B: Ch 7.1.5
RN, Chapter 18.5

Computational Learning Theory

Computational Learning Theory

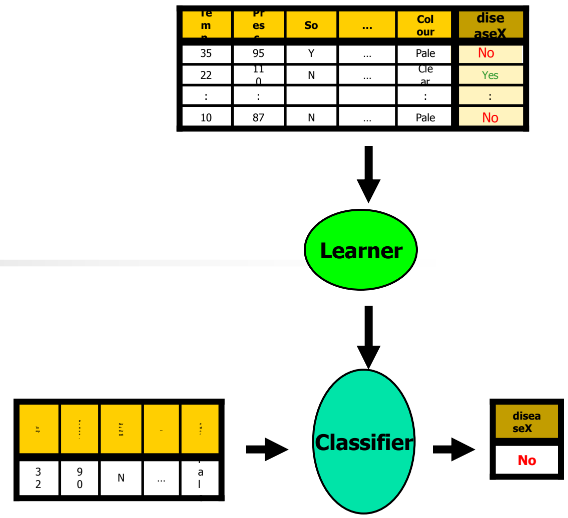
- Inductive Learning
  - Protocol
  - Error
- Probably Approximately Correct Learning
  - Consistency Filtering
  - Sample Complexity
  - Eg: Conjunction, Decision List
- Issues
  - Bound
  - Other Models

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(…) !

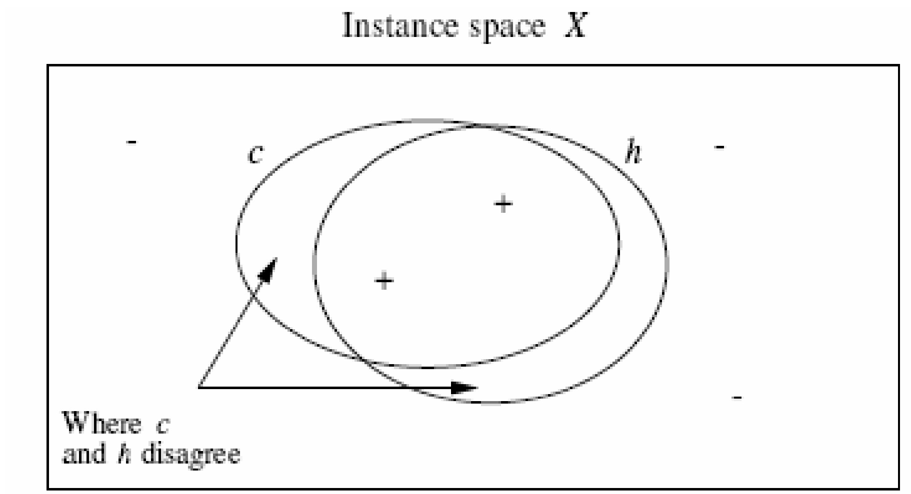
Protocol



- 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 $c(.)$, 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

≡ probability that h will misclassify instance drawn from D

$$err_D(h) = Pr_{x \in D}[\, c(x) \neq h(x) \,]$$

Probably Approximately Correct

Goal:

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

$err_D(\hat{h}) \approx 0$

with high probability

$P(\ err_D(\hat{h}) \approx 0\ ) \approx 1$

- Double “hedging"
  - approximately
  - probably

  Need both!

PAC-Learning

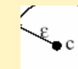
- Learner L can draw labeled instance $\langle x, c(x) \rangle$ in unit time
  $x \in X$ drawn from distribution D labeled by target concept $c \in C$

Def'n: Learner L PAC-learns class C (by H)
if
1. for any target concept $c \in C$,
   any distribution D, any ε, $\delta > 0$,
   L returns $h \in H$ s.t.
   w/ prob. $\geq 1 - \delta$, $err_D(h) < \varepsilon$
2. L's run-time (and hence, sample complexity)
   is $poly(|x|, size(c), 1/\varepsilon, 1/\delta)$

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

  Often $C = H$

Simple Learning Algorithm: Consistency Filtering



- Draw $m_H(\varepsilon, \delta)$ 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: $m_H(\varepsilon, \delta)$
- Q2: Need to decide if $h \in H$ is consistent w/ all S_m

  ... efficiently ...

Boolean Functions (≡ 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

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

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

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

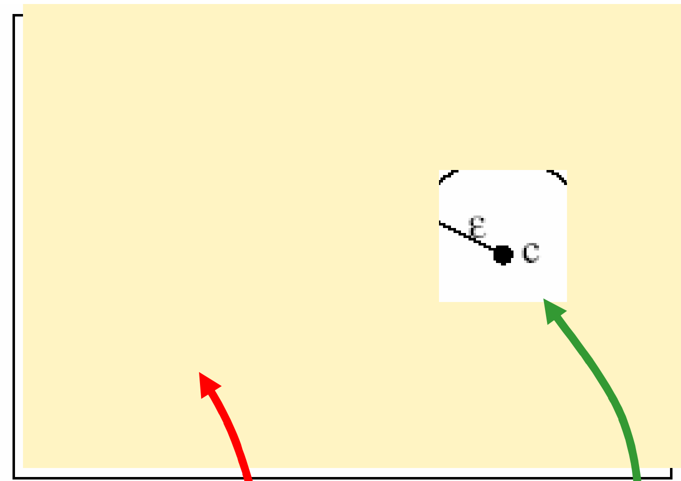
Labeled-tuple is $\left\{ \begin{array}{c} \textit{Consist} \\ \textit{InConsistent} \end{array} \right\}$ w/ hyp.

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

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

Consistent with $h_{X_2 \vee X_3}$

Bad Hypotheses



Idea: Find $m = m_H(\varepsilon, \delta)$ s.t.

after seeing *m* examples,

every BAD hypothesis h ($err_{D,c}(h) > \varepsilon$)

will be ELIMINATED

with high probability ($\approx 1 - \delta$)

leaving only good hypotheses

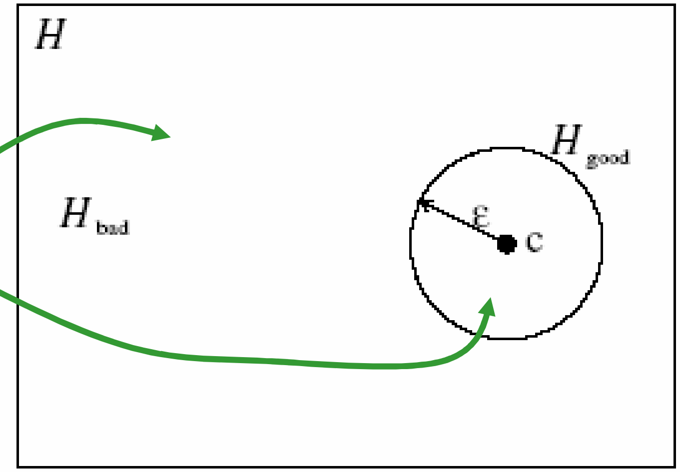
Eliminate
$H_{bad} = \{ h \in H \mid err_D(h) > \varepsilon \}$

