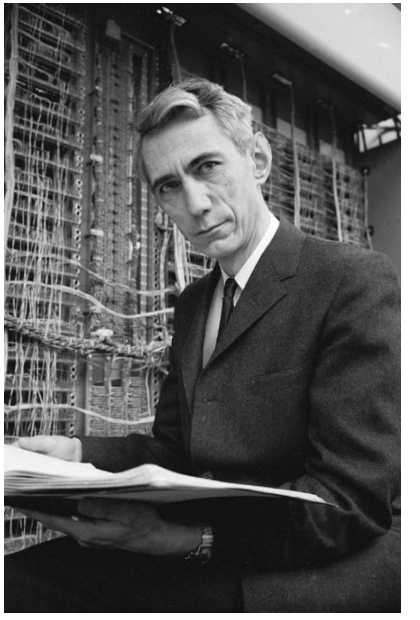
Classifiers, Decision Trees, Entropy



Ariel Shamir

Zohar Yakhini

IDC

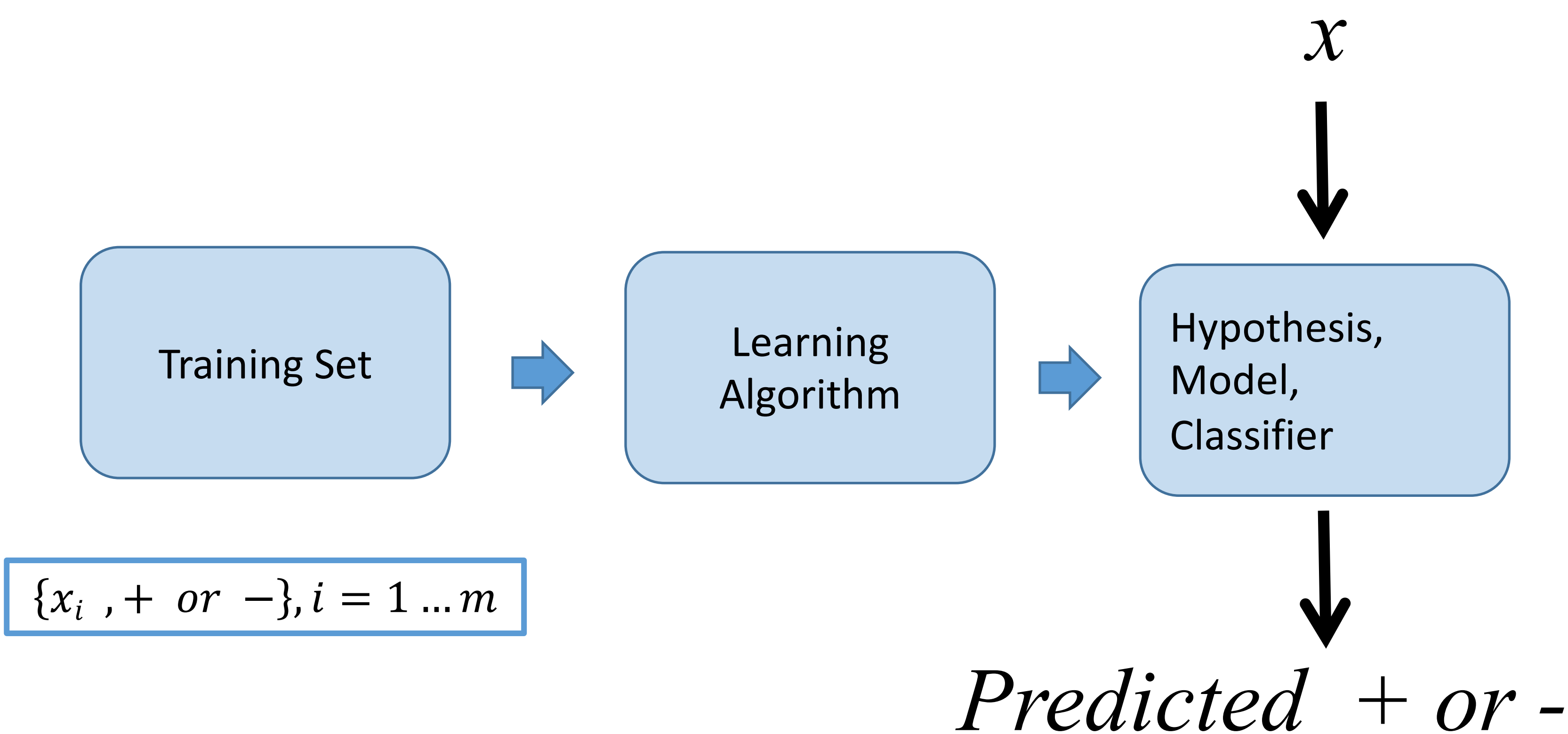
Outline

- Classification
- Formalities
- Decision trees
- Constructing a decision tree
- Impurity/uncertainty criteria & Goodness of Split
- Entropy, Gini, information gain
- More issues

Types of Learning Questions

- Regression
  - Given $\{x_i, y_i\}$ find f such that $y = f(x)$
- Classification
  - Given $\{x_i, y_i\}$ where $y_i \in \{0,1\}$ as training data, determine for a new x if $x \in C_0$ or $x \in C_1$
- Density Estimation
  - Given $\{x_i\}$ find a PDF that best explains the data
- Clustering
  - Given $\{x_i\}$ find a partition to subsets under some constraints

Classification



Formalities (Discrete Space)

- X is a space of instance representations $x \in X$ (feature vectors)
- A **concept** c is a subset in this space.
  That is:
  $c \in 2^X = H$ (the power set of X)
- A **training set** D is a set of pairs <x,c(x)> where $x \in X$ and $c(x) \in \{+1/-1\}$
  (or later maybe several categories)
- Consistent learning:
  We are looking for a **hypothesis** (or **model**) $h \in H$ such that $h(x)=c(x)$ for all x represented in D
- Agnostic learning:
  We are looking for a **hypothesis** (or **model**) $h \in H$ such that minimizes the error as reflected in D

Example

- x is defined by an (ordered) set of attributes (features) each having a range of values: $x=(x_1,x_2,\ldots x_n)$ where each $x_i \in A_i = \{a_1,a_2,\ldots a_{m_i}\}$

- Instances:

  <male, tall, brown eyes, black hair>

  <female, short, blue eyes, blond hair>

  <female, tall, green eyes, red hair>

  <female, tall, green eyes, gray hair>

  <male, tall, green eyes, red hair>

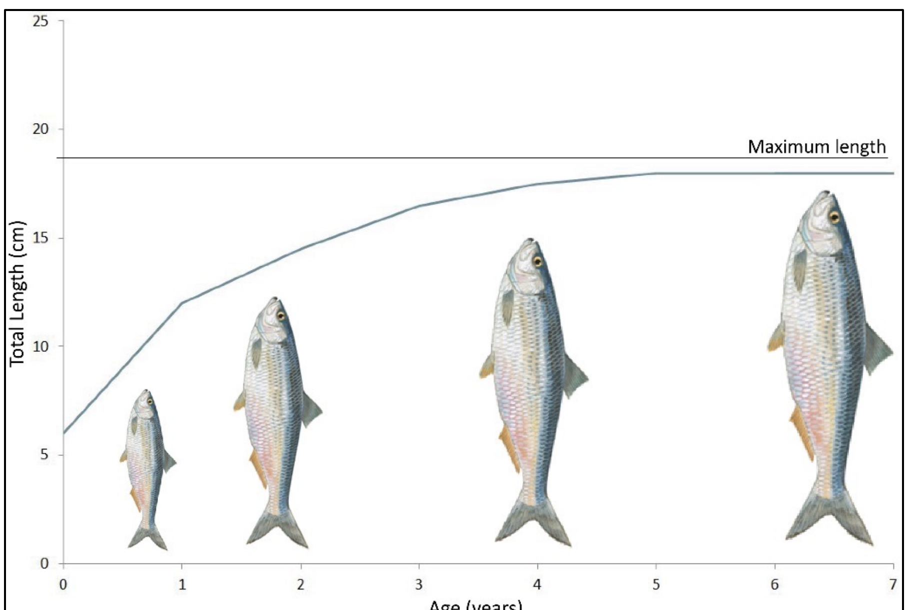
Discrete Attributes

- Gender: {Male, Female}
- Height: {Tall, Medium, Short}
- Eyes: {Blue, Brown, Green, Hazel, Yellow}
- Hair: {Gray, Black, Blond, Red, Brown}

Possible concept: “all cute cats”

Discrete vs. Continuous Feature Spaces

- A feature/attribute can have discrete values such as "eye color" (blue, green, brown,…)
- A feature can have a continuous value such as height, length, age, temperature, sq-ft of a house



If we have n features that can take on real values, our ***instance space*** is $\mathbb{R}^n$.

Any instance is simply a point in this space.

