

RN, Chapter  
18.5



# Computational Learning Theory

---



# Computational Learning Theory

---

- Inductive Learning
  - Protocol
  - Error
- Probably Approximately Correct Learning
  - Consistency Filtering
  - Sample Complexity
  - Eg: Conjunction, Decision List
- Issues
  - Bound
  - Other Models

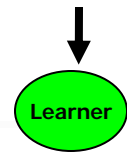
# What General Laws constrain Inductive Learning?

- Sample Complexity
  - How many training examples are sufficient to learn target concept?
- Computational Complexity
  - Resources required to learn target concept?
- Want theory to relate:
  - Training examples
    - Quantity
    - Quality
    - How presented
  - Complexity of hypothesis/concept space
  - Accuracy of approx to target concept
  - Probability of successful learning

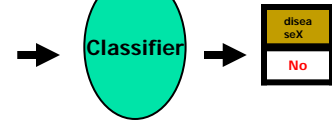
These results only useful wrt  $O(\dots)$  !

# Protocol

Age	Sex	Smokes	...	Color	diseaseX
35	95	Y	...	Pale	No
22	11	N	...	Clear	Yes
...	...	...	...	...	...
10	87	N	...	Pale	No



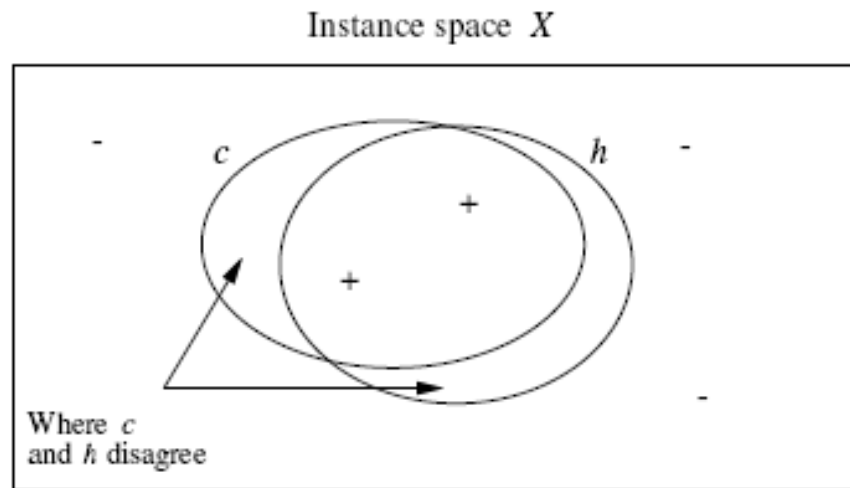
Age	Sex	Smokes	...	Color
3	2	9	0	N
...	...	...	...	...
...	...	...	...	...



diseaseX
No

- Given:
  - set of examples  $X$
  - fixed (unknown) distribution  $D$  over  $X$
  - set of hypotheses  $H$
  - set of possible target concepts  $C$
- Learner observes sample  $S = \{ \langle x_i, c(x_i) \rangle \}$ 
  - instances  $x_i$  drawn from distr.  $D$
  - labeled by target concept  $c \in C$   
(Learner does NOT know  $c(\cdot), D$ )
- Learner outputs  $h \in H$  estimating  $c$ 
  - $h$  is evaluated by performance on subsequent instances drawn from  $D$
- For now:
  - $C = H$  (so  $c \in H$ )
  - Noise-free data

# True Error of Hypothesis



Def'n: The true error of hypothesis  $h$  wrt

- target concept  $c$
- distribution  $D$

$\equiv$  probability that  $h$  will misclassify instance drawn from  $D$

$$\text{err}_D(h) = \Pr_{x \in D} [ c(x) \neq h(x) ]$$



# Probably Approximately Correct

---

Goal:

PAC-Learner produces hypothesis  $\hat{h}$  that  
is approximately correct,

$$\text{err}_D(\hat{h}) \approx 0$$

with high probability

$$P(\text{err}_D(\hat{h}) \approx 0) \approx 1$$

- Double "hedging"
  - approximately
  - probably

Need both!



