Decision Tree Learning

Dr. Xiaowei Huang

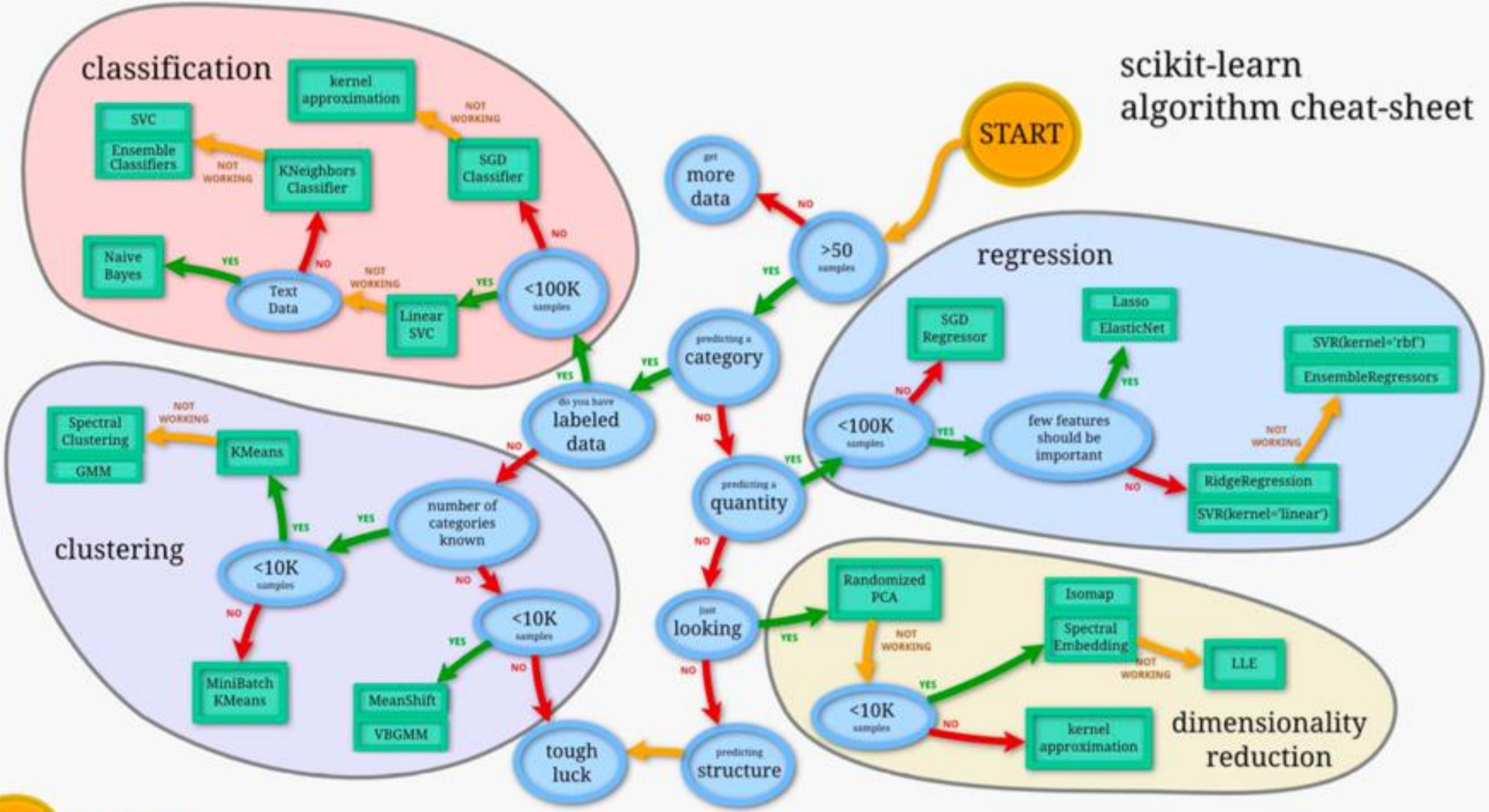
https://cgi.csc.liv.ac.uk/~xiaowei/

After two weeks,

- Remainders
  - Should work harder to enjoy the learning procedure

- slides
  - Read slides before coming, think them through after class
  - Slides will be adapted according to the prior classes, and will be updated to the newest asap after class

