Decision Trees

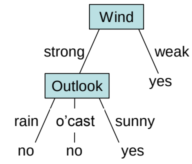
Keith Cheverst

Overview...

Target Function:
When do I play golf on a Saturday?

- Inductive Method
  - Generalises to learn rules from observed "training" data
    - IF the Wind is weak THEN PlayGolf is yes
    - IF the Outlook is overcast AND the wind is strong THEN PlayGolf is no
  - May turn out to be wrong when used on further data
    - Next Sunday the wind is strong and the outlook overcast but I do play golf...
  - DTs can be used for working out whether you get that cheap loan...

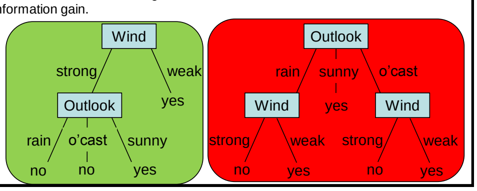
Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no



How do we produce Decision Trees?

- Basic Algorithm is recursive...
  - Determine which of the candidate attributes to use as the top most node of the decision tree (candidates being "Wind" or "Outlook" in the first step using this example)
    - do this by calculating the information gain for each candidate attribute
      - pick the attribute with the highest information gain.

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no



Information Gain

- To understand Information Gain we firstly have to understand the concept of Entropy...

Entropy

"characterizes the impurity of an arbitrary collection of examples" [Mitchell,1997]

- So if the impurity or randomness of a collection (with respect to the target classifier) is high then the entropy is high
- But if there is no randomness (complete uniformity with respect to the target classifier) then the entropy is zero

Entropy(S) ≡ $-p_+ \log_2 p_+ \ - \ p_- \log_2 p_-$

Where target classification is boolean
p_+ :the proportion of positive examples in collection S
p_- :the proportion of negative examples in collection S

Entropy(S) ≡ $-p_+ \log_2 p_+ \ - \ p_- \log_2 p_-$

Entropy([2+,0-]) = 0
(Max uniformity)

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Strong	Yes

Entropy([1+,1-]) = 1
(Min uniformity)

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Weak	No

Entropy([2+,1-]) = 0.92
(Working to 2 decimal places)

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Strong	Yes
Sunny	Weak	No

Entropy Example 1...

$$\text{Entropy}(S) \equiv -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy([2+,0-]) $= -(2/2)\log_2(2/2) - (0/2)\log_2(0/2)$
$= -1\log_2(1) - (0/2)\log2(0/2)$
$= -1{*}0 - 0{*}0$
$= 0 - 0$
$= 0$

Recall that $2^0 = 1$

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Strong	Yes

(Max Uniformity, Min Randomness)

Entropy Example 2...

$$\text{Entropy}(S) \equiv -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy([1+,1-]) $= -(1/2)\log_2(1/2) - (1/2)\log_2(1/2)$
$= -0.5 * \log_2(0.5) - 0.5 * \log_2(0.5)$
$= (-0.5 * -1) - (0.5 * -1)$
$= 0.5 - (-0.5)$
$= 1$

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Weak	No

(Max Randomness)

Entropy Example 3...

$$\text{Entropy}(S) \equiv -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy([2+,1-]) $= -(2/3)\log_2(2/3) - (1/3)\log_2(1/3)$
$= -0.67 * \log_2(0.67) - (0.33) * \log_2(0.33)$
$= -(0.67 * -0.58) - (0.33 * -1.6)$
$= 0.39 - (-0.53)$
$= 0.92$

Outlook	Wind	WearCoat
Rain	Weak	Yes
Sunny	Strong	Yes
Sunny	Weak	No

N.B. For reasons of speed/brevity, I'm just working to 2 decimal places!

Entropy of our Play Golf collection...

$$\text{Entropy}(S) \equiv -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy([4+,3-]) $= -(4/7)\log_2(4/7) - (3/7)\log_2(3/7)$
$= -0.57 * \log_2(0.57) - (0.43) * \log_2 (0.43)$
$= -0.57 * (-0.81) - (0.43) * (-1.22)$
$= 0.46 - (-0.52)$
$= 0.98$

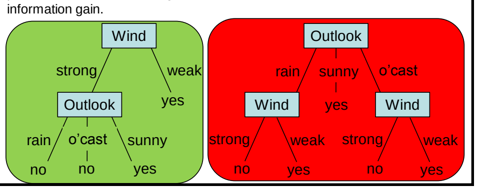
(High randomness so would expect high entropy)

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

How do we produce decision trees?

