

# Machine Learning

## Lecture 4

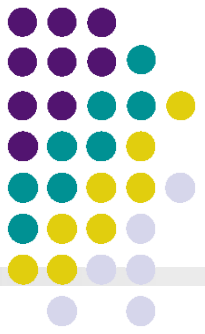
Yang Yang  
Department of Computer Science & Engineering  
Shanghai Jiao Tong University

# Computational Learning Theory

Reading: Andrew's lecture notes Part VI Learning theory  
《机器学习》 Chap. 12

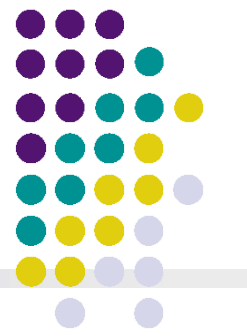
# Generalizability of Learning

---



- ❖ In machine learning it's really the generalization error that we care about, but most learning algorithms fit their models to the training set.
- ❖ Why should doing well on the training set tell us anything about generalization error? Specifically, can we relate error on training set to generalization error?
- ❖ Are there conditions under which we can actually prove that learning algorithms will work well?

# 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?
- ❖ What theory to relate:
  - Training examples
    - Quantity
    - Quality
    - How presented
  - Complexity of hypothesis / concept space
  - Accuracy of approx to target concept
  - Probability of successful learning

$m$

$H$

$\epsilon$

$\delta$

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

# Two Basic Competing Models



## PAC framework

Sample labels are consistent with some  $h$  in  $H$

Learner's hypothesis required to meet *absolute* upper bound on its error

## Agnostic framework

*No prior restriction on the sample labels*

*The required upper bound on the hypothesis error is only relative (to the best hypothesis in the class)*

# 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 $S = \{ \langle x^{(i)}, c(x^{(i)}) \rangle \}$

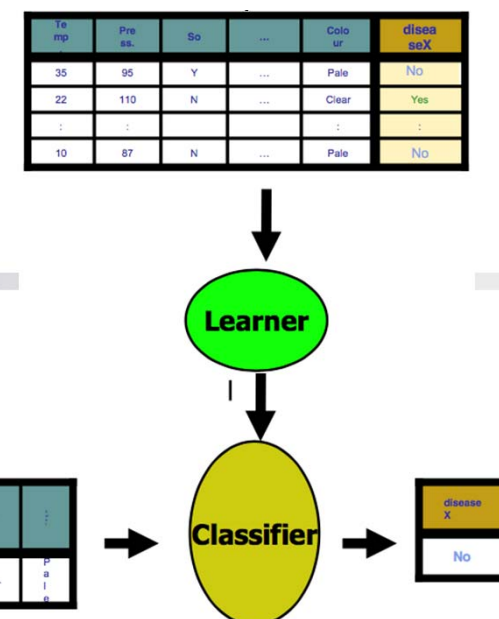
- instances  $x^{(i)}$  drawn from distribution  $D$
- labeled by target concept  $c \in C$

## ❖ Learner outputs $h \in H$ estimating $c$

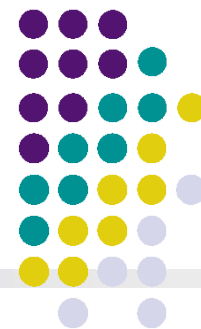
- $h$  is evaluated by performance on subsequent instances drawn from  $D$

## ❖ For now:

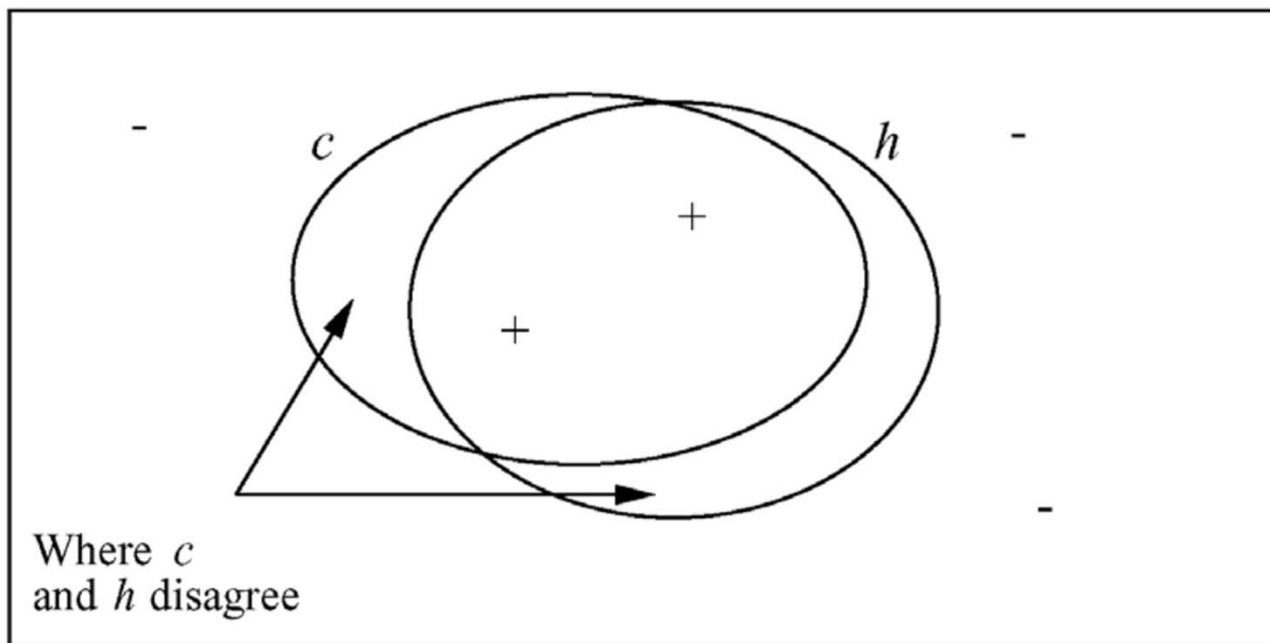
- $C = H$  ( so  $c \in H$  )
- noise-free data