- Lab
  - First assignment to be disclosed this week
  - Learning experience: type in the code, and try to adapt the code to see the different results



Back

Note: only a subset of ML methods

Figure from scikit-learn.org

Decision Tree up to now,

- Decision tree representation
- A general top-down algorithm
- How to do splitting on numeric features

Top-down decision tree learning

MakeSubtree(set of training instances D)

C = DetermineCandidateSplits(D)

if stopping criteria met

make a leaf node N

determine class label/probabilities for N

else

make an internal node N

S = FindBestSplit(D, C)

for each outcome k of S

D_k = subset of instances that have outcome k

k^{th} child of N = MakeSubtree(D_k)

return subtree rooted at N

The focus of this lecture

In the next lecture

Explained in the last lecture

Topics

- Occam's razor
- entropy and information gain
- types of decision-tree splits

Finding the best split

- How should we select the best feature to split on at each step?

- Key hypothesis: the simplest tree that classifies the training instances accurately will work well on previously unseen instances

Occam’s razor

- attributed to 14th century William of Ockham
- “Nunquam ponenda est pluralitis sin necesitate”

- “Entities should not be multiplied beyond necessity”
- “when you have two competing theories that make exactly the same predictions, the simpler one is the better”

But a thousand years earlier, I said, “We consider it a good principle to explain the phenomena by the simplest hypothesis possible.”

Occam’s razor and decision trees

- Why is Occam’s razor a reasonable heuristic for decision tree learning?
  - there are fewer short models (i.e. small trees) than long ones
  - a short model is unlikely to fit the training data well by chance
  - a long model is more likely to fit the training data well coincidentally

Finding the best splits

- Can we find and return the smallest possible decision tree that accurately classifies the training set?

This is an NP-hard problem

[Hyafil & Rivest, *Information Processing Letters, 1976]*

- Instead, we'll use an information-theoretic heuristics to greedily choose splits

Expected Value (Finite Case)

- Let X be a random variable with a finite number of finite outcomes x_1, x_2, …, x_k occurring with probability p_1, p_2, …, p_k, respectively. The expectation of X is defined as

$$E[X] = p_1x_1 + p_2x_2 + \ldots + p_kx_k$$

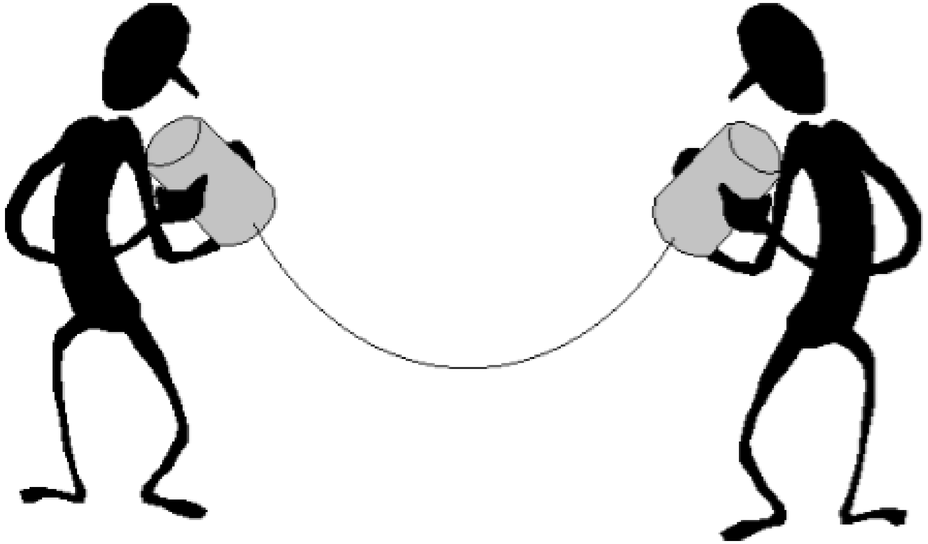
- Expectation is a weighted average

Expected Value Example

- Let X represent the outcome of a roll of a fair six-sided die
- Possible values for X include {1,2,3,4,5,6}
- Probability of them are {1/6, 1/6, 1/6, 1/6, 1/6, 1/6}
- The expected value is $\mathrm{E}[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = 3.5$