# PAC-Learning

- Learner  $L$  can draw labeled instance  $\langle x, c(x) \rangle$  in unit time  
 $x \in X$  drawn from distribution  $D$  labeled by target concept  $c \in C$

**Def'n:** Learner  $L$  PAC-learns class  $C$  (by  $H$ )

if

1. for any target concept  $c \in C$ ,  
any distribution  $D$ , any  $\varepsilon, \delta > 0$ ,  
 $L$  returns  $h \in H$  s.t.  
w/ prob.  $\geq 1 - \delta$ ,  $\text{err}_D(h) < \varepsilon$
2.  $L$ 's run-time (and hence, sample complexity)  
is  $\text{poly}(|X|, \text{size}(c), 1/\varepsilon, 1/\delta)$

- Sufficient:
  1. Only  $\text{poly}(\dots)$  training instances –  $|H| = 2^{\text{poly}(\dots)}$
  2. Only  $\text{poly}$  time / instance ...

Often  $C = H$

# Simple Learning Algorithm: Consistency Filtering



- Draw  $m_H(\epsilon, \delta)$  random (labeled) examples  $S_m$
- Remove every hyp. that contradicts any  $\langle x, y \rangle \in S_m$
- Return any remaining (consistent) hypothesis

Challenges:

- Q1: Sample size:  $m_H(\epsilon, \delta)$
- Q2: Need to decide if  $h \in H$  is consistent w/ all  $S_m$   
... efficiently ...



# Boolean Functions ( $\equiv$ Concepts)

Eg:  $h_{X_1 \vee \neg X_2}(X_1, X_2, X_3) = \begin{cases} 1 & \text{if } X_1 \vee \neg X_2 \\ 0 & \text{otherwise} \end{cases}$

$X_1$	$X_2$	$X_3$	$h_{X_1 \vee \neg X_2}(X_1, X_2, X_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$h_{X_1 \vee \neg X_2}(0, 1, 1) = 0$

$h_{X_1 \vee \neg X_2}(1, 1, 0) = 1$

Note: Hypothesis maps unlabeled-tuple to  $\{0, 1\}$

Labeled-tuple is  $\left\{ \begin{array}{l} \text{Consistent} \\ \text{Inconsistent} \end{array} \right\}$  w/ hyp.

So  $\langle (0, 1, 1), 1 \rangle$  is

Inconsistent with  $h_{X_1 \vee \neg X_2}$

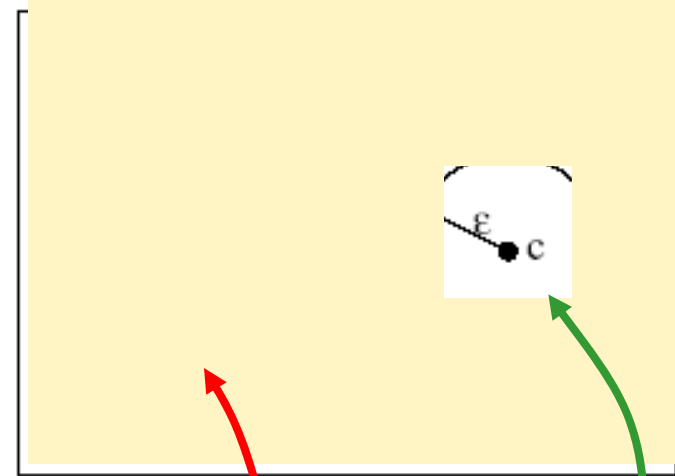
Consistent with  $h_{X_2 \vee X_3}$

# Bad Hypotheses

Idea: Find  $m = m_H(\epsilon, \delta)$  s.t.  
after seeing  $m$  examples,  
every BAD hypothesis  $h$  ( $\text{err}_{D,c}(h) > \epsilon$ )  
will be ELIMINATED  
with high probability ( $\approx 1 - \delta$ )  
leaving only good hypotheses

