Data Mining, Data Warehouse

Decision Tree Learning

Lesson #4

Attribute Selection Measure in ID3
Information Gain

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

Select the attribute *A* that best classifies the *examples*

…An Example

Weekend (Example)	Weather	Parents	Money	Decision (Class)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Adopted from: http://www.doc.ic.ac.uk/~sgc/teaching/v231/lecture11.html

Using the ID3 algorithm to build a decision tree

- Figure out which attribute will be put into the node at the top of our tree:

  Weather, *parents* or *money*.

- To do this, we first need to calculate:

Entropy(S) =

$-p_{cinema} \log_2(p_{cinema}) -p_{tennis} \log_2(p_{tennis}) -p_{shopping} \log_2(p_{shopping}) -p_{stay\_in} \log_2(p_{stay\_in})$

$= -(6/10) * \log_2(6/10) -(2/10) * \log_2(2/10) -(1/10) * \log_2(1/10) -(1/10) * \log_2(1/10)$

$= 1.571$

Table

Using the ID3 algorithm… (cont.)

Now lets calculate the gain for the *weather* attribute:

Gain(S, weather) = 1.571 - ($|S_{sunny}|$/10)*Entropy(S_{sunny}) - ($|S_{windy}|$/10)*Entropy(S_{windy}) - ($|S_{rainy}|$/10)*Entropy(S_{rainy})

Where,

Entropy(S_{sunny}) = $-p_{cinema|sunny}\log_2(p_{cinema|sunny})-p_{tennis|sunny}\log_2(p_{tennis|sunny})$

$= -(1/3)*\log_2 (1/3)-(2/3)*\log_2 (2/3) = 0.918$

Entropy(S_{windy}) = $-p_{cinema|windy}\log_2(p_{cinema|windy})$ $-p_{shopping|windy}\log_2(p_{shopping|windy})$

$= -(1/4)*\log_2 (1/4)-(3/4)*\log_2 (3/4) = 0.811$

Entropy(S_{rainy}) = $-p_{cinema|rainy}\log_2(p_{cinema|wind})$ $-p_{stay\_in|rainy}\log_2(p_{stay\_in|rainy})$

$= -(2/3)*\log_2 (2/3)-(1/3)*\log_2 (1/3) = 0.918$

Gain(S, weather) = 1.571 - (3/10)*Entropy(S_{sunny}) - (4/10)*Entropy(S_{windy}) - (3/10)*Entropy(S_{rainy}) = **0.69**

Table

Using the ID3 algorithm to build a decision tree (cont.)

Now lets calculate the gain for the *parents* attribute:

Gain(S, parents) = 1.571 - ($|S_{yes}|$/10)*Entropy(S_{yes}) - ($|S_{no}|$/10)*Entropy(S_{no})

Where,

Entropy(S_{yes}) = $-p_{cinema|yes}\log_2(p_{cinema|yes})$

$= -(5/5)*\log_2(5/5)= 0$

Entropy(S_{no}) = $-p_{tennis|no}\log_2(p_{tennis|no}) - p_{stay\_in|no}\log_2(p_{stay\_in|no})$
$-p_{cinema|no}\log_2(p_{cinenema|no})\ p_{shopping|no}\log_2(p_{shopping|no})$

$= -(2/5)*\log_2 (2/5)-(1/5)*\log_2 (1/5)- (1/5)*\log_2 (1/5)-(1/5)*\log_2(1/5) = 1.922$

Gain(S, parents) = 1.571 - (5/10)*Entropy(S_{yes}) - (5/10)*Entropy(S_{no}) = **0.61**

Table

Using the ID3 algorithm… (cont.)

