杭州電子科技大學
HANGZHOU DIANZI UNIVERSITY

媒体智能实验室
Media Intelligence Lab

03 Decision Trees

Machine Learning in Action

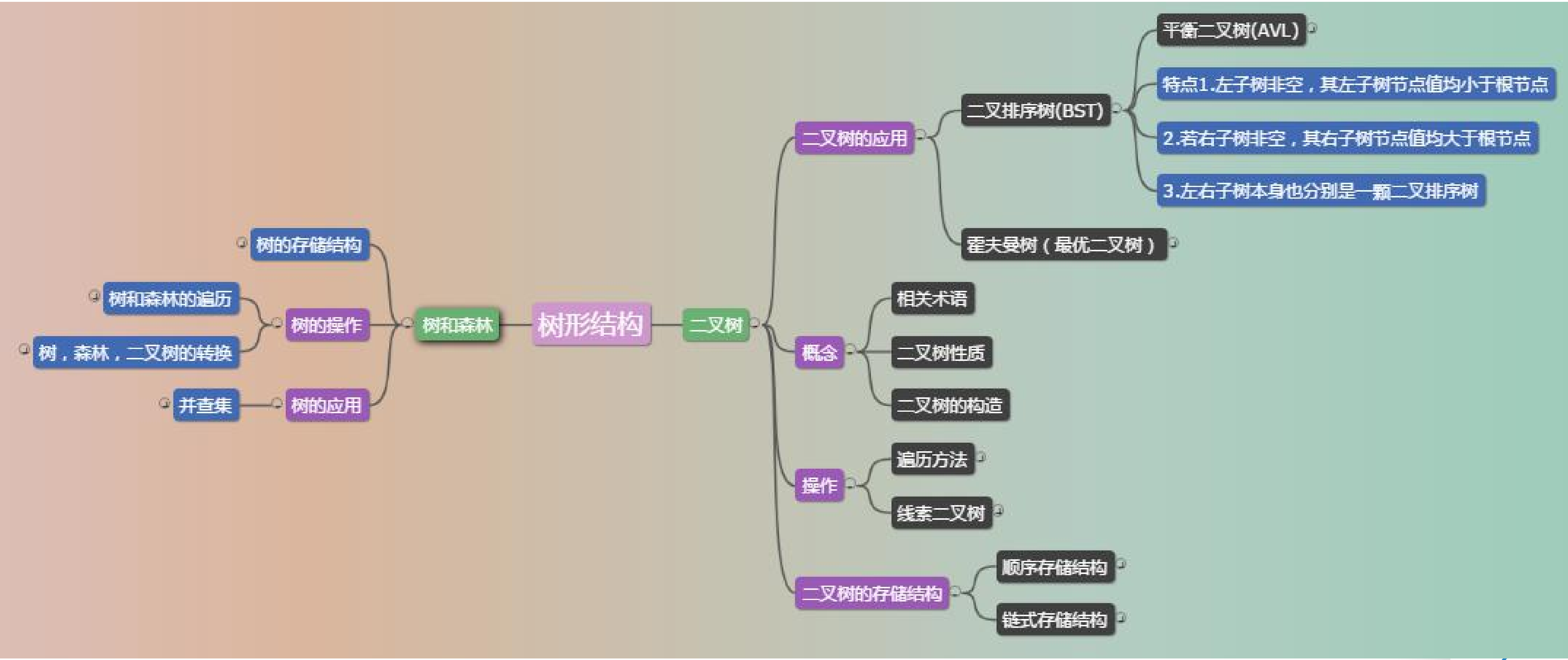
- MIL, College of Computer Science and Technology, Hangzhou Dianzi University
- Suguo Zhu
- zsg2016@hdu.edu.cn

K-Nearest Neighbors

- The k-NN algorithm did a great job of classifying, but it didn't lead to any major insights about the data.
- One of the best things about decision trees is that humans can easily understand the data.

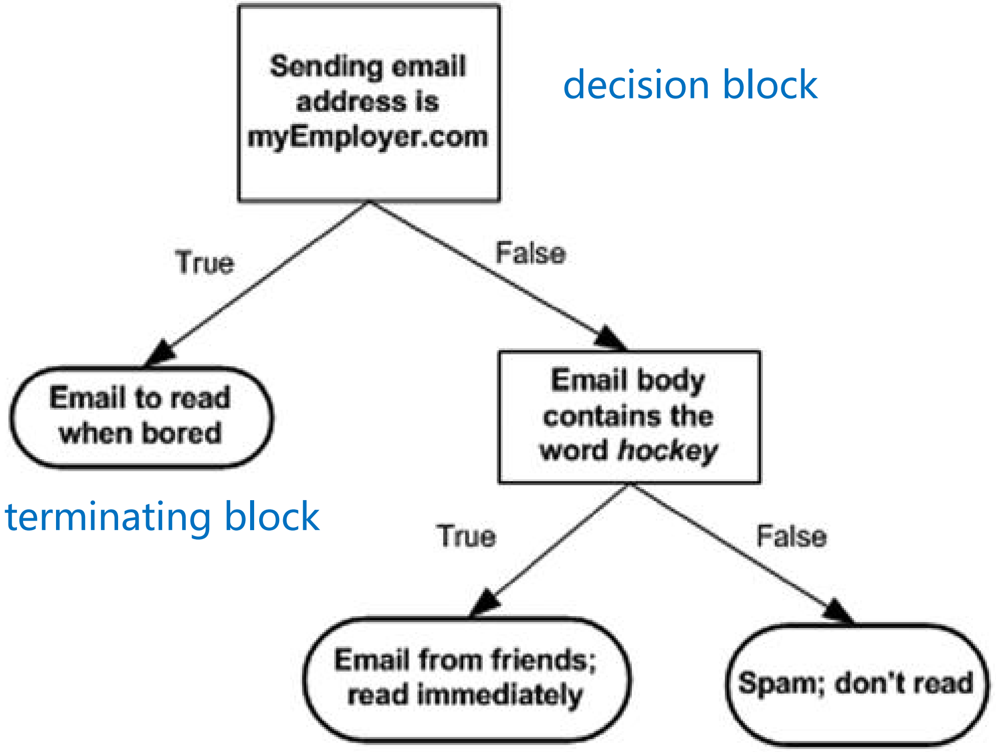
- What is Decision Tree ?

Data Structure——Trees



Twenty Questions

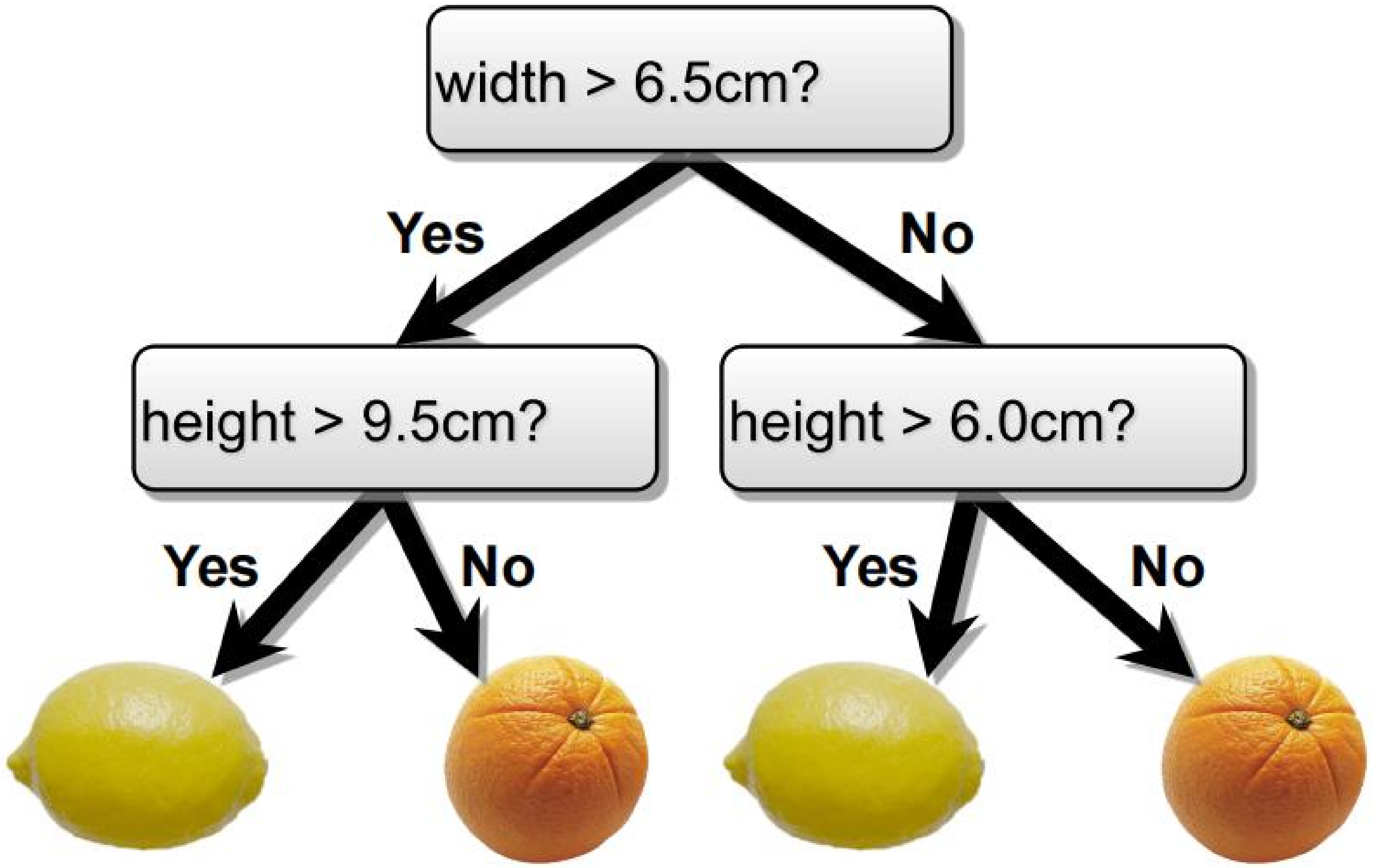
Publicity photo from the game show Twenty Questions. Photo from DuMont advertising the show, with 14-year-old Dick Harrison, Herb Polesie, Fred Van Deventer, Florence Rinard, and actor Aldo Ray as guest panelist (February 1, 1954)



