

CS6375: Machine Learning

Gautam Kunapuli

Machine Learning Theory

*Most of these slides are slightly modified from the originals created by **Malik-Magdon Ismail** for the **Machine Learning from Data** course at Rensselaer Polytechnic Institute and the textbook **Learning from Data: A Short Course** (co-authored with Yaser S. Abu-Mostafa and Hsuan-Tien Lin).*

Please see <http://www.cs.rpi.edu/~magdon/courses/learn/slides.html> for the original slides and <http://amlbook.com/index.html> for additional resources and the textbook.



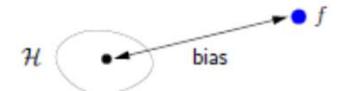
THE UNIVERSITY OF TEXAS AT DALLAS

Erik Jonsson School of Engineering and Computer Science

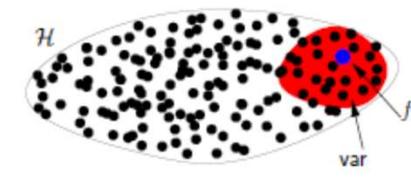
The Bias-Variance Decomposition

$$\begin{aligned}
 \text{test error} & \quad \text{learned model} \quad \text{true model} \\
 E_{\text{out}}(\mathbf{x}) &= \mathbb{E}_{\mathcal{D}} [(g^{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2] \\
 &= \mathbb{E}_{\mathcal{D}} [g^{\mathcal{D}}(\mathbf{x})^2 - 2g^{\mathcal{D}}(\mathbf{x})f(\mathbf{x}) + f(\mathbf{x})^2] \\
 &= \mathbb{E}_{\mathcal{D}} [g^{\mathcal{D}}(\mathbf{x})^2] - 2\bar{g}(\mathbf{x})f(\mathbf{x}) + f(\mathbf{x})^2 \quad \leftarrow \text{understand this; the rest is just algebra} \\
 &= \mathbb{E}_{\mathcal{D}} [g^{\mathcal{D}}(\mathbf{x})^2] - \bar{g}(\mathbf{x})^2 + \bar{g}(\mathbf{x})^2 - 2\bar{g}(\mathbf{x})f(\mathbf{x}) + f(\mathbf{x})^2 \\
 &= \mathbb{E}_{\mathcal{D}} [g^{\mathcal{D}}(\mathbf{x})^2] - \bar{g}(\mathbf{x})^2 + (\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2 \\
 &\quad \underbrace{\hspace{10em}}_{\text{var}(\mathbf{x})} \quad \underbrace{\hspace{10em}}_{\text{bias}(\mathbf{x})}
 \end{aligned}$$

$$E_{\text{out}}(\mathbf{x}) = \text{bias}(\mathbf{x}) + \text{var}(\mathbf{x})$$



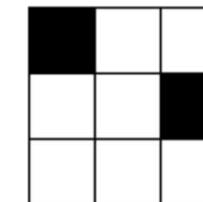
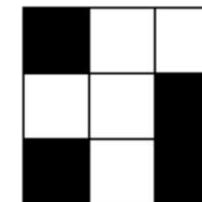
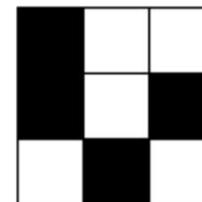
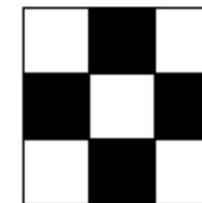
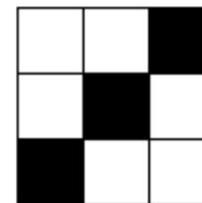
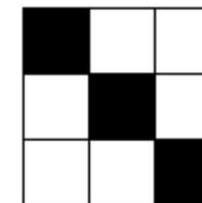
Very small model



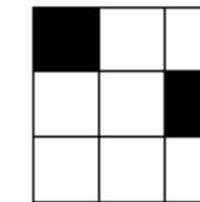
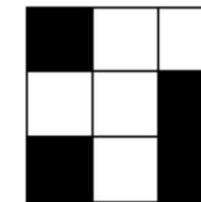
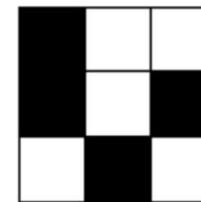
Very large model

