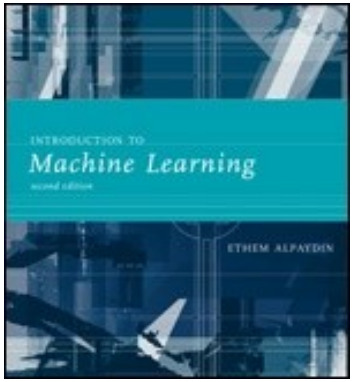
Lecture Slides for

INTRODUCTION TO

Machine Learning

2nd Edition



ETHEM ALPAYDIN, modified by Leonardo Bobadilla and some parts from http://www.cs.tau.ac.il/~apartzin/MachineLearning/ and www.cs.princeton.edu/courses/archive/fall01/cs302/notes/11.../EM.ppt

 alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2m

Outline

Previous class
Ch 8: Clustering
This class:
Ch 9: Decision Trees

CHAPTER 7:

Clustering

Example:

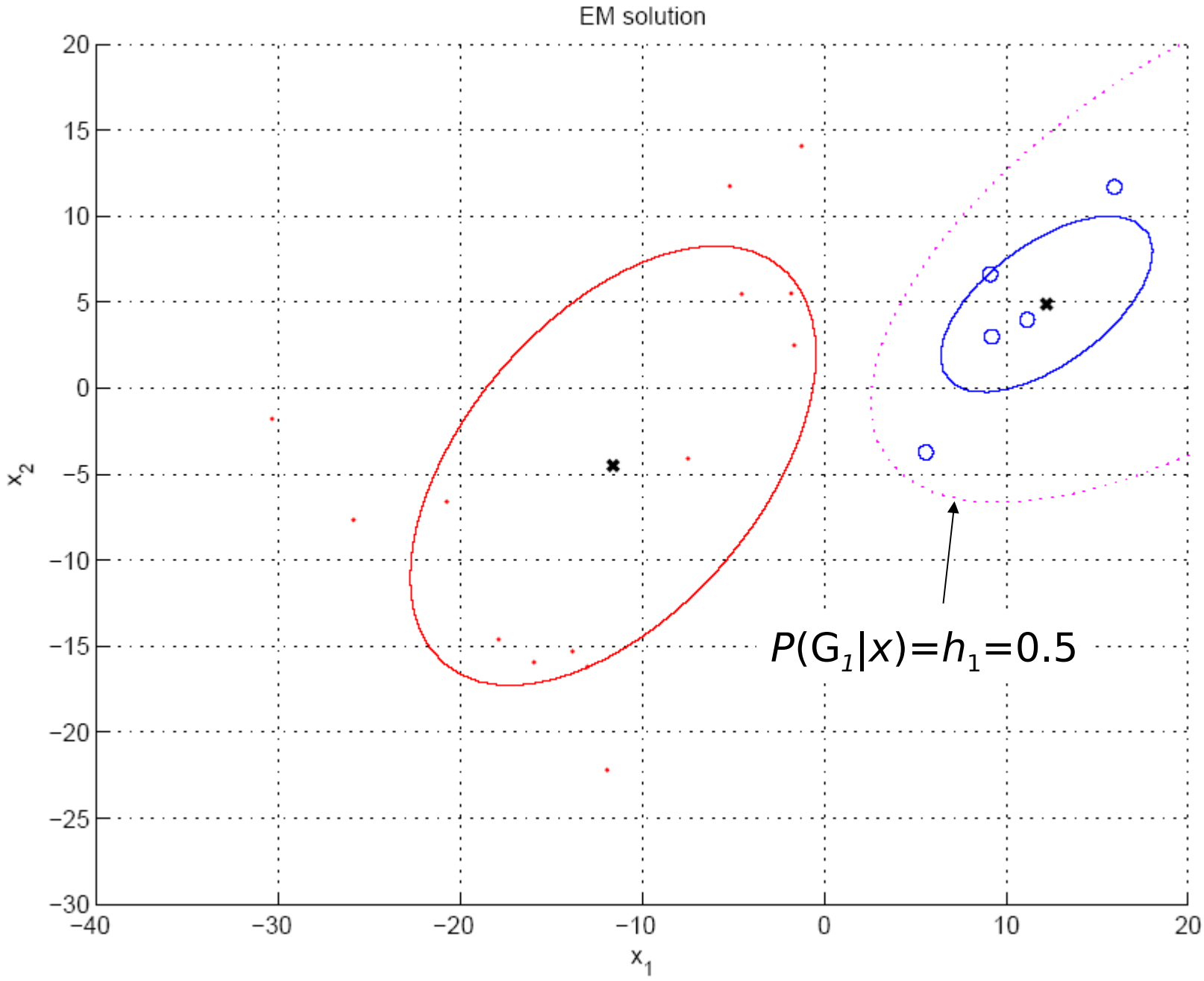
- Data came from mix of Gaussians $\hat{p}_i(\mathbf{x}^t|\Phi) \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$,
- Maximize likelihood assuming we know latent "indicator variable"

$$\mathbf{m}_i^{l+1} = \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t}$$

$$\mathbf{S}_i^{l+1} = \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i^{l+1})(\mathbf{x}^t - \mathbf{m}_i^{l+1})^T}{\sum_t h_i^t}$$

- E-step: expected value of indicator variables

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_i)^T \mathbf{S}_i^{-1} (\mathbf{x}^t - \mathbf{m}_i)]}{\sum_j \pi_j |\mathbf{S}_j|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_j)^T \mathbf{S}_j^{-1} (\mathbf{x}^t - \mathbf{m}_j)]}$$


EM solution
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−40
−30
−20
−10
0
10
20
x2
x1
P(G1|x)=h1=0.5

.1)

EM for Gaussian mixtures

- Assume all groups/clusters are Gaussians
- Multivariate Uncorrelated
- Same Variance
- Harden indicators
  - EM: expected values are between 0 and 1
  - K-means: 0 or 1
- Same as k-means

Dimensionality Reduction vs. Clustering

- Dimensionality reduction methods find correlations between features and group features
  - Age and Income are correlated

- Clustering methods find similarities between instances and group instances
  - Customer A and B are from the same cluster

Clustering: Usage for supervised learning

- Describe data in terms of cluster
  - Represent all data in cluster by cluster mean
  - Range of attributes

- Map data into new space(preprocessing)
  - d- dimension original space
  - k- number of clusters
  - Use indicator variables as data representations
  - k might be larger then d

Mixture of Mixtures

- In classification, the input comes from a mixture of classes (supervised).
- If each class is also a mixture, e.g., of Gaussians, (unsupervised), we have a mixture of mixtures:

$$p(\boldsymbol{x}|\mathrm{C}_i) = \sum_{j=1}^{k_i} p(\boldsymbol{x}|\mathrm{G}_{ij}) P(\mathrm{G}_{ij})$$

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|\mathrm{C}_i) P(\mathrm{C}_i)$$

Hierarchical Clustering

- Probabilistic view
  - Fit mixture model to data
  - Find codewords minimizing reconstruction error

- Hierarchical clustering
  - Group similar items together
  - No specific model/distribution
  - Items in groups is more similar to each other than instances in different groups

Hierarchical Clustering

Minkowski (L_p) (Euclidean for $p = 2$)

$$d_m\left(\boldsymbol{x}^r, \boldsymbol{x}^s\right) = \left[\sum_{j=1}^{d}\left(x_j^r - x_j^s\right)^p\right]^{1/p}$$

