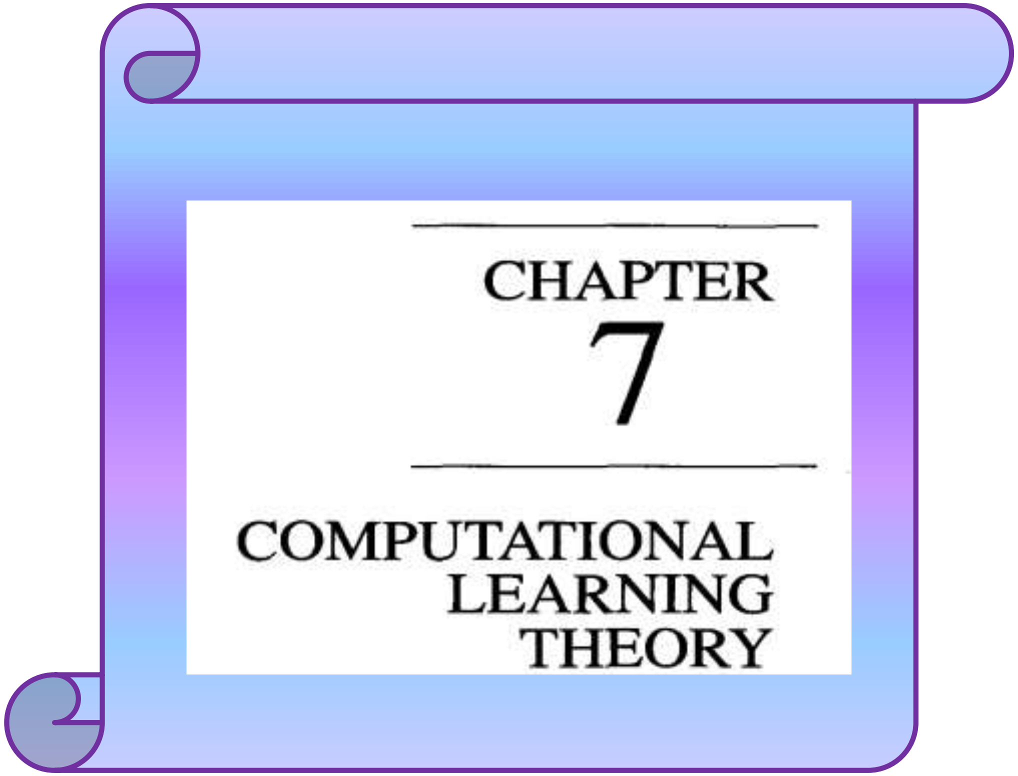
COmputational Learning Theory (COLT)



CHAPTER

7

COMPUTATIONAL
LEARNING
THEORY

B. Tech III Year
CS 701 PC

Unit III Part B Computational learning theory

7.1	**Introduction**
7.2	**PROBABLY LEARNING AN APPROXIMATELY CORRECT HYPOTHESIS** **PAC**
7.3	**SAMPLE COMPLEXITY (m) FOR FINITE HYPOTHESIS SPACES**
7.4	**SAMPLE COMPLEXITY(m) FOR INFINITE HYPOTHESIS SPACES**
7.5	**THE MISTAKE BOUND MODEL OF LEARNING**

COmputational Learning Theory (COLT)

This chapter presents a

- **theoretical characterization the difficulty** of types of ML problems
- the **capabilities** of several types of ML algorithms.

Theory seeks to answer questions such as:

1. Under what conditions is successful learning **possible** and **impossible**?

2. Under what conditions is a particular learning algorithm **assured** of learning successfully?

Two specific frameworks *for analyzing* learning algorithms are considered.

1. **Within the *Probably Approximately Correct* (PAC) framework,**

   It identify classes of hypotheses that **can** and **cannot** be learned from a polynomial number of training examples and we define a natural measure of complexity for hypothesis spaces that allows bounding the number of training examples required for inductive learning.

2. **Within the Mistake Bound framework,**

   It examine the number of training errors that will be made by a learner before it determines the correct hypothesis.

7.1 Introduction

When studying machine learning it is natural to wonder what **general laws** may govern ***machine*** (and *nonmachine*) learners

what **general laws govern by ML's**

- Is it possible to identify classes of learning problems that are inherently difficult or easy, independent of the learning algorithm?
- Can one characterize the number of training examples necessary or sufficient to assure successful learning?
- How is this number affected if the learner is allowed to pose queries to the trainer, versus observing a random sample of training examples?
- Can one characterize the number of mistakes that a learner will make before learning the target function?
- Can one characterize the inherent computational complexity of classes of learning problems?

Although **general answers** to all these questions are not yet known,

fragments of a COmputational Theory of Larning (COLT) have begun to emerge.

➢The solution for the problem of inductively learning an unknown target function, given only training examples of this target function and a space of candidate hypotheses.

➢ Within this setting, we will be chiefly concerned with questions such as

- how many training examples are sufficient to successfully learn the target function,
- how many mistakes will the learner make before succeeding.

It is possible to set quantitative bounds on these measures, depending on attributes of the learning problem such as:

- ***size or complexity of the hypothesis space considered by learner***
- ***accuracy to which the target concept must be approximated***
- ***probability that the learner will output a successful hypothesis***
- ***manner in which training examples are presented to the learner***

Our goal is to answer questions such as:

❖***Sample complexity.***

How many training examples are needed for a learner to converge (with high probability) to a successful hypothesis?

❖***Computational complexity.***

How much computational effort is needed for a **learner to converge (with high probability) to a successful hypothesis?**

❖ ***Mistake bound.***

How many training examples will the learner misclassify before converging to a successful hypothesis?

7.2 PROBABLY LEARNING AN APPROXIMATELY CORRECT

A particular setting for the learning problem, called the probably approximately correct (PAC) learning model.

Specifying the problem setting that defines the PAC learning model, consider the questions of

- how many training examples
- how much computation are required in order to learn various classes of target functions within this PAC model.

For the sake of simplicity,

- the discussion to the case of learning boolean valued concepts from noise-free training data.
- However, many of the results can be extended to learning real-valued target functions (see, for example, Natarajan 1991),
- and some can be extended to learning from certain types of noisy data (see, for example, Laird 1988; Kearns and Vazirani 1994).

(PAC) HYPOTHESIS

The Problem Setting

Let X refer to the set of all possible instances over which target functions may be defined. E.g., X might represent the set of all people, each described by the attributes age (e.g., **young** or old) and height (short or **tall**).

Let ***C*** *refer to some set of target concepts that our learner might be called* upon to learn. Each target concept ***c*** *in* ***C*** *corresponds to some subset of X, or* equivalently to some Boolean-valued function ***c : X -> {0, 1}.***

E.g., *one* target concept ***c*** *in* ***C*** *might be the concept* ***"people who are skiers."*** *If* ***x*** *is a* positive example of c, then we will write ***c(x) = 1;*** *if* ***x*** *is a negative example,* ***c(x) = 0.***

We assume instances are generated at random from **X** according to some probability distribution ***D.***

E.g. ***D*** *might be the distribution of instances* generated by observing people who walk out of the largest sports store in Switzerland!

(What about in India!)

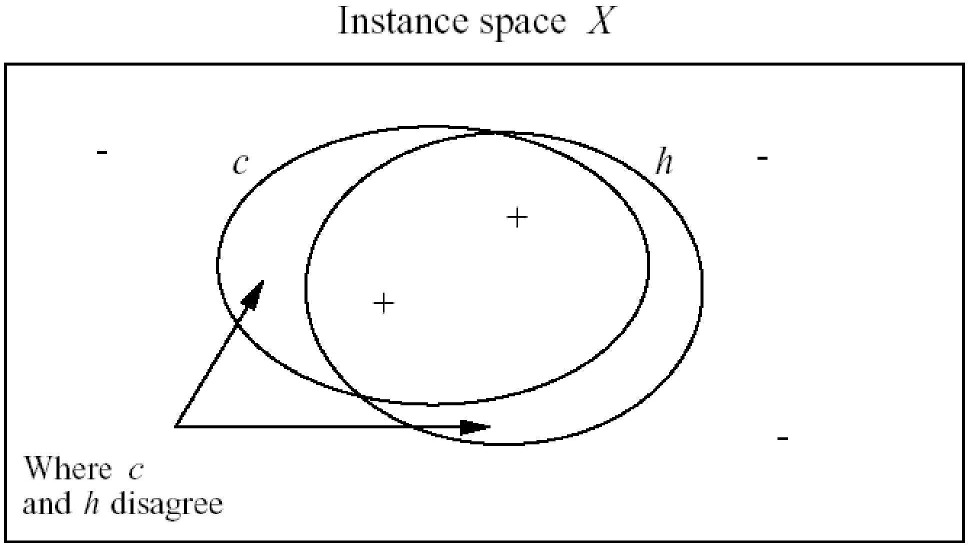
In general, ***D*** *may be any distribution, and it will not generally be known* to the learner.

All that we require of ***D*** is that it be ***stationary; i.e.,*** *the* distribution not change over time. Training examples are generated by drawing an instance ***x*** at random according to ***D,*** then presenting x along with its target value, ***c(x), to the learner*** describable by conjunctions of the attributes ***age*** *and* ***height.***

After observing a sequence of training examples of the target concept ***c, L*** *must output some* hypothesis **h** from **H**, which is its estimate of **c**. To be fair, we evaluate the success of L by the performance of ***h*** over new instances drawn randomly from X according to *D, the same probability distribution used to generate the training* data.

Within this setting, we are interested in characterizing the performance of various learners L using various hypothesis spaces H, when learning individual target concepts drawn from various classes C. Because we demand that L be general enough to learn any target concept from ***C regardless of the distribution*** of training examples, we will often be interested in worst-case analyses over all possible target concepts from ***C and all possible instance distributions D.***

a.True Error of a Hypothesis



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

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

b.PAC Learning

- Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.
  - *Definition*: C is **PAC-learnable** by L using H if for all $c \in C$, distributions D over X, ε such that $0 < \varepsilon < 1/2$, and δ such that $0 < \delta < 1/2$
  - learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_D(h) \leq \varepsilon$, in time that is polynomial in $1/\varepsilon$, $1/\delta$, n, and $size(c)$.