If you take average over \mathbf{x} : $E_{\text{out}} = \text{bias} + \text{var}$

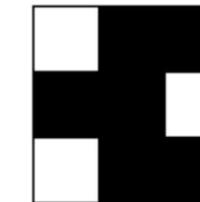
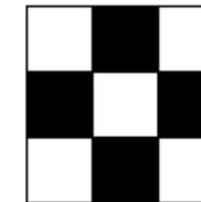
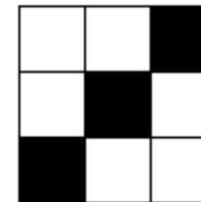
Example: A Simple Visual Learning Problem

 $f = -1$  $f = +1$  $f = ?$

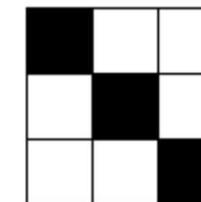
Example: A Simple Visual Learning Problem



$$f = -1$$



$$f = +1$$

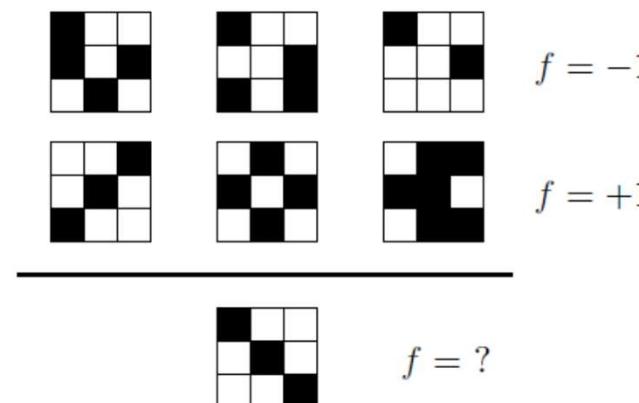


$$f = ?$$

- Did you say $f = +1$? (f is measuring symmetry.)
- Did you say $f = -1$? (f only cares about the top left pixel.)

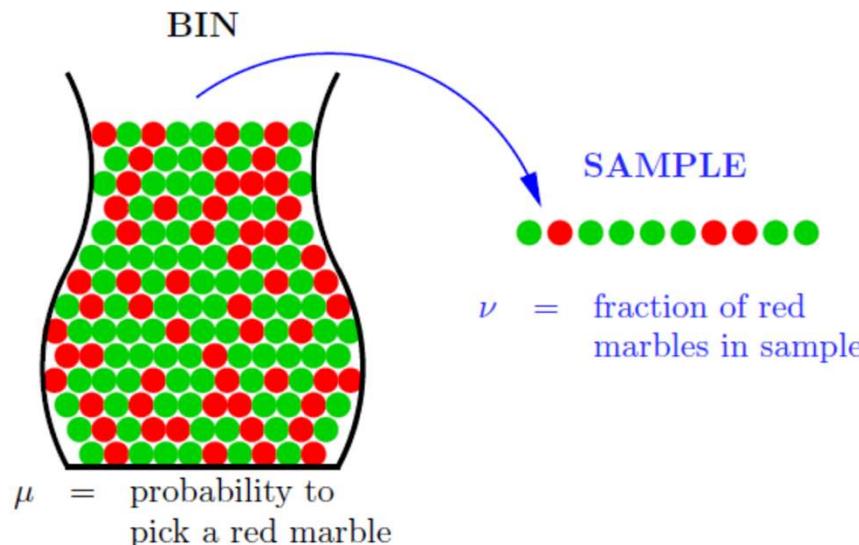
Who is correct? – we cannot *rule out either possibility*.

Example: A Simple Visual Learning Problem



- An easy visual learning problem just got very messy.
For *every* f that fits the data and is “+1” on the new point, there is one that is “−1”.
Since f is *unknown*, it can take on any value outside the data, no matter how large the data.
- This is called **No Free Lunch (NFL)**.
You cannot know anything *for sure* about f outside the data set *without* making assumptions.
- What now!
Is there *any hope* to know *anything* about f outside the data set *without* making assumptions about f ?
Yes, if we are willing to give up the “for sure”.