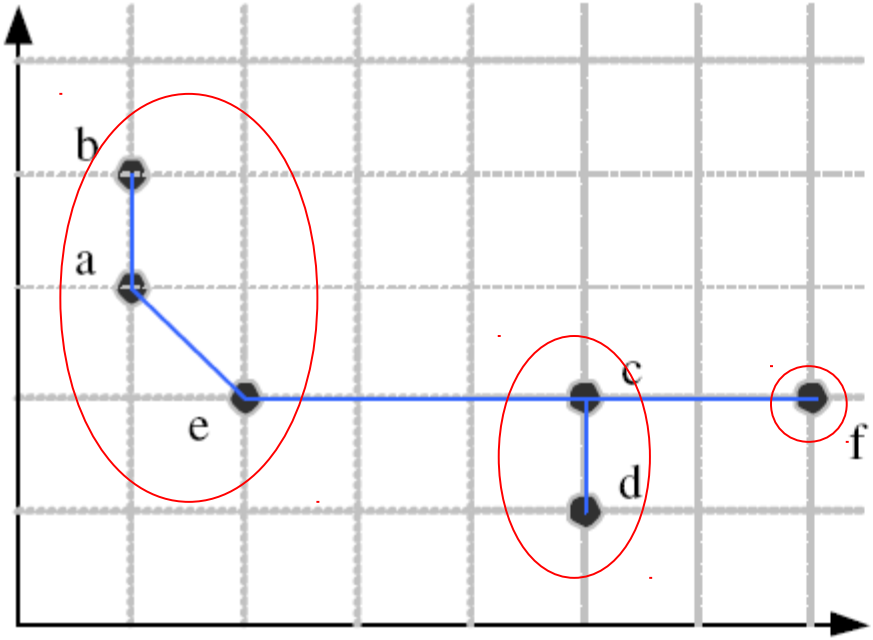
City-block distance

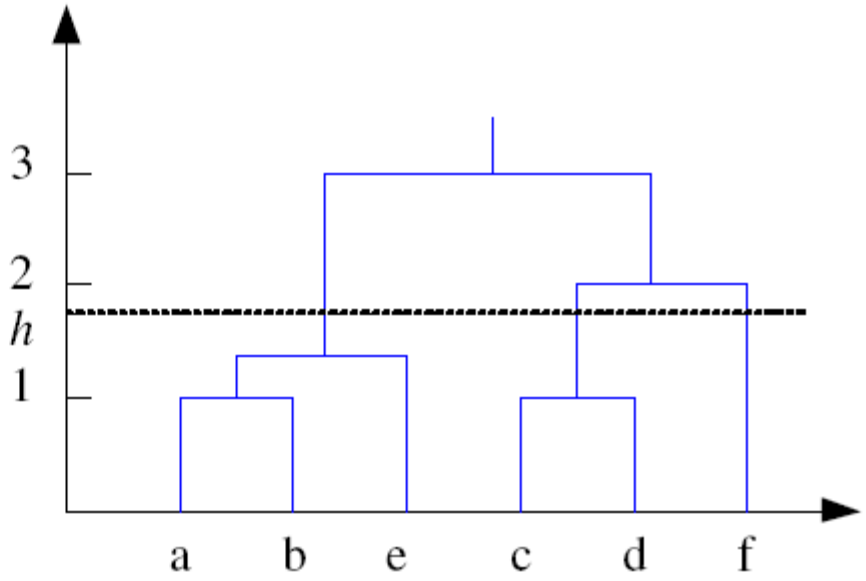
$$d_{cb}\left(\boldsymbol{x}^r, \boldsymbol{x}^s\right) = \sum_{j=1}^{d}\left|x_j^r - x_j^s\right|$$

Agglomerative Clustering

- Start with clusters each having single point
- At each step merge similar clusters
- Measure of similarity
  - Minimal distance(single link)
    - Distance between closest points in 2 groups
  - Maximal distance(complete link)
    - Distance between most distant points in 2 groups
  - Average distance
    - Distance between group centers

Example: Single-Link Clustering





Dendrogram

Choosing k

- Defined by the application, e.g., image quantization
- Plotting data in two dimensions using PCA

- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)

CHAPTER 9:

Decision Trees

Slides from: Blaž Zupan and Ivan Bratko

magix.fri.uni-lj.si/predavanja/uisp

Motivation

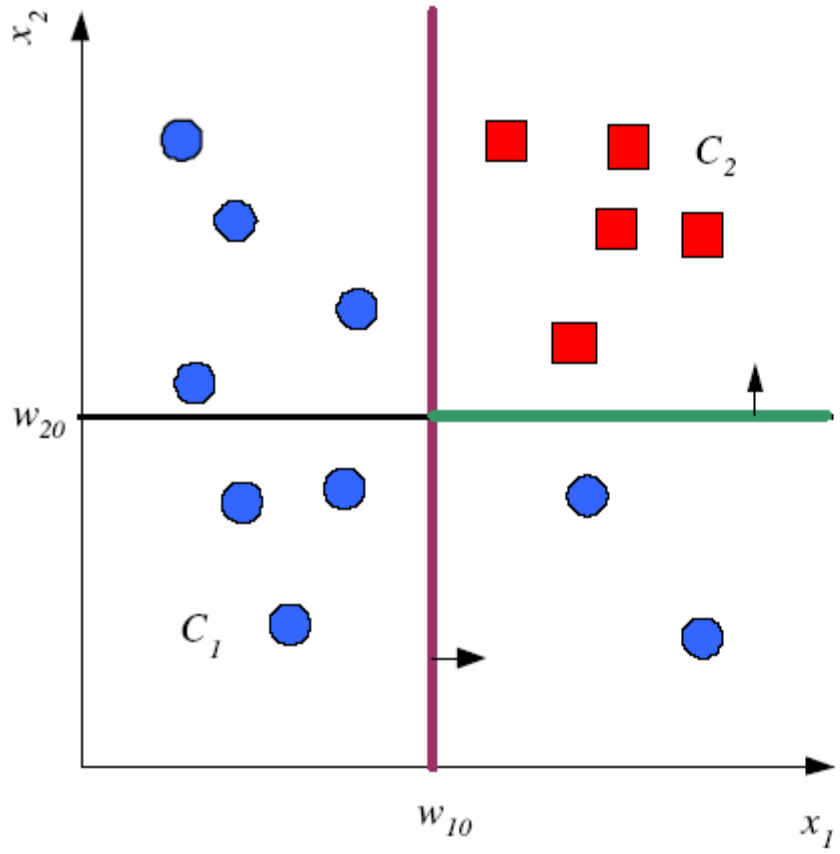
- Parametric Estimation
  - Assume model for class probability or regression
  - Estimate parameters from all data

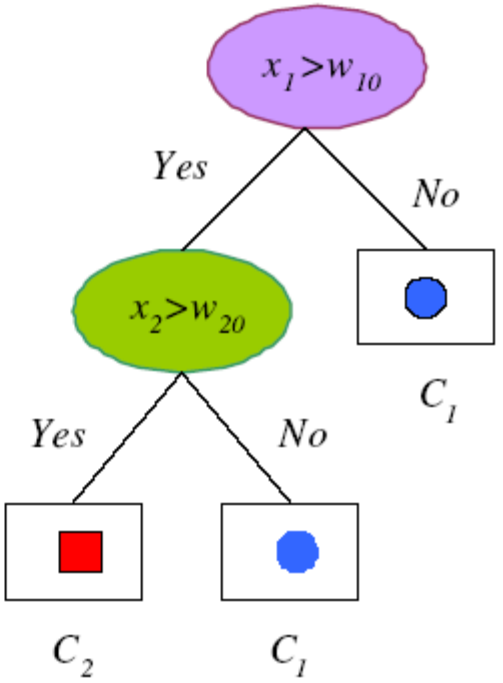
Motivation

- Pre-split training data into region using small number of simple rules organized in hierarchical manner

- Decision Trees
  - Internal decision nodes have splitting rule
  - Terminal leaves have class labels for classification problem or values for regression problem

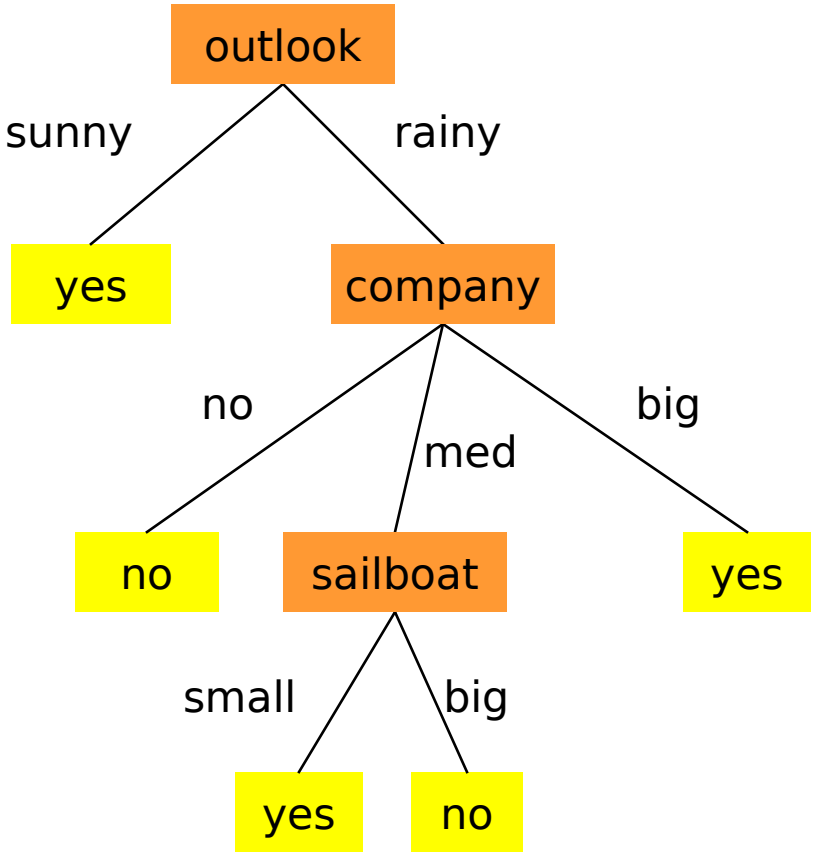
Tree Uses Nodes, and Leaves





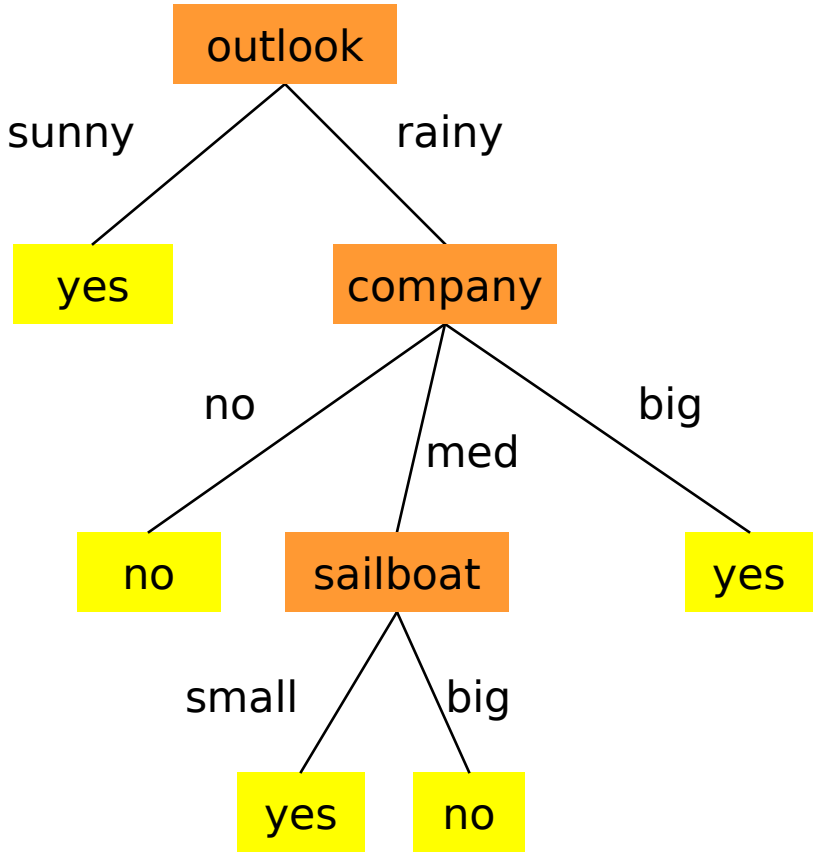
An Example Data Set and Decision Tree

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	?