# True Error of a Hypothesis



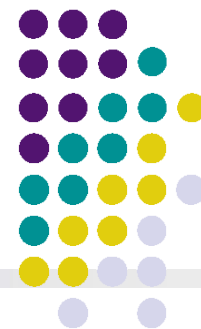
Instance space  $X$



- ❖ Definition: The **true error** (denoted  $\varepsilon_D(h)$ ) of hypothesis  $h$  with to target concept  $c$  and distribution  $D$  is the probability that  $h$  will misclassify an instance drawn at random according to  $D$ .

$$\varepsilon_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)]$$

# Two Notions of Error



- ❖ Training error ( a.k.a., empirical risk or empirical error ) of hypothesis  $h$  with respect to target concept  $c$ 
  - How often  $h(x) \neq c(x)$  over training instance from  $S$

$$\hat{\epsilon}_S(h) \equiv \Pr_{x \in S}[c(x) \neq h(x)] \equiv \frac{\sum_{x \in S} \delta(c(x) \neq h(x))}{|S|}$$

- ❖ True error of ( a.k.a., generalization error, test error ) hypothesis  $h$  with respect to  $c$ 
  - How often  $h(x) \neq c(x)$  over future random instances drew i.i.d. from  $D$

$$\epsilon_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)]$$

**Can we bound**

$$\hat{\epsilon}_D(h)$$

**in terms of**

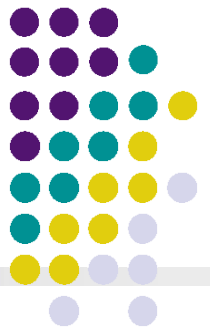
$$\hat{\epsilon}_S(h)$$

**??**



# The Union Bound

---

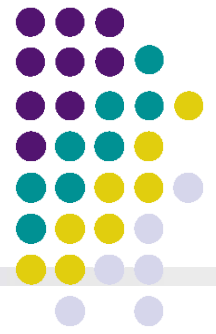


- ❖ Lemma. ( The union bound ). Let  $A_1, A_2, \dots, A_k$  be  $k$  different events ( that may not be independent ). Then

$$P(A_1 \cup A_2 \cup \dots \cup A_k) \leq P(A_1) + P(A_2) + \dots + P(A_k)$$

- In probability theory, the union bound is usually stated as an axiom ( and thus we won' t try to prove it ), but it also makes intuitive sense: The probability of any one of  $k$  events happening is at most the sums of the probabilities of the  $k$  different events.

# Hoeffding Inequality



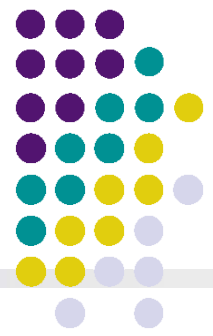
- ❖ Lemma. ( Hoeffding Inequality ) Let  $Z_1, \dots, Z_m$  be  $m$  independent and identically distributed ( iid ) random variables drawn from a Bernoulli( $\phi$ ) distribution, i.e.,  $P( Z_i = 1 ) = \phi$  and  $P( Z_i = 0 ) = 1 - \phi$

Let  $\hat{\phi} = \frac{1}{m} \sum_{i=1}^m Z_i$  be the mean of these random variables, and let any  $\gamma > 0$  be fixed. Then

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

- This lemma ( which in learning theory is also called the Chernoff bound ) says that if we take  $\hat{\phi}$  — the average of  $m$  Bernoulli( $\phi$ ) random variables — to be our estimate of  $\phi$ , then the probability of our being far from the true value is small, so long as  $m$  is large.

# Version Space



- ❖ A hypothesis  $h$  is consistent with a set of training examples  $S$  of target concept  $c$  if and only if  $h(x) = c(x)$  for each training example  $\langle x^{(i)}, c(x^{(i)}) \rangle$  in  $S$

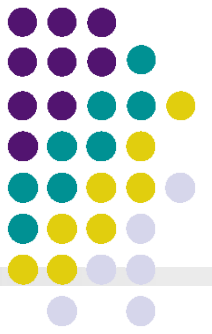
$$Consistent(h, S) \models h(x) = c(x), \forall \langle x, c(x) \rangle \in S$$

- ❖ The version space,  $VS_{H,S}$ , with respect to hypothesis space  $H$  and training examples  $S$  is the subset of hypotheses from  $H$  consistent with all training examples in  $S$ .

$$VS_{H,S} \equiv \{h \in H \mid Consistent(h, S)\}$$

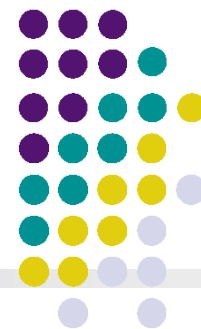
# Consistent Learner

---



- ❖ A learner is **consistent** if it outputs hypothesis that perfectly fits the training data
  - This is a quite reasonable learning strategy
- ❖ Every consistent learning outputs a hypothesis belonging to the version space
- ❖ We want to know how such hypothesis generalizes

