Note to other teachers and users of these slides. Andrew would be delighted if you found this source material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. PowerPoint originals are available. If you make use of a significant portion of these slides in your own lecture, please include this message, or the following link to the source repository of Andrew's tutorials: http://www.cs.cmu.edu/~awm/tutorials . Comments and corrections gratefully received.

PAC-learning

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Copyright © 2001, Andrew W. Moore Nov 30th, 2001

Probably Approximately Correct (PAC) Learning

- Imagine we're doing classification with categorical inputs.
- All inputs and outputs are binary.
- Data is noiseless.
- There's a machine $f(x,h)$ which has H possible settings (a.k.a. hypotheses), called $h_1, h_2 .. h_H$.

Copyright © 2001, Andrew W. Moore

Example of a machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm that contain only logical ands.
- Example hypotheses:
- X1 ^ X3 ^ X19
- X3 ^ X18
- X7
- X1 ^ X2 ^ X3 ^ … ^ Xm
- Question: if there are 3 attributes, what is the complete set of hypotheses in f?

Copyright © 2001, Andrew W. Moore

Example of a machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm that contain only logical ands.
- Example hypotheses:
- X1 ^ X3 ^ X19
- X3 ^ X18
- X7
- X1 ^ X2 ^ X3 ^ … ^ Xm
- Question: if there are 3 attributes, what is the complete set of hypotheses in f? (H = 8)

True	X2	X3	X2 ^ X3
X1	X1 ^ X2	X1 ^ X3	X1 ^ X2 ^ X3

Copyright © 2001, Andrew W. Moore

And-Positive-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm that contain only logical ands.
- Example hypotheses:
- X1 ^ X3 ^ X19
- X3 ^ X18
- X7
- X1 ^ X2 ^ X3 ^ … ^ Xm
- Question: if there are m attributes, how many hypotheses in f?

Copyright © 2001, Andrew W. Moore

And-Positive-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm that contain only logical ands.
- Example hypotheses:
- X1 ^ X3 ^ X19
- X3 ^ X18
- X7
- X1 ^ X2 ^ X3 ^ … ^ Xm
- Question: if there are m attributes, how many hypotheses in f? ($H = 2^m$)

Copyright © 2001, Andrew W. Moore

And-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm or their negations that contain only logical ands.
- Example hypotheses:
- X1 ^ ~X3 ^ X19
- X3 ^ ~X18
- ~X7
- X1 ^ X2 ^ ~X3 ^ … ^ Xm
- Question: if there are 2 attributes, what is the complete set of hypotheses in f?

Copyright © 2001, Andrew W. Moore

And-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm or their negations that contain only logical ands.
- Example hypotheses:
- X1 ^ ~X3 ^ X19
- X3 ^ ~X18
- ~X7
- X1 ^ X2 ^ ~X3 ^ … ^ Xm
- Question: if there are 2 attributes, what is the complete set of hypotheses in f? (H = 9)

True		True
True		X2
True		~X2
X1		True
X1	^	X2
X1	^	~X2
~X1		True
~X1	^	X2
~X1	^	~X2

Copyright © 2001, Andrew W. Moore

And-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm or their negations that contain only logical ands.
- Example hypotheses:
- X1 ^ ~X3 ^ X19
- X3 ^ ~X18
- ~X7
- X1 ^ X2 ^ ~X3 ^ … ^ Xm
- Question: if there are m attributes, what is the size of the complete set of hypotheses in f?

True		True
True		X2
True		~X2
X1		True
X1	^	X2
X1	^	~X2
~X1		True
~X1	^	X2
~X1	^	~X2

Copyright © 2001, Andrew W. Moore

And-Literals Machine

- f(x,h) consists of all logical sentences about X1, X2 .. Xm or their negations that contain only logical ands.
- Example hypotheses:
- X1 ^ ~X3 ^ X19
- X3 ^ ~X18
- ~X7
- X1 ^ X2 ^ ~X3 ^ … ^ Xm
- Question: if there are m attributes, what is the size of the complete set of hypotheses in f? ($H = 3^m$)

True		True
True		X2
True		~X2
X1		True
X1	^	X2
X1	^	~X2
~X1		True
~X1	^	X2
~X1	^	~X2

Copyright © 2001, Andrew W. Moore

Lookup Table Machine

- f(x,h) consists of all truth tables mapping combinations of input attributes to true and false
- Example hypothesis:

X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

- Question: if there are m attributes, what is the size of the complete set of hypotheses in f?