... then pick ANY of the remaining good ( $\text{err}_{D,c}(h) < \epsilon$ ) hyp's

Find  $m$  large number that  
very small chance that a "bad" hypothesis is consistent with  $m$  examples

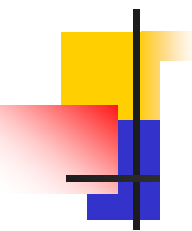


Eliminate

$$H_{\text{bad}} = \{ h \in H \mid \text{err}_D(h) > \epsilon \}$$

Leave

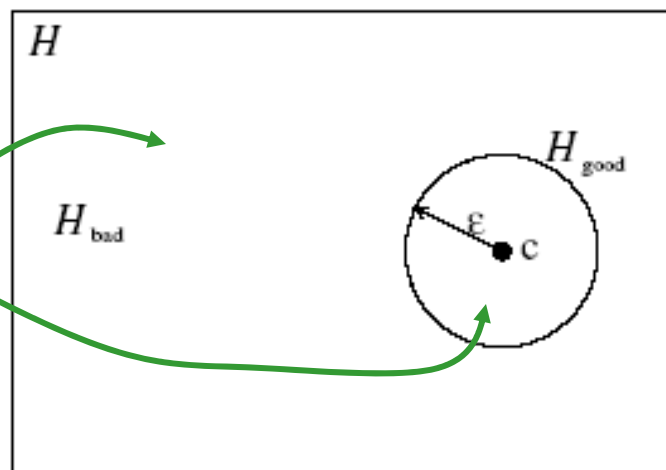
$$H_{\text{Good}} = \{ h \in H \mid \text{err}_D(h) \leq \epsilon \}$$



$$\mathcal{H}_{\text{bad}} = \{h \in \mathcal{H} \mid \text{err}_{\mathcal{D}}(h) > \epsilon\}$$

$$\mathcal{H}_{\text{good}} = \{h \in \mathcal{H} \mid \text{err}_{\mathcal{D}}(h) \leq \epsilon\}$$

Note  $|\mathcal{H}_{\text{Bad}}| \leq |\mathcal{H}|$



	$x_1$	$x_2$	$\dots$	$x_m$	Good/Bad
$h_1$	$\checkmark$	$\checkmark$	$\dots$	$\checkmark$	$< \epsilon$
$h_2$	$\checkmark$	$-$	$\dots$	$\checkmark$	$< \epsilon$
$\vdots$			$\vdots$		$\vdots$
$h_k$	$\checkmark$	$\checkmark$	$\dots$	$-$	$< \epsilon$
$h_{\text{bad},1}$	$-$	$-$	$\dots$	$\checkmark$	$> \epsilon$
$\vdots$			$\vdots$		$\vdots$
$h_{\text{bad},r}$	$-$	$-$	$\dots$	$-$	$> \epsilon$

Ok

Bad

$\checkmark$ :  $h_i(x_j) = c(x_j)$

$-$ :  $h_i(x_j) \neq c(x_j)$



# Sample Bounds – Derivation

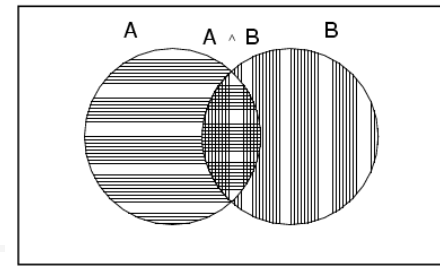
---

- Let  $h_1$  be  $\varepsilon$ -bad hypothesis ...  $\text{err}(h_1) > \varepsilon$ 
  - $\Rightarrow h_1$  mis-labels example w/prob  $P(h_1(x) \neq c(x)) > \varepsilon$
  - $\Rightarrow h_1$  **correctly** labels random example w/prob  $\leq (1 - \varepsilon)$
- As examples drawn **INDEPENDENTLY**
  - $P(h_1 \text{ correctly labels } m \text{ examples}) \leq (1 - \varepsilon)^m$

