Decision Trees

Boris Velichkov

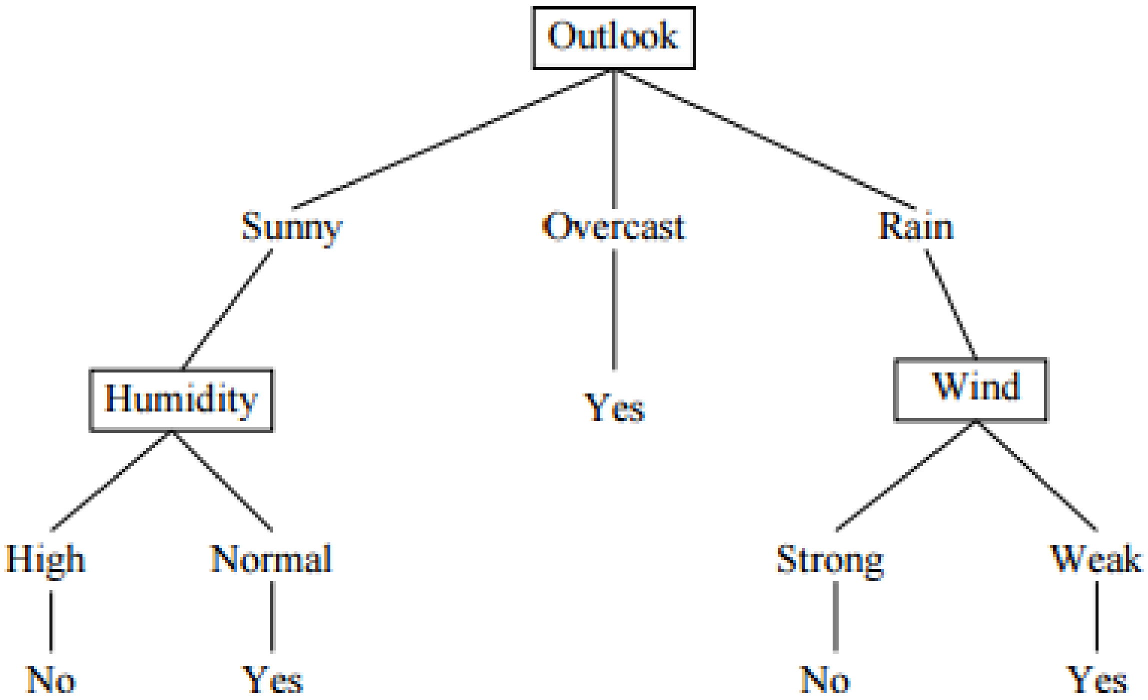
Decision Tree

- Machine Learning
  - Supervised Learning
    - Classification
    - Regression
  - Global Learning
  - Model-Based Learning
  - Eager Learning

Decision Tree

- Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented as a decision tree
- Decision tree representation:
  - Each internal node tests an attribute
  - Each branch corresponds to attribute value
  - Each leaf node assigns a classification
- Re-representation as if-then rules: disjunction of conjunctions of constraints on the attribute value instances

Decision Tree for Play Tennis



Logical Formulation: (Outlook = Sunny ∧ Humidity = Normal)
∨ (Outlook = Overcast)
∨ (Outlook = Rain ∧ Wind = Weak)

When to Consider Decision Trees

- Instances describable by attribute–value pairs
- Target function is discrete valued
- Disjunctive hypothesis may be required
- Possibly noisy training data