A decision tree in flowchart form

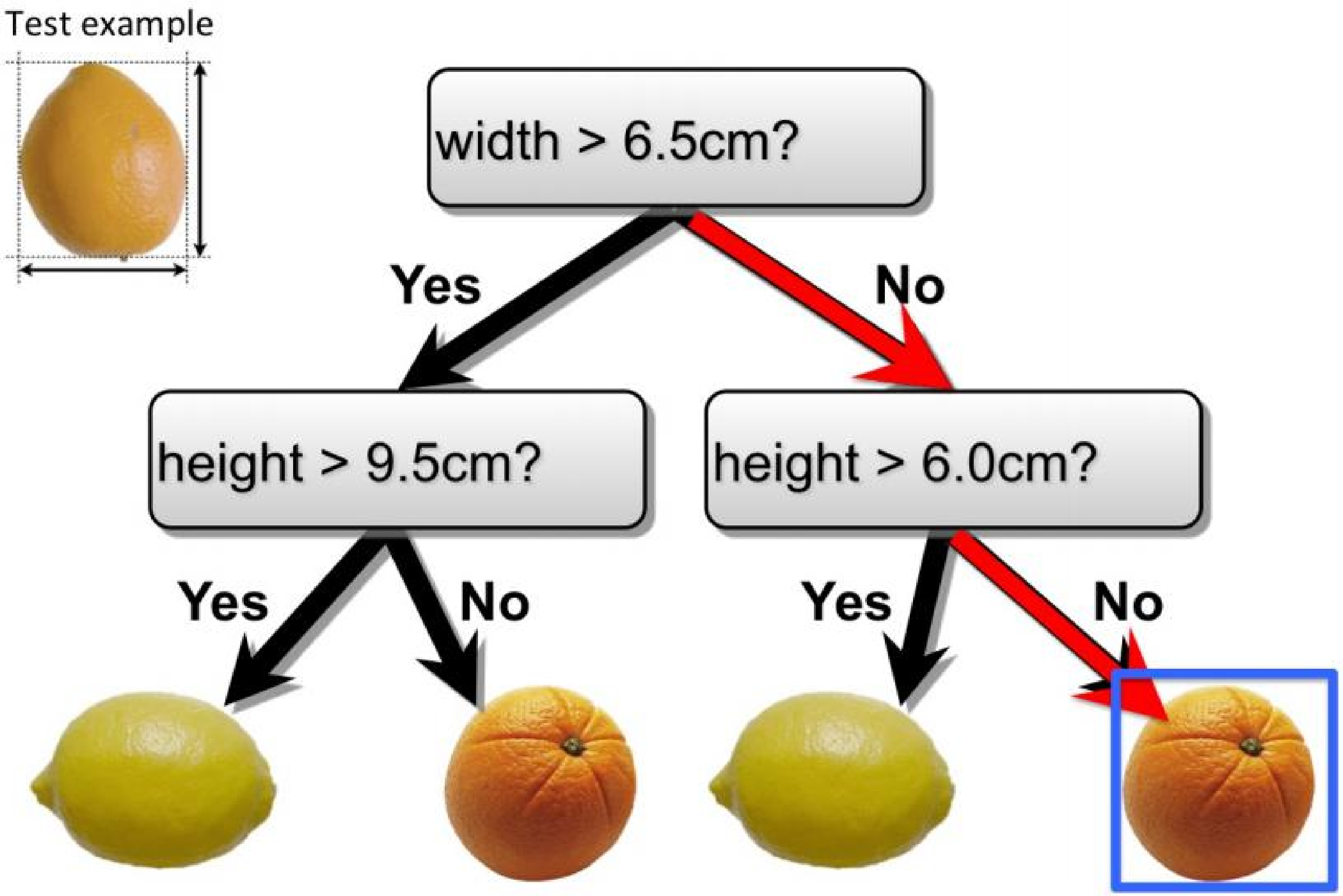
Decision Trees

- Make predictions by splitting on features according to a tree structure.



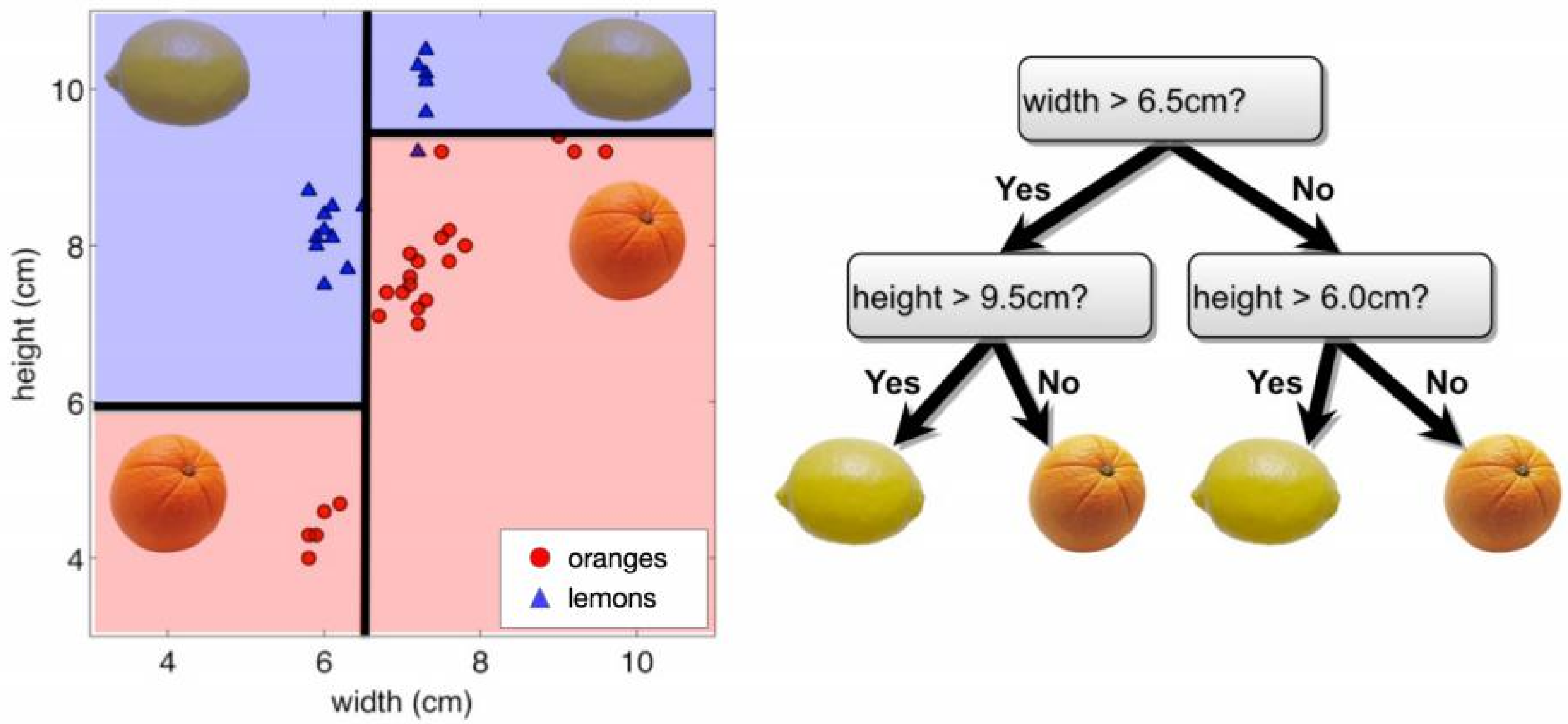
Decision Trees

- Make predictions by splitting on features according to a tree structure.