# Sample Bounds

## – Derivation II

True

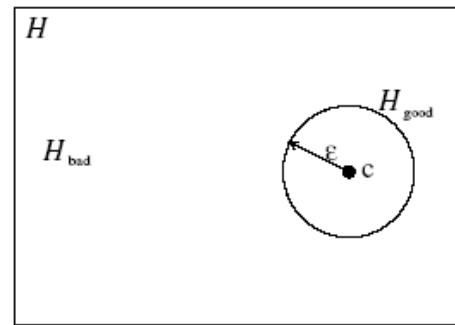


- Let  $h_2$  be another  $\varepsilon$ -bad hypothesis
- What is probability that *either*  $h_1$  or  $h_2$  survive  $m$  random examples?

$$\begin{aligned} P(h_1 \vee h_2 \text{ survives}) &= P(h_1 \text{ survives}) + P(h_2 \text{ survives}) \\ &\quad - P(h_1 \& h_2 \text{ survives}) \\ &\leq P(h_1 \text{ survives}) + P(h_2 \text{ survives}) \\ &\leq 2(1 - \varepsilon)^m \end{aligned}$$

- If  $k$   $\varepsilon$ -bad hypotheses  $\{h_1, \dots, h_k\}$ :  
 $P(h_1 \vee \dots \vee h_k \text{ survives}) \leq k(1 - \varepsilon)^m$

# Sample Bounds, con't



Let  $H_{\text{bad}} = \{ h \in H \mid \text{err}(h) > \varepsilon \}$

- Probability that any  $h \in H_{\text{bad}}$  survives is

$$P(\text{any } h_b \text{ in } H_{\text{bad}} \text{ is consistent with } m \text{ exs.}) \\ \leq |H_{\text{bad}}| (1 - \varepsilon)^m \leq |H| (1 - \varepsilon)^m$$

- This is  $\leq \delta$  if  $|H| (1 - \varepsilon)^m \leq \delta$

$\Rightarrow$

$$m_H(\varepsilon, \delta) \geq \left( \log \frac{|H|}{\delta} \right) / -\log(1 - \varepsilon) \geq \frac{1}{\varepsilon} \left( \log \frac{|H|}{\delta} \right)$$

- $m_H(\varepsilon, \delta)$  is "Sample Complexity" of hypothesis space  $H$
- Fact: For  $0 \leq \varepsilon \leq 1$ ,  $(1 - \varepsilon) \leq e^{-\varepsilon}$

# Sample Complexity

- Hypothesis Space (expressiveness):  $H$
- Error Rate of Resulting Hypthesis:  $\varepsilon$ 
  - $\text{err}_{D,c}(h) = P( h(x) \neq c(x) ) \leq \varepsilon$
- Confidence of being  $\varepsilon$ -close:  $\delta$ 
  - $P( \text{err}_{D,c}(h) \leq \varepsilon ) > 1 - \delta$
- Sample size:  $m_H( \varepsilon, \delta )$

- Any hypothesis consistent with

$$m_H( \varepsilon, \delta ) = \frac{1}{\varepsilon} \left( \log \frac{|H|}{\delta} \right)$$

examples,

has error of at most  $\varepsilon$ , with prob  $\leq 1 - \delta$

# Boolean Function... Conjunctions

- Boolean Instance:  $\langle x_1, \dots, x_n \rangle$   
 $\langle 1, 0, 1, 1 \rangle$  for  $\langle x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1 \rangle$
- Boolean Function:  $f(\langle x_1, \dots, x_n \rangle) \in \{0, 1\}$
- Conjunction (type of Boolean function)

$$f_{+-0-0+}(X) = x_1 \bar{x}_2 \bar{x}_4 x_6$$
$$= \begin{cases} 1 & \text{if } x_1(X) = t, x_2(X) = f, x_4(X) = f, \\ & \text{and } x_6(X) = t \\ 0 & \text{otherwise} \end{cases}$$

$$f_{+-0-0+}(\langle \underline{1}, \underline{0}, 1, \underline{0}, 0, \underline{1} \rangle) = 1$$

$$f_{+-0-0+}(\langle \underline{0}, \underline{0}, 1, \underline{0}, 0, \underline{1} \rangle) = 0$$

