Introduction to Machine Learning

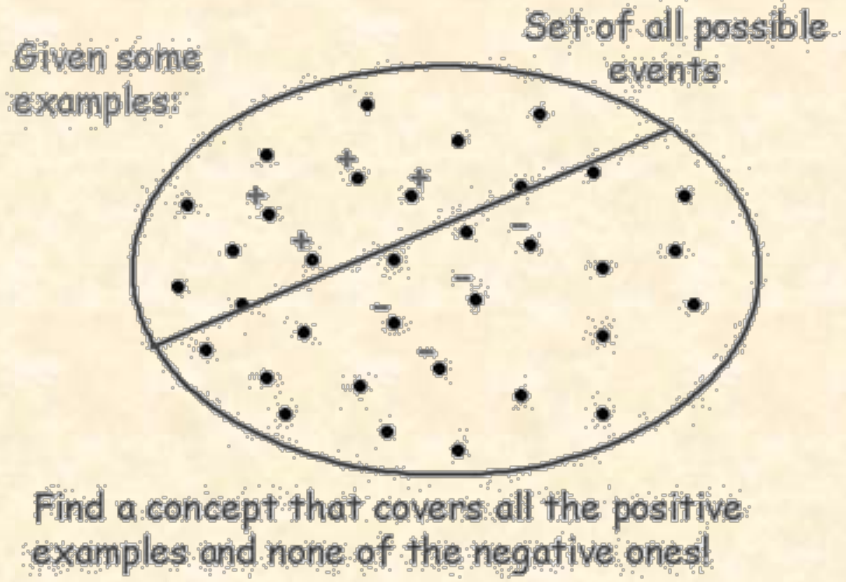
Decision Trees

Inas A. Yassine
Systems and Biomedical Engineering Department,
Faculty of Engineering - Cairo University
iyassine@eng.cu.edu.eg

Concept Learning

Concept Learning as Search

- Concept Learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.
- Selecting a Hypothesis Representation is an important step since it restricts (or *biases*) the space that can be searched. [For example, the hypothesis "If the air temperature is cold **or** the humidity high then it is a good day for water sports" cannot be expressed in our chosen representation.]



Find- S algorithm

- Determine the maximally specific hypothesis
    - Start form a very specific hypothesis and begin to relax it
    - Start form a very general hypothesis and begin to specify it

Find-S, a Maximally Specific Hypothesis Learning Algorithm

- Initialize h to the most specific hypothesis in H
- For each <u>positive</u> training instance x
  - For each attribute constraint a_i in h

    If the constraint a_i is satisfied by x

    then do nothing

    else replace a_i in h by the next more general constraint that is satisfied by x
- Output hypothesis h

Example for the Find-S Algorithm

- Initially:
  - S_0=<0,0,0,0,0,0>
- X_1^+=< Sunny, Warm, Normal, Strong, Warm, Same >
  - S_1= < Sunny, Warm, Normal, Strong, Warm, Same >
- X_2^+=< Sunny, Warm, High, Strong, Warm, Same>
  - S_2= < Sunny, Warm, ?, Strong, Warm, Same >
- X_3^-=< Rainy, Cold, High, Strong, Warm, Change >
  - S_3= < Sunny, Warm, ?, Strong, Warm, Same >
- X_4^+=< Sunny, Warm, High, Strong, Cool, Change >
  - S4= < Sunny, Warm, ?, Strong, ?,? >

Shortcomings of Find-S

- Although Find-S finds a hypothesis consistent with the training data, it does not indicate whether that is the only one available
- Is it a good strategy to prefer the most specific hypothesis?
- What if the training set is inconsistent (*noisy*)?
- What if there are several maximally specific consistent hypotheses? Find-S cannot backtrack!

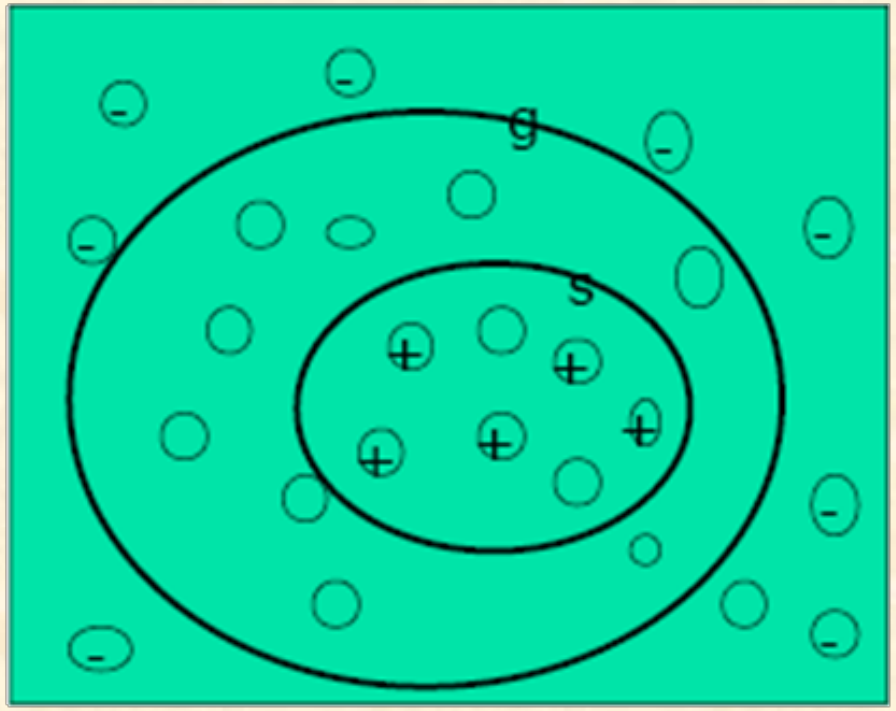
Candidate-Elimination Learning Algorithm

- The candidate-Elimination algorithm computes the version space containing all (and only those) hypotheses from *H* that are consistent with an observed sequence of training examples.

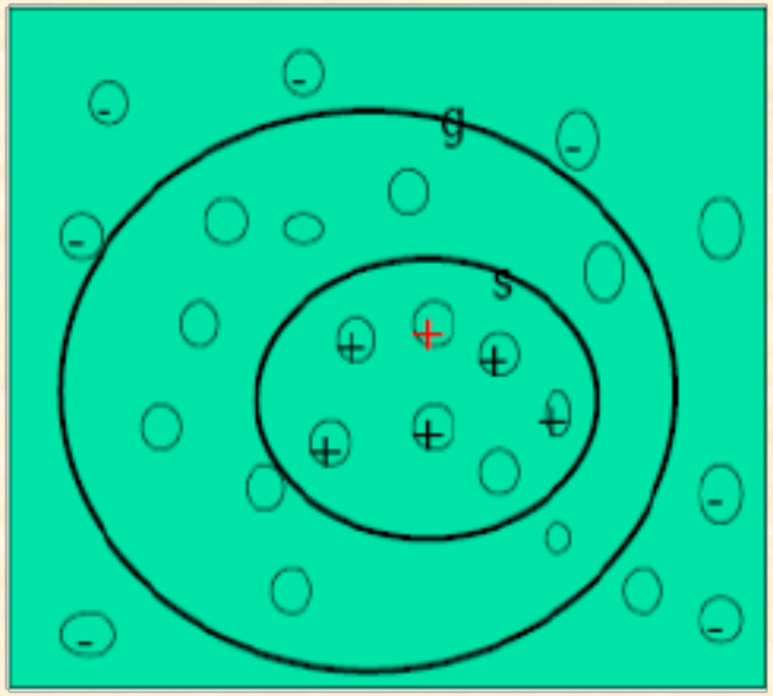
Basic Ideas of Candidate Elimination Algorithm