Information theory background

- consider a problem in which you are using a code to communicate information to a receiver
- example: as bikes go past, you are communicating the manufacturer of each bike



Information theory background

- suppose there are only four types of bikes
- we could use the following code

type	code
Trek	11
Specialized	10
Cervelo	01
Serrota	00

$$\frac{1}{4} \times 2 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2 + \frac{1}{4} \times 2 = 2$$

- expected number of bits we have to communicate:
  - 2 bits/bike

Information theory background

- we can do better if the bike types aren't equiprobable

Type/probability	# bits	code
$P(\text{Trek}) = 0.5$	1	1
$P(\text{Specialized}) = 0.25$	2	01
$P(\text{Cervelo}) = 0.125$	3	001
$P(\text{Serrota}) = 0.125$	3	000

Information theory background

Type/probability	# bits	code
$P(\text{Trek}) = 0.5$	1	1
$P(\text{Specialized}) = 0.25$	2	01
$P(\text{Cervelo}) = 0.125$	3	001
$P(\text{Serrota}) = 0.125$	3	000

- expected number of bits we have to communicate

$$0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75 < 2$$

Information theory background

Type/probability	# bits	code
$P(\text{Trek}) = 0.5$	1	1
$P(\text{Specialized}) = 0.25$	2	01
$P(\text{Cervelo}) = 0.125$	3	001
$P(\text{Serrota}) = 0.125$	3	000

$$0.5 \times 1 + 0.25 \times 2 + 0.125 \times 3 + 0.125 \times 3 = 1.75 < 2$$

$$= 0.5 \times \log_2 0.5 + 0.25 \times \log_2 0.25 + 0.125 \times \log_2 0.125 + 0.125 \times \log_2 0.125$$

$$= \; - \sum_{y \in \text{values}(Y)} P(y) \log_2 P(y)$$

Information theory background

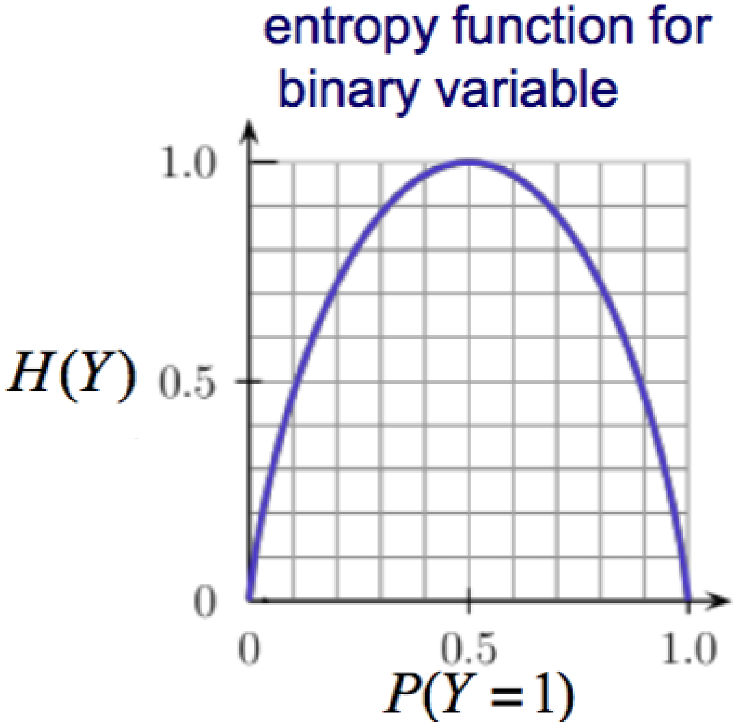
$$-\sum_{y \in \text{values}(Y)} P(y) \log_2 P(y)$$