Leave
$H_{Good} = \{ h \in H \mid err_D(h) \leq \varepsilon \}$

… then pick ANY of the remaining good ($err_{D,c}(h) < \varepsilon$) hyp's

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

$$\mathcal{H}_{bad} = \{ h \in \mathcal{H} \mid err_{\mathcal{D}}(h) > \epsilon \}$$
$$\mathcal{H}_{good} = \{ h \in \mathcal{H} \mid err_{\mathcal{D}}(h) \leq \epsilon \}$$
Note $|\mathcal{H}_{Bad}| \leq |\mathcal{H}|$



		x_1	x_2	$\cdots$	x_m	Good/Bad
	h_1	$\surd$	$\surd$	$\cdots$	$\surd$	$< \epsilon$
	h_2	$\surd$	$-$	$\cdots$	$\surd$	$< \epsilon$
	$\vdots$			$\vdots$		$\vdots$
$\mathcal{H}$	h_k	$\surd$	$\surd$	$\cdots$	$-$	$< \epsilon$
	$h_{bad,1}$	$-$	$-$	$\cdots$	$\surd$	$> \epsilon$
	$\vdots$			$\vdots$		$\vdots$
	$h_{bad,r}$	$-$	$-$	$\cdots$	$-$	$> \epsilon$

Ok

Bad

$\surd: \quad h_i(x_j) = c(x_j)$

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

Sample Bounds – Derivation

- Let h_1 be ε-bad hypothesis … $\text{err}(h_1) > \varepsilon$

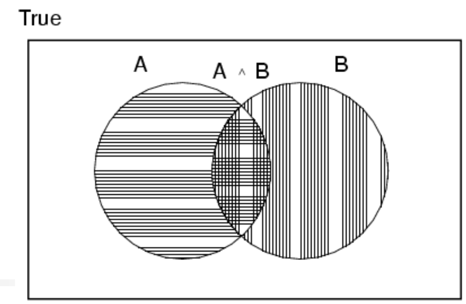
$\Rightarrow$ h_1 mis-labels example w/prob $P(h_1(x) \neq c(x)) > \varepsilon$

$\Rightarrow$ h_1 **correctly** labels random example w/prob $\leq (1 - \varepsilon)$

- As examples drawn INDEPENDENTLY

$P(h_1$ correctly labels *m* examples $) \leq (1 - \varepsilon)^m$

Sample Bounds – Derivation II



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

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

- If k ε-bad hypotheses $\{h_1, \ldots, h_k\}$:

$$P(h_1 \vee \ldots \vee h_k \text{ survives}) \leq k\,(1-\varepsilon)^m$$

Sample Bounds – Derivation

- Let h_1 be ε-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

  $P(h_1 \text{ correctly labels m examples}) \leq (1-\varepsilon)^m$
- Let h_2 be another ε-bad hypothesis
- What is probability that either h_1 or h_2 survive m random examples?

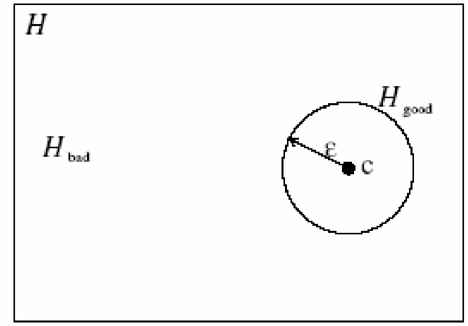
  $P(h_1 \vee h_2 \text{ survives})$

  $= P(h_1 \text{ survives}) + P(h_2 \text{ survives}) - P(h_1 \text{ \& } h_2 \text{ survives})$

  $\leq P(h_1 \text{ survives}) + P(h_2 \text{ survives})$

  $\leq 2(1-\varepsilon)^m$

Sample Bounds, con't



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

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

  P(any h_b in H_{bad} is consistent with *m* exs.)

  $\leq |H_{bad}| (1 - \varepsilon)^m \leq |H| (1 - \varepsilon)^m$

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

  $\Rightarrow$

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

- $m_H(\varepsilon, \delta)$ 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: ε
  - $err_{D,c}(h) = P(\ h(x) \neq c(x)\ ) \leq \varepsilon$
- Confidence of being ε-close: δ
  - $P(\ err_{D,c}(h) \leq \varepsilon\ ) > 1 - \delta$
- Sample size: $m_H(\ \varepsilon,\ \delta\ )$

- Any hypothesis consistent with

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

examples,

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

Boolean Function... Conjunctions

- Boolean Instance: $\langle x_1, \ldots, x_n \rangle$

  $\langle 1, 0, 1, 1 \rangle$ for $\langle x_1 = 1,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1 \rangle)$

- Boolean Function: $f(\langle x_1, \ldots, x_n \rangle) \in \{0, 1\}$

- Conjunction (type of Boolean function)

$$f_{+-0-0+}(X) \quad = \quad x_1\, \bar{x_2}\, \bar{x_4}\, x_6$$

$$= \begin{cases} 1 & \text{if } x_1(X) = t,\ x_2(X) = f,\ x_4(X) = f, \\ & \quad \text{and } x_6(X) = t \\ 0 & \text{otherwise} \end{cases}$$

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

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

(Ie, must match each literal mentioned)

- Only 3^n possible conjunctions

  out of 2^{2^n} boolean functions!