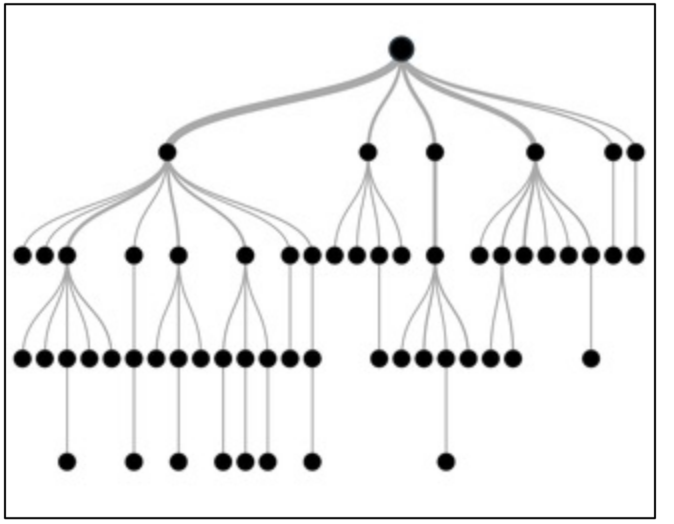
Concept as a Dichotomy

- A **dichotomy** is a partition of a set into two parts (subsets) that are**:**
  - o **<u>jointly exhaustive</u>**: everything must belong to one part or the other, and
  - o **<u>mutually exclusive</u>**: nothing can belong simultaneously to both parts.
- For simple +/- classifications we can define a concept as a dichotomy over the space X:

$c = X^+$, all instances labeled as +

$X = X^+ \cup X^-$ and $X^+ \cap X^- = \emptyset$

Decision Trees



A first approach to classification

Attributes (features):
- Size ϵ {big, med, small}
- Shape ϵ {thin, round}
- Color ϵ {green, yellow, red}

They all have a corresponding set of possible attribute values

A Decision Tree

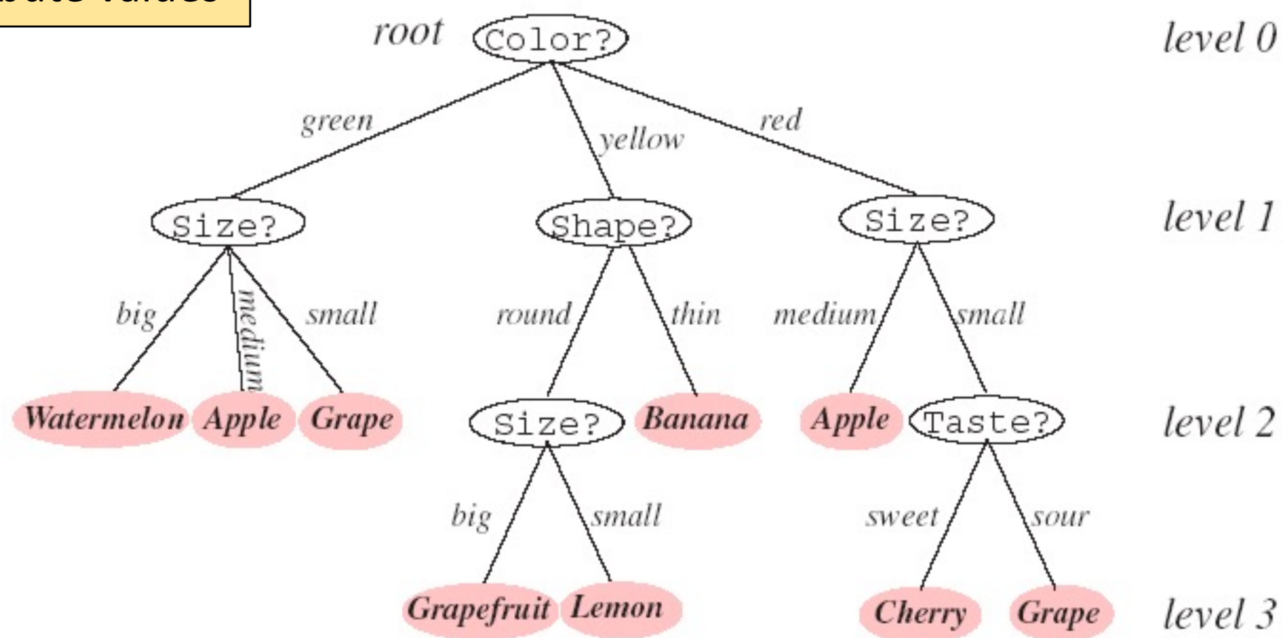
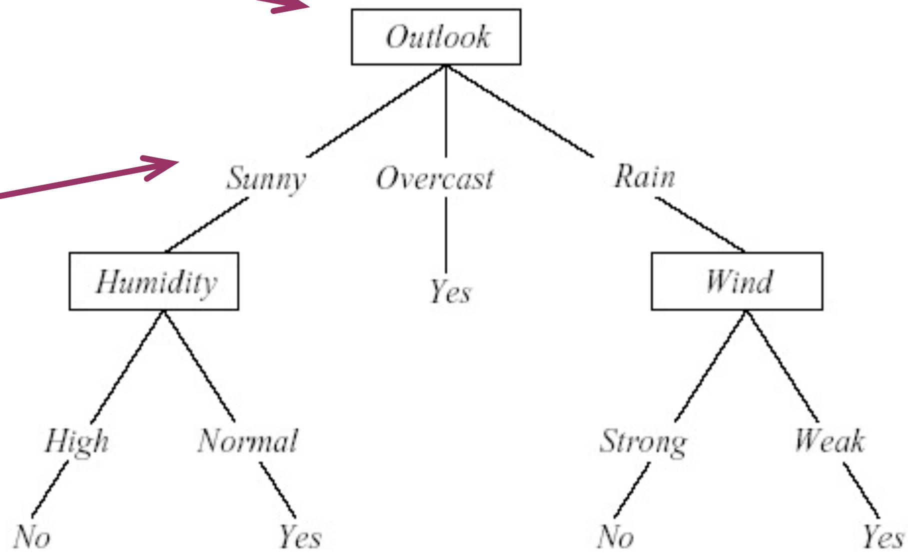


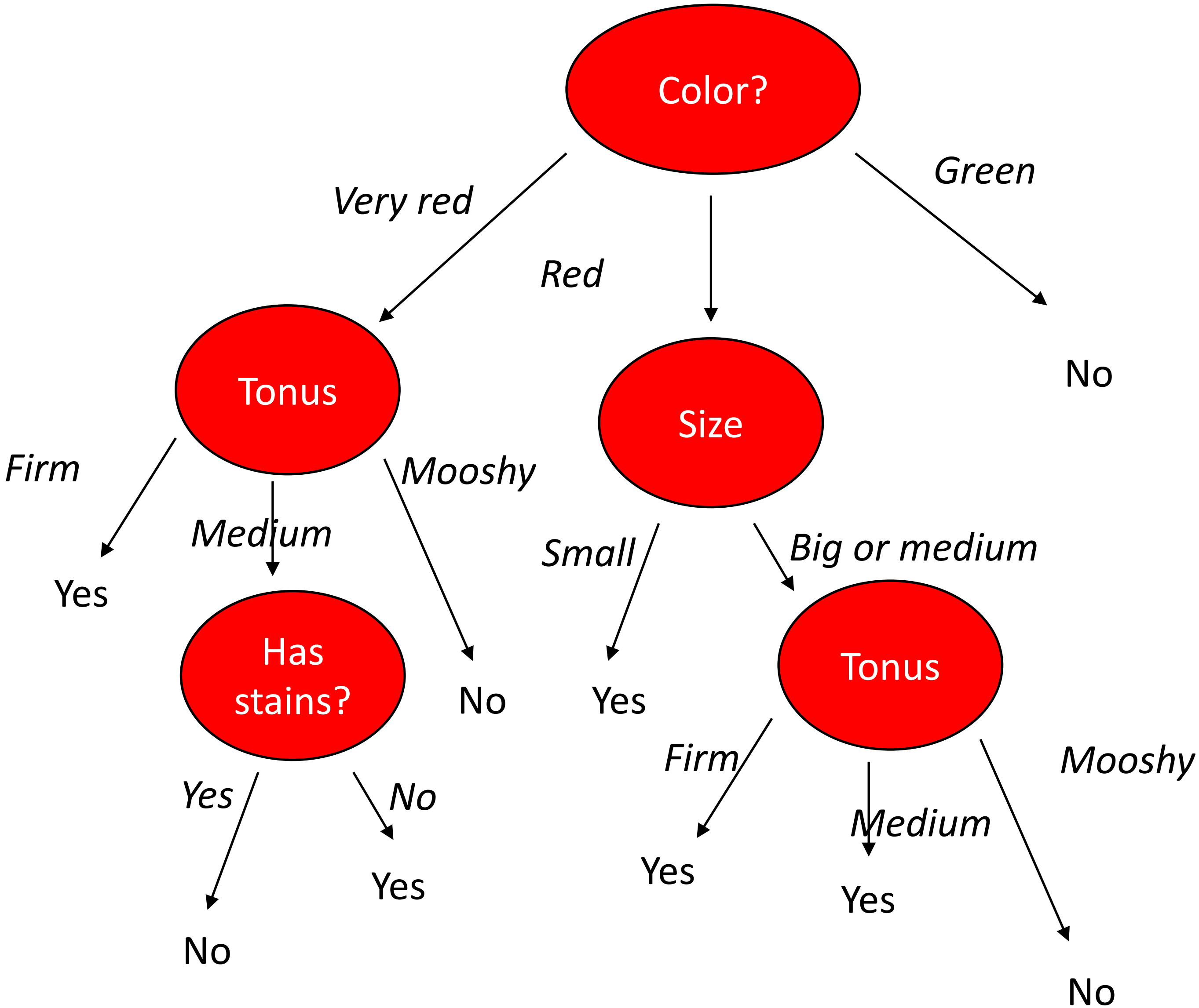
FIGURE 8.1. Classification in a basic decision tree proceeds from top to bottom. The questions asked at each node concern a particular property of the pattern, and the downward links correspond to the possible values. Successive nodes are visited until a terminal or leaf node is reached, where the category label is read. Note that the same question, `Size?`, appears in different places in the tree and that different questions can have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled by the same category (e.g., **Apple**). From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Decision Trees Components

- Each non-terminal node tests an attribute
(the decision attribute at that node)
- Each branch corresponds to an attribute value
- Each leaf node assigns a classification value



Tomatoes or Not a Tomato? (Binary Concept = Dichotomy)



