

CSCE 478/878 Lecture 3: Learning Decision Trees

Stephen D. Scott
(Adapted from Tom Mitchell's slides)

September 10, 2008

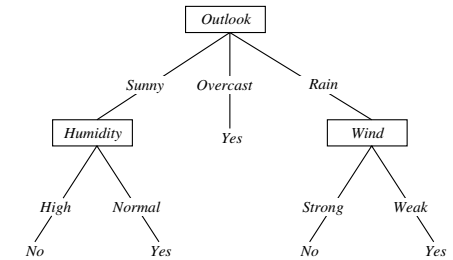
1

Outline

- Decision tree representation
- ID3 learning algorithm
- Entropy, Information gain
- Overfitting and pruning
- Continuous, many-valued, unknown, and cost-associated attributes

2

Decision Tree for *PlayTennis*



3

Decision Tree Representation

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent:

- $\wedge, \vee, \text{XOR}$
- $(A \wedge B) \vee (C \wedge \neg D \wedge E)$

4

When to Consider Decision Trees

- Instances describable by attribute–value pairs
- Target function is discrete-valued
- Disjunctive hypothesis may be required
- Possibly noisy training data
- Human readability of result is important

Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences

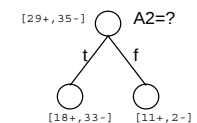
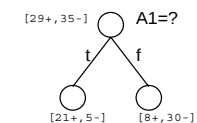
5

Top-Down Induction of Decision Trees (ID3 Algorithm, Table 3.1)

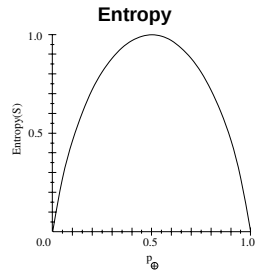
Main loop:

1. $A \leftarrow$ the “best” decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A , create new descendant of *node*
4. Sort (partition) training examples over children based on A 's value
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



6



- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S
- $p_{\ominus}$ is the proportion of negative examples in S
- **Entropy** measures the impurity of S

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

7

Entropy (cont'd)

- $Entropy(S)$ = expected number of bits needed to encode class ($\oplus$ or $\ominus$) of randomly drawn member of S (under the optimal, shortest-length code)

Why?

- Information theory: optimal length code assigns $-\log_2 p$ bits to message having probability p

- So, expected number of bits to encode $\oplus$ or $\ominus$ of random member of S :

$$p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus})$$

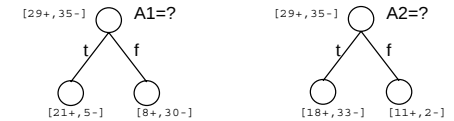
$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

8

Information Gain

- $Gain(S, A)$ = expected reduction in entropy due to partitioning on A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



9

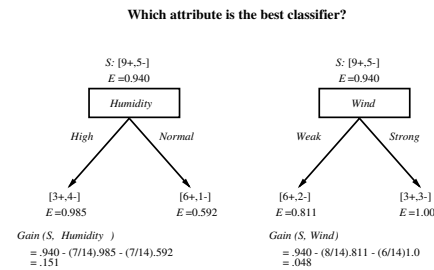
Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

10

Selecting the First Attribute

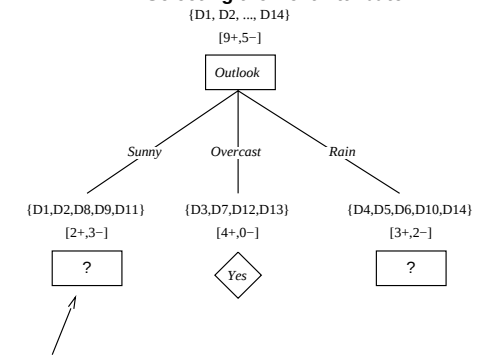
- Comparing *Humidity* to *Wind*:



- Other gain values: $Gain(S, Outlook) = 0.246$,
 $Gain(S, Temperature) = 0.029$

11

Selecting the Next Attribute



Which attribute should be tested here?

$S_{Sunny} = \{D1, D2, D8, D9, D11\}$

$$Gain(S_{Sunny}, Humidity) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970$$

$$Gain(S_{Sunny}, Temperature) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$Gain(S_{Sunny}, Wind) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

12

The diagram shows a binary tree structure representing a decision process. At the top is the root node, which branches into three nodes. The middle node branches into two nodes, labeled A1 and A2. Node A2 branches into two nodes, labeled A3 and A4. Each internal node has two outgoing arrows, one solid and one dashed. Leaf nodes are represented by circles containing '+' or '-' signs.

