CS 367 Tutorial
15 September 2008
Week 7 (tutorial #5)
Carl Schultz

Material is taken from lecture notes (http://www.cs.auckland.ac.nz/compsci367s2c/lectures/index.html).

- Test
    - Tuesday 23 September 2008
    - 12:00pm
    - *(next week)*

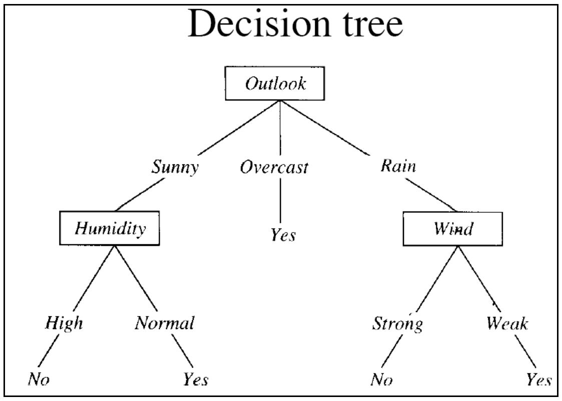
- Assignment
    - Friday 26 September 2008
    - 10:00pm
    - *(next week)*

- Version spaces
    - more good material in the course text book
    - Russell, Norvig "Artificial Intelligence: A modern approach"
    - chapter : "Knowledge in learning" (chap. 19 in second edition) → particularly 19.1 subsection "Current-best-hypothesis search"

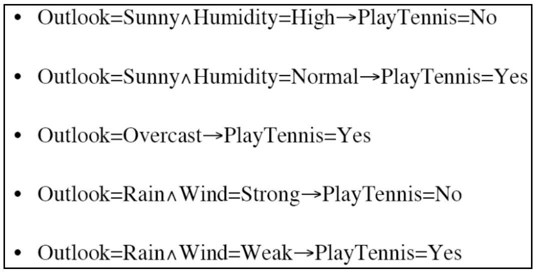
- assignment summary
    - review "experimental design and evaluating hypothesis" lecture notes
    - five datasets: credit-g, heart-c, hepatitis, vowel, and zoo
    - run four algorithms: unpruned decision trees, pruned decision trees, naïve bayes and instance-based learning
    - decide on your experiment methodology (refer to the lecture notes)
    - run a number of experiments – compare different algorithms for each dataset
        - e.g. training versus testing splits, number of runs, statistics taken, …
    - report
        - the task: describe the algorithms, datasets, and parameters
        - explain your experimentation methodology
        - present your experiments and results
            - specific parameters you used for each algorithm
        - analyse results: explain why you think certain algorithms performed better than others

- Weka tutorial
    - "zoo.arff" – what does the data look like?
    - "ExplorerGuide.pdf" (access from "Documentation.html" after installing)

- Decision trees



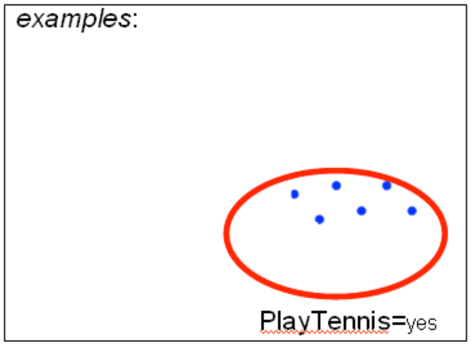
Decision tree



- Outlook=Sunny∧Humidity=High→PlayTennis=No
- Outlook=Sunny∧Humidity=Normal→PlayTennis=Yes
- Outlook=Overcast→PlayTennis=Yes
- Outlook=Rain∧Wind=Strong→PlayTennis=No
- Outlook=Rain∧Wind=Weak→PlayTennis=Yes

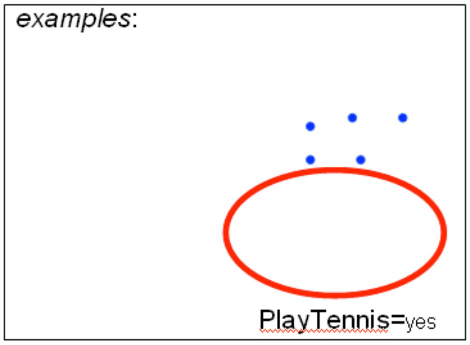
- Entropy
  - firstly consider the following: assume you don't know which objects (e.g. days) are positive examples (days we play tennis) or negative examples (days we don't) – but you do know the ratio of positives to negatives

  - what if all objects are positive examples? e.g.