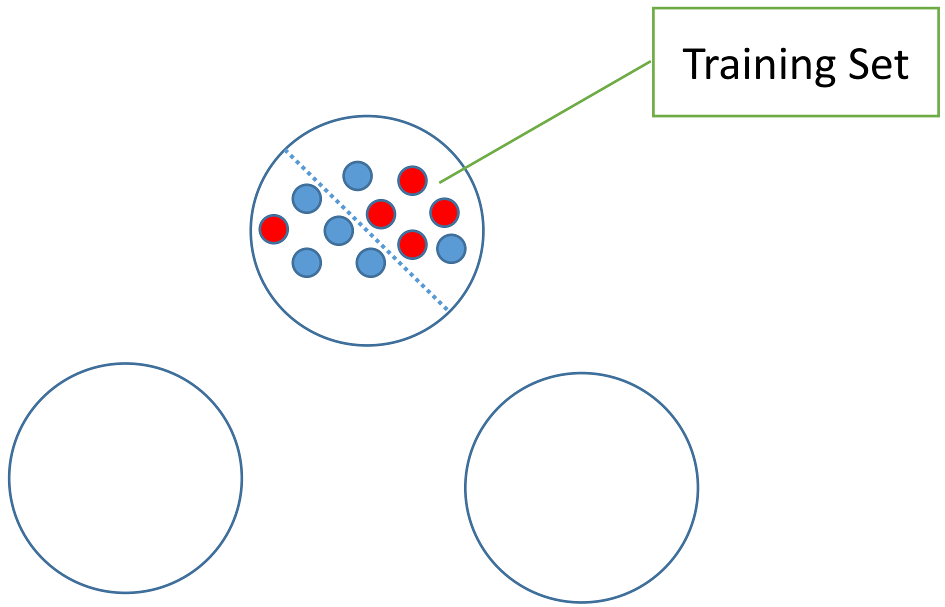
Example: Another Dichotomy

- Play Tenis Today (or Not)?

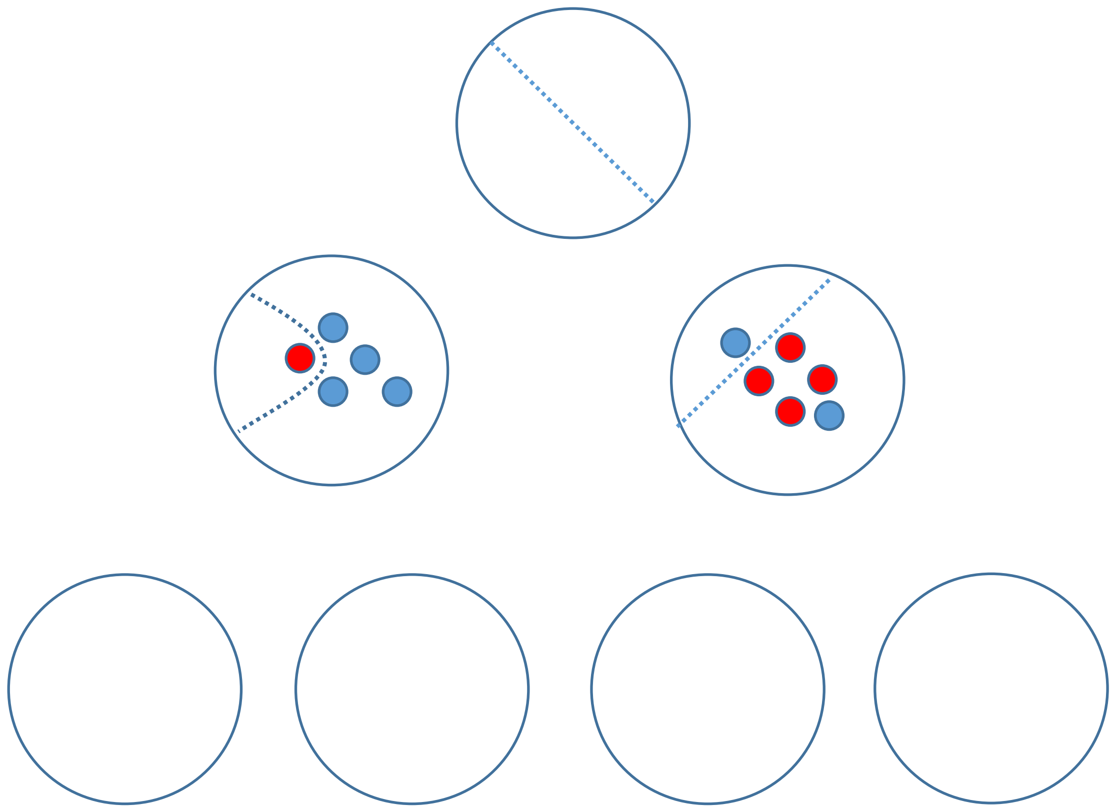
Training Set

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

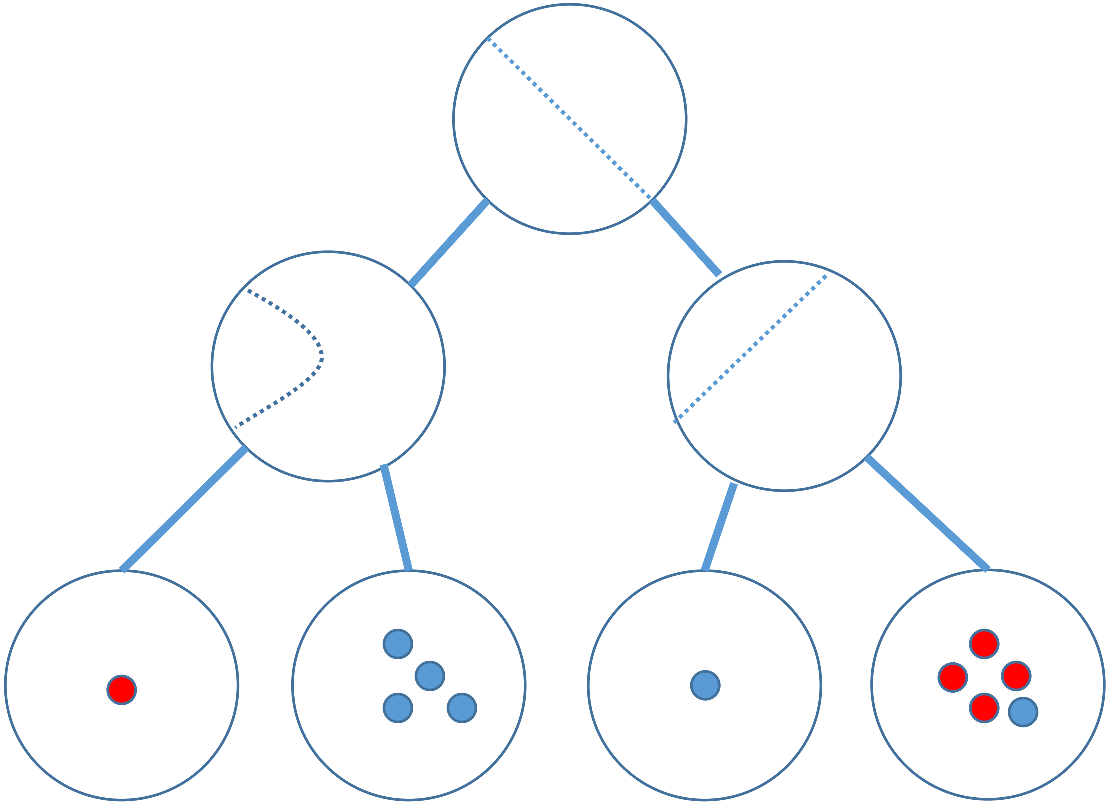
Growing a Tree



Creating a Tree



Creating a Tree



Learning (constructing a tree)

Will be relaxed

Will be discussed

Create the root node with all samples

Insert the node to initialize a queue, Q

While there are more incomplete nodes in Q do:

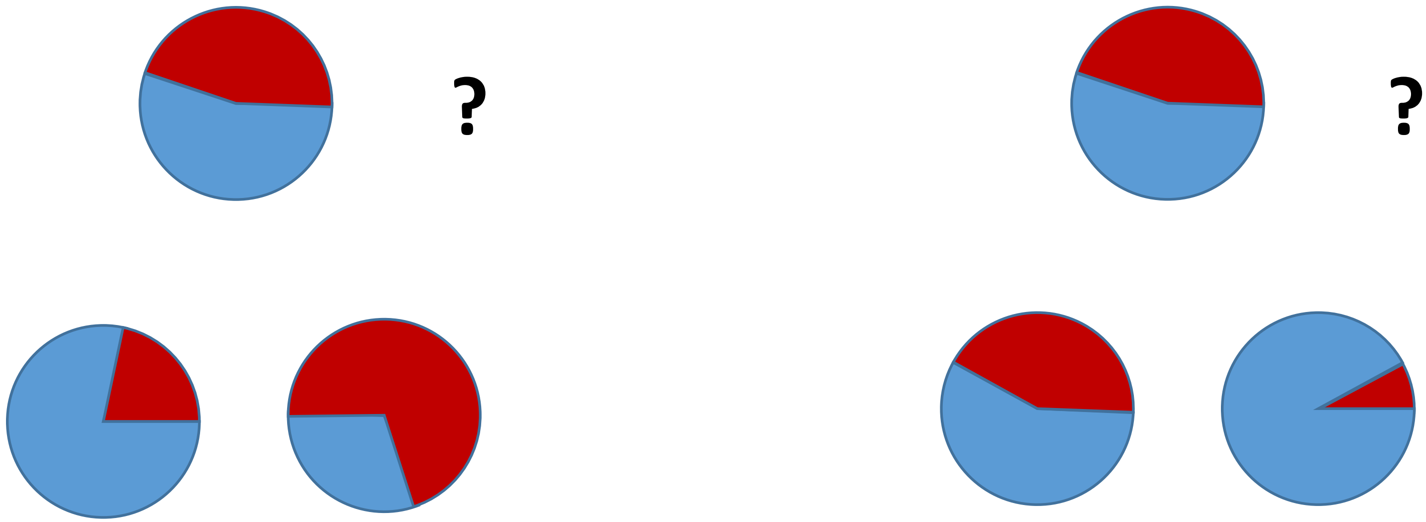
- Get next ***node n***
- If the training examples in ***n*** are perfectly classified
  Then mark node as complete and continue to next node in Q
- *Else* ***A*** ← the "best" decision **attribute** (and **boundary values**) for the set in ***n***
  - Assign ***A*** to be the decision attribute for ***n***
  - For each **interval of values** of ***A***, create a new descendant of ***n***
  - Distribute training examples to descendant nodes
  - Insert all (non empty) descendant nodes to Q

Which is the Best Attribute??