Now lets calculate the gain for the *money* attribute:

$\text{Gain}(S, \text{money}) = 1.571 - (|S_{rich}|/10)*\text{Entropy}(S_{rich}) - (|S_{poor}|/10)*\text{Entropy}(S_{poor})$

Where,

$\text{Entropy}(S_{rich}) = -p_{cinema|rich}\log_2(p_{cinema|rich}) - p_{tennis|rich}\log_2(p_{tennis|rich})$
$- p_{stay\_in|rich}\log_2(p_{stay\_in|rich}) - p_{shopping}\log_2(p_{shopping})$
$= -(3/7)*\log_2(3/7)-(2/7)*\log_2(2/7)-(1/7)*\log_2(1/7)-(1/7)*\log_2(1/7)= 1.842$

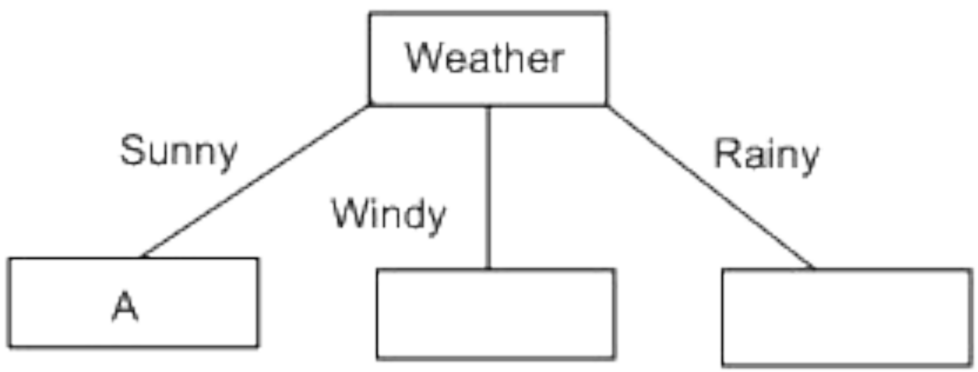
$\text{Entropy}(S_{poor}) = -p_{cinema|no}\log_2(p_{cinema|no})$
$= -(3/3)*\log_2 (3/3) = 0$

$\text{Gain}(S, \text{money}) = 1.571 - (7/10)*\text{Entropy}(S_{rich}) - (3/10)*\text{Entropy}(S_{poor}) = \mathbf{0.28}$

Table

Using the ID3 algorithm… (cont.)

- The first node in the decision tree will be the *weather* attribute.
- Now we look at the first branch. S_{sunny} = {W1, W2, W10}. This is not empty, so we do not put a default categorization leaf node here.



Using the ID3 algorithm… (cont.)

- Now we have to fill in the choice of attribute A, which we know cannot be weather, because we've already removed that from the list of attributes to use.
- So, we need to calculate the values for Gain(S_{sunny}, parents) and Gain(S_{sunny}, money).

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

Using the ID3 algorithm… (cont.)

- Hence we can calculate:

- Gain(S_{sunny},parents) = 0.918 - (|S_{yes}|/|S|)*Entropy(S_{yes}) - (|S_{no}|/|S|)*Entropy(S_{no})
  = 0.918 - (1/3)*0 - (2/3)*0 = **0.918**

(Entropy(S_{sunny}

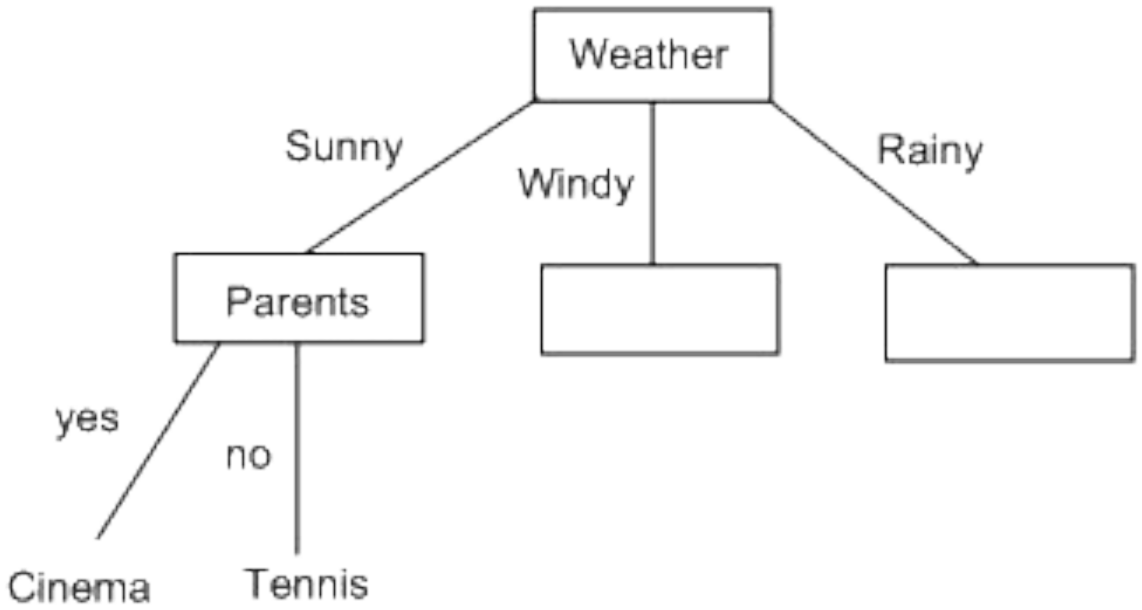
- Gain(S_{sunny},money) = 0.918 - (|S_{rich}|/|S|)*Entropy(S_{rich}) - (|S_{poor}|/|S|)*Entropy(S_{poor})
  = 0.918 - (3/3)*0.918 - 0 = 0.918 - 0.918 = 0