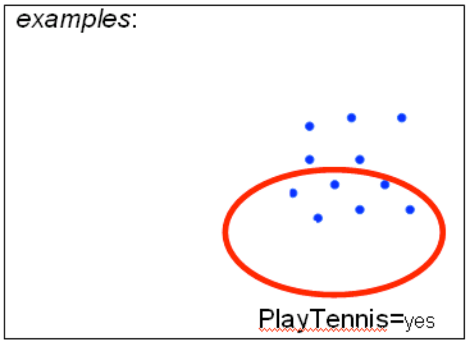
    - …the probability of picking a positive example randomly is 100%
    - so if I tell you – "look, here's a positive example" you don't care, because you already knew that they are all positives

- what if all objects are negative examples? e.g.



  - …similar to before, but the probability of picking a negative example randomly is 100%
  - so if I tell you – “look, here’s a negative example” you don’t care, because you already knew that they are all negatives

- what if the set is a mix of positives and negatives? e.g.



  - …now the probability of picking a positive example randomly is less than 100%
  - so if I tell you – “look, here’s a positive example” you *do* care, because you didn’t know before I showed you
  - this means my comment had some information content

- what if there are 80% positives and 20% negatives?
  - then you can guess that I’ll probably find a positive to show you (but I *might* find a negative)
  - so **on average** the next thing I’ll say will be somewhat informative

- what if there are 50% positives and 50% negatives?
  - you have no idea what I’m about to say; no idea whether I’ll find a positive or negative – it could equally be either
  - **on average** this is the most informative position I can be

- Shannon quantified this with a formula – we can interpret this as:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

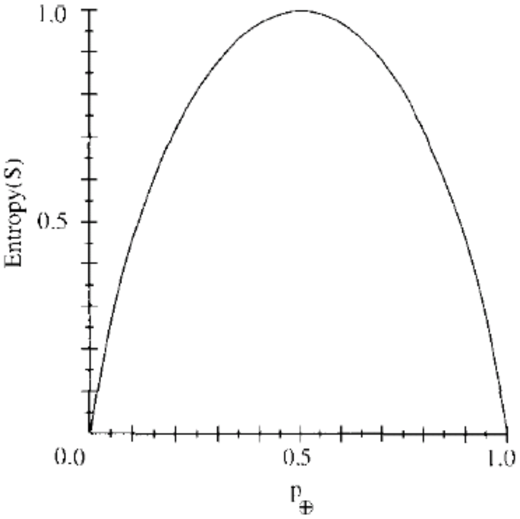
  - p_i …proportion of objects in class *i*
  - c …number of classes (e.g. in our case above c=2)
  - S …set of objects

  - notes:
    - the log of any the fraction between 0 and 1 is negative, so "minus" simply keeps the entropy positive
    - define that 0log0=0 (normally log0 is undefined)

- Boolean case (two values, c=2) is:

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

- …and if we plot this on a graph for ascending values of "P_+" we get:



  - look at the extremes where P_+ is 0.0 and 1.0
    - the entropy is minimum at these points
    - they correspond to the cases where you already knew what I was going to say next (100% certainty, zero surprise, zero information content)

[exercise]

a) Let “animal” be an attribute that can take the value of either “cat” or “dog”. If we have 8 cats and 8 dogs in our set (16 animals all together), what is the entropy wrt the “animal” attribute? *(show working)*

b) Now we only have 8 cats. What is the new entropy? *(show working)*

c) We are classifying days that it will snow. In our dataset we have 12 days when it does snow (positive examples) and 3 days when it doesn’t (negative examples). Calculate the entropy wrt to this attribute “will snow”.

