Decision Trees

CSC 461: Machine Learning

Fall 2020

Prof. Marco Alvarez
University of Rhode Island

Introduction

Learning Components

- Data instance
  - ✓ in general, $x \in \mathbb{R}^d$ is a feature vector of discrete values, but continuous values can also be handled
  - ✓ $y \in \{1,2,\ldots,k\}$
- Hypothesis
  - ✓ each hypothesis **g** is a decision tree

$$g : \mathscr{X} \mapsto \mathscr{Y}, g \in \mathscr{H}$$

Tennis dataset

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

Machine Learning, Tom Mitchell, McGraw Hill, 1997

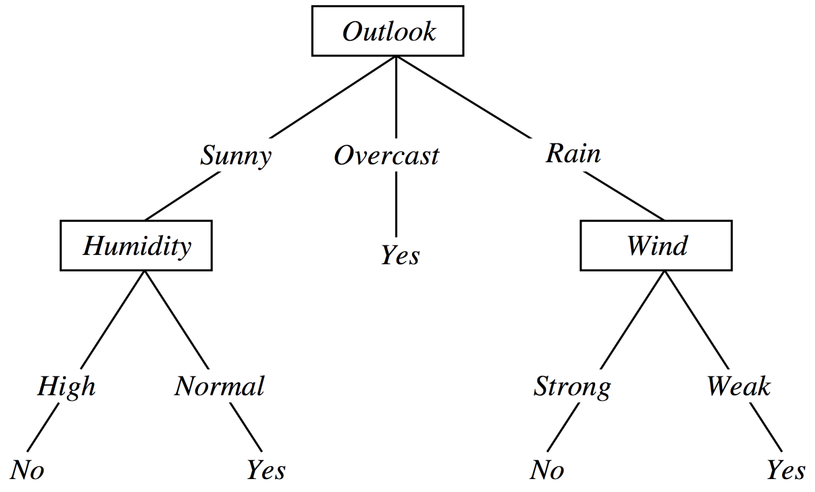
Warmup questions

How many rows are possible with these four features?

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

How many rows with 500 binary features?

Example (Decision Tree)



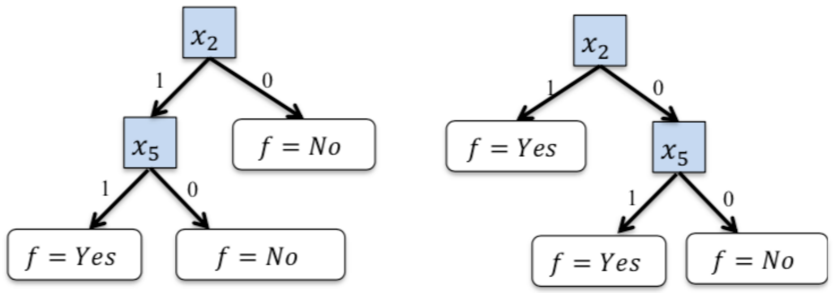
Machine Learning, Tom Mitchell, McGraw Hill, 1997

Representation

- Nodes test features/attributes
- Branches represent possible values for a feature
- Leaves represent outputs (classes)
- Assuming boolean variables, draw the trees:

$A \wedge B$
$A \vee B$
$(A \wedge B) \vee (C \wedge \neg D \wedge E)$

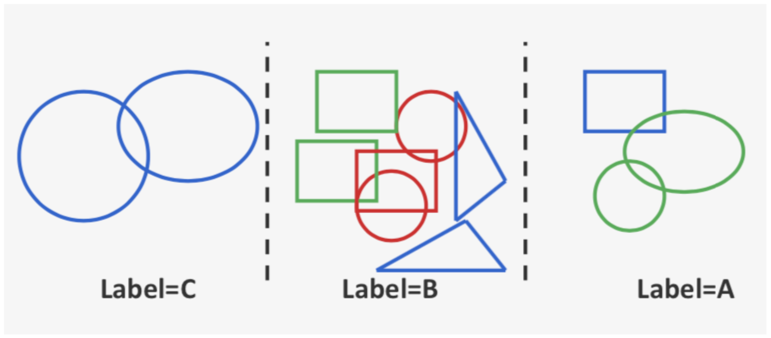
What functions are represented?