Population Mean From Sample Mean



The BIN Model

- Bin with red and green marbles.
- Pick a sample of N marbles *independently*.
- μ : probability to pick a red marble.
 ν : fraction of red marbles in the sample.

Sample \longrightarrow the data set $\longrightarrow \nu$
BIN \longrightarrow outside the data $\longrightarrow \mu$

Can we say anything about μ (**outside the data**) after observing ν (**the data**)?

ANSWER: No. It is *possible* for the sample to be all green marbles and the bin to be mostly red.

Then, why do we trust polling (e.g. to predict the outcome of the presidential election).

ANSWER: The bad case is *possible*, but not **probable**.

Probability to the Rescue: Hoeffding's Inequality

Hoeffding/Chernoff proved that, most of the time, ν cannot be too far from μ :

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

$$\mathbb{P}[|\nu - \mu| \leq \epsilon] \geq 1 - 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

We get to select any ϵ we want.

Probability to the Rescue: Hoeffding's Inequality

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

$$\mathbb{P}[|\nu - \mu| \leq \epsilon] \geq 1 - 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

box it and
memorize it



Example: $N = 1,000$; draw a sample and observe ν .

99% of the time $\mu - 0.05 \leq \nu \leq \mu + 0.05$ ($\epsilon = 0.05$)

99.9999996% of the time $\mu - 0.10 \leq \nu \leq \mu + 0.10$ ($\epsilon = 0.10$)

What does this mean? If I repeatedly pick a sample of size 1,000, observe ν and claim that

$$\mu \in [\nu - 0.05, \nu + 0.05], \quad (\text{the error bar is } \pm 0.05)$$

I will be right 99% of the time. On any particular sample you may be wrong, but not often.

How Did Probability Rescue Us?

this allows us to use the

- Key ingredient samples must be *independent*. *tools of probability*

If the sample is constructed in some arbitrary fashion, then indeed we cannot say anything.

Even with independence, ν can take on arbitrary values; but some values are way more likely than others.

This is what allows us to learn *something* – it is likely that $\nu \approx \mu$.

- The bound $2e^{-2\epsilon^2 N}$ does not depend on μ or the size of the bin

The bin can be infinite.

It's great that it does not depend on μ because μ is unknown; and we mean unknown.

dependence of the bound on the number of samples allows

- The key player in the bound $2e^{-2\epsilon^2 N}$ is N . *us to characterize the sample complexity*

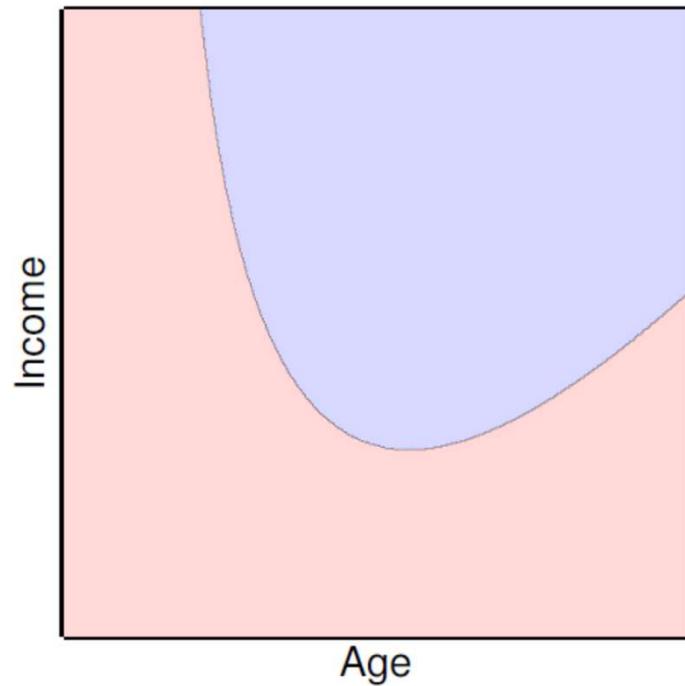
If $N \rightarrow \infty$, $\mu \approx \nu$ with very very very ... high probability, *but not for sure*.

