Artificial Intelligence

Traditional AI

Donghui Wang
AI Institute@ZJU
2015.4

Contents

- Decision Tree

References:

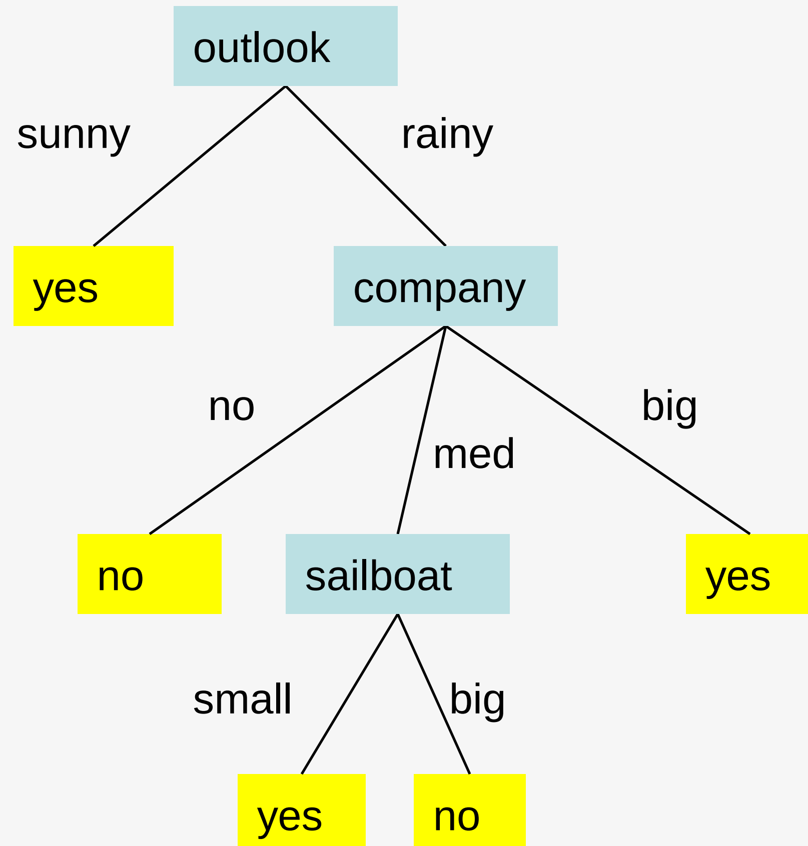
1. *Stuart J. Russell and Peter Norvig. "Artificial Intelligence: A Modern Approach", Chapter 12,18. 2011*
2. *Tom M. Michell. "Machine Learning". Chapter 3. McGraw-Hill,1997.*
3. *http://coitweb.uncc.edu/~ras/courses/Decision-Trees.ppt*

Decision Tree

(non-linear, non-parametric classifier)

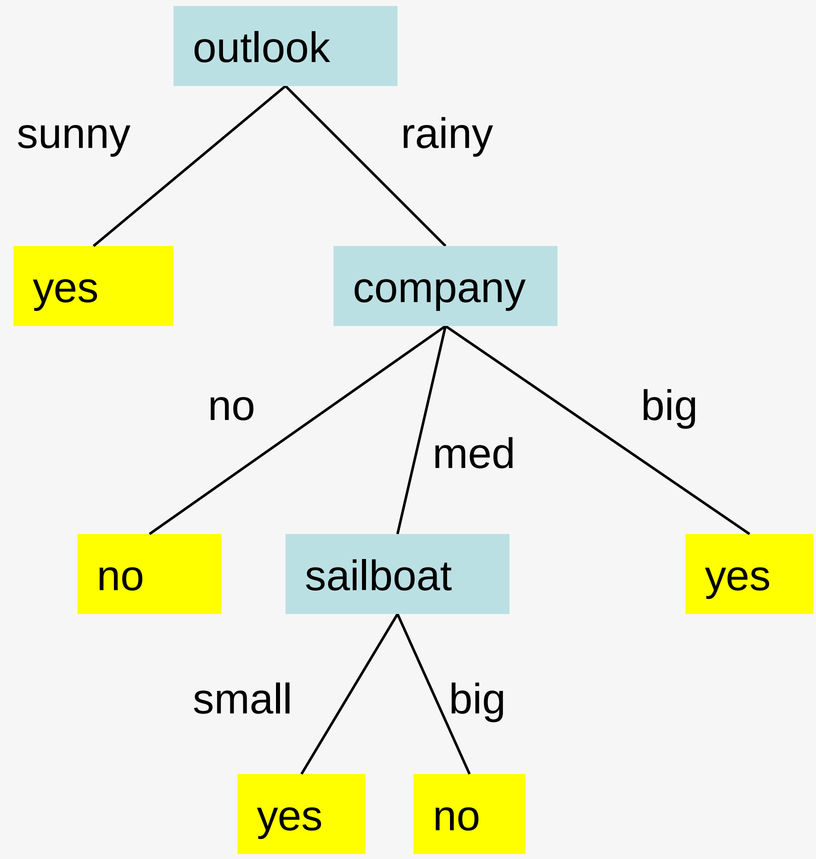
An Example Data Set and Decision Tree

#	Attribute			Class
	Outlook	Company	Sailboat	Sail?
1	sunny	big	small	yes
2	sunny	med	small	yes
3	sunny	med	big	yes
4	sunny	no	small	yes
5	sunny	big	big	yes
6	rainy	no	small	no
7	rainy	med	small	yes
8	rainy	big	big	yes
9	rainy	no	big	no
10	rainy	med	big	no



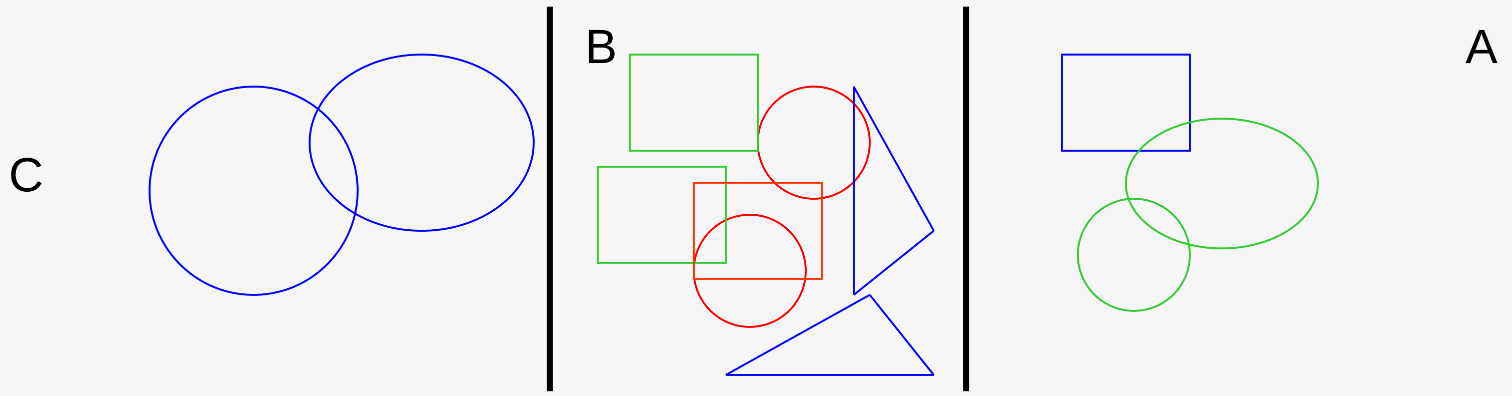
Classification

#	Attribute			Class
	Outlook	Company	Sailboat	Sail?
1	sunny	no	big	?
2	rainy	big	small	?



Decision Trees

- A hierarchical data structure that represents data by implementing a divide and conquer strategy
- Can be used as a non-parametric classification and regression method
- Given a collection of examples, learn a decision tree that represents it.
- Use this representation to classify new examples



Decision Trees: The Representation

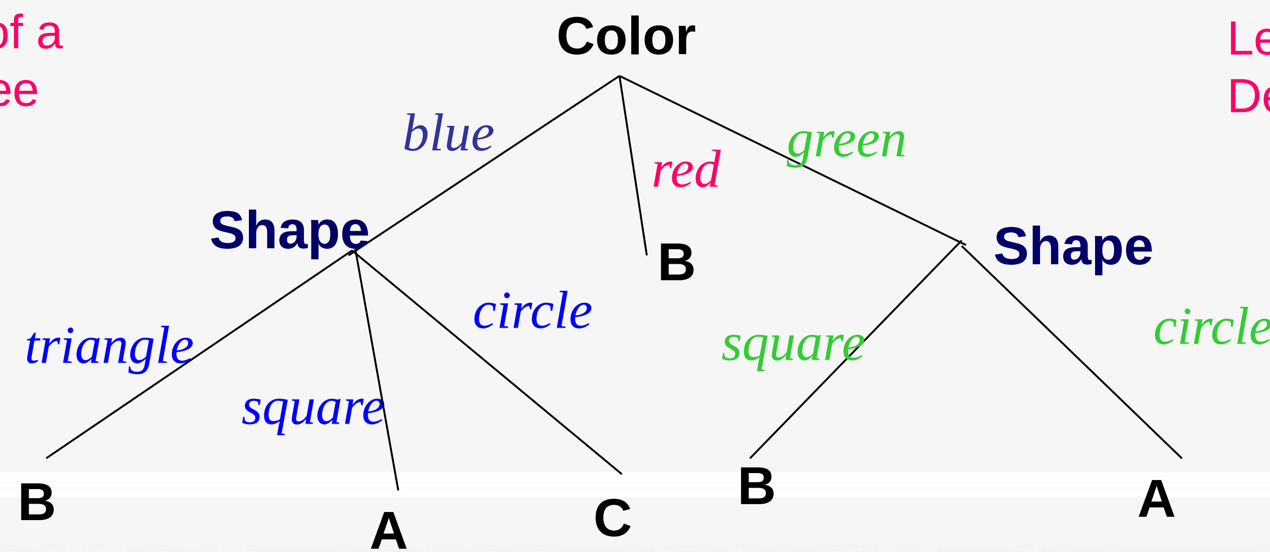
- Decision Trees are classifiers for instances represented as features vectors.