Copyright © 2001, Andrew W. Moore

Lookup Table Machine

- f(x,h) consists of all truth tables mapping combinations of input attributes to true and false
- Example hypothesis:

X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

- Question: if there are m attributes, what is the size of the complete set of hypotheses in f?

$$H = 2^{2^m}$$

Copyright © 2001, Andrew W. Moore

A Game

- We specify f, the machine
- Nature chooses hidden random hypothesis h*
- Nature randomly generates R datapoints
  - How is a datapoint generated?
    1. Vector of inputs $\boldsymbol{x}_k = (x_{k1}, x_{k2}, x_{km})$ is drawn from a fixed unknown distrib: D
    2. The corresponding output $y_k = f(\boldsymbol{x}_k, h^*)$
- We learn an approximation of h* by choosing some h^{est} for which the training set error is 0

Copyright © 2001, Andrew W. Moore

Test Error Rate

- We specify f, the machine
- Nature chooses hidden random hypothesis h*
- Nature randomly generates R datapoints
  - How is a datapoint generated?
    1. Vector of inputs $\boldsymbol{x}_k = (x_{k1}, x_{k2}, x_{km})$ is drawn from a fixed unknown distrib: D
    2. The corresponding output $y_k = f(\boldsymbol{x}_k, h^*)$
- We learn an approximation of h* by choosing some h^{est} for which the training set error is 0
- For each hypothesis *h* ,
- Say *h* is Correctly Classified (CCd) if *h* has zero training set error
- Define TESTERR(*h*)
  - = Fraction of test points that *h* will classify correctly
  - = P(*h* classifies a random test point correctly)
- Say *h* is BAD if TESTERR(*h*) > ε

Copyright © 2001, Andrew W. Moore

Test Error Rate

- We specify f, the machine
- Nature chooses hidden random hypothesis h*
- Nature randomly generates R datapoints
  - How is a datapoint generated?
    1. Vector of inputs $\mathbf{x}_k = (x_{k1}, x_{k2}, x_{km})$ is drawn from a fixed unknown distrib: D
    2. The corresponding output $y_k = f(\mathbf{x}_k, h^*)$
- We learn an approximation of h* by choosing some h^{est} for which the training set error is 0
- For each hypothesis *h* ,
- Say *h* is Correctly Classified (CCd) if *h* has zero training set error
- Define TESTERR(*h*)
  - = Fraction of test points that *h* will classify correctly
  - = P(*h* classifies a random test point correctly)
- Say *h* is BAD if TESTERR(*h*) > ε

$$P(h \text{ is CCd} \mid h \text{ is bad}) =$$

$$P(\forall k \in \text{Training Set}, f(x_k, h) = y_k \mid h \text{ is bad})$$

$$\leq (1-\varepsilon)^R$$

Copyright © 2001, Andrew W. Moore

Test Error Rate

- We specify f, the machine
- Nature chooses hidden random hypothesis h*
- Nature randomly generates R datapoints
  - How is a datapoint generated?
    1. Vector of inputs $\mathbf{x}_k = (x_{k1}, x_{k2}, x_{km})$ is drawn from a fixed unknown distrib: D
    2. The corresponding output $y_k = f(\mathbf{x}_k, h^*)$
- We learn an approximation of h* by choosing some h^{est} for which the training set error is 0
- For each hypothesis *h* ,
- Say *h* is Correctly Classified (CCd) if *h* has zero training set error
- Define TESTERR(*h*)
  - = Fraction of test points that *h* will classify correctly
  - = P(*h* classifies a random test point correctly)
- Say *h* is BAD if TESTERR(*h*) > ε

$$P(h \text{ is CCd} \mid h \text{ is bad}) =$$

$$P(\forall k \in \text{Training Set}, f(x_k, h) = y_k \mid h \text{ is bad})$$

$$\leq (1-\varepsilon)^R$$

$$P(\text{we learn a bad } h) \leq$$

$$P\begin{pmatrix}\text{the set of CCd } h\text{'s} \\ \text{contains a bad } h\end{pmatrix} =$$