# Probably Approximately Correct



## ❖ Goal:

- PAC-Learner produces hypothesis  $\hat{h}$

that is approximately correct

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

with high probability

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

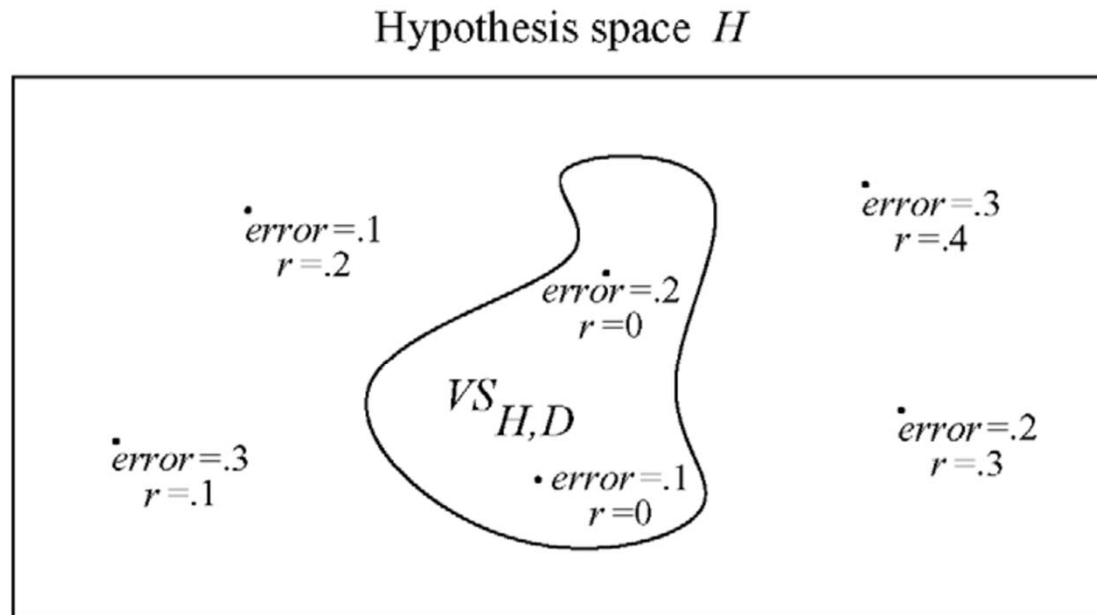
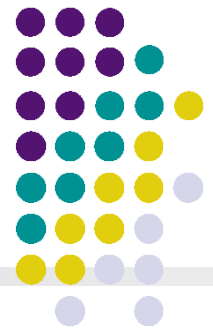
## ❖ Double

“hedging”

- approximately
- probably

Need both !

# Exhausting the Version Space

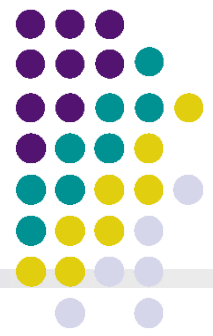


( $r$  = training error,  $\text{error}$  = true error)

- ❖ Definition: The version space  $VS_{H,S}$  is said to be  $\varepsilon$ -exhausted with respect to  $c$  and  $S$ , if every hypothesis  $h$  in  $VS_{H,S}$  has true error less than  $\varepsilon$  with respect to  $c$  and  $D$ .

$$\forall h \in VS_{H,S}, \quad \hat{\varepsilon}_D(h) < \varepsilon$$

# How Many Examples Will $\epsilon$ -exhaust the VS



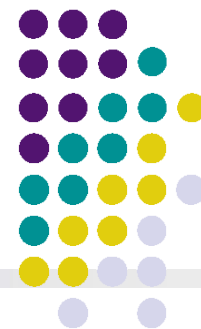
❖ Theorem: [Haussler, 1988].

- If the hypothesis space  $H$  is finite, and  $S$  is a sequence of  $m \geq 1$  independent random examples of some target concept  $c$ , then for **ANY**  $0 \leq \epsilon \leq 1/2$ , the probability that **the version space with respect to  $H$  and  $S$  is not  $\epsilon$ -exhausted** ( **with respect to  $c$**  ) is less than

$$|H|e^{-\epsilon m}$$

- This bounds the probability that any consistent learner will output a hypothesis  $h$  with  $\epsilon(h) \geq \epsilon$

# What It Means



- ❖ [Haussler, 1988]: probability that the version space is not  $\varepsilon$ -exhausted after  $m$  training examples is at most  $|H|e^{-\varepsilon m}$

$$Pr(\exists h \in H, \text{ s.t. } (error_{train}(h) = 0) \cap (error_{true}(h) > \varepsilon)) \leq |H|e^{-\varepsilon m}$$

Suppose we want this probability to be at most  $\delta$

$$|H|e^{-\varepsilon m} \leq \delta$$

1. How many training examples suffice ?  $m \geq \frac{1}{\varepsilon} \left( \ln|H| + \ln\left(\frac{1}{\delta}\right) \right)$

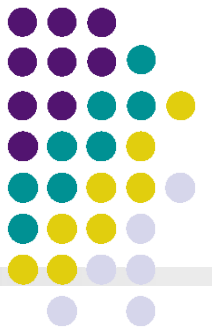
2. If  $error_{train}(h) = 0$  then with probability at least  $1 - \delta$

$$error_{true} \leq \frac{1}{m} \left( \ln|H| + \ln\left(\frac{1}{\delta}\right) \right)$$



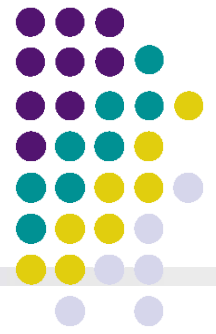
# PAC Learnability

---



- ❖ A learning algorithm is PAC learnable if it
  - Requires no more than polynomial computation per training example, and
  - no more than polynomial number of samples
- ❖ Theorem: conjunctions of Boolean literals is PAC learnable