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

Build your own tree

- Assume instances with two features
  - ✓ color and shape



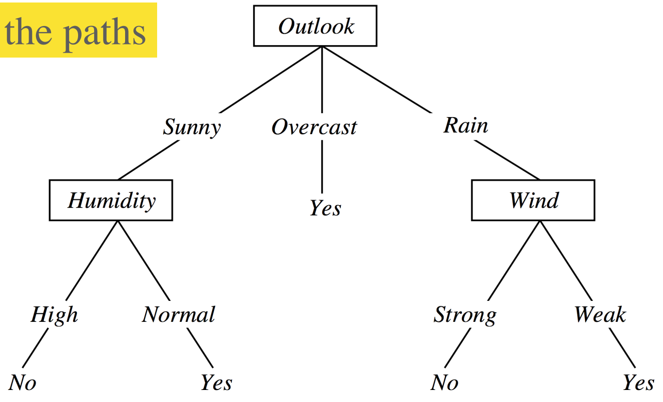
from: CS 5350 Machine Learning, Vivek Srikumar, University of Utah, Fall 2019

Test your tree

- What are the labels for a red triangle and a green triangle?

Extracting rules from the tree

Look at the paths



Machine Learning, Tom Mitchell, McGraw Hill, 1997

Disjunction of conjunctions

$$\ldots \vee (\ldots \wedge \ldots) \vee (\ldots \wedge \ldots) \vee \ldots$$

If …

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

then it belongs to class YES

Consistent hypotheses

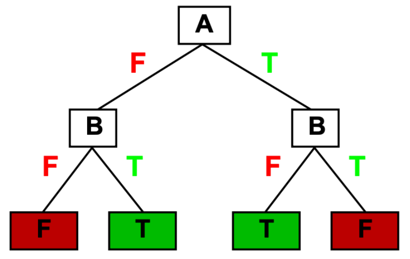
- A hypothesis **h** is consistent with a set of training examples **S** if and only if **h(x) = y** for all pairs **(x, y)** in **S**
  - ✓ our hope: if **h** is consistent with training data, then it would be accurate on new instances

- There is a tree consistent with any training set (just list all paths) — **it may not generalize well**
- Preferably we want more **compact** trees that can **generalize** better

Expressiveness

- A decision tree can represent any boolean/ discrete function (discrete input/discrete output)

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



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

Hypothesis space

How many distinct decision trees can be created with n=5 boolean features?

x					y
0	0	0	0	0	T/F
0	0	0	0	1	T/F
0	0	0	1	0	T/F
0	0	0	1	1	T/F
0	0	1	0	0	T/F
…					T/F
1	1	1	1	1	T/F

2^5 = 32 entries

how many boolean functions with 5 features are there, given that entries can be T/F?

$$2^{2^5}$$

Try n == 10

Hypothesis space

- More expressive hypothesis space …
  - ✓ allows learning complex target functions
  - ✓ increases number of consistent hypotheses
  - ✓ may not **generalize**, due to **overfitting**
- DT learning
  - ✓ find a small tree consistent with the training data
  - ✓ **NP-complete** (polynomial algorithm may not exist)

Learning a Decision Tree

Goal

- (small) Hypothesis **g** that best approximates **f**

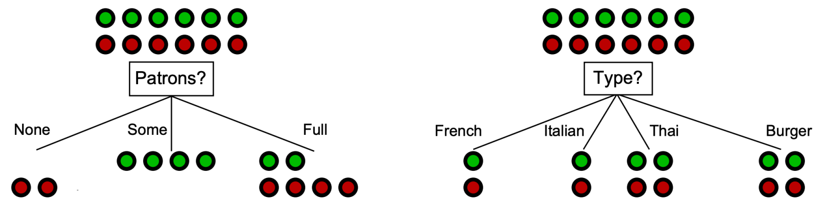
$$\forall (x_i, y_i) \sim P \text{ and } g \in \mathscr{H}$$

$$g(x) \approx f(x)$$

Induction of a Decision Tree

- Build the tree using a **top-down** approach
- **Greedy** algorithm
  - ✓ makes the **optimal** (feature) choice at each step
  - ✓ the greedy nature of the algorithm cannot guarantee **optimality (smallest tree consistent with the data)**
- **NP-complete** problem
  - ✓ “Although a solution to an NP-complete problem can be verified "quickly", there is no known way to find a solution quickly” [wikipedia]

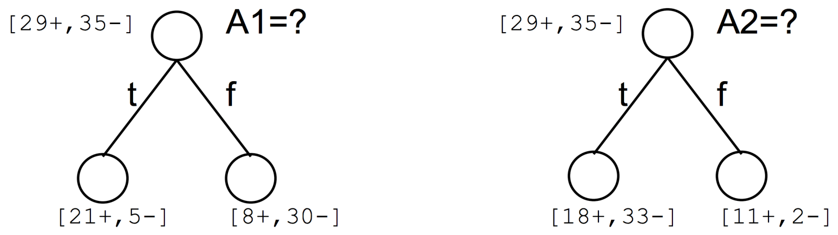
Which feature is better? Why?