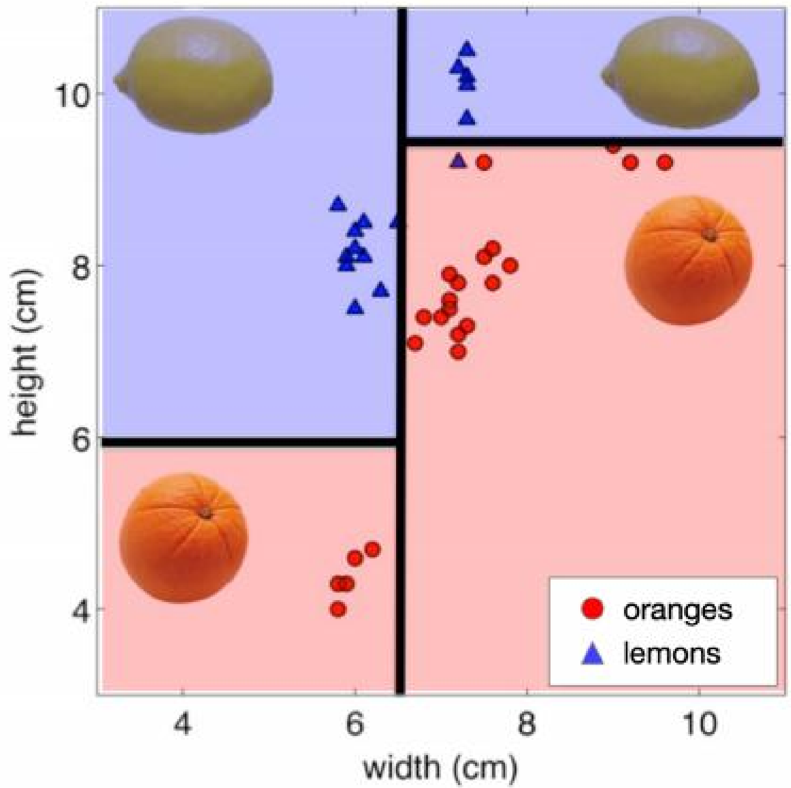
Decision Trees—Continuous Features

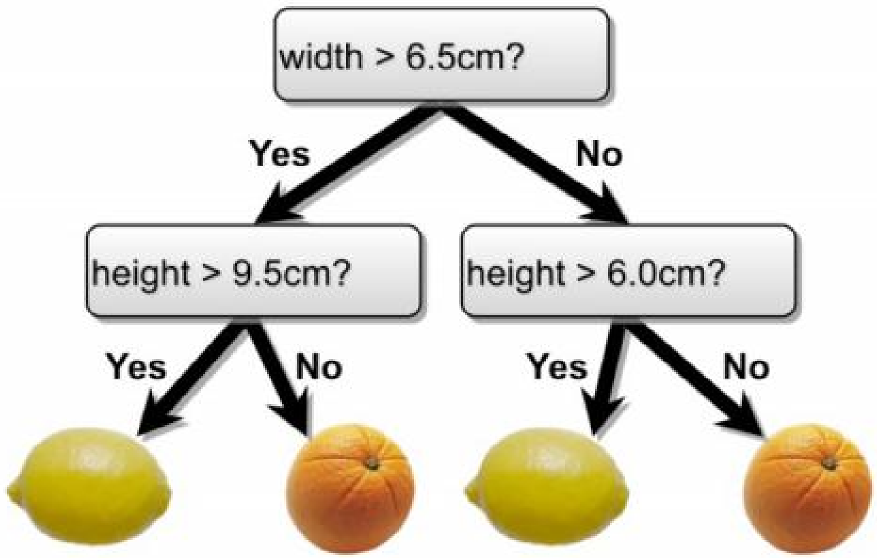
- Split *continuous features* by checking whether that feature is greater than or less than some threshold.
- Decision boundary is made up of axis-aligned planes.

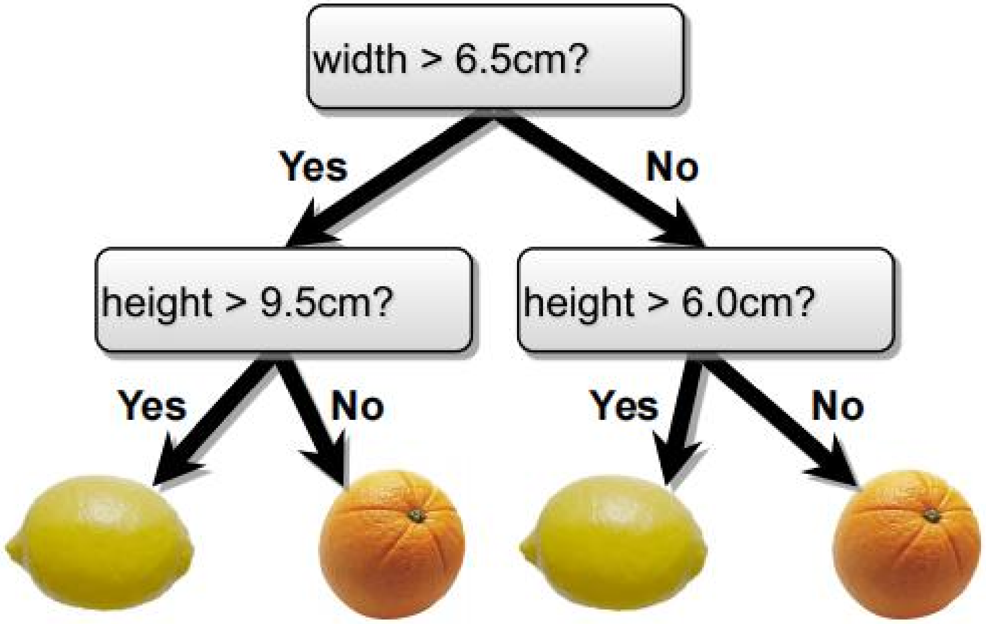


Decision Trees—Continuous Features

- Split *continuous features* by checking whether that feature is greater than or less than some threshold.
- Decision boundary is made up of axis-aligned planes.

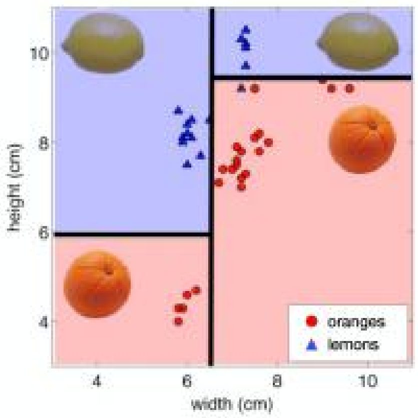






- Internal nodes test a feature
- Branching is determined by the feature value
- Leaf nodes are outputs (predictions)