(color= ;shape= ;label=)

- Nodes are tests for feature values;
- There is one branch for each value of the feature
- Leaves specify the categories (labels)
- Can categorize instances into multiple disjoint categories

(color= red ;shape=triangle)

Evaluation of a Decision Tree



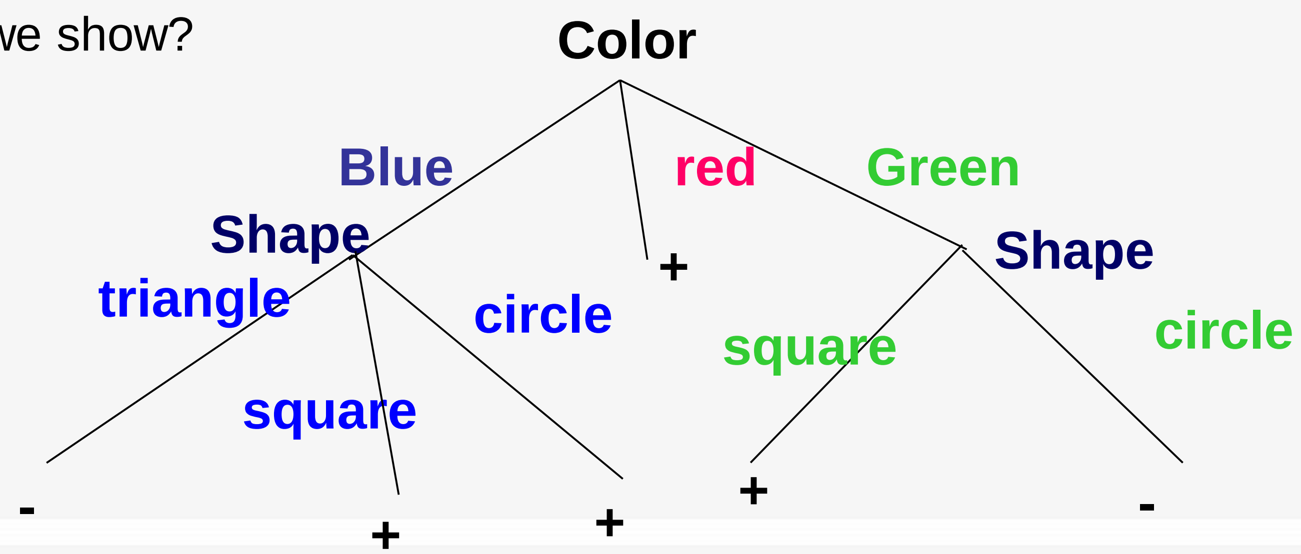
Learning a Decision Tree

Boolean Decision Trees

They can represent any Boolean function.

- Can be rewritten as rules in Disjunctive Normal Form (DNF)
- green $\wedge$ square $\rightarrow$ positive
- blue $\wedge$ circle $\rightarrow$ positive
- blue $\wedge$ square $\rightarrow$ positive
- The disjunction of these rules is equivalent to the Decision Tree

•What did we show?



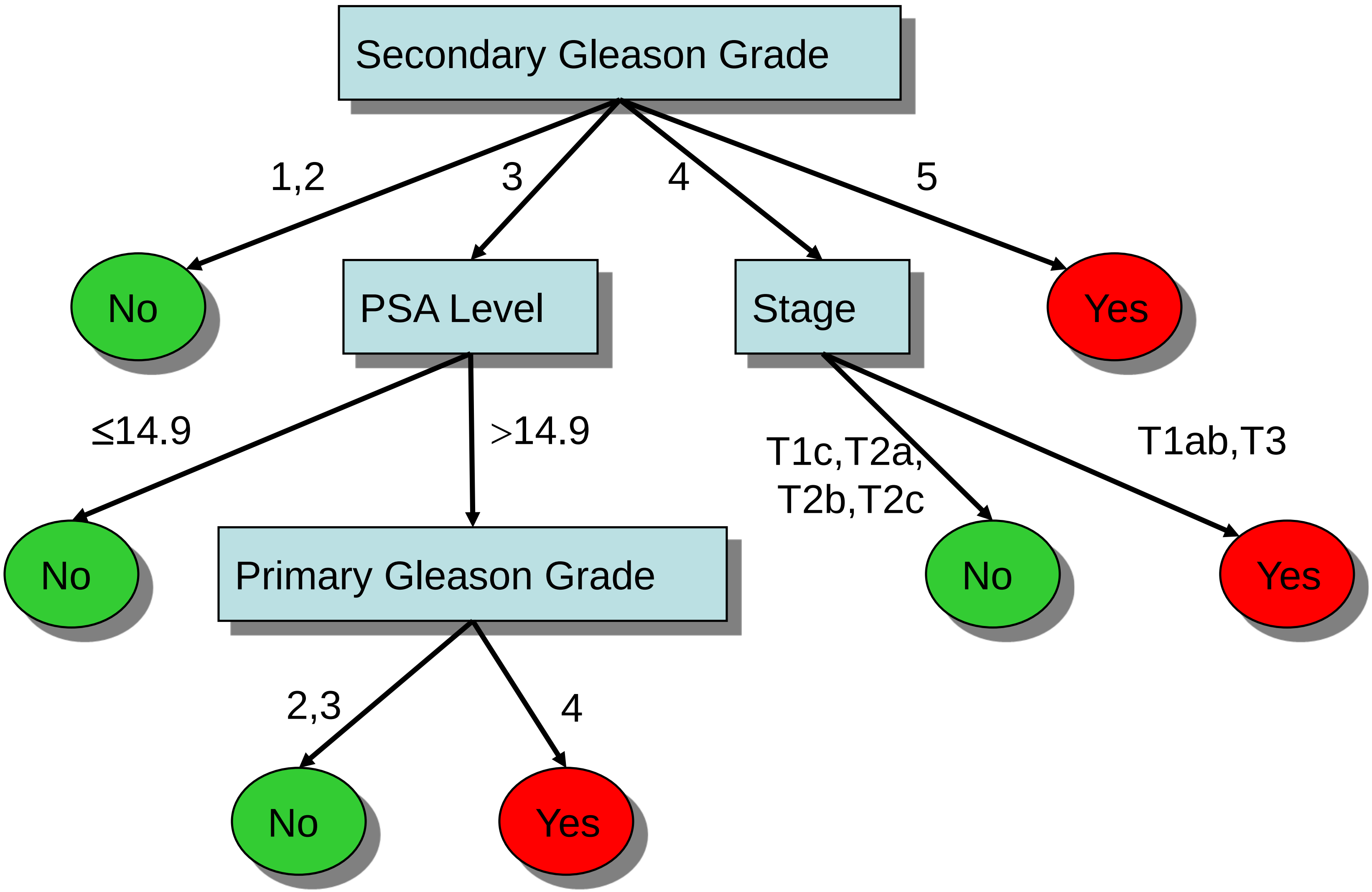
Induction of Decision Trees

- Data Set (Learning Set)
  - Each example = Attributes + Class
- Induced description = Decision tree
- TDIDT
  - *Top Down Induction of Decision Trees*
- Recursive Partitioning

Some TDIDT Systems

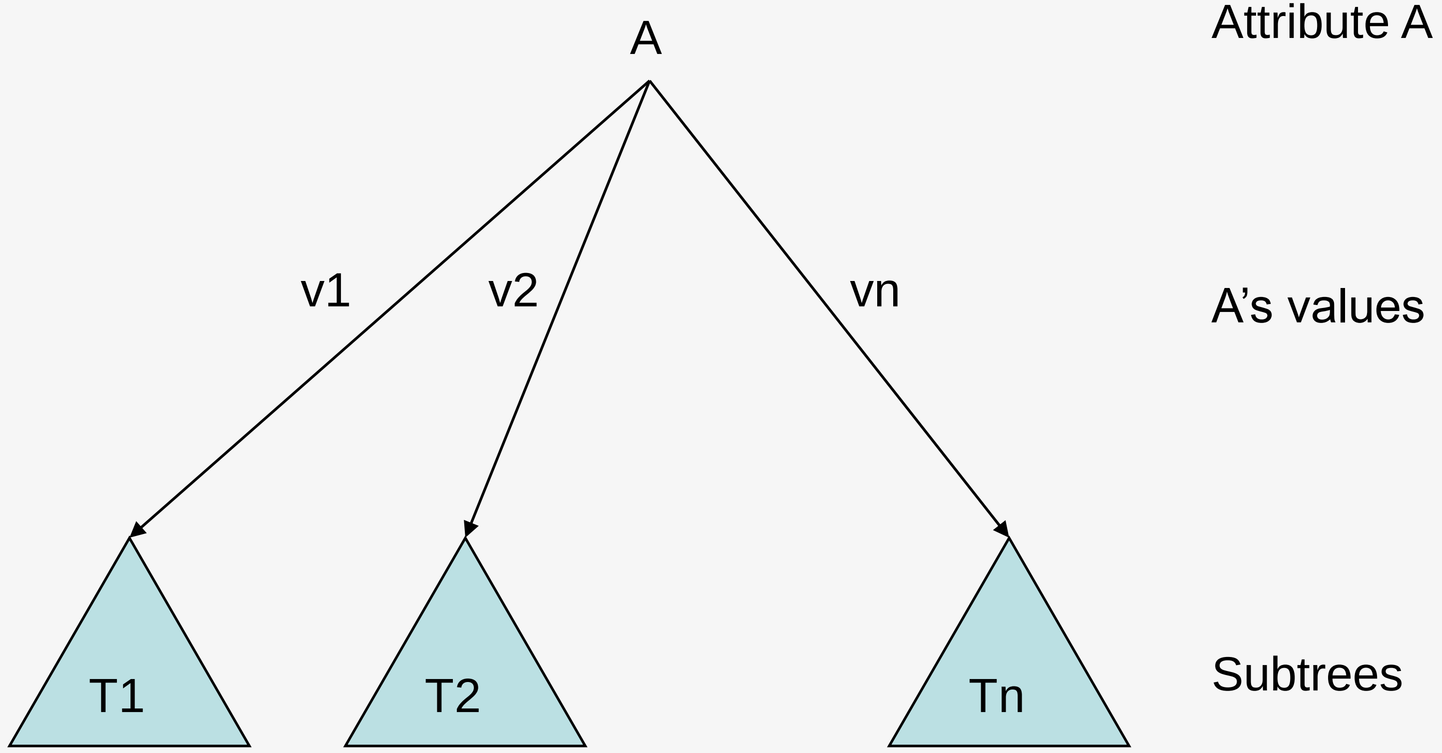
- ID3 (Quinlan 79)
  - Iterative Dichotomiser 3
- CART (Brieman et al. 84)
  - Classification & Regression Tree
- Assistant (Cestnik et al. 87)
- C4.5 (Quinlan 93)
  - Successor of ID3
- See5 (Quinlan 97)
  - C5.0
- …

