LECTURE 10: Decision tree learning

Were there some restriction in concept learning?

Example 1: Recall most specific hypothesis for these entries:

							EnjoySport
1.	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2.	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3.	Rainy	Warm	Normal	Strong	Cool	Change	No

S_1: <Sunny, Warm, Normal, Strong, Cool, Change>
S_2: < ? , Warm, Normal, Strong, Cool, Change>
S_3: $\emptyset$

⇒ More <u>expressive</u> hypothesis space needed!

E.g.

<Sunny, ?, ?, ?, ?, ?> $\vee$ <Cloudy, ?, ?, ?, ?, ?>

In concept learning variables can be fixed or free, but this is not sufficient for all tasks (Example 1). Allowing disjunctions ($\vee$) does not work in concept learning, and therefore, we need new <u>learning model</u> and <u>method</u>!

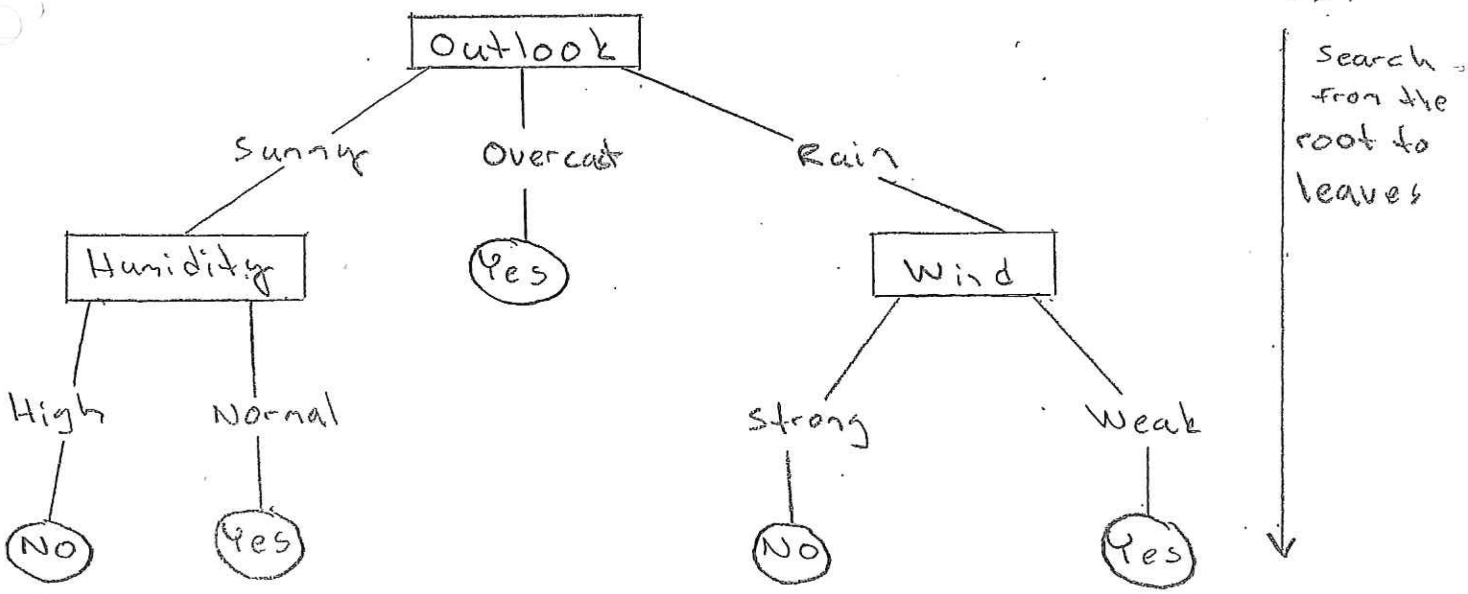
1. Decision tree learning model (representation)
<SLIDE>

Decisio tree represents knowledge (concept) by sorting values down the tree from the root to some leaf node, which provides information if the concept is satisfied or not.

Node: Test of some attribute

Branch: Possible value of an attribute

Example 2 PlayTennis decision tree (from TABLE 3.2 in Mitchell (SLIDE))



Decision trees represent learnt information as a disjunction of conjunctions of constraints on the attribute values of instances

Example 3 PlayTennis as disjunctions of conjunctions (left-first search)

(Outlook == Sunny ∧ Humidity == norma)
∨ (Outlook == Overcast)
∨ (Outlook == Rain ∧ Wind == Weak)

Remember: brute force!

2.ii. Decision tree learning (ID3 algorithm)

It is straightforward to implement a learning method if a single question can be answered: <SLID

"Which attribute should be tested at the root of the tree?"

If this can be solved, then the ID3 algorithm works (See $\boxed{\text{Table 3.1}}$ and note recursive structure)!

Top-Down induction: <SLIDE>

iii. Selecting the best attribute to be tested

We should seek answer from information theory: which attribute provides largest information gain?