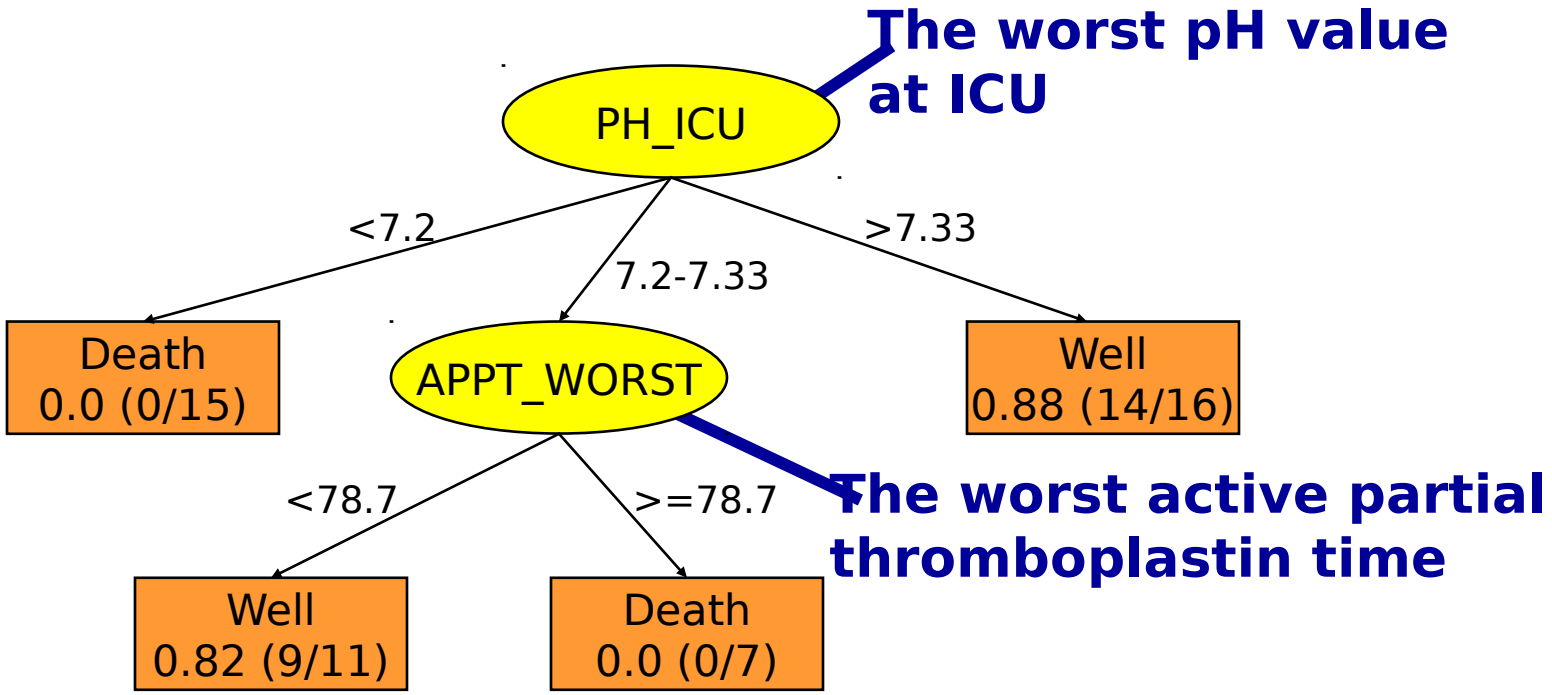


Learning Decision Trees

- Data Set (Training Set)
  - Each example = Attributes + Class
- Description = Decision tree

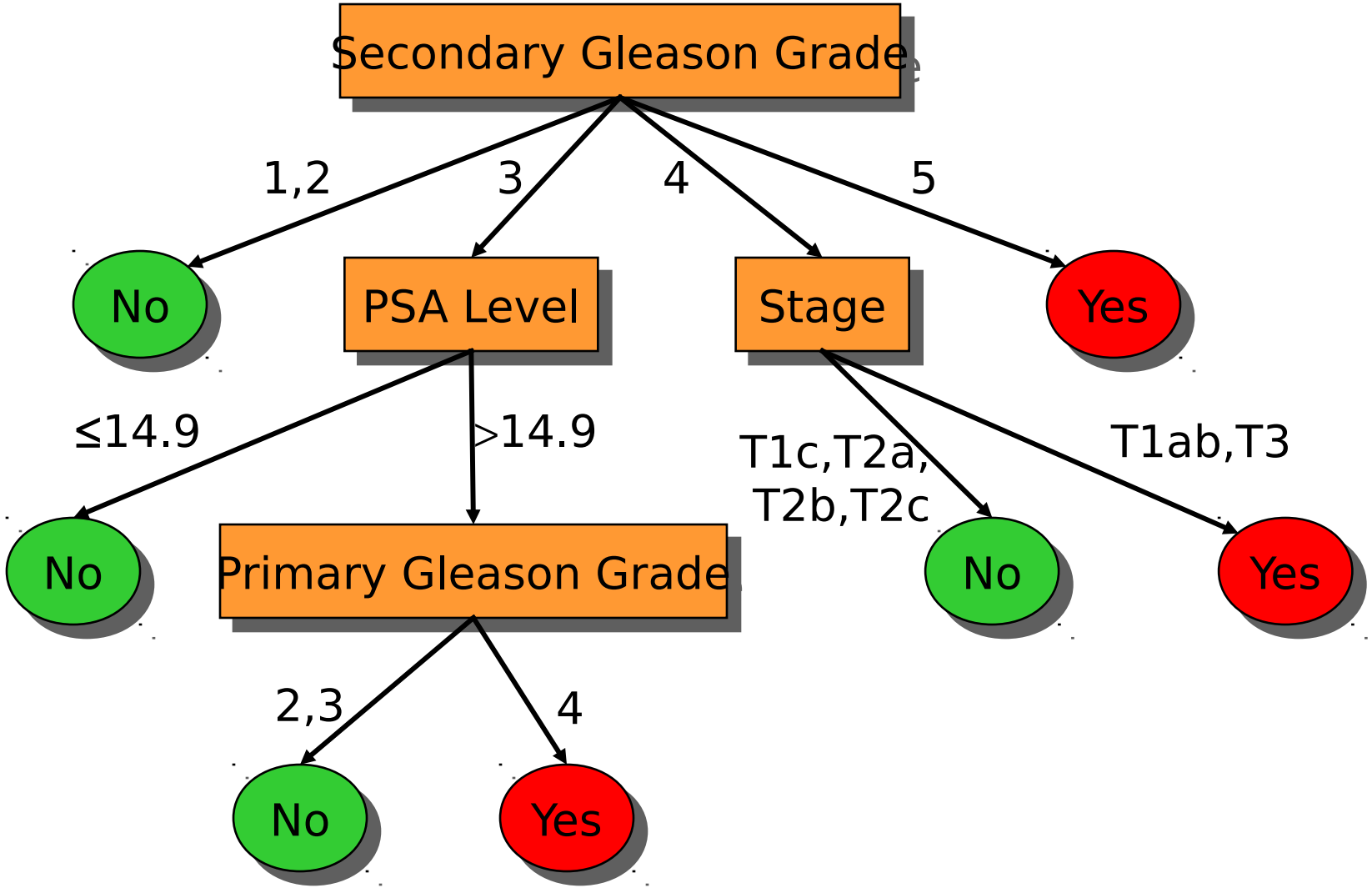
- Recursive Partitioning

Analysis of Severe Trauma Patients Data



PH_ICU and APPT_WORST are exactly the two factors (theoretically) advocated to be the most important ones in the study by Rotondo et al., 1997.