- Initialize G to the set of maximally general hypotheses in H
- Initialize S to the set of maximally specific hypotheses in H
- For each training example d=<x,c(x)> modify G and S so that G and S are consistent with d

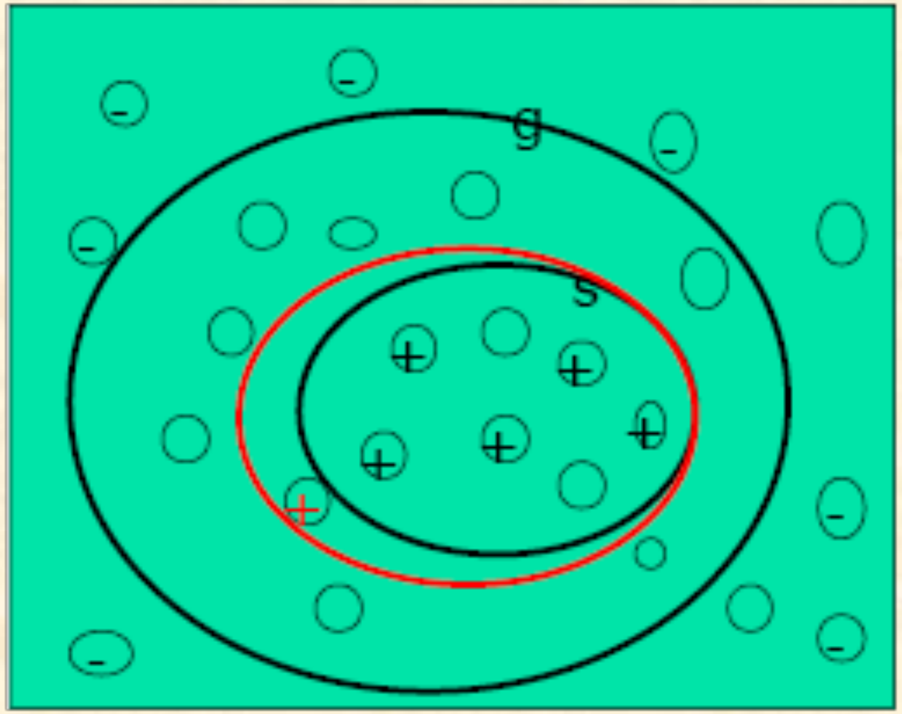
Specific and General Boundaries



Occurrence of Positive Example

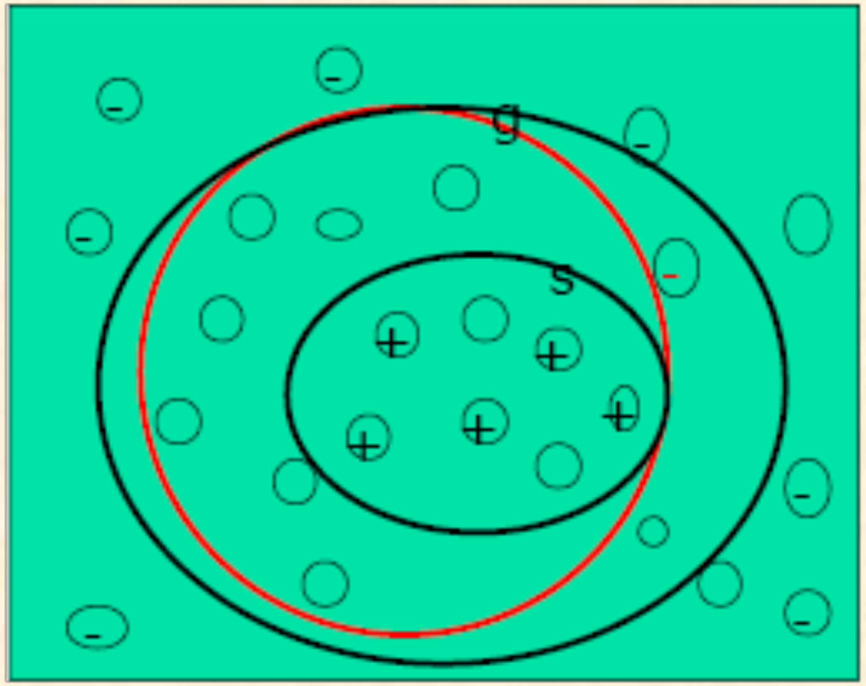


Occurrence of Positive Example



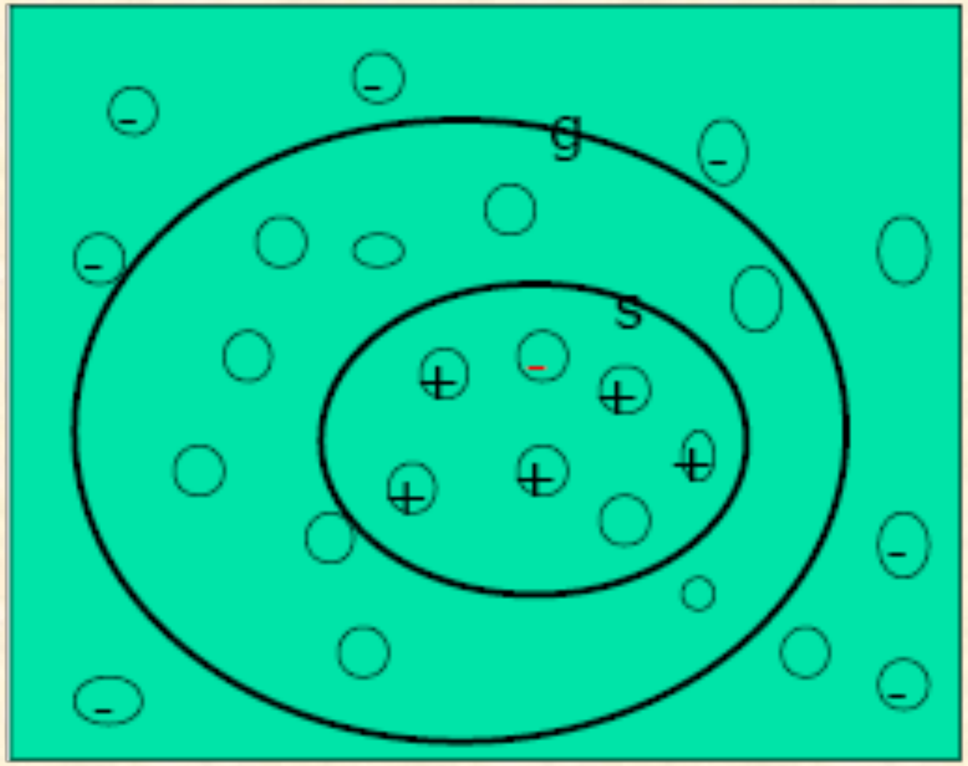
- generalize s

Occurrence of Negative Example



- specialize g

Occurrence of Negative Example



- remove s
- remove g

Candidate Elimination Algorithm

- Initialization:
    - G<- Maximally General Hypotheses in H
    - S<- Maximally specific hypotheses in H
- Learning:
    - For each training example d, do:
        - If d is a positive example:
            - Remove from G any hypothesis with d
            - For each hypothesis s in S the is not consistent with d
                - Remove s from S
                - Add to S all minimal generalizations h of s such that
                    - h is consistent with d, and
                    - Some member of G is more general than h
                - Remove from S any hypothesis that is more general than another hypothesis in S
        - If d is negative example:
            - Remove from S any hypothesis inconsistent with d
            - For each hypothesis g in G that is not consistent with d
                - Remove g from G
                - Add to G all minimal specializations h of g such that
