Splitting Criteria

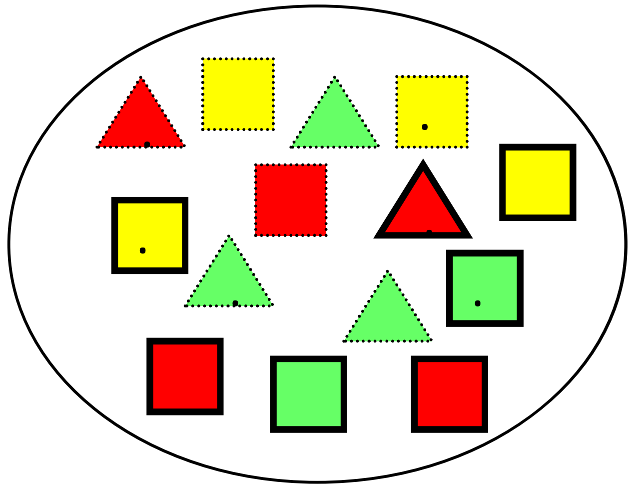
I. Splitting Criterion

- Central Idea : Select attribute which partitions the learning set into subsets as “pure” as possible
- A partition is PURE if all of the observations in it belong to the same class.

Example: Triangles and Squares

#	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



I. Entropy & Information Gain – C4.5

Shannon entropy

Measure of uncertainty

$$E(Y) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \times \log_2\left(\frac{n_{k.}}{n}\right)$$

Condition entropy

Expected entropy of Y knowing the values of X

$$E(Y/X) = -\sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \times \log_2\left(\frac{n_{kl}}{n_{.l}}\right)$$