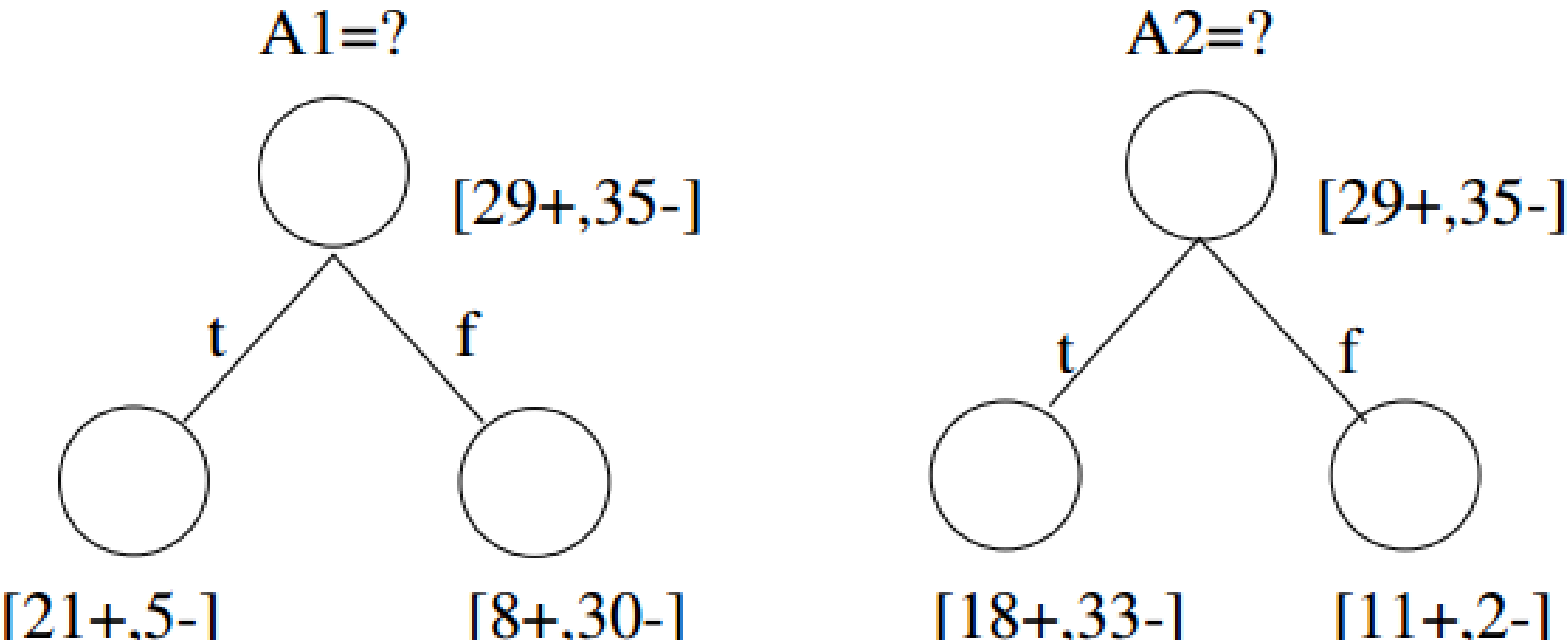
Examples:

- Equipment or medical diagnosis
- Credit risk analysis
- Modelling calendar scheduling preferences

Top-Down Induction of Decision Trees (ID3)

1. A is the “best” decision attribute for next node
2. Assign A as decision attribute for node
3. For each value of A, create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



Entropy

- S is a sample of training examples
- P+ is the proportion of positive examples in S
- P- is the proportion of negative examples in S
- Entropy measures the impurity of S

Entropy(S) ≡ H(S) ≡ −p+ log2 p+ − p- log2 p-

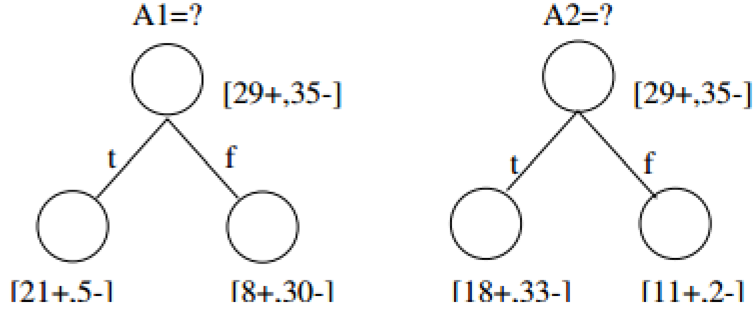
- H(S) = 0 if sample is pure (all + or all -), H(S) = 1 bit if p+ = p- = 0.5

Information Gain

- Gain(S, A) = expected reduction in entropy due to sorting on A

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

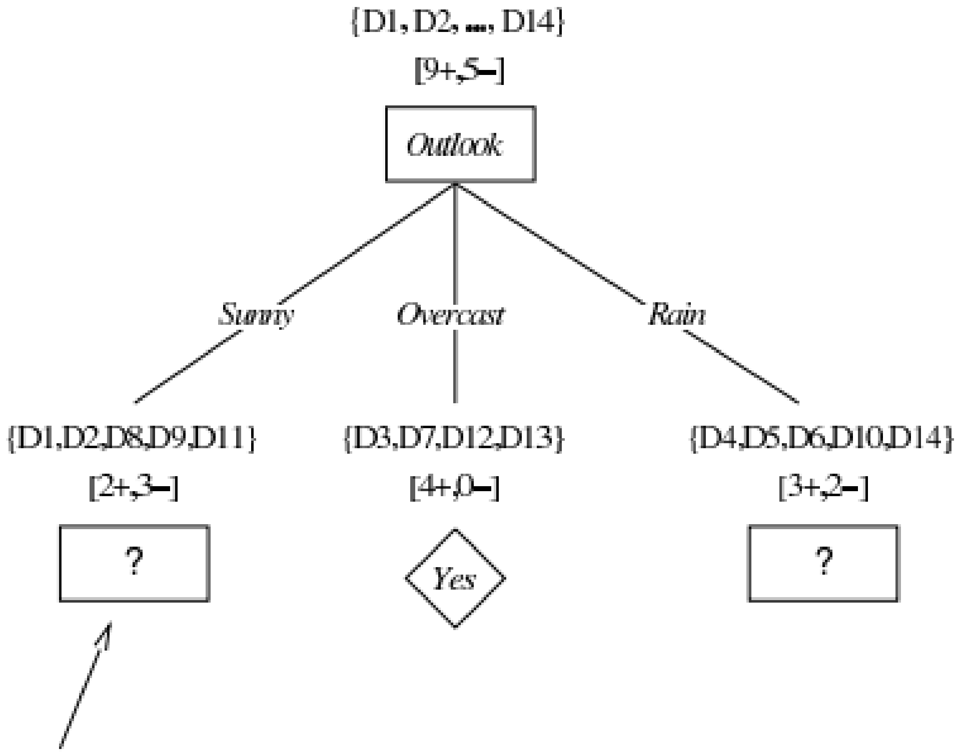
- Information gain is also called the mutual information between A and the labels of S