- Basic Algorithm is recursive...
  - Determine which attribute to use as the top most node of the decision tree (candidates being "Wind" or "Outlook" in the first step using this example)
    - do this by calculating the information gain for each candidate attribute
      - pick the attribute with the highest information gain.

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no



Step 1.

- Determine which attribute to use as the top most node of the decision tree – do this by calculating the information gain for each candidate attribute – pick the attribute with the highest information gain.

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

The information Gain of attribute **A** in collection **S** where **Values(A)** is the set of possible values for attribute **A** and S_v is the subset of **S** for which attribute **A** has the value **v**

$$\textbf{Entropy(S)} \equiv -p_+\log_2 p_+ - p_-\log_2 p_-$$

Where target classification is boolean
p_+ :the proportion of positive examples in collection S
p_- :the proportion of negative examples in collection S

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

Calculating the IG for each candidate...

Gain(S, Outlook) ≡ 0.98 – $\sum_{v\ \epsilon Values(rain,sunny,overcast)}$ (| S_v | / | S |)Entropy(S_v)

Gain(S, Wind) ≡ 0.98 – $\sum_{v\ \epsilon Values(weak,strong)}$ (| S_v | / | S |)Entropy(S_v)

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

Calculating the IG of the Outlook Attribute...

Gain(S, Outlook) ≡ 0.98 – $\sum_{v\ \epsilon Values(rain,sunny,overcast)}$ (| S_v | / | S |) Entropy(S_v)

Entropy(S) ≡ $-p_+\log_2 p_+ - p_-\log_2 p_-$

rain sunny overcast

(3/7)Entropy([1+,2-])+(2/7)Entropy([2+,0-]) +(2/7)Entropy([1+,1-])

(3/7) Entropy([1+,2-]) + (2/7) * 0 + (2/7) * 1
(3/7) * (-(1/3)* $\log_2$ (1/3)) – ((2/3) * $\log_2$ (2/3)) + 0 + (2/7)
0.43 * ((-0.33 * -1.6) – (0.66 * -0.6)) + 0 + 0.29
0.43 * ((0.53) - (-0.4)) + 0 + 0.29
0.43 * (0.93) + 0 + 0.29
0.4 + 0 + 0.29
0.69

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

Gain(S, Outlook) ≡ 0.98 – 0.69
Gain(S, Outlook) ≡ 0.29

Calculating the IG of the Wind Attribute...

Gain(S, Wind) ≡ 0.98 – $\sum_{v\ \epsilon Values(weak,\ strong)}$ (| S_v | / | S |) Entropy(S_v)

Entropy(S) ≡ $-p_+\log_2 p_+ - p_-\log_2 p_-$

weak strong

(3/7)Entropy([3+,0-]) + (4/7)Entropy([1+,3-])

(3/7) * 0.0 + (4/7) Entropy([1+,3-])
0 + (4/7) * (-(1/4) * $\log_2$ (1/4)) – ((3/4) * $\log_2$ (3/4))
0 + 0.57 * (-(0.25 * -2.0) – (0.75 * -0.41))
0 + 0.57 * (-(-0.5) – (-0.31))
0 + 0.57 * ((0.5) – (-0.31))
0 + 0.57 * (0.81)
0 + 0.46
0.46

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no

Gain(S, Wind) ≡ 0.98 – 0.46
Gain(S, Wind) ≡ 0.52

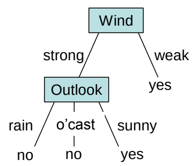
Compare the IGs of each attribute...

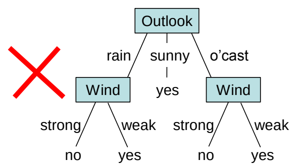
Gain(S, Outlook) ≡ 0.98 – 0.69
Gain(S, Outlook) ≡ 0.29

Gain(S, Wind) ≡ 0.98 – 0.46
Gain(S, Wind) ≡ 0.52

Wind has highest Information Gain so make it top node

Outlook	Wind	PlayGolf
rain	strong	no
sunny	weak	yes
overcast	weak	yes
rain	weak	yes
sunny	strong	yes
rain	strong	no
overcast	strong	no





Let's consider another set of Training Data...

- This time with 15 rows and 4 attributes not including the target attribute 'PlayTennis'
- Note that this is still an extremely small sample of training data – it's not uncommon to run decision tree learning algorithms on census data!
  - See, UCI Machine Learning Repository for much larger set's of training data that you can play with…
  - http://www.ics.uci.edu/~mlearn/MLRepository.html

Sample Training Data…

Day	Outlook	Temp	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

Step 1…

- Which attribute should we have as the top most node of our decision tree
- Determine the information gain for each candidate attribute…
  - Gain(S, Outlook) = 0.246
  - Gain(S, Humidity) = 0.151
  - Gain(S, Wind) = 0.048
  - Gain(S, Temperature) = 0.029
- So would have Outlook as top Node

After Step 1...