Prostate cancer recurrence



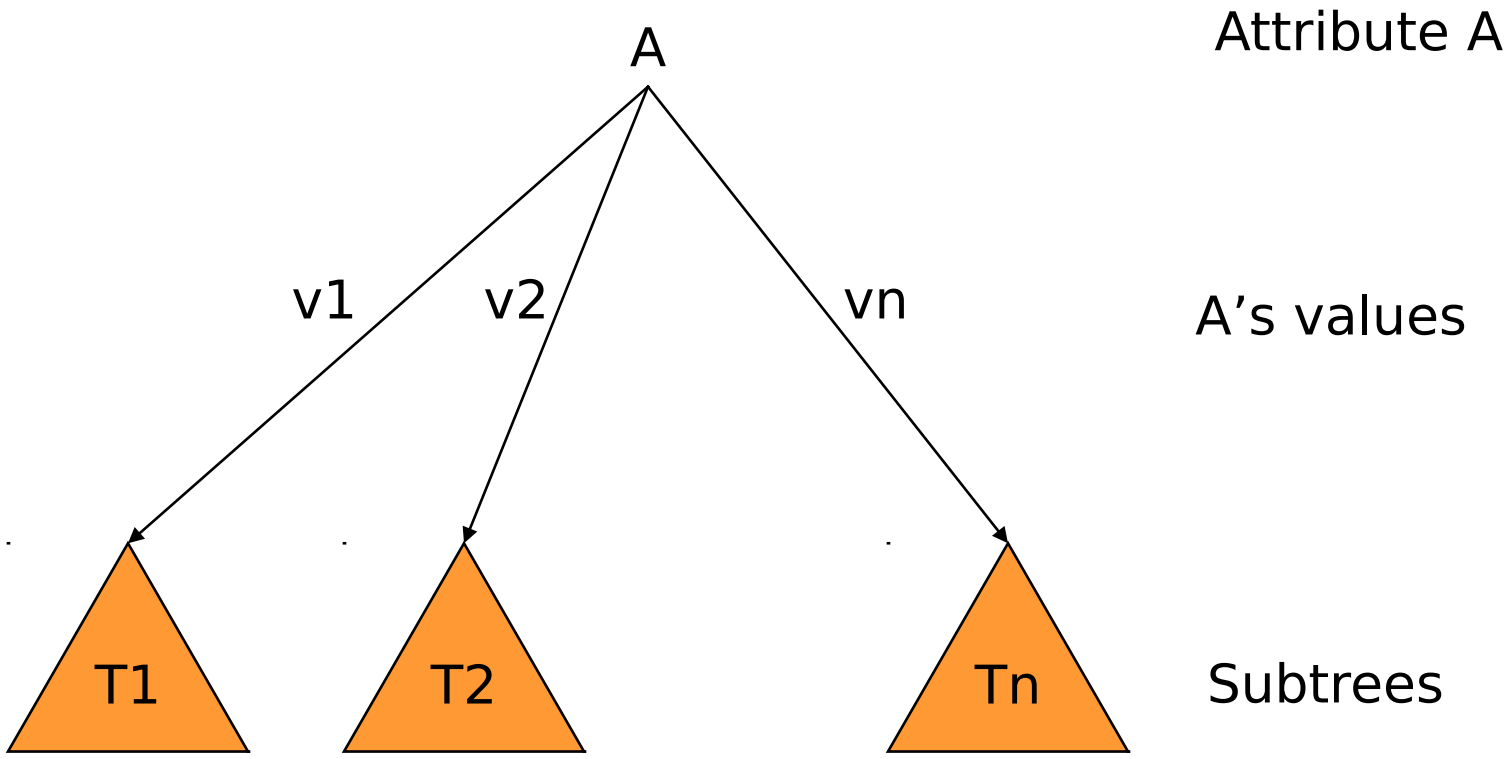
ID3 Algorithm

To construct decision tree T from learning set X:

- **If** all examples in X belong to some class C
  Then
  make leaf labeled C
- **Otherwise**
  - select the “most informative” attribute A
  - partition S according to A’s values
  - recursively construct subtrees T1, T2, ..., for the subsets of S

ID3 Algorithm

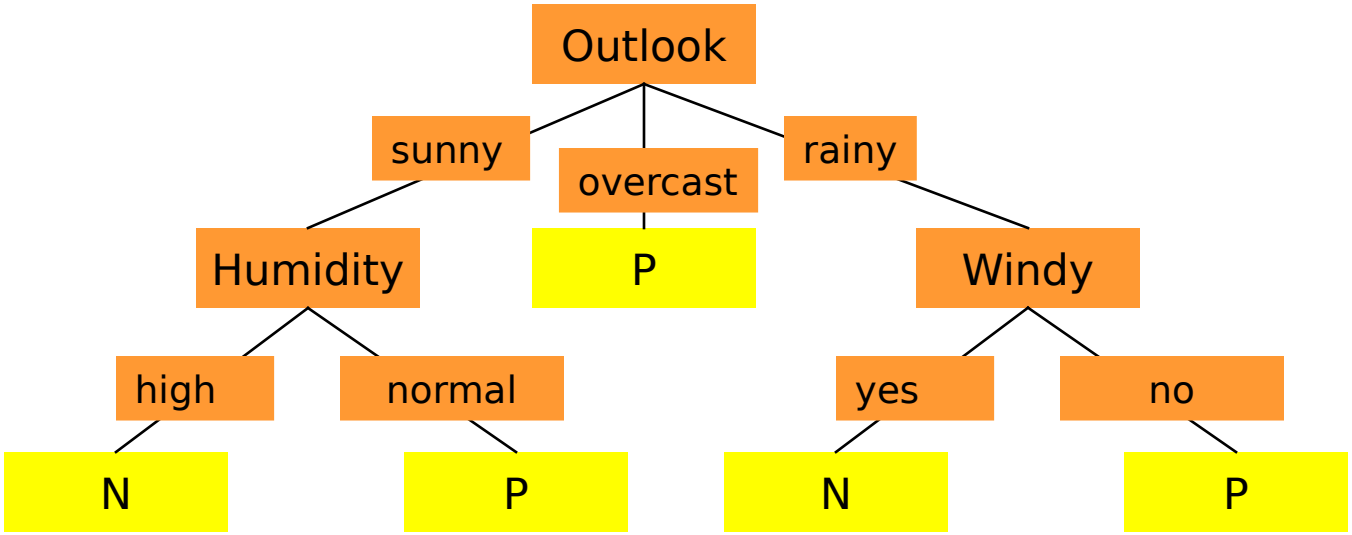
- Resulting tree T is:



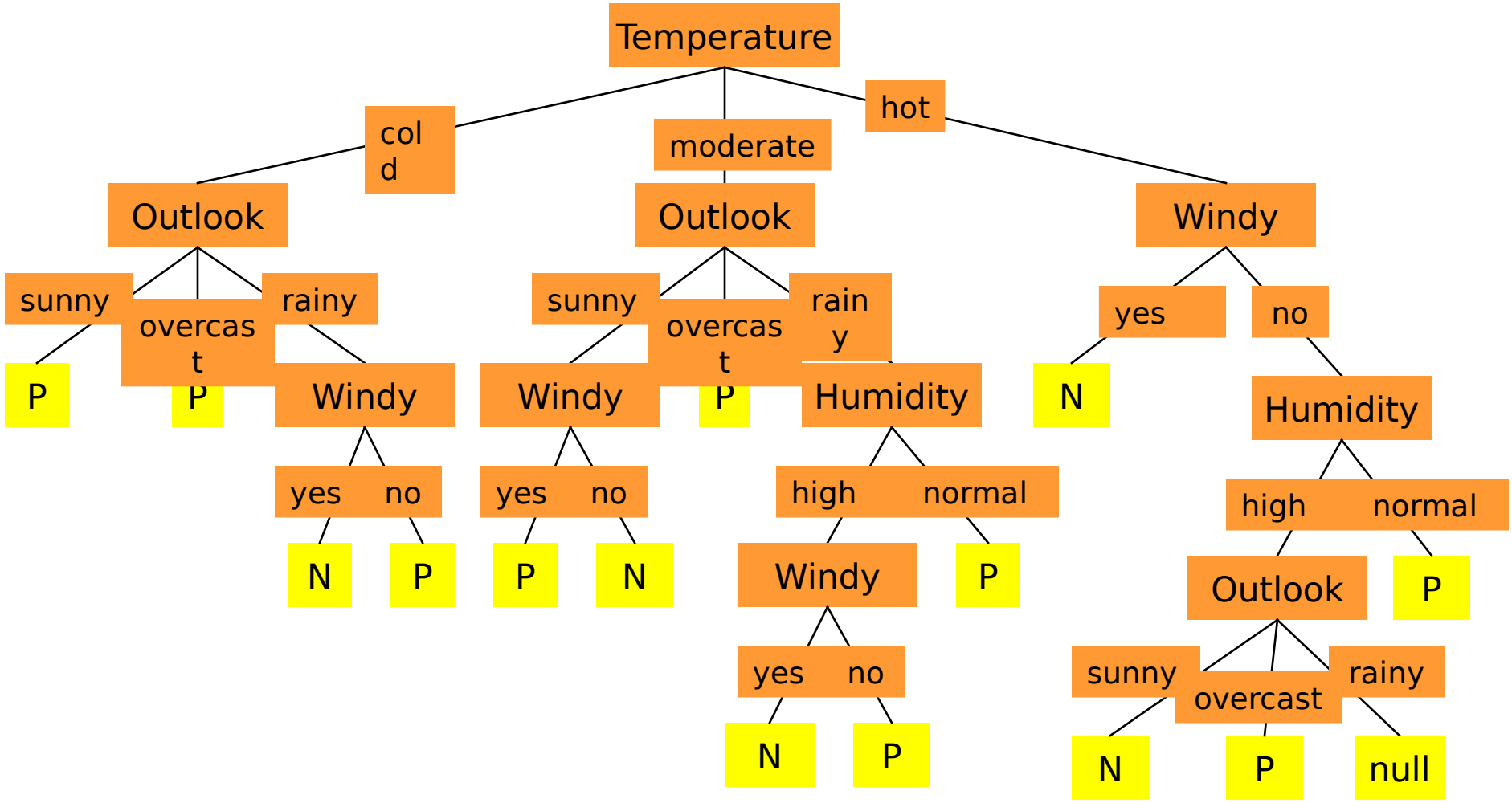
Another Example

#	Attribute				Class
	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	no	N
2	sunny	hot	high	yes	N
3	overcast	hot	high	no	P
4	rainy	moderate	high	no	P
5	rainy	cold	normal	no	P
6	rainy	cold	normal	yes	N
7	overcast	cold	normal	yes	P
8	sunny	moderate	high	no	N
9	sunny	cold	normal	no	P
10	rainy	moderate	normal	no	P
11	sunny	moderate	normal	yes	P
12	overcast	moderate	high	yes	P
13	overcast	hot	normal	no	P
14	rainy	moderate	high	yes	N

Simple Tree


Outlook
sunny
overcast
rainy
Humidity
P
Windy
high
normal
yes
no
N
P
N
P

Complicated Tree



Attribute Selection Criteria

- Main principle
  - Select attribute which partitions the learning set into subsets as “pure” as possible
- Various measures of purity
  - Information-theoretic
  - Gini index
  - ….