                    - h is consistent with d, and
                    - some members of S is more specific than h
                - Remove from G any hypothesis that is less general than another hypothesis in G

Remarks on Candidate-Elimination

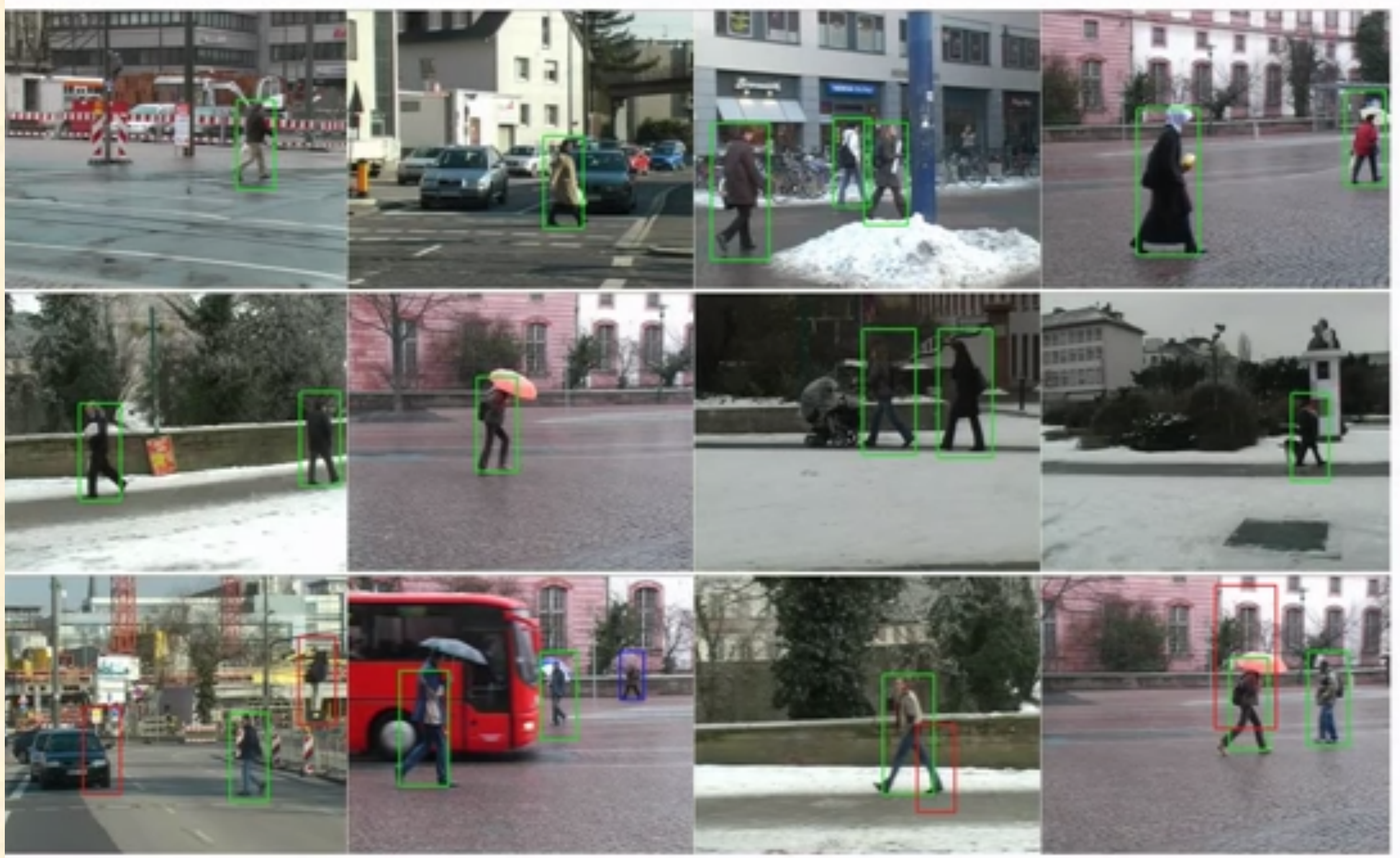
- The Candidate-Elimination Algorithm will converge toward the hypothesis that correctly describes the target concept provided: (1) There are no errors in the training examples; (2) There is some hypothesis in *H* that correctly describes the target concept.
- Convergence can be speeded up by presenting the data in a strategic order. The best examples are those that satisfy exactly half of the hypotheses in the current version space.
- Version-Spaces can be used to assign certainty scores to the classification of new examples

Example for the Candidate Elimination Algorithm

- Initially:
  - S_0=<0,0,0,0,0,0>
  - G_0=<?,?,?,?,?,?,?>
- X_1^+=< Sunny, Warm, Normal, Strong, Warm, Same >
  - S_1= < Sunny, Warm, Normal, Strong, Warm, Same >
  - G_1=<?,?,?,?,?,?,?>
- X_2^+=< Sunny, Warm, High, Strong, Warm, Same>
  - S_2= < Sunny, Warm, ?, Strong, Warm, Same >
  - G_2=<?,?,?,?,?,?,?>
- X_3^-=< Rainy, Cold, High, Strong, Warm, Change >
  - S_3= < Sunny, Warm, ?, Strong, Warm, Same >
  - G_3={<Sunny,?,?,?,?,?>, <?, Warm,?,?,?,?>,<?,?,?,?,?, Same>}
- X_4^+=< Sunny, Warm, High, Strong, Cool, Change >
  - S4= < Sunny, Warm, ?, Strong, ?,? >
  - G4={<Sunny,?,?,?,?,?>, <?, Warm,?,?,?,?>}