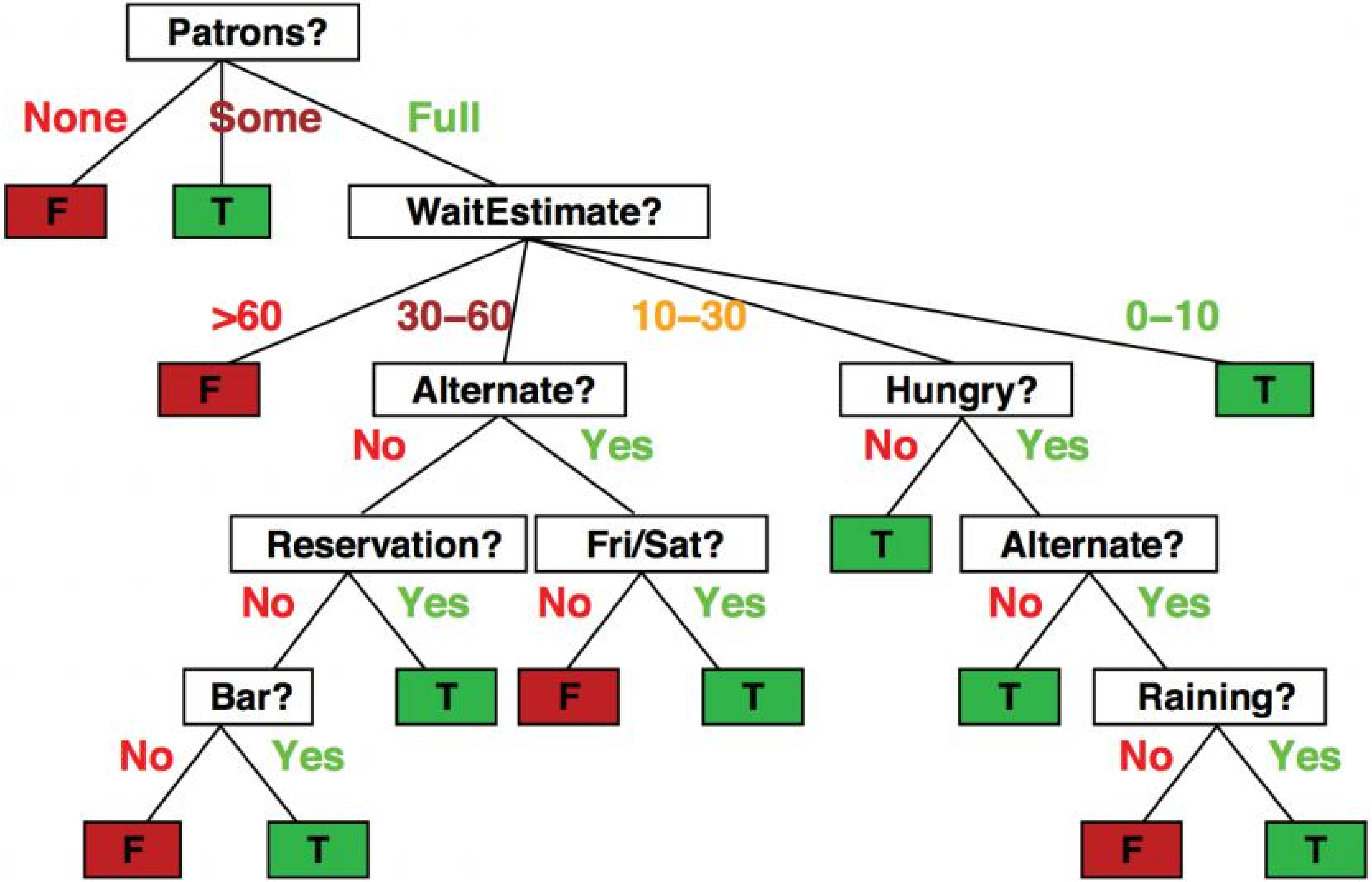
Decision Trees—Classification and Regression



- Each path from root to a leaf defines a region R_m of input space
- Let $\{(x^{(m_1)}, t^{(m_1)}), \ldots, (x^{(m_k)}, t^{(m_k)})\}$ be the training examples that fall into R_m
- Classification tree (we will focus on this):
  - discrete output
  - leaf value y^m typically set to the most common value in $\{t^{(m_1)}, \ldots, t^{(m_k)}\}$
- Regression tree:
  - continuous output
  - leaf value y^m typically set to the mean value in $\{t^{(m_1)}, \ldots, t^{(m_k)}\}$

Decision Trees—Discrete Features

- Will I eat at this restaurant?



Decision Trees—Discrete Features

- Split *discrete features* into a partition of possible values.

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} =$ *No*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} =$ *No*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} =$ *Yes*

Features:

1.	Alternate: whether there is a suitable alternative restaurant nearby.
2.	Bar: whether the restaurant has a comfortable bar area to wait in.
3.	Fri/Sat: true on Fridays and Saturdays.
4.	Hungry: whether we are hungry.
5.	Patrons: how many people are in the restaurant (values are None, Some, and Full).
6.	Price: the restaurant's price range ($, $$, $$$).
7.	Raining: whether it is raining outside.
8.	Reservation: whether we made a reservation.
9.	Type: the kind of restaurant (French, Italian, Thai or Burger).
10.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Decision Tree Learning

Example: learn concept **PlayTennis** (i.e., decide whether our friend will play tennis or not in a given day)

simple training dataset

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Learning Decision Trees

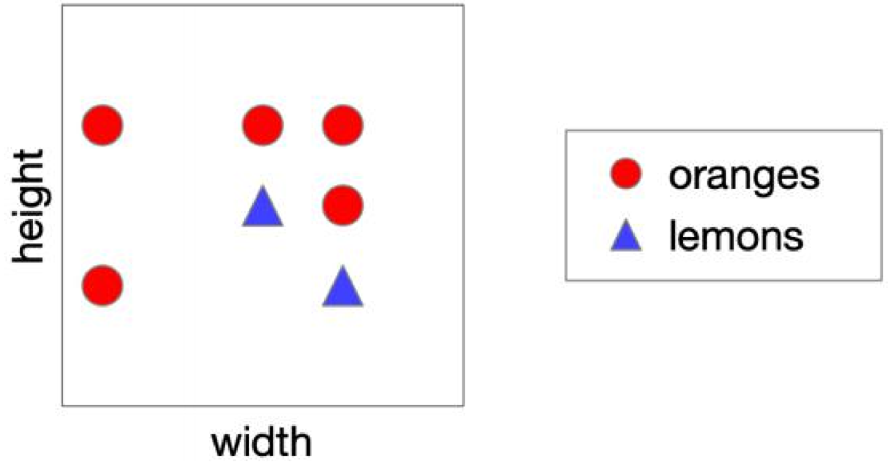
- For any training set we can construct a decision tree that has exactly the one leaf for every training point, but it probably won't generalize.
  - Decision trees are universal function approximators.
- But, finding the smallest decision tree that correctly classifies a training set is NP complete.
  - If you are interested, check: Hyafil & Rivest'76.
- So, how do we construct a useful decision tree?

Learning Decision Trees

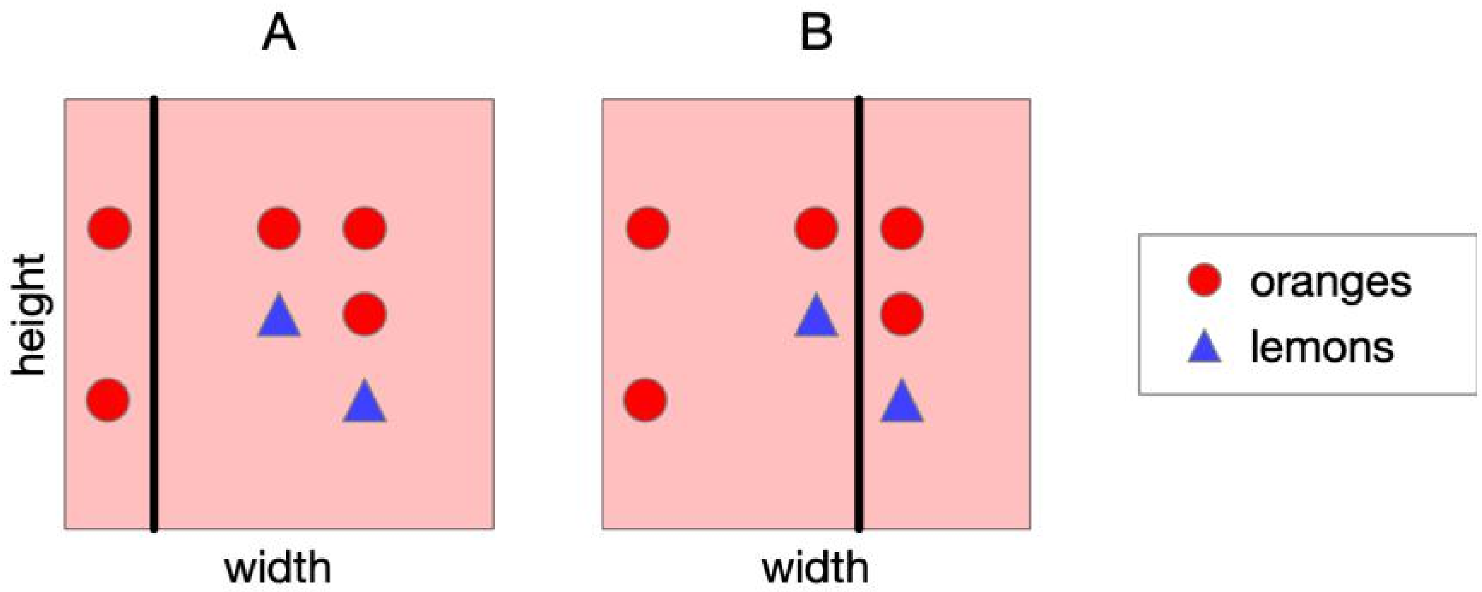
- Resort to a greedy heuristic:
  - Start with the whole training set and an empty decision tree.
  - Pick a feature and candidate split that would most reduce the loss.
  - Split on that feature and recurse on subpartitions.
- Which loss should we use?
  - Let's see if misclassification rate is a good loss.