The main focus of the learning algorithm for decision trees is deciding <u>which attribute is best to use when we split the data</u>

(what should we use as the decision attribute at a given node)

Which is the Best Attribute??



The colors represent the distribution of the class values.
Circle sizes do NOT represent the size of the nodes

Best Attribute?

- One that takes us as close as possible to perfect classification, considering all descendants, <u>in the training data</u>.
- How do we measure this property?
- If we can divide all samples, using one question (attribute), into two groups of "purely +" and "purely –" samples then that's the best. (<u>full certainty</u>)
- Alternatively, we want to **reduce** the **impurity** or the **confusion** or the **uncertainty** that still remains in the next step
- **How do we measure impurity?**

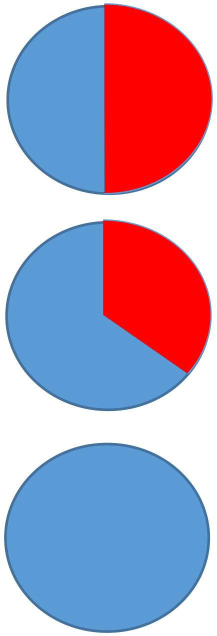
Uncertainty Criteria

An <u>uncertainty measure</u> is a function $\varphi\colon [0,1]^k \to \mathbb{R}$ that is defined for probability distributions $P = (p_1, \dots, p_k) \in [0,1]^k$, $\sum_{i=1}^{k} p_i = 1$.
An uncertainty measure will satisfy the following conditions:

- $\varphi(P) \geq 0$
- The minimal value is attained when $\exists i$ s.t $p_i = 1$.
- The maximal value is attained when $1 \leq \forall i \leq k$, $p_i = {}^{1}/_{k}$.
- It is symmetric with respect to the components of P
- It is smooth (infinitely differentiable) in the relevant range

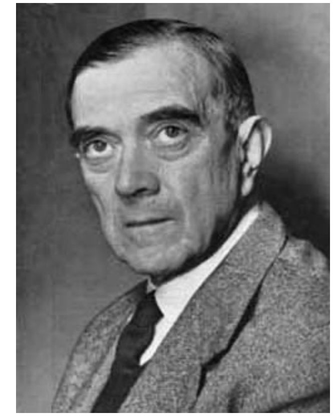
Example for k=2

- We have instances with two values + and –
- If in a node S we have P(+)=P(-)=1/2 then the impurity $\phi(S)$ is at maximum
(and it is the worst for us in the decision tree context)
- If S has a blue majority (say) then the impurity is reduced (better …)
- If S has only blue or only red then the impurity $\phi(S)$ is at minimum – there is full certainty in the node
(and that is the best descendant for us)



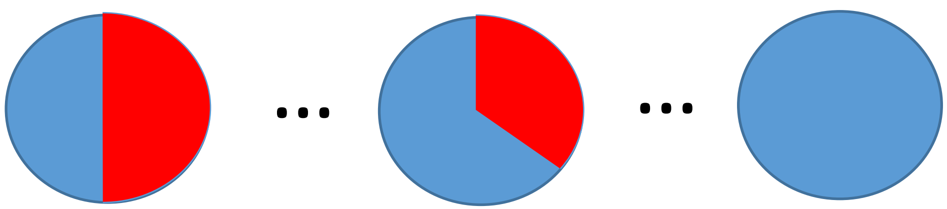
Gini Impurity

Corrado Gini
1884-1965
Italian statistician
and sociologist



- Measures dispersion.
- Ranges from 0 to 1. Low values mean less dispersion while high value mean more dispersion:

$$G(S) \equiv 1 - \sum_{i=1}^{c} (p_i)^2 = 1 - \sum_{i=1}^{c} \left(\frac{|S_i|}{|S|}\right)^2$$



$$G(S) = 1 - \frac{1}{4} - \frac{1}{4} = \frac{1}{2}$$

If we have k values equi-distributed then
G(S)=1- k(1/k)^2 =1-1/k=~1

$$G(S) = 1 - 1 - 0 = 0$$

If we have k values but only one value really present then
G(S)=1-(k/k)^2 = 0

Main Idea for Choosing the Splitting Attribute (Feature)

- For all attributes:
  - Measure the uncertainty/impurity before splitting according to attribute
  - Measure the uncertainty/impurity after (in the children)
- Choose the attribute that produces the largest difference!

Using Gini Impurity for Splitting

We search for the attribute A that will yield the best average Gini after the split, and therefore define:

$$GiniGain(S,A) = \Delta G(S,A) \equiv G(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} G(S_v)$$

Change in Impurity

Impurity Before Split

Weighted Average of Impurity of All Groups After Splitting

More Generally: Split Quality

- Given an uncertainty criterion $\phi(S)$
- The Split-Quality due to a discrete attribute A is defined as the **reduction in uncertainty** $\phi(S)$ when comparing the values before and after the full splitting of S according to the values of the attribute A:

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

Uncertainty Before Split

Weighted Average of Uncertainty After Split

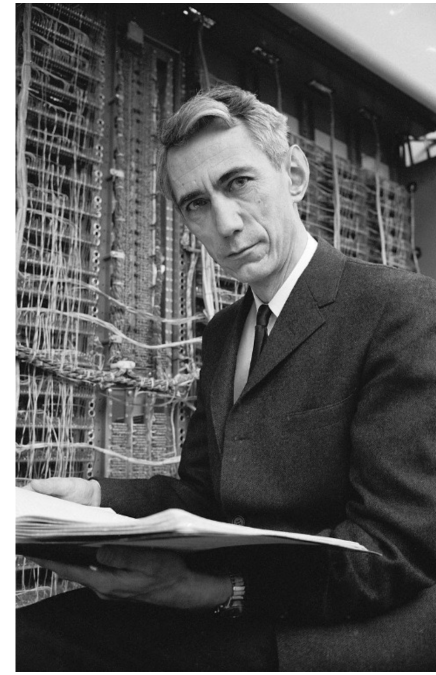
Entropy Example: A ”System” of Balls

- Assume that you have 10 balls that can either be black or white with probability 0.5.
- If I tell you all of them are white, then you have full information, no uncertainty – then the **entropy** is 0.
- If you know half of them are white, you know some information but still you have $\binom{10}{5} = \frac{10!}{5!5!}$ possibilities – a lot of uncertainty so **entropy** is large.
- If you have no information about the system: each ball can be black or white, then you have 2^{10} possibilities which means your uncertainty is maximal… and so is the **entropy.**

Another Measure: Shannon's Entropy

Shannon's entropy for a random variable X that takes n distinct values with probabilities p_i:

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i)$$



Claude Shannon
1916-2001
American mathematician, electrical engineer, statistician.
The founder of the field of Information Theory

Claude Shannon: ~~A~~ The Mathematical Theory of Communication
Bell System Technical Journal, 1948

Entropy

- Measures the average information content associated to an outcome of a random variable
- Intuitively: a measure for uncertainty
- Consider a set S and data with c (possibly more than 2) classes
- p_i is the proportion of class i in the set S
- The Entropy of S:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i = \sum_{i=1}^{c} -\frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

When is Entropy Maximal?

Johan Jensen's Inequality 1906 (inverted):