Can you live with 10^{-100} probability of error?

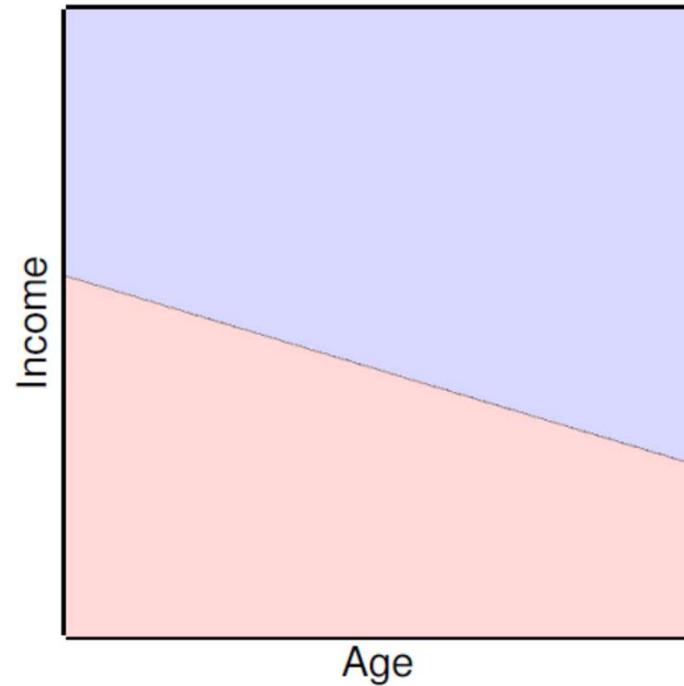
Relating the Bin to Learning

Note that we are currently considering just a single hypothesis

Target Function f



Fixed hypothesis h

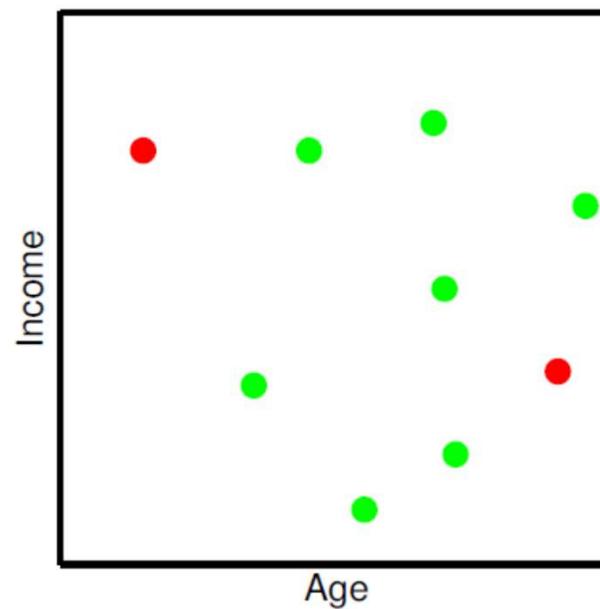
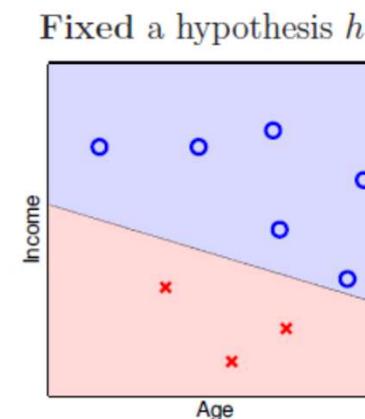
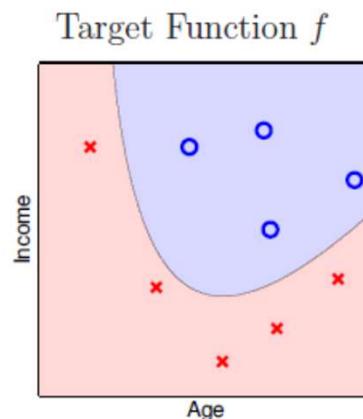


UNKNOWN

KNOWN

In learning, the unknown is an entire function f ; in the bin it was a single number μ .

