CSC 461: Machine Learning

Fall 2024

Decision Trees

Prof. Marco Alvarez, Computer Science
University of Rhode Island

Introduction

- Decision trees
  - hierarchical models for classification and regression
    - tree-like structure of decisions
  - key components:
    - root node, internal nodes, leaf nodes
  - gained prominence in the 80s, still relevant in modern ML, particularly as foundation for ensemble methods

Preliminaries

Tennis dataset (example)

Classic dataset for illustrating decision trees

Goal: Predict whether to play tennis based on weather conditions

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

14 examples
4 discrete features
2 possible labels

How many possible combinations of inputs?

$3 \times 3 \times 2 \times 2$

How many possible combinations if your dataset has 500 binary features?

2^{500}

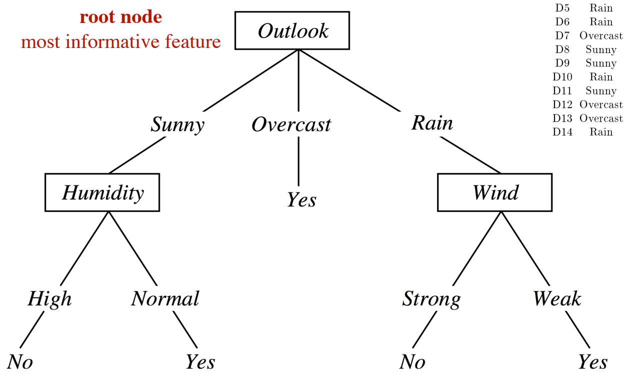
3273390607896141870013189696827599152216642046043064789483291368096133796404674554883270092325904157150886684127560071009217256545885393053328527589376

Machine Learning, Tom Mitchell, McGraw Hill, 1997

Tennis dataset (decision tree)

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

root node
most informative feature

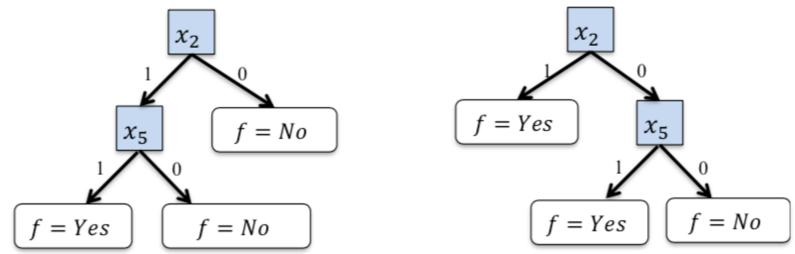


nodes
splits on feature values

leaf nodes
final decisions

Machine Learning, Tom Mitchell, McGraw Hill, 1997

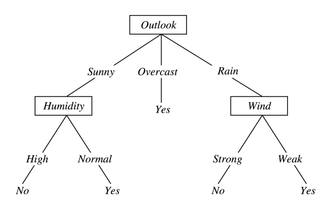
What logical functions these trees represent?



from: 10-315 Machine Learning, Maria-Florina (Nina) Balcan, CMU, Spring 2019

Interpretability

- Decision trees offer high interpretability
  - every path is a rule
    - if (Outlook = Sunny) ∧ (Humidity = Normal) then YES
  - rules are conjunctions
    - … ∧ … ∧ …
  - classes can be represented as disjunctions of conjunctions
    - … ∨ (… ∧ …) ∨ (… ∧ …) ∨ …



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

Expressiveness

- DTs can represent any boolean/discrete function
  - handle discrete input/discrete output scenarios
  - continuous variables can be discretized
- Search space complexity
  - how many distinct combinations of inputs?
    - $2^5 = 32$
  - how many boolean functions with 5 inputs and a binary output?
    - 2^{2^5}