Prostate cancer recurrence



TDIDT Algorithm

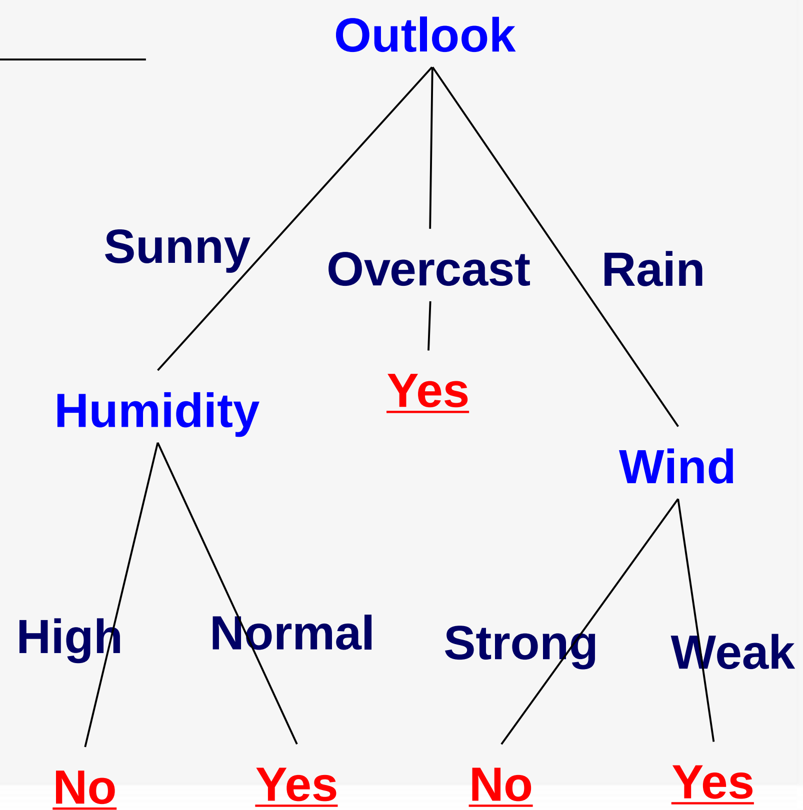
Resulting tree T is:



Basic Decision Trees Learning Algorithm

- Data is processed in Batch (i.e., all the data is available).
- Recursively build a decision tree top-down.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



Learning Algorithm

- *DT(Examples, Attributes)*

If all Examples have same label: return a leaf node with Label

Else

    If Attributes is empty: return a leaf with majority Label

    Else

        Pick an attribute A as root

        For each value v of A

            Let Examples(v) be all the examples for which A=v

            Add a branch out of the root for the test A=v

            If Examples(v) is empty

                create a leaf node labeled with the majority label in Examples

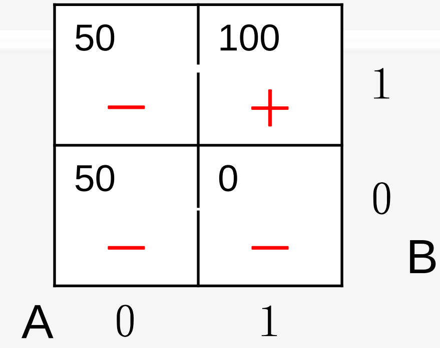
            Else recursively create subtree by calling

                DT(Examples(v), Attribute-{A})

Picking the Root Attribute

- The goal is to have the resulting decision tree as small as possible (Occam's Razor)
- Finding the minimal decision tree consistent with the data is NP-hard
- The recursive algorithm is a greedy heuristic search for a simple tree, but cannot guarantee optimality.
- The main decision in the algorithm is the selection of the next attribute to condition on.

Picking the Root Attribute

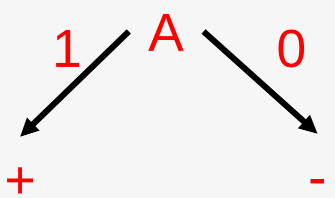


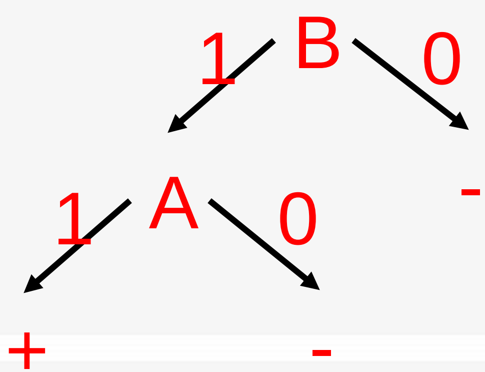
- Consider data with two Boolean attributes (A,B).

  < (A=0,B=0), - >: 50 examples
  < (A=0,B=1), - >: 50 examples
  < (A=1,B=0), - >: 0 examples
  < (A=1,B=1), + >: 100 examples

- What should be the first attribute we select?

- Splitting on A: we get purely labeled nodes.





- Splitting on B: we don't get purely labeled nodes.

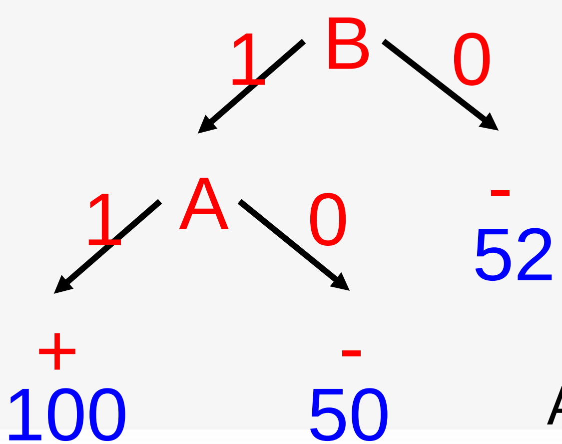
- What if we have: <(A=1,B=0), - >: 2 examples

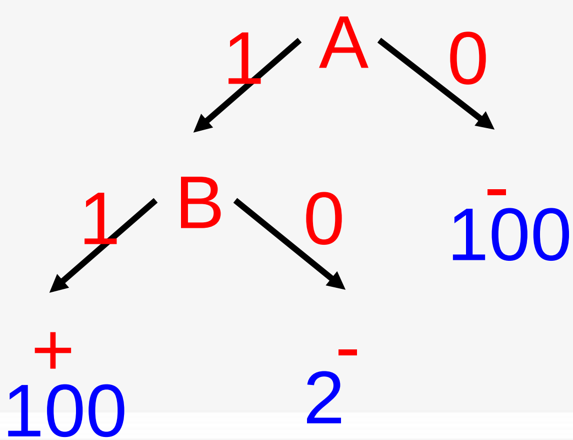
Picking the Root Attribute

- Consider data with two Boolean attributes (A,B).

< (A=0,B=0), - >: 50 examples
< (A=0,B=1), - >: 50 examples
< (A=1,B=0), - >: ~~0 examples~~ 2 examples
< (A=1,B=1), + >: 100 examples

- Trees looks structurally similar; which attribute should we choose?





Advantage A. But…
Need a way to quantify things

Picking the Root Attribute

- *The goal is to have the resulting decision tree as small as possible (Occam's Razor)*

- *The main decision in the algorithm is the selection of the next attribute to condition on.*

- We want attributes that split the examples to sets that are relatively pure in one label; this way we are closer to a leaf node.

- The most popular heuristics is based on information gain, originated with the ID3 system of Quinlan.

Entropy

- Entropy (impurity, disorder) of a set of examples, S, relative to a binary classification is:

$$Entropy(S) = -p_+ \log(p_+) - p_- \log(p_-)$$

  - Where p_+ is the proportion of positive examples in S and is p_- the proportion of negatives.
  - If all the examples belong to the same category: *Entropy* = 0
  - If the examples are equally mixed (0.5,0.5) *Entropy* = 1

•In general, when p_i is the fraction of examples labeled i:

$$\boxed{Entropy(\{p_1, p_2, \ldots, p_k\}) = -\sum_{i=1}^{k} p_i \log(p_i)}$$

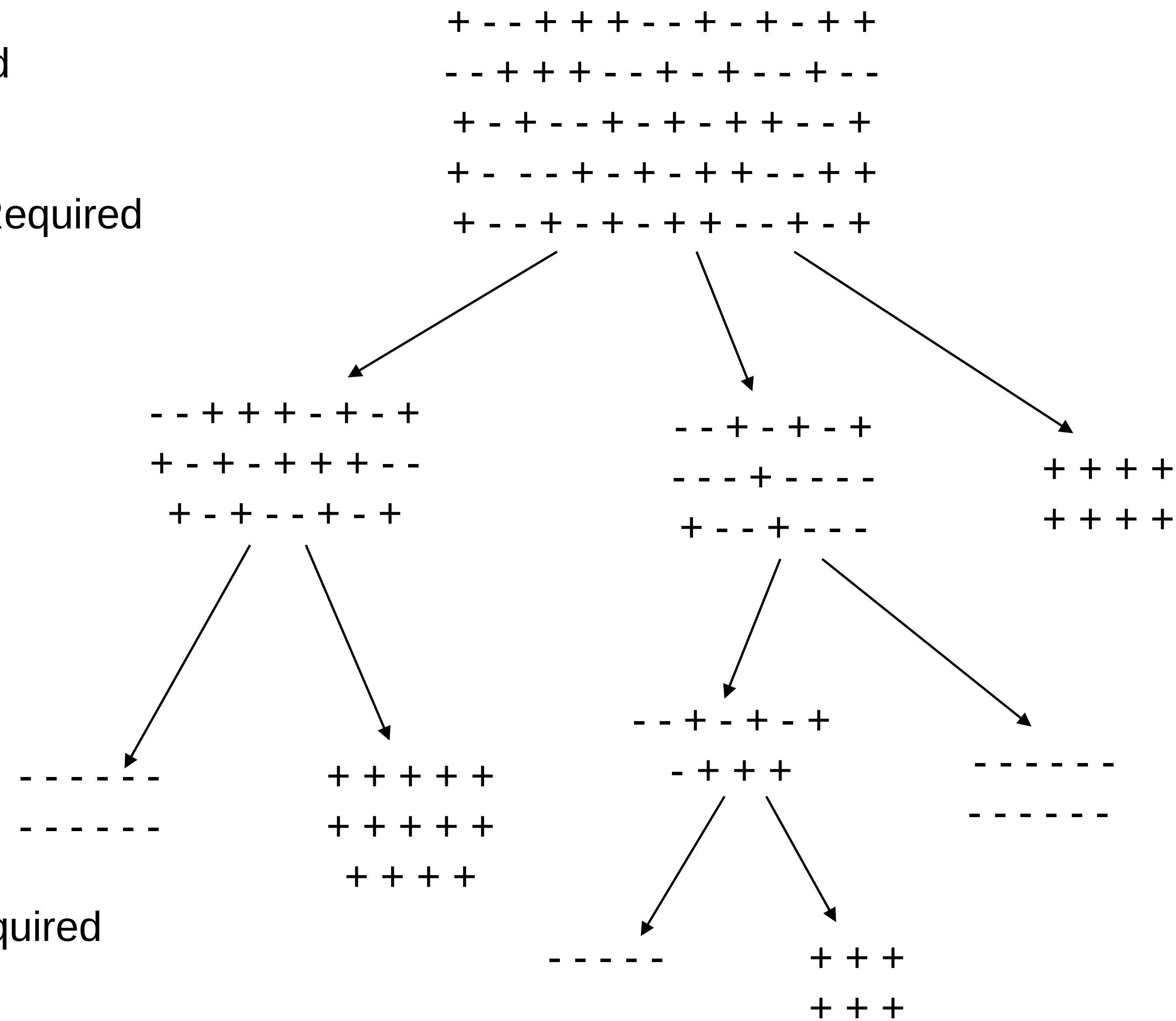
Entropy can be viewed as the number of bits required, on average, to encode the class of labels. If the probability for + is 0.5, a single bit is required for each example; if it is 0.8 -- can use less then 1 bit.