Relating the Bin to Learning: Training Error



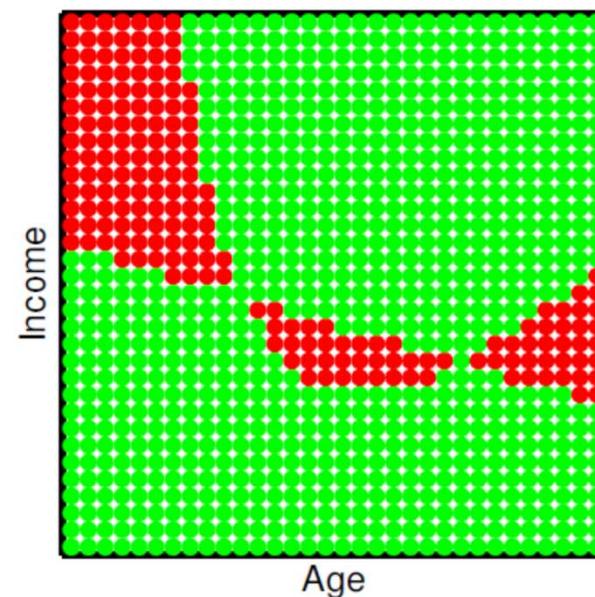
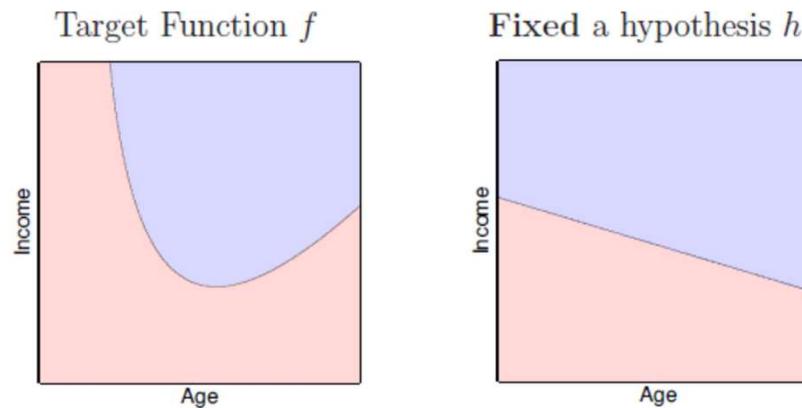
green data: $h(\mathbf{x}_n) = f(\mathbf{x}_n)$ correctly classified
 red data: $h(\mathbf{x}_n) \neq f(\mathbf{x}_n)$ misclassified

$E_{\text{in}}(h)$ = fraction of red data

←
in-sample ↑
 misclassified

KNOWN!

Relating the Bin to Learning: Test Error



green “marble”: $h(\mathbf{x}) = f(\mathbf{x})$ correctly classified
 red “marble”: $h(\mathbf{x}) \neq f(\mathbf{x})$ misclassified
 BIN: \mathcal{X}

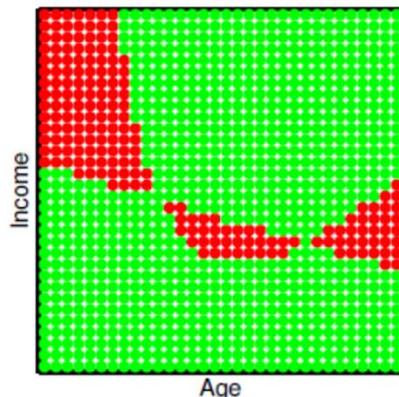
$$E_{\text{out}}(h) = \mathbb{P}_{\mathbf{x}}[h(\mathbf{x}) \neq f(\mathbf{x})]$$



out-of-sample

UNKNOWN

Relating the Bin to Learning: The Error Function



Unknown f and $P(x)$, fixed h

Learning

input space \mathcal{X}

$\textcolor{green}{x}$ for which $h(\textcolor{green}{x}) = f(\textcolor{green}{x})$

