
Chapter 3: Decision Tree Learning (part 2)

CS 536: Machine Learning
Littman (Wu, TA)

Administration

Books?

Two on reserve in the math library.

iCML-03: instructional Conference on
Machine Learning

mailing list! (mailing Weka
instructions)

What is ID3 Optimizing?

How would you find a tree that minimizes:

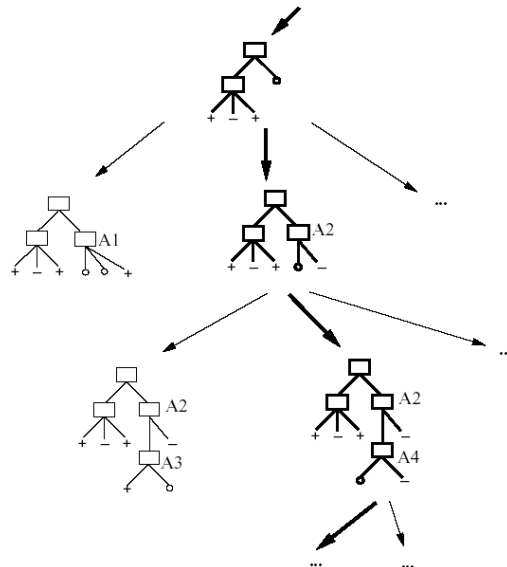
- misclassified examples?
- expected entropy?
- expected number of tests?
- depth of tree given a fixed accuracy?
- etc.?

How decide if one tree beats another?

Hypothesis Space Search by ID3

ID3:

- representation
: trees
- scoring
: entropy
- search
: greedy



Hypothesis Space Search by ID3

- Hypothesis space is complete!
 - Target function surely in there...
- Outputs a single hypothesis (which one?)
 - Can't play 20 questions...
- No back tracking
 - Local minima...
- Statically-based search choices
 - Robust to noisy data...
- Inductive bias \approx "prefer shortest tree"

Inductive Bias in ID3

Note H is the power set of instances X

- Unbiased?

Not really...

- Preference for short trees, and for those with high information gain attributes near the root
- Bias is a *preference* for some hypotheses, rather than a *restriction* of hypothesis space H
- Occam's razor: prefer the shortest hypothesis that fits the data

Occam's Razor

Why prefer short hypotheses?

Argument in favor:

- Fewer short hyps. than long hyps.
 - a short hyp that fits data unlikely to be coincidence
 - a long hyp that fits data might be coincidence

Argument opposed:

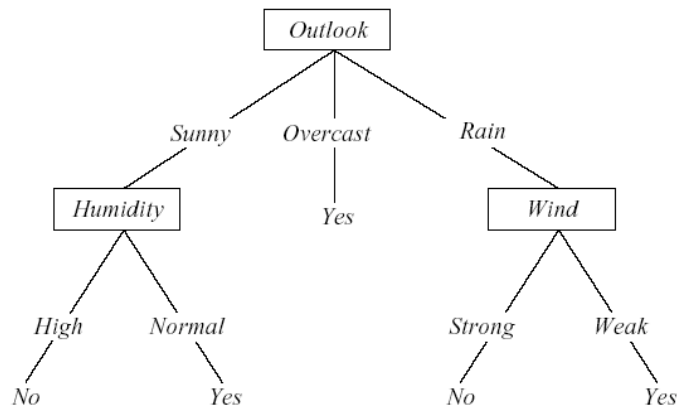
- There are many ways to define small sets of hyps
- e.g., all trees with a prime number of nodes that use attributes beginning with "Z"
- What's so special about small sets based on size of hypothesis??

Overfitting

Consider adding noisy training example #15:

Sunny, Hot, Normal, Strong, PlayTennis = No

What effect on earlier tree?