- Which feature is more **informative**?
- Which provides the minimum **0/1 loss** if we use the majority vote for classifying new instances?

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

Which feature is better?



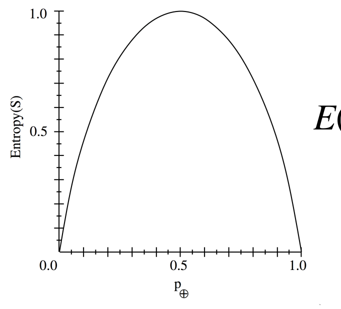
How to choose the splitting feature?

- Information Gain
  - ✓ used in **ID3**
- Gain Ratio
  - ✓ used in C4.5
- Gini Measure
  - ✓ used in CART

ID3 was invented by Ross Quinlan

Entropy

- Assume a set S of positive/negative instances
  - ✓ entropy measures the **impurity** of S



w.r.t. a binary variable

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

Entropy

- Assuming **k** possible values, each with different probabilities, then:

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

What is the entropy if all instances belong to the same category?

Information Gain

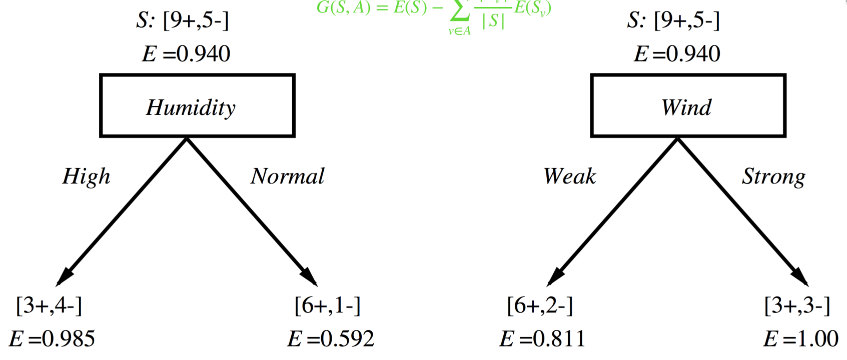
- Expected reduction in **Entropy** after splitting

$$G(S,A) = E(S) - \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$$

- **Information gain** increases for low entropy values

Calculate the Information Gain

$G(S,A) = E(S) - \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$



Induction of a Decision Tree

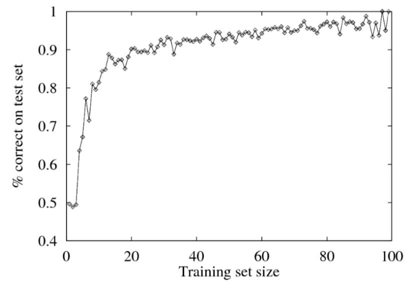
```
Algorithm GrowTree($D, F$)
Input  : data $D$; set of features $F$.
Output : feature tree $T$ with labelled leaves.
if Homogeneous($D$) then return Label($D$) ;      // Homogeneous, Label: see text
$S \leftarrow$ BestSplit($D, F$) ;                 // e.g., BestSplit-Class (Algorithm 5.2)
split $D$ into subsets $D_i$ according to the literals in $S$;
for each $i$ do
    if $D_i \neq \emptyset$ then $T_i \leftarrow$ GrowTree($D_i, F$) else $T_i$ is a leaf labelled with
    Label($D$);
end
return a tree whose root is labelled with $S$ and whose children are $T_i$
```

Induction of a Decision Tree

```
Algorithm BestSplit-Class($D, F$) – find the best split for a decision tree.
Input  : data $D$; set of features $F$.
Output : feature $f$ to split on.
$I_{\min} \leftarrow 1$;
for each $f \in F$ do
    split $D$ into subsets $D_1, \ldots, D_l$ according to the values $v_j$ of $f$;
    if Imp($\{D_1, \ldots, D_l\}$) $< I_{\min}$ then
        $I_{\min} \leftarrow$ Imp($\{D_1, \ldots, D_l\}$);
        $f_{\text{best}} \leftarrow f$;
    end
end
return $f_{\text{best}}$
```