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

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

Alg:

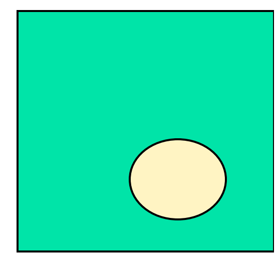
> Collect $m_{\mathcal{H}_C}(\epsilon, \delta) = \frac{1}{\epsilon}\left[n \ln 3 + \ln \frac{1}{\delta}\right]$ labeled samples
> Let $h = x_1 \bar{x}_1 x_2 \bar{x}_2 \cdots x_n \bar{x}_n$
> For each +-example $y = \bigwedge_i \pm_i x_i$
> Remove from h any literal NOT included in y

Data	Current Hyp					
	x_1	$\bar{x}_1$	x_2	$\bar{x}_2$	x_3	$\bar{x}_3$
$\langle\langle 1 \; 0 \; 1 \rangle +\rangle$						

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 poly(n, 1/ε, 1/δ) steps
  $\Rightarrow$ algorithm is *efficient!*
- Does NOT explicitly build all 3^n conjunctions, then throw some out…

PAC-Learning k-CNF

- $CNF \quad \equiv \quad$ Conjunctive Normal Form

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

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

  1-CNF $\equiv$ Conjunctions

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

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

$$\binom{n}{k} = O\left(n^k\right)$$

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

Alg: Consistency Filtering:

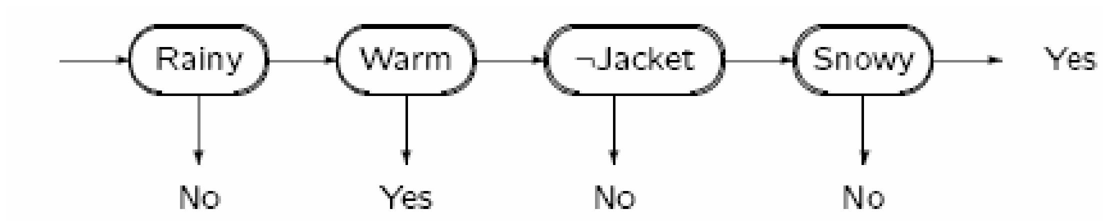
> Let T = all $O(\binom{n}{k} 3^k)$ possible k-clauses.
> After each $+$-example y,
>   Remove from T all clauses INCONSISTENT w/ y
> Return $\bigwedge T$

- 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 $|H_{DL}| \leq (4n)!$

  (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 $h = \langle \neg \mathbf{x_1} \mapsto \mathbf{B} \rangle$
   Eliminate i_2, i_4
2. When $x_2 = 1$, class is "A"
   Form $h = \langle \neg \mathbf{x_1} \mapsto \mathbf{B}; \mathbf{x_2} \mapsto \mathbf{A} \rangle$
   Eliminate i_3, i_5
3. When $x_4 = 1$, class is "A"
   Form $h = \langle \neg \mathbf{x_1} \mapsto \mathbf{B}; \mathbf{x_2} \mapsto \mathbf{A}; \mathbf{x_4} \mapsto \mathbf{A} \rangle$
   Eliminate i_1
4. Always have class "B"
   Form $h = \langle \neg \mathbf{x_1} \mapsto \mathbf{B}; \mathbf{x_2} \mapsto \mathbf{A}; \mathbf{x_4} \mapsto \mathbf{A}; \mathbf{t} \mapsto \mathbf{B} \rangle$
   Eliminate rest (i_6)

PAC-Learning Decision Lists

Let: S = set of

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

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 ≥ 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)

- "Covering" Algorithm

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 L satisfied by ≥ 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 ≥ 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, δ
     $h(x)$ is probably correct on x
2. If example x appears rarely ε
   ($P(x) \approx 0$)
   then h suffers only small penalty for being wrong.

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

Why is Bound so Lousy!

- Assumes error of all ε-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 $|H_{bad}| = |H|$ ... 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 (ε-cover)
    Gaussian, . . .
  - Look at samples! $\Rightarrow$ Sequential PAC Learning

If λ-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 (ε) 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	$\frac{1}{\varepsilon}\ln\frac{\lvert H\rvert}{\delta}$	Nested Class	VC dim
Agnostic	$O\left(\frac{1}{\varepsilon^2}\ln\frac{\lvert H\rvert}{\delta}\right)$	–	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

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

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 $f(x) \in \{ 1, \ldots, K \}$ 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

Computational Learning Theory

- Framework/Protocols

1. Finite $\mathcal{H}$, Realizable case
2. Finite $\mathcal{H}$, Unrealizable case
3. Infinite $\mathcal{H}$
   (Vapnik-Chervonenkis Dimension)
4. Variable size Hypothesis Space
5. Data-dependent Bounds
   (Max Margin)

- Topics:
  - Extensions to PAC
  - Other Learning Models
  - Occam Algorithms

6. Mistake Bound (Winnow)

Case 2: Finite $\mathcal{H}$, Unrealizable

- What if perfect classifier $\notin$ hyp. space $\mathcal{H}$?
  - either none exists (data inconsistent) or
  - hypothesis space is restricted
- Let: $h^* = \text{argmin}_{h\in\mathcal{H}} \{ \text{err}_D(h) \}$ be optimal $h\in\mathcal{H}$
- Want: $\hat{h}$ s.t. $\text{err}_D(\hat{h}) \leq \text{err}_D(h) + \varepsilon$
- Alg:

  Draw $m = m(\varepsilon,\delta)$ instances S
  Return $\hat{h} = \text{argmin}_{h\in\mathcal{H}} \{ \underline{\text{err}}_S(h) \}$
  with optimal empirical score, over S

  ($\underline{\text{err}}_S(h) = 1/m \sum_{x\in S} \text{err}(h, x)$ is EMPIRICAL score)
- Issues:
  - 1. How many instances?
  - Computational cost of $\text{argmin}_{h\in\mathcal{H}} \{ \underline{\text{err}}_S(h) \}$

Sample Complexity

Goal: Want enough instances that, w/prob $\geq 1-\delta$

$\boxed{\hat{h} = \text{argmin}_{h\in\mathcal{H}}\{\, \underline{err}_S(h) \,\} \text{ is within } \varepsilon \text{ of } h^* = \text{argmin}_{h\in\mathcal{H}}\{\, err_D(h) \,\}}$

- Step1: Sufficient to estimate ALL h's to within $\varepsilon/2$.

  $$|\, err_D(h) - \underline{err}_S(h)| \leq \varepsilon/2$$

  If so, then

  $$e_D(\hat{h}) - e_D(h^*)$$
  $$= e_D(\hat{h}) - \underline{e}_S(\hat{h}) + \underline{e}_S(\hat{h}) - \underline{e}_S(h^*) + \underline{e}_S(h^*) - e_D(h^*)$$
  $$\leq \quad \varepsilon/2 \quad + \quad 0 \quad + \quad \varepsilon/2 \quad = \varepsilon$$

Sample Complexity, con't

Goal: Want enough instances that, w/prob $\geq 1 - \delta$

$\hat{h} = \text{argmin}_{h \in \mathcal{H}} \{ \underline{err}_S(h) \}$ is within ε of $h^* = \text{argmin}_{h \in \mathcal{H}} \{ err_D(h) \}$

- Step2: Sufficient to estimate EACH h's to within $\varepsilon/2$ with prob $\geq 1 - \delta / |\mathcal{H}|$

  If so, then

  $$P(\exists\, h \in \mathcal{H} \mid err_D(h) - \underline{err}_S(h)| \leq \varepsilon/2)$$
  $$\leq \sum_{h \in \mathcal{H}} P(err_D(h) - \underline{err}_S(h)| \leq \varepsilon/2)$$
  $$\leq |\mathcal{H}| \quad \delta / |\mathcal{H}| = \delta$$

- Step3: How many instances s.t.