Overfitting

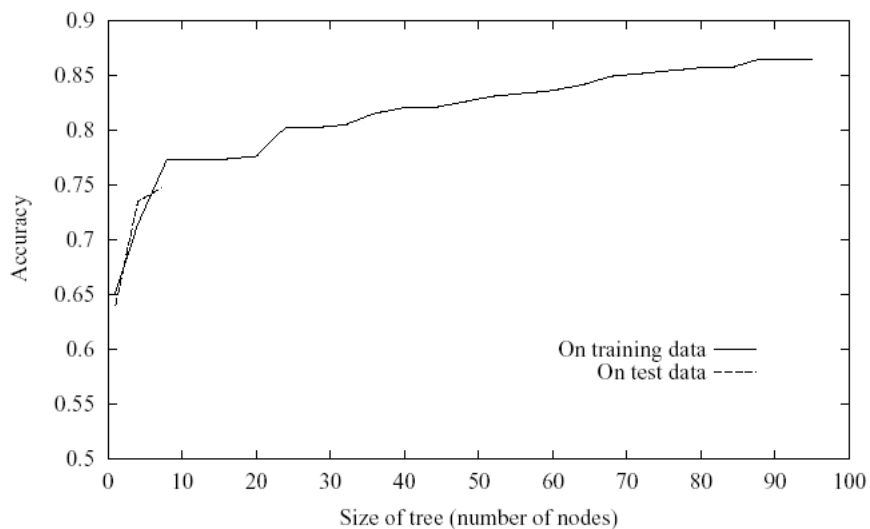
Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution D of data: $error_D(h)$

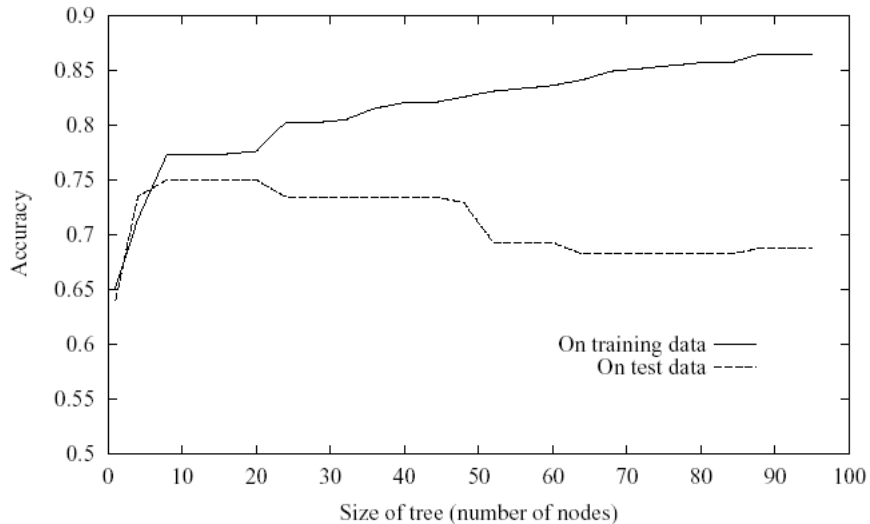
Hypothesis h in H **overfits** training data if there is an alternative hypothesis h' in H such that

- $error_{train}(h) < error_{train}(h')$, and
- $error_D(h) > error_D(h')$

Overfitting in Learning



Overfitting in Learning



Avoiding Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune (DP alg!)

How to select “best” tree:

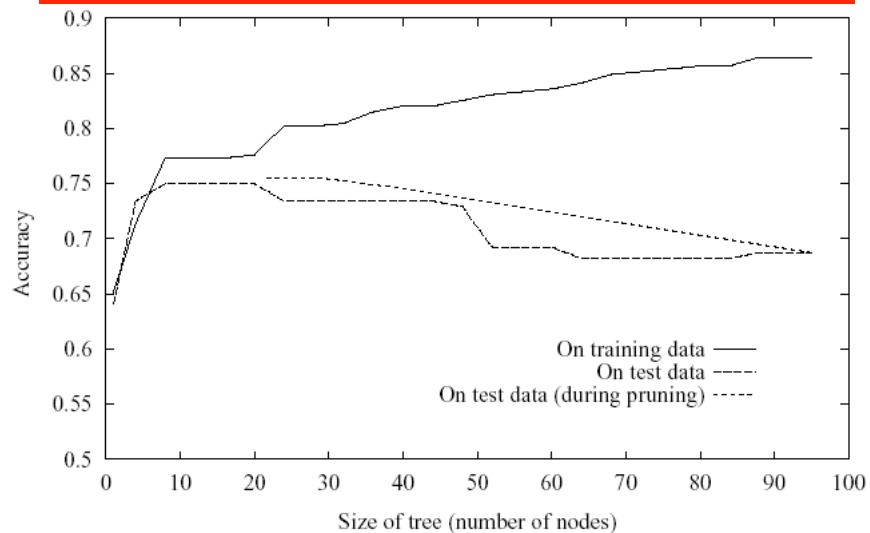
- Measure performance over training data
- Measure performance over separate validation data set
- MDL: minimize
 $size(tree) + size(misclassifications(tree))$