Information-Theoretic Approach

- To classify an object, a certain information is needed
  - I, information
- After we have learned the value of attribute A, we only need some remaining amount of information to classify the object
  - I_{res}, residual information
- Gain $Gain(A) = I - I_{res}(A)$

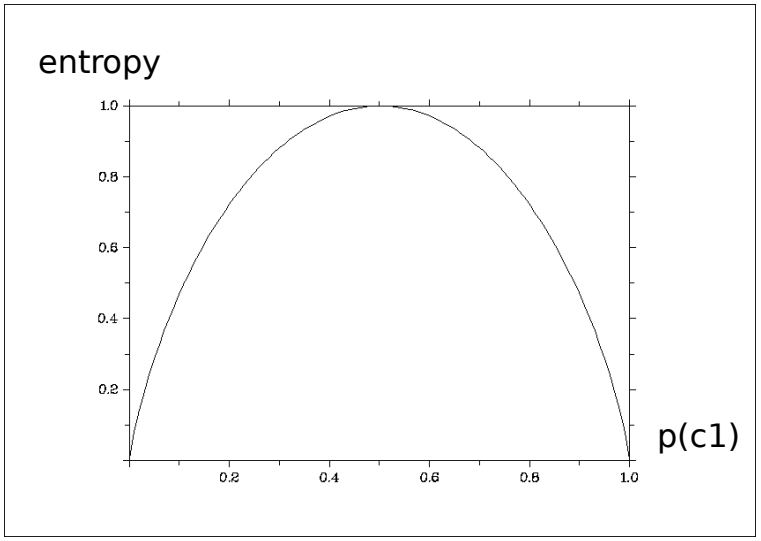
The most ‘informative’ attribute is the one that minimizes I_{res}, *i.e.*, maximizes Gain

Entropy

- The average amount of information *I* needed to classify an object is given by the entropy measure

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

- For a two-class problem:



Residual Information

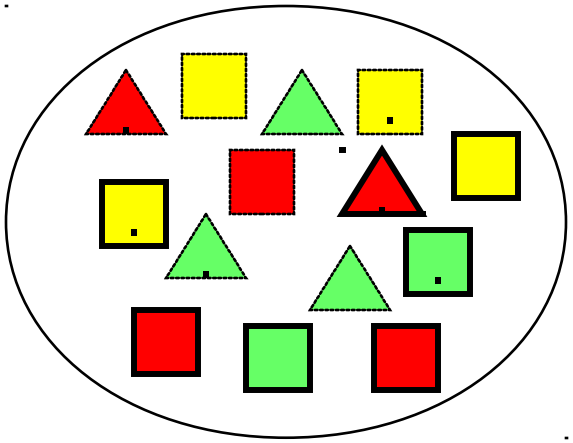
- After applying attribute *A*, *S* is partitioned into subsets according to values *v* of *A*
- I_{res} is equal to weighted sum of the amounts of information for the subsets