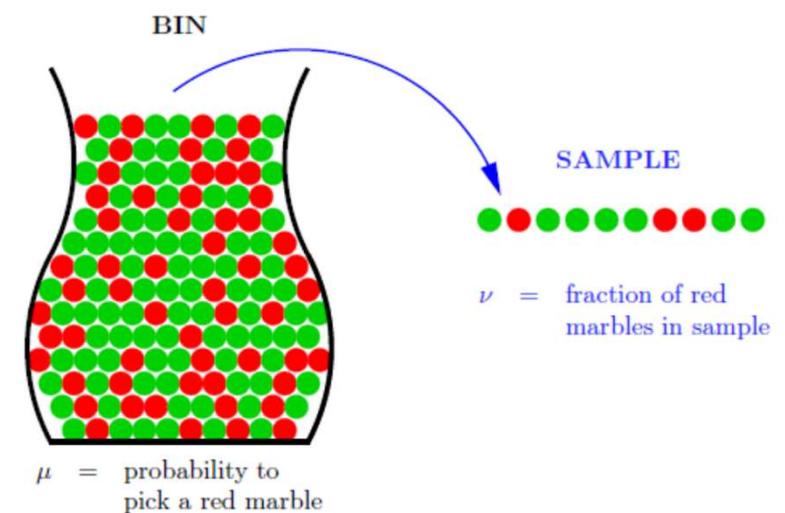
$\textcolor{red}{x}$ for which $h(\textcolor{red}{x}) \neq f(\textcolor{red}{x})$

$P(x)$

data set \mathcal{D}

Out-of-sample Error: $E_{\text{out}}(h) = \mathbb{P}_x[h(x) \neq f(x)]$

In-sample Error: $E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^N \llbracket h(x_n) \neq f(x_n) \rrbracket$



Bin Model

Bin

● green marble

● red marble

randomly picking a marble

sample of N marbles

μ = probability of picking a red marble

ν = fraction of red marbles in the sample

Hoeffding Inequality for Learning

$$\mathbb{P} [|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

$$\mathbb{P} [|E_{\text{in}}(h) - E_{\text{out}}(h)| \leq \epsilon] \geq 1 - 2e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

E_{in} is **random, but known**; E_{out} **fixed, but unknown**.

- If $E_{\text{in}} \approx 0 \implies E_{\text{out}} \approx 0$ (with high probability), i.e. $\mathbb{P}_{\mathbf{x}}[h(\mathbf{x}) \neq f(\mathbf{x})] \approx 0$;
We have learned something about the *entire* f : $f \approx h$ over \mathcal{X} (outside \mathcal{D})
- If $E_{\text{in}} \gg 0$, we're out of luck.
But, we have still learned something about the entire f : $f \not\approx h$; it is not very useful though.

That's Verification, Not Learning

Recall that we are currently considering just a single hypothesis

The entire previous argument assumed a **FIXED** h and then came the data.

- Given $h \in \mathcal{H}$, a sample can **verify** whether or not it is good (w.r.t. f):

if E_{in} is small, h is good, with high confidence.

if E_{in} is large, h is bad with high confidence.

We have no control over E_{in} . It is what it is.

- In learning, you actually try to **fit** the data, as with the perceptron model g results from searching an entire hypothesis set \mathcal{H} for a hypothesis with small E_{in} .

<u>Verification</u>	<u>Real Learning</u>
Fixed single hypothesis h	Fixed hypothesis set \mathcal{H}
h to be certified	g to be certified
h does not depend on \mathcal{D}	g results after searching \mathcal{H} to fit \mathcal{D}
No control over E_{in}	Pick best E_{in}

Bounding Learning with Finite Hypothesis Spaces

We can bound the performance of the best hypothesis by considering all the hypotheses in the hypothesis space

$$\mathbb{P}[|E_{\text{in}}(\mathbf{g}) - E_{\text{out}}(\mathbf{g})| > \epsilon] \leq 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

$$\mathbb{P}[|E_{\text{in}}(\mathbf{g}) - E_{\text{out}}(\mathbf{g})| \leq \epsilon] \geq 1 - 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

We don't care how g was obtained, *as long as it is from \mathcal{H}*

This is called the union bound, and can often be too loose to be meaningful or informative

Some Basic Probability

Events A, B

Implication

If $A \implies B$ ($A \subseteq B$) then $\mathbb{P}[A] \leq \mathbb{P}[B]$.