  $P(err_D(h) - \underline{err}_S(h)| \leq \varepsilon/2) \leq \delta / |\mathcal{H}|$?

Complexity of "Agnostic Learning"

- **Sample Complexity**: Good news!
- Hoeffding Inequality ⇒ Need only $m(\varepsilon,\delta) = \dfrac{2}{\varepsilon^2}\ln\dfrac{2|H|}{\delta}$

  instances to estimate EACH h's to within $\varepsilon/2$
  with prob $\geq 1 - \delta / |\mathcal{H}|$

  $P(err_D(h) - \underline{err_S}(h)| \leq \varepsilon/2)$

  $\leq 2 \exp(-2\ m\ (\varepsilon/2)^2) \leq \delta / |\mathcal{H}|$
- **Computational Complexity**: Bad news!

  NP-hard to find

  CONJUNCTION $h \in \mathcal{H}$ that is BEST FIT to DNF $c \in C$

  (target space = DNF; hypothesis space = Conjunctions)
- Note: Sample size typically poly;
  Hardness tends to be Consistency/Optimization

Case 3: ∞ Hypothesis Spaces ⇒ VC Dim

- Learning an initial subinterval.
  "Factory ok iff Temperature ≤ a"
  for some (unknown) $a \in [0, 100]$
  ⇒ target concept is some initial interval
  $C = H = \{ [0, a] \mid a \in [0, 100] \}$

Observe M instances Return [0, b] , where b is largest positive example seen.

- Clearly poly time per example.
  How many examples?

Sample Complexity of Learning Initial Segment

- Approach#1: Use $m_H(\varepsilon,\delta) = \frac{1}{\varepsilon}\left(\log\frac{|H|}{\delta}\right)$ instances ?

  But $\mathcal{H}$ is UNCOUNTABLE!

- Approach#2:
  - Let a_ε be real value < a s.t. $[a_\varepsilon, a]$ has probability ε

- Alg succeeds *iff* it sees example in $[a_\varepsilon, a]$

  P(failure) = P(none of M examples in $[a_\varepsilon, a]$) = $(1 - \varepsilon)^M$

  So for P(failure) $\leq \delta$, need

$$\boxed{M \geq \frac{1}{\varepsilon}\ln\frac{1}{\delta}}$$

Uniform Convergence

- Simultaneously estimating all $\{ [a_\varepsilon , a] \mid a \in [0, 100] \}$!

Q: Why possible?

A: Only one "degree of freedom"

⇒ each sample provides LOTS of information about many hypothesis

Q: How much is a degree of freedom worth?

Are they all worth the same?

A: Look at "effective number" of concepts,
as fn of number of data points seen.

Only grows linearly....

- Number of "effective degrees of freedom":

called "VC-dimension"

Shattering a Set of Instances

- Hypothesis class $\mathcal{H}$ trivially fit

  $\mathbf{X} = \{x_1, \ldots, x_k\}$

  if

  $\forall$ labeling of examples in **X**,

  $\exists\, h \in \mathcal{H}$ matching labeling

$$\begin{array}{c|ccccc} & h_1 & h_2 & \cdots & h_{2^k-1} & h_{2^k} \\ \hline \mathrm{x}_1 & 0 & 0 & \cdots & 1 & 1 \\ \mathrm{x}_2 & 0 & 0 & \cdots & 1 & 1 \\ \vdots & \vdots & & & \vdots & \vdots \\ \mathrm{x}_k & 0 & 1 & \cdots & 0 & 1 \end{array}$$

- k instances; $|\mathcal{H}| \geq 2^k$

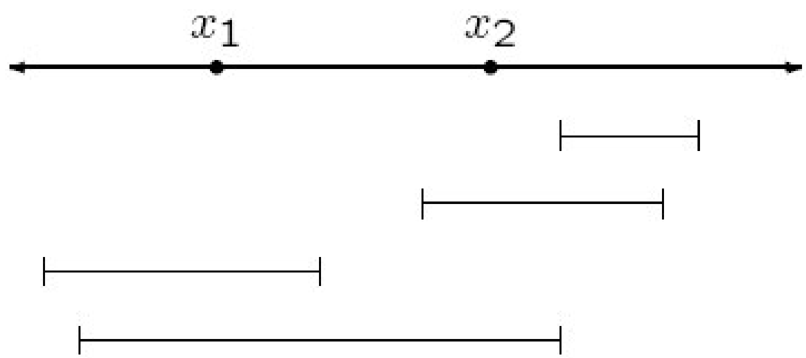
  Any subset of size k – 1 is unconstrained!

- Defn: Set of points $\mathbf{X} = \{x_i\}$ is shattered by hypothesis class $\mathcal{H}$ if

  $\forall\, S \subset X, \exists\, h_S \in \mathcal{H}$ s.t.
  - $h_S(x) = 1 \quad \forall\, x \in S$
  - $h_S(x) = 0 \quad \forall\, x \notin S$

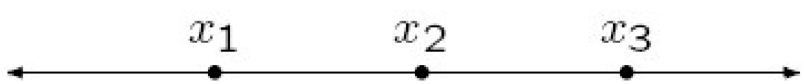
Example of Shattering

- $\mathcal{H} = \{ [a, b] \mid a < b \}$ = intervals on real line
- Can shatter (any!) 2 points:



x_1	x_2
0	0
0	1
1	0
1	1

- $\exists$ 3 points that can NOT be shattered:



x_1	x_2	x_3
1	0	1

Vapnik-Chervonenkis Dimension

- *Def'n*: VCdim of concept class $\mathcal{H}$

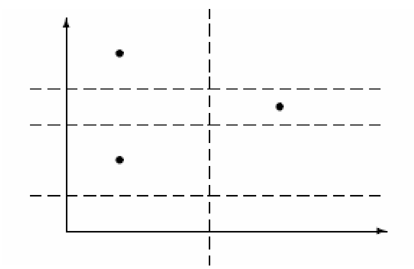
  $\equiv$ largest # of points shattered by $\mathcal{H}$
  - If arbitrarily large finite sets of **X** shattered by $\mathcal{H}$, then $\text{VCdim}(\mathcal{H}) = \infty$
  - $\text{VCdim}(\mathcal{H}) = d \quad \Leftrightarrow$

