers

- Basic idea
  - Build different “experts”, and let them vote
- Advantages
  - Improve predictive performance
  - Easy to implement
  - No too much parameter tuning
- Disadvantages
  - The combined classifier is not transparent and interpretable
  - Not a compact representation

Practical Applications

Goal: predict how a user will rate a movie

- Based on the user's ratings for other movies
- and other peoples' ratings
- with no other information about the movies

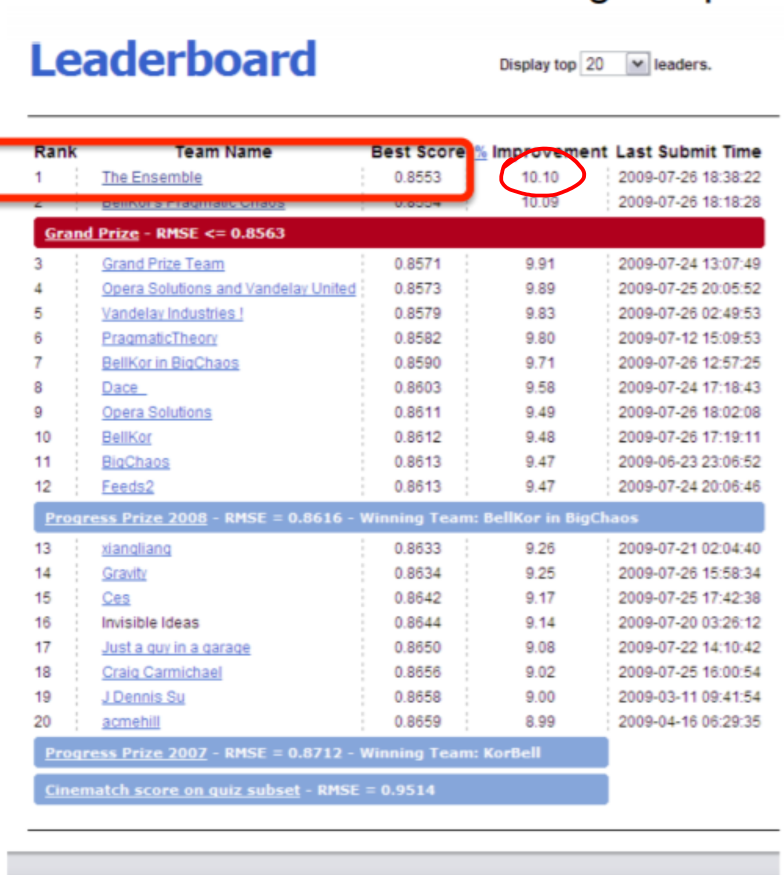
This application is called "collaborative filtering"

Netflix Prize: $1M to the first team to do 10% better then Netflix' system (2007-2009)