7.3 SAMPLE COMPLEXITY FOR FINITE HYPOTHESIS SPACES

the PAC- learnability is largely determined by the number of training examples required by the learner.

The growth in the number of required training examples with problem size, called the ***sample complexity of the learning problem,*** is the characteristic that is usually of greatest interest.

The reason is that in most practical settings the factor that most limits success of the learner is the *limited availability of training data.*

Consistent Learners

A general bound on the sample complexity for a very broad class of learners, called ***consistent learners.***

A learner is **consistent** if it outputs hypotheses that perfectly fit the training data, whenever possible.

It is quite reasonable to ask that a learning algorithm be consistent, given that we typically prefer a hypothesis that fits the training data over one that does not.

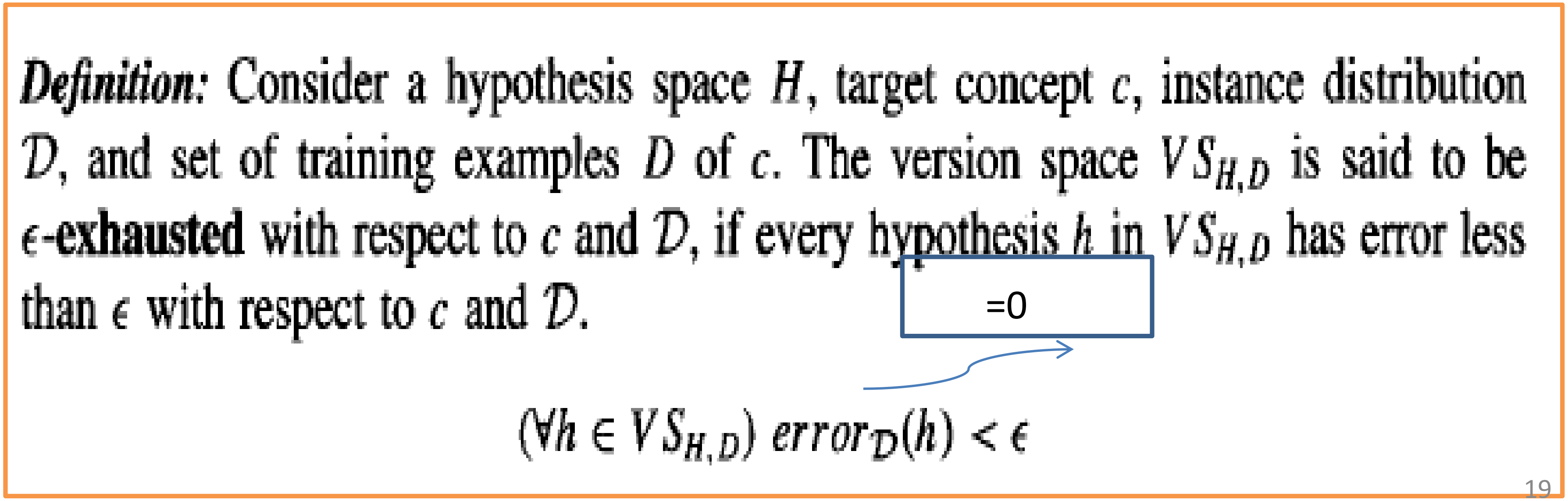
Can we derive a bound on the number of training examples required by **any consistent learner, independent of the specific algorithm it uses to derive a *consistent hypothesis?***

The answer is yes. To accomplish this, it is useful to recall the definition of version space from Chapter 2. There we defined the version space, ***$VS_{H,D}$*** *to be the set of all hypotheses* ***$h \in H$*** *that correctly classify the*

$$VS_{H,D} = \{h \in H | (\forall \langle x, c(x) \rangle \in D)\ (h(x) = c(x))\}$$

The significance of the version space here is that every consistent learner outputs a hypothesis belonging to the version space, regardless of the instance space **X**, hypothesis space H, or training data D. The reason is simply that by definition the version space ***V S*** $_{H,D}$ contains every consistent hypothesis in H.

Therefore, to bound the number of examples needed by any consistent learner, we need only bound the number of examples needed to assure that the version space contains no unacceptable hypotheses.

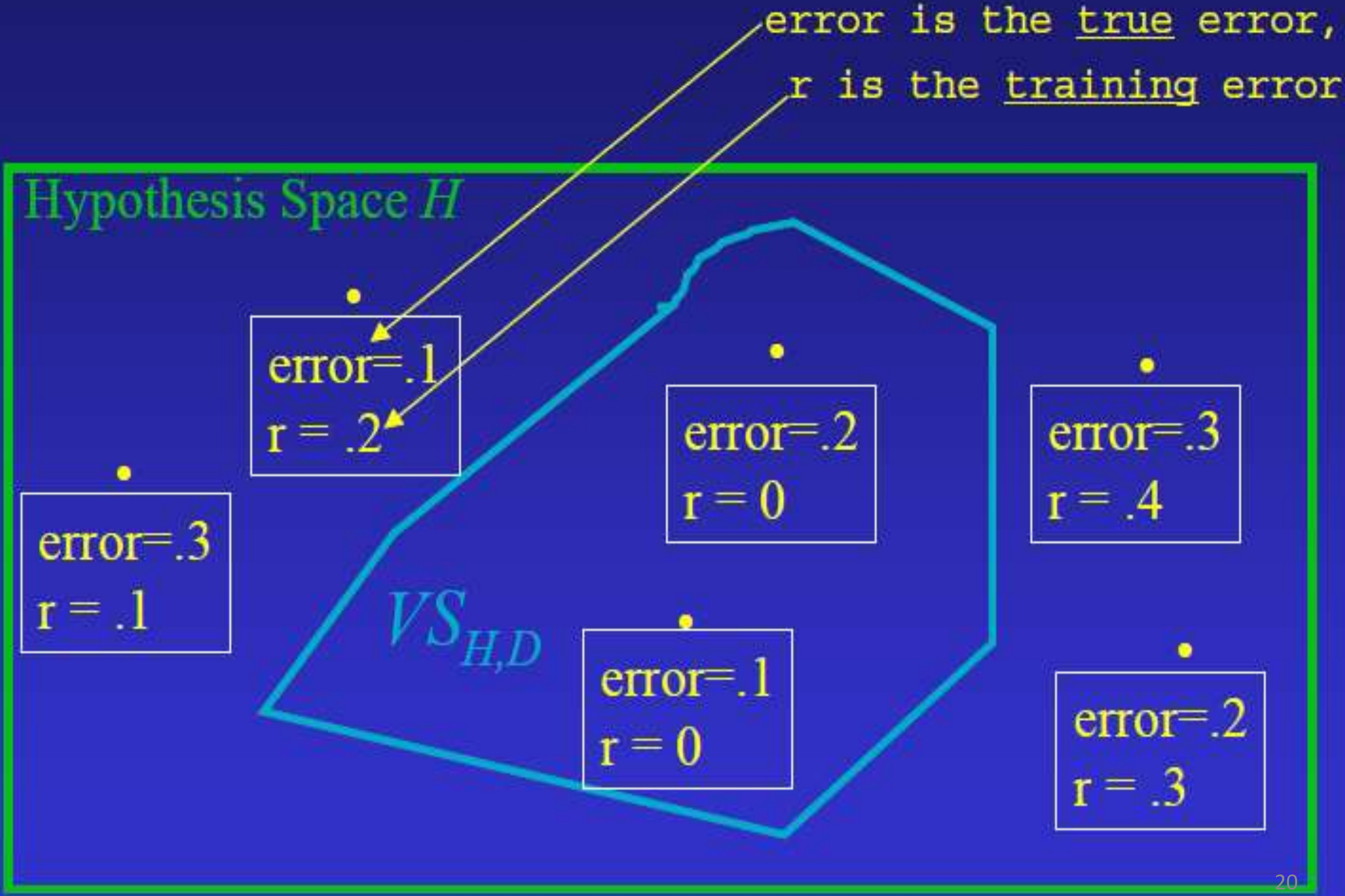


Definition: Consider a hypothesis space H, target concept c, instance distribution $\mathcal{D}$, and set of training examples D of c. The version space $VS_{H,D}$ is said to be ϵ-**exhausted** with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ϵ with respect to c and $\mathcal{D}$.