Information gain

Reduction of uncertainty

$$G(Y/X) = E(Y) - E(Y/X)$$

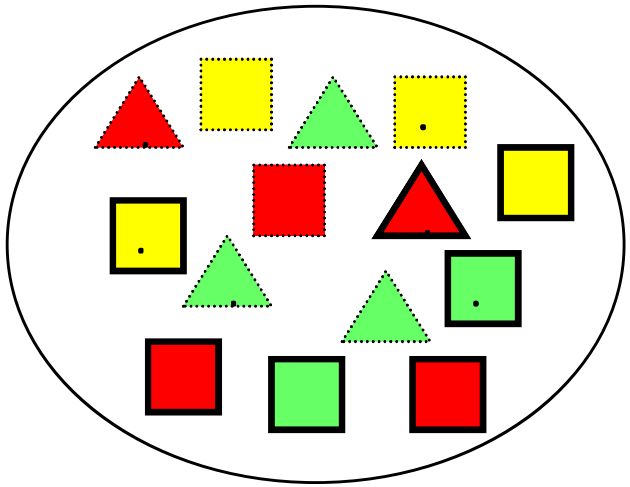
(Information) Gain ratio

Favors the splits with low number of leaves

$$GR(Y/X) = \frac{E(Y) - E(Y/X)}{E(X)}$$

Example: Entropy of given dataset

- 5 triangles
- 9 squares
- class probabilities