# PAC-Learning



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

**Def' n:** Learner  $L$  PAC-learns class  $C$  using hypothesis space  $H$  if

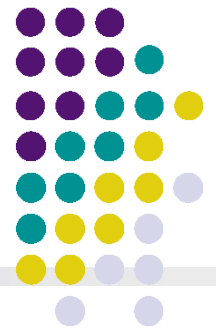
1. for any target concept  $c \in C$ , any distribution  $D$ , any  $\epsilon$  such that  $0 < \epsilon < 1/2$ ,  $\delta$  such that  $0 < \delta < 1/2$ ,  $L$  returns  $h \in H$  s.t. w/ prob.  $\geq 1 - \delta$   $\text{err}_D(h) < \epsilon$
2.  $L$ 's run-time (and hence, sample complexity) is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

- ❖ Sufficient:

1. Only  $\text{poly}(\dots)$  training instances —  $|H| = 2^{\text{poly}()}$   $m \geq \frac{1}{\epsilon} \left( \ln |H| + \ln\left(\frac{1}{\delta}\right) \right)$
2. Only  $\text{poly}(\dots)$  time / instance ...

Often  $C = H$

# Agnostic Learning



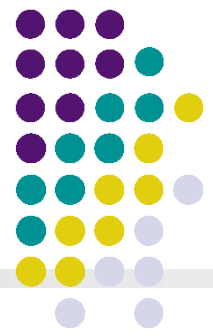
- ❖ So far, assumed  $c \in H$
- ❖ Agnostic learning setting: don't assume  $c \in H$
- ❖ What do we want then ?
  - The hypothesis  $h$  that makes fewest errors on training data
- ❖ What is sample complexity in this case ?

$$m \geq \frac{1}{2\epsilon^2} \left( \ln |H| + \ln\left(\frac{1}{\delta}\right) \right)$$

derived from Hoeffding bounds:

$$\Pr[\text{error}_D(h) > \text{error}_S(h) + \epsilon] \leq e^{-2m\epsilon^2}$$

# Empirical Risk Minimization Paradigm



- ❖ Choose a **Hypothesis Class**  $H$  of subsets of  $X$
- ❖ For an input sample  $S$ , find some  $h$  in  $H$  that fits  $S$  “well”
- ❖ For a new point  $x$ , predict a label according to its membership in  $h$

$$\hat{h} = \operatorname{argmin}_{h \in H} \hat{\epsilon}_S(h)$$

## ❖ Example:

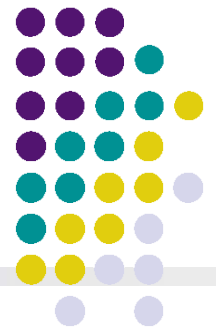
- Consider linear classification, and let  $h_\theta(x) = 1\{\theta^T x \geq 0\}$   
Then  $H = \{h_\theta : h_\theta(x) = 1\{\theta^T x \geq 0\}, \theta \in R^{n+1}\}$

$$\hat{\theta} = \operatorname{argmin}_{\theta} \hat{\epsilon}_S(h_\theta)$$

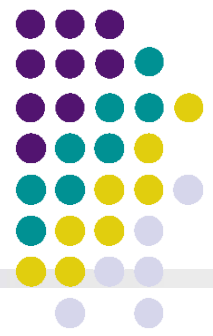
- We think of ERM as the most “basic” learning algorithm, and it will be this algorithm that we focus on in the remaining.
- In our study of learning theory, it will be useful to abstract away from the specific parameterization of hypotheses and from issues such as whether we’re using a linear classifier or an ANN

# The Case of Finite H

---



- ❖  $H = \{h_1, \dots, h_k\}$  consisting of  $k$  hypotheses.
- ❖ We would like to give guarantees on the generalization error of  $\hat{h}$ .
- ❖ First, we will show that  $\hat{\epsilon}(h)$  is a reliable estimate of  $\epsilon(h)$  for all  $h$ .
- ❖ Second, we will show that this implies an upper-bound on the generalization error of  $\hat{h}$ .



# Misclassification Probability

❖ The outcome of a binary classifier can be viewed as a Bernoulli random variable  $Z$ :  $Z = 1\{h_i(x) \neq c(x)\}$

❖ For each sample:  $Z_j = 1\{h_i(x^{(j)}) \neq c(x^{(j)})\}$

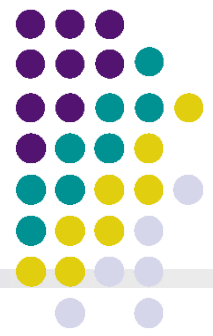
$$\hat{\varepsilon}(h_i) = \frac{1}{m} \sum_{j=1}^m Z_j$$

❖ Hoeffding inequality

$$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2e^{-2\gamma^2 m}$$

- This shows that, for our particular  $h_i$ , training error will be close to generalization error with high probability, assuming  $m$  is large.

# Uniform Convergence



- ❖ But we don't just want to guarantee that  $\hat{\varepsilon}(h_i)$  will be close  $\varepsilon(h_i)$  (with high probability) for just only one particular  $h_i$ . We want to prove that this will be true for simultaneously for all  $h_i \in H$

- ❖ For  $k$  hypothesis:

$$\begin{aligned} P(\exists h \in H, |\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) &= P(A_1 \cup \dots \cup A_k) \\ &< \sum_{i=1}^k P(A_i) \\ &= \sum_{i=1}^k 2e^{-2\gamma^2 m} \\ &= 2ke^{-2\gamma^2 m} \end{aligned}$$

- ❖ This means:

$$\begin{aligned} P(\neg \exists h \in H, |\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) &= P(\forall h \in H, |\varepsilon(h_i) - \hat{\varepsilon}(h_i)| \leq \gamma) \\ &= 1 - 2ke^{-2\gamma^2 m} \end{aligned}$$

# Uniform Convergence, cont.

---



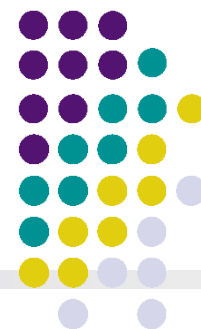
- ❖ In the discussion above, what we did was, for particular values of  $m$  and  $\gamma$ , given a bound on the probability that for some  $h_i \in H$

$$|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma$$

- ❖ There are three quantities of interest here:  $m$  and  $\gamma$ , and probability of error; we can bound either one in terms of the other two.