Choosing a Good Split

- Consider the following data. Let's split on width.



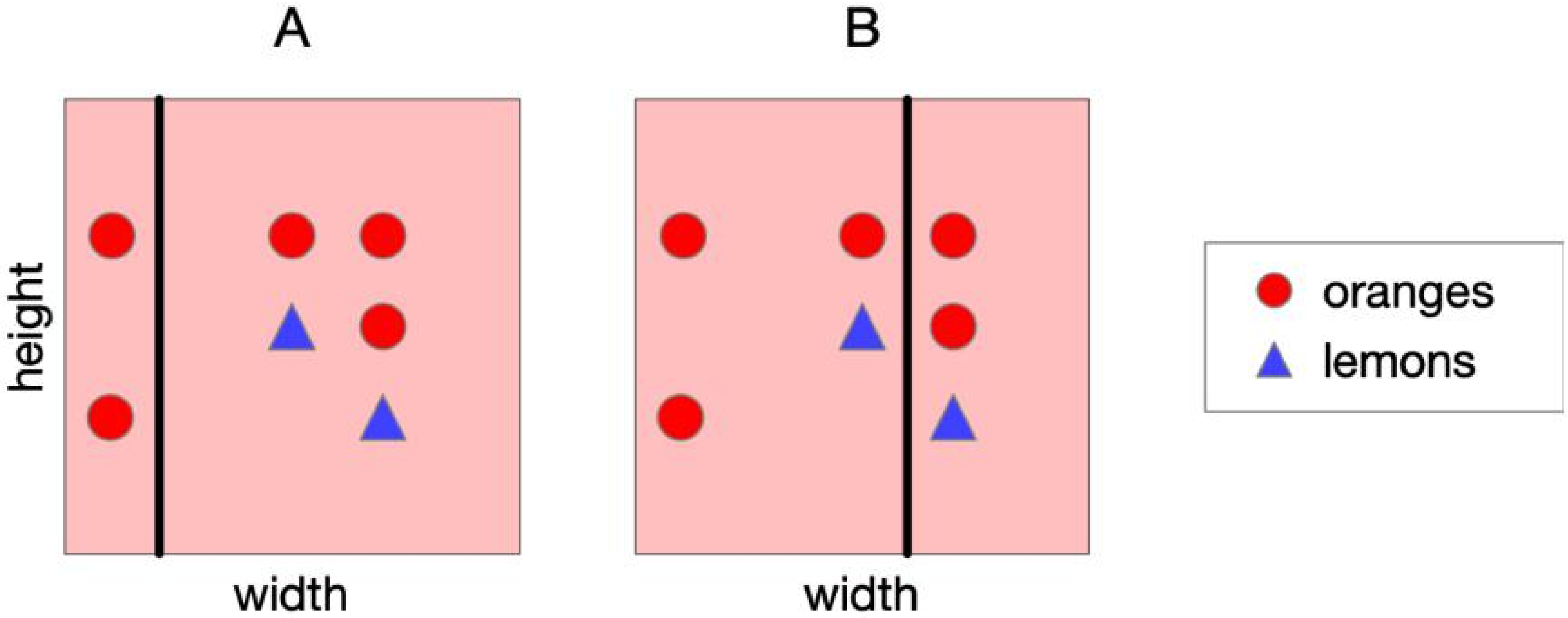
- Recall: classify by majority.



A and B have the same misclassification rate, so which is the best split? Vote!

Choosing a Good Split

- A feels like a better split, because the left-hand region is very certain about whether the fruit is an orange.



- Can we quantify this?

Choosing a Good Split

- How can we quantify uncertainty in prediction for a given leaf node?
  - If all examples in leaf have same class: good, low uncertainty
  - If each class has same amount of examples in leaf: bad, high uncertainty
- **Idea:** Use counts at leaves to define probability distributions; use a probabilistic notion of uncertainty to decide splits.
- A brief detour through information theory...

Quantifying Uncertainty

- The entropy of a discrete random variable is a number that quantifies the uncertainty inherent in its possible outcomes.
- The mathematical definition of entropy that we give in a few slides may seem arbitrary, but it can be motivated axiomatically.
  - If you're interested, check: *Information Theory* by Robert Ash.

Entropy

- More generally, the entropy of a discrete random variable Y is given by

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$$

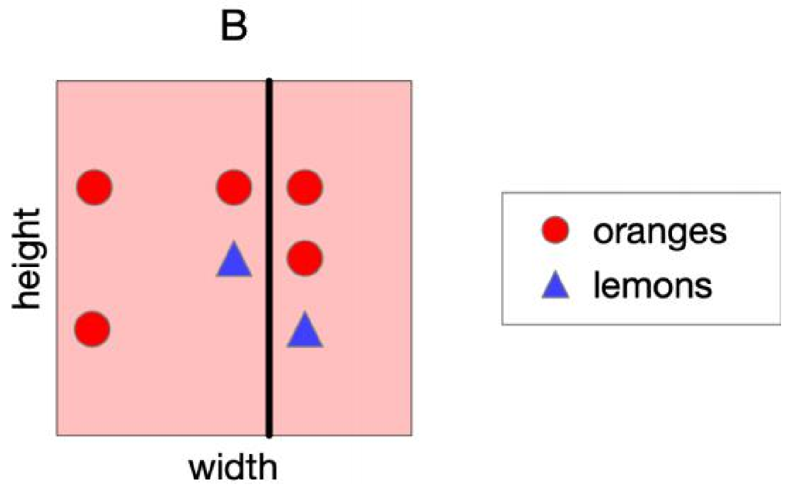
- **"High Entropy"**:
  - Variable has a uniform like distribution over many outcomes
  - Flat histogram
  - Values sampled from it are less predictable
- **"Low Entropy"**
  - Distribution is concentrated on only a few outcomes
  - Histogram is concentrated in a few areas
  - Values sampled from it are more predictable

Information gain

- Information gain measures the informativeness of a variable, which is exactly what we desire in a decision tree split!
- The information gain of a split: how much information (over the training set) about the class label Y is gained by knowing which side of a split you're on.

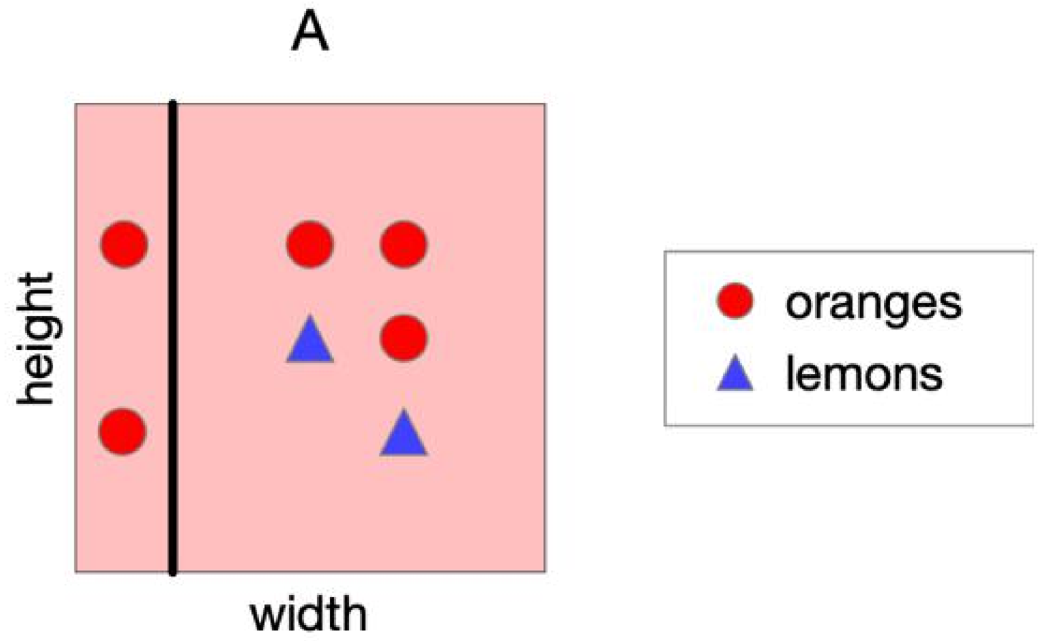
Revisiting Our Original Example

- What is the information gain of split B? Not terribly informative...



