3. Decision Trees

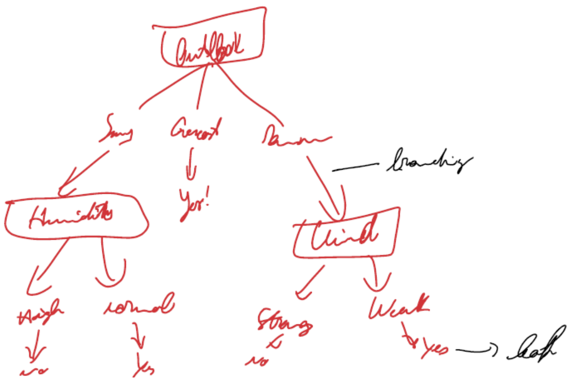
References
T. Mitchell. Machine Learning. Chapter 3

To compute the "best" consistent hypothesis with respect to (wrt) D:

1. Define htpothesis Space H
2. Implement an algorithm that searches for the best hypothesis

Given a discrete input space with m attributes (A1*...*Am) and a classification problem f: X → C
decision tree has 3 characteristics:

1. Internal node → attribute Ai
2. Branch → value of ai,j in Ai
3. Leaf → assign a classification value c in C



Decision trees represent a **disjunction of conjunctions of constraints** on the attribute values of instances.

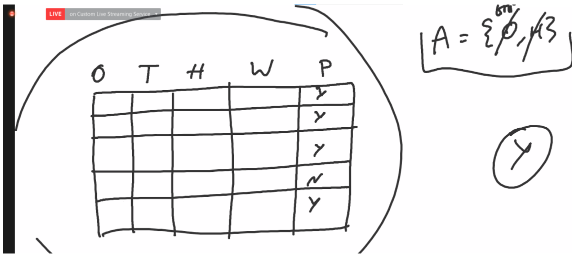
(Outlook = Sunny $\wedge$ Humidity = Normal) $\vee$ (Outlook = Overcast) $\vee$ (Outlook = Rain $\wedge$ Wind = Weak)

A **rule** is generated for **each path to a leaf node.**
IF (Outlook = Sunny) $\wedge$ (Humidity = High)
THEN PlayTennis = No

ID3 Algorithm

1 Create a Root node for the tree
2 if all Examples are **positive** (you always play tennis), then return the node Root with **label +**

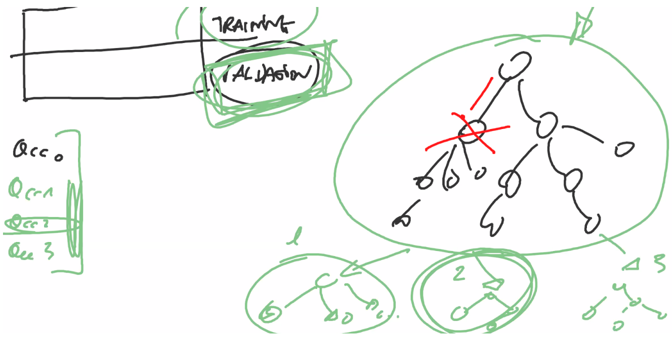
3 if all Examples are **negative**, then return the node Root with **label –**



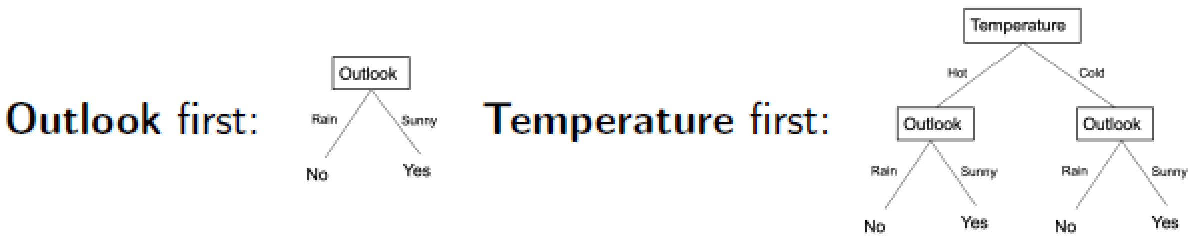
4 if **Attributes** is **empty**, then **return the node Root with label = most common value** of Target attribute in Examples
5 Otherwise

- For each value vi of A
    - if Examples vi is empty then add a leaf node with label = most common value of Target attribute in Examples
    - else
      add the tree ID3(Examplesvi , Target attribute, Attributes–{A})

If there isn't an attribute, it means that we don't consider that attribute important for the choose.