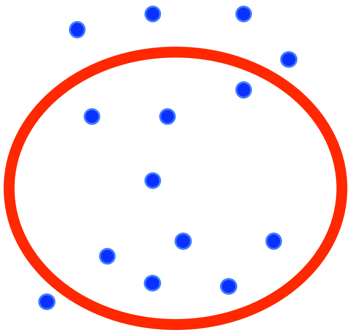
d) We are classifying days when we like to play tennis according to the weather. One attribute to help us is “Wind” which can take the values “strong” or “weak”.

    a. If our dataset has 9 days when we play tennis and 5 days when we don’t, what is the classification entropy?
    b. When we just look at “Wind=weak”, we play tennis on 6 of those days, and don’t play tennis on 2 days. What is the entropy wrt “Wind=weak”?
    c. Calculate the entropy wrt “Wind=strong” *(hint: all days in the dataset either have Wind=weak or Wind=strong…use (a.) and (b.) to figure out the number of positive and negative examples for Wind=strong)*

e) We are classifying sheep as “greedy” and “not greedy” grass eaters (i.e. greedy eaters eat lots of grass). One attribute is “eye colour” that can take the values “brown”, “green” or “blue”. If we have 12 sheep with brown eyes, where 2 of those are greedy eaters (positive examples), calculate the classification entropy wrt to the “brown” eye colour value.

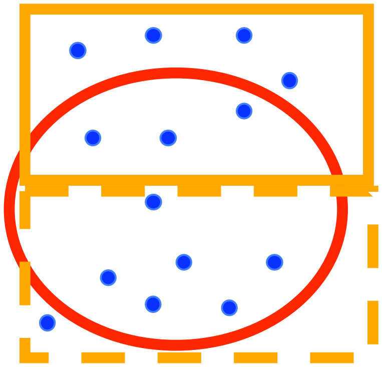
- Information **Gain**
    - Consider the example from lectures. We are trying to classify days on which we play tennis according to a number of attributes about the weather.
    - Which attribute should we choose first? (the root of our decision tree)
        - the one that leaves us with either mostly positives (days we play) or mostly negatives (days we don’t) depending on the value
        - i.e. the one that removes the most uncertainty (entropy) about our classification

- In set S we have 9 positive examples (inside circle) and 5 negative:



- …giving Entropy(S) =-(9/14)$\log_2$(9/14) – (5/14)$\log_2$(5/14) = 0.940

- Now imagine that 3 positive examples and 4 negative examples have "Humidity=high" (top rectangle):



  - entropy S[3+,4-] = 0.985 *(working in lecture notes)*

- …and the rest have "Humidity=normal" (bottom rectangle)
  - entropy S[6+,1-] = 0.592 *(working in lecture notes)*

- If we start from the original dataset where E(S)=0.94, and apply the attribute value "Humidity=normal", the entropy drops to S_v=0.592
  - the difference in entropy is:
    $E(S) – E(S_v) = 0.94 – 0.592 = 0.348$
  - …or our gain in information was 0.348

- But, if instead we apply the other attribute value "Humidity=high", the entropy changes to $E(S_v)$=0.985
  - this time we have more entropy!
  - now the split between positive and negative examples [3+,4-] is more balanced than the original set [9+5-]
  - that's bad – we want more certainty about our classification, not less

- Now let’s generalise for “Humidity” – given a day that we want to classify (as either play tennis day or not), according to our training set:
  - on average if you check the humidity, the probability of the value being “normal” is
    - $|S_{normal}| / |S| = 7/14 = 0.5$
  - and the probability of the humidity being “high” is
    - $|S_{high}| / |S| = 7/14 = 0.5$
  - so that means:
    - 50% of the time our entropy will drop to 0.592 (when humidity=normal), and
    - 50% of the time our entropy will rise to 0.985
  - so, **on average** the amount of entropy we’ll end up with by asking about humidity will be:
    - 50% × 0.592 + 50% × 0.985 = 0.7885
  - …and the gain in information (the decrease in entropy) on average will be:
    - $E(S) - (E(S_{normal})^* |S_{normal}| / |S| + E(S_{high})^* |S_{high}| / |S|)$
    - = 0.94 – 0.7885
    - = 0.1515

- Thus the formula for information gain is:

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

- work through the example from slide 27 of “Decision Tree Learning” in the lecture notes

- Finally, read up on “Inductive bias”
  - preference (search) bias
  - restriction (language) bias
  - Occam’s razor