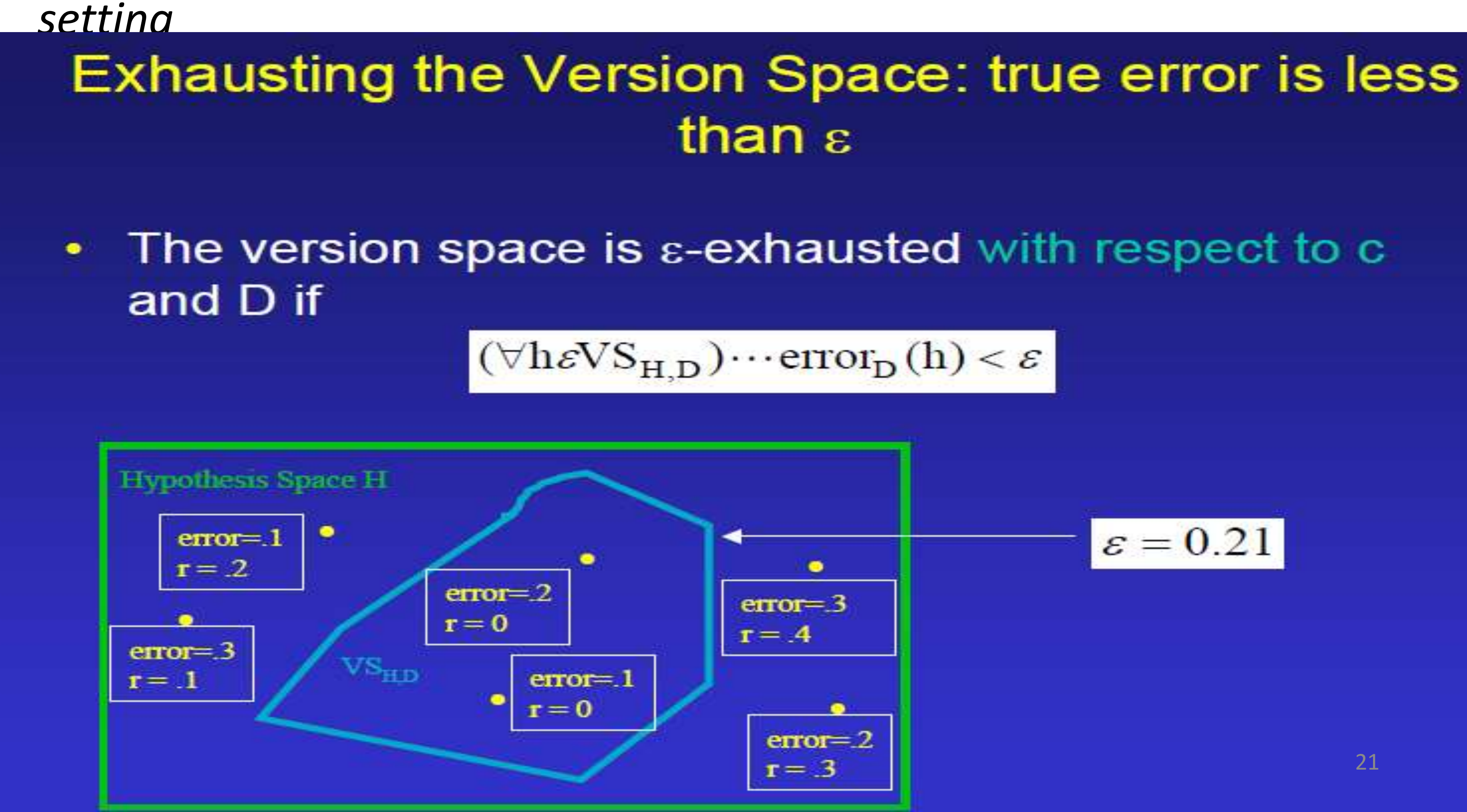
=0

$$(\forall h \in VS_{H,D})\ error_{\mathcal{D}}(h) < \epsilon$$

Version Space with associated errors



A **version space is epsilon-*exhausted*** iff all remaining hypothesis have low error. Then we can choose any of the hypothesis and be acceptably fine according to your chosen error rate. In other words, it is Probably Approximately Correct-------------→ *making problem setting*



Exhausting the Version Space: true error is less than ε

- The version space is ε-exhausted with respect to c and D if

$$(\forall h \varepsilon VS_{H,D}) \cdots \mathrm{error}_D(h) < \varepsilon$$

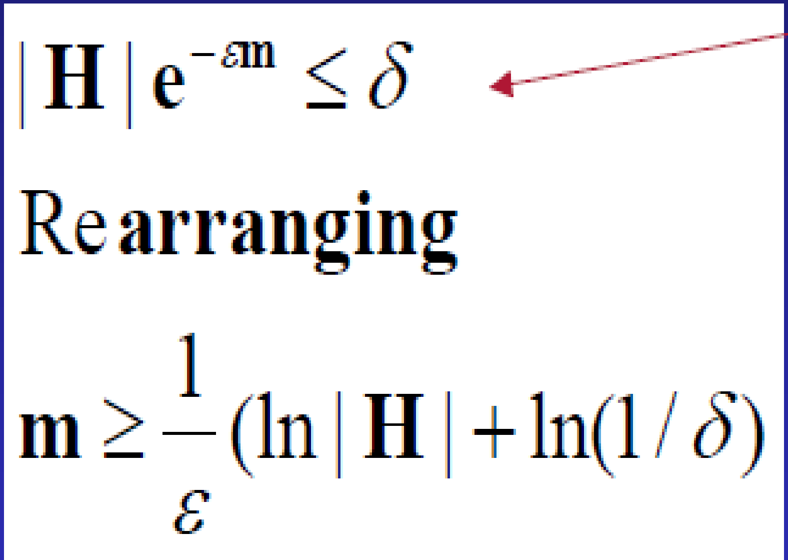
Upper bound on probability of not ε-exhausted

- **Theorem:**
    - If the hypothesis space H is finite
    - D is a sequence of $m \geq 1$ independent random samples of concept c
    - Then for any $0 < \varepsilon < 1$
    - Probability of not ε-exhausted (with respect to c) is less than or equal to

$$|\mathbf{H}|\,\mathbf{e}^{-\varepsilon \mathbf{m}}$$

- Bounds the probability that m training samples will fail to eliminate all “bad” hypotheses

Number of training samples required



$$|\mathbf{H}|\,e^{-\varepsilon m} \le \delta$$

Rearranging

$$\mathbf{m} \ge \frac{1}{\varepsilon}(\ln|\mathbf{H}| + \ln(1/\delta))$$

Probability of failure is below some desired level

(7.2)

- Provides general bound on the no. of training samples
  - sufficient for any consistent learner to learn any target concept in H for
  - any desired values of δ and ε

For concept Smiling Face

Concept features:

- Eyes {round,square} → RndEyes, ¬RndEyes
- **Nose {triangle,square} → TriNose, ¬TriNose**
- **Head {round,square} → RndHead, ¬RndHead**
- **FaceColor {yellow,green,purple} → YelFace, ¬YelFace, GrnFace, ¬GrnFace, PurFace, ¬PurFace**
- **Hair {yes,no} → Hair, ¬Hair**

Size of $|H|$ = 3^7 = 2187

If we want to assure that with probability 95%, *VS* contains only hypotheses $error_D(h) \leq .1$, then sufficient to have *m* examples, where

***m* ≥ *(1/ .1)*(ln*(2187)*+ ln*(1/ .05)*)**

***m* ≥ *10*(ln*(2187)*+ ln*(20)*)**

7.3.1 Agnostic Learning and Inconsistent Hypotheses

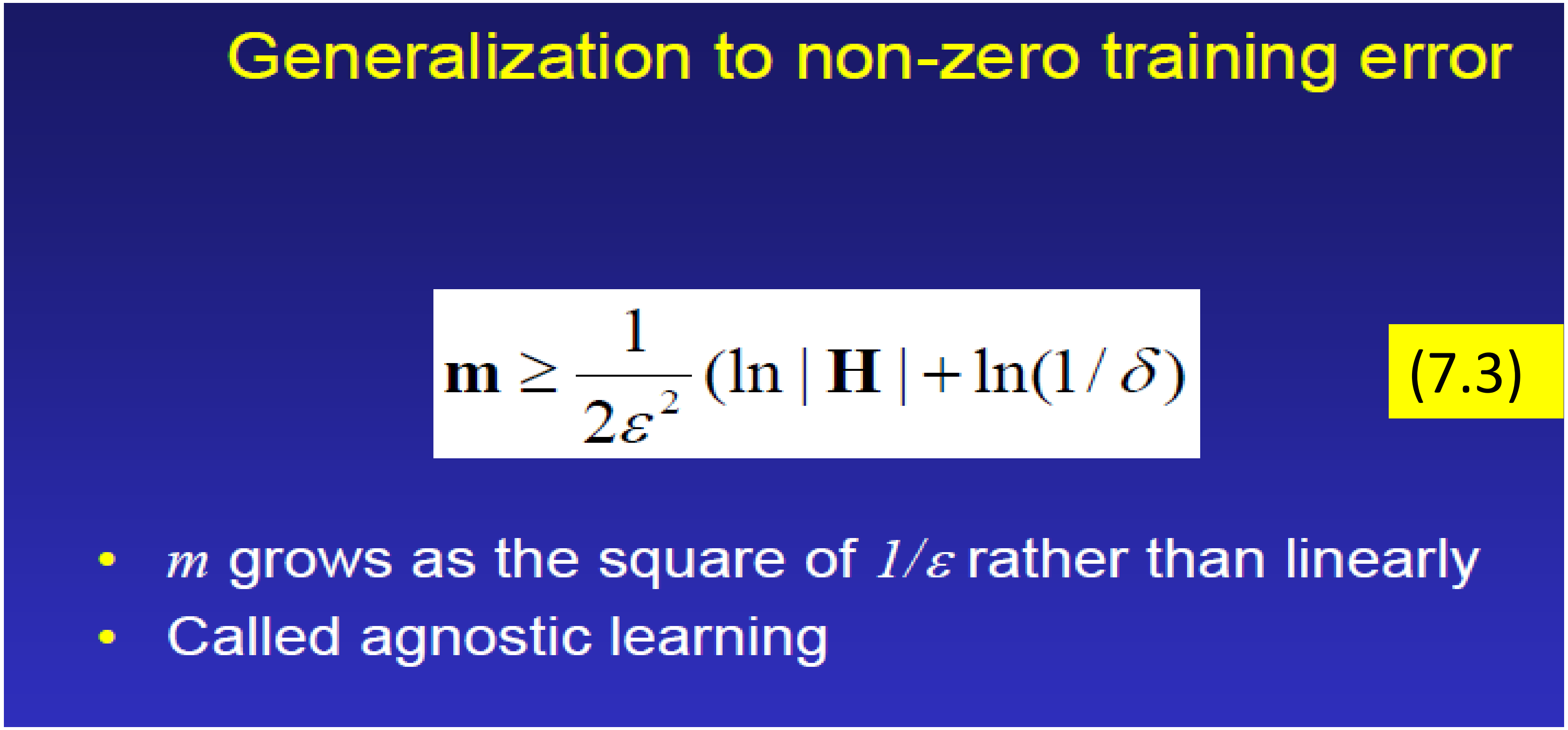
Equation (7.2) is important because it tells us how many training examples suffice to ensure (with probability (**1 - δ**)) that every hypothesis in **H** having zero training error will have a true error of at most ***ϵ.***

$$\mathbf{m} \geq \frac{1}{\varepsilon}(\ln|\mathbf{H}| + \ln(1/\delta)) \qquad (7.2)$$

Unfortunately, if H does not contain the target concept *c, then a zero-error hypothesis cannot always be found*.

In this case, the most we might ask of our learner is to output the hypothesis from ***H***that has the minimum error over the training examples. A learner that makes no assumption that the target concept is representable by ***H*** and that simply finds the hypothesis with minimum training error, is often called an ***agnostic*** learner, because it makes no prior commitment about whether or not $C \subseteq H$.