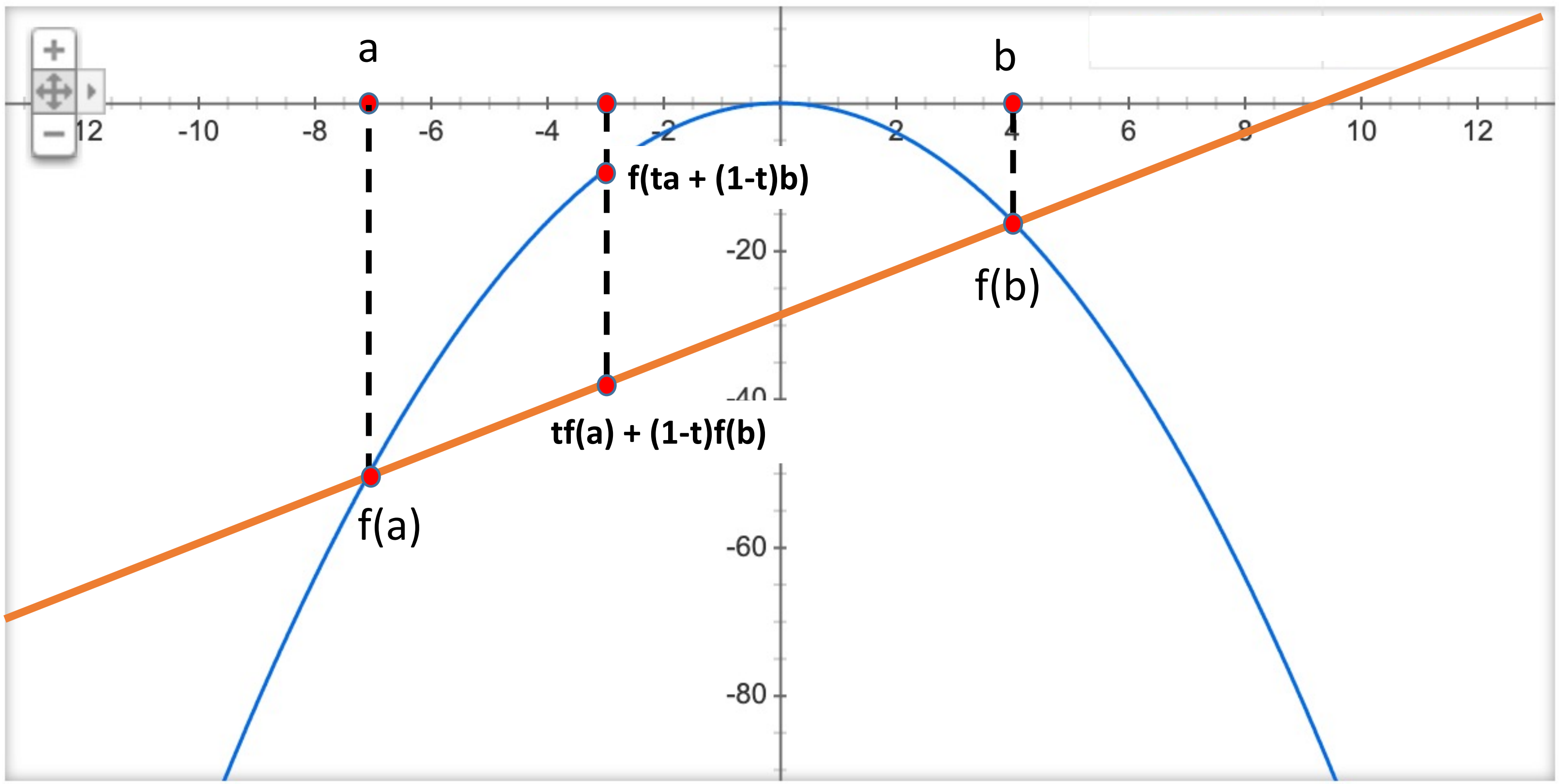
The following holds for any concave ("sad") function f:

$\forall x_1, \ldots x_k$ and $\forall \lambda_1 \ldots \lambda_k \in [0,1]$ s.t $\sum \lambda_i = 1$.

$$\sum_{i=1}^{k} \lambda_i f(x_i) \leq f\left(\sum_{i=1}^{k} \lambda_i x_i\right).$$

See the excellent explanation (of a generalized version) by John Tsitsiklis, MIT: https://ocw.mit.edu/RES-6-012S18

Average of the Function ≤ Function of the Average



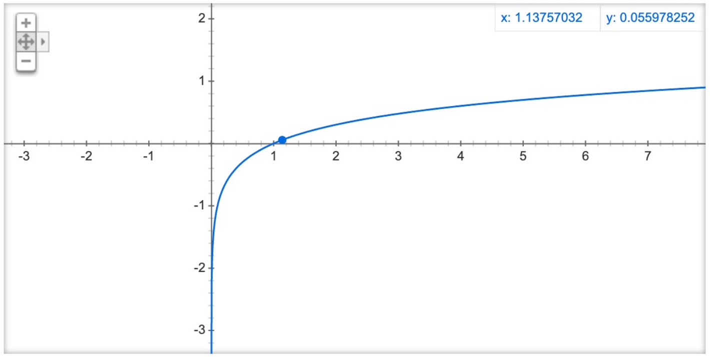
$$tf(a) + (1-t)f(b) \leq f(ta + (1-t)b)$$

When is Entropy Maximal?

$f(x) = \log(x)$ is a concave ("sad") function.

Since:

$$f''(x) = -\frac{1}{x^2} < 0$$



When is Entropy Maximal?

$$H(p_1, \ldots p_k) = -\sum_{i=1}^{k} p_i \log(p_i) = \sum_{i=1}^{k} p_i \log\left(\frac{1}{p_i}\right)$$

$$\leq \log\left(\sum_{i=1}^{k} p_i \cdot \frac{1}{p_i}\right) = \log k$$

$$\log\left(\frac{1}{x}\right) = -\log(x)$$

Convex combination of the function values

The function evaluated at a convex combination of the points

$$\sum_{i=1}^{k} \lambda_i f(x_i) \leq f\left(\sum_{i=1}^{k} \lambda_i x_i\right)$$

When is Entropy Maximal?

$$H\left(P = \left(\frac{1}{k}, \dots, \frac{1}{k}\right)\right) = \log k$$

Like the entropy of a fair coin or a fair die.

Information Gain = Split-Quality using Entropy

Gain(S,A) = expected reduction in entropy due to dividing according to attribute A:

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

$$\text{InfoGain}(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

Should I Play Tennis Today?

Training Data

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

$$\frac{9}{14}, \frac{5}{14}$$

Outlook Attribute

$$\frac{5}{14}, \frac{5}{14}, \frac{4}{14}$$

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

Temperature Attribute

$$\frac{4}{14}, \frac{4}{14}, \frac{6}{14}$$

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

Humidity Attribute

$$\frac{7}{14}, \frac{7}{14}$$

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

Wind Attribute

$$\frac{8}{14}, \frac{6}{14}$$

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

Base Entropy

$$E = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14} = 0.904$$

Training Data

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

$$\frac{9}{14}, \frac{5}{14}$$

Selecting Humidity Attribute

$$E = -\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7} = 0.985$$

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
D8	Sunny	Mild	High	Weak	No
D12	Overcast	Mild	High	Strong	Yes
D14	Rain	Mild	High	Strong	No

$\frac{4}{7}, \frac{3}{7}$

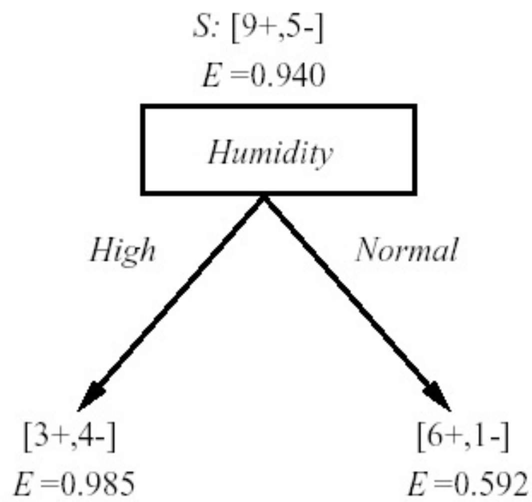
Selecting Humidity Attribute

$$E = -\frac{1}{7}\log\frac{1}{7} - \frac{6}{7}\log\frac{6}{7} = 0.592$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

$$\frac{1}{7}, \frac{6}{7}$$

Selecting Humidity Attribute



$$Gain(humidity) = 0.940 - \frac{7}{14}0.985 - \frac{7}{14}0.592$$

Selecting Wind Attribute

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes

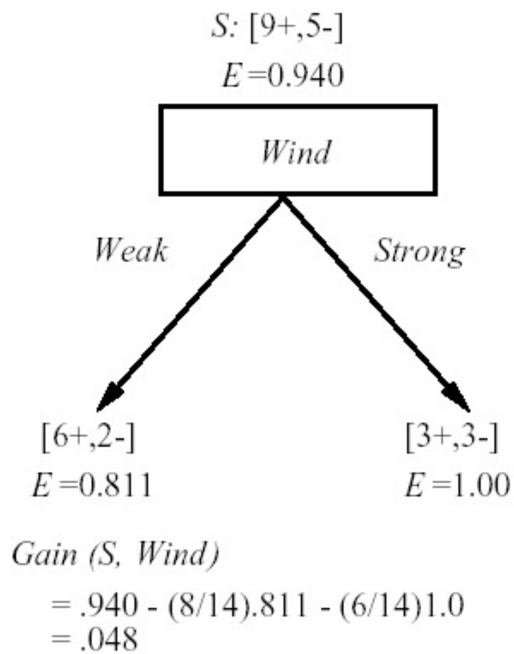
$$\frac{2}{8}, \frac{6}{8}$$

Selecting Wind Attribute

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D2	Sunny	Hot	High	Strong	No
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D14	Rain	Mild	High	Strong	No

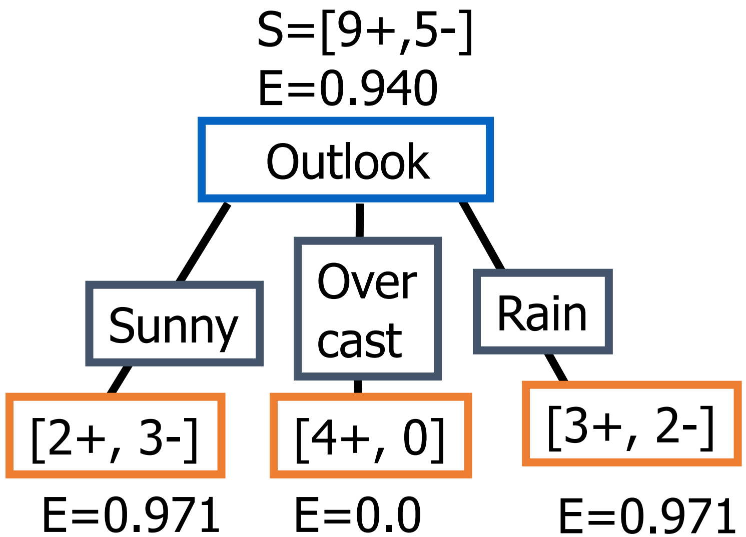
$$\frac{3}{6}, \frac{3}{6}$$

Selecting Wind Attribute



$$Gain(wind) = 0.940 - \frac{8}{14}0.811 - \frac{6}{14}1.0$$

Selecting Outlook Attribute



$$Gain(outlook) = 0.940 - \frac{5}{14}0.971 - \frac{4}{14}0.0 - \frac{5}{14}0.097$$