- optimal code uses $-\log_2 P(y)$ bits for event with probability $P(y)$

Entropy

- entropy is a measure of uncertainty associated with a random variable
- defined as the expected number of bits required to communicate the value of the variable

$$H(Y) = -\sum_{y \in \text{values}(Y)} P(y) \log_2 P(y)$$



Conditional entropy

- **conditional entropy** (or equivocation) quantifies the amount of information needed to describe the outcome of a random variable given that the value of another random variable is known.
- What's the entropy of Y if we condition on some other variable X?

$$H(Y \mid X) = \sum_{x \in \text{values}(X)} P(X = x) H(Y \mid X = x)$$

similar as the expected value?

- Where

$$H(Y|X = x) = -\sum_{y \in \text{values}(Y)} P(Y = y|X = x) \log_2 P(Y = y|X = x)$$

Similar as entropy

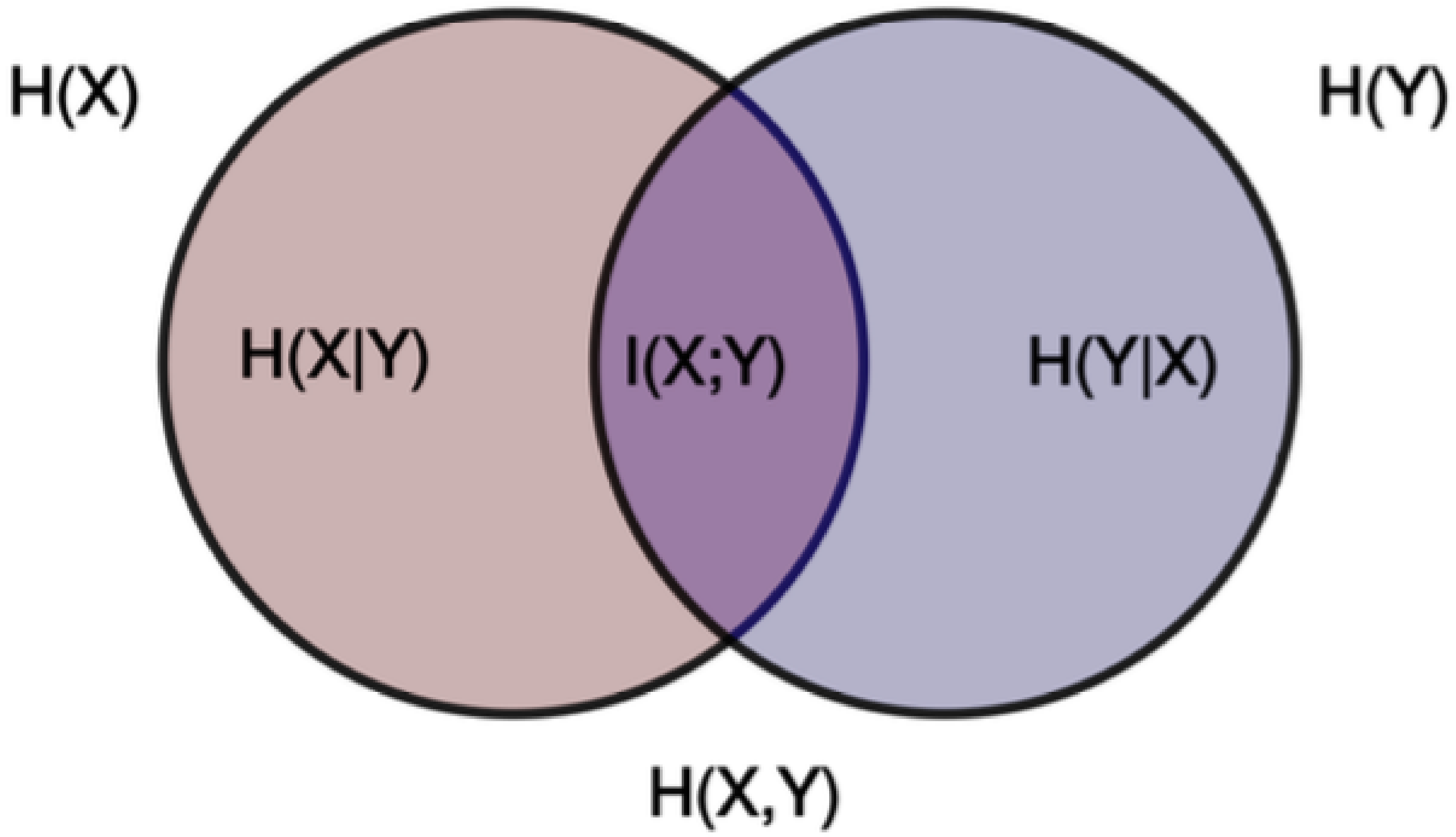
Information gain (a.k.a. mutual information)