$$P(\exists h.\, h \text{ is CCd} \wedge h \text{ is bad}) =$$

$$P\begin{pmatrix}(h_1 \text{ is CCd} \wedge h_1 \text{ is bad}) \vee \\ (h_2 \text{ is CCd} \wedge h_2 \text{ is bad}) \vee \\ \vdots \\ (h_H \text{ is CCd} \wedge h_H \text{ is bad})\end{pmatrix} \leq$$

$$\sum_{i=1}^{H} P(h_i \text{ is CCd} \wedge h_i \text{ is bad}) \leq$$

$$\sum_{i=1}^{H} P(h_i \text{ is CCd} \mid h_i \text{ is bad}) =$$

$$H \times P(h_i \text{ is CCd} \mid h_i \text{ is bad}) \leq H(1-\varepsilon)^R$$

Copyright © 2001, Andrew W. Moore

PAC Learning

- Choose *R* such that with probability less than δ we'll select a bad h^{est} (i.e. an h^{est} with 0 train error but which will have test error ε or worse)

$$P(\text{error of } h \text{ is} \geq \varepsilon) \leq \delta$$

- Probably Approximately Correct
- As we just saw, this can be achieved by choosing *R* such that

$$H(1-\varepsilon)^R \leq \delta$$

- i.e. *R* such that

$$R \geq \frac{1}{\varepsilon}\left(\log_2 H + \log_2 \frac{1}{\delta}\right)$$

Whoah wait...
Why is that?
$(1-\varepsilon) \leq e^{-\varepsilon}$

Copyright © 2001, Andrew W. Moore

PAC in action

Machine	Example Hypothesis	H	R required to PAC-learn
And-positive-literals	X3 ^ X7 ^ X8	2^m	$\frac{1}{\varepsilon}\left(m + \log_2 \frac{1}{\delta}\right)$
And-literals	X3 ^ ~X7	3^m	$\frac{1}{\varepsilon}\left((\log_2 3)m + \log_2 \frac{1}{\delta}\right)$
Lookup Table		2^{2^m}	$\frac{1}{\varepsilon}\left(2^m + \log_2 \frac{1}{\delta}\right)$
And-lits or And-lits	(X1 ^ X5) v (X2 ^ ~X7 ^ X8)	$(3^m)^2 = 3^{2m}$	$\frac{1}{\varepsilon}\left((2\log_2 3)m + \log_2 \frac{1}{\delta}\right)$

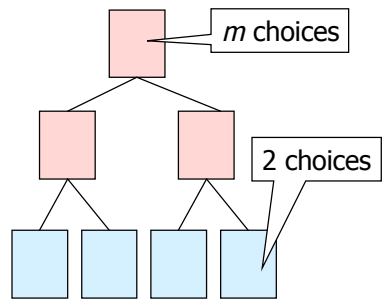
X1	X2	X3	X4	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Copyright © 2001, Andrew W. Moore

PAC for decision trees of depth k

- Assume *m* binary attributes, binary class labels
- H_k = Number of decision trees of depth *k*

Example: k = 2



- $H_0 = 2$
- H_k = (#choices of root attribute) * (# possible left subtrees) * (# possible right subtrees) $= m * H_{k-1} * H_{k-1}$
- Write $L_k = \log_2 H_k$
- $L_0 = 1$
- $L_k = \log_2 m + 2L_{k-1}$
- So $L_k = (2^k-1)(1+\log_2 m) +1$
- So to PAC-learn, need

$$R \geq \frac{1}{\varepsilon}\left((2^k-1)(1+\log_2 m)+1+\log_2\frac{1}{\delta}\right)$$

Copyright © 2001, Andrew W. Moore

What you should know

- Be able to understand every step in the math that gets you to

$$P(\text{we learn a bad } h) \leq H(1-\varepsilon)^R$$

- Understand that you thus need this many records to PAC-learn a machine with H different hypotheses

$$R \geq \frac{1}{\varepsilon}\left(\log_2 H + \log_2\frac{1}{\delta}\right)$$

- Understand examples of deducing H for various machines (i.e. counting hypotheses.)

Copyright © 2001, Andrew W. Moore