log(1/3) –*1/3-
(2/3*log(2/3

- Note: Entropy(S_{yes}) and Entropy(S_{no}) were both zero, because S_{yes} contains examples which are all in the same category (cinema), and S_{no} contains examples which are all in the same category (tennis).
- This should make it more obvious why we use information gain to choose attributes to put in nodes.

Using the ID3 algorithm… (cont.)

- Given our calculations, attribute A should be taken as *parents*.



Using the ID3 algorithm… (cont.)

- Now we need to calculate the values for Gain(S_{windy}, parents) and Gain(S_{windy}, money).

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W3	Windy	Yes	Rich	**Cinema**
W7	Windy	No	Poor	**Cinema**
W8	Windy	No	Rich	**Shopping**
W9	Windy	Yes	Rich	**Cinema**

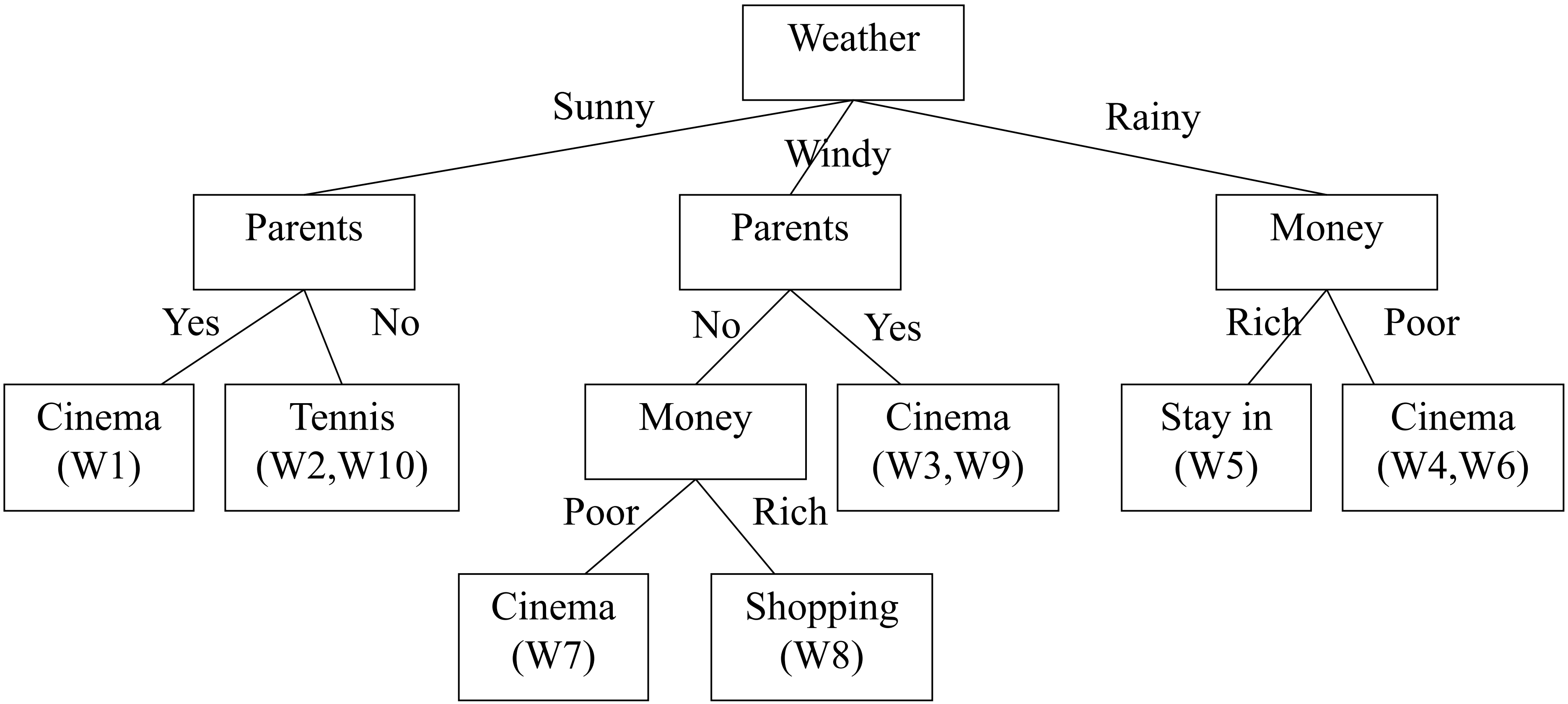
Using the ID3 algorithm… (cont.)