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

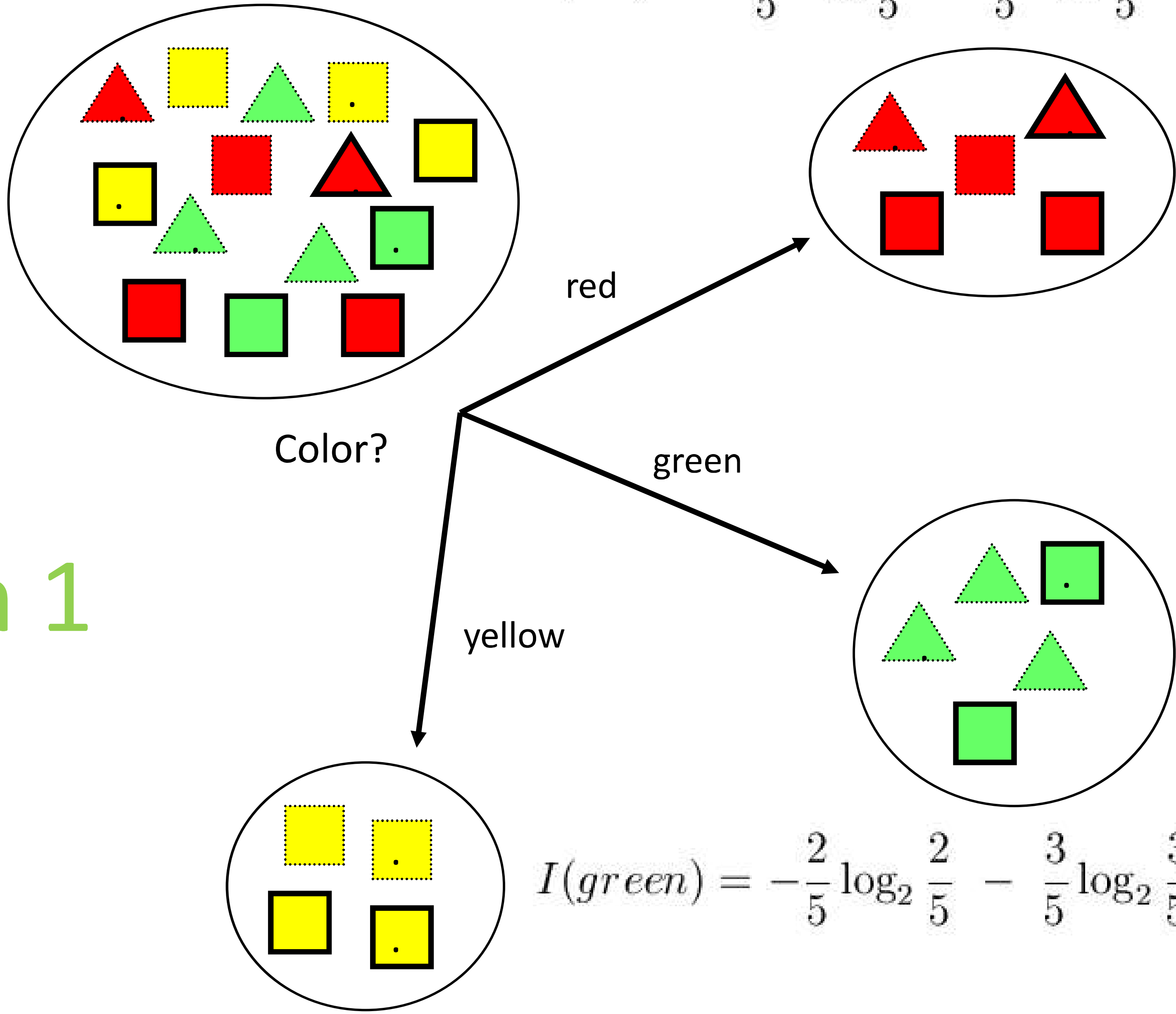
$$p(\triangle) = \frac{5}{14}$$

- entropy

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

Depth 1

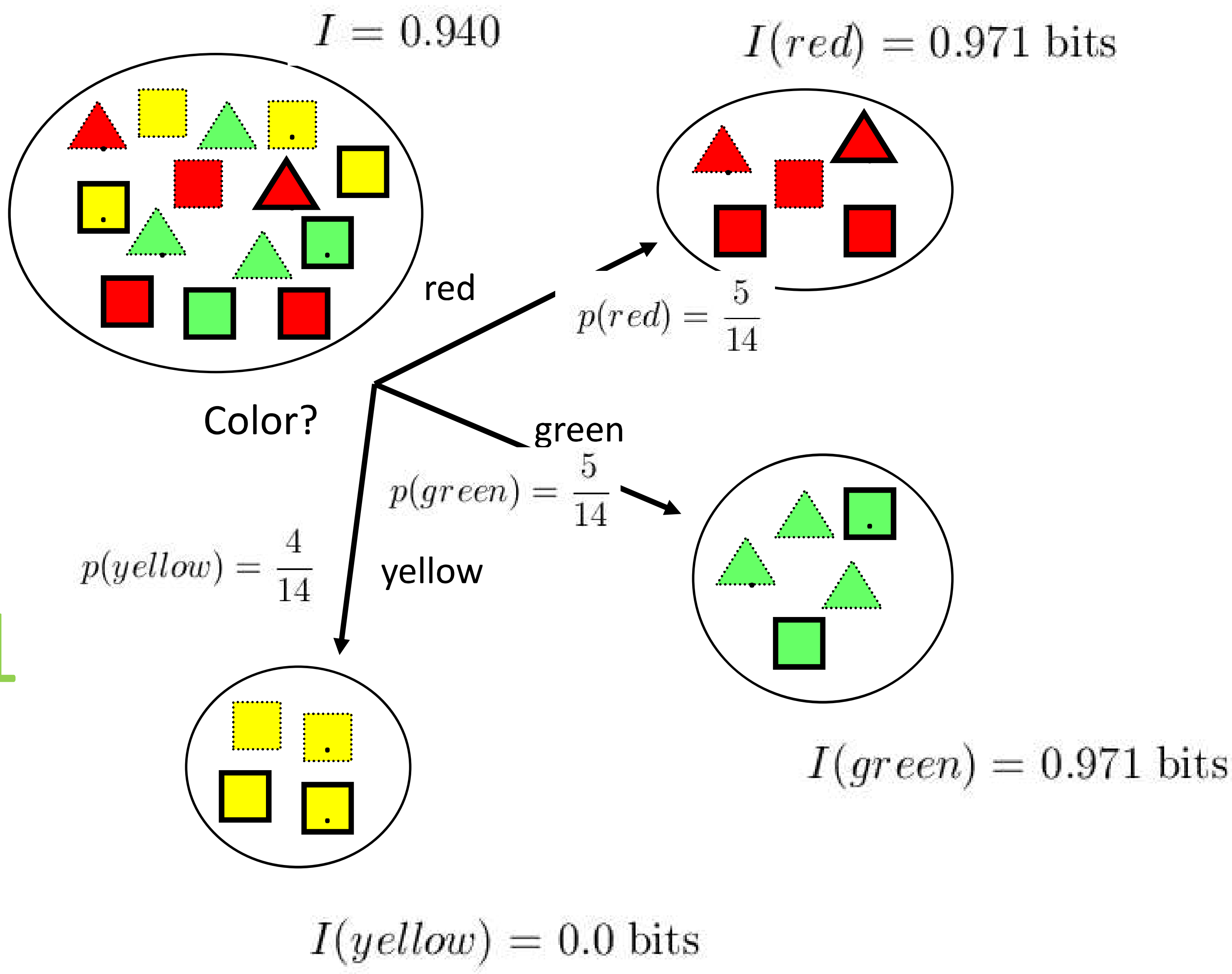
$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$