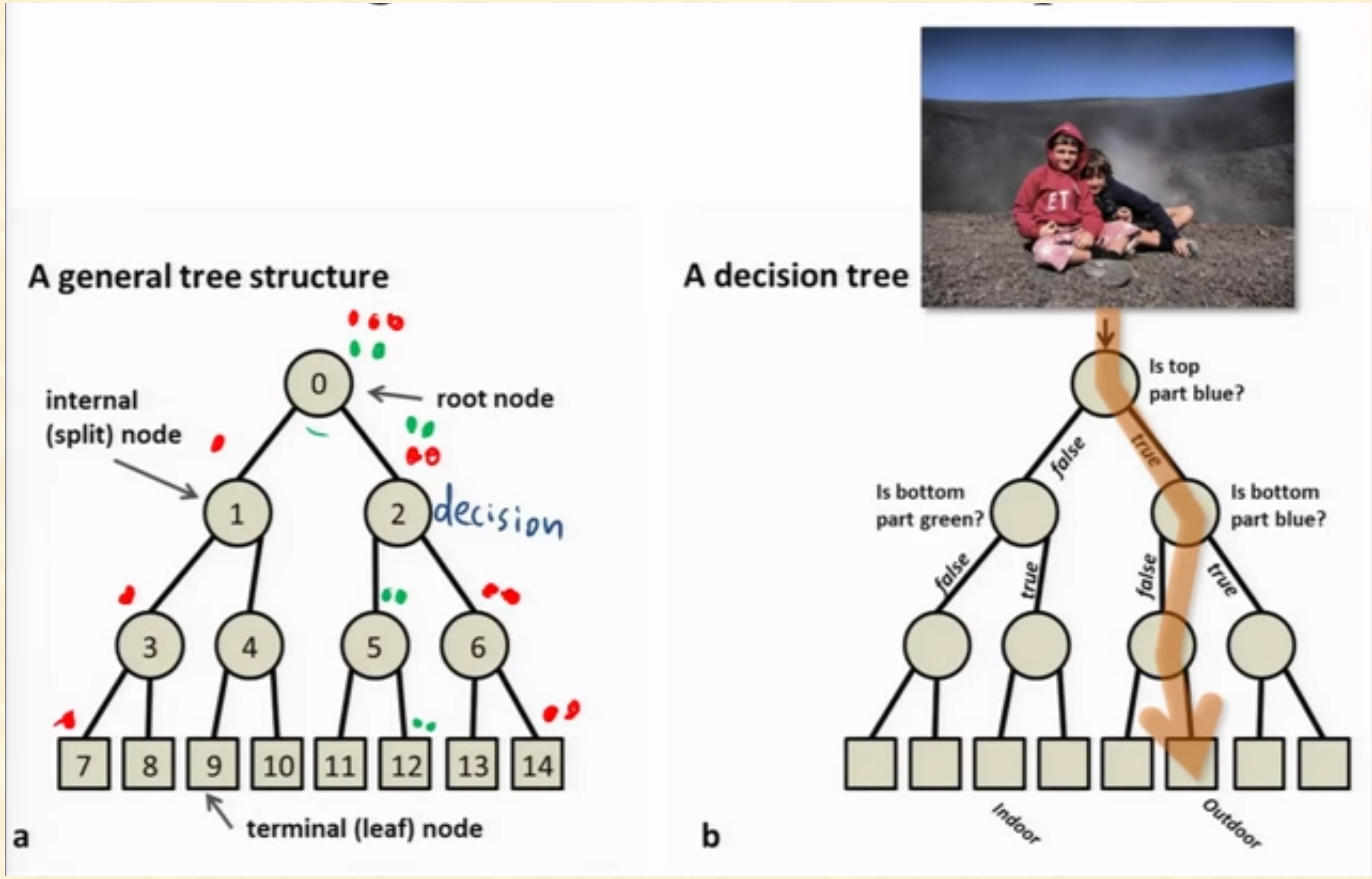
Decision Trees

Application 1: Object Detection

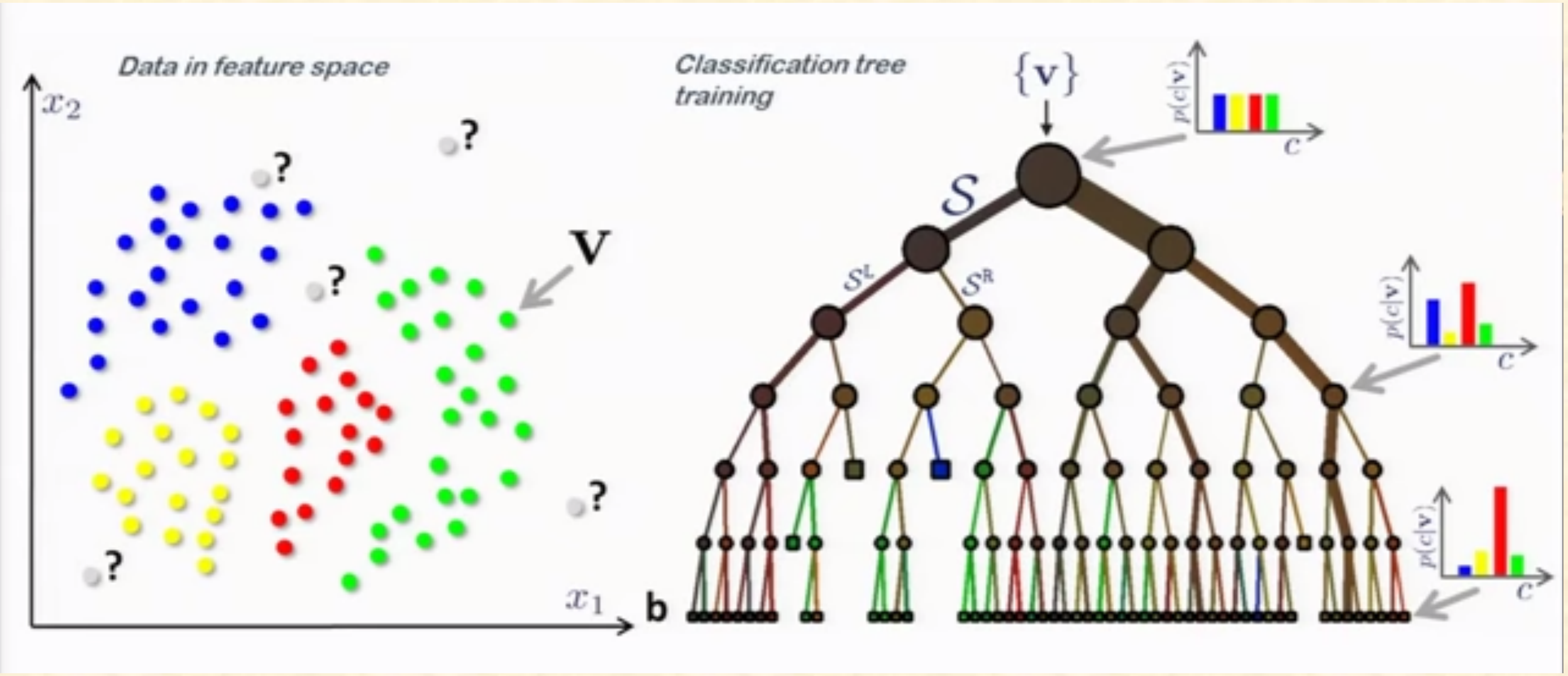


Application 2: Kinect

Image Classification



Classification Tree



Let’s see another typical machine learning dataset

48,000 records, 16 attributes [Kohavi 1995]