Union Bound

$\mathbb{P}[A \text{ or } B] = \mathbb{P}[A \cup B] \leq \mathbb{P}[A] + \mathbb{P}[B]$.

Bayes' Rule

$\mathbb{P}[A|B] = \frac{\mathbb{P}[B|A] \cdot \mathbb{P}[A]}{\mathbb{P}[B]}$

Proof: Let $M = |\mathcal{H}|$.

The event “ $|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon$ ” implies
“ $|E_{\text{in}}(h_1) - E_{\text{out}}(h_1)| > \epsilon$ ” OR … OR “ $|E_{\text{in}}(h_M) - E_{\text{out}}(h_M)| > \epsilon$ ”

So, by the implication and union bounds:

$$\begin{aligned} \mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon] &\leq \mathbb{P}\left[\bigcup_{m=1}^M |E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon\right] \\ &\leq \sum_{m=1}^M \mathbb{P}[|E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon], \\ &\leq 2Me^{-2\epsilon^2 N}. \end{aligned}$$

(The last inequality is because we can apply the Hoeffding bound to each summand)

Probably Approximately Correct Bound

$$\mathbb{P}[|E_{\text{in}}(\mathbf{g}) - E_{\text{out}}(\mathbf{g})| > \epsilon] \leq 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

$$\mathbb{P}[|E_{\text{in}}(\mathbf{g}) - E_{\text{out}}(\mathbf{g})| \leq \epsilon] \geq 1 - 2|\mathcal{H}|e^{-2\epsilon^2 N}, \quad \text{for any } \epsilon > 0.$$

Theorem. With probability at least $1 - \delta$,

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}.$$

bias variance

We don't care how g was obtained, *as long as* $g \in \mathcal{H}$

Drawbacks of this bound

- Bound is **too loose** to be practical as it considers the size of the hypothesis space to measure hypothesis **complexity**
- Bound **breaks down for infinite hypothesis spaces**, for example, linear classifiers

Proof: Let $\delta = 2|\mathcal{H}|e^{-2\epsilon^2 N}$. Then

$$\mathbb{P}[|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon] \geq 1 - \delta.$$

In words, with probability at least $1 - \delta$,

$$|E_{\text{in}}(g) - E_{\text{out}}(g)| \leq \epsilon.$$

This implies

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \epsilon.$$

From the definition of δ , solve for ϵ :

$$\epsilon = \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}.$$

Probably Approximately Correct Bound

- (1) $E_{\text{out}}(g) \approx E_{\text{in}}(g)$.
- (2) $E_{\text{in}}(g) \approx 0$.

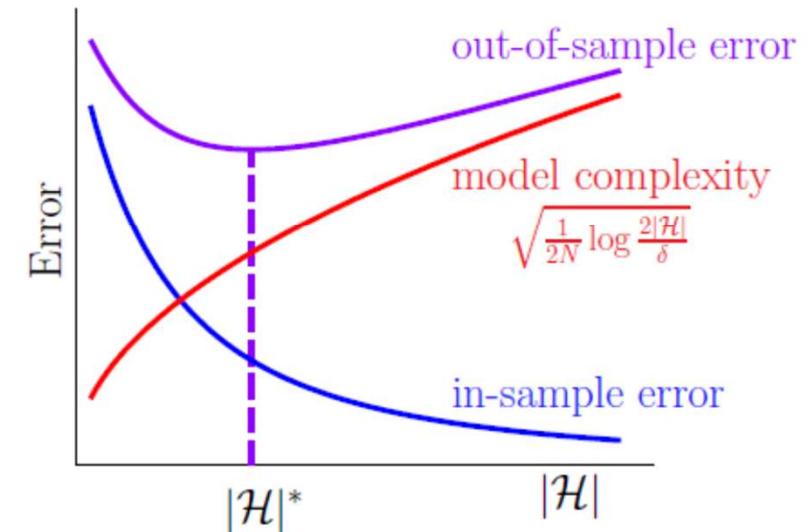
Together, these ensure $E_{\text{out}} \approx 0$.