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

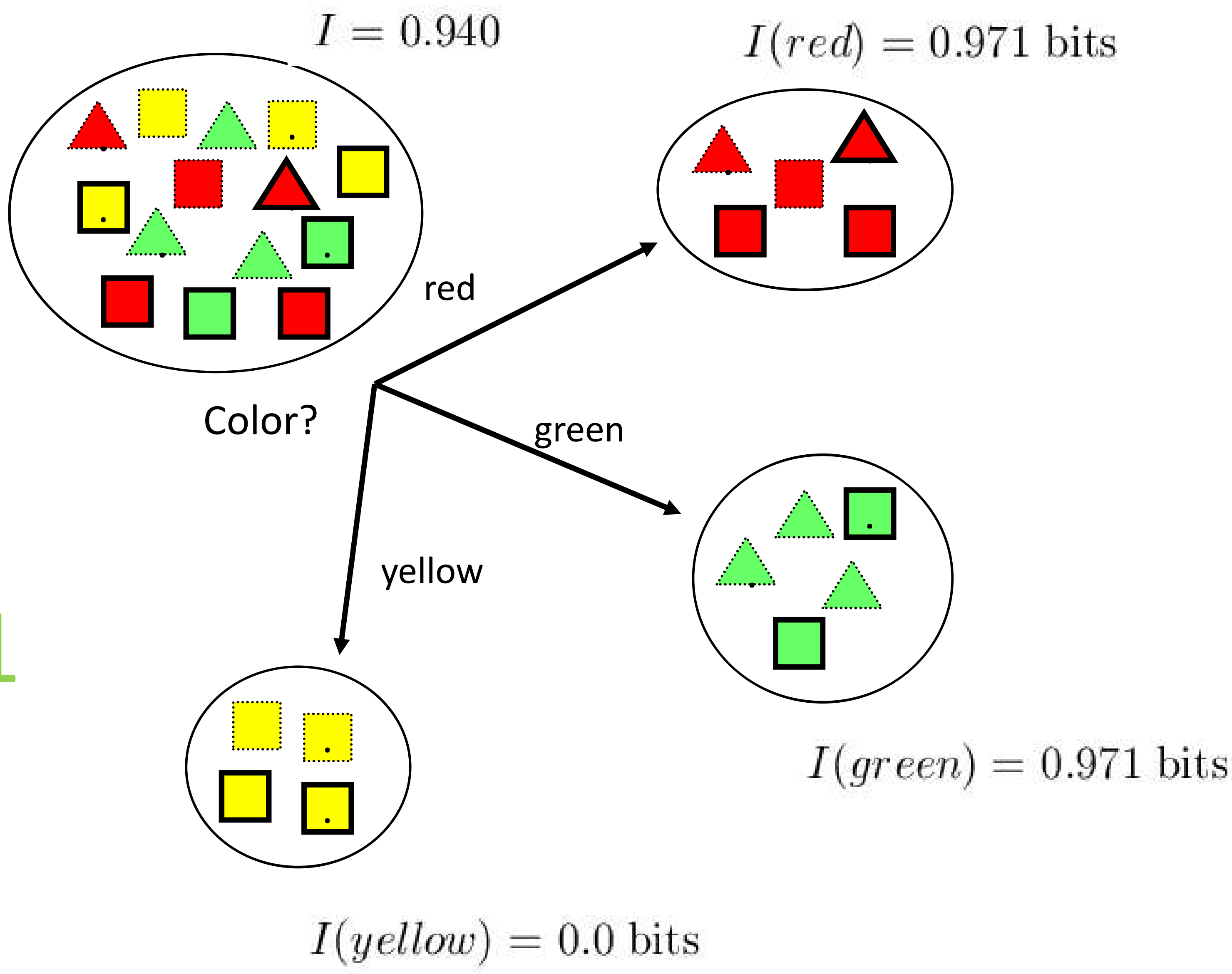
$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$