$$I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

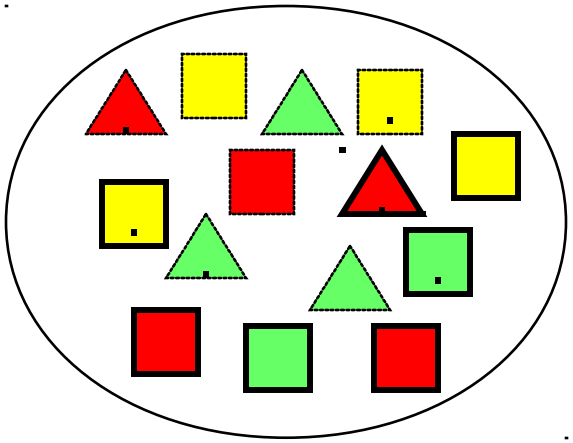
Triangles and Squares

#	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



Entropy



- 5 triangles
- 9 squares
- class probabilities

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

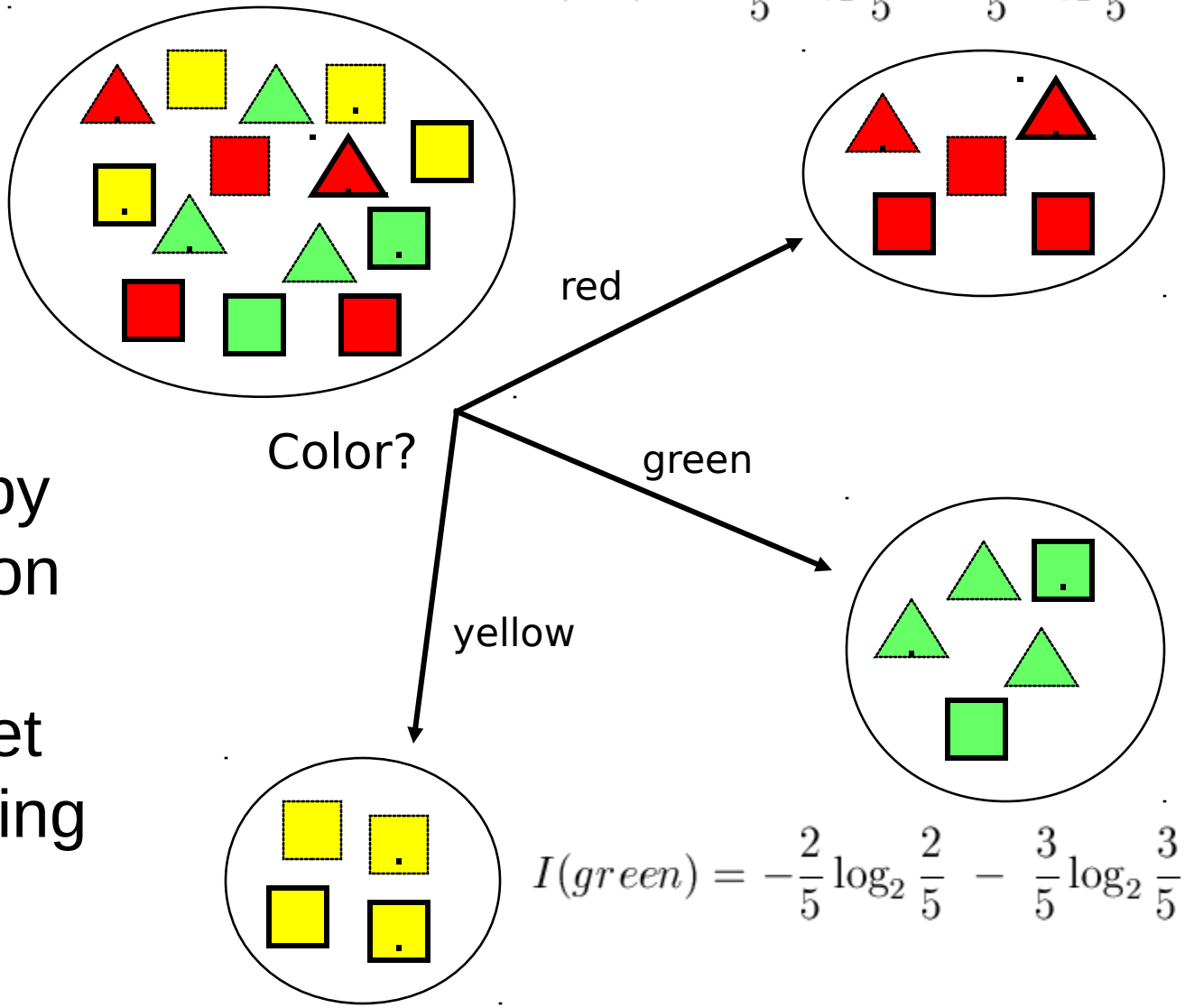
$$p(\triangle) = \frac{5}{14}$$

- entropy

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

Entropy reduction by data set partitioning

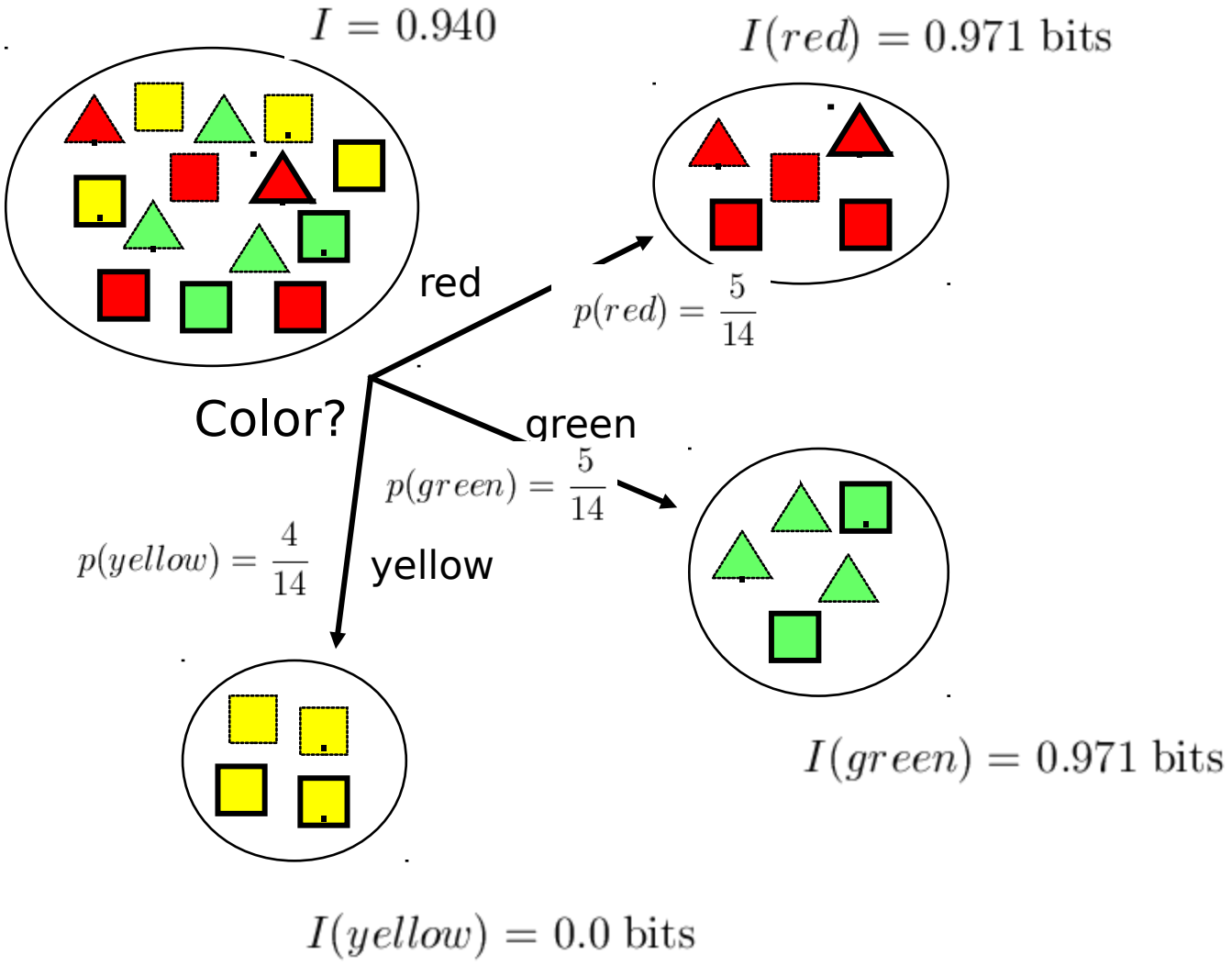
$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$



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

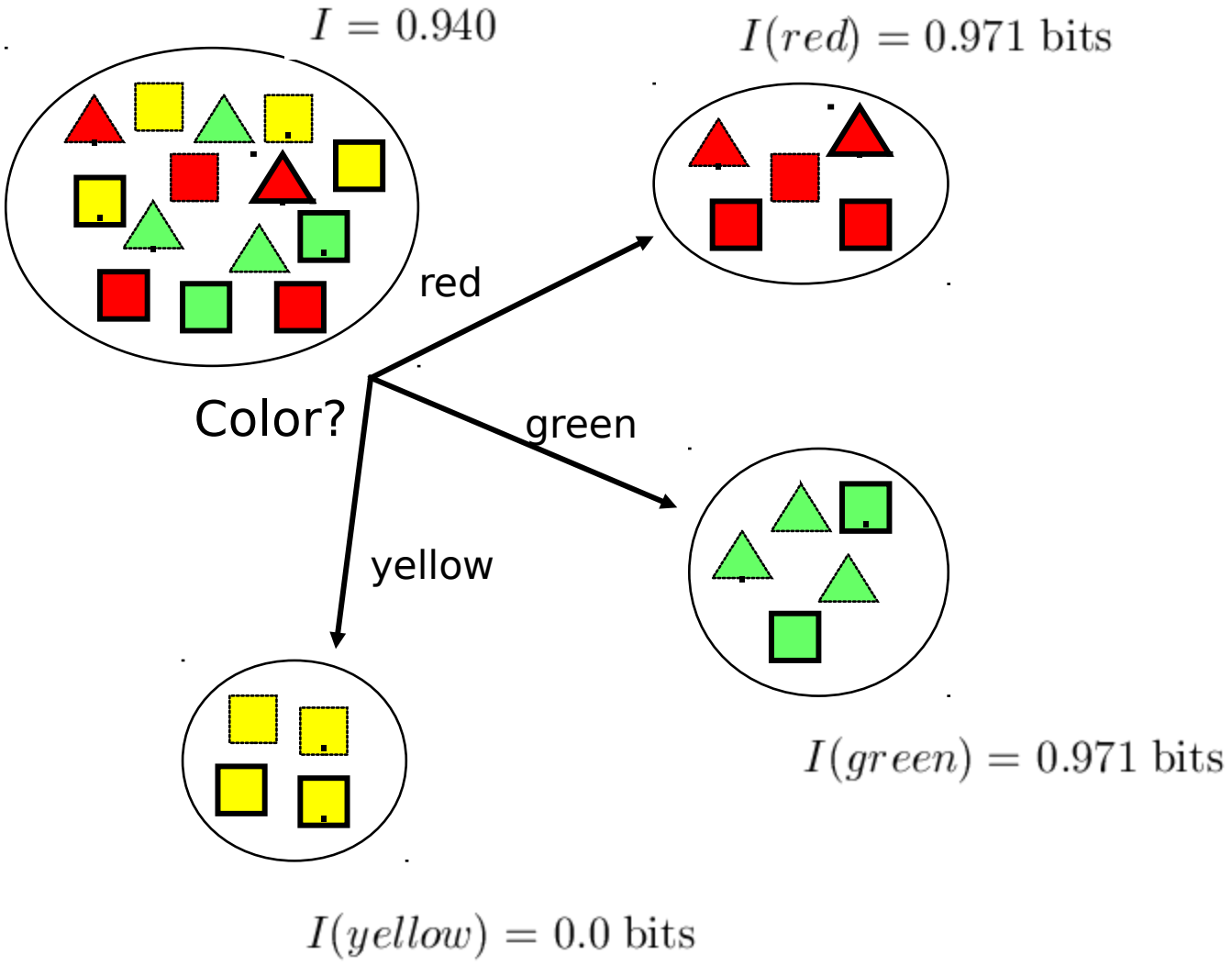
$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$

