# Machine Learning 10-701

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

January 11, 2011

#### Today:

- What is machine learning?
- Decision tree learning
- · Course logistics

#### Readings:

- "The Discipline of ML"
- Mitchell, Chapter 3
- · Bishop, Chapter 14.4

# Machine Learning:

Study of algorithms that

- improve their performance P
- at some task T
- with experience E

well-defined learning task: <P,T,E>

# Learning to Predict Emergency C-Sections [Sims et al., 2000] Data: Putient 103 time=1 Putient 103 time=2 Putient 103 time=n

9714 patient records,

each with 215 features

Age: 23 Age: 23 Age: 23 Age: 23 Age: 23 PitstPregnancy: no Anemia: no Diabetes: no Diabetes: YES Diabetes: no Diabetes: YES Diabetes: no Ultrasound: 2 Ultrasound: 2 Ultrasound: 2 Ultrasound: 2 Ultrasound: 2 Elective C-Section: no Emergency C-Section: 7 Emergency C-Section: 7 Emergency C-Section: Yes

One of 18 learned rules:

If No previous vaginal delivery, and
Abnormal 2nd Trimester Ultrasound, and
Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

Over training data: 26/41 = .63, Over test data: 12/20 = .60

# Learning to detect objects in images

(Prof. H. Schneiderman)

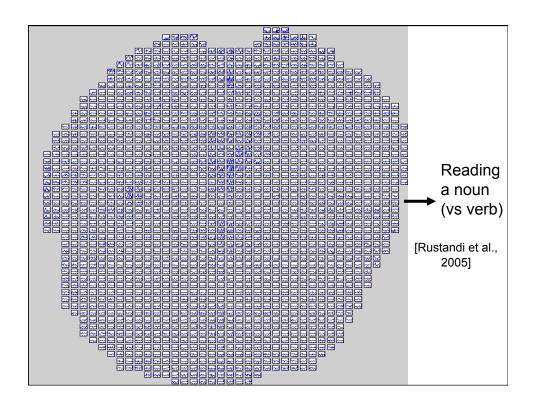


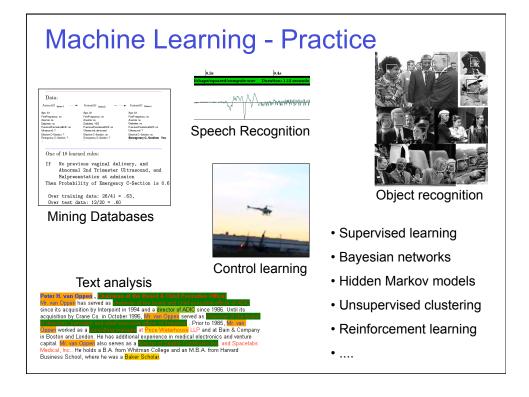


Example training images for each orientation

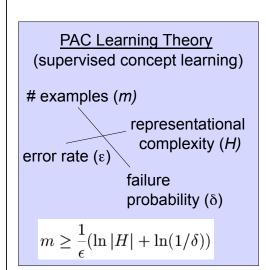


# Learning to classify text documents At Note the Company Outside Administration Company Outs



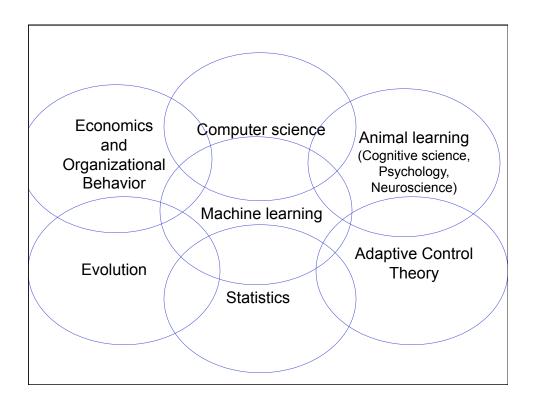






Other theories for

- Reinforcement skill learning
- · Semi-supervised learning
- · Active student querying
- ..
- ... also relating:
- # of mistakes during learning
- · learner's query strategy
- convergence rate
- asymptotic performance
- bias, variance



# Machine Learning in Computer Science

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - **–** ...

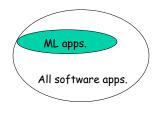
ML apps.

All software apps.

• This ML niche is growing (why?)

## Machine Learning in Computer Science

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - **–** ...



- · This ML niche is growing
  - Improved machine learning algorithms
  - Increased data capture, networking, new sensors
  - Software too complex to write by hand
  - Demand for self-customization to user, environment

Function Approximation and Decision tree learning

# Function approximation

#### **Problem Setting:**

- Set of possible instances X
- Unknown target function  $f: X \rightarrow Y$
- Set of function hypotheses  $H = \{ h \mid h : X \rightarrow Y \}$

#### Input:

superscript: ith training example

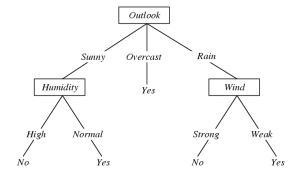
• Training examples  $\{\langle x^{(i)}, y^{(i)} \rangle\}$  of unknown target function f

#### Output:

• Hypothesis  $h \in H$  that best approximates target function f

#### A Decision tree for

F: <Outlook, Humidity, Wind, Temp> → PlayTennis?