Selecting the First Attribute

The information gain values for the 4 attributes are:

- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

where S denotes the collection of training examples

After Split with Outlook Attribute

	Day	Outlook	Temperature	Humidity	Wind	PlayTennis
	D1	Sunny	Hot	High	Weak	No
	D2	Sunny	Hot	High	Strong	No
Node 1	D8	Sunny	Mild	High	Weak	No
	D9	Sunny	Cool	Normal	Weak	Yes
	D11	Sunny	Mild	Normal	Strong	Yes
	D3	Overcast	Hot	High	Weak	Yes
Node 2	D7	Overcast	Cool	Normal	Strong	Yes
	D12	Overcast	Mild	High	Strong	Yes
	D13	Overcast	Hot	Normal	Weak	Yes
	D4	Rain	Mild	High	Weak	Yes
	D5	Rain	Cool	Normal	Weak	Yes
Node 3	D6	Rain	Cool	Normal	Strong	No
	D10	Rain	Mild	Normal	Weak	Yes
	D14	Rain	Mild	High	Strong	No

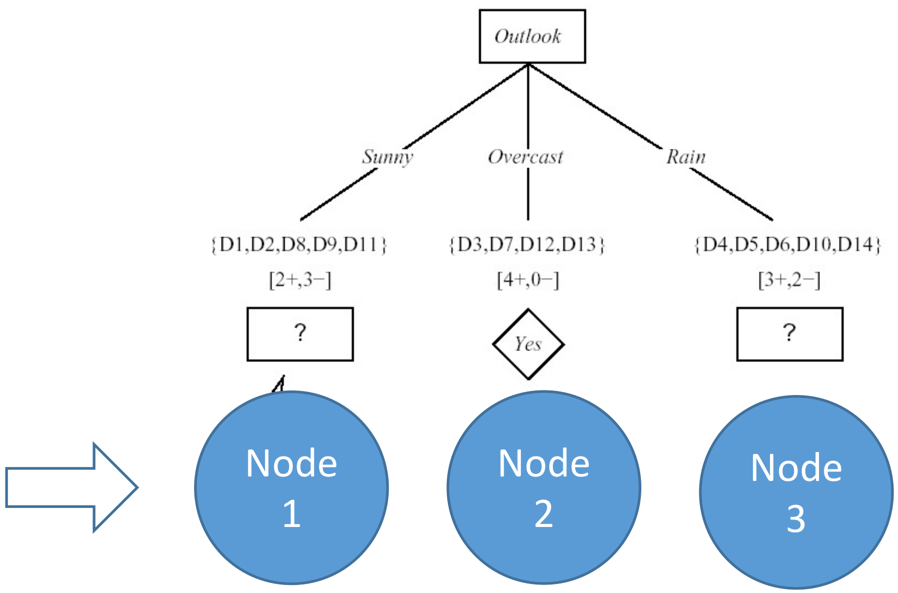
And the Next Attribute...

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

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

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

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



ID3 Algorithm (Iterative Dichotomiser 3)

- Recursively builds the decision tree based on information gain
- Note that different attributes can be selected at different nodes at the same level.
- Note that for categorical data the same attribute will not be selected twice on the same branch down the tree

ID3 pseudocode

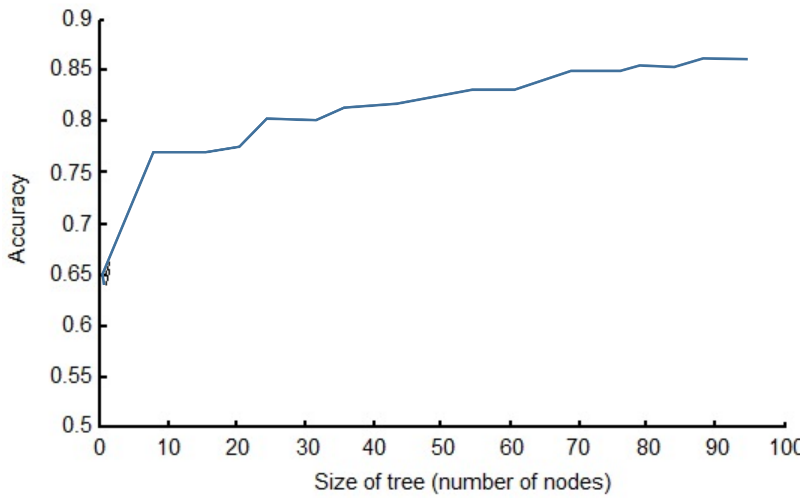
```
id3(instances, attributes)

# instances are the training examples. attributes is a list of available attributes that
may be tested by the learned decision tree.
# Returns a tree that correctly classifies all the given examples.
# targetAttribute, which is the attribute whose value is to be predicted by the tree, is a
dichotomy variable which is designated for all training instances

node = DecisionTreeRoot(instances)
# is the sample set monochromatic?
dictionary = summarizeExamples(instances.targetAttribute)
for key in dictionary:
        if dictionary[key] == total number of examples
        node.label = key
        return node
# test for number of examples to avoid overfitting and see if more attributes are available
if attributes is empty or number of examples < minimum allowed per branch:
        node.label = most common value in examples
        return node
# split the node using the best decision attribute
bestA = the attribute with the most information gain
node.decisionAtt = bestA
for each possible value v of the attribute bestA:
        subset = the subset of instances that have value v for bestA
        if subset is not empty:
                node.addBranch(id3(subset, attributes-bestA))
return node
```

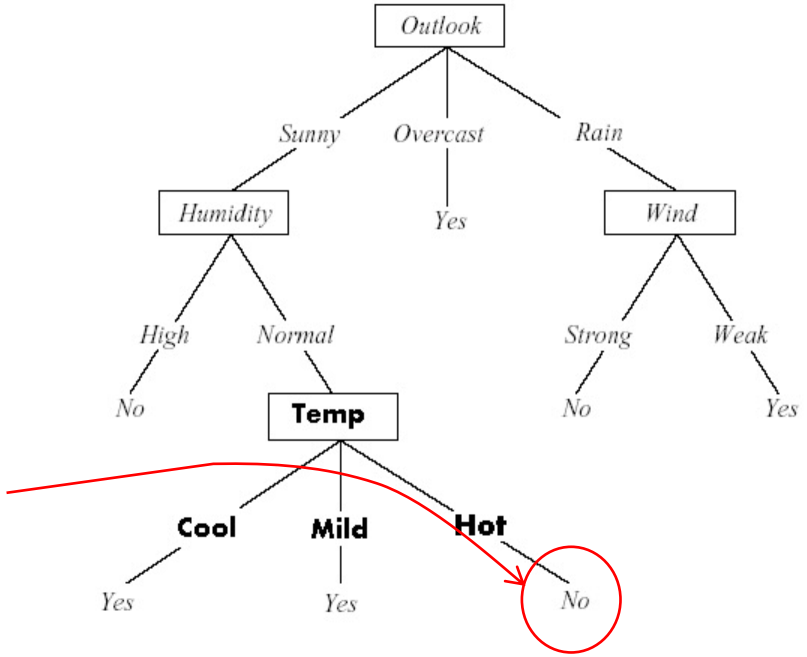
Optional

Larger Trees



- Larger trees fit the training data better!
- Is this good or bad?

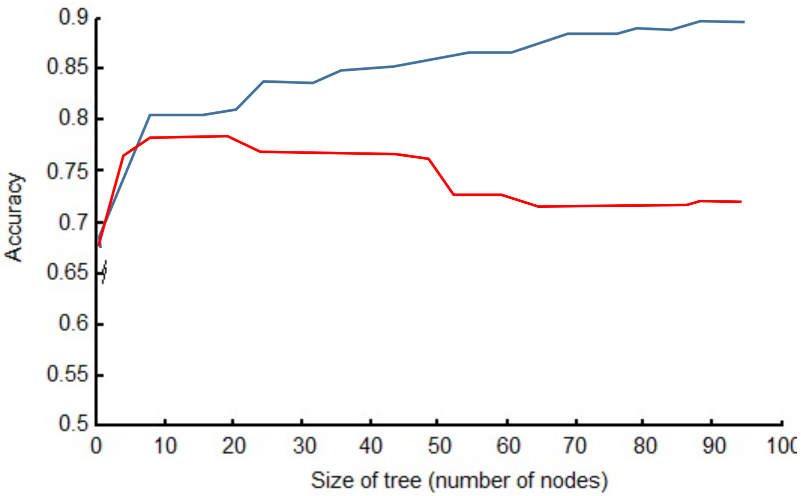
Overfitting in Decision Trees



This can be an error!
meaning the classifier adheres too much to the training set.
If these conditions (Sunny, Normal, Hot) occur again the tree will produce a wrong prediction

How do we know how to stop learning?

- Use another data set that is called the "validation" set
- Learn using the training set but measure the error on the validation set.
- What will happen?