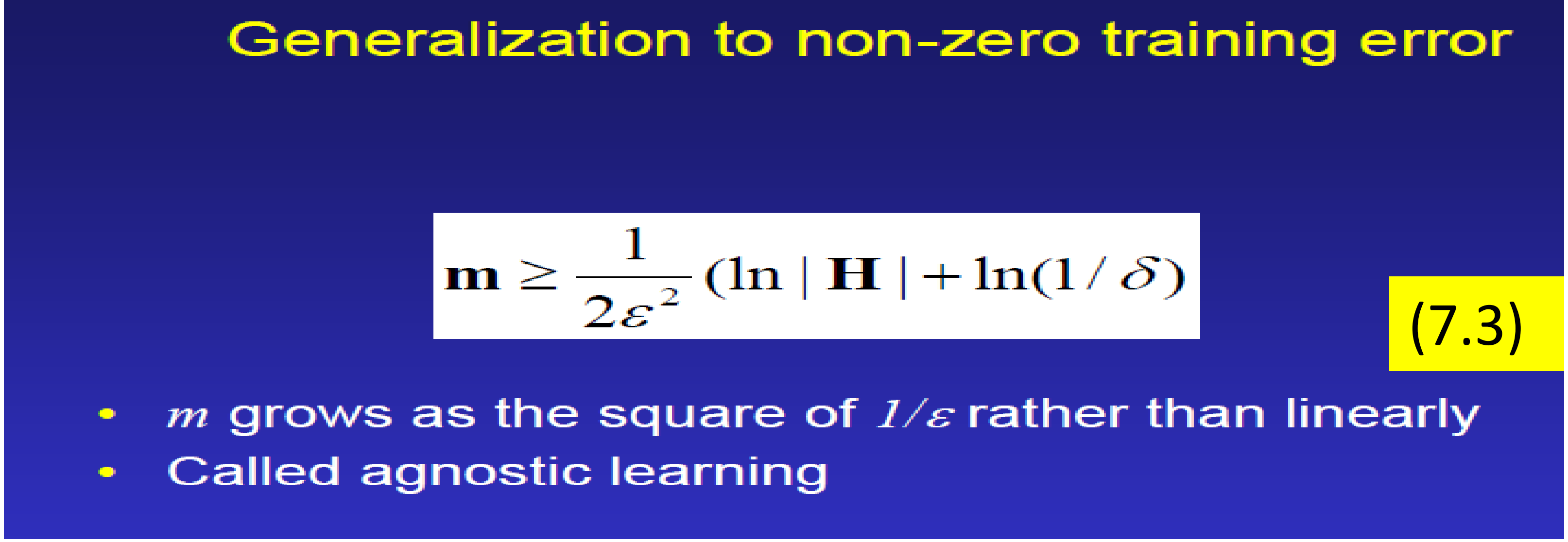
In probability theory, **Hoeffding's inequality** provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount. Hoeffding's inequality was proven by Wassily Hoeffding in 1963.



Generalization to non-zero training error

$$\mathbf{m} \geq \frac{1}{2\varepsilon^2}(\ln|\mathbf{H}| + \ln(1/\delta)) \qquad (7.3)$$

- m grows as the square of $1/\varepsilon$ rather than linearly
- Called agnostic learning



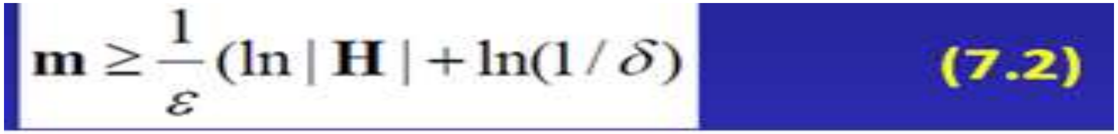
Generalization to non-zero training error

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

(7.3)

- m grows as the square of $1/\varepsilon$ rather than linearly
- Called agnostic learning

This is the generalization of Equation (7.2) to the case in which the learner still picks the best hypothesis **h ϵ H,** but where the best hypothesis may have nonzero
training error.



$$\mathbf{m} \geq \frac{1}{\varepsilon}(\ln|\mathbf{H}| + \ln(1/\delta)) \qquad \textbf{(7.2)}$$

Notice that m depends logarithmically on H and on 1/δ, as it did in the more restrictive case of Eqn (7.2). However, in this less restrictive situation m now grows as the square of 1/δ, rather than linearly with 1/δ.

7.4 SAMPLE COMPLEXITY FOR INFINITE HYPOTHESIS SPACES

In previous section we have seen that sample complexity for PAC learning grows as the logarithm of the size of the hypothesis space.

While Equation (7.2) is quite useful, there are two drawbacks to characterizing sample complexity in terms of IHI.

$$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta)) \qquad (7.2)$$

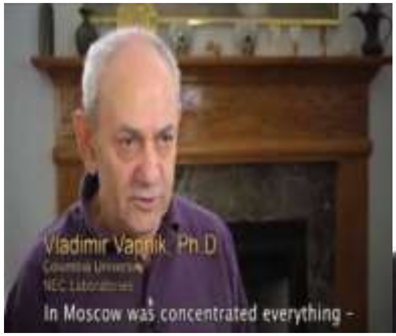
(i) it can lead to quite weak bounds (recall that the bound on *δ can be* significantly greater than 1 for large I H I).

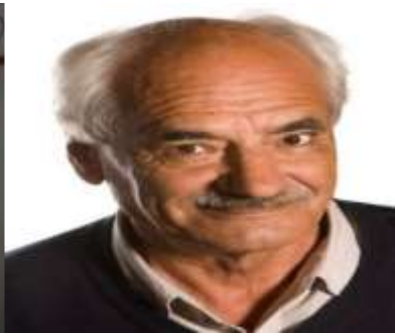
(ii) in the case of infinite hypothesis spaces we cannot apply Equation (7.2) at all!

Finite ----→ |H|

Infinite ------→ |X|

Now let us look at a second measure of the complexity of H, called the **Vapnik-Chervonenkis** dimension of H (VC dimension, or VC(H).





Intuition : We can state bounds on sample complexity that use VC(H) rather than IHI. In many cases, the sample complexity bounds based on VC(H) will be tighter than those from Eqn. (7.2).

$$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta)) \tag{7.2}$$

In addition, these bounds allow us to characterize the sample complexity of many infinite hypothesis spaces, and can be shown to be fairly tight.

7.4.1 Shattering a Set of Instances

The VC dimension measures the complexity of the hypothesis space H, not by the number of distinct hypotheses |H|, but instead by the number of distinct instances from X that can be completely discriminated using H.

To make this notion more precise, we first define the notion of **shattering** a set of instances. Consider some subset of instances S ϵ X.

Instance space X

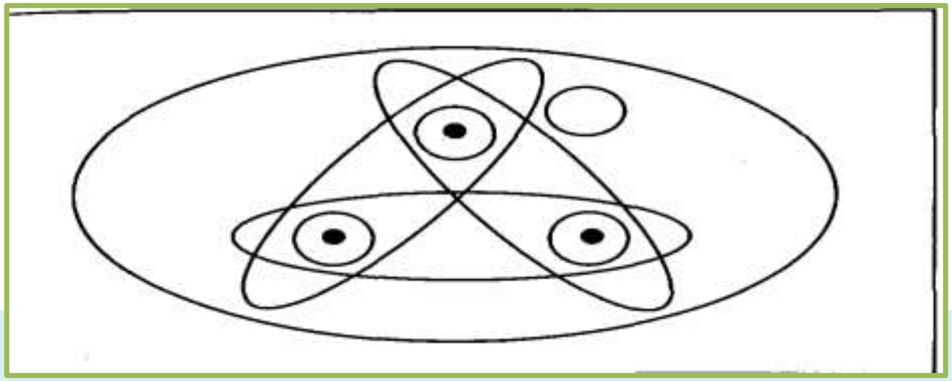


FIGURE 7.3

A set of three instances shattered by eight hypotheses. For every possible dichotomy of the instances, there exists a corresponding hypothesis

Each hypothesis h from H imposes some

dichotomy on S; that is, h partitions S into the two subsets

{ x ϵ S | h(x) = 1) and

{x ϵ S |h(x) = 0).

Given some instance set S, there are 2ISI possible **dichotomies**,

though H may be unable to represent some of these.

We say that H shatters S if every possible dichotomy of S can be

represented by some hypothesis from H.

Definition: *A set of instances **S** is shattered by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.*

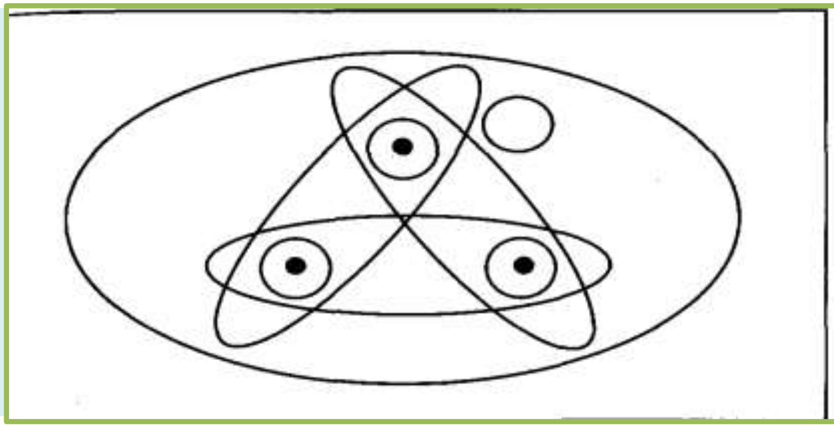


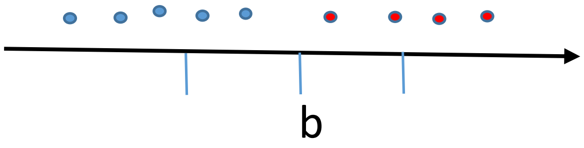
Fig.7.3 illustrates a set S of three instances that is **shattered** by the hypothesis space. Notice that each of the **2^3 dichotomies** of these three instances is covered by some hypothesis.