# Sample Complexity



- ❖ How many training examples we need in order to make a guarantee ?

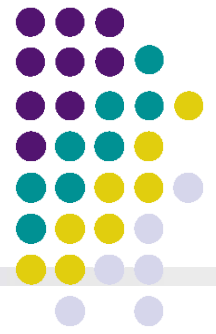
$$P(\exists h \in H, |\varepsilon(h) - \hat{\varepsilon}(h)| > \gamma) = 2ke^{-2\gamma^2 m}$$

- ❖ We find that if  $m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$

then with probability at least  $1 - \delta$ , we have that  $|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| \leq \gamma$  for all  $h_i \in H$

- ❖ The key property of the bound above is that the number of training examples needed to make this guarantee is only **logarithmic in k**, the number of hypotheses in  $H$ . This will be important later.

# Generalization Error Bound



- ❖ Similarly, we can also hold  $m$  and  $\delta$  fixed and solve for  $\gamma$  in the previous equation, and show [again, convince yourself that this is right !] that with probability  $1 - \delta$ , we have that for all  $h_i \in H$

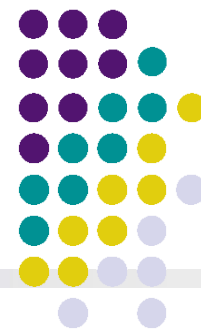
$$|\hat{\varepsilon}(h) - \varepsilon(h)| \leq \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

- ❖ Define  $h^* = \operatorname{argmin}_{h \in H} \varepsilon(h)$  to be the best possible hypothesis in  $H$ .

$$\begin{aligned} \varepsilon(\hat{h}) &\leq \hat{\varepsilon}(\hat{h}) + \gamma \\ &\leq \hat{\varepsilon}(\hat{h}^*) + \gamma \\ &\leq \varepsilon(\hat{h}^*) + 2\gamma \end{aligned}$$

- If uniform convergence occurs, then the generalization error of  $\varepsilon(\hat{h})$  is at most  $2\gamma$  worse than the best possible hypothesis in  $H$  !

# Summary



Theorem. Let  $|H| = k$ , and let any  $m, \delta$  be fixed. Then with probability at least  $1 - \delta$ , we have that

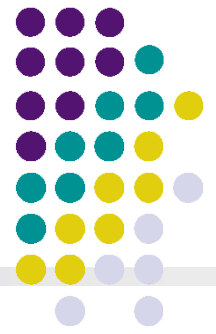
$$\varepsilon(\hat{h}) \leq \left( \min_{h \in H} \varepsilon(h) \right) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

Corollary. Let  $|H| = k$ , and let any  $\delta, \gamma$  be fixed. Then for  $\varepsilon(\hat{h}) \leq \min_{h \in H} \varepsilon(h) + 2\gamma$  to hold with probability at least  $1 - \delta$ , it suffices that

$$\begin{aligned} m &\geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta} \\ &= O\left(\frac{1}{2\gamma^2} \log \frac{k}{\delta}\right) \end{aligned}$$

# What If $H$ is Not Finite ?

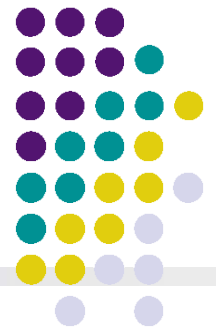
---



- ❖ Can't use our result for infinite  $H$
- ❖ Need some other measure of complexity for  $H$  — Vapnik-Chervonenkis (VC) dimension !
- ❖ Some Informal Derivation
  - Suppose we have an  $H$  that is parameterized by  $d$  real numbers. Since we are using a computer to represent real numbers, and IEEE double-precision floating point (double's in C) uses 64 bits to represent a floating point number, this means that our learning algorithm, assuming we're using double-precision floating point, is parameterized by  $64d$  bits
- ❖ Parameterization

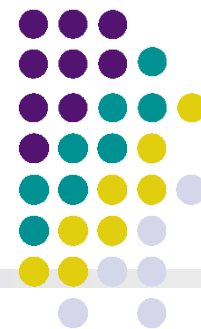
# How do we characterize “power” ?

---



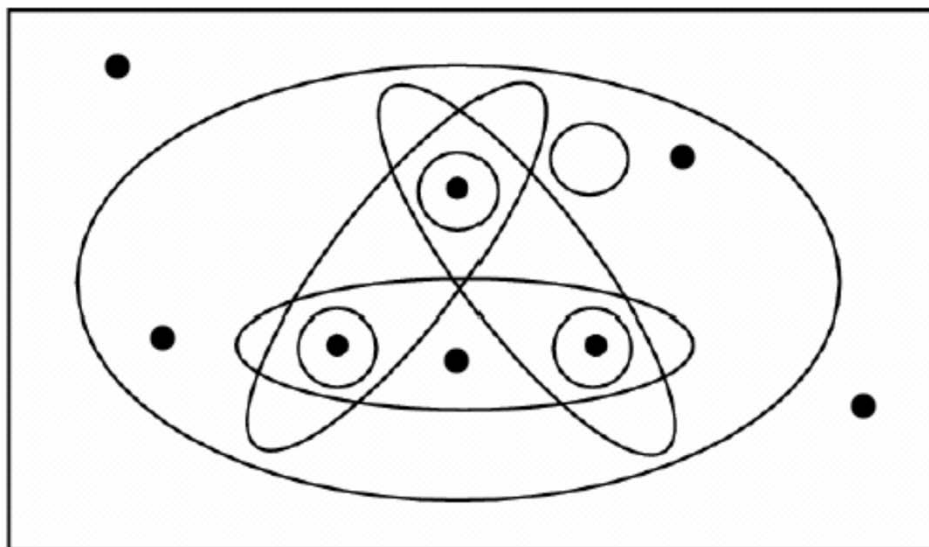
- ❖ Different machines have different amounts of “power”
- ❖ Tradeoff between:
  - More power: Can model more complex classifiers but might overfit
  - Less power: Not going to overfit, but restricted in what it can model
- ❖ How do we characterize the amount of power ?