age	employme	education	edun	marital	…	job	relation	race	gender	hour	country	wealth
					…							
39	State_gov	Bachelors	13	Never_mar	…	Adm_cleri	Not_in_fam	White	Male	40	United_Sta	poor
51	Self_emp_	Bachelors	13	Married	…	Exec_man	Husband	White	Male	13	United_Sta	poor
39	Private	HS_grad	9	Divorced	…	Handlers_	Not_in_fam	White	Male	40	United_Sta	poor
54	Private	11th	7	Married	…	Handlers_	Husband	Black	Male	40	United_Sta	poor
28	Private	Bachelors	13	Married	…	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	…	Exec_man	Wife	White	Female	40	United_Sta	poor
50	Private	9th	5	Married_sp	…	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp_	HS_grad	9	Married	…	Exec_man	Husband	White	Male	45	United_Sta	rich
31	Private	Masters	14	Never_mar	…	Prof_speci	Not_in_fam	White	Female	50	United_Sta	rich
42	Private	Bachelors	13	Married	…	Exec_man	Husband	White	Male	40	United_Sta	rich
37	Private	Some_coll	10	Married	…	Exec_man	Husband	Black	Male	80	United_Sta	rich
30	State_gov	Bachelors	13	Married	…	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	…	Adm_cleri	Own_child	White	Female	30	United_Sta	poor
33	Private	Assoc_acd	12	Never_mar	…	Sales	Not_in_fam	Black	Male	50	United_Sta	poor
41	Private	Assoc_voc	11	Married	…	Craft_repai	Husband	Asian	Male	40	*MissingV	rich
34	Private	7th_8th	4	Married	…	Transport_	Husband	Amer_Indi	Male	45	Mexico	poor
26	Self_emp_	HS_grad	9	Never_mar	…	Farming_fi	Own_child	White	Male	35	United_Sta	poor
33	Private	HS_grad	9	Never_mar	…	Machine_c	Unmarried	White	Male	40	United_Sta	poor
38	Private	11th	7	Married	…	Sales	Husband	White	Male	50	United_Sta	poor
44	Self_emp_	Masters	14	Divorced	…	Exec_man	Unmarried	White	Female	45	United_Sta	rich
41	Private	Doctorate	16	Married	…	Prof_speci	Husband	White	Male	60	United_Sta	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

What can we do with a dataset?

- Well, you can look at histograms…

Value	Frequency
Female	16192
Male	32650

Gender

Value	Frequency
Divorced	6633
Married_AF_spouse	37
Married	22379
Married_spouse_absent	628
Never_married	16117
Separated	1530
Widowed	1518

Marital Status

Contingency Tables

- A better name for a histogram:

A One-dimensional Contingency Table

- Recipe for making a k-dimensional contingency table:

1. Pick *k* attributes from your dataset. Call them $a_1, a_2, \ldots a_k$.
2. For every possible combination of values, $a_1,=x_1$, $a_2,=x_2, \ldots a_k,=x_k$,record how frequently that combination occurs

Fun fact: A database person would call this a "k-dimensional datacube"

A 2D Contingency Table

- For each pair of values for attributes (age group, wealth) we can see how many records match.

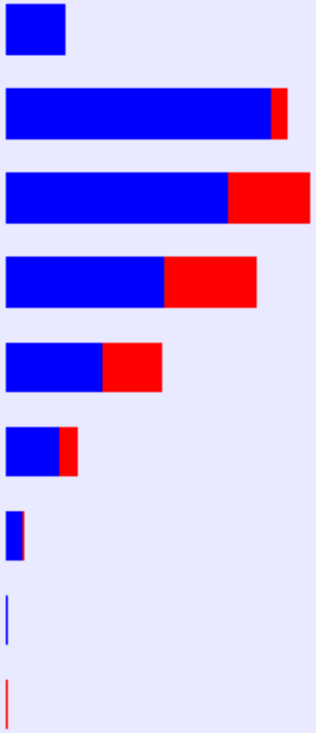
wealth values:		poor	rich
agegroup	10s	2507	3
	20s	11262	743
	30s	9468	3461
	40s	6738	3986
	50s	4110	2509
	60s	2245	809
	70s	668	147
	80s	115	16
	90s	42	13

A 2D Contingency Table

- Easier to appreciate graphically

wealth values: poor rich

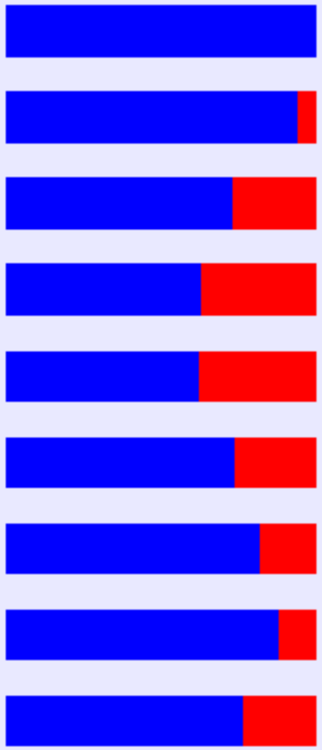
		poor	rich
agegroup	10s	2507	3
	20s	11262	743
	30s	9468	3461
	40s	6738	3986
	50s	4110	2509
	60s	2245	809
	70s	668	147
	80s	115	16
	90s	42	13



A 2D Contingency Table

- Easier to see "interesting" things if we stretch out the histogram bars

wealth values:		poor	rich
agegroup	10s	2507	3
	20s	11262	743
	30s	9468	3461
	40s	6738	3986
	50s	4110	2509
	60s	2245	809
	70s	668	147
	80s	115	16
	90s	42	13

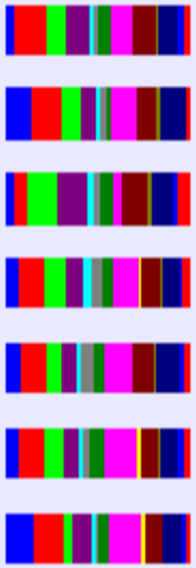


A bigger 2D contingency table

job values: Adm_clerical Craft_repair Farming_fishing Machine_op_inspct Priv_house_serv Protective_serv Tech_support

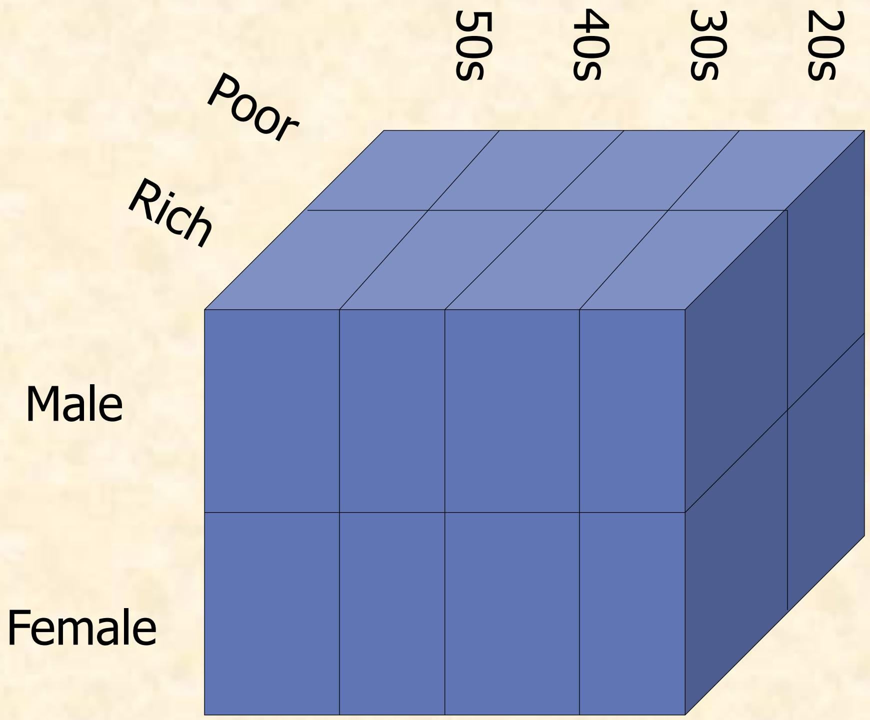
MissingValue Armed_Forces Exec_managerial Handlers_cleaners Other_service Prof_specialty Sales Transport_moving