Output tree depends on attribute order



Information gain measures **how well a given attribute separates** the training examples according to their target classification.

ID3 selects the attribute that induces **highest information gain.**

Entropy

Information gain measured as reduction in **entropy** (how much a dataset is impure).

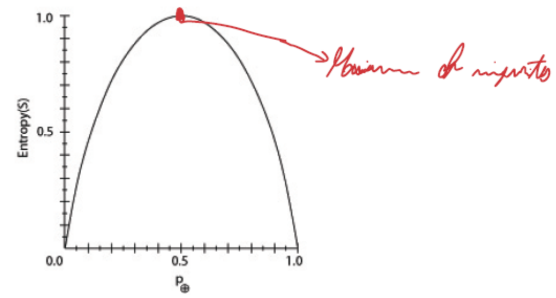
- p(+) is the proportion of positive examples in S (+/N)
- p- (= 1 – p(+)) is the proportion of negative examples in S
- Entropy measures the impurity of S

Example

Consider the set S = [9+, 5−]

Entropy(S) = −(9/14)log2(9/14) – (5/14)log2(5/14) = 0.940

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



In case of multi-valued target functions (c-wise classification)

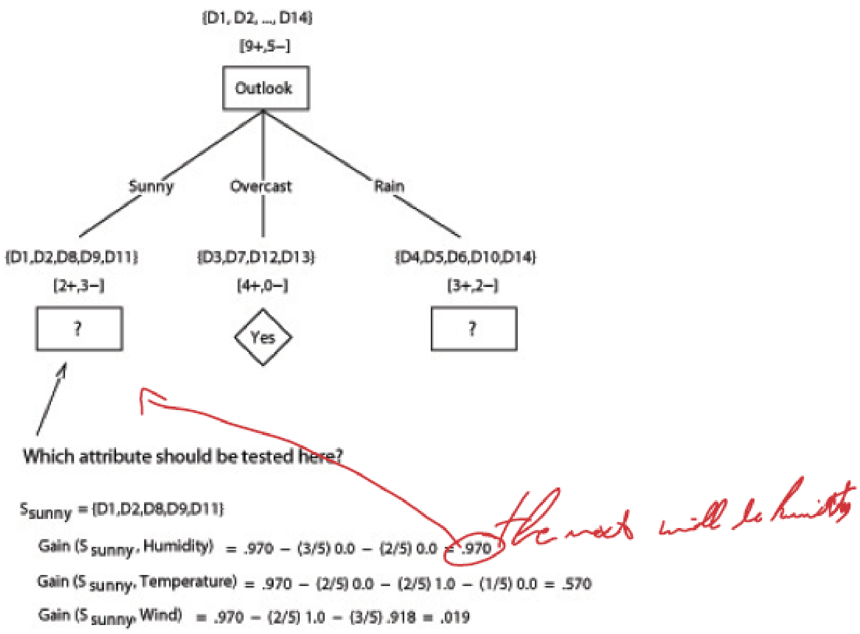
$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

Prop of examples classified as c.

Gain(S, A) = expected reduction in entropy of S caused by knowing the value of attribute A.

Gain(S, A) ≡ Entropy(S) – Sum[(|Sv|/|S|) Entropy(Sv)]

Sv = {s in S|A(s) = v}



If you end before the leaf you will choose with less accuracy

Overfitting in Decision Trees

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

Prune

Reduced-Error pruning

Split data into training and validation set

- Do until further pruning (potatura) is harmful (decreases accuracy):
  1 Evaluate impact on validation set of pruning each possible node
  2 Greedily remove the one that most improves validation set accuracy

Rule Post-Pruning

- Convert the learned tree into a set of rules
- Generalize each rule independently
- Sort rules for use

Specific Attributes

- Attributes with Many Values

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

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

- Attributes with Costs

  - Tan and Schlimmer (1990)

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

  - Nunez (1988) ($w \in [0, 1]$ determines importance of cost)

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

- Unknown Attribute Values

  Assign most common value