# The Vapnik-Chervonenkis Dimension



- ❖ Definition: The **Vapnik-Chervonenkis dimension**,  $VC(H)$ , of hypothesis space  $H$  defined over instance space  $X$  is the size of the **largest finite subset** of  $X$  shattered by  $H$ . If arbitrarily large finite sets of  $X$  can be shattered by  $H$ , then  $VC(H) \equiv \infty$ .

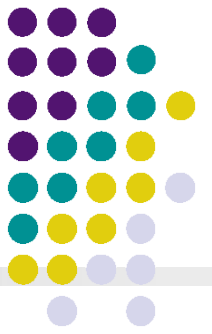
Instance space  $X$



*Definition:*

Given a set  $S = \{x(1), \dots, x(d)\}$  of points  $x(i) \in X$ , we say that  $H$  **shatters**  $S$  if  $H$  can realize any labeling on  $S$ .

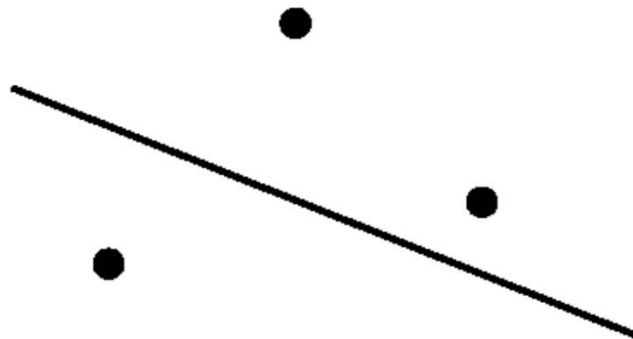
# VC Dimension: Examples



❖ Consider  $X = \mathbb{R}^2$ , want to learn  $c: X \rightarrow \{0, 1\}$

• What is VC dimension of lines in a plane ?

$$H = \{ ( (wx+b) > 0 \rightarrow y=1 ) \}$$

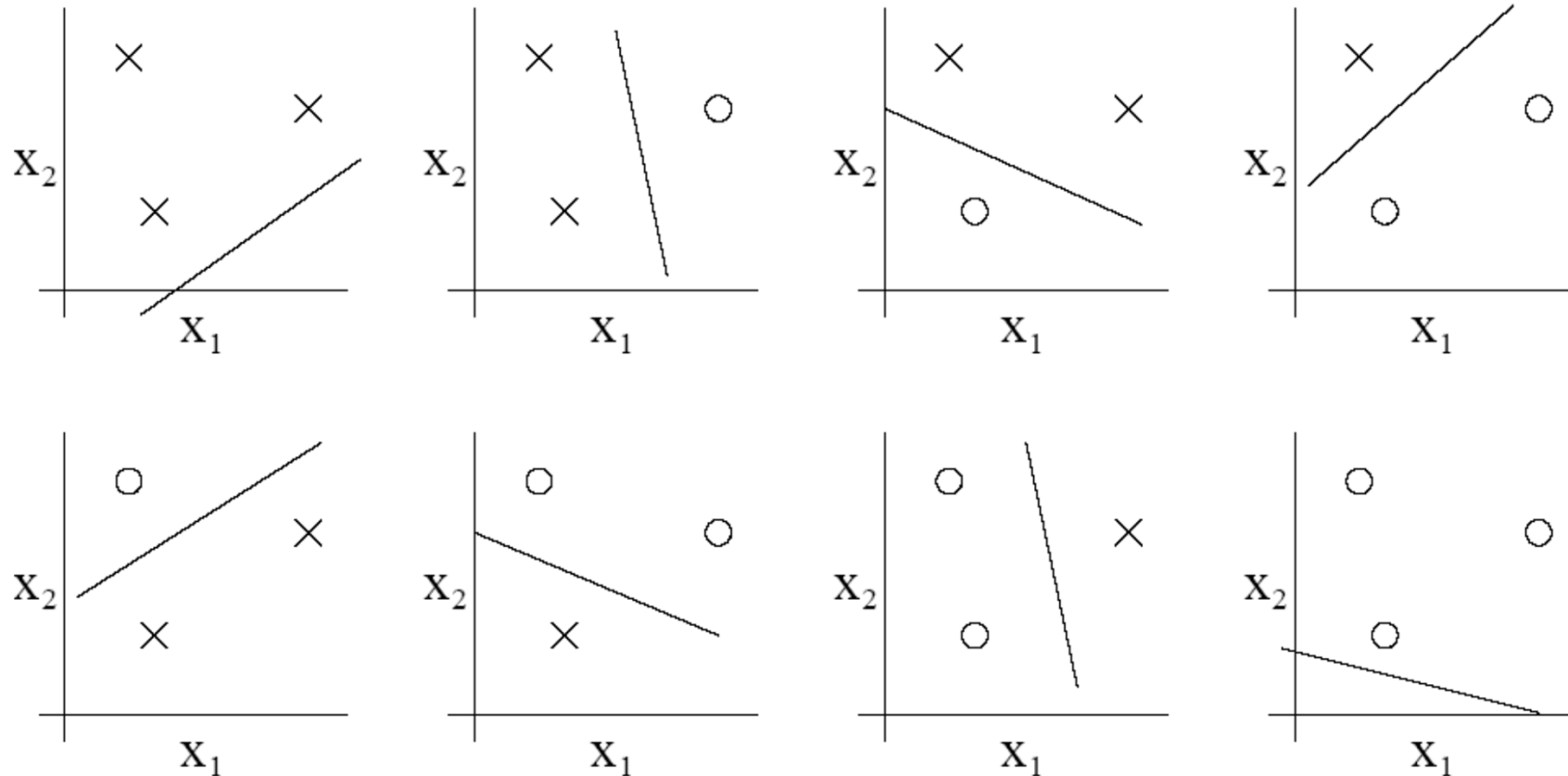
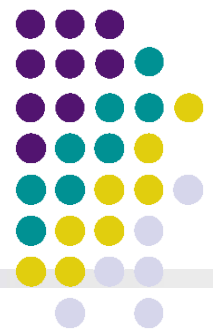