$$Entropy(\{p_1, p_2, \ldots, p_k\}) = -\sum_{i=1}^{k} p_i \log(p_i)$$

Highly Disorganized

High Entropy

Much Information Required

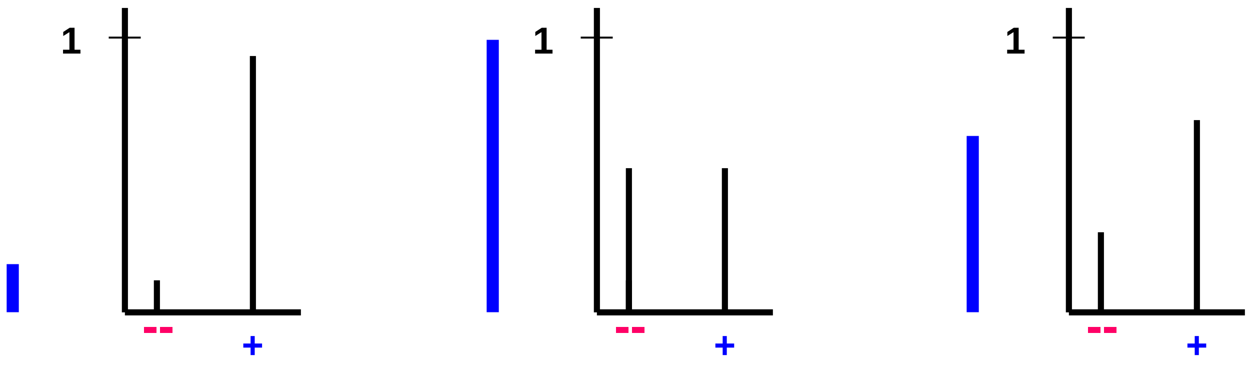


Highly Organized

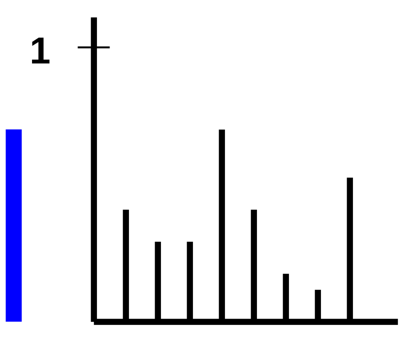
Low Entropy

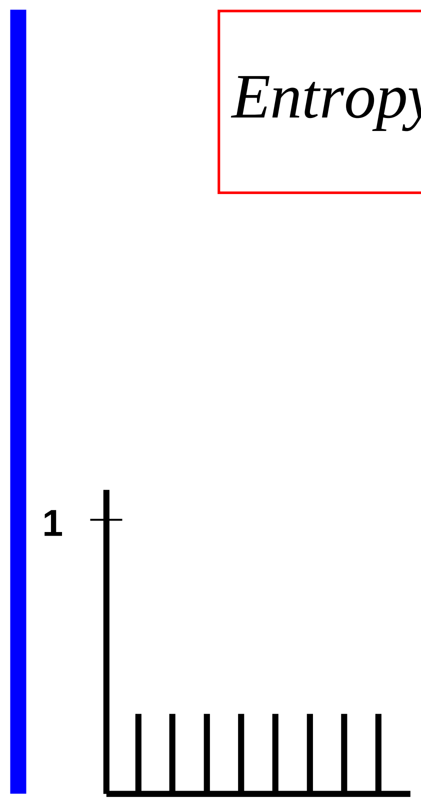
Little Information Required

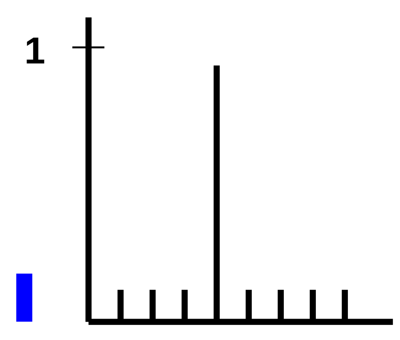
$$Entropy(S) = -p_+ \log(p_+) - p_- \log(p_-)$$



$$\boxed{Entropy(\{p_1, p_2, \ldots, p_k\}) = -\sum_{i=1}^{k} p_i \log(p_i)}$$



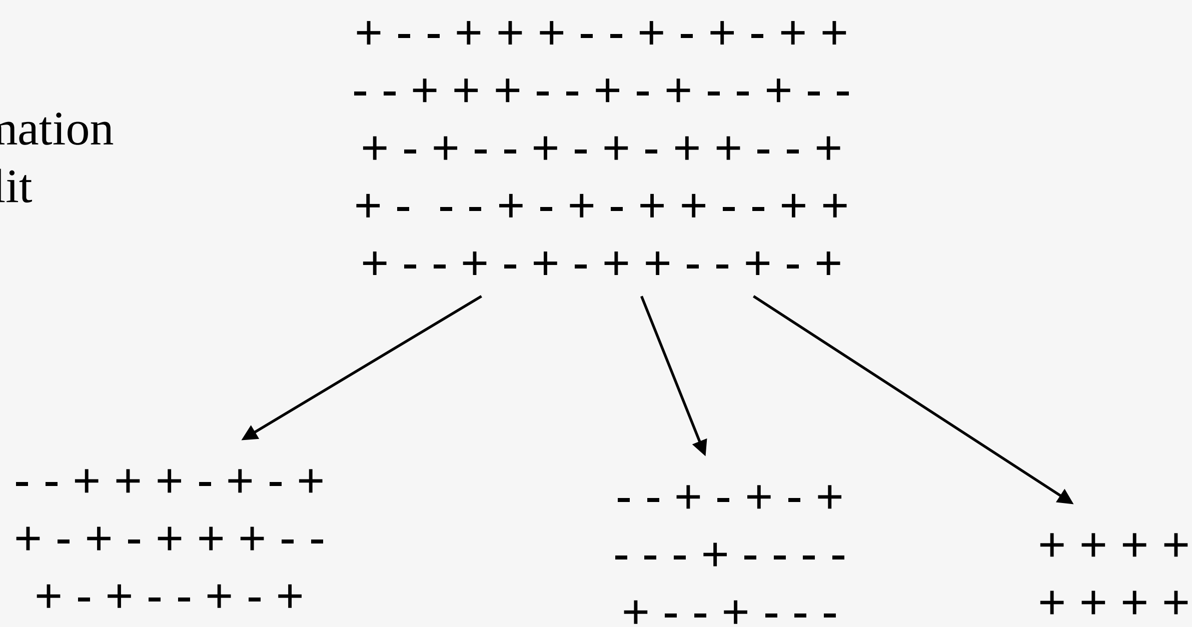




Information Gain

- For Information Gain, Subtract Information required after split from before

Some Expected Information required before the split



Some Expected Information required after the split

Information Gain

- The information gain of an attribute *a* is the expected reduction in entropy caused by partitioning on this attribute.

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

Where S_v is the subset of S for which attribute *a* has value *v*

and the entropy of partitioning the data is calculated by weighing the entropy of each partition by its size relative to the original set

Partitions of low entropy lead to high gain

An Illustrative Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

An Illustrative Example(2)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

$$\text{Entropy}(S) = -\tfrac{9}{14}\log\left(\tfrac{9}{14}\right) - \tfrac{5}{14}\log\left(\tfrac{5}{14}\right) = 0.94$$

Entropy

9+,5-

An Illustrative Example(2)

Humidity	Wind	PlayTennis
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Entropy

9+,5-

E=.94

An Illustrative Example(2)

Humidity

High | Normal

3+,4- | 6+,1-

E=.985 | *E*=.592

Humidity	Wind	PlayTennis
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Entropy

9+,5-

E=.94

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

An Illustrative Example(2)

Humidity	Wind	PlayTennis
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Humidity

High: 3+,4-, E=.985

Normal: 6+,1-, E=.592

Wind

Weak: 6+2-, E=.811

Strong: 3+,3-, E=1.0

Entropy

9+,5-

E=.94

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

An Illustrative Example(2)

Humidity	Wind	PlayTennis
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Entropy

9+,5-

E=.94

Humidity

High	Normal
3+,4-	6+,1-
E=.985	*E*=.592

Wind

Weak	Strong
6+2-	3+,3-
E=.811	*E*=1.0

*Gain(S,*Humidity*)*=
.94 - 7/14 0.985
- 7/14 0.592=
0.151

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

An Illustrative Example(2)

Humidity	Wind	PlayTennis
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Entropy