Depth 1



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

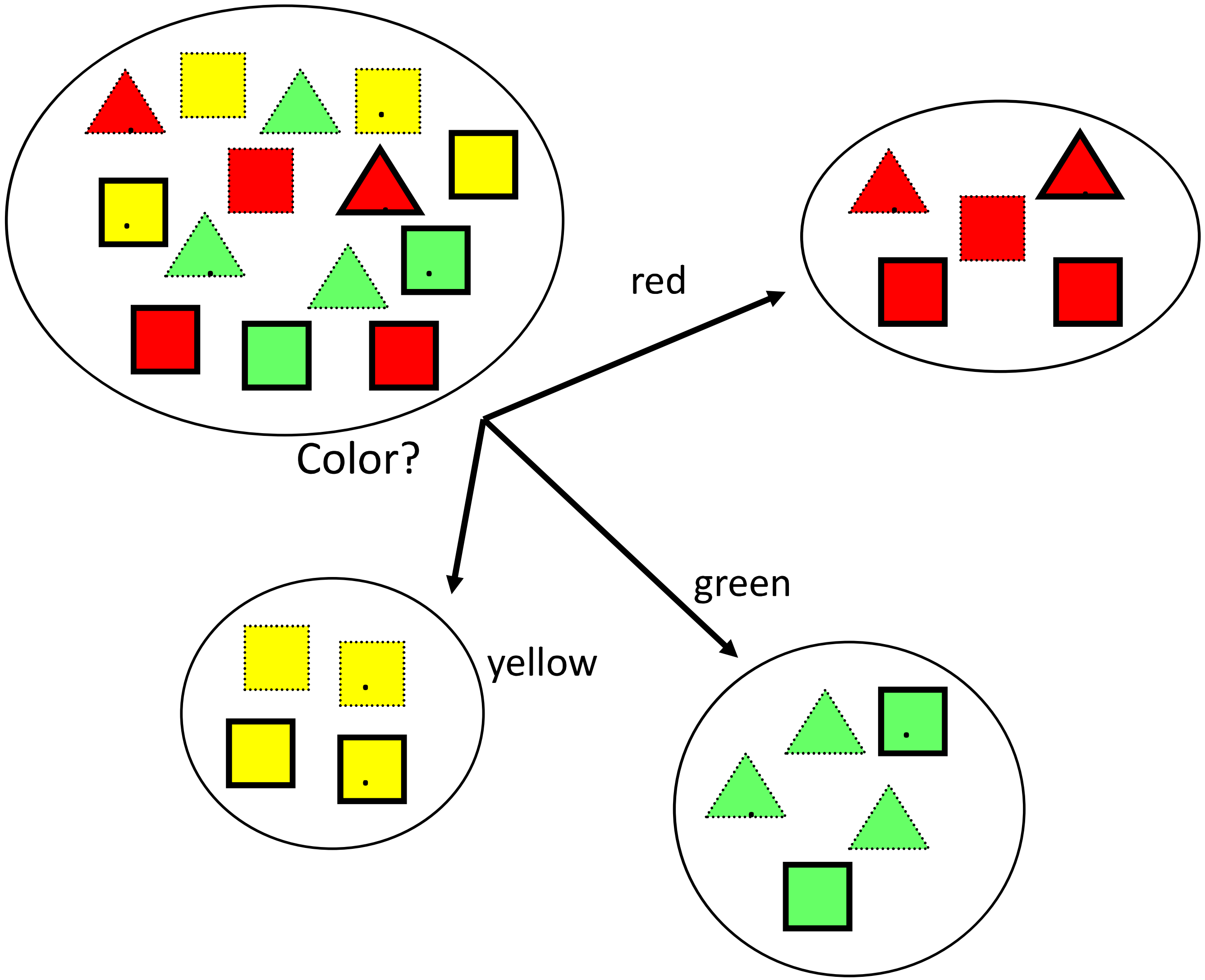
Depth 1



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

Depth1 :Information Gains

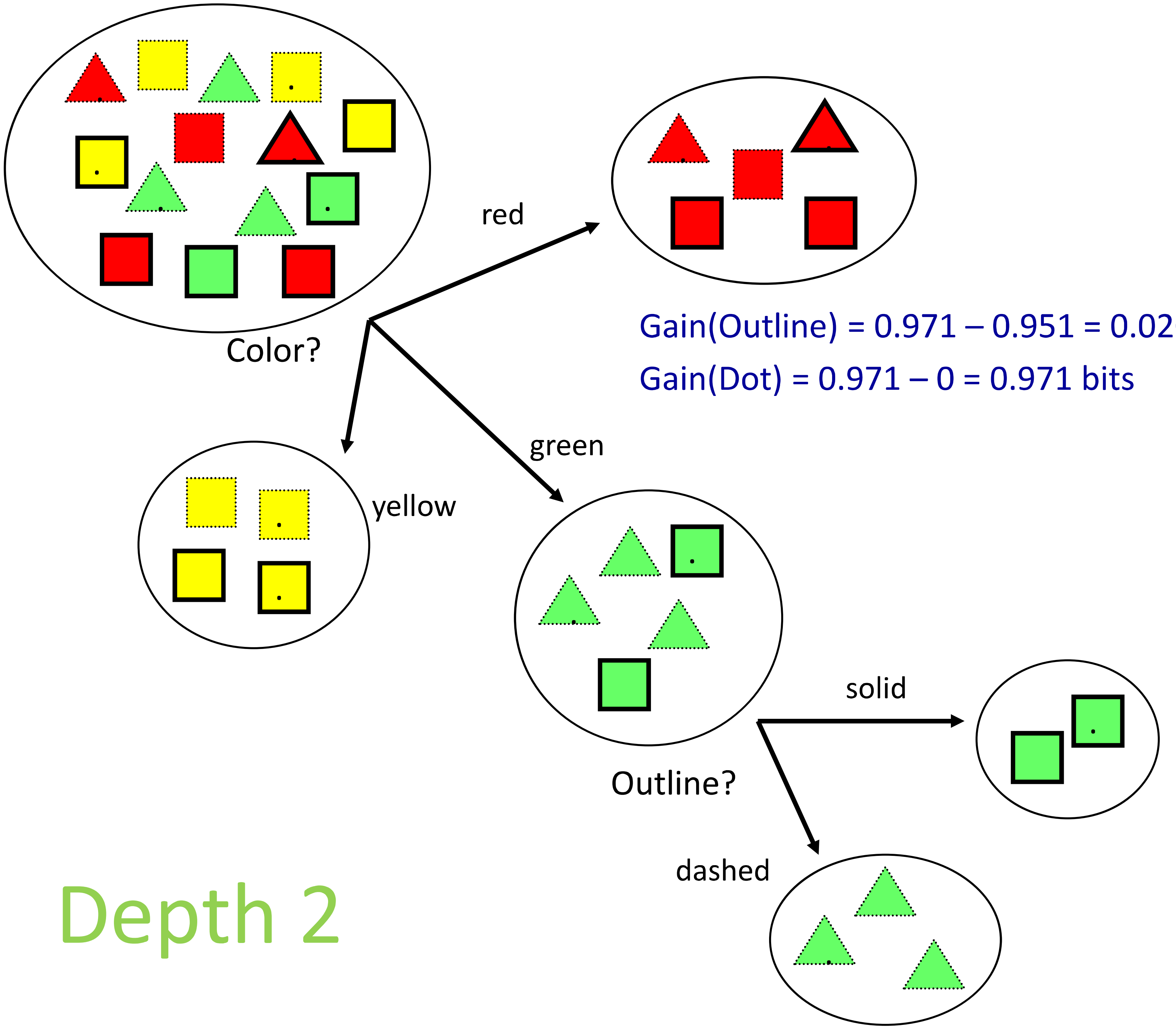
- Attributes
  - Gain(Color) = 0.246
  - Gain(Outline) = 0.151
  - Gain(Dot) = 0.048
- The attribute with the highest gain is chosen
- This heuristics is local (local minimization of impurity)