marital	Divorced	270	1192	0	679	890	90	197	434	762	46	795	121	664	239	254
	Married_AF_spouse	5	6	0	4	3	1	1	1	5	0	4	1	5	0	1
	Married	928	1495	7	3818	3600	869	724	1469	1088	27	3182	583	2491	609	1489
	Married_spouse_absent	45	84	0	77	52	35	32	37	92	9	64	7	55	9	30
	Never_married	1242	2360	8	1301	1260	434	1029	872	2442	99	1849	237	1992	506	486
	Separated	97	224	0	160	126	23	63	123	275	21	145	23	146	48	56
	Widowed	222	250	0	73	155	38	26	86	259	40	133	11	151	35	39



3-d contingency tables

- These are harder to look at!



Data Mining

- Data Mining is all about automating the process of searching for patterns in the data.

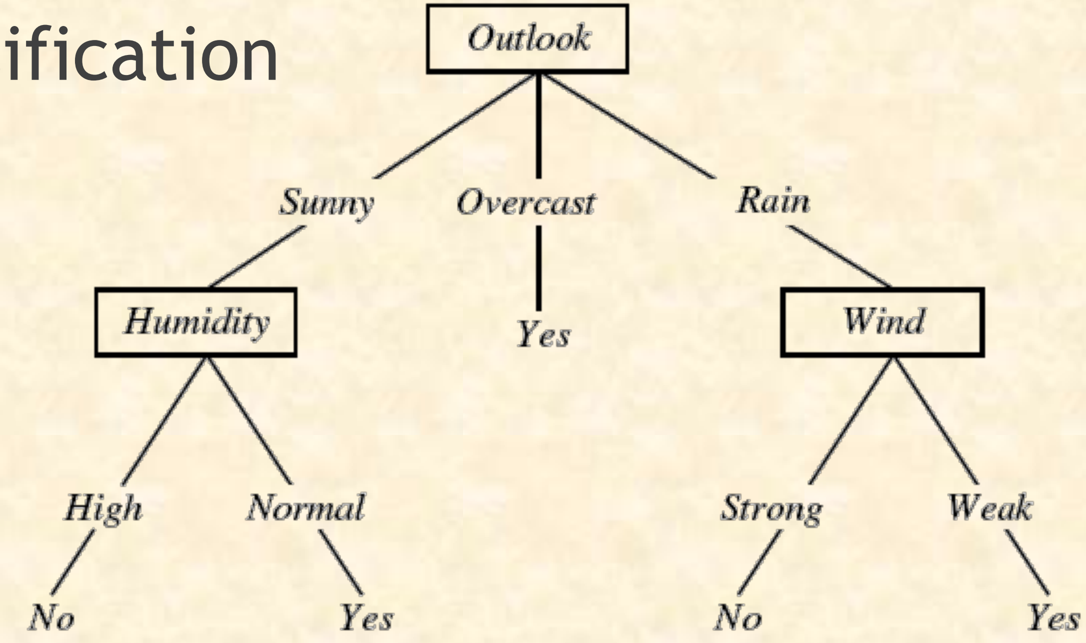
Which patterns are interesting?

Which might be mere illusions?

And how can they be exploited?

Decision Tree Representation

- Each Internal Node Tests an Attribute
- Each Branch Corresponds to Attribute value
- Each Leaf Node assigns a classification



Entropy Decision Tree Example

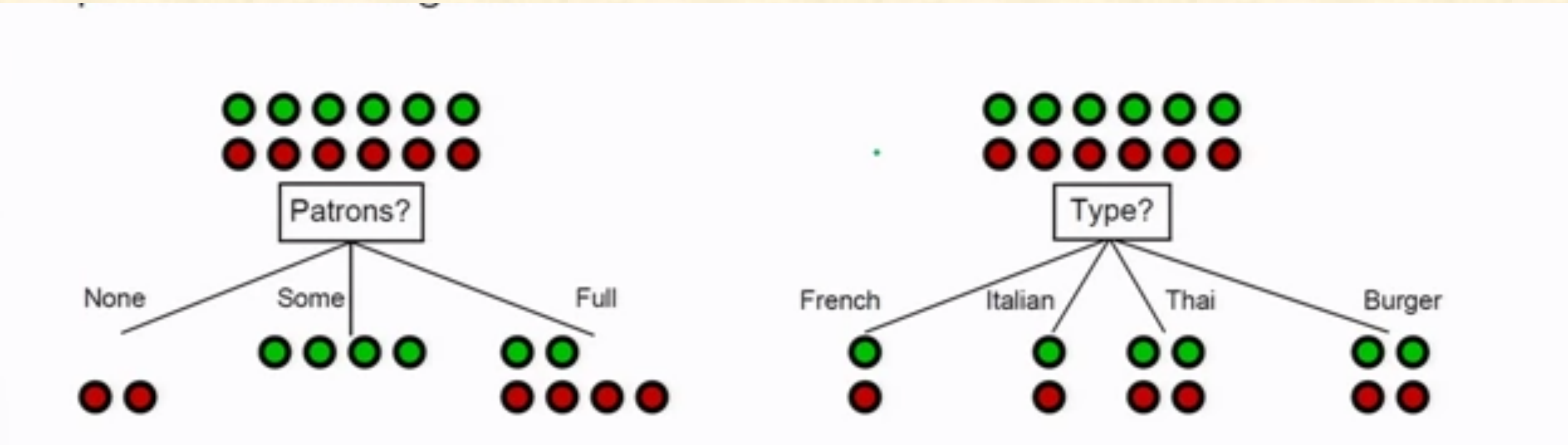
No.	Student	First last year?	Male?	Works hard?	Drinks?	First this year?
7	Matthew	no	yes	no	yes	??
8	Mary	no	no	yes	yes	??

```
              1st-last-year?
               /          \
          yes /            \ no
             /              \
    1+, 2+, 5+            drinks?
       YES                /     \
                      no /       \ yes
                        /         \
                       3+        4-, 6-
                      YES          NO
```

Data to be Classified …

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

How to Construct the tree



What Attribute to choose to "best" split a node?