(a)



(b)

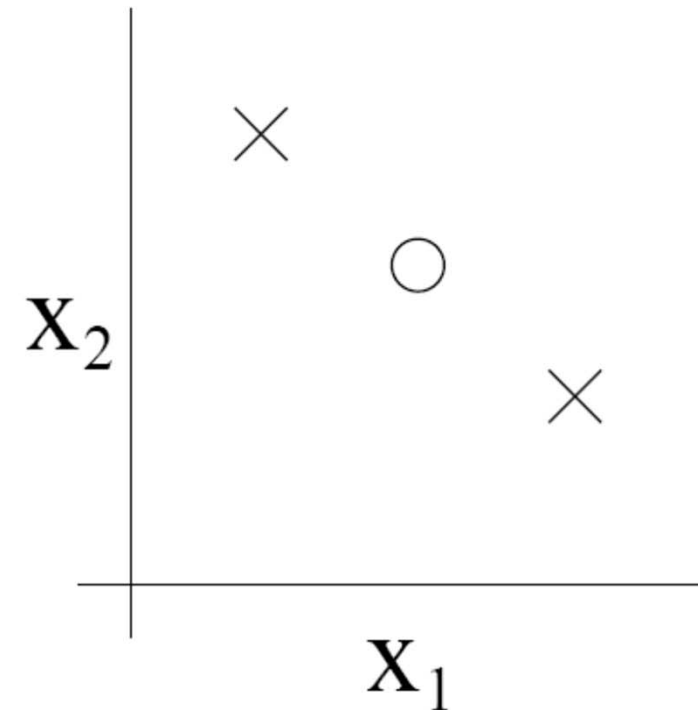
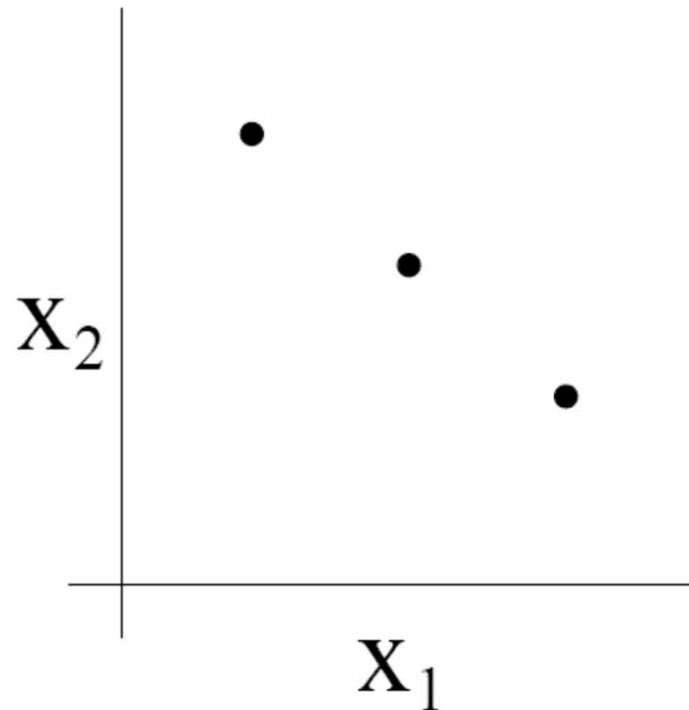
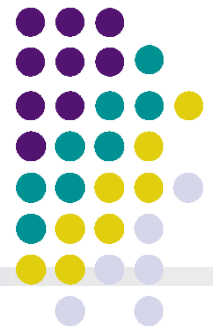
# VC Dimension: Examples



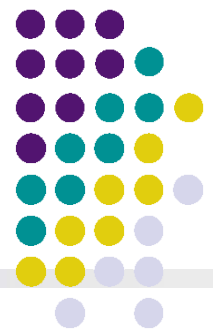
- For any of the eight possible labeling of these points, we can find a linear classifier that obtains “zero training error” on them.
- Moreover, it is possible to show that there is no set of 4 points that this hypothesis class can shatter.



# VC Dimension: Examples



- The VC dimension of  $H$  here is 3 even though there may be sets of size 3 that it cannot shatter.
- Under the definition of the VC dimension, in order to prove that  $VC(H)$  is at least  $d$ , we need to show only that there's at least one set of size  $d$  that  $H$  can shatter.