9+,5-

E=.94

Humidity

High	Normal
3+,4-	6+,1-
E=.985	*E*=.592

Wind

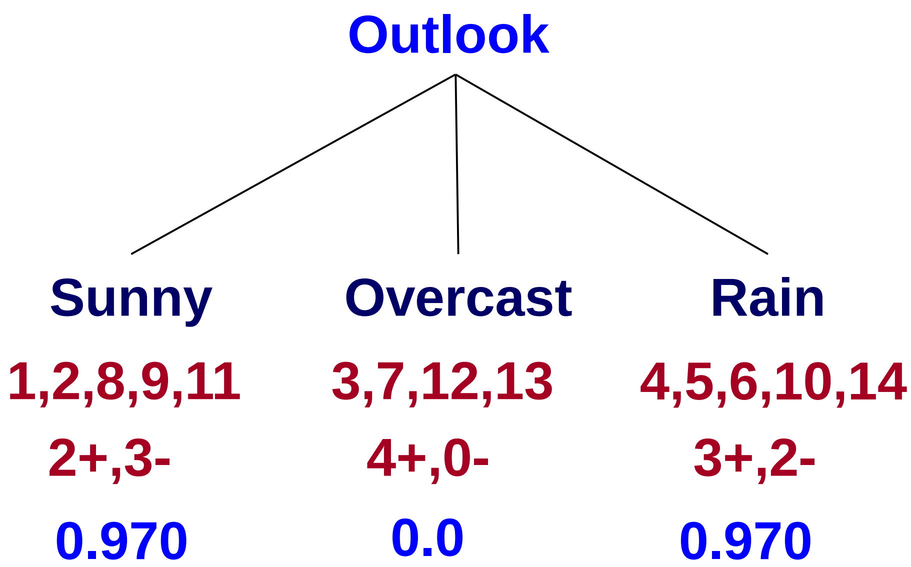
Weak	Strong
6+2-	3+,3-
E=.811	*E*=1.0

*Gain(S,*Humidity*)*=
.94 - 7/14 0.985
- 7/14 0.592=
0.151

*Gain(S,*Wind*)*=
.94 - 8/14 0.811
- 6/14 1.0 =
0.048

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

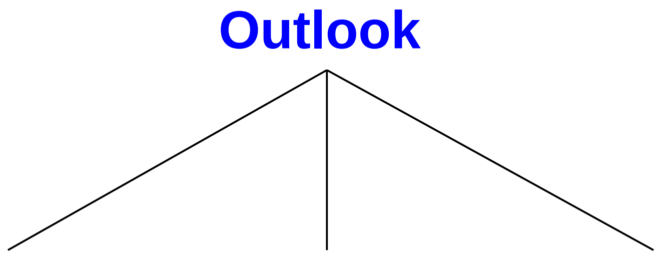
An Illustrative Example(3)



*Gain(S,*Outlook*)*=
0.246

Day	Outlook	PlayTennis
1	Sunny	No
2	Sunny	No
3	Overcast	Yes
4	Rain	Yes
5	Rain	Yes
6	Rain	No
7	Overcast	Yes
8	Sunny	No
9	Sunny	Yes
10	Rain	Yes
11	Sunny	Yes
12	Overcast	Yes
13	Overcast	Yes
14	Rain	No

An Illustrative Example(3)



*Gain(S,*Humidity*)*=0.151

*Gain(S,*Wind*)*=0.048

*Gain(S,*Temperature*)*=0.029

*Gain(S,*Outlook*)*=0.246

An Illustrative Example(3)

Outlook

Sunny	Overcast	Rain
1,2,8,9,11	3,7,12,13	4,5,6,10,14
2+,3-	4+,0-	3+,2-
?	Yes	?

Day	Outlook	PlayTennis
1	Sunny	No
2	Sunny	No
3	Overcast	Yes
4	Rain	Yes
5	Rain	Yes
6	Rain	No
7	Overcast	Yes
8	Sunny	No
9	Sunny	Yes
10	Rain	Yes
11	Sunny	Yes
12	Overcast	Yes
13	Overcast	Yes
14	Rain	No

Continue until:

- **Every attribute is included in path, or,**
- **All examples in the leaf have same label**

An Illustrative Example(4)

$Gain(S_{sunny}, Humidity) = .97-(3/5)\ 0-(2/5)\ 0 = .97$

$Gain(S_{sunny}, Temp) = .97-\ 0-(2/5)\ 1 = .57$

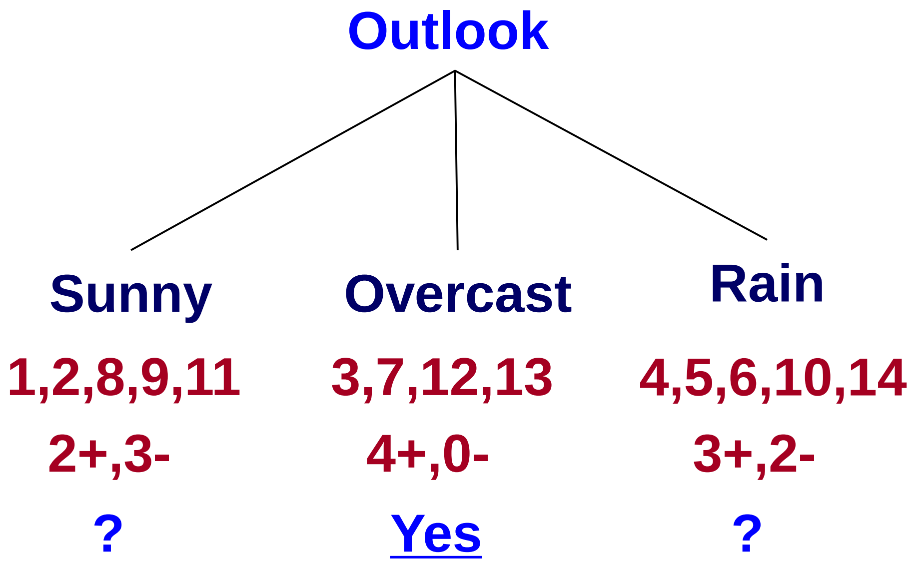
$Gain(S_{sunny}, Wind) = .97-(2/5)\ 1 - (3/5)\ .92 = .02$

Outlook

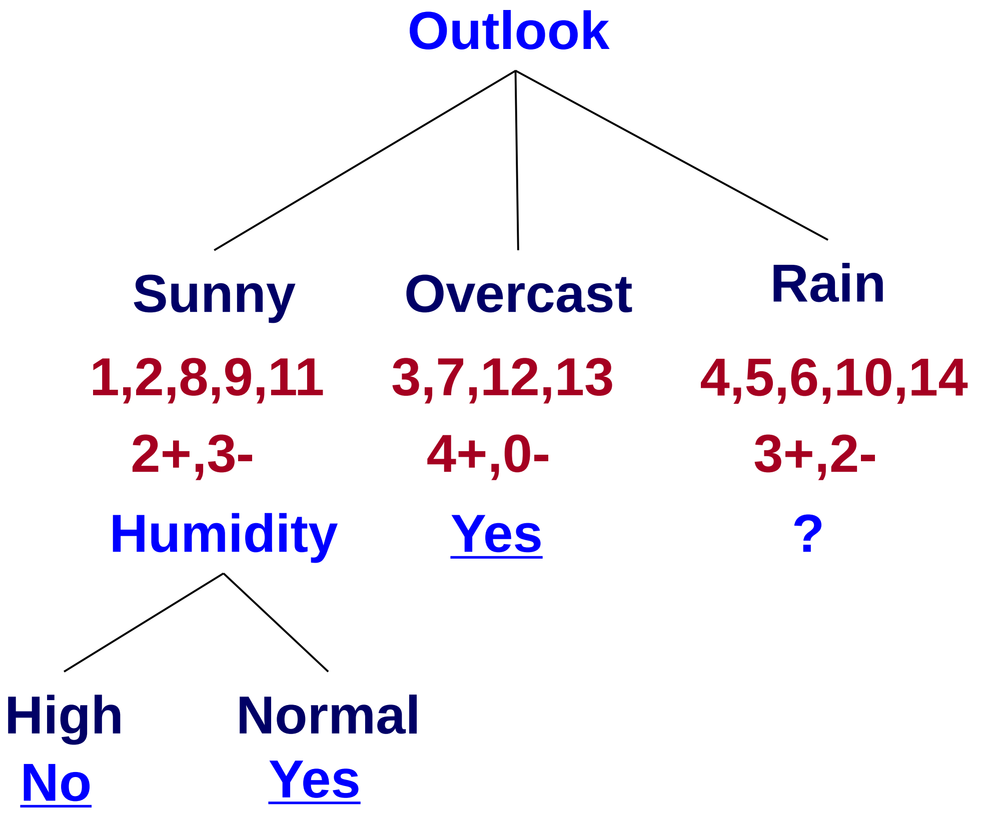
Sunny	Overcast	Rain
1,2,8,9,11	3,7,12,13	4,5,6,10,14
2+,3-	4+,0-	3+,2-
?	Yes	?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

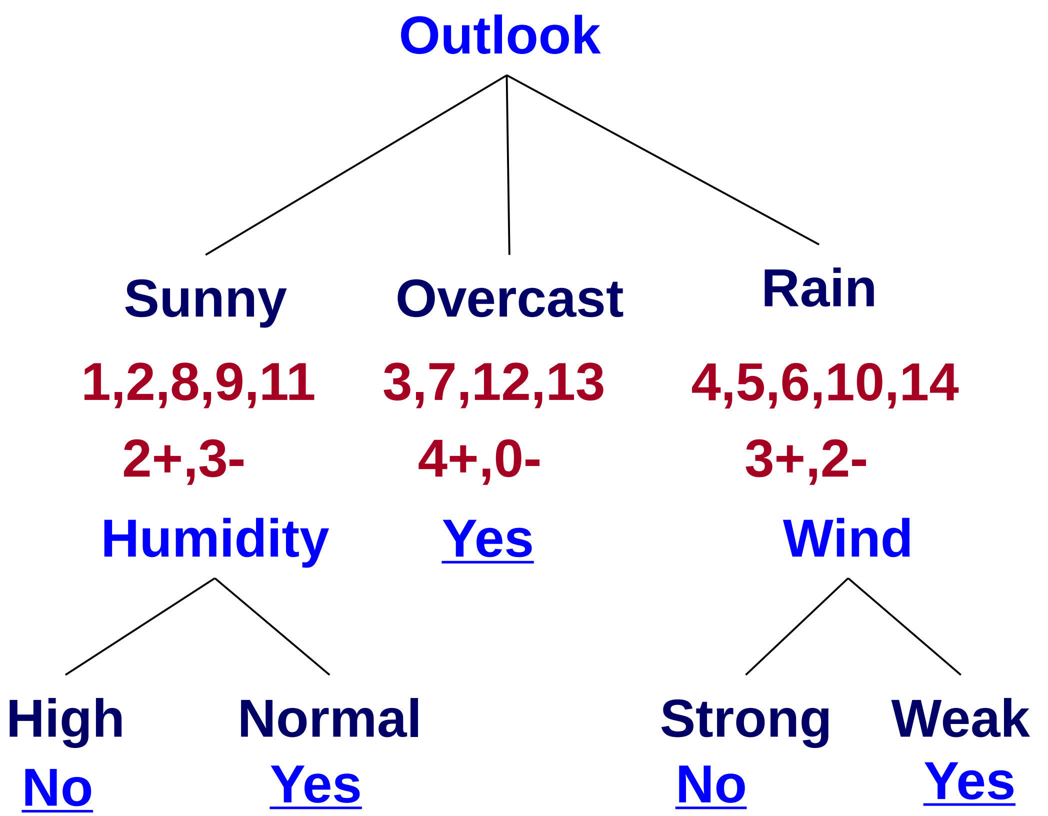
An Illustrative Example(5)