(Ie, must match each literal mentioned)

- Only  $3^n$  possible conjunctions  
out of  $2^{2^n}$  boolean functions!



- $\mathcal{H}_C$  = conjunctions of literals

- $|\mathcal{H}_C| = 3^n$ :  $\left( \begin{array}{l} \text{Each variable can be} \\ \circ \text{ included positively } "x_i", \\ \circ \text{ included negatively } "\bar{x}_i", \\ \circ \text{ excluded} \end{array} \right)$

$$\Rightarrow m_{\mathcal{H}_C}(\epsilon, \delta) = \frac{1}{\epsilon} \left[ n \ln 3 + \ln \frac{1}{\delta} \right]$$

Alg:

Collect  $m_{\mathcal{H}_C}(\epsilon, \delta) = \frac{1}{\epsilon} \left[ n \ln 3 + \ln \frac{1}{\delta} \right]$  labeled samples

Let  $h = x_1 \bar{x}_1 x_2 \bar{x}_2 \cdots x_n \bar{x}_n$

For each +-example  $y = \bigwedge_i \pm_i x_i$

Remove from  $h$  any literal NOT included in  $y$

Data	Current Hyp						
	$x_1$	$\bar{x}_1$	$x_2$	$\bar{x}_2$	$x_3$	$\bar{x}_3$	
$\langle (1 \ 0 \ 1) + \rangle$							Never true
							True only for "101"
							True only for "10*"

- Just uses +-examples!
  - Finds "smallest" hypothesis (true for as few +-examples as possible)
  - ... No mistakes on -examples
- As each step is efficient  $O(n)$ , only  $\text{poly}(n, 1/\epsilon, 1/\delta)$  steps  
 $\Rightarrow$  algorithm is *efficient!*
- Does NOT explicitly build all  $3^n$  conjunctions, then throw some out...

# PAC-Learning k-CNF

- $CNF \equiv$  Conjunctive Normal Form

$$(x_1 \vee \bar{x}_2 \vee x_7) \wedge (x_2 \vee x_4 \vee \bar{x}_9) \wedge \dots \wedge (x_7 \vee \bar{x}_8 \vee \bar{x}_9)$$

- $k-CNF \equiv$  CNF where each clause has  $\leq k$  literals  
1-CNF  $\equiv$  Conjunctions

- As  $\exists O\left(\binom{n}{k} 3^k\right)$  possible  $\leq k$ -clauses,

$$|\mathcal{H}_{k-CNF}| = 2^{O\left(\binom{n}{k} 3^k\right)}$$

(n choose k) =  $O(n^k)$

$$\Rightarrow m_{\mathcal{H}_{k-CNF}} = O\left(\frac{1}{\epsilon} \left[(3n)^k + \ln \frac{1}{\delta}\right]\right)$$

Alg: Consistency Filtering:

Let  $T =$  all  $O\left(\binom{n}{k} 3^k\right)$  possible  $k$ -clauses.

After each  $\pm$ -example  $y$ ,

Remove from  $T$  all clauses INCONSISTENT w/  $y$

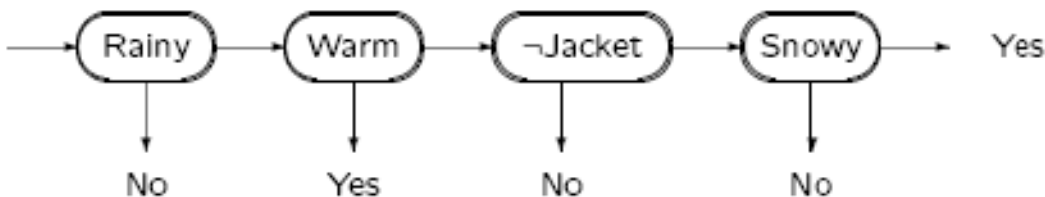
Return  $\bigwedge T$

- Similar for Disjunctions,  $k$ -DNF, ...