- choosing splits in ID3: select the split S that most reduces the conditional entropy of Y for training set D

$$\text{InfoGain}(D,S) = H_D(Y) - H_D(Y \mid S)$$

D indicates that we're calculating probabilities using the specific sample *D*

Relations between the concepts



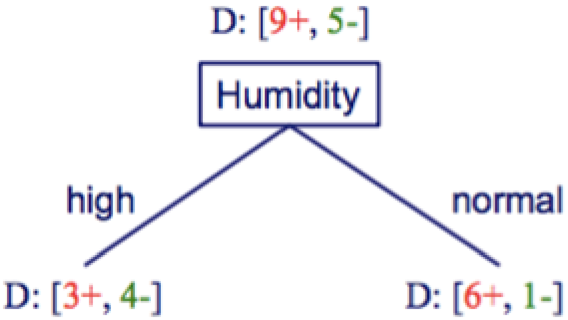
Information gain example

PlayTennis: 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

Information gain example

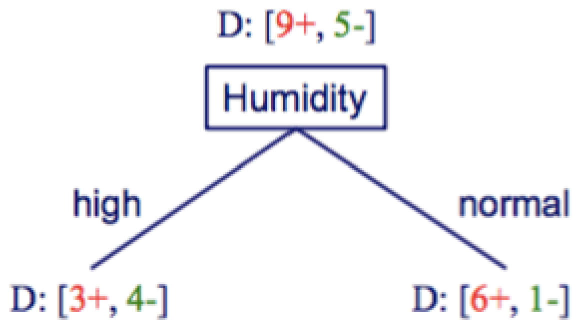
- What's the information gain of splitting on Humidity?



$$\text{InfoGain}(D, \text{Humidity}) = H_D(Y) - H_D(Y \mid \text{Humidity})$$

$$\text{InfoGain}(D,S) = H_D(Y) - H_D(Y \mid S)$$

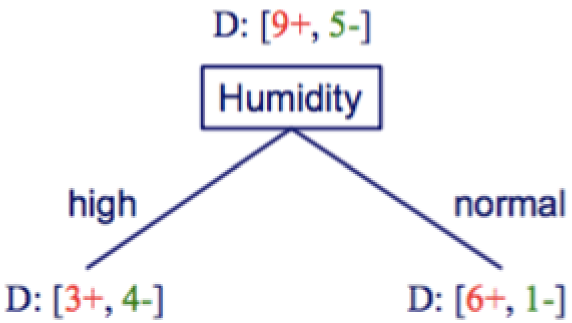
Information gain example



$$H_D(Y) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

$$H(Y) = -\sum_{y \in \text{values}(Y)} P(y)\log_2 P(y)$$

Information gain example



$H_D(Y \mid \text{Humidity})$ = P(Humidity=high)H_D(Y|Humidity=high) + P(Humidity=normal)H_D(Y|Humidity=normal)