Hypothesis Space Search by ID3 (cont'd)

- 14

- Note H is the power set of instances X

- Not really:

- 15

Why prefer short hypotheses?

- Fewer short hyps. than long hyps.

⇒ a long hyp that fits data might be coincidence

- Are many ways to define small sets of hyps

- 16

- Consider adding noisy training example #15:

- What effect on earlier tree?

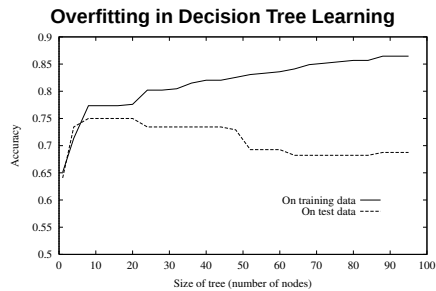


- Consider error of hypothesis h over

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

and

18



19

Avoiding Overfitting

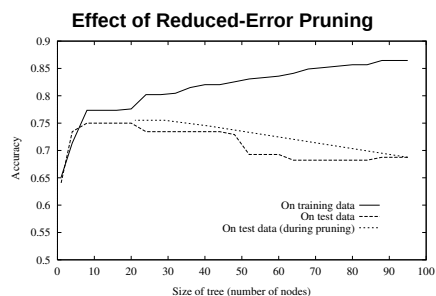
- How can we avoid overfitting?
 - Stop growing when data split doesn't help
 - Grow full tree, then post-prune
- How to select “best” tree:
 - Measure performance over training data
 - Measure performance over separate validation data set
 - MDL (minimum description length principle): minimize $size(tree) + size(misclassifications(tree))$ based on some size measure of trees and examples (Chapt. 6)

20

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 (with respect to validation set)
- What if data is limited?

21



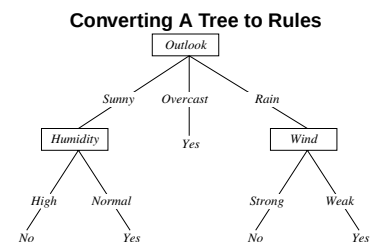
22

Rule Post-Pruning

1. Convert tree to equivalent set of rules
2. Prune each rule independently of others by removing selected preconditions (the ones that improve accuracy the most)
3. Sort final rules into desired sequence for use

Perhaps most frequently used method (e.g. C4.5)

23



IF $(Outlook = Sunny) \wedge (Humidity = High)$
 THEN $PlayTennis = No$

IF $(Outlook = Sunny) \wedge (Humidity = Normal)$
 THEN $PlayTennis = Yes$

...

24

Continuous-Valued Attributes

Use threshold to map continuous to boolean, e.g.
 $(Temperature > 72.3) \in \{t, f\}$

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

- Can show that threshold maximizing inf. gain must lie between two adjacent attribute values in training set such that label changed, so try all such values, e.g. $(48 + 60)/2 = 54$ and $(80 + 90)/2 = 85$
- Now (dynamically) replace continuous attribute with boolean attributes $Temperature_{>54}$ and $Temperature_{>85}$ and run algorithm normally
- Other options: Split into multiple intervals rather than two; use thresholded linear combinations of continuous attributes

25

Attributes with Many Values

Problem:

- If attribute has many values, *Gain* will select it
- E.g. if *Date* is attribute, inf. gain will be high because several very small subsets will be created

One approach: use *GainRatio* instead:

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

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

where S_i is subset of S for which A has value v_i (measures how broadly and uniformly A splits data)

26

Attributes with Costs

- Medical diagnosis, *BloodTest* has cost \$150
- Robotics, *Width_from_1ft* has cost 23 sec.

How to learn a consistent tree with low expected cost?

One approach: replace gain by

- Tan and Schlimmer (1990)

$$\frac{Gain^2(S, A)}{Cost(A)}.$$

- Nunez (1988)

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

where $w \in [0, 1]$ determines importance of cost

27

Unknown Attribute Values

What if some examples are missing values of A ?

Use them anyway (sift each through tree)

- If node n tests A , assign most common value of A among other training examples sifted to node n
- Assign most common value of A among other examples with same target value (either overall or at node n)
- Assign probability p_i to each possible value v_i of A
 - Assign fraction p_i of example to each descendant in tree

Classify new examples in same fashion

28

Topic summary due in 1 week!

29