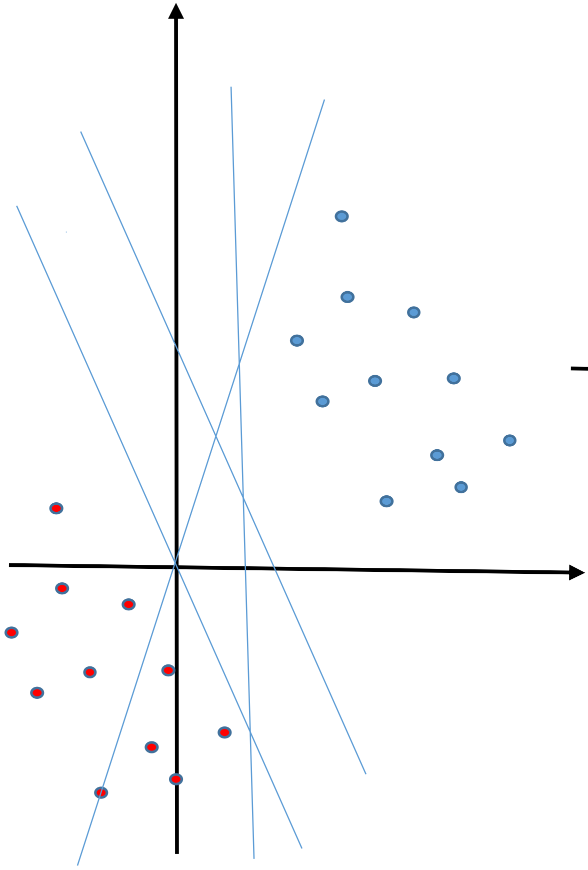
Note that if a set of instances is not shattered by a hypothesis space, then there must be some concept (dichotomy) that can be defined over the instances, but that cannot be represented by the hypothesis space.

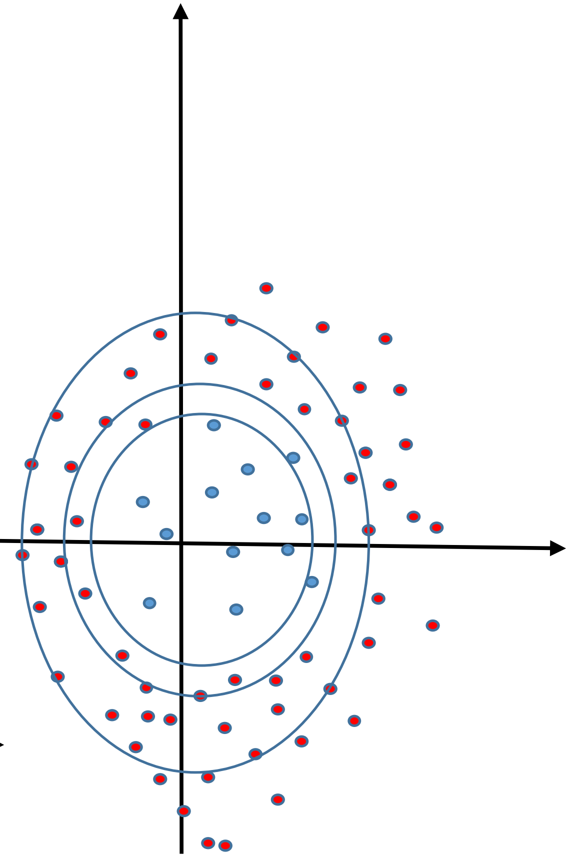
The ability of H to shatter a set of instances is thus a measure of its capacity to represent target concepts defined over these instances.