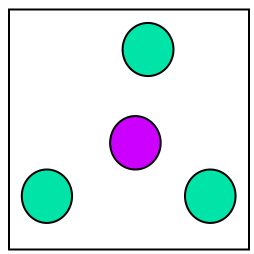
    $\exists$ set of d points that can be shattered,
    but no set of d+1 points can be shattered
- Note: $\text{VCdim}(\mathcal{H}) \leq \log_2 | \mathcal{H} |$
- $\text{VCdim}(\mathcal{H})$ measures complexity of $\mathcal{H}$

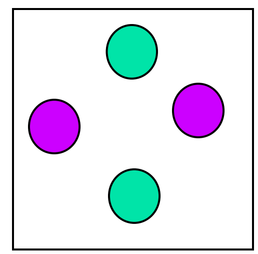
  ... how many distinctions can its elements exhibit

VC-dimension: Linear Separator

- $\mathcal{H}_{\mathcal{LS}2} = \{ [w_0, w_1, w_2] \in \Re^3 \}$
  = linear separators in 2-D
- Trivial to fit (any non-linear!) 3 points







- But cannot shatter ANY set of 4 points
  - If one point inside convex hull of others, can not make outsides " –" and inside "+"
  - Otherwise, label alternatingly in cycle

$\Rightarrow$ VC ($\mathcal{H}_{\mathcal{LS}2}$)=VC(LinearSeparator in 2Dim) = 3

Some VC Dims

- VCdim(LinearSeparator in k-Dim) = k +1

- Multi-layer perceptron network over n inputs of depth s:
  $d \leq 2(n+1)s(1+\ln s)$
- Exact value for sigmoid units is ?unknown?
  … probably slightly larger...
- Typically VCdim(model) ≈
  # of non-redundant tunable parameters

VCdim of . . .

- H_{int} = { intervals of real line }
- H_{box} = {axis-parallel boxes in 2-D}
- H_{md} = {monotone disjunctions (n features)}
- H_{all} = {all functions on n features }

How does VCdim measure Complexity?

- Def'n: $\mathfrak{H}[m]$ = maximum number of ways to split m points using concepts in $\mathfrak{H}$
- For $m \leq VCdim(\mathfrak{H})$, $\mathfrak{H}[m] = 2^m$

  For $m \geq VCdim(\mathfrak{H})$, . . .
- Theorem: $\mathfrak{H}[m] = O(m^{VCdim(\mathfrak{H})})$
  - Ie, only $C[m]$ "different" concepts in $\mathfrak{H}$ wrt any set of m examples.

$\Rightarrow^?$ Replace $\ln(|\mathfrak{H}|)$ by $\ln(\mathfrak{H}[m])$ in PAC bounds

YES (kinda)! . . . but NOT OBVIOUS, since different data $\Rightarrow$ different concepts

Upper/Lower Bounds using VCdim

- **Theorem 1**: Given class C,
  for any distribution D, target concept in C,
  given a sample size:

  $$\boxed{\frac{1}{\epsilon}\left(4\log_2\left(\frac{2}{\delta}\right) + 8\,\mathrm{VCdim}(\mathcal{C})\log_2\left(\frac{13}{\epsilon}\right)\right)}$$

  then with prob $\geq 1 - \delta$,
  any consistent $h \in C$ has error $\leq \varepsilon$.

- **Theorem 2:** If $|C| \leq 2$, then
  for any learning alg A,
  $\exists$ distribution D over X, distribution over C s.t.
  expected error of A is $> \varepsilon$
  if A sees sample of size under $\boxed{\dfrac{\mathrm{VCdim}(\mathcal{C}) - 1}{32\epsilon}}$

Comments on VC Dimension

- VCdim provides good measure of complexity of class:

  Upper/Lower (worst case) bounds:

$$\boxed{\tilde{\Theta}(\, VC\dim(C)\,)}$$

- Does this mean. . .
  - ... can't learn classes of infinite VCdimension?

    A: No: just use poly dependence on size(c)
  - ... complicated hypotheses are bad?

    A: No. Just need a lot of data to learn complicated concept classes...

Proof of Theorem#2 (Sketch)

- Theorem 2: ... need at least $m = \frac{\text{VCdim}(\mathcal{C}) - 1}{8\epsilon}$

  (#examples needed for uniform convergence
  . . . for all bad $h \in C$ to look bad . . .)

Proof: Consider $d = \text{VCdim}(C)$ points $\{ x_1, x_2, \ldots, x_d \}$ that can be shattered by target concepts $\{ c_i \}_{i=1}^{2^k}$

- Define distribution D:
  - $1 - 4\varepsilon$ on x_1
  - $4\varepsilon / (d - 1)$ on each other
- Given m instances, expect to see only ½ of $\{ x_2, \ldots, x_d \}$ so $E[\#notSeen] \geq (d - 1) / 2$
- As can only do 50/50 on instances NOT seen, expected error is $\#notSeen \; ½ \; 4\varepsilon / (d - 1) = \varepsilon$

Summary of Training vs Test Error

- ϵ = "true" error of hyp h
  ϵ^* = minimum true error of any member of $\mathcal{H}$
  ϵ_T = "training set" error of hyp h

- After m examples, w/ probability $\geq 1-\delta$, ...
  - Finite Hypothesis Class; "Realizable"
  $$\epsilon \quad \leq \quad \frac{1}{m}\left[\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right]$$
  - Finite Hypothesis Class; "UnRealizable"
  $$\epsilon \quad \leq \quad \epsilon^* + \sqrt{\frac{1}{2m}\left[\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right]}$$
  - $d = \text{VCdim}(\mathcal{H})$
  $$\epsilon \quad \leq \quad 2\epsilon_T + \frac{4}{m}\left[d\log\frac{2e\,m}{d} + \ln\frac{4}{\delta}\right]$$

Case 4: Why SINGLE Hypothesis Space?