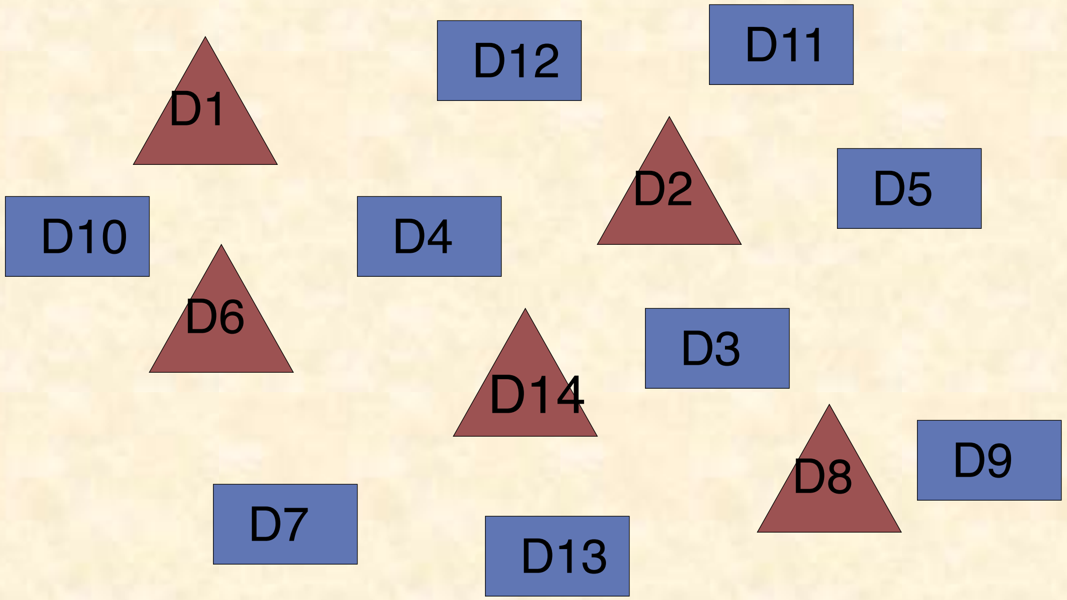
- Choose the attribute that minimize the **Disorder (or Entropy)** in the subtree rooted at a given node.
- **Disorder** and **Information** are <u>related</u> as follows: the more disorderly a set, the more information is required to correctly guess an element of that set.
- **Information:** What is the best strategy for guessing a number from a finite set of possible numbers? i.e., how many questions do you need to ask in order to know the answer (we are looking for the <u>minimal</u> number of questions). Answer Log_2(S), where S is the set of numbers and lSl, its cardinality.

E.g.: 0 1 2 3 | 4 5 6 | 7 8 9 10

Q2 Q1

Q1: is it smaller than 5?
Q2: is it smaller than 2?

ID3: The Basic Decision Tree Learning Algorithm

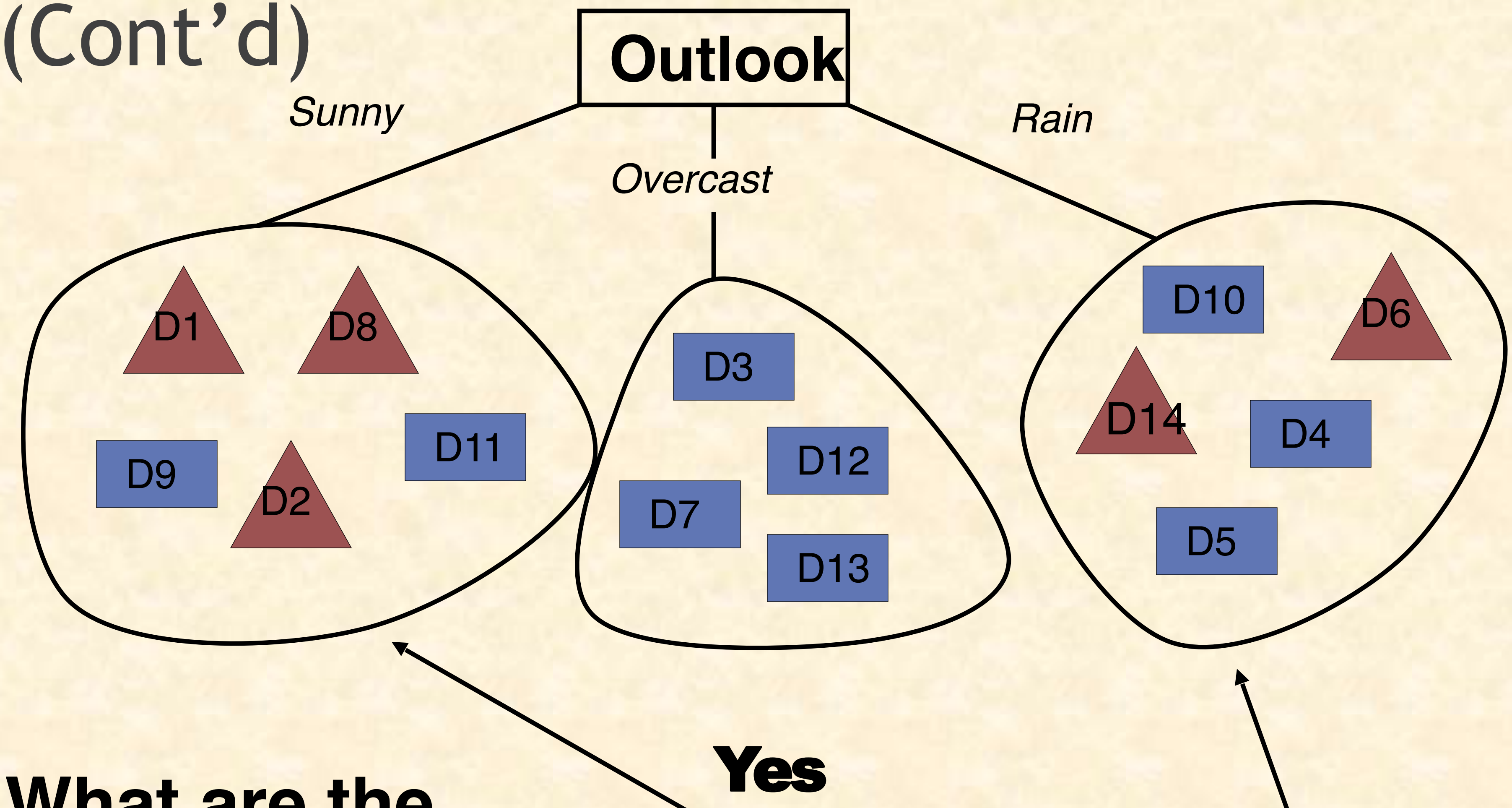


What is the "best" attribute?