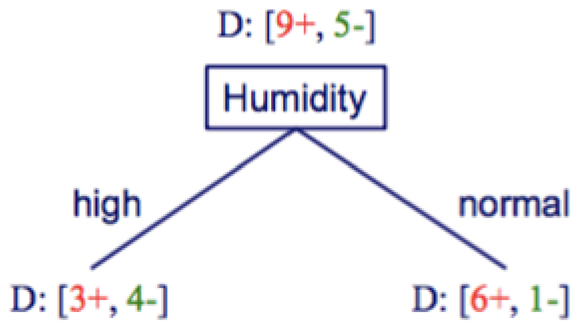
$$H(Y \mid X) = \sum_{x \in \text{values}(X)} P(X = x) H(Y \mid X = x)$$

$$H_D(Y \mid \text{high}) = -\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right)$$
$$= 0.985$$

$$H_D(Y \mid \text{normal}) = -\frac{6}{7}\log_2\left(\frac{6}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right)$$
$$= 0.592$$

$$H(Y|X = x) = -\sum_{y \in \text{values}(Y)} P(Y = y|X = x) \log_2 P(Y = y|X = x)$$

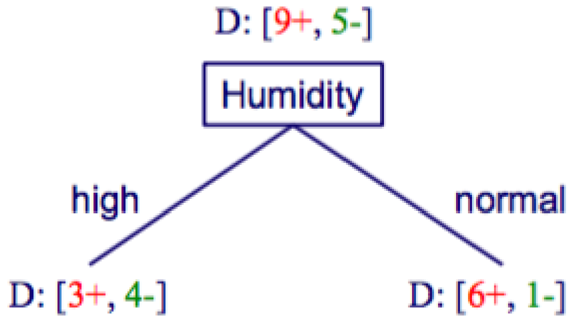
Information gain example

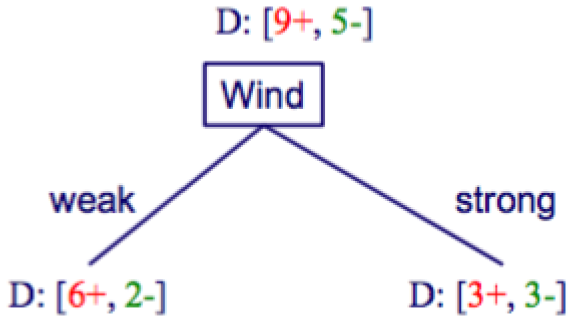


$$
\begin{aligned}
\text{InfoGain}(D, \text{Humidity}) &= H_D(Y) - H_D(Y \mid \text{Humidity}) \\
&= 0.940 - \left[\frac{7}{14}(0.985) + \frac{7}{14}(0.592)\right] \\
&= 0.151
\end{aligned}
$$

Information gain example

- Is it better to split on Humidity or Wind?





$H_D(Y \mid \text{weak}) = 0.811$ $\quad H_D(Y \mid \text{strong}) = 1.0$

$$\text{InfoGain}(D, \text{Humidity}) = 0.940 - \left[\frac{7}{14}(0.985) + \frac{7}{14}(0.592)\right]$$
$$= 0.151$$

$$\text{InfoGain}(D, \text{Wind}) = 0.940 - \left[\frac{8}{14}(0.811) + \frac{6}{14}(1.0)\right]$$
$$= 0.048$$

One limitation of information gain

- information gain is biased towards tests with many outcomes
- e.g. consider a feature that uniquely identifies each training instance
  - splitting on this feature would result in many branches, each of which is "pure" (has instances of only one class)
  - maximal information gain!

Gain ratio

- to address this limitation, C4.5 uses a splitting criterion called *gain ratio*
- gain ratio normalizes the information gain by the entropy of the split being considered

$$\text{GainRatio}(D,S) = \frac{\text{InfoGain}(D,S)}{H_D(S)} = \frac{H_D(Y) - H_D(Y \mid S)}{H_D(S)}$$