An Illustrative Example(5)



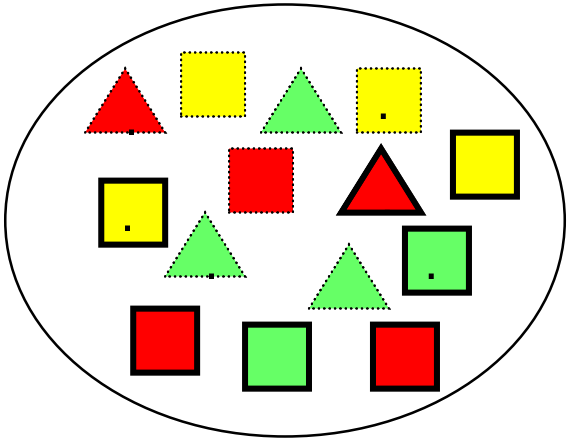
An Illustrative Example(5)



Another Example: Triangles & Sqaures

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange

Data Set:
A set of classified objects



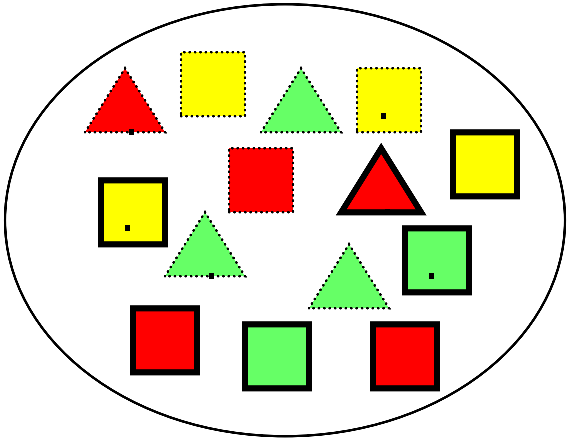
Another Example: Triangles & Sqaures

- 5 triangles
- 9 squares
- class probabilities

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

Data Set:
A set of classified objects

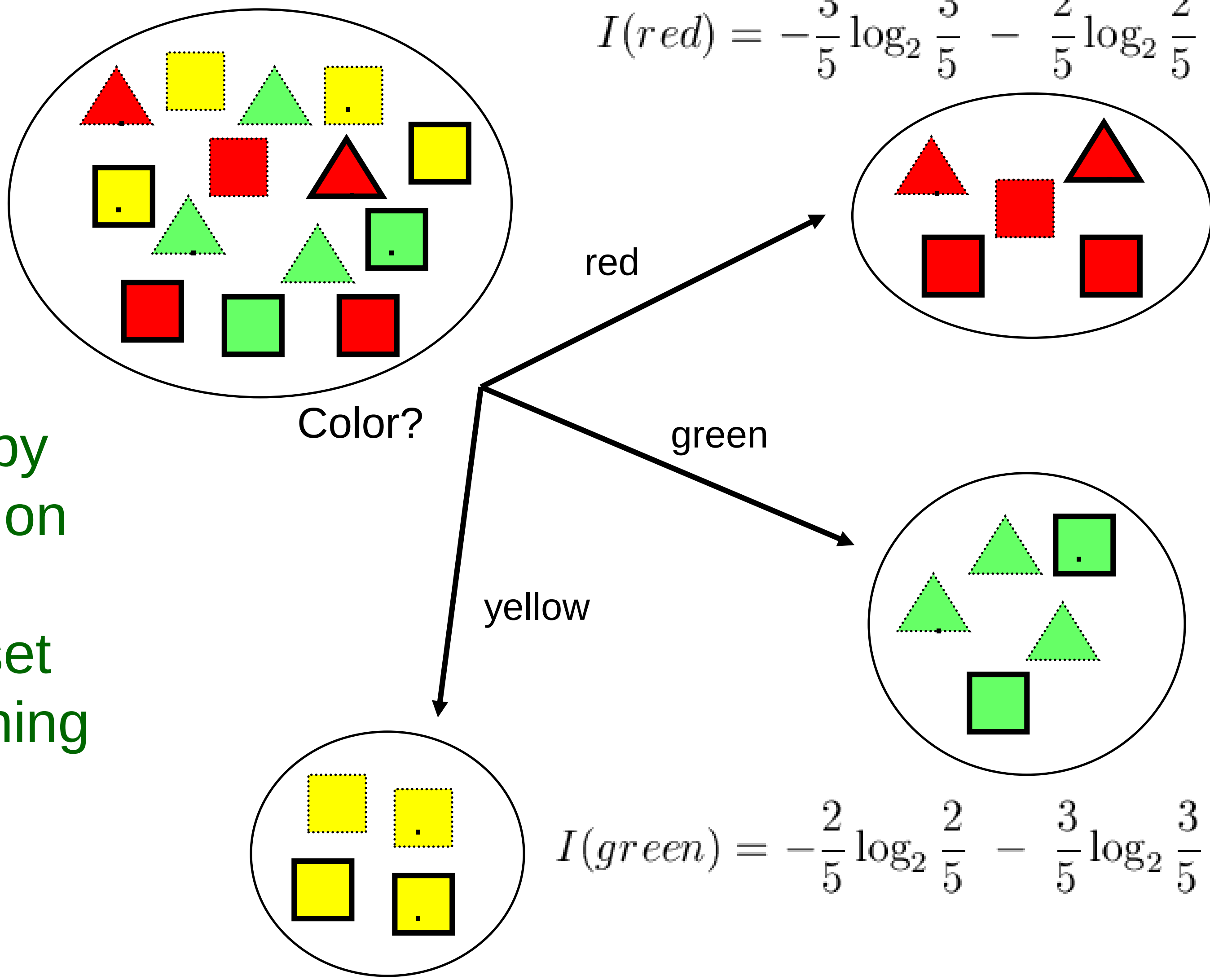


- entropy

$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits}$$

Another Example: Triangles & Sqaures

$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$

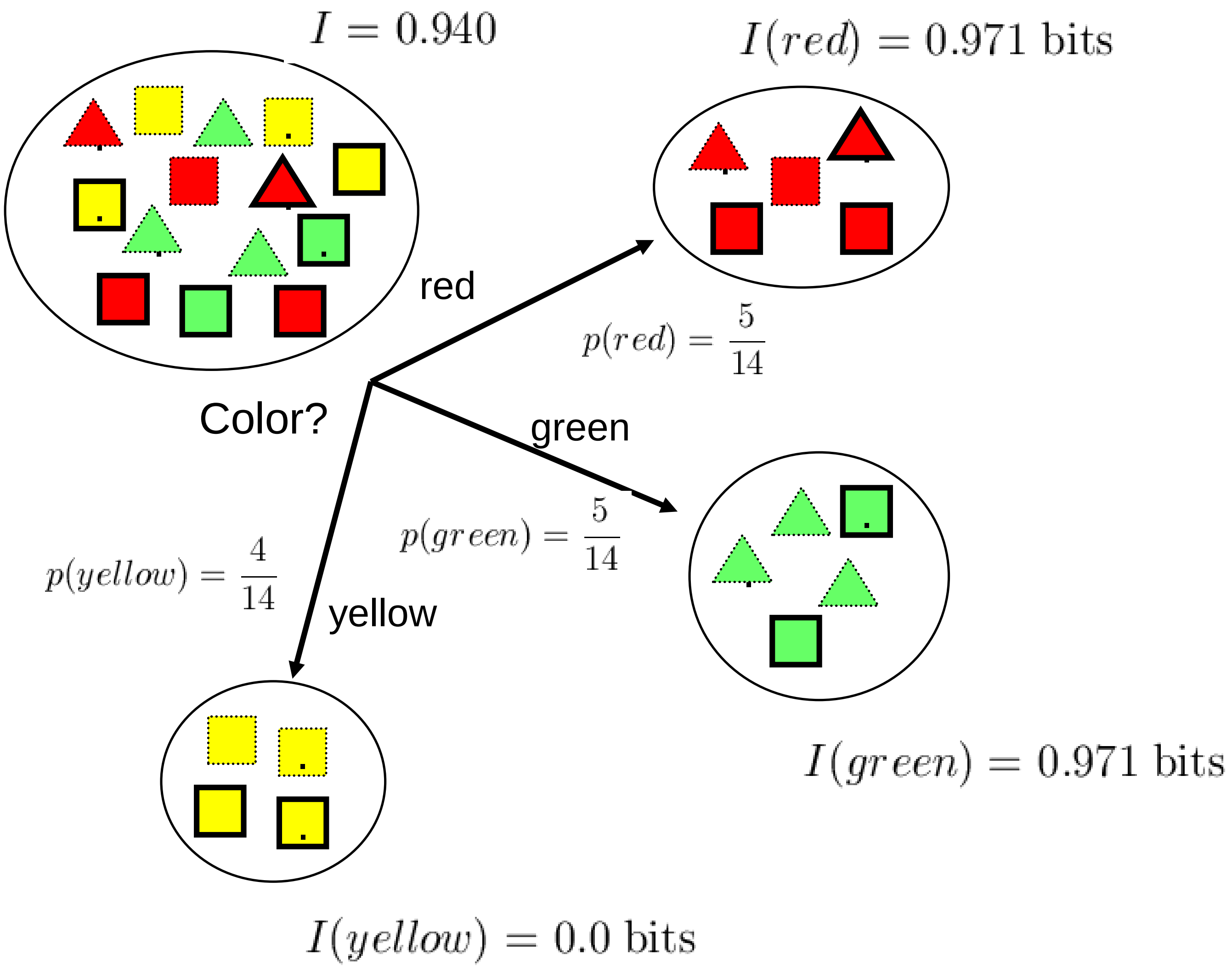


Entropy
reduction
by
data set
partitioning

$$I(green) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \text{ bits}$$

$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$

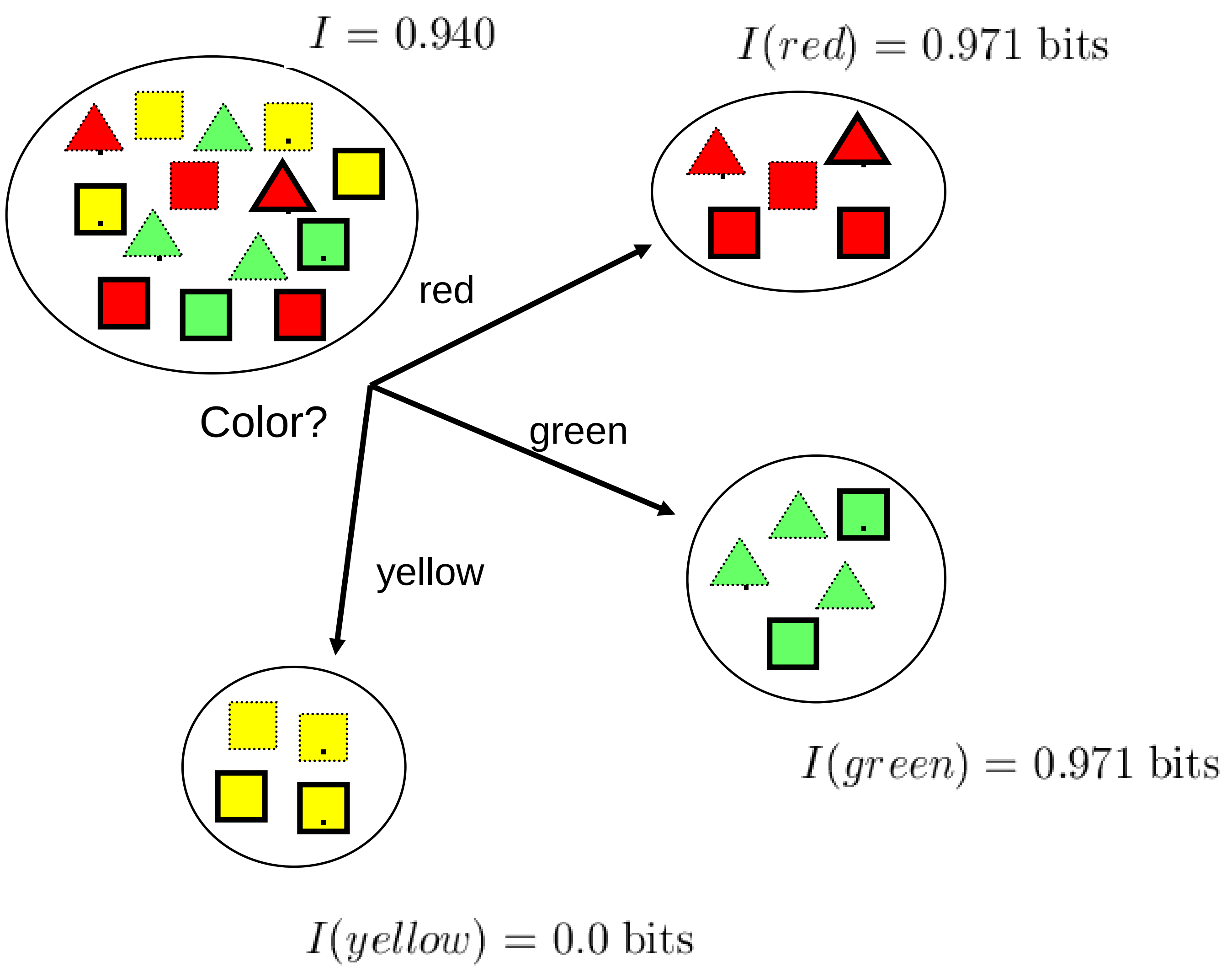
Another Example: Triangles & Sqaures



$$I_{res}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694\ bits$$