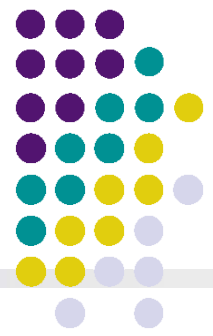


❖ **Theorem:** Consider some set of  $m$  points in  $\mathbb{R}^n$ . Choose any one of the points as origin. Then the  $m$  points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent.

❖ **Corollary:** The VC dimension of the set of oriented hyperplanes in  $\mathbb{R}^n$  is  $n+1$ .

Proof: we can always choose  $n + 1$  points, and then choose one of the points as origin, such that the position vectors of the remaining  $n$  points are linearly independent, but can never choose  $n + 2$  such points ( since no  $n + 1$  vectors in  $\mathbb{R}^n$  can be linearly independent ).

# The VC Dimension and the Number of Parameters

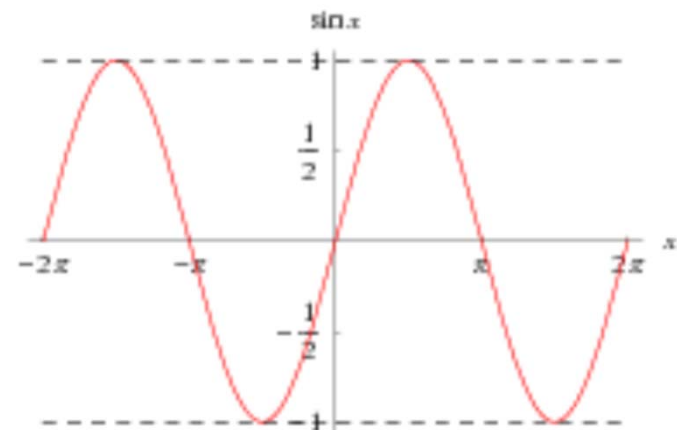


- ❖ The VC dimension thus gives concreteness to the notion of the capacity of a given set of  $h$ .
- ❖ Is it true that learning machines with many parameters would have high VC dimension, while learning machines with few parameters would have low VC dimension ?

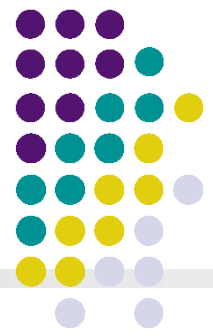
An infinite-VC function with just one parameter !

$$f(x, \alpha) \equiv \theta(\sin(\alpha x)), \quad x, \alpha \in \mathbb{R}$$

where  $\theta$  is an indicator function



# An Infinite-VC Function with Just One Parameter

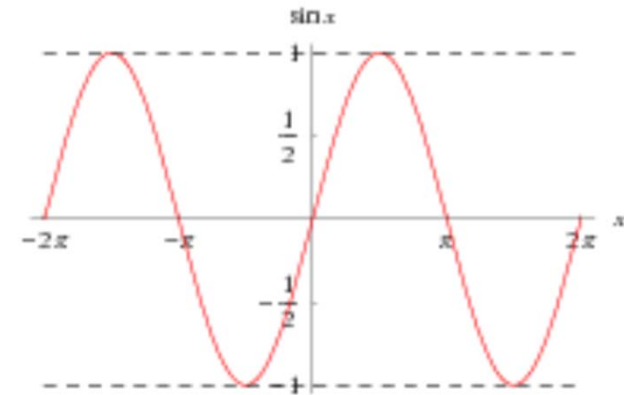


- ❖ You choose some number  $l$ , and present me with the task of finding  $l$  points that can be shattered. I choose them to be

$$x_i = 10^{-i} \quad i = 1, \dots, l,$$

- ❖ You specify any labels you like:

$$y_1, y_2, \dots, y_l, \quad y_i \in \{-1, 1\}$$

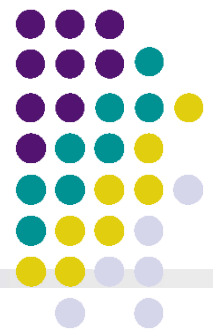


- ❖ Then  $f(\alpha)$  gives this labeling if I choose  $\alpha$  to be

$$\alpha = \pi \left( 1 + \sum_{i=1}^l \frac{(1 - y_i)10^i}{2} \right)$$

- ❖ Thus the VC dimension of this machine is infinite.

# Sample Complexity from VC Dimension



- ❖ How many randomly drawn examples suffice to  $\varepsilon$ -exhaust  $VS_{H,S}$  with probability at least  $1 - \delta$  ?

i.e., to guarantee that any hypothesis that perfectly fits the training data is probably  $1 - \delta$  approximately ( $\varepsilon$ ) correct on testing data from the same distribution

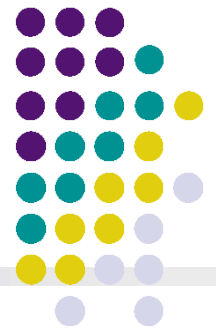
$$m \geq \frac{1}{\varepsilon} \left( 4 \log_2 \frac{2}{\delta} + 8VC(H) \log_2 \frac{13}{\varepsilon} \right)$$

Compare to our earlier results based on  $|H|$ :

$$m \geq \frac{1}{2\varepsilon^2} \left( \ln |H| + \ln \frac{1}{\delta} \right)$$

# What You Should Know

---



- ❖ Sample complexity varies with the learning setting
  - Learner actively queries trainer
  - Examples provided at random
- ❖ Within the PAC learning setting, we can bound the probability that learner will output hypothesis with given error
  - For ANY consistent learner (case where  $c \in H$ )
  - For ANY “best fit” hypothesis (agnostic learning, where perhaps  $c$  not in  $H$ )
- ❖ VC dimension as measure of complexity of  $H$