- Large $\mathfrak{H}$ is likely to include (approx to) target c

  but . . .
- w/few examples, cannot reliably find good hypothesis from large hypothesis space
- That is...
  - Underfitting: Every $h \in \mathfrak{H}$ has high ε_T

    $\Rightarrow$ consider larger hypothesis space $\mathfrak{H}' \supset \mathfrak{H}$
  - Overfitting: Many $h \in \mathfrak{H}$ have $\varepsilon_T \approx 0$

    $\Rightarrow$ consider smaller $\mathfrak{H}'' \subset \mathfrak{H}$ to get lower *d*

$\Rightarrow$ To learn effectively ($> 1 - \varepsilon$) from m instances, only consider $\mathfrak{H}$ s.t. $| \mathfrak{H} | \approx e^{\varepsilon m}$

How Learning Algorithms Manage This Tradeoff

S1: Start with small hypothesis space $\mathcal{H}_1$

S2: Grow hypothesis space

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \subset \ldots$$

until finding a good (nearly consistent) hypothesis

Eg1 $\mathcal{H}_1 =$ "leaf", then
$\mathcal{H}_2 =$ "one DecTree node", then
$\mathcal{H}_3 =$ "two DecTree nodes", then ...

Eg2 $\mathcal{H}_1 =$ "constants", then
$\mathcal{H}_2 =$ "linear functions", then
$\mathcal{H}_3 =$ "quadratic functions", then ...

Approaches

1. Easy: $\bigcup_i \mathcal{H}_i$ countable, and realizable
2. General: Structural Risk Minimization
3. "Occam Algorithms"

#4a: Dealing w/∞ Set of Hypotheses

- Incremental algorithms:

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \ldots \subset \mathcal{H}_n \subset \ldots$$

$$1-DNF \;\subset\; 2-DNF \;\subset\; 3-DNF \;\subset\; \ldots$$

Assume: $m(\mathcal{H}_i, \epsilon, \delta)$ instances sufficient to PAC(ϵ, δ)-learn $\mathcal{H}_i$

Alg? *Assume target in* $\mathcal{H}_1$

Draw $m(\mathcal{H}_1,\ \epsilon,\ \delta\ )$ instances

Stop if find good $h_1 \in \mathcal{H}_1$

Otherwise. . .

Assume target in $\mathcal{H}_2$

Draw $m(\mathcal{H}_2,\ \epsilon,\ \delta\ )$ more instances

Stop if find good $h_2 \in \mathcal{H}_2$

Otherwise. . .

. . .

Assume target in $\mathcal{H}_i$

Draw $m(\mathcal{H}_i,\ \epsilon,\ \delta\ )$ more instances

Stop if find good $h_i \in \mathcal{H}_i$

Otherwise. . .

. . .

Correct Algorithm?

- Q: Suppose find "good" h_k at iteration k.
  What is prob of making mistake?
- A: $P(\text{ mistake }) = \sum_{i=1..k} P(\text{ mistake @ iteration } i\ )$
  $\leq \sum_{i=1..k} \delta \leq k\,\delta$

$\Rightarrow$ Need to use δ_i s.t. $\sum_{i=1..k} \delta_i \leq \delta$ for any k

- Eg: $\delta_i = \delta/2^i$
  - Note: $P(\text{ mistake }) \leq \sum_{i=1..k} \delta_i = \delta \sum_{i=1..k} ½^{\,i} = \delta$

- Takes k bits to identify member of 2^k-size hypothesis space
  - . . . takes k bits just to express such a hypothesis

$\Rightarrow$ reasonable to allow learning alg'm time poly in
$1/\varepsilon$, $1/\delta$ and SIZE OF HYPOTHESIS

#4b: Structural Risk Minimization

- Consider
  - nested series: $\mathfrak{H}_1 \subset \mathfrak{H}_2 \subset \ldots \subset \mathfrak{H}_k \subset \ldots$
  - with VCdim: $d_1 \leq d_2 \leq \ldots \leq d_k \leq \ldots$
  - training errors: $\varepsilon_1 \geq \varepsilon_2 \geq \ldots \geq \varepsilon_k \geq \ldots$

- Choose $h_k \in \mathfrak{H}_k$ that minimizes

$$\boxed{\epsilon \quad \leq \quad 2\epsilon^k + \frac{4}{m}\left[d_k \log \frac{2e\,m}{d_k} + \ln \frac{4}{\delta}\right]}$$

Structural Risk Minimization

For $h \in \mathcal{H}$

$L(h)$	Probability of miss-classification
$\hat{L}_n(h)$	Empirical fraction of miss-classifications

Vapnik and Chervonenkis 1971: For **any** distribution with prob. $1-\delta$, $\forall h \in \mathcal{H}$,

$$L(h) < \underbrace{\hat{L}_n(h)}_{\text{emp. error}} + \underbrace{c\sqrt{\frac{\mathrm{VCdim}(\mathcal{H})\log n + \log\frac{1}{\delta}}{n}}}_{\text{complexity penalty}}$$

An Improved VC Bound II

Canonical hyper-plane:

$$\min_{1 \leq i \leq n} |\mathbf{w}^\top \mathbf{x}_i + b| = 1$$

(No loss of generality)

Improved VC Bound (Vapnik 95) VC dimension of set of canonical hyper-planes such that

$$\|\mathbf{w}\| \leq A$$

$$\mathbf{x}_i \in \text{Ball of radius } L$$

is

$$\boxed{\text{VCdim} \leq \min(A^2L^2, d) + 1}$$

Observe: Constraints reduce VC-dim bound

Canonical hyper-planes with **minimal norm** yields best bound

Suggestion: Use hyper-plane with minimal norm

Case 5: Data Dependent Bounds

- So far, bounds on depend only on
  - ε_T
  - quantities computed prior to seeing S
    (eg, size of $\mathfrak{H}$)

  $\Rightarrow$ "worst case"

  as must work for all but δ of possible training sets
- Data dependent bounds consider how h fits data
  - If S is not worst case training set

$\Rightarrow$ tighter error bound!

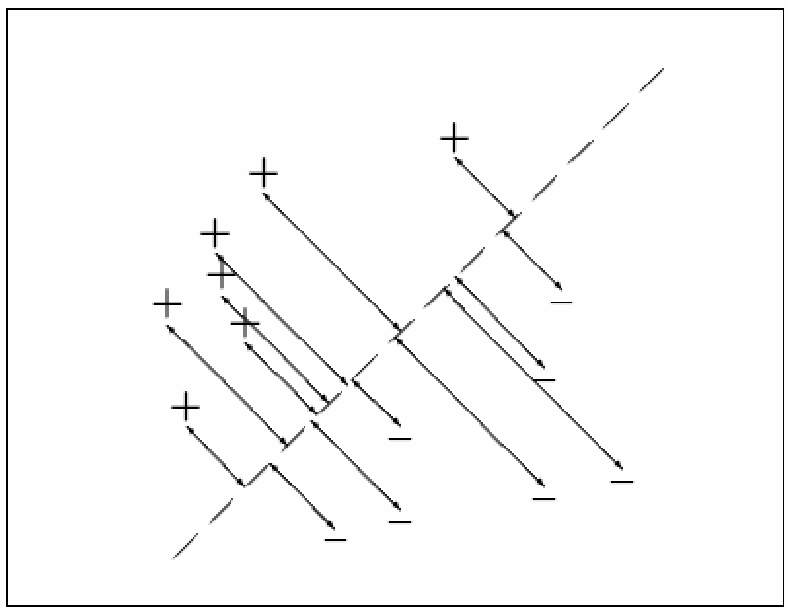
Margin Bounds

- $g(x)$ is real-valued function
  "thresholded at 0" to produce $h(x)$:

$$\begin{array}{lcl} g(x) > 0 & \Rightarrow & h(x) = +1 \\ g(x) < 0 & \Rightarrow & h(x) = -1 \end{array}$$