? What about CNF  $\equiv$   $n$ -CNF ?

# Decision Lists

- When to go for walk?
  - Vars: *rainy*, *warm*, *jacket*, *snowy*
  - Don't go for walk if *rainy*.  
Otherwise, go for walk if *warm* or  
if I *jacket* and it is *snowy*.



Def'n: A *DL*  $\equiv$  list of "if-then rules"  
where  $\left\{ \begin{array}{l} \text{condition} \equiv \text{a literal} \\ \text{consequent is } + \text{ or } - \end{array} \right\}$

( $\equiv$  decision tree with just one long path)

- How many DLs?
  - 4n possible "rules", each of form " $\pm x_i \Rightarrow \pm$ "
  - $\Rightarrow (4n)!$  orderings, so  $|H_{DL}| \cdot (4n)!$
  - (Actually:  $\leq n! 4^n$ )

# Example of Learning DL

Data:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Label
$i_1$	1	0	0	1	1	A
$i_2$	0	1	1	0	0	B
$i_3$	1	1	1	0	0	A
$i_4$	0	0	0	1	0	B
$i_5$	1	1	0	1	1	A
$i_6$	1	0	0	0	1	B

1. When  $x_1 = 0$ , class is "B"

Form  $h = \langle \neg x_1 \mapsto B \rangle$

Eliminate  $i_2, i_4$

2. When  $x_2 = 1$ , class is "A"

Form  $h = \langle \neg x_1 \mapsto B; x_2 \mapsto A \rangle$

Eliminate  $i_3, i_5$

3. When  $x_4 = 1$ , class is "A"

Form  $h = \langle \neg x_1 \mapsto B; x_2 \mapsto A; x_4 \mapsto A \rangle$

Eliminate  $i_1$

4. Always have class "B"

Form  $h = \langle \neg x_1 \mapsto B; x_2 \mapsto A; x_4 \mapsto A; t \mapsto B \rangle$

Eliminate rest ( $i_6$ )

# PAC-Learning Decision Lists

Let:  $S$  = set of

$$m_{DL} = O\left(\frac{1}{\epsilon}[n \ln(n) + \ln \frac{1}{\delta}]\right)$$

training instances

$h$  = empty list

$R$  = all  $4n$  possible rules

While  $S \neq \{\}$  do

1. Find  $r \in R$  s.t.
  - + consistent w/  $S$
  - +  $r$  applies to  $\geq 1$   $s \in S$

(If none, halt w/ "Failure")

2.  $h := h \circ r$

(Put rule at BOTTOM of hypothesis)

3.  $S := S - \{s \mid s \text{ classified by } h\}$

(Throw out examples classified by current hypothesis)



# Proof (PAC-Learn DL)

- *Correctness#1: Enough data?*

Yes.  $\frac{1}{\epsilon} \ln \frac{|\mathcal{H}_{DL}|}{\delta}$

- *Correctness#2: Consistency?*

If  $\exists$  DL consistent w/data...

1.  $\exists \geq 1$  choice for step 1  
(e.g., first rule in  $\mathcal{L}$  satisfied by  $\geq 1$  example)
2.  $\exists$  DL consistent w/ remaining data  
— original DL!

- *Efficiency:*

Algorithm runs in poly time, since

- each iteration requires poly time, and
- each iteration removes  $\geq 1$  example  
(only *poly* examples)

- Generalization:  $k$ -DL

... whose nodes each contain

CONJUNCTION of  $\leq k$  literals

(So earlier DL  $\equiv$  1-DL.)