Entropija vrednosti atributa



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

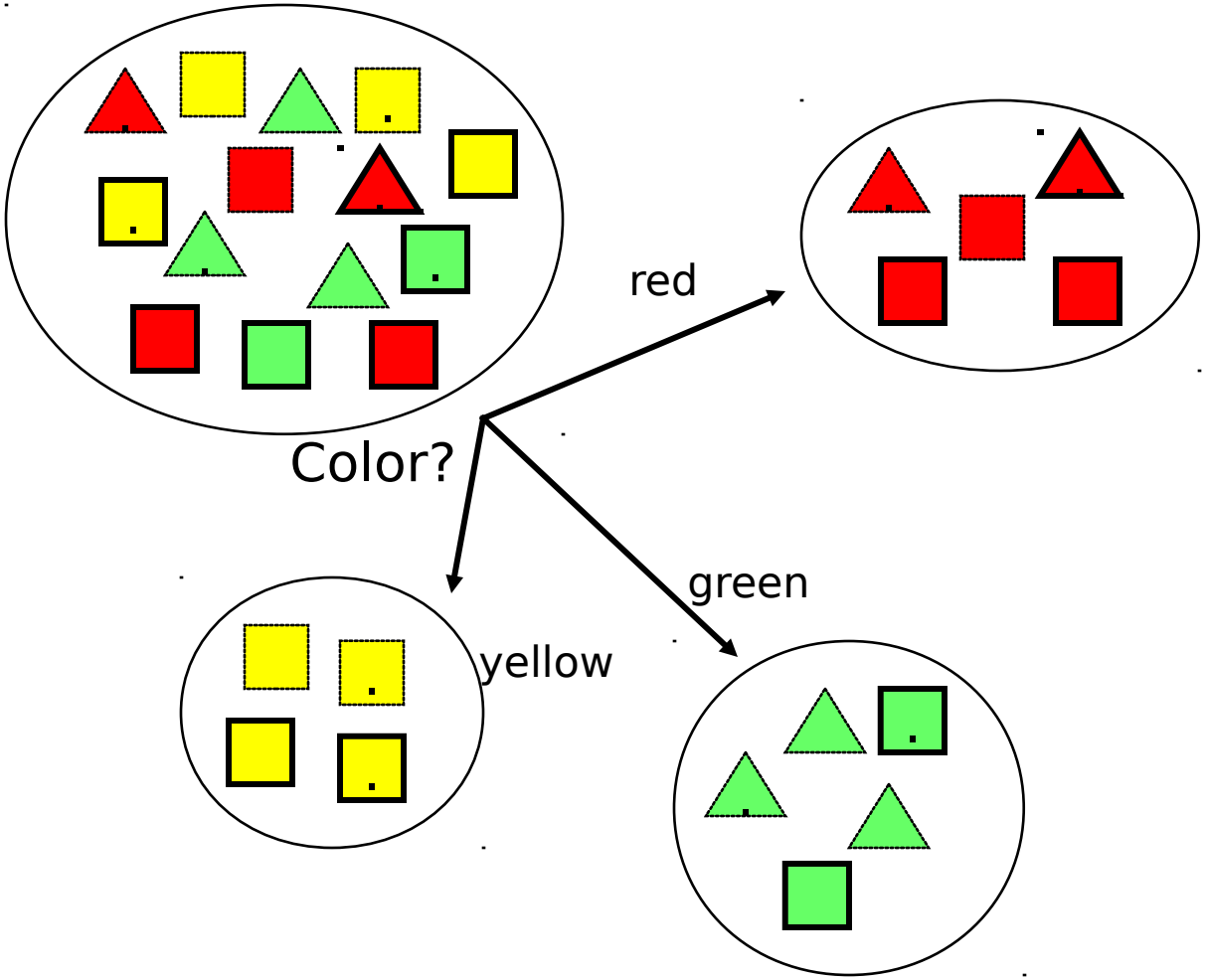
Information Gain



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

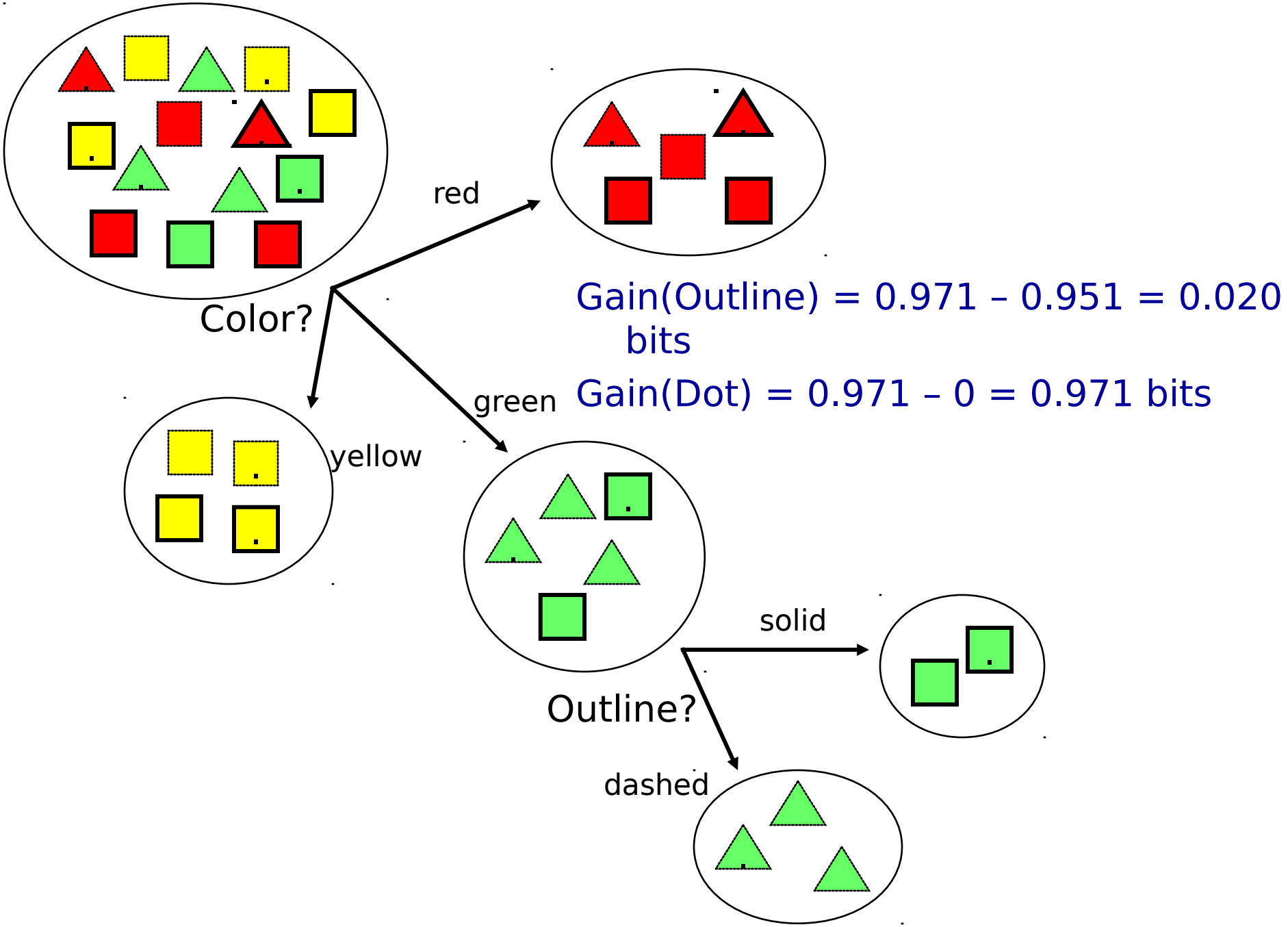
Information Gain of The Attribute

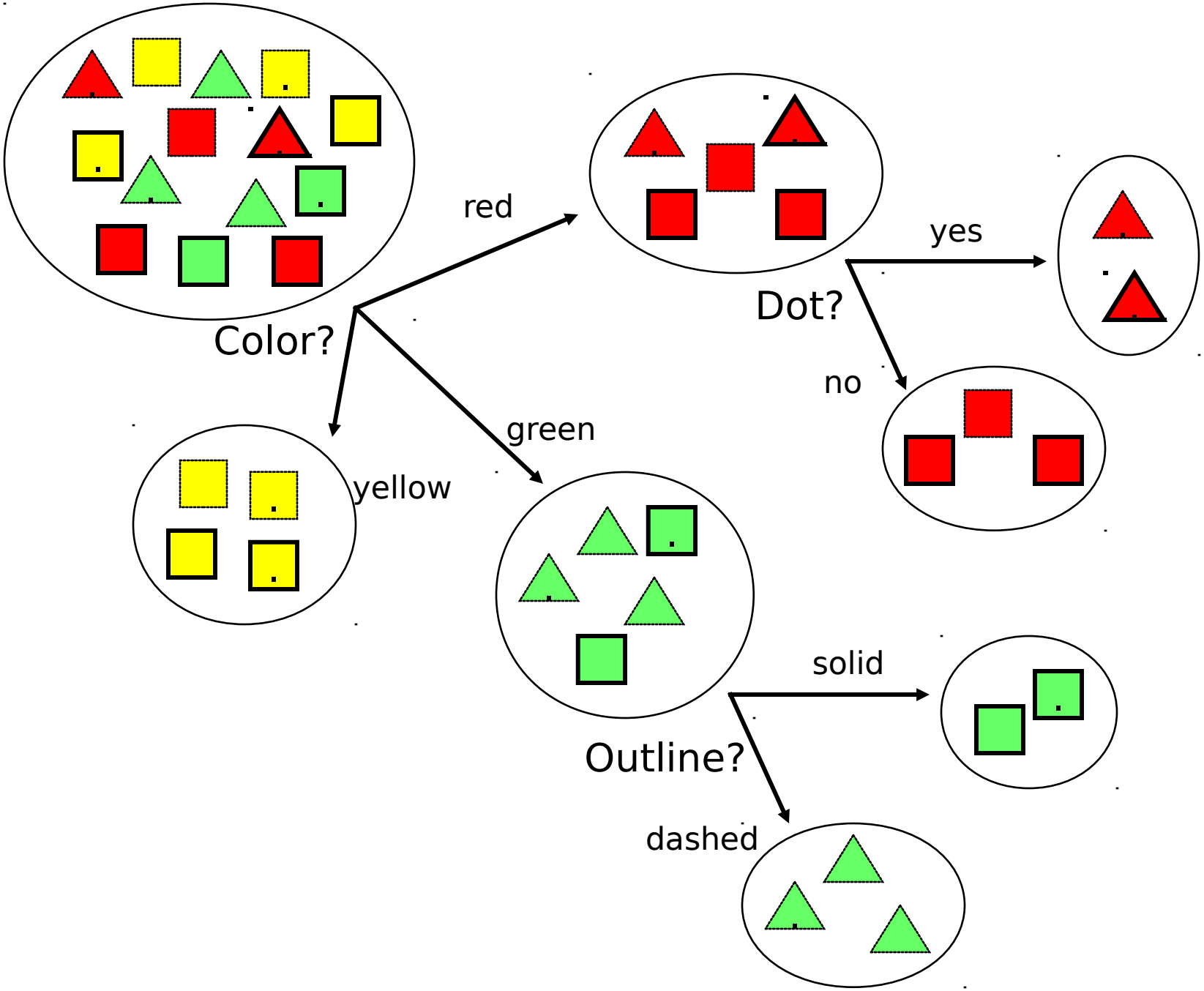
- Attributes
  - Gain(Color) = 0.246
  - Gain(Outline) = 0.151
  - Gain(Dot) = 0.048
- Heuristics: attribute with the highest gain is chosen
- This heuristics is local (local minimization of impurity)