- Root entropy of class outcome: $H(Y) = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) \approx 0.86$
- Leaf conditional entropy of class outcome: $H(Y|left) \approx 0.81$, $H(Y|right) \approx 0.92$
- $IG(split) \approx 0.86 - (\frac{4}{7} \cdot 0.81 + \frac{3}{7} \cdot 0.92) \approx 0.006$

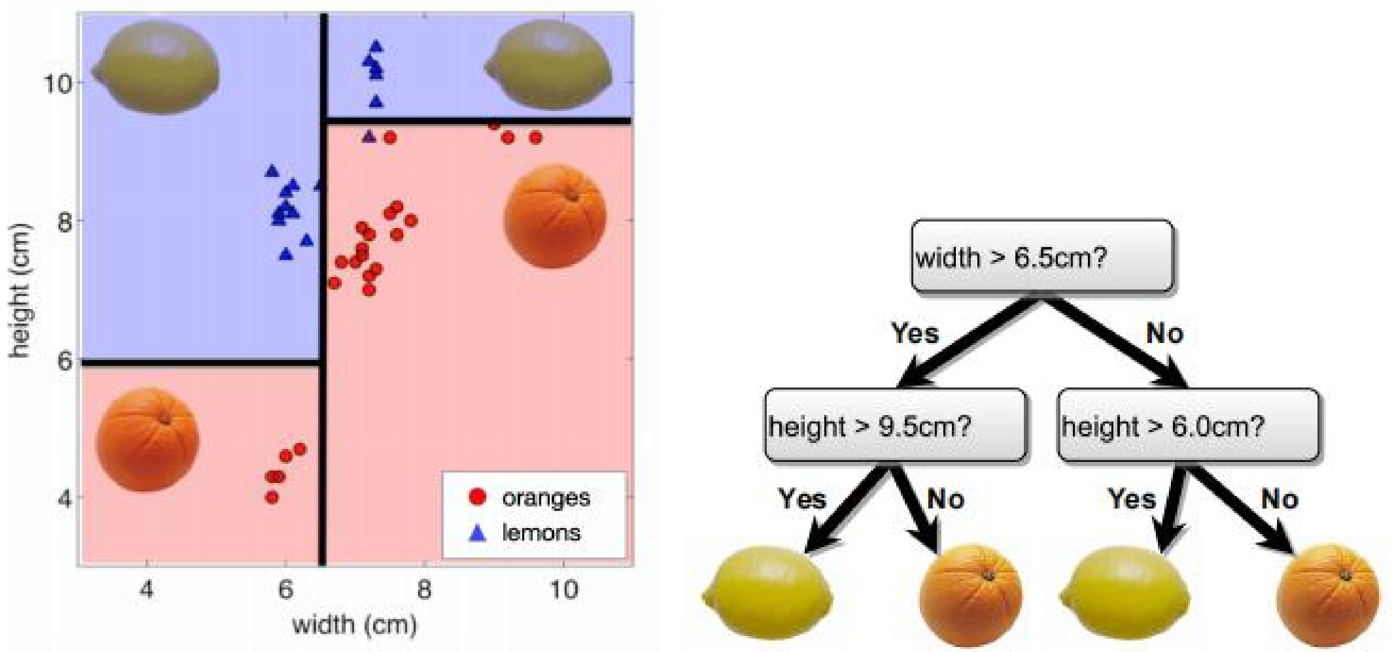
- What is the information gain of split A? Very informative!



- Root entropy of class outcome: $H(Y) = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) \approx 0.86$
- Leaf conditional entropy of class outcome: $H(Y|left) = 0$, $H(Y|right) \approx 0.97$
- $IG(split) \approx 0.86 - (\frac{2}{7}\cdot 0 + \frac{5}{7}\cdot 0.97) \approx 0.17!!$

Constructing Decision Trees

- At each level, one must choose:
  1. Which feature to split.
  2. Possibly where to split it.
- Choose them based on how much information we would gain from the decision! (choose feature that gives the highest gain)



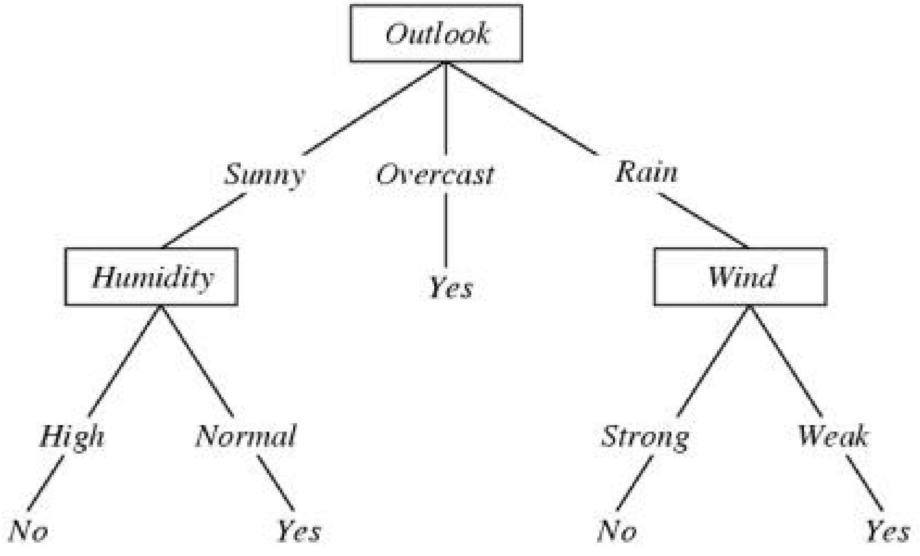
Decision Tree Construction Algorithm

- Simple, greedy, recursive approach, builds up tree node-by-node
    1. pick a feature to split at a non-terminal node
    2. split examples into groups based on feature value
    3. for each group:
        - if no examples – return majority from parent
        - else if all examples in same class – return class
        - else loop to step 1
- Terminates when all leaves contain only examples in the same class or are empty.

Decision Tree Learning

- Each internal node: test one (discrete-valued) attribute X_i
- Each branch from a node: corresponds to one possible values for X_i
- Each leaf node: predict Y (or $P(Y=1|x \in \text{leaf})$)

Example: A Decision tree for
f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

E.g., x=(
Outlook=sunny, Temperature-Hot,
Humidity=Normal,Wind=High), f(x)=Yes

Problem Setting

Input: Training labeled examples $\{(x^{(i)},y^{(i)})\}$ of unknown target function f

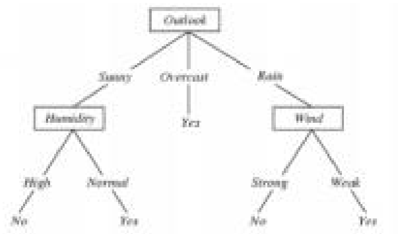
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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

- Examples described by their values on some set of features or attributes
  - E.g. 4 attributes: *Humidity, Wind, Outlook, Temp*
    - e.g., <*Humidity=High, Wind=weak, Outlook=rain, Temp=Mild*>
  - Set of possible instances X (a.k.a instance space)

- Unknown target function $f : X \rightarrow Y$
  - e.g., $Y=\{0,1\}$ label space
  - e.g., 1 if we play tennis on this day, else 0

Output: Hypothesis $h \in H$ that (best) approximates target function f

- Set of function hypotheses $H=\{\ h \mid h : X \rightarrow Y\ \}$
  - each hypothesis h is a decision tree



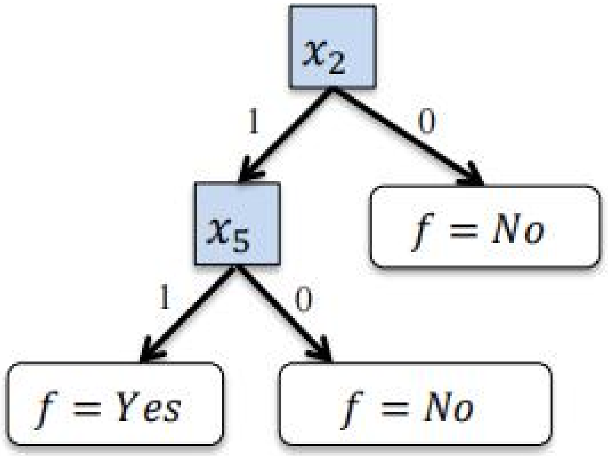
Problem Setting

Suppose $X = \langle x_1, \ldots x_n \rangle$

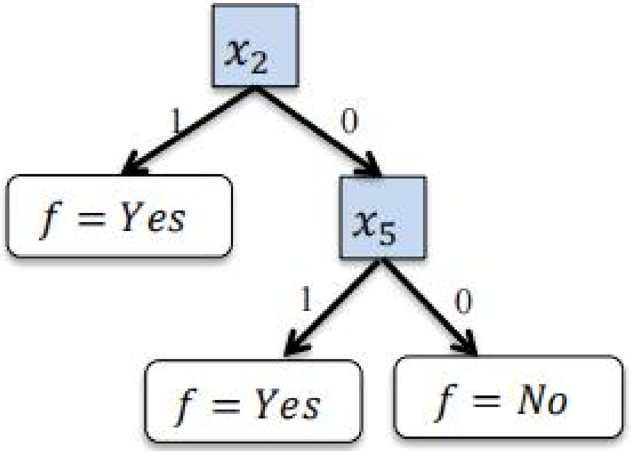
where x_i are boolean-valued variables

How would you represent the following as DTs?