Gain(Outline) = 0.971 – 0 = 0.971 bits

Gain(Dot) = 0.971 – 0.951 = 0.020 bits

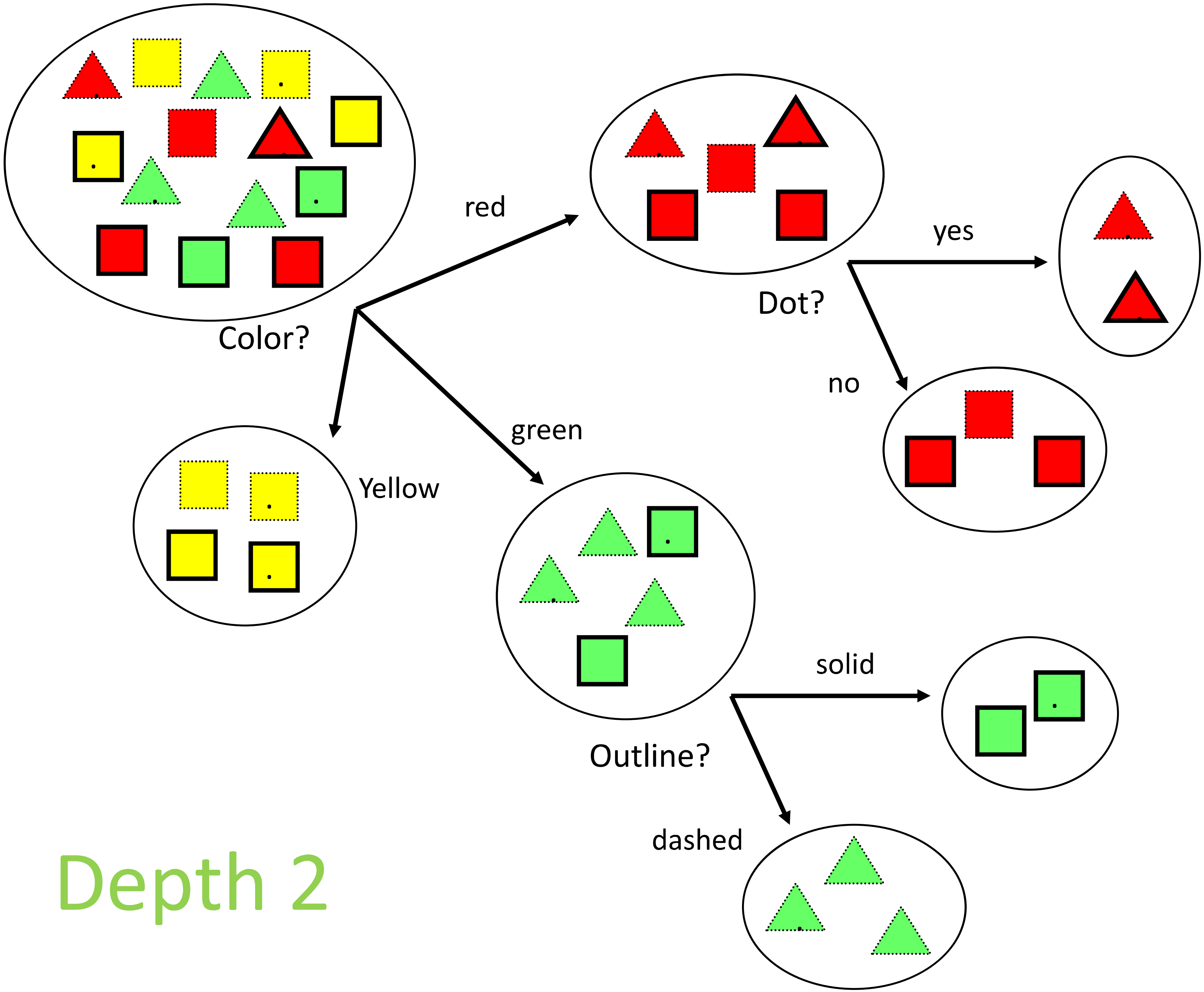
Depth 2



Gain(Outline) = 0.971 – 0.951 = 0.020 bits

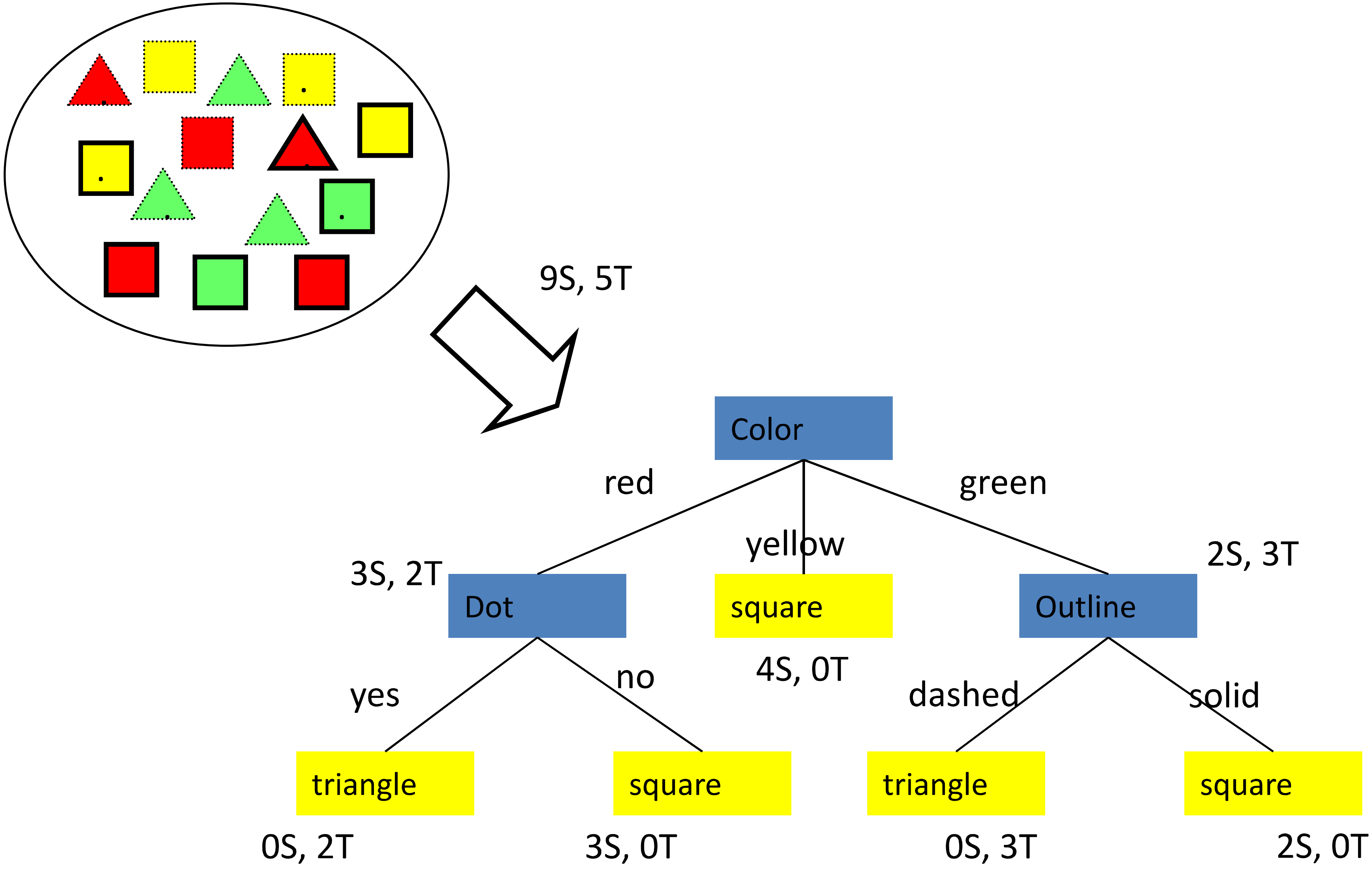
Gain(Dot) = 0.971 – 0 = 0.971 bits

Depth 2



Depth 2

Final Decision Tree



I. Gini Gain – CART

Gini index

Measure of impurity

$$I(Y) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \times \left(1 - \frac{n_{k.}}{n}\right)$$