Hypothesis space

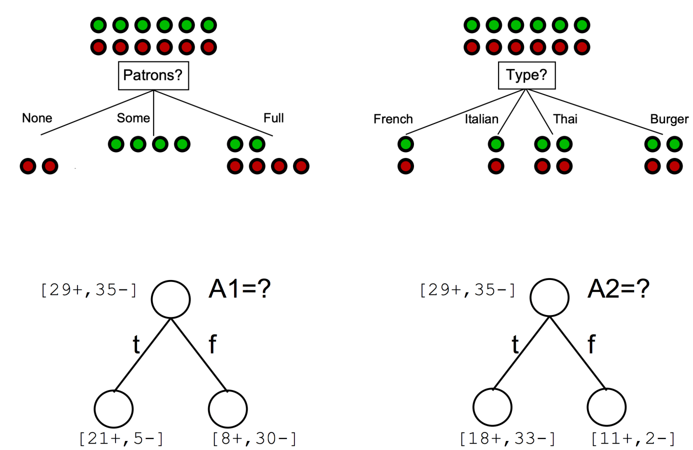
- More expressive hypothesis space …
  - allows learning complex target functions
  - increases number of consistent hypotheses
  - **risk of overfitting**: may not **generalize** well to unseen data
- DT learning goals
  - find a **small tree** consistent with **training data**
  - achieve good generalization
- **NP-hard** problem
  - no known polynomial-time algorithm for finding optimal tree
  - heuristic approaches used in practice

Consistent hypothesis

- Definition
  - h is consistent with $\mathscr{D}$ if $h(\mathbf{x}) = y, \forall(\mathbf{x}, y) \in \mathscr{D}$
- Expected behavior
  - if h is consistent with training data, then it would be accurate on new instances
- Note
  - a consistent tree always exists for any training data set
    - e.g., can just list all paths
    - may not generalize well
- Goal
  - find compact trees that generalize to unseen data

Entropy and information gain

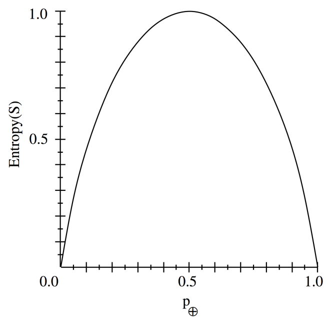
Select the "best" feature



http://aima.eecs.berkeley.edu/slides-pdf/chapter18.pdf

Entropy

- Assume a set $\mathcal{S}$ of positive/negative instances
  - **entropy** measures the impurity or uncertainty in $\mathcal{S}$



assuming k possible values each with different probabilities:

$$E(\mathcal{S}) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

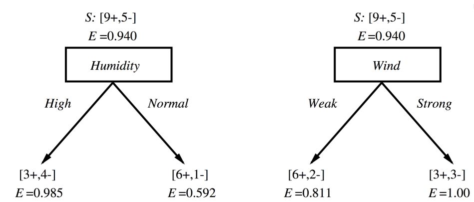
$$E(\mathcal{S}) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Information gain

- Expected reduction in **entropy** after splitting on an attribute (feature)

$$IG(\mathcal{S}, A) = E(\mathcal{S}) - \sum_{v \in A} \frac{|\mathcal{S}_v|}{|\mathcal{S}|} E(\mathcal{S}_v)$$

IG tends to increase for attributes with low entropy values



`Calculate the IG for both splits`

Learning a decision tree

Setup

- Data instances
  - every data instance $x \in \mathbb{R}^d$ is typically a **feature vector** of discrete values
    - continuous values can also be handled
  - $y \in \{1,2,\ldots,k\}$
- Hypothesis
  - each solution (hypothesis) is a **decision tree**

$$h : \mathcal{X} \mapsto \mathcal{Y}, h \in \mathcal{H}$$

Approach

- Build the tree using a **top-down** approach
  - select best feature to split on
  - create child nodes for each feature value
  - recursively apply steps above to child nodes
- Use a **greedy algorithm**
  - makes locally optimal choice at each step
  - cannot guarantee optimality (smallest consistent tree)
  - efficient, but may lead to suboptimal solutions