Each internal node: test one attribute X<sub>i</sub>

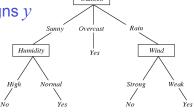
Each branch from a node: selects one value for X<sub>i</sub>

Each leaf node: predict Y (or  $P(Y|X \in leaf)$ )

# **Decision Tree Learning**

#### **Problem Setting:**

- Set of possible instances *X* 
  - each instance x in X is a feature vector
  - e.g., <Humidity=low, Wind=weak, Outlook=rain, Temp=hot>
- Unknown target function  $f: X \rightarrow Y$ 
  - Y is discrete valued
- Set of function hypotheses  $H = \{ h \mid h : X \rightarrow Y \}$ 
  - each hypothesis h is a decision tree
  - trees sorts x to leaf, which assigns y



# **Decision Tree Learning**

#### **Problem Setting:**

- Set of possible instances *X* 
  - each instance x in X is a feature vector  $x = \langle x_1, x_2 \dots x_n \rangle$
- Unknown target function  $f: X \rightarrow Y$ 
  - Y is discrete valued
- Set of function hypotheses  $H = \{ h \mid h : X \rightarrow Y \}$ 
  - each hypothesis h is a decision tree

#### Input:

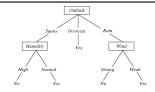
Training examples {<x(i),y(i)>} of unknown target function f

#### Output:

• Hypothesis  $h \in H$  that best approximates target function f

# **Decision Trees**

Suppose  $X = \langle X_1, ... X_n \rangle$ where  $X_i$  are boolean variables



How would you represent  $Y = X_2 X_5$ ?  $Y = X_2 \vee X_5$ 

How would you represent  $X_2 X_5 \vee X_3 X_4 (\neg X_1)$ 

#### A Tree to Predict C-Section Risk

Learned from medical records of 1000 women Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .10-
| | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

#### Top-Down Induction of Decision Trees

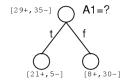
[ID3, C4.5, Quinlan]

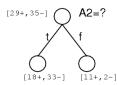
node = Root

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?





# **Entropy**

Entropy H(X) of a random variable X

# of possible values for X

$$H(X) = -\sum_{i=1}^{n} P(X=i) \log_2 P(X=i)$$

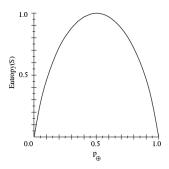
H(X) is the expected number of bits needed to encode a randomly drawn value of X (under most efficient code)

Why? Information theory:

- Most efficient code assigns -log<sub>2</sub>P(X=i) bits to encode the message X=i
- So, expected number of bits to code one random *X* is:

$$\sum_{i=1}^{n} P(X = i)(-\log_2 P(X = i))$$

#### Sample Entropy



- $\bullet$  S is a sample of training examples
- $p_{\oplus}$  is the proportion of positive examples in S
- ullet  $p_{\ominus}$  is the proportion of negative examples in S
- $\bullet$  Entropy measures the impurity of S

$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

# **Entropy**

Entropy H(X) of a random variable X

$$H(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

Specific conditional entropy H(X|Y=v) of X given Y=v:

$$H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v) \log_2 P(X = i|Y = v)$$

Conditional entropy H(X|Y) of X given Y:

$$H(X|Y) = \sum_{v \in values(Y)} P(Y = v)H(X|Y = v)$$

Mututal information (aka Information Gain) of X and Y:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Information Gain is the mutual information between input attribute A and target variable Y

Information Gain is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$$

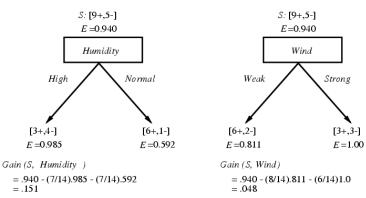


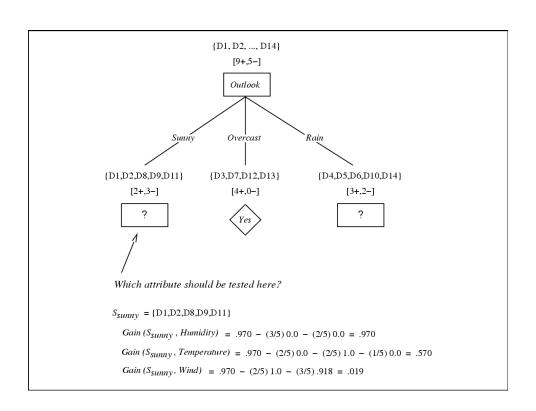
#### Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTenr
D1	Sunny	Hot	High	Weak	No
D2	Sunny	$\operatorname{Hot}$	$\operatorname{High}$	Strong	No
D3	Overcast	$\operatorname{Hot}$	$\operatorname{High}$	Weak	Yes
D4	Rain	Mild	$\operatorname{High}$	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	$\operatorname{Rain}$	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	$\operatorname{Sunny}$	Mild	$\operatorname{High}$	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	$\operatorname{Sunny}$	Mild	Normal	Strong	Yes
D12	Overcast	Mild	$\operatorname{High}$	Strong	Yes
D13	Overcast	$\operatorname{Hot}$	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# Selecting the Next Attribute