Training Examples

Day	Outlook	Temperature	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

Building the Decision Tree



Which attribute should be tested here?

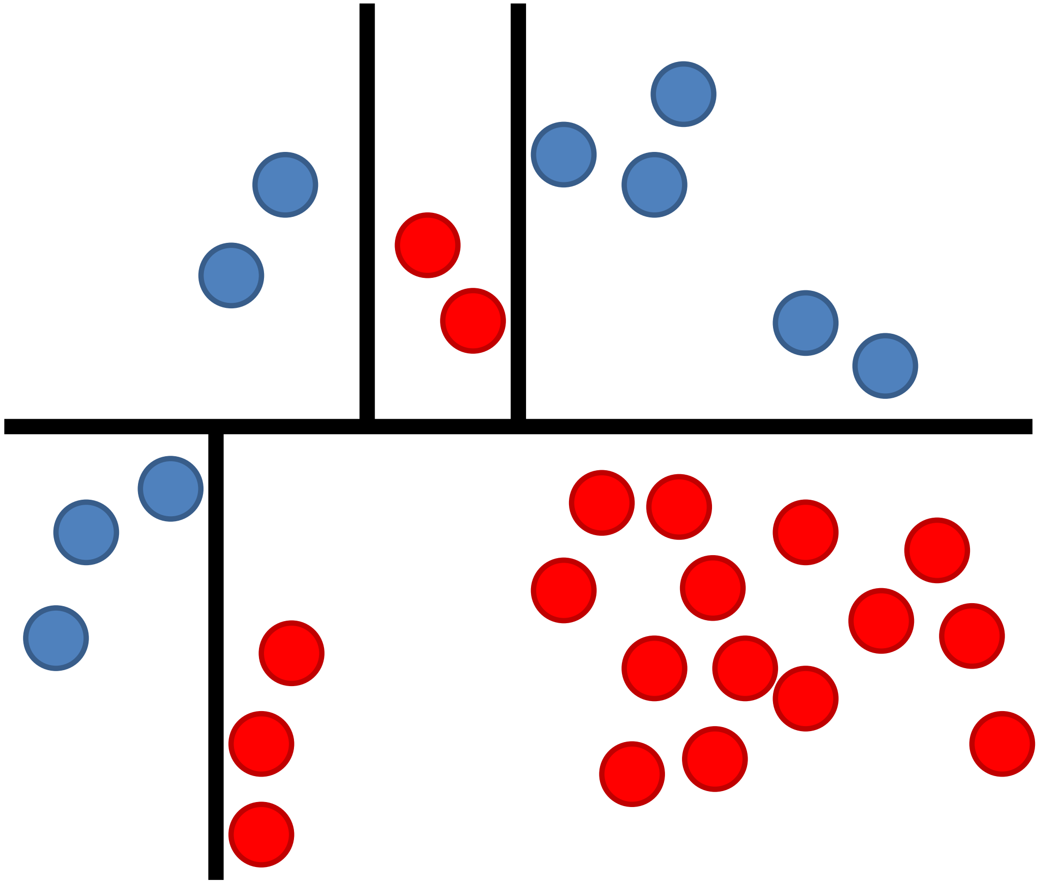
$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Decision Tree Visualization



How to Fix the Overfitting?

- Common technics
  - Getting more training data
  - Cross-validation sampling
  - Reducing number of features
  - Pruning
  - Regularization
    - Increase Regularization term

How to Fix the Overfitting?

- Mostly for Decision Trees
  - Pruning
    - Pre-Pruning
    - Post-Pruning
  - Ensemble Learning
    - Random Forest
    - XGBoost

Example

https://www.saedsayad.com/decision_tree.htm