Resulting tree

- Tree is expected to be small and consistent with training examples $g \in \mathcal{H}$
- Tree does not necessarily agree with the correct function (bigger training sets help) $f \in \mathcal{H}$



Final Remarks

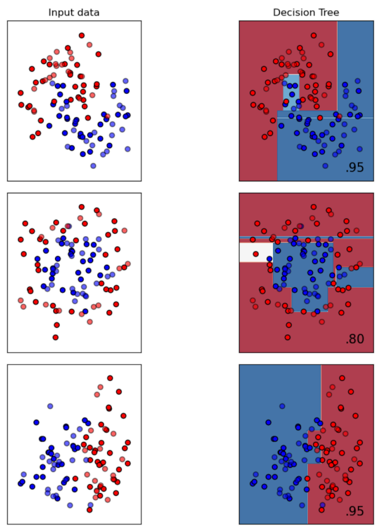
Continuous features

- Transform continuous into discrete features
  - ✓ use thresholds defined by domain experts or automatically calculated from training data
- For example:
  - ✓ sort values (training set)
  - ✓ find split points where class changes

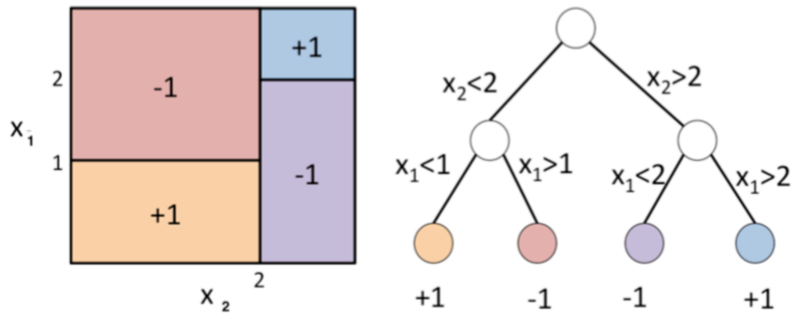
Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

54 85

Nonlinear Decision Boundary



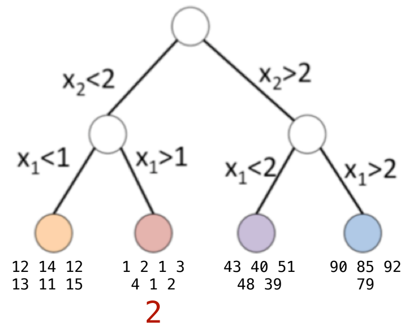
Nonlinear Decision Boundary



from: CS260 Machine Learning Algorithms, Cho-Jui Hsieh, UCLA, 2019

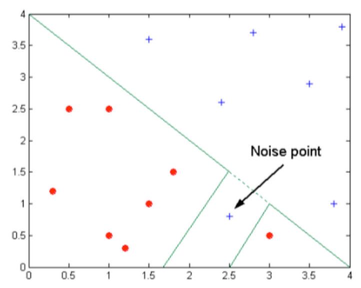
Continuous outputs

- Regression trees
  - ✓ can assign a continuous value to a leaf
  - ✓ e.g. the **average** of all **y** values that fall into the leaf

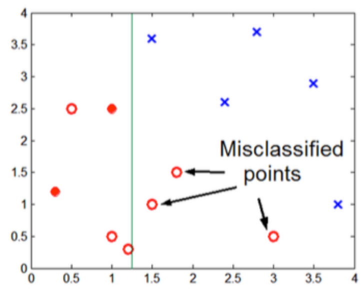


Model overfitting

A hypothesis **h1** is said to **overfit** the training data if there exists some alternative hypothesis **h2** such that **h1** has smaller error than **h2** over the training examples, but **h2** has a smaller error than **h1** over the entire distribution of instances



due to noise



due to lack of representative instances

from: Data Mining I, Eirini Ntoutsi, Leibniz University, Summer 2019

Preventing overfitting (DTs)

- Remove irrelevant features
- Add more data
- Stop growing branches during training
  - ✓ hard thresholds or statistical measures
- Prune the tree post-training

Additional thoughts on DTs

- **Nonlinear** classifiers, which can also provide **interpretability**
- Training may be **slow** but inference is **fast**
  - ✓ what is the big-O of inference?
- Although trees can be small, certain functions will require an exponentially large decision tree
  - ✓ e.g. **majority** (1 if n inputs are positive), **parity** (1 if even number of inputs is positive)