Winner: BellKor's Pragmatic Chaos – an ensemble of more than 800 rating systems

Netflix Prize

Machine learning competition with a $1 million prize



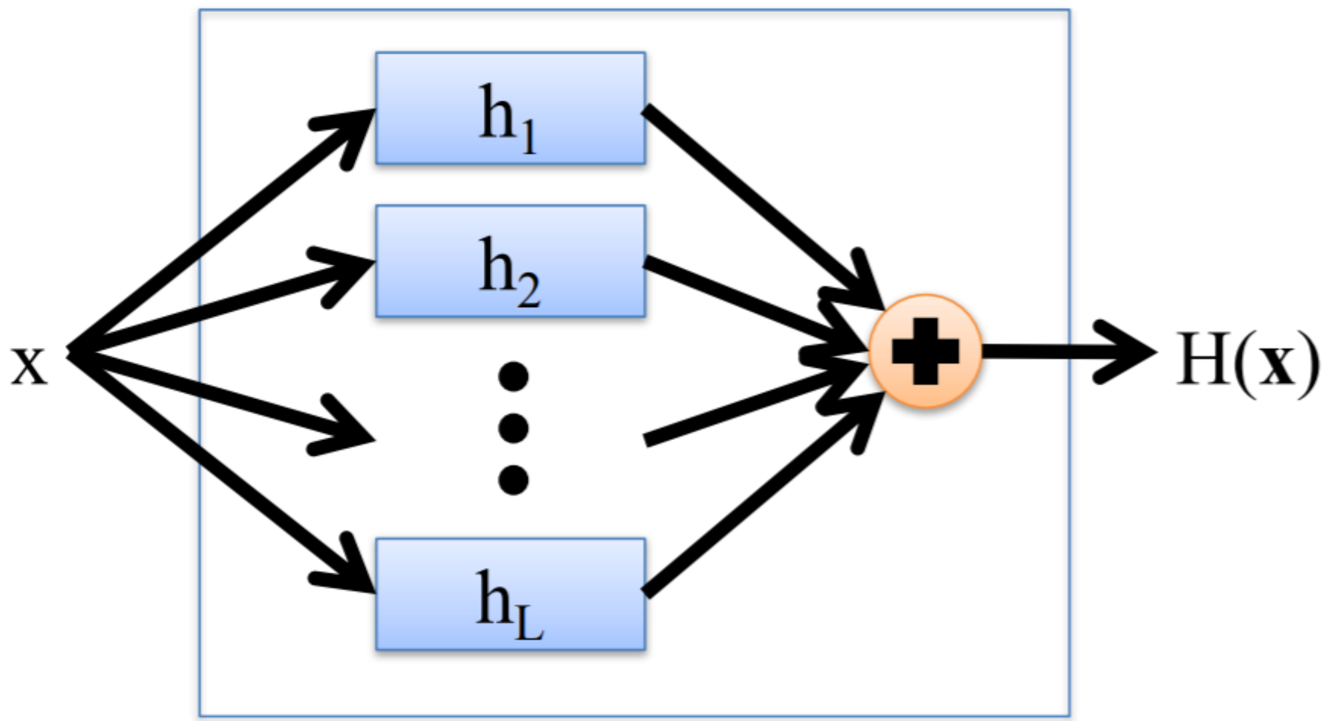
Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28
Grand Prize - RMSE <= 0.8563				
3	Grand Prize Team	0.8571	9.91	2009-07-24 13:07:49
4	Opera Solutions and Vandelay United	0.8573	9.89	2009-07-25 20:05:52
5	Vandelay Industries !	0.8579	9.83	2009-07-26 02:49:53
6	PragmaticTheory	0.8582	9.80	2009-07-12 15:09:53
7	BellKor in BigChaos	0.8590	9.71	2009-07-26 12:57:25
8	Dace_	0.8603	9.58	2009-07-24 17:18:43
9	Opera Solutions	0.8611	9.49	2009-07-26 18:02:08
10	BellKor	0.8612	9.48	2009-07-26 17:19:11
11	BigChaos	0.8613	9.47	2009-06-23 23:06:52
12	Feeds2	0.8613	9.47	2009-07-24 20:06:46
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8633	9.26	2009-07-21 02:04:40
14	Gravity	0.8634	9.25	2009-07-26 15:58:34
15	Ces	0.8642	9.17	2009-07-25 17:42:38
16	Invisible Ideas	0.8644	9.14	2009-07-20 03:26:12
17	Just a guy in a garage	0.8650	9.08	2009-07-22 14:10:42
18	Craig Carmichael	0.8656	9.02	2009-07-25 16:00:54
19	J Dennis Su	0.8658	9.00	2009-03-11 09:41:54
20	acmehill	0.8659	8.99	2009-04-16 06:29:35
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell				
Cinematch score on quiz subset - RMSE = 0.9514				