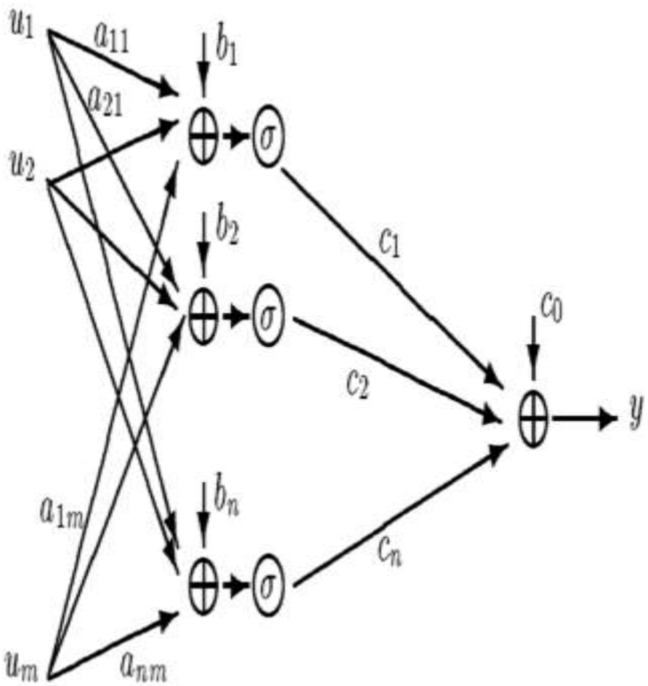
Hypothesis Class

- An **hypothesis class H** is a set of models



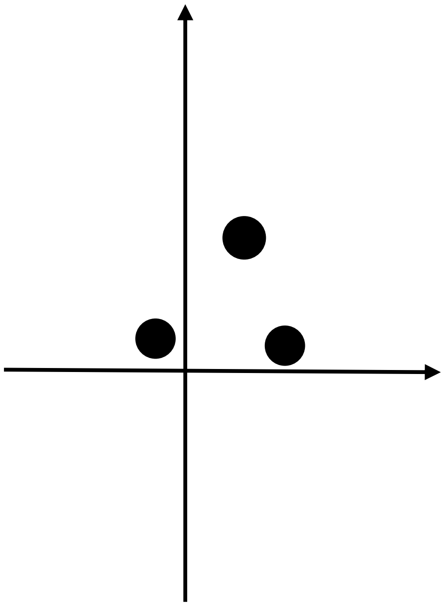


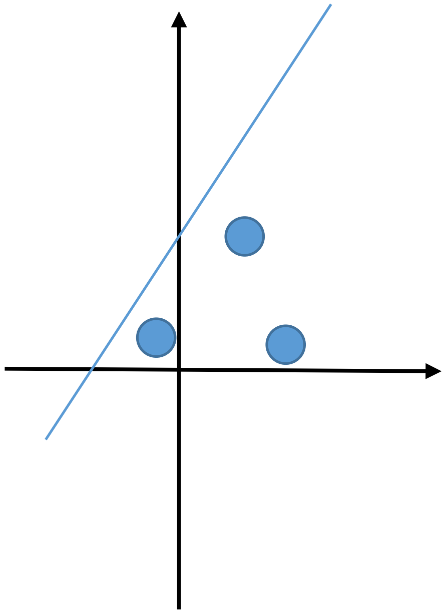




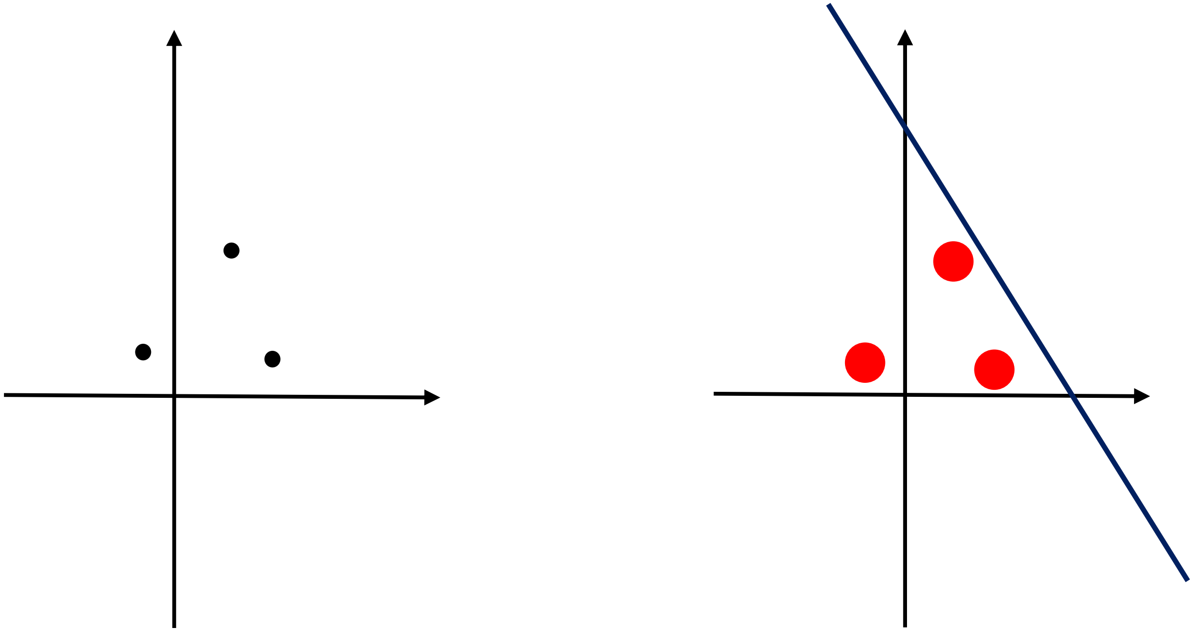
$sign(x^2$

VC Dimension – Shattering

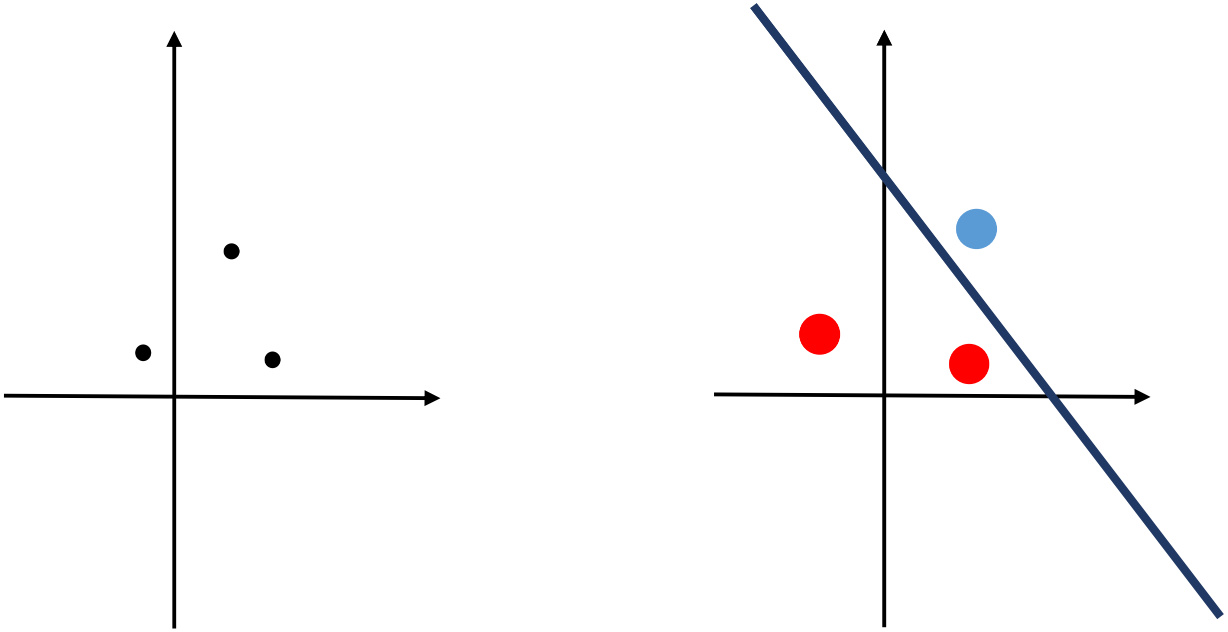




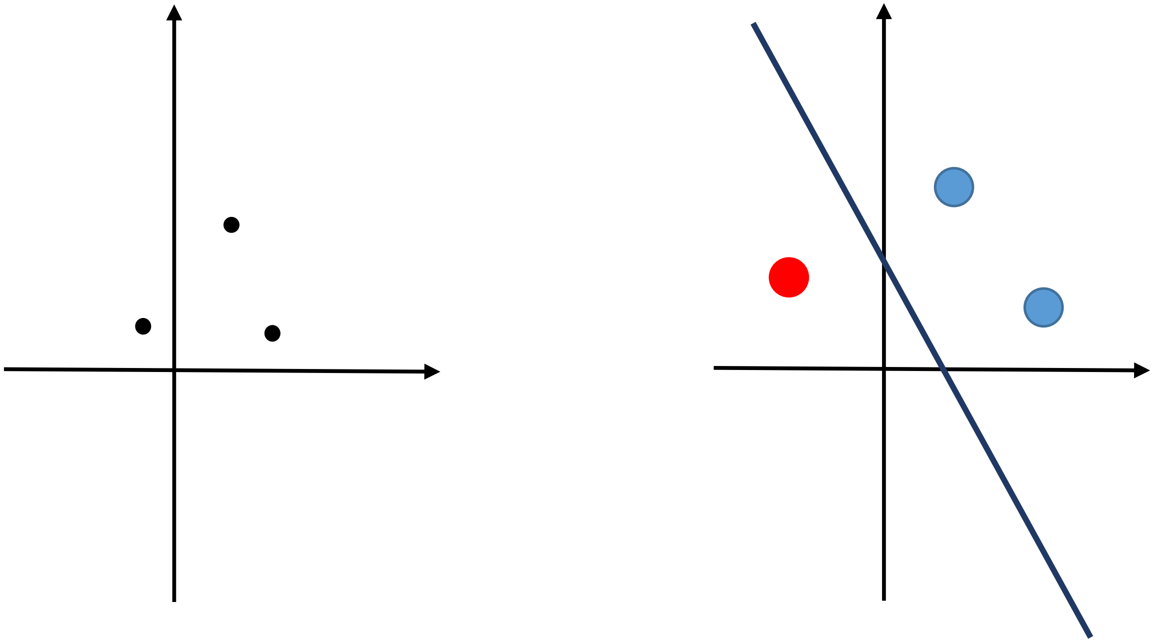
VC Dimension – Shattering



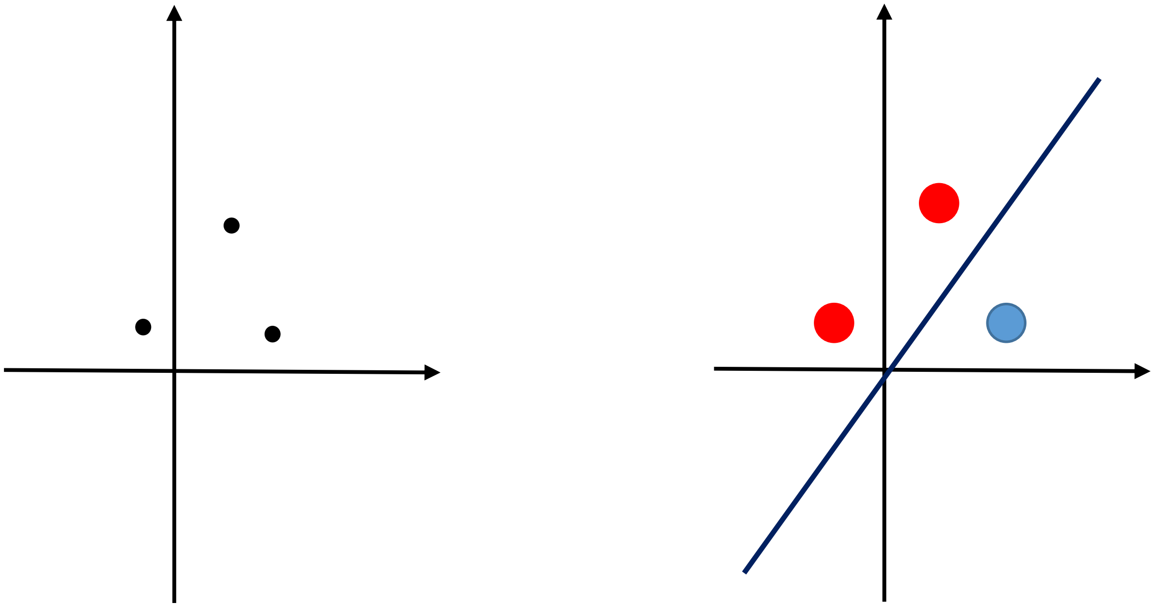
VC Dimension – Shattering



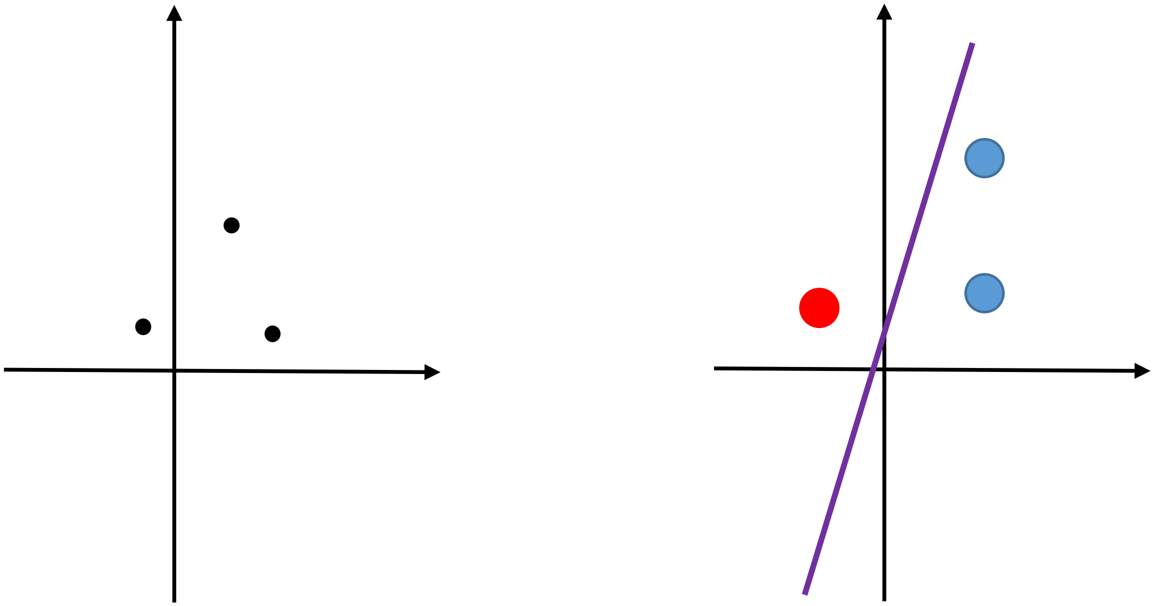
VC Dimension – Shattering



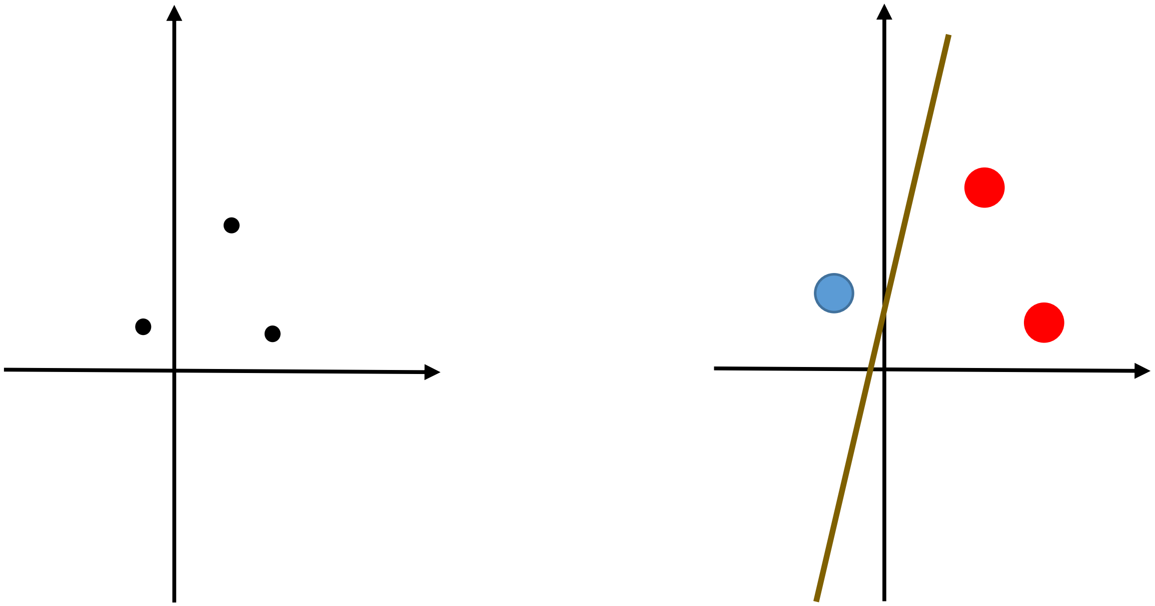
VC Dimension – Shattering



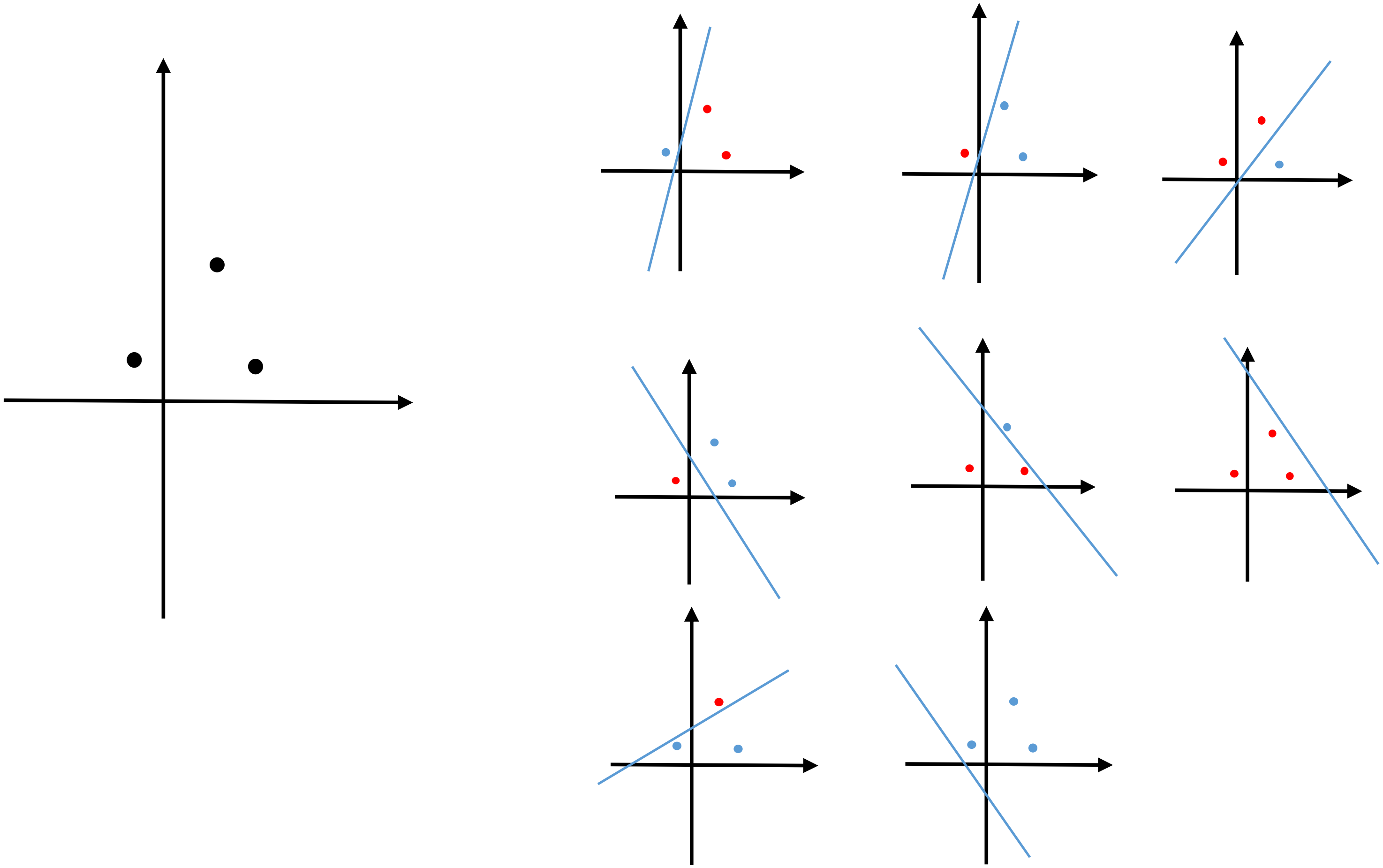
VC Dimension – Shattering

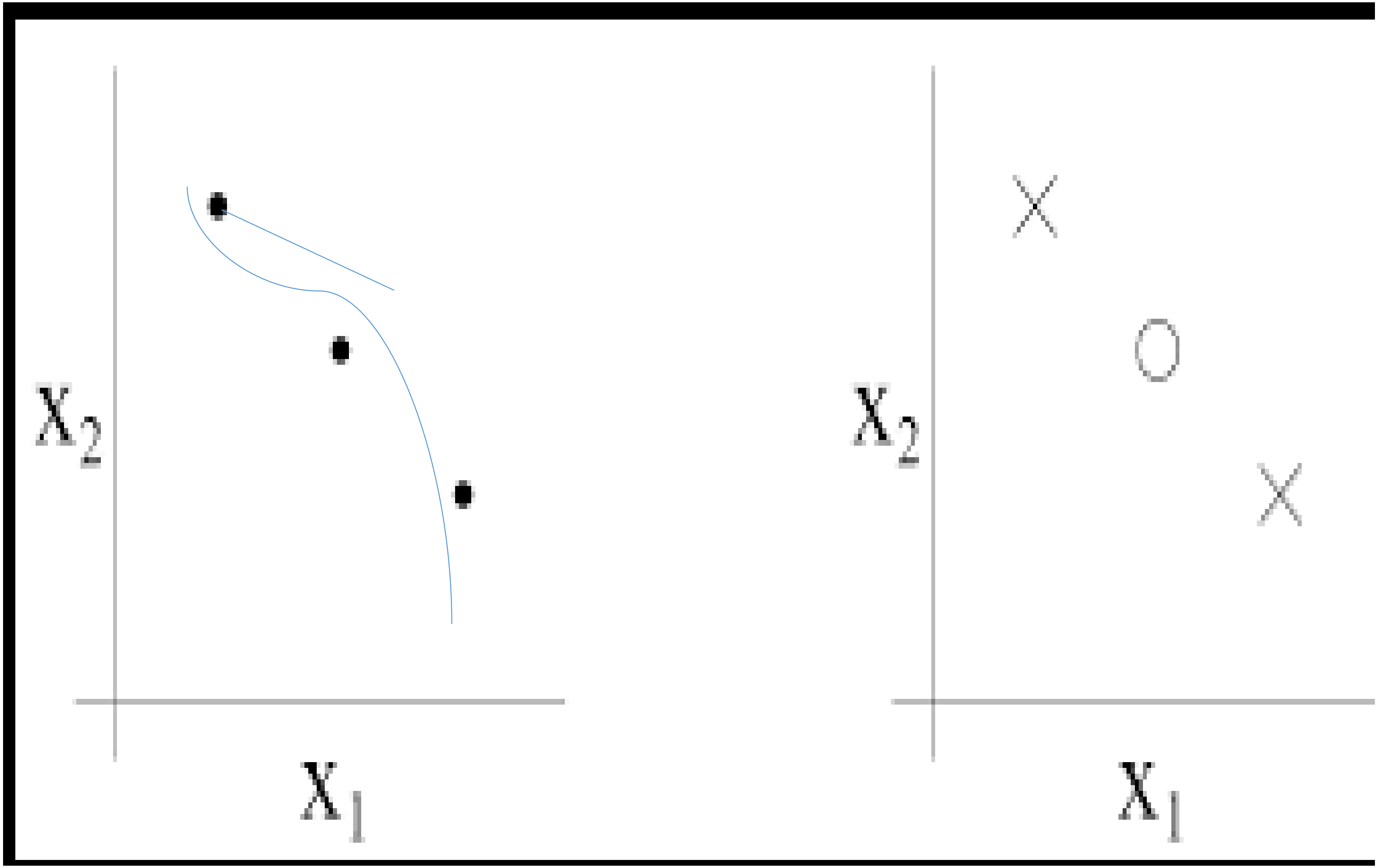


VC Dimension – Shattering



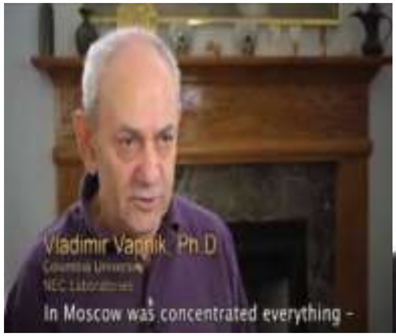
VC Dimension – Shattering



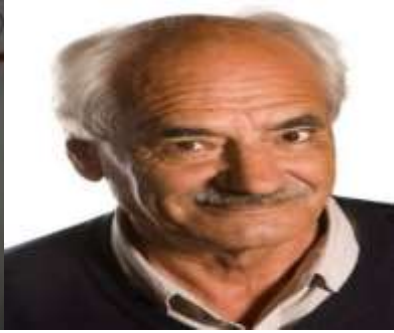

X_2
X_1
X_2
X
O
X
X_1

7.4.2 The Vapnik-Chervonenkis Dimension

The ability to shatter a set of instances is closely related to the inductive bias of a hypothesis space.



Vapnik



Chervonenkis

Background(From Chapter 2) that an unbiased hypothesis space is one capable of representing every possible concept (dichotomy) definable over the instance space X.

Put briefly, an unbiased hypothesis space H is one that shatters the instance space X.

What if H cannot shatter X, but can shatter some large subset S of X?

Intuitively, it seems reasonable to say that the larger the subset of X that can be shattered, the more expressive H.
The VC dimension of H is precisely this measure.