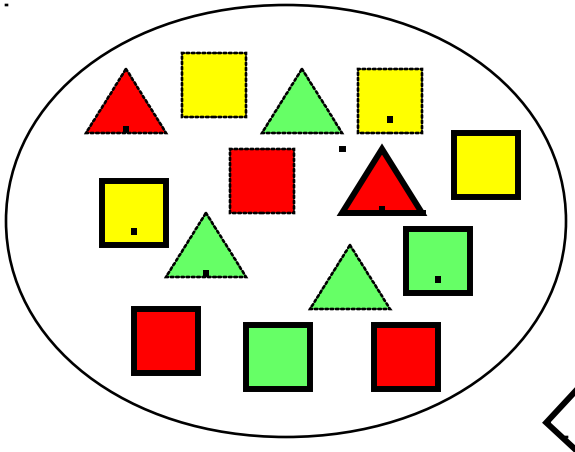
Gain(Outline) = 0.971 – 0 = 0.971 bits

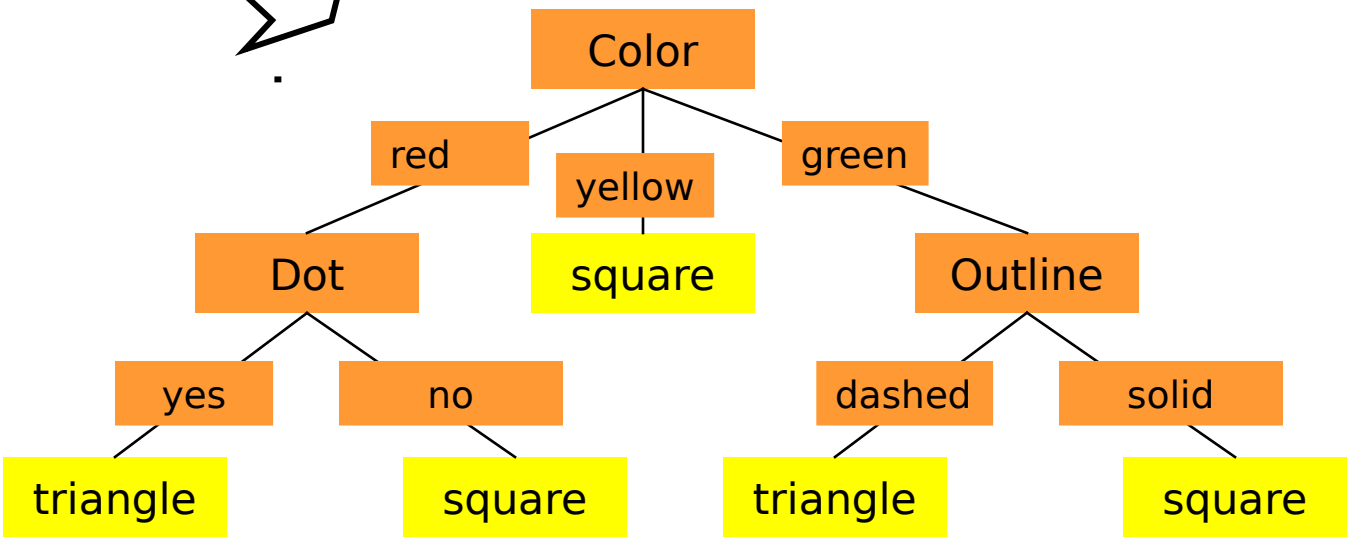
Gain(Dot) = 0.971 – 0.951 = 0.020 bits


red
Color?
green
yellow
Gain(Outline) = 0.971 – 0.951 = 0.020 bits
Gain(Dot) = 0.971 – 0 = 0.971 bits
solid
Outline?
dashed


Color?
red
yellow
green
Dot?
yes
no
Outline?
solid
dashed

Decision Tree





Decision Trees

- Start from univariate decision trees
  - Each node looks only at single input feature
- Want smaller decision trees
  - Less memory for representation
  - Less computation for a new instance
- Want smaller generalization error

Decision and Leaf Node

- Implement simple test function $f_m(x)$
- Output: labels of branches
- $f_m(x)$ discriminant in d-dimensional space
- Complex discriminant is broken down into hierarchy of simple decisions
- Leaf node describes a region in d-dimensional space with same value
  - Classification label
  - Regression value

Classification Trees

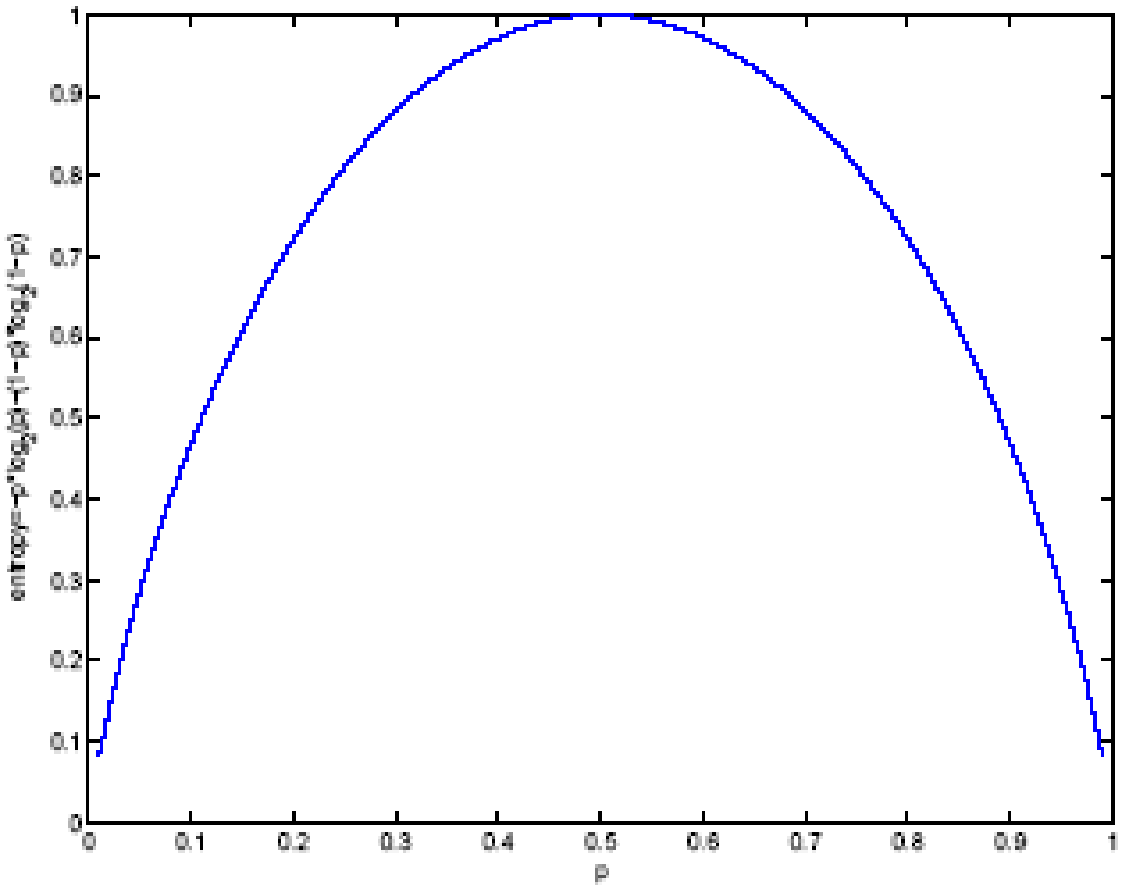
- What is the good split function?
- Use Impurity measure
- Assume N_m training samples reach node m
- N_m^i of N_m belong to class C_i, with $\sum_i N_m^i = N_m$.
- $$\hat{P}(C_i|\mathbf{x}, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$
- Node m is pure if for all classes either 0 or 1

Entropy

- Measure amount of uncertainty on a scale from 0 to 1
- Example: 2 events
- If p1=p2=0.5, entropy is 1 which is maximum uncertainty
- If p1=1=1-p0, entropy is 0 , which is no uncertainty

$$\mathcal{I}_m = -\sum_{i=1}^{K} p_m^i \log_2 p_m^i$$

Entropy



Best Split

- Node is impure, need to split more
- Have several split criteria (coordinates), have to choose optimal
- Minimize impurity (uncertainty) after split
- Stop when impurity is small enough
  - Zero stop impurity=>complex tree with large variance
  - Larger stop impurity=>small tress but

Best Split

- Impurity after split: N_{mj} of N_m take branch j.
- N^i_{mj} belong to C_i

$$\hat{P}(C_i \mid \boldsymbol{x}, m, j) \equiv p^i_{mj} = \frac{N^i_{mj}}{N_{mj}}$$

$$I'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p^i_{mj} \log_2 p^i_{mj}$$

- Find the variable and split that min impurity
  - among all variables
  - split positions for numeric variables

```
GenerateTree(X)
    If NodeEntropy(X)< θ_I /* eq. 9.3
        Create leaf labelled by majority class in X
        Return
    i ← SplitAttribute(X)
    For each branch of x_i
        Find X_i falling in branch
        GenerateTree(X_i)
SplitAttribute(X)
    MinEnt← MAX
    For all attributes i = 1,...,d
        If x_i is discrete with n values
            Split X into X_1,...,X_n by x_i
            e ← SplitEntropy(X_1,...,X_n) /* eq. 9.8 */
            If e<MinEnt MinEnt ← e; bestf ← i
        Else /* x_i is numeric */
            For all possible splits
                Split X into X_1, X_2 on x_i
                e←SplitEntropy(X_1, X_2)
                If e<MinEnt MinEnt ← e; bestf ← i
    Return bestf
```