⇒ Wanted information is division of example instances to two classes

A perfect attribute would be the one which divides examples to exactly positive and negative examples. The worst attribute holds equally for the both hating them completely mixed.

iii.1 Measure of homogeneity: entropy

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

$p_{\oplus}$ proportion of positive examples

$p_{\ominus}$ proportion of negative examples

Example 4 14 samples including 9 positive and 5 negative ([9+,5-]). Compute entropy (Table 3.2)

$$\text{Entropy}([9+,5-]) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

What does entropy tell? What do maximum and minimum values denote?

$p_\oplus$ and $p_\ominus$ may have values in $[0,1]$ such that

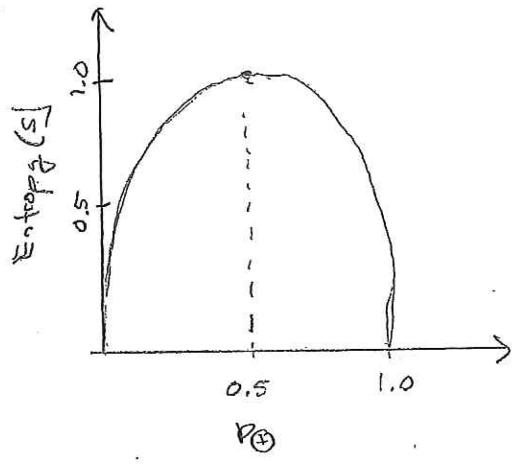
$p_\oplus + p_\ominus = 1$ always.

Min. Entropy(S) = 0, when

$p_\oplus = 1$ or $p_\ominus = 1$ $\left(-1\cdot\underbrace{\log_2(1)}_{=0} - 0\cdot\log_2(0)\right)$

Max. Entropy(S) = 1, when

$p_\oplus = p_\ominus = \frac{1}{2}$ $\left(-\frac{1}{2}\cdot\underbrace{\log_2\frac{1}{2}}_{=-1} - \frac{1}{2}\cdot\log_2(\frac{1}{2})\right)$



Entropy in general represents how many bits on average are required to represent information.

For two classes ($\oplus$ and $\ominus$) the maximum is 1. For C classes $\lceil \log_2 C \rceil$ and the entropy is computed as

$$\text{Entropy}(S) = \sum_{i=1}^{C} -p_i \log_2 p_i$$ [Note relation to probabilities]

III.2 Information gain

Entropy measures non-homogeneity of data, i.e. how mixed the data is. Our goal in best attribute selection is to "un-mix" data. We want to <u>reduce entropy</u>:

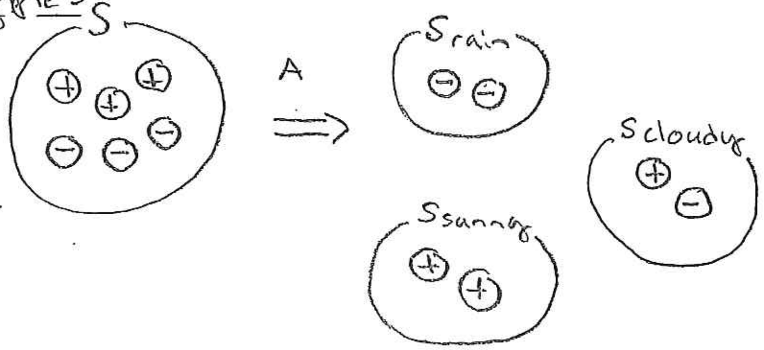
Information gain == reduction in entropy caused by partitioning examples according to given attribute

Definition 1

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

where $Values(A)$ is all possible values of the attribute A. and S_v subset of S for which A has value v.

Example 5



$Entropy(S) = 1.0$

$Entropy(S_{rain}) = 0.0$

$Entropy(S_{sunny}) = 0.0$

$Entropy(S_{cloudy}) = 1.0$

$$Gain(S,A) = 1.0 - 0.0 - 0.0 - \frac{2}{6} \cdot 1.0 = \frac{4}{6} = \underline{\underline{\frac{2}{3}}}$$

Example 6 Computing Gain() (Cont. Example 4)

$Values(Wind) = \{Weak, Strong\}$

$S_{Weak} \leftarrow [6+, 2-]$

$S_{Strong} \leftarrow [3+, 3-]$

$$Gain(S, Wind) = Entropy(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(S) - \frac{8}{14} Entropy(S_{Weak}) - \frac{6}{14} Entropy(S_{Strong})$$

$$= 0.940 - \frac{8}{14} \cdot 0.811 - \frac{6}{14} \cdot 1.00 = \underline{\underline{0.048}}$$

Finally, ID3 algorithm can be implemented using Gain(S,A)-function <SLIDES>

Example 7 ID3 for the data in TABLE 3.2

First step uses whole data S:

$\boxed{Gain(S, Outlook) = 0.246}$, $Gain(S, Humidity) = 0.151$, $Gain(S, Wind) = 0.048$

$Gain(S, Temp) = 0.029$ ... See Figures 3.3 and 3.4

...

Real example: C-Section risk <SLIDES>

3. Issues in DT Learning <SLIDES>

Reduced-error pruning - not very effective

RANDOMISATION IN ML (RANDOM FORESTS)

1. Intro <SLIDES>

2. Decision trees + Randomisation = random forests

   <SLIDES>