Note:  $k$ -DL  $\supset$   $k$ -CNF,  $k$ -DNF,  $k$ -depth DecTree, ...



# Why Learning May Succeed

- Learner  $L$  produces classifier  $h = L(S)$  that does well on training data  $S$

Why?

1. If  $x$  appears a lot

- then  $x$  probably occurs in training data  $S$
- As  $h$  does well on  $S$ ,  $\delta$   
 $h(x)$  is probably correct on  $x$

2. If example  $x$  appears rarely

(  $P(x) \approx 0$  )

then  $h$  suffers only small penalty for being wrong.  $\epsilon$

- Assumption: Distribution is "stationary"
  - distr. for testing = distr. for training

# Comments on Model

$$m_H(\varepsilon, \delta) = \frac{1}{\varepsilon} \left( \log \frac{|H|}{\delta} \right)$$

## Simplify task:

- 1\*. Assume  $c \in H$ , where  $H$  known
  - (Eg, lines, conjunctions, . . . )
- 2\*. Noise free training data
- 3. Only require approximate correctness:
  - $h$  is " $\varepsilon$ -good":  $P_x( h(x) \neq c(x) ) < \varepsilon$
- 4. Allow learner to (rarely) be completely off
  - If examples NOT representative, cannot do well.
  - $P( h_L \text{ is } \varepsilon\text{-good} ) \leq 1 - \delta$

## Complicate task:

- 1. Learner must be computationally efficient
- 2. Over any instance distribution



# Comments: Sample Complexity

$$m_H(\varepsilon, \delta) = \frac{1}{\varepsilon} \left( \log \frac{|H|}{\delta} \right)$$

- If  $k$  parameters,  $\langle v_1, \dots, v_k \rangle$ 
  - $\Rightarrow |H_k| \approx B^k$
  - $\Rightarrow m_{H_k} \approx \log(B^k)/\varepsilon \approx k/\varepsilon$
- Too GENEROUS:
  - Based on pre-defined  $C = \{c_1, \dots\} = H$   
Where did this come from???
  - Assumes  $c \in H$ , noise-free
  - If  $\text{err} \neq 0$ , need  $O(1/\varepsilon^2 \dots)$

# Why is Bound so Lousy!

- Assumes error of all  $\varepsilon$ -bad hypotheses  $\approx \varepsilon$   
(Typically most bad hypotheses are really bad  
 $\Rightarrow$  get thrown out much sooner)
- Uses  $P(A \text{ or } B) \leq P(A) + P(B)$ .  
(If hypotheses are correlated, then if one inconsistent,  
others probably inconsistent too)
- Assumes  $|H_{\text{bad}}| = |H|$  ... see VCdimension
- WorstCase:
  - over all  $c \in C$
  - over all distribution  $D$  over  $X$
  - over all presentations of instances (drawn from  $D$ )
- Improvements
  - "Distribution Specific" learning  
Known single dist ( $\varepsilon$ -cover)  
Gaussian, . . .
  - Look at samples!  $\Rightarrow$  Sequential PAC Learning

If  $\lambda$ -bad, takes  $\approx 1/\lambda$  to see evidence

# Fundamental Tradeoff in Machine Learning

$$m_H(\varepsilon, \delta) = \frac{1}{\varepsilon} \left( \log \frac{|H|}{\delta} \right)$$

- Larger  $H$  is more likely to include
  - (approx to) target  $f$
  - but it requires more examples to learn
- w/few examples, cannot reliably find good hypothesis from large hypothesis space
- To learn effectively ( $\varepsilon$ ) from small # of samples ( $m$ ), only consider  $H$  where  $|H| \approx e^{\varepsilon m}$
- Restrict form of Boolean function to reduce size of hypotheses space.
  - Eg, for  $H_C$  = conjunctions of literals,  $|H_C| = 3^n$ , so only need poly number of examples!
  - Great if target concept is in  $H_C$ , but . . .