- **Margin** of $h(x)$ wrt S is

$$\gamma(g, S) \quad = \quad \min_i\{y_i\, g(x_i)\}$$

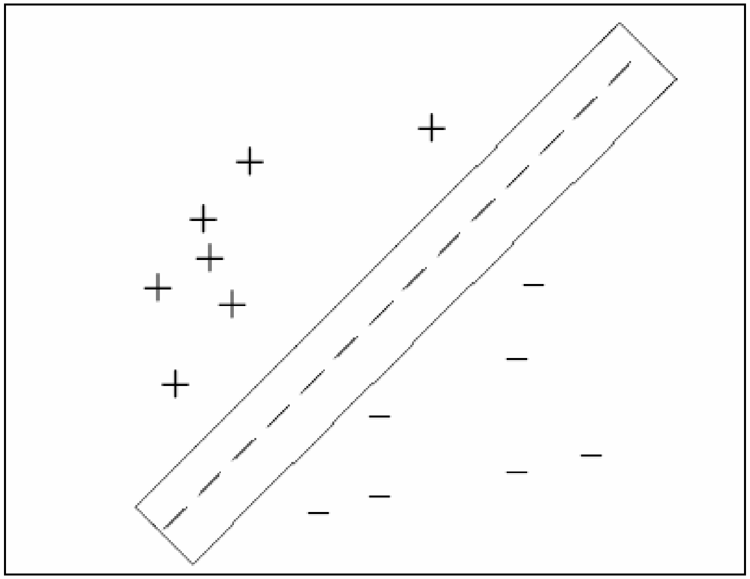


Margin Bounds: Key Intuition

Let $G = \{g(x)\}$ = set of real-valued functions that can be thresholded at 0 to give $h(x)$.

- Consider "thickening" each $g \in G$

  ...must correctly classify every point w/ margin $\geq \gamma$



- **fat shattering dimension**: $\text{fat}_\gamma(G)$

  $\equiv$ VCdim of these "fat" separators

Note $\text{fat}_\gamma(G) \leq \text{VCdim}(G)$

Noise Free Margin Bound

- Spse find $g \in G$

  with margin $\gamma = \gamma(g, S)$

  for a training set of size m

- Then, with probability $1 - \delta$

$$\epsilon \quad \leq \quad \frac{2}{m}\left[d \log \frac{2e\,m}{d\gamma} \log \frac{32m}{\gamma^2} + \log \frac{4}{\delta}\right]$$

$d = \text{fat}_{\gamma/8}(G)$ with margin $\gamma/8$

- Note fat.(G) kinda-like VCdim(G) !

Soft Margin Classification (2)

- Error rate of linear separator with
  unit weight vector and
  margin γ on training data
  lying in a sphere of radius R

  is, with probability $\geq 1 - \delta$,

$$\boxed{\epsilon \quad \leq \quad \frac{C}{m}\left[\frac{R^2 + \|\xi\|^2}{\gamma^2}\log^2 m + \log\frac{1}{\delta}\right]}$$

(constant C)

$\Rightarrow$ we should

- maximize margin γ
- minimize slack $\|\xi\|^2$

...see support vector machines!

Fat Shattering for Linear Separators: Noise-Free

Spse support for $P(\mathbf{x})$ within sphere of radius R

$$\|\mathbf{x}\| \leq R$$

$$G = \{\, g \mid g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} \;\&\; \|w\| = 1\}$$

Then $\text{fat}_\gamma(G) \;=\; \left(\frac{R}{\gamma}\right)^2$

$$\begin{aligned}\Rightarrow \quad \epsilon \quad &\leq \quad \frac{2}{m}\left[\frac{64R^2}{\gamma^2} \log \frac{em\gamma}{8R^2} \log \frac{32m}{\gamma^2} \;+\; \log \frac{4}{\delta}\right] \\ &\in \quad \tilde{O}\left(\frac{R^2}{m\gamma^2}\right)\end{aligned}$$

$\Rightarrow$ For fixed R, m:

seek g that maximizes γ !

maximum margin classifier

- Even with kernel $K(\cdot,\cdot)$... where $\|\mathrm{x}\| = \sqrt{K(\mathrm{x},\mathrm{x})}$

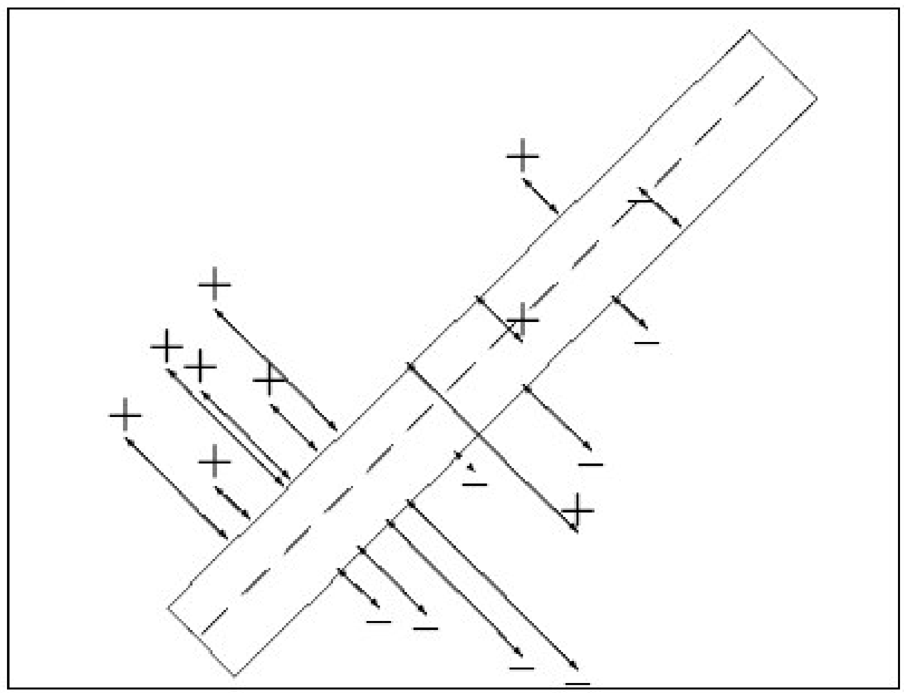
Soft Margin Classification

- Extension of margin analysis:
  When data is not linearly separable:

- $\xi_i \;=\; \max\{0,\ \gamma - y_i,\ g(\mathbf{x}_i\}$
  "*margin slack variable*" for $\langle \mathbf{x}_i, y_i \rangle$

  Note: $\xi_i \;> \gamma \quad \Rightarrow \mathbf{x}_i$ misclassified by h

- $\xi \;=\; \langle \xi_1, \ldots, \xi_m \rangle$
  "*margin slack vector* for h on S"



Irrelevant Features

- Consider learning $CD(n)$ = disjunction of n features
  "List-then-Eliminate" makes $O(n)$ mistakes
  - PAC-learning: $O(\, n/\varepsilon \log(1/\delta)\,)$
- Spse n is HUGE
  - Words in text
  - Boolean combination of "atomic" features
  - Features extracted in 480x560 image

  . . . but only $r \ll n$ features "relevant"
  - Eg: concept $x_4 \vee \neg x_{91} \vee \neg x_{203} \vee x_{907}$
- $\exists$ learning alg that makes $O(r \ln n)$ mistakes!
  "Winnow"

Winnow Algorithm

- Initialize weights $w_1, \ldots, w_n$ to 1
- Do until bored:
  - Given example $\mathbf{x} = [x_1, \ldots, x_n]$,
    If $w_1x_1 + w_2x_2 + \ldots + w_nx_n \geq n$
    output 1 otherwise 0
  - If mistake:
    - (a) If predicts 0 on 1-example, then
      for each $x_i = 1$, set $w_i := w_i \times 2$
    - (b) If predicts 1 on 0-example, then
      for each $x_i = 1$, set $w_i := w_i / 2$

Winnow's Effectiveness

Theorem Winnow MB-learns $CD(n)$, making at most $2+3r(1+\lg n)$ mistakes when target concept is disjunction of r var's.

Proof: 1. Any mistake made on 1-example must double

- ≥ 1 weights in target function (the relevant weights),
- & mistake on 0-example will not halve these weights.

- Each "relevant" weight can be doubled $\leq 1+\lg n$ times, since only weights $\leq n$ can be doubled.

  (Never double any weight $w_i > n$ as that weight alone $\Rightarrow$ class is 1)

$\Rightarrow$ Winnow makes $\leq r(1+\lg n)$ mistakes on 1-examples

2. Negative examples?

- Let sw_t be sum of weights $\sum w_i = n$, at time t.

  Initially $sw_0 = n$.

  Each mistake on 1-example increases sw by $\leq n$

  (. . . before doubling, we know $w_1x_1 + w_2x_2 + \ldots + w_nx_n < n$)

  Each mistake on 0-example decreases sw by $\geq n/2$

  (. . . before halving, we know $w_1x_1 + w_2x_2 + \ldots + w_nx_n \geq n$)

- As sw $\geq$ 0, number of mistakes made on 0-examples $\leq$ 2+ 2number of mistakes made on 1-examples.

- So total # of mistakes is $r(1+\ln n) + [2+2r(1+\lg n)]$

Incorporating Winnow Into PAC Model

- Given a MB(M)-learner, can PAC(ε,δ)-learn
  - Return any h_i that makes $\frac{1}{\epsilon}\log(\frac{M}{\delta})$ correct predictions
  - Requires $m = \frac{M}{\epsilon}\log(\frac{M}{\delta}) = \frac{r\log(n)}{\epsilon}\log(\frac{r\log(n)}{\delta})$ instances
- Better PAC-learner: $O(\ \frac{1}{\epsilon}[r\log(n) + \log(\frac{1}{\delta})]\ )$

1. Draw $m_1 = 4/\varepsilon \max\{ M, 2\ln(2/\delta) \}$ instances, S_1
2. Run Winnow (a MB-learner) on S_1, generating $\leq M$ hypotheses $H = \{ h_1, \ldots, h_M \}$
3. Draw $m_2 = O(\ 8/\varepsilon \log(2M/\delta)\ )$ more instances S_2
4. Use S_2 to find best hypothesis, h^* in H
5. Return h^*

- Why: Most ε-bad hypotheses have error >> ε
  ⇒ reveal "badness" in < $\frac{1}{\epsilon}\log(\frac{M}{\delta})$ instances

Proof

- m_1 guarantees that ≥ 1 of H is good
  m_2 distinguishes good h^* from bad members of H.

- After m_1 instances, ≥ 1 of H has error $\leq \varepsilon/2$
  PROOF: Spse first k – 1 hyp's all have error $> \varepsilon/2$,
  and h_k had error $\leq \varepsilon/2$
  What is prob that h_k occurs after m_1 instances?

Worst if k = M and each $err_D(h_i) = \varepsilon/2$
Chernoff bounds $\Rightarrow \delta/2$:

- Consider flipping (sequence of M) $\varepsilon/2$ weighted coins
- (each "head" $\equiv$ error)
- After m_1 flips, expect $m_1 \times \varepsilon/2 \leq 2M$ "heads"
- Prob of getting under M ($\leq$ ½ exp. number) heads
  $\leq P(Y_M \leq (1 - ½)\, \varepsilon/2) \leq \exp(-M\, \varepsilon/2\, ½)/2)$
  $\leq \exp(-M\, \varepsilon/8) \leq \delta$

Proof (II)

Use m_2, select h^* w/ $err_S(h^*) \leq \frac{3}{4}\,\varepsilon$

With prob $\geq 1 - \delta/2$ $err_D(h^*) \leq \varepsilon$

PROOF: Need to show $err_S(h_i)$

[average # mistakes made by h_i over m_2 samples]

is within 3/4 of $\mu_i = err_D(h_i)$

- $P(err_S(h_i) < err_D(h_i) \times (1 - 1/4)) \leq \exp(-(m_2\, \varepsilon\, \frac{1}{4})/2) \leq \delta / (2M)$
- So prob ANY $h_i \in H$ is off by < 3/4 is under $\delta/2$
- m_1 is leading term

  $\Rightarrow O(1/\varepsilon\ [r \log(n)+\log(1/\delta)])$

- Best known bound for learning *r* of *n* disjuncts!
- Note: Might NOT find 0 error r-disjunction. . .