#### Which attribute is the best classifier?

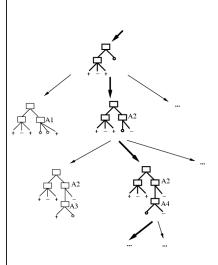




# **Decision Tree Learning Applet**

 http://www.cs.ualberta.ca/%7Eaixplore/learning/ DecisionTrees/Applet/DecisionTreeApplet.html

# Which Tree Should We Output?



- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

Occam's razor: prefer the simplest hypothesis that fits the data

Why Prefer Short Hypotheses? (Occam's Razor)
Arguments in favor:
Arguments opposed:

## Why Prefer Short Hypotheses? (Occam's Razor)

#### Argument in favor:

- Fewer short hypotheses than long ones
- → a short hypothesis that fits the data is less likely to be a statistical coincidence
- → highly probable that a sufficiently complex hypothesis will fit the data

#### Argument opposed:

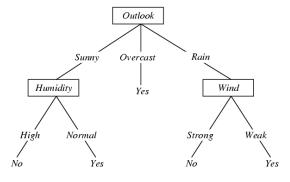
- Also fewer hypotheses with prime number of nodes and attributes beginning with "Z"
- What's so special about "short" hypotheses?

#### Overfitting in Decision Trees

Consider adding noisy training example #15:

 $Sunny,\ Hot,\ Normal,\ Strong,\ PlayTennis=No$ 

What effect on earlier tree?



# Overfitting

Consider error of hypothesis h over

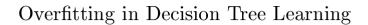
- training data:  $error_{train}(h)$
- ullet entire distribution  $\mathcal D$  of data:  $error_{\mathcal D}(h)$

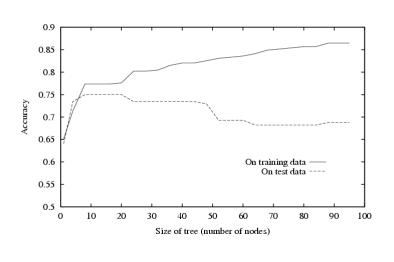
Hypothesis  $h \in H$  overfits training data if there is an alternative hypothesis  $h' \in H$  such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$





# Avoiding Overfitting

How can we avoid overfitting?

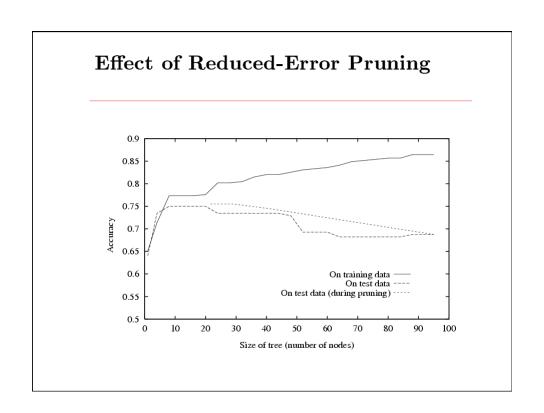
- stop growing when data split not statistically significant
- $\bullet$  grow full tree, then post-prune

#### Reduced-Error Pruning

Split data into training and validation set

Create tree that classifies *training* set correctly Do until further pruning is harmful:

- 1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
- 2. Greedily remove the one that most improves validation set accuracy
- produces smallest version of most accurate subtree
- What if data is limited?



#### Continuous Valued Attributes

Create a discrete attribute to test continuous

- $\bullet Temperature = 82.5$
- (Temperature > 72.3) = t, f

Temperature: 40 48 60 72 80 90 Play Tennis: No No Yes Yes Yes No

#### Attributes with Many Values

Problem:

- If attribute has many values, Gain will select it
- Imagine using  $Date = Jun_{-}3_{-}1996$  as attribute

One approach: use GainRatio instead

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

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

where  $S_i$  is subset of S for which A has value  $v_i$ 

# What you should know:

- · Well posed function approximation problems:
  - Instance space, X
  - Sample of labeled training data { <x(i), y(i)>}
  - Hypothesis space, H = { f: X→Y }
- Learning is a search/optimization problem over H
  - Various objective functions
    - minimize training error (0-1 loss)
    - among hypotheses that minimize training error, select smallest (?)
- · Decision tree learning
  - Greedy top-down learning of decision trees (ID3, C4.5, ...)
  - Overfitting and tree/rule post-pruning
  - Extensions...