<u>Answer:</u> Outlook

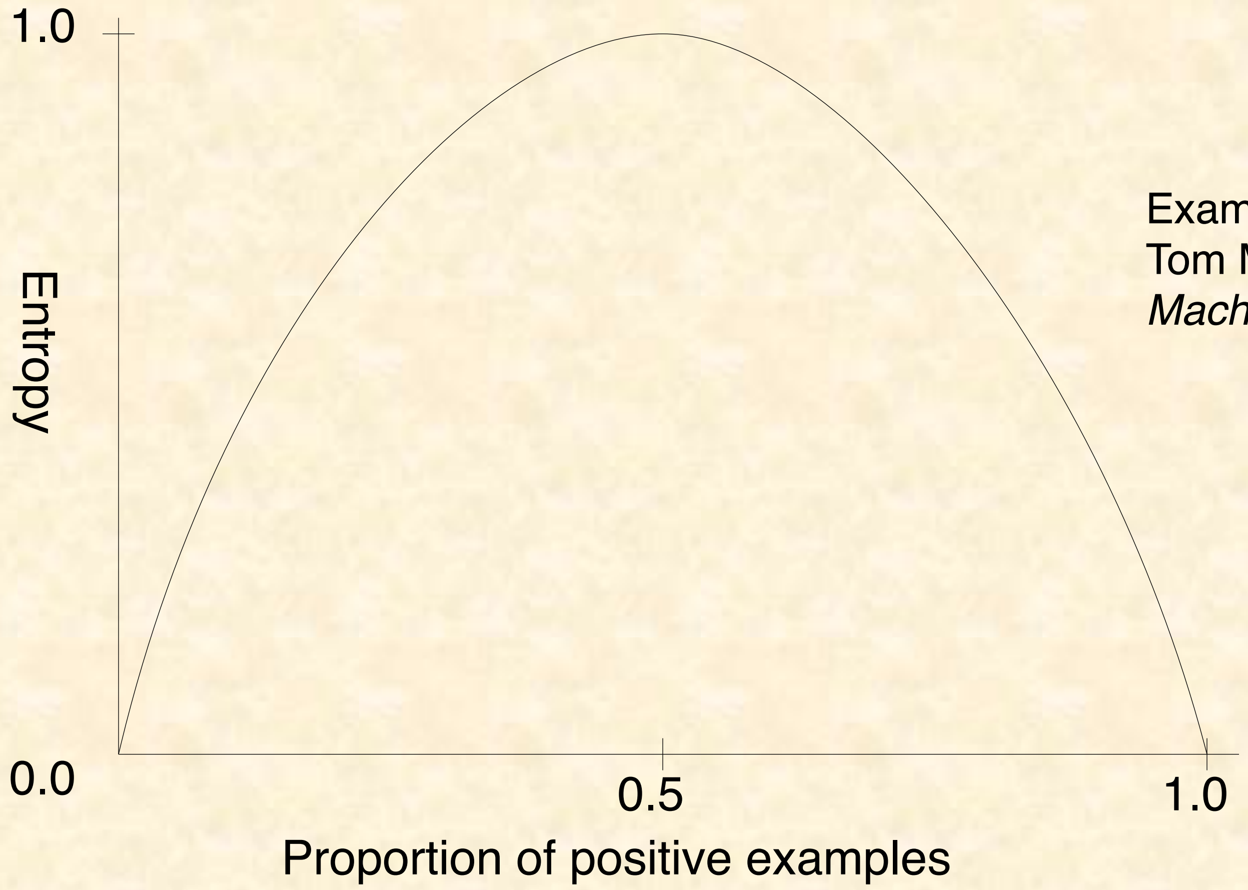
["best" = with highest information gain]

ID3 (Cont'd)



What are the "best" attributes? Humidity and Wind

The Entropy Function Relative to Boolean Classification



Example taken from
Tom Mitchell's
Machine Learning

Entropy Based selection

$$Entropy(decision) = P_+ \log_2 P_+ + P_- \log_2 P_-$$

$$Entropy(decision) = \sum P(D_i)(P_+(D_i)(\log_2 P_+(D_i))$$

$$+ P_-(D_i)(\log_2 P_-(D_i)))$$

Entropy Calculation Example

- Entropy for a dataset
  - Portion of Examples belonging to a certain class
  - $E(S)= \frac{-9}{14} \log\frac{9}{14} - \frac{5}{14}\log \frac{5}{14} = 0.94$
  - No of +ve examples= No of –ve examples
    - Entropy =1;
  - No of +ve examples= 0;
    - Entropy =0;
  - No of –ve examples=0;
    - Entropy =0;

Entropy Example

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

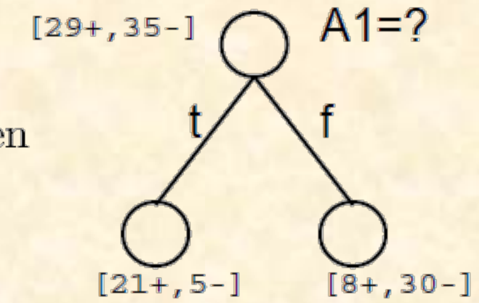
Deciding whether a pattern is interesting

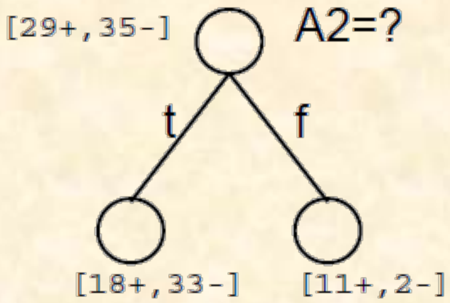
- We will use information theory
- A very large topic, originally used for compressing signals
- But more recently used for data mining…

Top-Down Induction of Decision Tree

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes





Which attribute is best?

Information Gain

- The information gain of a feature *F* is the expected reduction in entropy resulting from splitting on this feature.

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where S_v is the subset of *S* having value *v* for feature *F*.

- Entropy of each resulting subset weighted by its relative size.

- <big, red, circle>: + , <small, red, circle>: + , <small, red, square>: – , <big, blue, circle>: –

2+, 2 –: E=1
size
big: 1+,1– E=1
small: 1+,1– E=1
Gain=1–(0.5·1 + 0.5·1) = 0

2+, 2 – : E=1
color
red: 2+,1– E=0.918
blue: 0+,1– E=0
Gain=1–(0.75·0.918 + 0.25·0) = 0.311

2+, 2 – : E=1
shape
circle: 2+,1– E=0.918
square: 0+,1– E=0
Gain=1–(0.75·0.918 + 0.25·0) = 0.311

Information Gain Calculation Example

- Entropy for a dataset
  - E(S)= $\frac{-9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14}$ =0.94
- Entropy for Humidity
  - High[3+,4-], Normal [6+,1-]
  - Entropy[S_H]= $\frac{-3}{7}\log\frac{3}{7} - \frac{4}{7}\log\frac{4}{7}$
  - Entropy[S_N]= $\frac{-6}{7}\log\frac{6}{7} - \frac{1}{7}\log\frac{1}{7}$
  - Gain= 0.94- [$\frac{7}{14}$ Entropy[S_H] $+\frac{7}{14}$Entropy[S_N]]=0.151
- Entropy for Wind
  - Strong [6+,2-], Weak [3+,3-]
  - Entropy[S_W]= $\frac{-3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1$
  - Entropy[S_s]= $\frac{-6}{8}\log\frac{6}{8} - \frac{2}{8}\log\frac{2}{8}$ =0.811
  - Gain= 0.94- [$\frac{8}{14}$Entropy[S_s] $+\frac{6}{14}$Entropy[S_w]]=0.048

Gain(S, Humidity)> Gain(S, Wind) , Humidity is chosen as the root

Thank You…