Conditional impurity

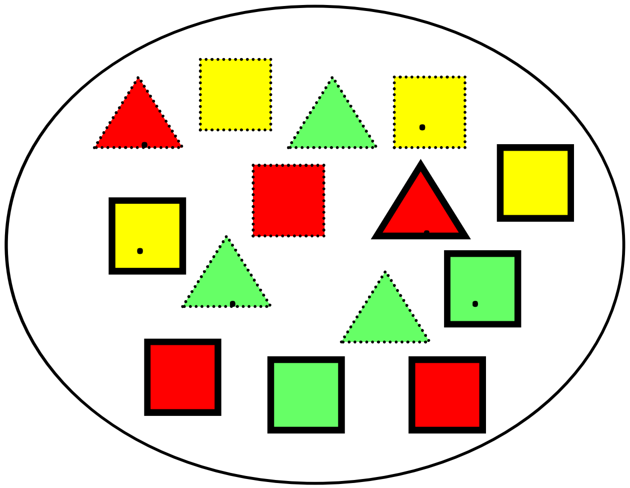
Average impurity of Y conditionally to X

$$I(Y/X) = -\sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \times \left(1 - \frac{n_{kl}}{n_{.l}}\right)$$

Gain

$$D(Y/X) = I(Y) - I(Y/X)$$

Gini Index



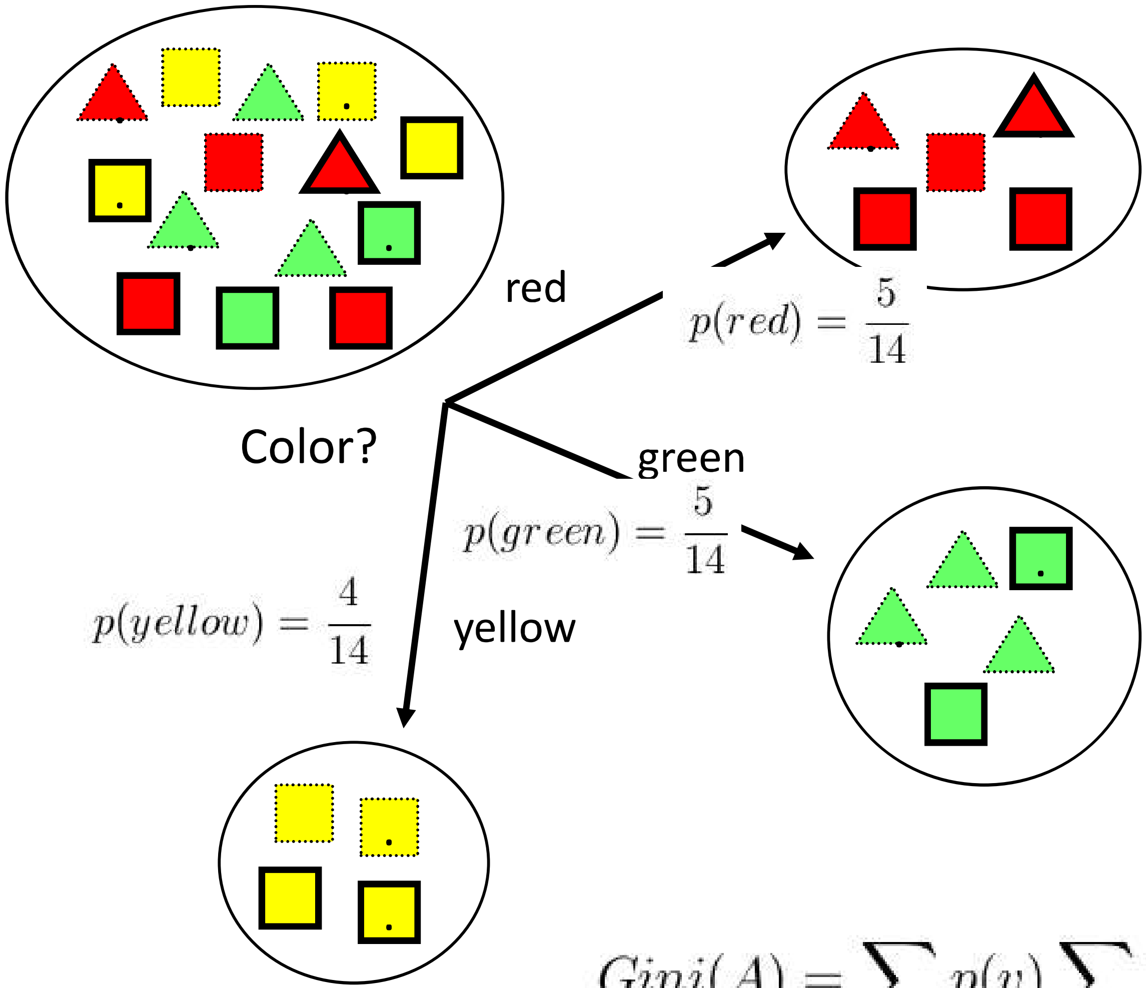
$$p(\square) = \frac{9}{14}$$

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

$$Gini = \sum_{i \neq j} p(i)p(j)$$

$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

Gini Index for Color



$p(red) = \frac{5}{14}$ $p(green) = \frac{5}{14}$ $p(yellow) = \frac{4}{14}$

$$Gini(A) = \sum_v p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

$$Gini(\text{Color}) = \frac{5}{14} \times (\frac{3}{5} \times \frac{2}{5}) + \frac{5}{14} \times (\frac{2}{5} \times \frac{3}{5}) + \frac{4}{14} \times (\frac{4}{4} \times \frac{0}{4}) = 0.171$$