$f(x) = x_2 \ AND \ x_5$?



$f(x) = x_2 \ OR \ x_5$



Hwk: How would you represent $X_2 X_5 \vee X_3 X_4 (\neg X_1)$?

Problem Setting

Input: Training labeled examples $\{(x^{(i)}, y^{(i)})\}$ of unknown target function f

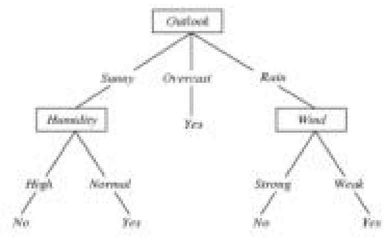
- Examples described by their values on some set of features or attributes
  - E.g. 4 attributes: *Humidity, Wind, Outlook, Temp*
    - e.g., <*Humidity=High, Wind=weak, Outlook=rain, Temp=Mild*>
  - Set of possible instances X (a.k.a instance space)

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

- Unknown target function $f : X \rightarrow Y$
  - e.g., $Y=\{0,1\}$ label space
  - e.g., 1 if we play tennis on this day, else 0

Output: Hypothesis $h \in H$ that (best) approximates target function f

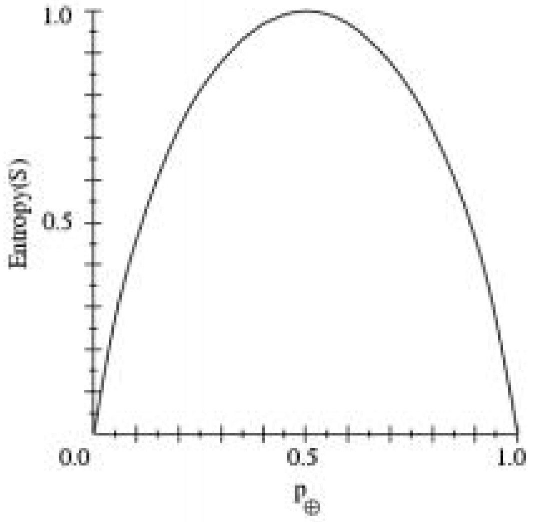
- Set of function hypotheses $H=\{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree



Sample Entropy of a Labeled Dataset

- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S.
- $p_\ominus$ is the proportion of negative examples in S.
- Entropy measures the impurity of S.

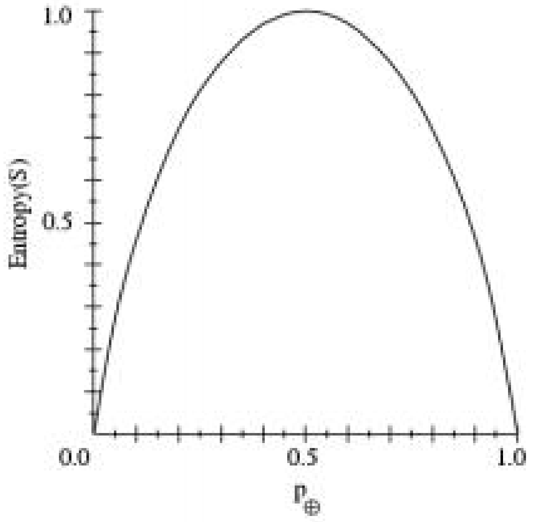
$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$



- E.g., if all negative, then entropy=0. If all positive, then entropy=0.
- If 50/50 positive and negative then entropy=1.
- If 14 examples with 9 positive and 5 negative, then entropy=.940

Sample Entropy of a Labeled Dataset

- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S.
- $p_{\ominus}$ is the proportion of negative examples in S.
- Entropy measures the impurity of S.



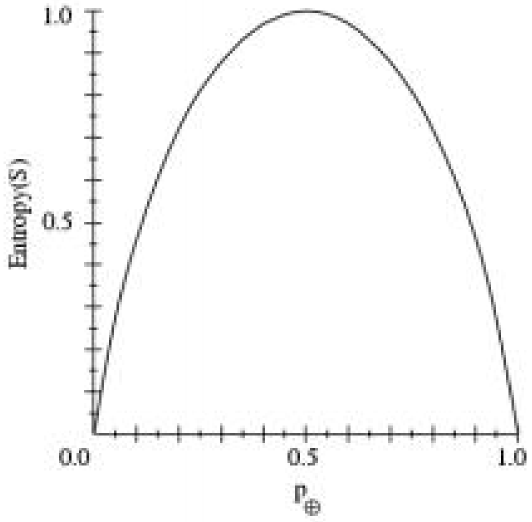
$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Interpretation from information theory: expected number of bits needed to encode label of a randomly drawn example in S.

- If S is all positive, receiver knows label will be positive, don't need any bits.
- If S is 50/50 then need 1 bit.
- If S is 80/20, then in a long sequence of messages, can code with less than 1 bit on average (assigning shorter codes to positive examples and longer codes to negative examples).

Sample Entropy of a Labeled Dataset

- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S.
- $p_\ominus$ is the proportion of negative examples in S.
- Entropy measures the impurity of S.



$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

If labels not Boolean, then $H(S) = \sum_{i \in Y} -p_i \log_2 p_i$

E.g., if c classes, all equally likely, then $H(S) = \log_2 c$

Information Gain

Given the definition of entropy, can define a measure of effectiveness of attribute in classifying training data:

Information Gain of A is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

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

entropy of original collection

Expected entropy after S is partitioned using attribute A

sum of entropies of subsets S_v weighted by the fraction of examples that belong to S_v.

Information Gain

Given the definition of entropy, can define a measure of effectiveness of attribute in classifying training data:

Information Gain of A is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

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

entropy of original collection

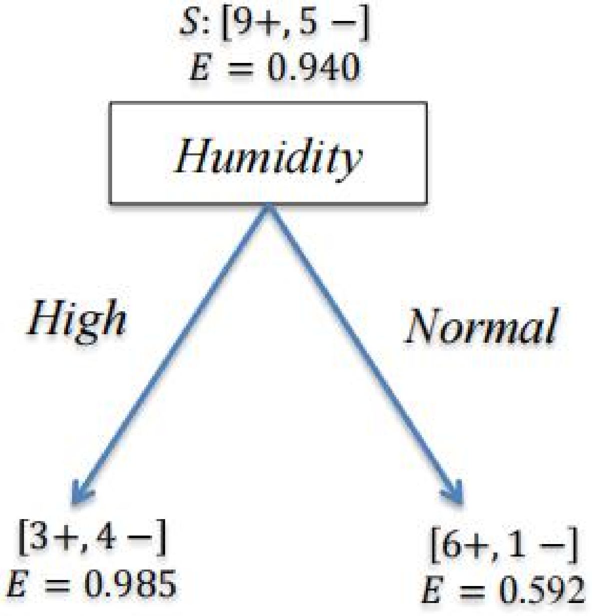
Expected entropy after S is partitioned using attribute A

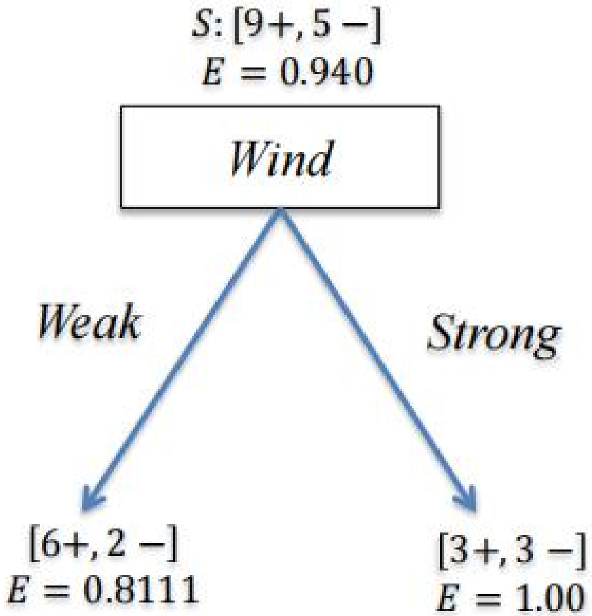
Gain(S,A) information provided about the target function, given the value of some other attribute A.

sum of entropies of subsets S_v weighted by the fraction of examples that belong to S_v.