Outlook

sunny o'cast rain

yes

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

[2+,3-] ? ← Temp, Humidity or Wind?

[3+,2-] ?

$\text{Gain}(S_{sunny}, \text{Humidity}) = .97 - (3/5)0.0 - (2/5)0.0 = 0.97$
$\text{Gain}(S_{sunny}, \text{Temp}) = .97 - (2/5)0.0 - (2/5)1.0 - (1/5)0.0 = 0.57$
$\text{Gain}(S_{sunny}, \text{Wind}) = .97 - (3/5)0.93 - (2/5)1.0 = 0.019$

$$\text{Entropy}([2+,3-]) = (-(2/5)^* \log_2(2/5)) - ((3/5) * \log_2(3/5))$$
$$(-0.4 * -1.32) - (0.6 * -0.74)$$
$$0.53 + 0.44 = 0.97$$

Again just 2 dec places being used here!

Discussion…

Ubiquitous Computing…

Decision Trees and their potential role in Ubiquitous Computing

- Consider the following problem...
- We wish to learn behaviour patterns of a user around a given task so that we can build a system to provide proactive support for that task.
- Sensors can be used to build up 'Context History' Tables
- Because its proactive we want the user to be able to query the system with questions such as *'why did you do that, what rule where you following?'*

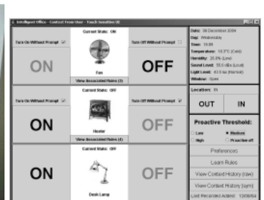
Comprehensibility

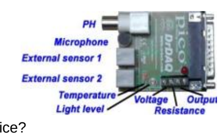
- When discussing key challenges in the ubicomp domain, (Abowd and Mynatt, 2000) make comments concerning the need for comprehensibility, :

  "One fear of users is the lack of knowledge of what some computing system is doing, or that something is being done 'behind their backs'".

Abowd, G.D. and E. D. Mynatt.: 2000, 'Charting Past, Present and Future Research in Ubiquitous Computing'. *ACM Transactions on Computer-Human Interaction, Special Issue on HCI in the New Millenium*, **7**(1), 29-58.

Example Proactive System: IOS…

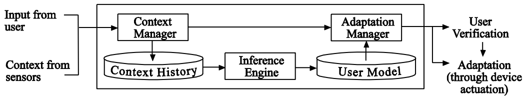




The contexts considered in the experiment were: temperature, humidity, noise level, light level, the status of window, the status of fan and the location of a user.

How does the owner tend to cool his/her office?

Overall approach…

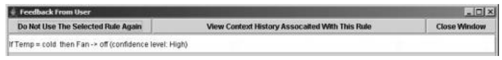


Date	Time	Temp	Noise Level	Humidity	Light	Window	Fan	Heater	Location
2004-26-11	14:36:01	23	55	30	52	closed	off	off	in
2004-26-11	14:37:01	24	55	30	49	closed	on	off	in
2004-26-11	14:38:01	25	55	30	51	closed	on	off	in
2004-26-11	14:39:01	26	55	30	50	closed	on	off	in
2004-26-11	14:40:01	22	68	30	50	closed	off	off	in
2004-26-11	14:41:01	22	62	30	50	closed	off	off	in
2004-26-11	14:42:01	21	55	30	49	closed	off	on	in
2004-26-11	14:43:01	20	55	30	50	closed	off	on	in
2004-26-11	14:44:01	18	55	30	50	closed	off	on	in
2004-26-11	14:45:01	19	76	30	50	closed	off	on	in

Note that 'raw' values would be concerted into symbolic values, e.g., temperatures below 20°C would be classified as cold. Process called *Discretisation*.

When a Rule for Device Actuation is Triggered

- When a suggestion prompt is issued (which occurs if the user has indicated that a prompt rather than automatic action is required) it is displayed on the main control GUI.
  - if the system suggests that the fan should be turned off, then the UI changes to that shown below - the text on the 'OFF' button flashes black and white.



Some Useful Links...

- Reading: Machine Learning, Mitchell, McGrawHill
  - Pages 52 to 63
- UCI Machine Learning Repository
  - http://www.ics.uci.edu/~mlearn/MLRepository.html
- Tools for learning Computational Intelligence
  - http://www.cs.ubc.ca/nest/lci/CIspace/Version4/dTree/
- Log base2 table
  - http://usl.sis.pitt.edu/trurl/log-table.html