Avoiding Overfitting – Practically …

- How can we avoid overfitting?
  1. Stop growing before full tree (before 0 error)
  2. Grow full tree, then post-prune
  3. Combine

- Stop growing?
  - Stop growing when data split does not lead to a distribution of class values which is different from that of the father in a statistically significant manner
    (either on the training set or on the validation set)
  - Stop growing when validation set accuracy starts going down
  - Other criteria
- Pruning: in recitation

Overfitting in polynomial regression

$$y = f(x) = \sin(\pi x)$$

Data generated by adding some small Gaussian noise

The figures show polynomial fits of different degrees (M)

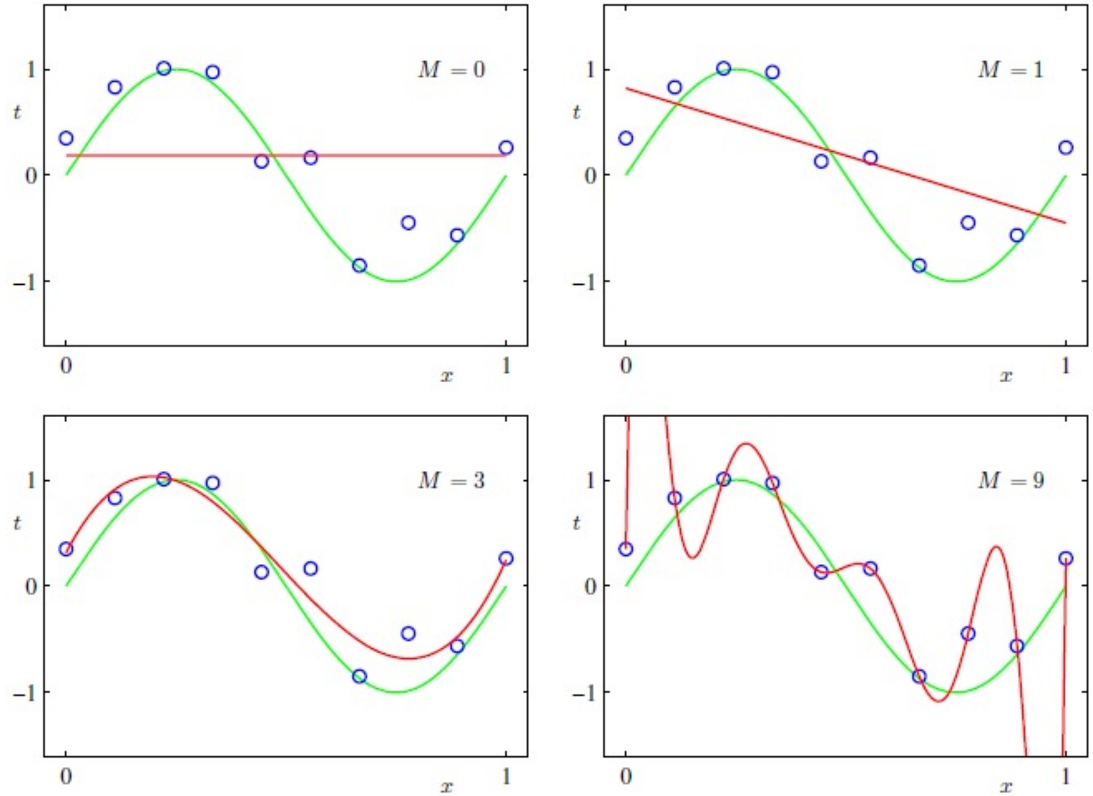


Image source: CM Bishop, Pattern Recognition and Machine Learning, Springer 2006

Overfitting in polynomial regression

$$y = f(x) = \sin(\pi x)$$

Data generated by adding some small Gaussian noise

The figures show polynomial fits of different degrees (M)

For an independent dataset the error obtained for M=9 is larger than that obtained for M=3

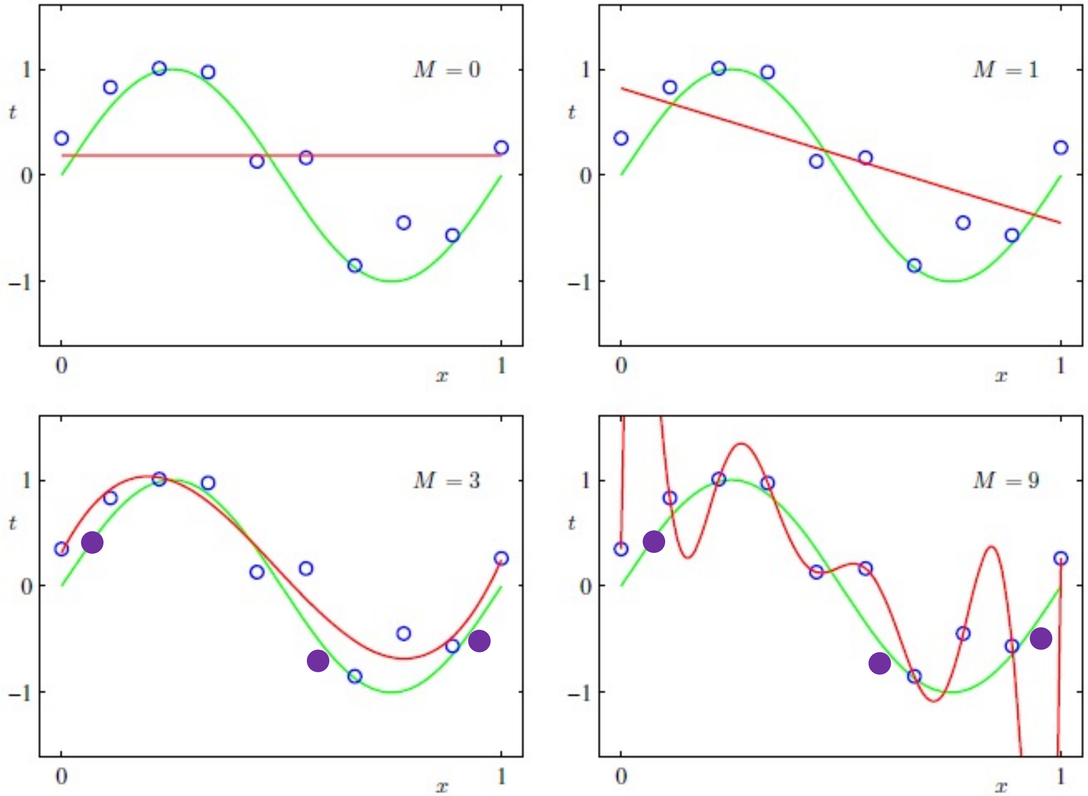
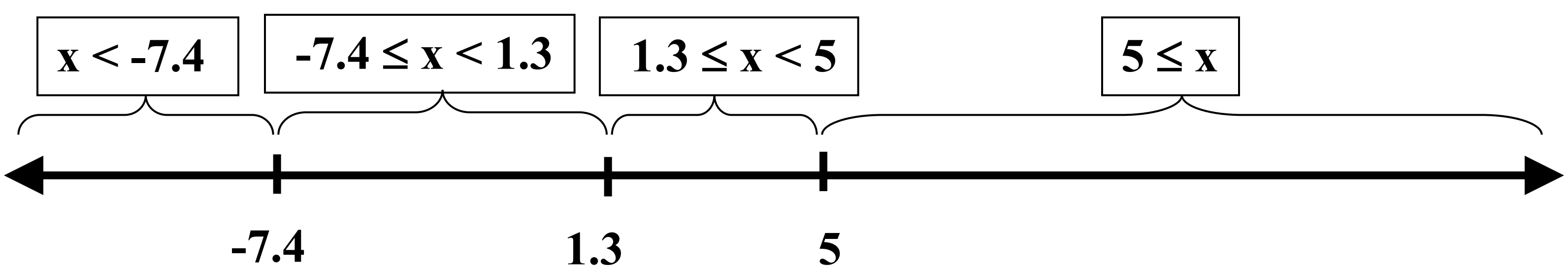


Image source: CM Bishop, Pattern Recognition and Machine Learning, Springer 2006

Discrete Vs. Continuous Attributes

- An attribute that can take on a continuous range of values can be discretized into intervals, each of which is considered one value:



Continuous Attributes

Create discrete partitions for continuous attributes

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

$S = [3+, 3-]$
$Entropy(S) = 1$

> 54

$S = [2+, 0-]$
$Entropy(S) = 0$

$S = [3+, 1-]$
$Entropy(S) = 0.8113$

$S = [3+, 3-]$
$Entropy(S) = 1$

> 85

$S = [3+, 2-]$
$Entropy(S) = 0.971$

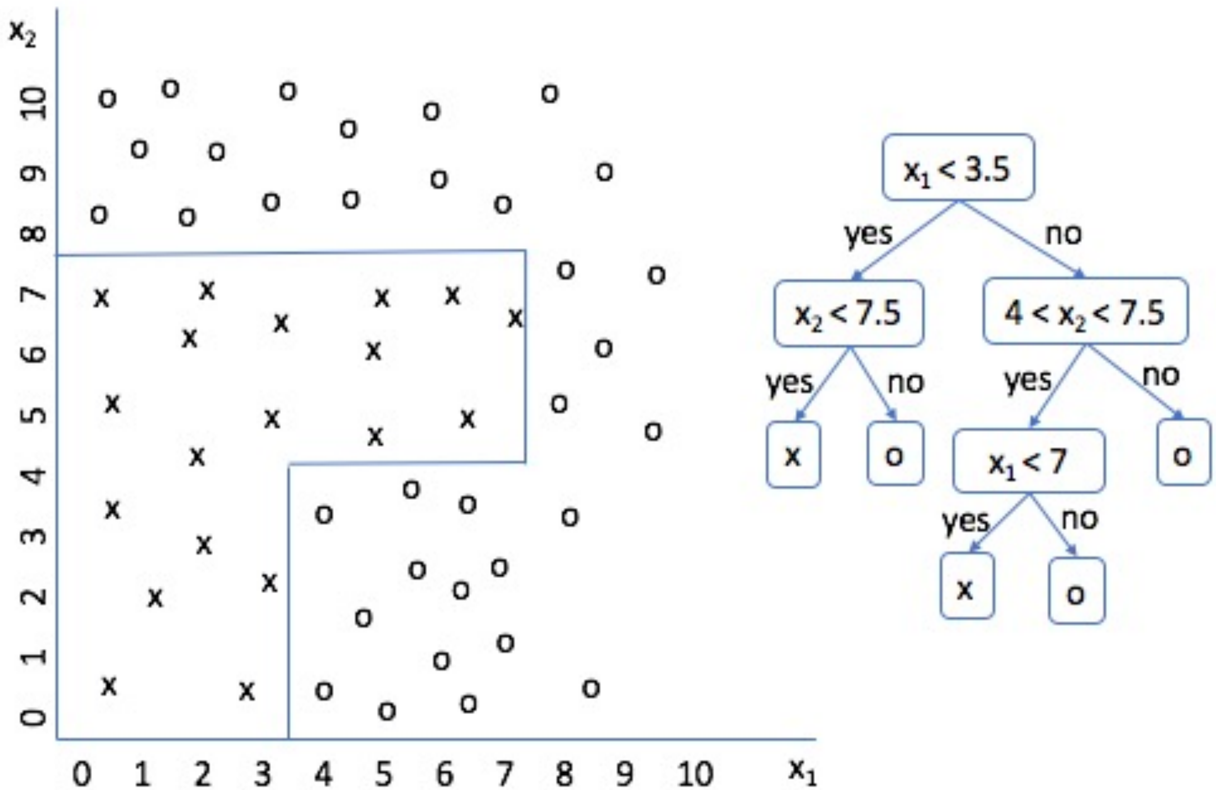
$S = [0+, 1-]$
$Entropy(S) = 0$

$$Gain(S, > 54) = 1 - \frac{2}{6}0 - \frac{4}{6}0.8113 = 0.4591$$

$$Gain(S, > 85) = 1 - \frac{5}{6}0.971 - \frac{1}{6}0 = 0.1908$$

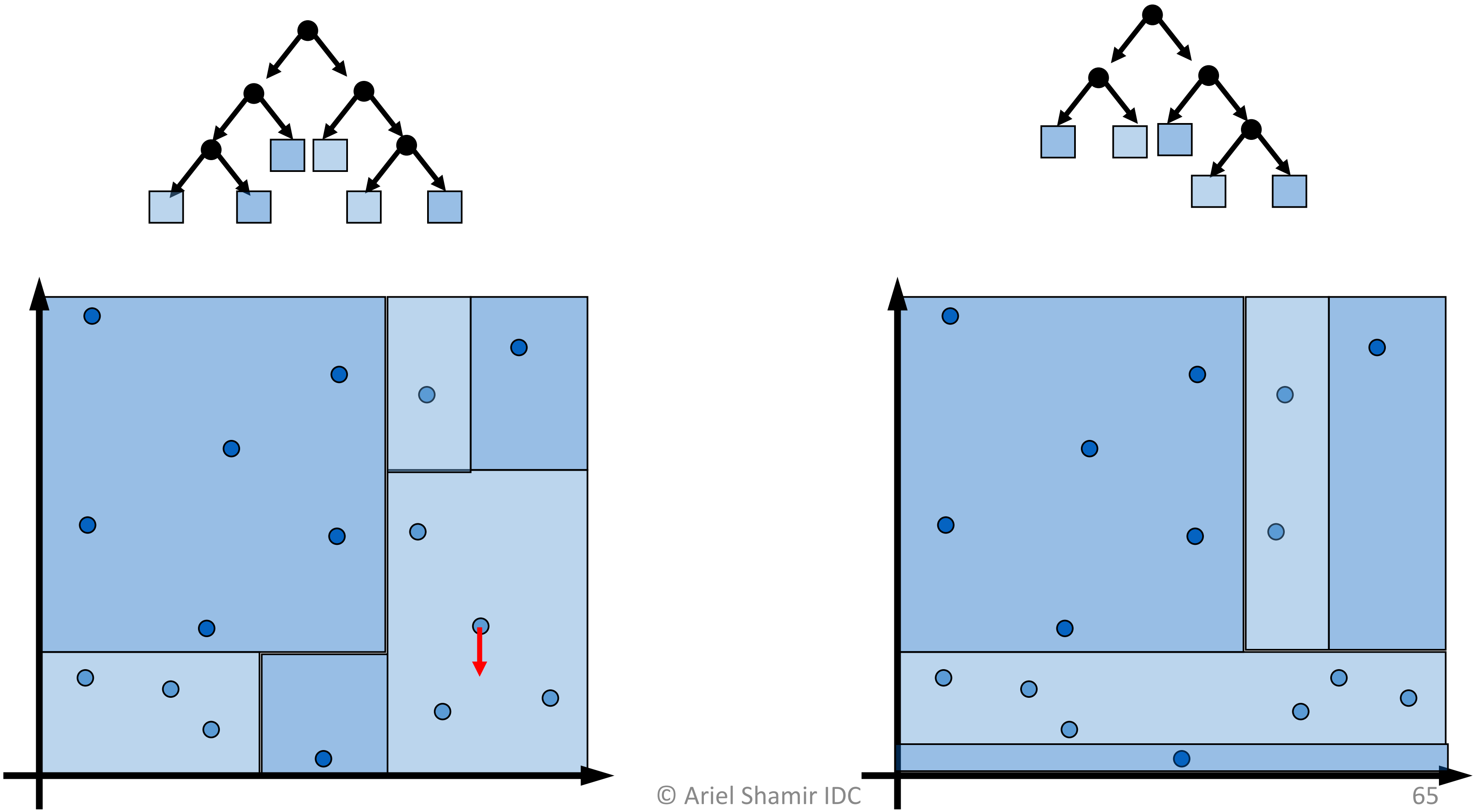
Additional computation: need to search for best splitting values among many possibilities

DTs in continuous space



Decision Tree Sensitivity

Small change in one of the values can cause large change in tree



More issues…

1. Greedy is not necessarily optimal
2. Dealing with **missing values** of some attributes
3. Split Information and **Gain Ratio** for attributes with many values
4. Including **Costs** of attaining attributes;
5. Working with a **weighted** error function (to be address in future weeks)
6. Complex **boundaries** → **Overfitting (later…)**

Imputation (replacing missing data with substituted values)

- *What if some examples in a subset miss values of some attribute A. How can we still compute Gain(S,A) and compare it to other attributes?*

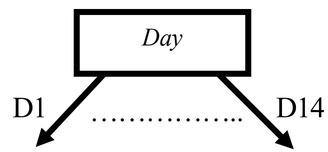
- Solution 1: Where missing, assign the most common value of *A*
  (either in the whole dataset or in this particular node)
- Solution 2: Assign probabilities to different values of A according to their distribution (in the current node? higher up in the tree?)
- Solution 3: kNN approach. To be discussed in future weeks
- More imputation approaches …

Attributes With Many Values

Problem:

- If attribute has many values, *Gain* is more likely to select it
- Imagine using the attribute DAY=[D1,…,.D14]

$$S = [9+, 5-]$$
$$Entropy(S) = 0.940$$



$$S = [0+, 1-] \qquad S = [0+, 1-]$$
$$Entropy(S) = 0 \qquad Entropy(S) = 0$$

$$Gain(S, Day) = 0.940 - \frac{1}{14}0 \ldots - \frac{1}{14}0 = 0.940$$

Split Information

- Given attribute A with c values let S_i be the subset of S which has the value i of A.

$$SplitInfo(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- This is just the entropy of A in S
  (NOT to be confused with the entropy of the label)

GainRatio

Instead of using Gain we can define and use the Gain Ratio:

$$GainRatio(S,A) \equiv \frac{Gain(S,A)}{SplitInfo(S,A)}$$

Example :

$$SplitInformation(S,Day) = -\sum_{i=1}^{14} \frac{1}{14} \log_2 \frac{1}{14} = -\log_2 \frac{1}{14} = 3.8074$$

$$GainRatio(S,Day) = \frac{0.940}{3.8074} = 0.2469$$

Attributes With Costs

- Medical diagnosis: different tests have different morbidity and/or financial costs.
- Robotics: cost (in time) of obtaining a sensory input as a feature.

- How to learn a consistent tree with low expected cost?
- Replace *Gain* by:

$$\frac{Gain^2(S,A)}{Cost(A)}$$

Or

$$\frac{2^{Gain(S,A)}-1}{(Cost(A)+1)^w} \qquad \text{where } w \in [0,1] \text{ determines importance of cost}$$

When to Consider Decision Trees?

- Instances describable by attribute(s)-value pairs
- Target function is discrete valued
  (Note there is DT regression that addresses numerical valued target attributes)
- Hypothesis with a logical structure may be required.
  Interpretation is important
- Possibly noisy (even inconsistent) training data

Summary

- Classifiers
- Linearly separable dichotomies – will be discussed in later classes
- Decision Trees
  + Construct by iteratively splitting nodes according to a measure of the quality of the split
  + Two measures of uncertainty: Gini and entropy
  + Inference algorithm – percolate through the constructed tree
  + More accessible interpretation
  + Gracefully and naturally handle categorical data
  + Can be generalized to continuous valued features
- DTs can be used for regression as well (not in scope)

Summary - cont

- DTs - additional topics:
  - \+ Greedy not opt
  - \+ Gain ratio
  - \+ Cost considerations

- Overfitting & Pruning (later)
- Random forests (not in this class's scope)