Selecting the Next Attribute

- **Which attribute is the best classifier?**

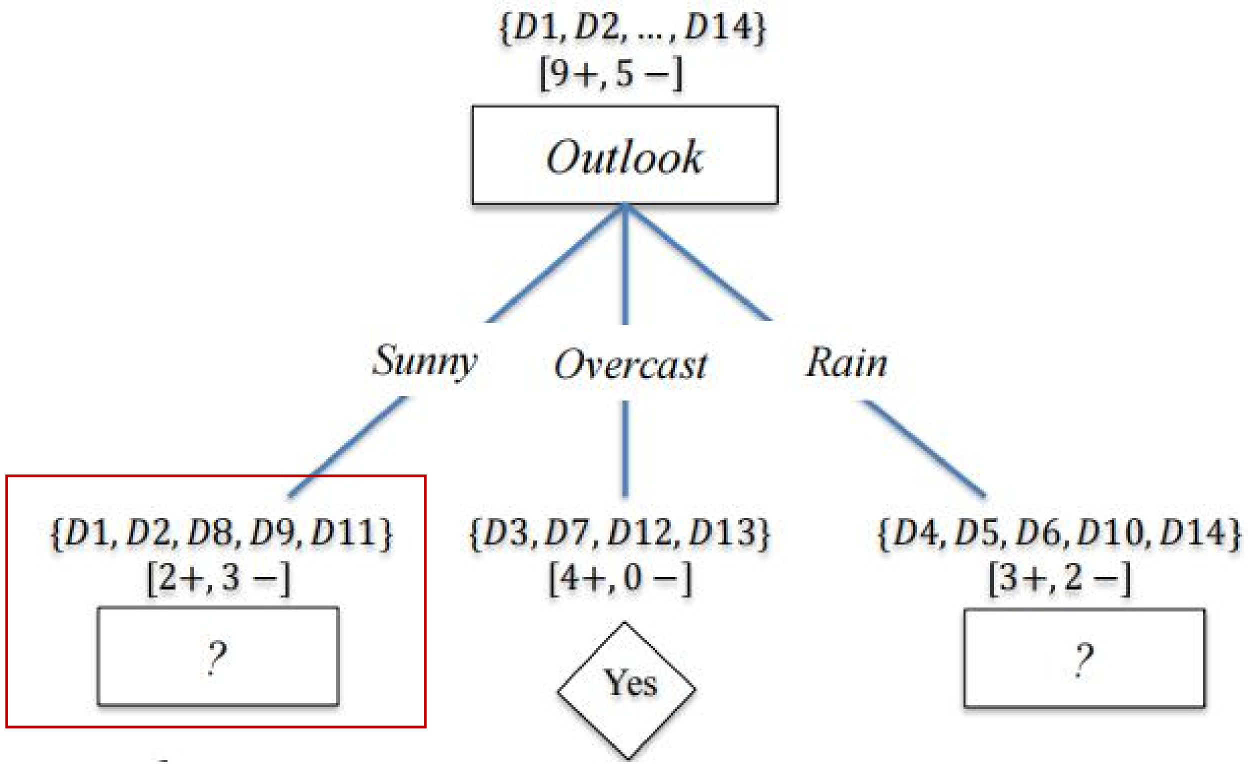




$$Gain(S, Humidity)$$
$$= .940 - \left(\frac{7}{14}\right).985 - \left(\frac{7}{14}\right).592$$
$$= .151$$

$$Gain(S, Wind)$$
$$= .940 - \left(\frac{8}{14}\right).811 - \left(\frac{6}{14}\right)1.0$$
$$= .048$$

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Which attribute should be tested here?

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

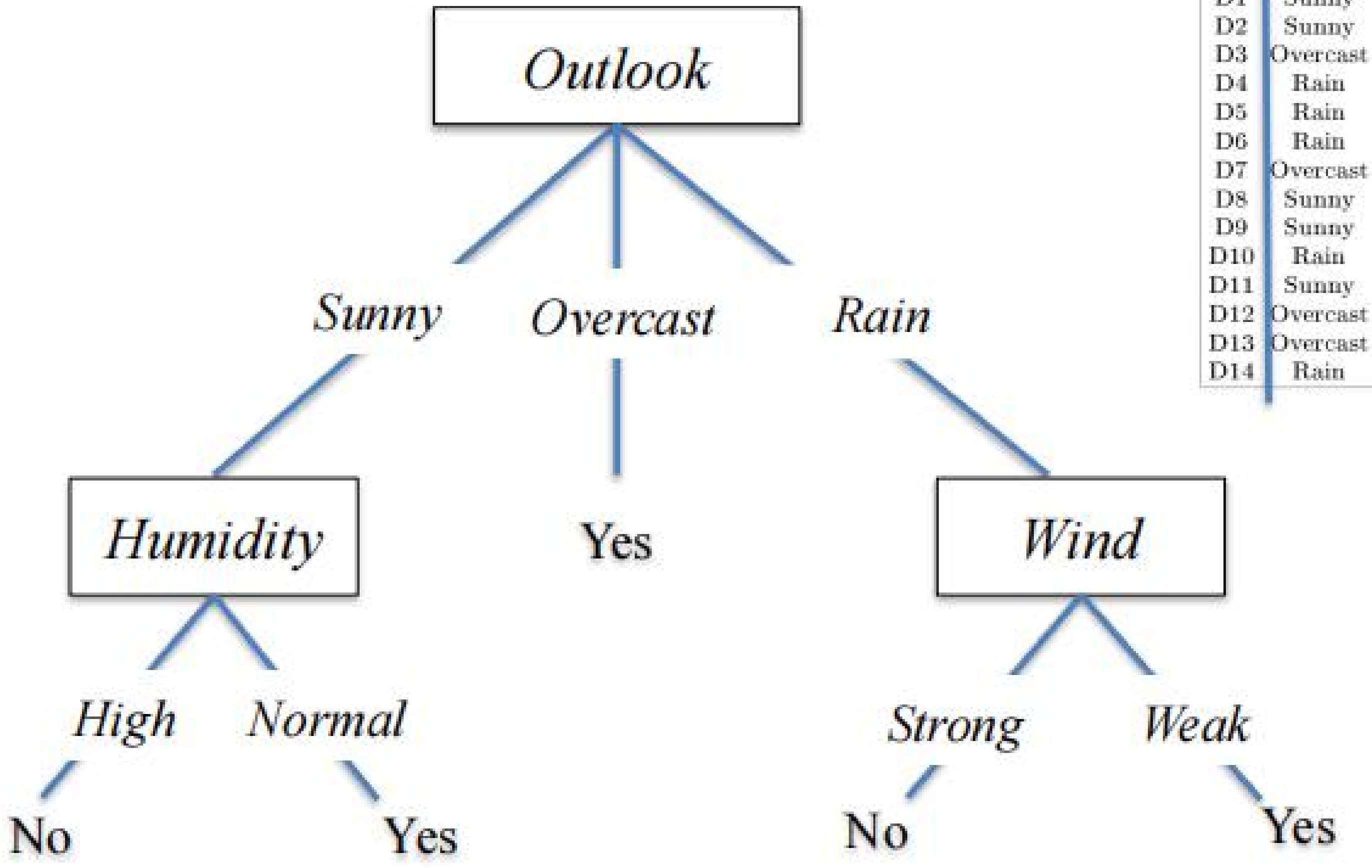
$$Gain(s_{sunny}, Humidity) = .970 - \left(\frac{3}{5}\right)0.0 - \left(\frac{2}{5}\right)0.0 = .970$$

$$Gain(s_{sunny}, Temperature) = .970 - \left(\frac{2}{5}\right)0.0 - \left(\frac{2}{5}\right)1.0 - \left(\frac{1}{5}\right)0.0 = .570$$

$$Gain(s_{sunny}, Wind) = .970 - \left(\frac{2}{5}\right)1.0 - \left(\frac{3}{5}\right).918 = .019$$

Final Decision Tree

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Comparison to some other classifiers

Advantages of decision trees over KNNs and neural nets

- Simple to deal with discrete features, missing values, and poorly scaled data
- Fast at test time
- More interpretable

Advantages of KNNs over decision trees

- Few hyperparameters
- Can incorporate interesting distance measures (e.g. shape contexts)

Advantages of neural nets over decision trees

- Able to handle attributes/features that interact in very complex ways (e.g. pixels)

Examples for Decision Tree

- What is the final decision tree ?

(1) Marine animal data

	Can survive without coming to surface?	Has flippers?	Fish?
1	Yes	Yes	Yes
2	Yes	Yes	Yes
3	Yes	No	No
4	No	Yes	No
5	No	Yes	No

Examples for Decision Tree

(2) Watermelon data

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否