Gain of Gini Index

$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

$$Gini(\mathrm{Color}) = \frac{5}{14} \times (\frac{3}{5} \times \frac{2}{5}) + \frac{5}{14} \times (\frac{2}{5} \times \frac{3}{5}) + \frac{4}{14} \times (\frac{4}{4} \times \frac{0}{4}) = 0.171$$

$$\boxed{GiniGain(\mathrm{Color}) = 0.230 - 0.171 = 0.058}$$

I. Un-biased measures

- Allows to alleviate the data fragmentation problem

- Gain Ratio corrects the bias of the information gain
- The Gini reduction in impurity is biased in favor of variables with more levels

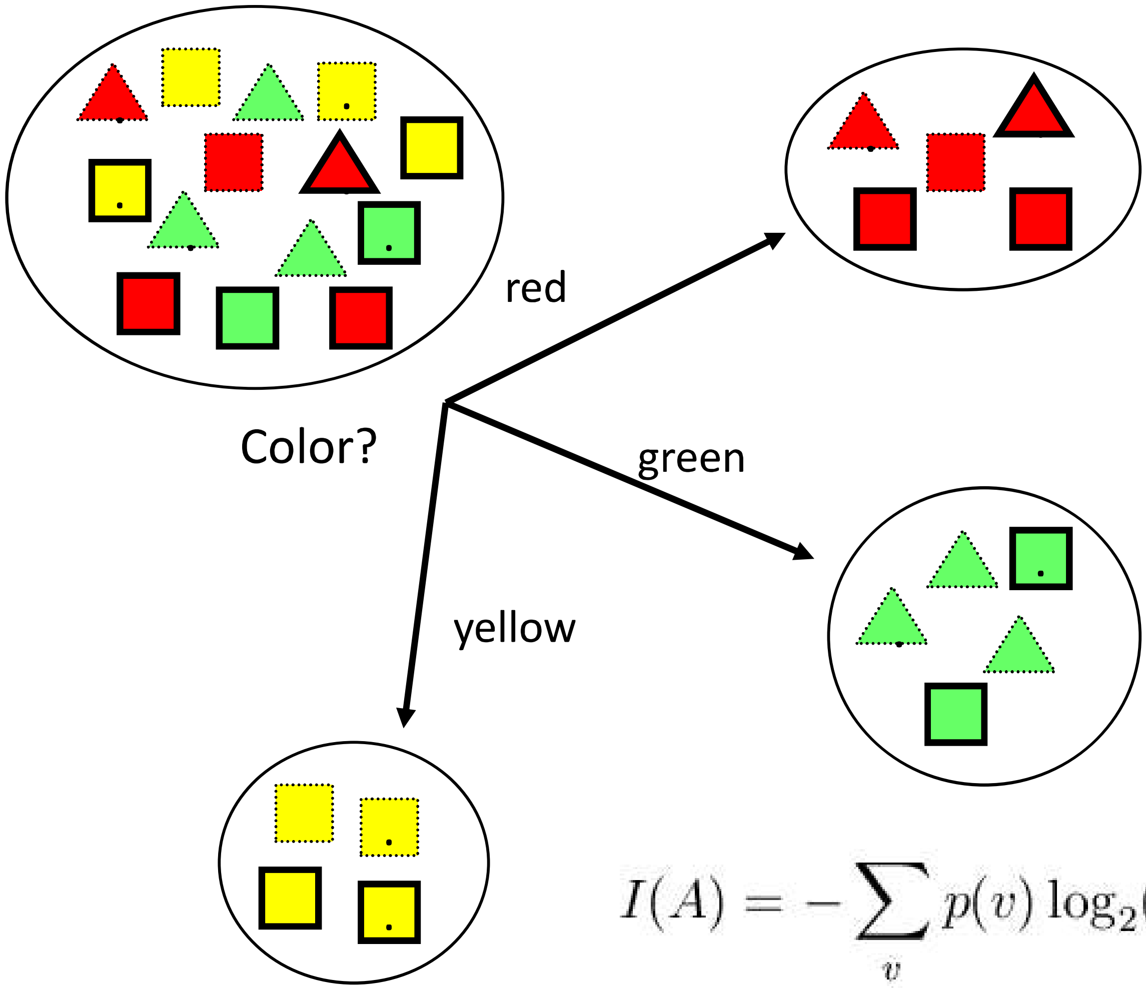
(but the CART algorithm constructs necessarily a binary decision tree)

Problems with Information Gain

Attributes which have a large number of possible values -> leads to many child nodes.

- Information gain is biased towards choosing attributes with a large number of values
- This may result in *overfitting* (selection of an attribute that is non-optimal for prediction)

Information Gain Ratio



$$I(A) = -\sum_{v} p(v) \log_2(p(v))$$

$$I(\text{Color}) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.58 \text{ bits}$$

$$GainRatio(\text{Color}) = \frac{Gain(\text{Color})}{I(\text{Color})} = \frac{0.940 - 0.694}{1.58} = 0.156$$

Information Gain and Information Gain Ratio

A	|v(A)|	Gain(A)	GainRatio(A)
Color	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

Three Impurity Measures

A	**Gain(A)**	**GainRatio(A)**	**GiniGain(A)**
Color	0.247	0.156	0.058
Outline	0.152	0.152	0.046
Dot	0.048	0.049	0.015