Definition: The ***VC(H) of hypothesis space H*** defined over instance space ***X*** *is the size of the largest finite subset of* ***X*** *shattered* by ***H.***

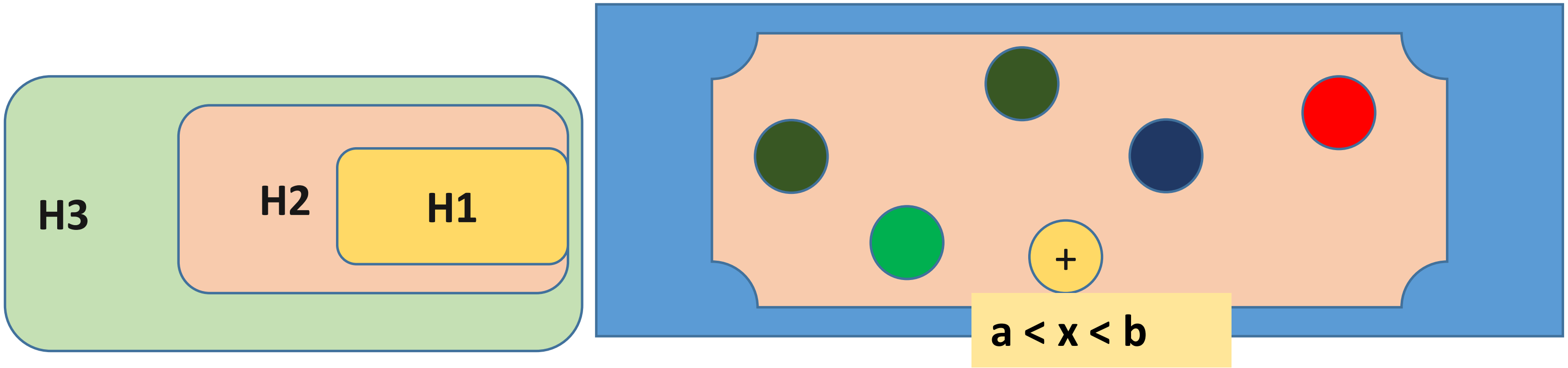
If arbitrarily large finite sets of X can be shattered by H, then VC(H) = ∞.

Note that for any finite H, $\mathbf{VC(H) \leq \log_2 IHI}$. *To see this, suppose that* VC(H) = d. Then H will require ***2^d distinct*** *hypotheses to shatter d instances.*

Hence, $\mathbf{2d \leq IHI}$, and $\mathbf{d = VC(H) \leq \log_2(H).}$

7.4.2.1 ILLUSTRATIVE EXAMPLES

Suppose the instance space X is the set of real Numbers $X = \Re$ *(e.g., describing the height of people), and H the set of intervals* on the real number line. In other words, H is the set of hypotheses of the form **a** < ***x* < *b*,** *where **a** and **b** may be any real constants. What is **VC(H)?***



To answer this question, we must find the largest subset of X that can be shattered by H.

$S = \{ 3.1 , 5.7 \}$ **Q. Can S be shattered by H?** **Ans: YES**

H1	**=**	$1 < x < 2$	**“ Neither “**
H2	**=**	$1 < x < 4$	**“ Covering ONE”**
H3	**=**	$4 < x < 7$	**“ Covering next (both) “**
H4	**=**	$1 < x < 7$	**“ covering All”**

Together, they represent each of the four dichotomies over S, covering neither instance, either one of the instances, and both of the instances, respectively. Since we have found a set of size two that can be shattered by H, we know the VC dimension of H is at least two.

Is there a set of size **three** that can be shattered?

Consider a set **S = { x0, x1, x2}** containing three arbitrary instances.

Assume **x0 < x1 < x2.** Clearly this set cannot be shattered,

because the dichotomy that includes **x0** and **x2**, but not **x1,**

cannot be represented by a single closed interval**.**

Therefore, no subset S of size three can be shattered,

and VC(H) = 2. **Note here that H is infinite, but VC(H) finite.**

Next consider the set X of instances corresponding to points on the x, y plane, (Fig. 7.4).

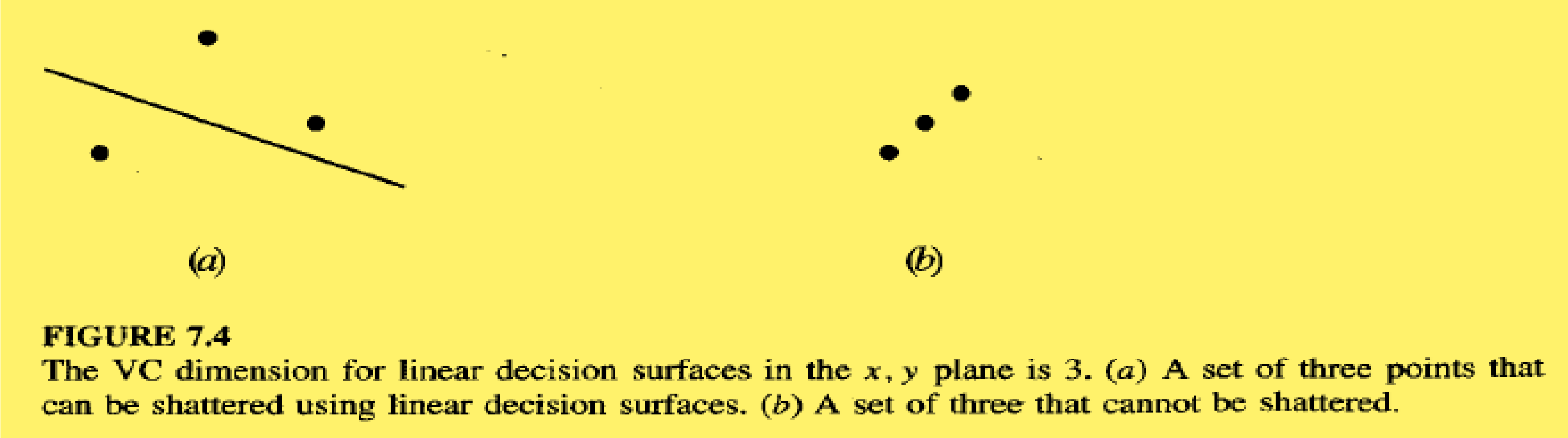


FIGURE 7.4
The VC dimension for linear decision surfaces in the x, y plane is 3. (*a*) A set of three points that can be shattered using linear decision surfaces. (*b*) A set of three that cannot be shattered.

Let H be the set of all linear decision surfaces in the plane. In other words, H is the hypothesis space corresponding to a single perceptron unit with two inputs.

What is the VC dimension of this H?

It is easy to see that any two distinct points in the plane can be shattered by H, because we can find four linear surfaces that include neither, either, or both points.

What about sets of 3 points? As long as the points are **not colinear,** we will be able to **find 2^3 linear surfaces** that shatter them. Of course 3 **colinear** points cannot be shattered (for the same reason that the 3 points on the real line could not be shattered in the previous example).

What is VC(H) in this case - 2 or 3? It is at least 3.

The definition of VC dimension indicates that if we find any set of instances of size **d** that can be shattered, then VC(H) ≥ *d.*

To show that VC(H) < d, we must show that no set of size **d** can be shattered. In this example, no sets of size four can be shattered, so VC(H) = 3. More generally, it can be shown that the VC dimension of linear decision surfaces in an **r** dimensional space (i.e., the VC dimension of a perceptron with r inputs) is **r + 1**.

As one final example, suppose each instance in X is described by the conjunction of exactly 3 Boolean literals, and suppose that each hypothesis in H is described by the conjunction of up to 3 Boolean literals. What is VC(H)?

We can show that it is at least 3, as follows.
Represent each instance by a 3-bit string
corresponding to the values of each of its three literals **b1**, **b2**, and **b3**. Consider the following set of three instances:

$\text{instance}_1 = 100$
$\text{instance}_2 = 010$
$\text{instance}_3 = 001$

This set of 3 instances can be shattered by H, because a hypothesis can be constructed for any desired dichotomy as follows: If the dichotomy is to exclude **instance$_i$** , add the literal **¬b**$_i$ to the hypothesis.

For example, suppose we wish to include instance$_2$, but exclude instance$_1$ and instance$_3$. Then we use the hypothesis ¬b1 Λ ¬ b3. This argument easily extends from 3 features to n. Thus, the VC dimension for conjunctions of n Boolean literals is at least n. In fact, it is exactly n, though showing this is more difficult, because it requires demonstrating that no set of **n + 1** instances can be shattered.

7.5 THE MISTAKE BOUND MODEL OF LEARNING

COLT considers a variety of different settings and questions. Different learning settings that have been studied vary by —

- how the training examples are generated (e.g., passive observation of random examples, active querying by the learner),

- noise in the data (e.g., noisy or error-free),

- the definition of success (e.g., the target concept must be learned exactly, or only probably and approximately),

- assumptions made by the learner (e.g., regarding the distribution of instances and whether $C \subseteq H$), and

the measure according to which the learner is evaluated (e.g., number of training examples, number of mistakes, total time).

In the mistake bound model of learning, the learner is evaluated by the total number of mistakes it makes before it converges to the correct hypothesis. As in the PAC setting, we assume the learner receives a sequence of training examples. However, here we demand that upon receiving each example x, the learner must predict the target value c(x), before it is shown the correct target value by the trainer.

The question considered is-

"*How many mistakes will the learner make in its predictions before it learns the target concept*?”

This question is significant in practical settings where learning must be done while the system is in actual use, rather than during some off-line training stage.

For example, if the system is to learn to predict which credit card purchases should be approved and which are fraudulent, based on data collected during use, then we are interested in minimizing the total number of mistakes it will make before converging to the correct target function. Here the total number of mistakes can be even more important than the total number of training examples.

Usually Skimming card readers or Skimmers will be placed in ATMs or POS machines . Skimmer is a small electronic device which is capable of capturing the data present in magnetic strips of the cards..

Skimming may take place during a legitimate transaction at a business. Such fraudulent activities can happen mostly in shopping outlets and restaurants. E.g., in a restaurant your card may be taken away when the bill is being settled and may use your card for regular transaction, also for capturing the card details. This captured card details will be misused by the scamsters