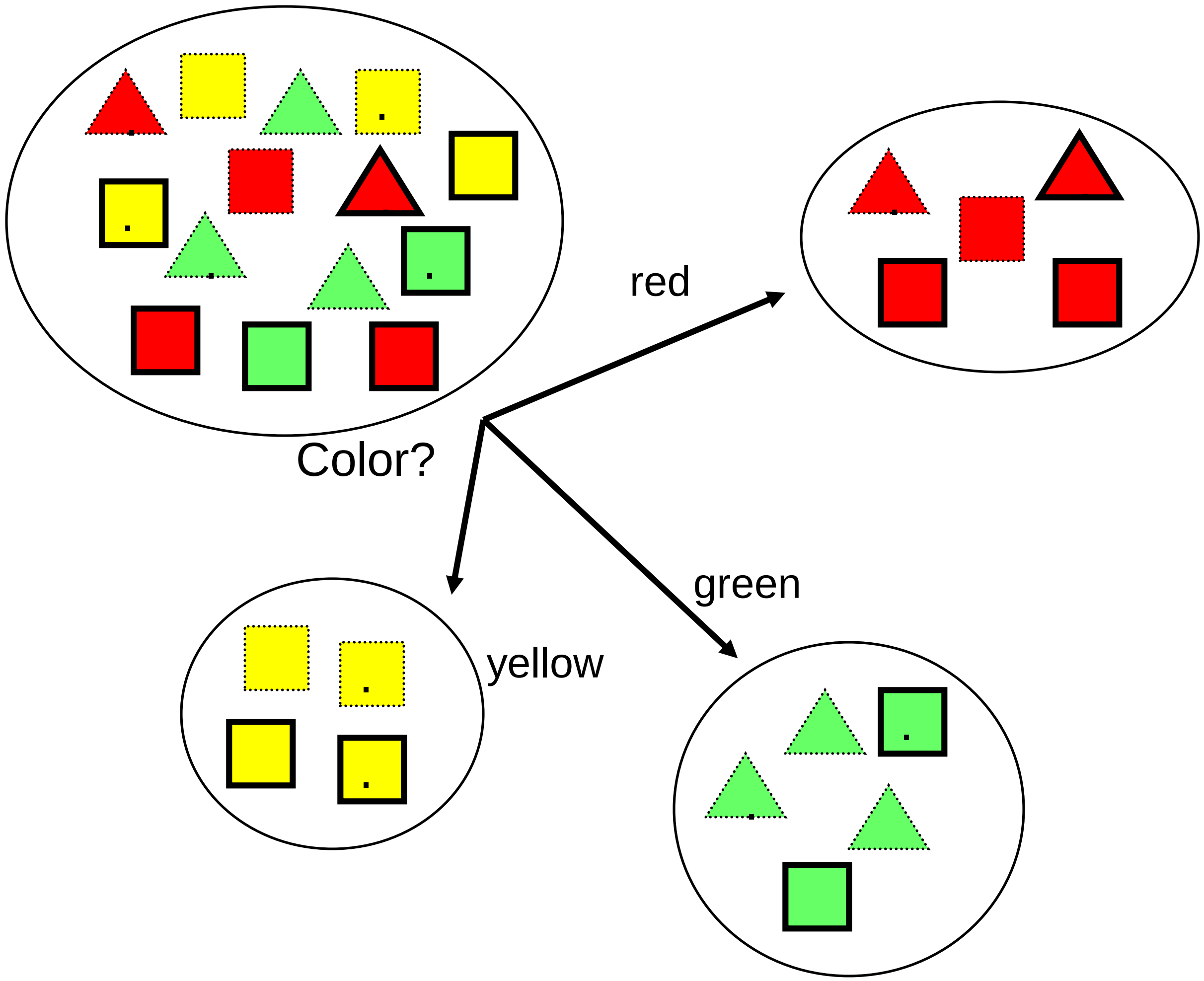
Another Example: Triangles & Sqaures

Information Gain



$$\boxed{Gain(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246\ bits}$$

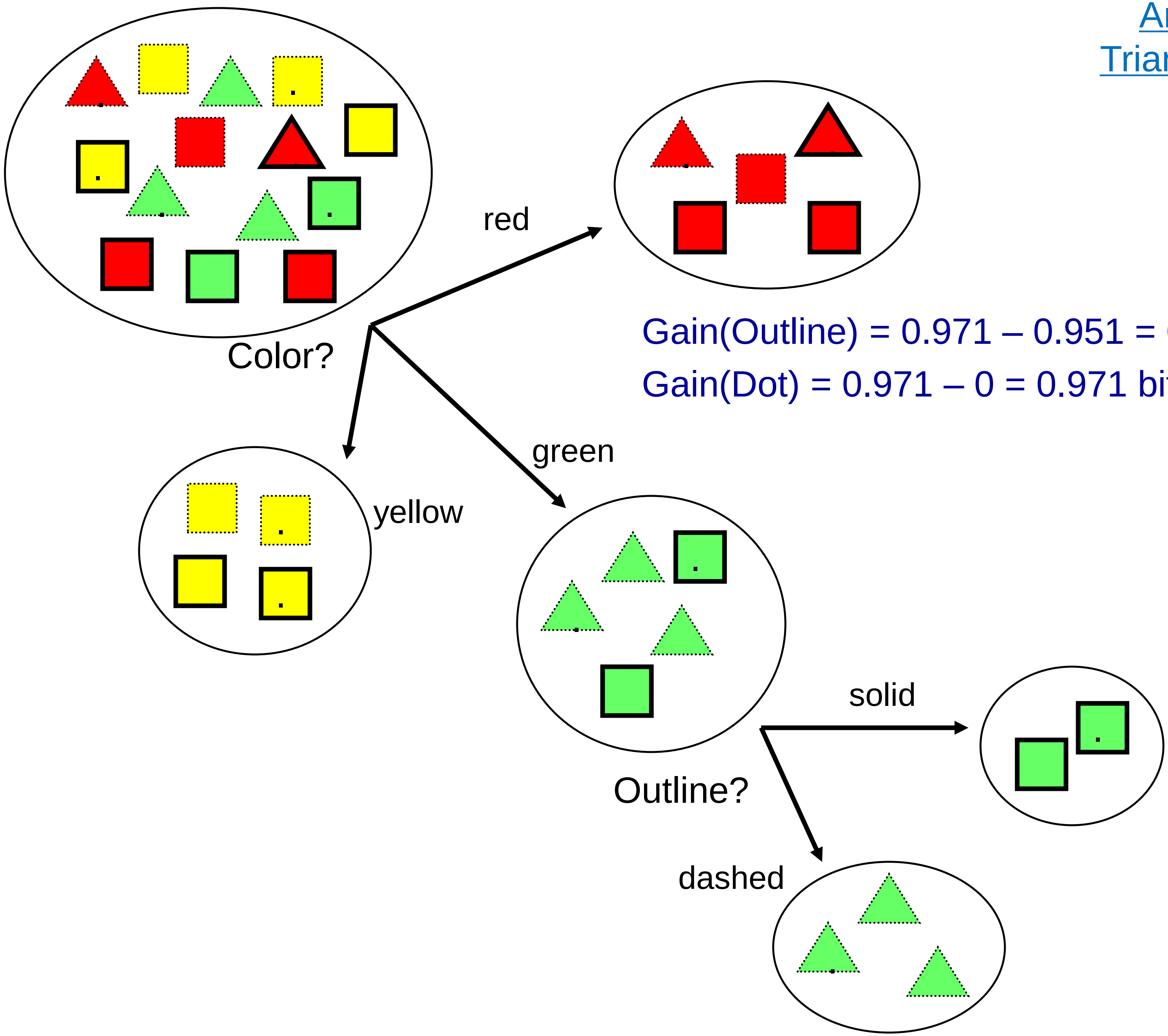
Another Example: Triangles & Sqaures



Gain(Outline) = 0.971 – 0 = 0.971 bits

Gain(Dot) = 0.971 – 0.951 = 0.020 bits

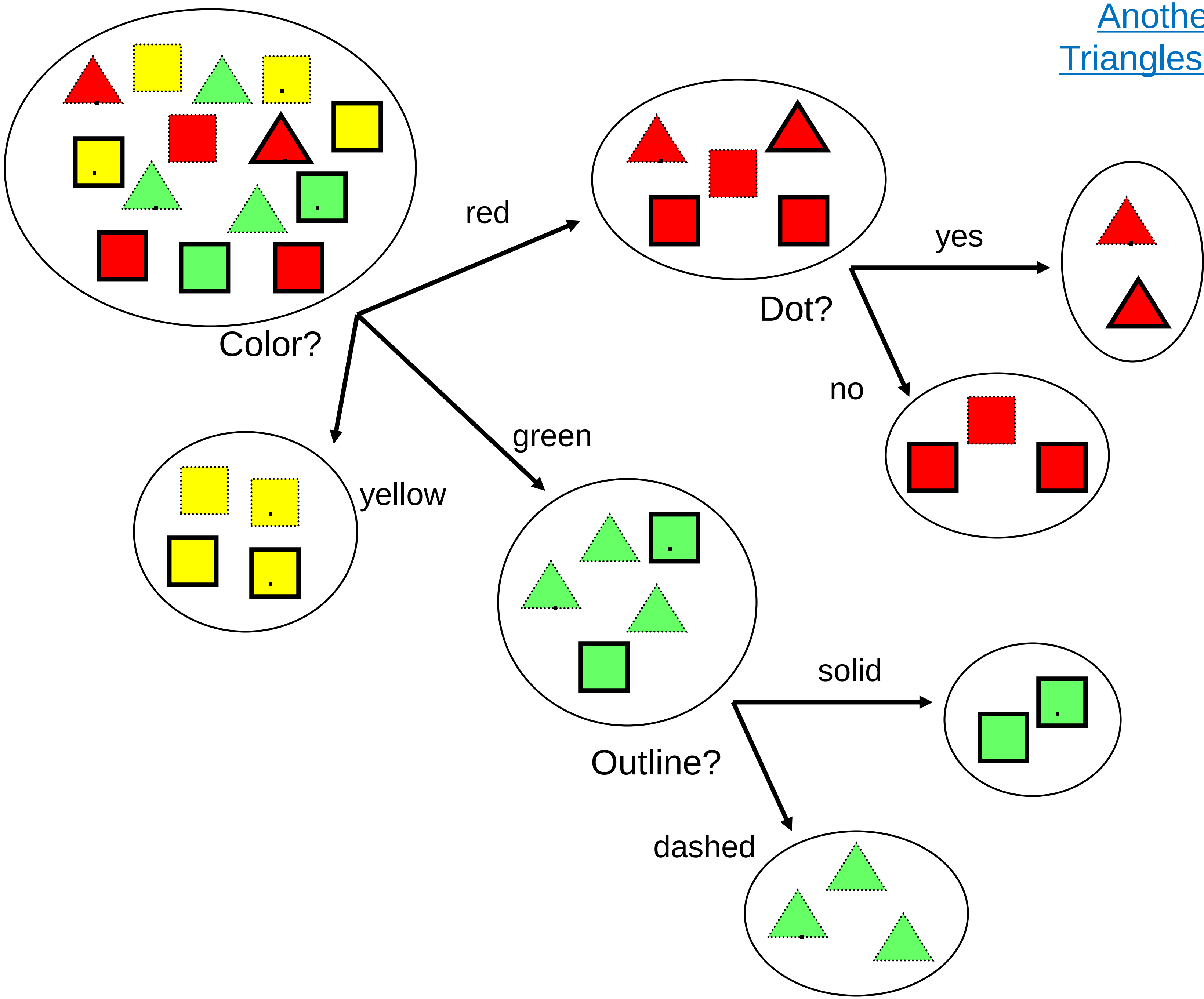
Another Example: Triangles & Sqaures



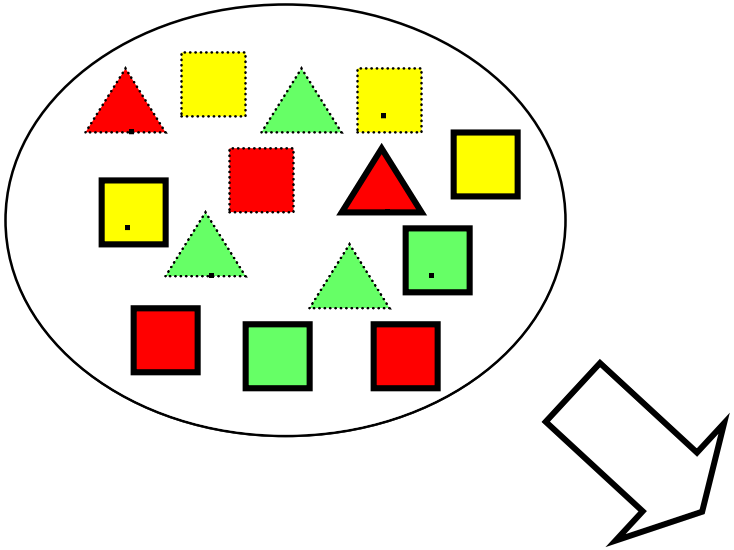
Gain(Outline) = 0.971 – 0.951 = 0.020 bits

Gain(Dot) = 0.971 – 0 = 0.971 bits

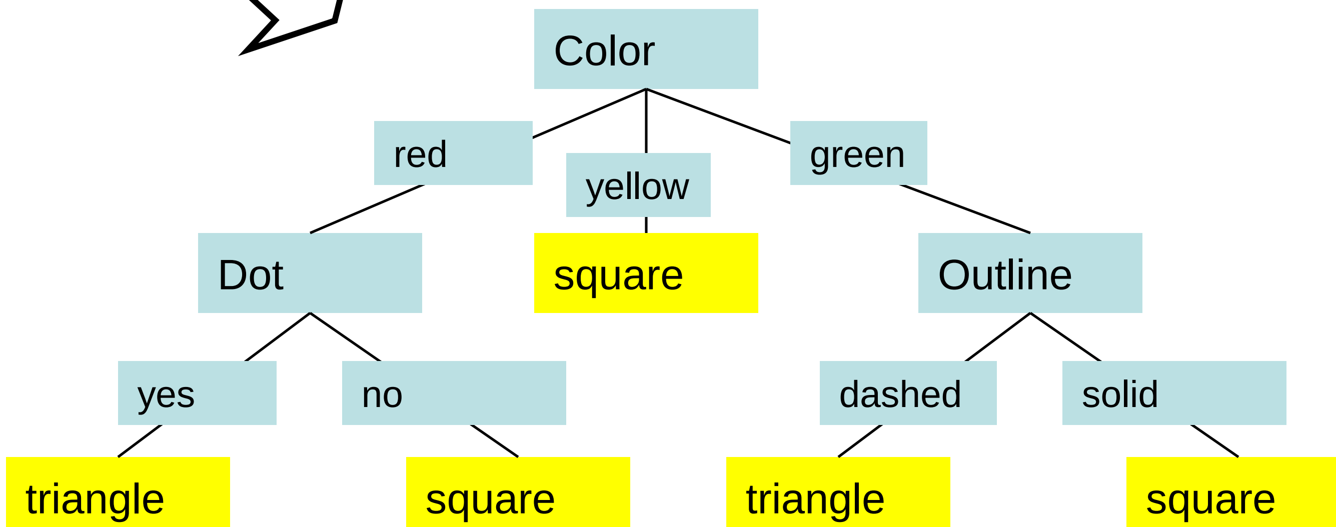
Another Example: Triangles & Sqaures



Another Example: Triangles & Sqaures



Decision Tree



ID3(Examples, Attributes, Label)

Let S be the set of Examples
Label is the target attribute (the prediction)
Attributes is the set of measured attributes
Create a Root node for tree
If all examples are labeled the same return a single node tree with Label
Otherwise Begin
A = attribute in Attributes that <u>*best*</u> classifies S
for each possible value v of A
Add a new tree branch corresponding to A=v
Let *Sv* be the subset of examples in S with A=v
if *Sv* is empty: add leaf node with the common value of Label in S
Else: below this branch add the subtree
ID3(*Sv*, Attributes - {a}, Label)
End
Return Root

History of Decision Tree Research

- Hunt and colleagues in Psychology used full search decision trees methods to model human concept learning in the 60's
- Quinlan developed ID3, with the information gain heuristics in the late 70's to learn expert systems from examples
- Breiman, Friedmans and colleagues in statistics developed CART (classification and regression trees) simultaneously
- A variety of improvements in the 80's: coping with noise, continuous attributes, missing data, non-axis parallel etc.
- Quinlan's updated algorithm, C4.5 (1993) is commonly used (New:C5)
- Boosting (or Bagging) over DTs is a very good general purpose algorithm

Next: Course review

- HW7 (Option):
  - Programming work: see course site for details