- The calculation:
- $\text{Gain}(S_{windy},\text{parents}) = 0.811 - (|S_{yes}|/|S|)*\text{Entropy}(S_{yes}) - (|S_{no}|/|S|)*\text{Entropy}(S_{no})$
  $= 0.811 - (2/4)*0 - (2/4)*1 =$ **0.311**
- $\text{Gain}(S_{windy},\text{money}) = 0.811 - (|S_{rich}|/|S|)*\text{Entropy}(S_{rich}) - (|S_{poor}|/|S|)*\text{Entropy}(S_{poor})$
  $= 0.811 - (3/4)*0.918 - (1/4)*0 = 0.122$

Meaning that this node will be split by *parents* too.

After calculating the Gain for *rainy* too, the final tree is…

Using the ID3 algorithm… (cont.)



Accuracy Estimation

- Training Accuracy Rate
  - The percentage of *training set* samples that are correctly classified by the model.
- Testing Accuracy Rate
  - The percentage of *test set* samples that are correctly classified by the model.
  - Test set is independent of training set, otherwise over-fitting will occur.
- Majority Rule Accuracy
  - To select a class for a terminal node, select the class A having the most examples (in training set).
  - *Majority rule accuracy* = $|S_A|/|S|$
  - The classification model should be more accurate than the *majority rule.*

Accuracy Estimation

- In our example, training accuracy is 10/10 = 100 %
- Take the following testing set for example:

Weekend (Example)	Weather	Parents	Money	Decision (Class)
W11	Sunny	Yes	Poor	**Cinema**
W12	Sunny	No	Rich	**Tennis**
W13	Windy	No	Rich	**Cinema**
W14	Rainy	Yes	Poor	**Cinema**

Testing accuracy is: 3/4 = 75%
Majority rule accuracy = 6/10 = 60%

eTre

Accuracy Estimation

- How can we tell if the training accuracy is significantly different than the testing accuracy?
- Normal Approximation to Binomial Distribution
  H0: The two accuracies are equal. We reject H0 if:

$$\hat{p} > p + Z \cdot \sqrt{\frac{p(1-p)}{n}}$$

number of rows in the testing set

Testing accuracy

$$1 > p + 1.96 \cdot \sqrt{\frac{p(1-p)}{4}}$$

If training accuracy > testing accuracy

Over-fitting

We can check in the same manner, majority rule vs. testing accuracy

p=0.51 □ Testing accuracy should be 51% at the most, for the two accuracies to be significantly different (to reject H0).