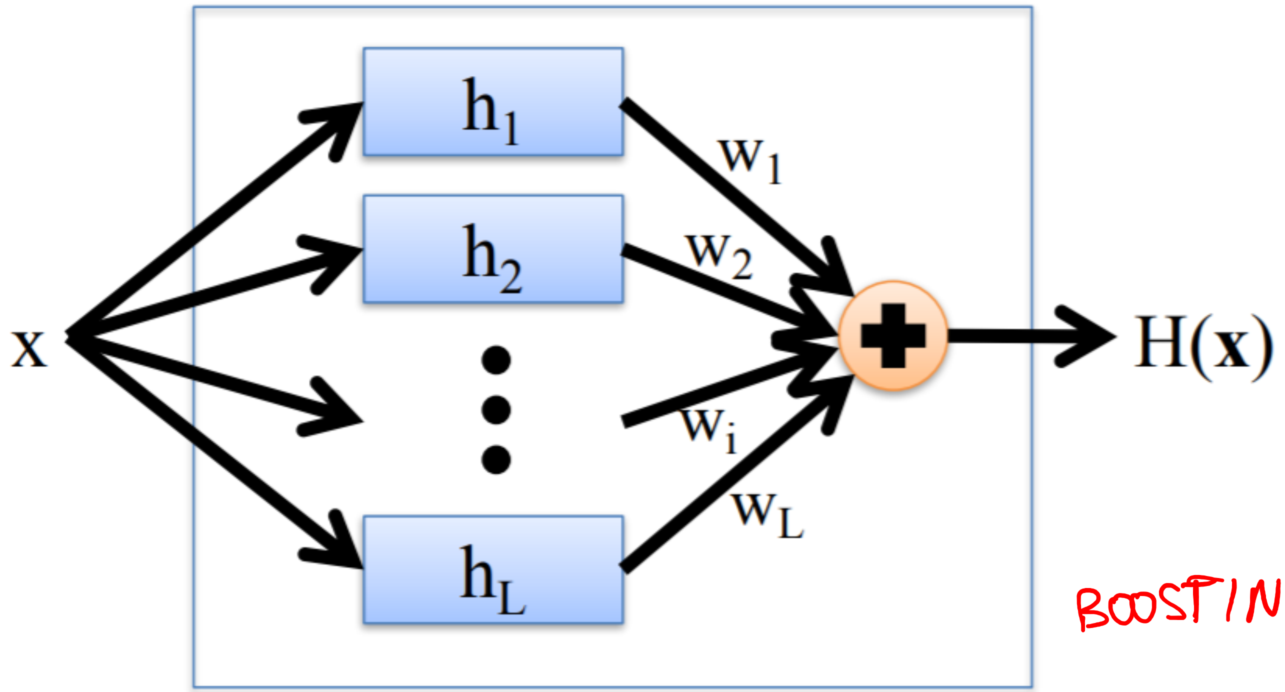
movies \ users						
	1		?	3	5	?
	?	1				2
		4		4	5	?

Combining Classifiers: Averaging



- Final hypothesis is a simple vote of the members

Combining Classifiers: Weighted Averaging



BOOSTING

- Coefficients of individual members are trained using a validation set

Reduce error

- Suppose there are 25 base classifiers

MAJ. VOTE

- Each classifier has error rate, $\varepsilon = 0.35$
- Assume independence among classifiers
- Probability that the ensemble classifier makes a wrong prediction:

ε^{13}

≥ 13 MODELS MAKE AN ERROR

$$P[\text{ENSEMBLE MAKES ERROR}] = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} \cdot (1-\varepsilon)^{25-i} \approx 0.06$$

$i = \#$ CLASSIFIERS WRONG

$\binom{25}{i}$ SETS

$\underbrace{0\ 0\ \ldots\ 0}_{i \text{ WRONG}}\ \underbrace{0\ \ldots\ 0}_{25-i \text{ CORRECT}}$

Reduce Variance

ENSEMBLES:

- REDUCE ERROR
- REDUCE VAR.

ASSUMPTION: MODELS ARE INDEPENDENT

REALITY: ONE TRAINING SET

HOW TO ACHEIVE DIVERSITY?

Acknowledgements

- Slides made using resources from:
  - Andrew Ng
  - Eric Eaton
  - David Sontag
- Thanks!