Reduced-Error Pruning

Split data into *training* and *validation* set
Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
 2. Greedily remove the one that most improves *validation* set accuracy
- produces smallest version of most accurate subtree
 - What if data is limited?

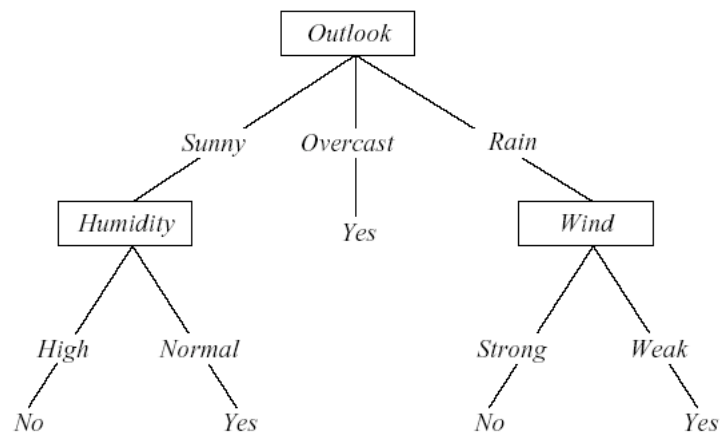
Effect of Pruning



Rule Post-Pruning

1. Convert tree to equivalent set of rules
 2. Prune each rule independently of others
 3. Sort final rules into desired sequence for use
- Perhaps most frequently used method (e.g., C4.5)

Converting Tree to Rules



The Rules

IF (Outlook = Sunny) \wedge (Humidity = High)

THEN PlayTennis = No

IF (Outlook = Sunny) \wedge (Humidity = Normal)

THEN PlayTennis = Yes

...

Continuous Valued Attributes

Create a discrete attribute to test
continuous

- $Temp = 82.5$
- $(Temp > 72.3) = t, f$

<i>Temp:</i>	40	48	60	72	80	90
<i>PlayTennis:</i>	No	No	Yes	Yes	Yes	No

Attributes with Many Values

Problem:

- If one attribute has many values compared to the others, *Gain* will select it
- Imagine using *Date = Jun_3_1996* as attribute

One approach: use *GainRatio* instead

$$\text{GainRatio}(S,A) \equiv \text{Gain}(S,A) / \text{SplitInfo}(S,A)$$

$$\text{SplitInfo}(S,A) \equiv -\sum_{i=1}^c |S_i|/|S| \log_2 |S_i|/|S|$$

where S_i is subset of S for which A has value v_i

Attributes with Costs

Consider

- medical diagnosis, *BloodTest* has cost \$150
- robotics, *Width_from_1ft* has cost 23 sec.

How to learn a consistent tree with low expected cost? Find min cost tree.

Another approach: replace gain by

- Tan and Schlimmer (1990)

$$\text{Gain}^2(S,A) / \text{Cost}(A)$$

- Nunez (1988) [w in $[0,1]$: importance]

$$(2^{\text{Gain}(S,A)} - 1) / (\text{Cost}(A) + 1)^w$$

Unknown Attribute Values

Some examples missing values of A ?

Use training example anyway, sort it

- If node n tests A , assign most common value of A among other examples sorted to node n
- assign most common value of A among other examples with same target value
- assign probability p_i to each possible value v_i of A (perhaps as above)
 - assign fraction p_i of example to each descendant in tree
- Classify new examples in same fashion