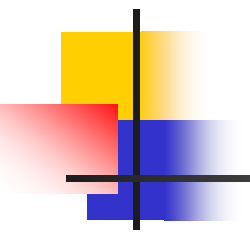


# Issues

---

- Computational Complexity
- Sampling Issues:

	Finite	Countable	Uncountable
Realizable	$\frac{1}{\epsilon} \ln \frac{ \mathcal{H} }{\delta}$	Nested Class	VC dim
Agnostic	$O\left(\frac{1}{\epsilon^2} \ln \frac{ \mathcal{H} }{\delta}\right)$	-	VC dim



# Learning = Estimation + Optimization

---

1. Acquire required relevant information by examining enough labeled samples
2. Find hypothesis  $h \in H$  consistent with those samples
  - . . . often "smallest" hypothesis
  - Spse  $H$  has  $2^k$  hypotheses  
Each hypothesis requires  $k$  bits
    - $\Rightarrow \log |H| \approx |h| = k$
    - $\Rightarrow$  SAMPLE COMPLEXITY not problematic
  - But optimization often is. . . intractable!
    - Eg, consistency for 2term-DNF is NP-hard, . . .
  - Perhaps find best hypothesis in  $F \supset H$ 
    - 2-CNF  $\supset$  2term-DNF
    - . . . easier optimization problem!



# Extensions to this Model

---

- Ockham Algorithm: Can PAC-learn  $H$  iff
  - can “compress” samples
  - have efficient consistency-finding algorithm

- Data Efficient Learner

Gathers samples sequentially, autonomously decides when to stop & return hypothesis

- Exploiting other information

- Prior background theory
- Relevance

- Degradation of Training/Testing Information

$\left\{ \begin{array}{l} \text{Errors} \\ \text{Omissions} \end{array} \right\}$  in  $\left\{ \begin{array}{l} \text{Training} \\ \text{Testing} \end{array} \right\}$   $\left\{ \begin{array}{l} \text{Attribute Value} \\ \text{Class Label} \end{array} \right\}$



# Other Learning Models

---

- Learning in the Limit [Recursion Theoretic]
  - Exact identification, no resource constraints
- On-Line learning
  - After seeing each unlabeled instance,
  - learner returns (proposed) label
  - Then correct label provided (learner penalized if wrong)
  - Q: Can learner converge, after making only  $k$  mistakes?
- Active Learners
  - Actively request useful information from environment
  - “Experiment”
- “Agnostic Learning”
  - What if target  $\neg[f \in H]$  ?
  - Want to find CLOSEST hypotheses. . .
  - Typically NP-hard. . .
- Bayesian Approach: Model Averaging, . . .

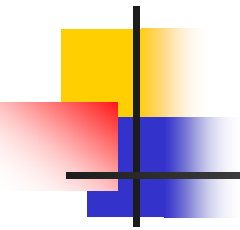


# Computational Learning Theory

---

- Inductive Learning is possible
  - With caveats: error, confidence
  - Depends on complexity of hypothesis space
- Probably Approximately Correct Learning
  - Consistency Filtering
  - Sample Complexity
  - Eg: Conjunctions, Decision\_Lists
- Many other meaningful models







# Terminology

---

- **Labeled example:** Example of form  $\langle x, f(x) \rangle$
- **Labeled sample:** Set of  $\{ \langle x_i; f(x_i) \rangle \}$
- **Classifier:** Discrete-valued function.  
Possible values  $f(x) \in \{ 1, \dots, K \}$  called "classes";  
"class labels"
- **Concept:** Boolean function.
  - $x$  s.t.  $f(x) = 1$  called "positive examples"
  - $x$  s.t.  $f(x) = 0$  called "negative examples"
- **Target function** (target concept): "True function"  $f$  generating the labels
- **Hypothesis:** Proposed function  $h$  believed to be similar to  $f$ .
- **Hypothesis Space:** Space of all hypotheses that can, in principle, be output by a learning algorithm