How to verify (1) since we do not know E_{out}
 – must *ensure* it theoretically - Hoeffding.

We can ensure (2) (for example PLA)
 – modulo that we can guarantee (1)

There is a tradeoff:

- Small $|\mathcal{H}| \implies E_{\text{in}} \approx E_{\text{out}}$
- Large $|\mathcal{H}| \implies E_{\text{in}} \approx 0$ is more likely.



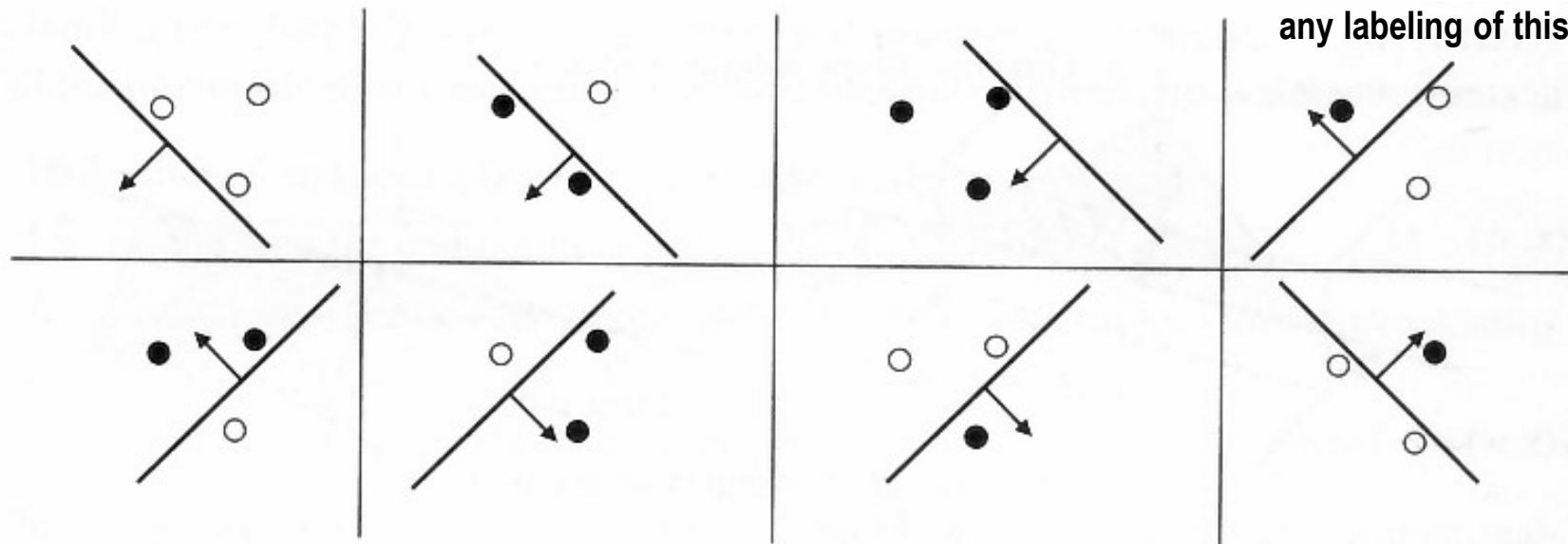
The Vapnik-Chervonenkis Dimension

The **VC dimension** is a measure of **complexity** of a learner, that is, it is a measure of its **representation power**

- **higher complexity** means a classifier has a greater possibility of capturing the true hypothesis, but more likely to **overfit**
- **lower complexity** means that the classifier has a smaller possibility of capturing the true hypothesis, but more likely to **underfit**

We say a classifier $f(\mathbf{x})$ can **shatter** a set of h points $\mathbf{x}_1, \dots, \mathbf{x}_h$ if and only if for all label assignments $y_1 \dots y_h$, $f(\mathbf{x})$ can achieve zero error on the training data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_h, y_h)$

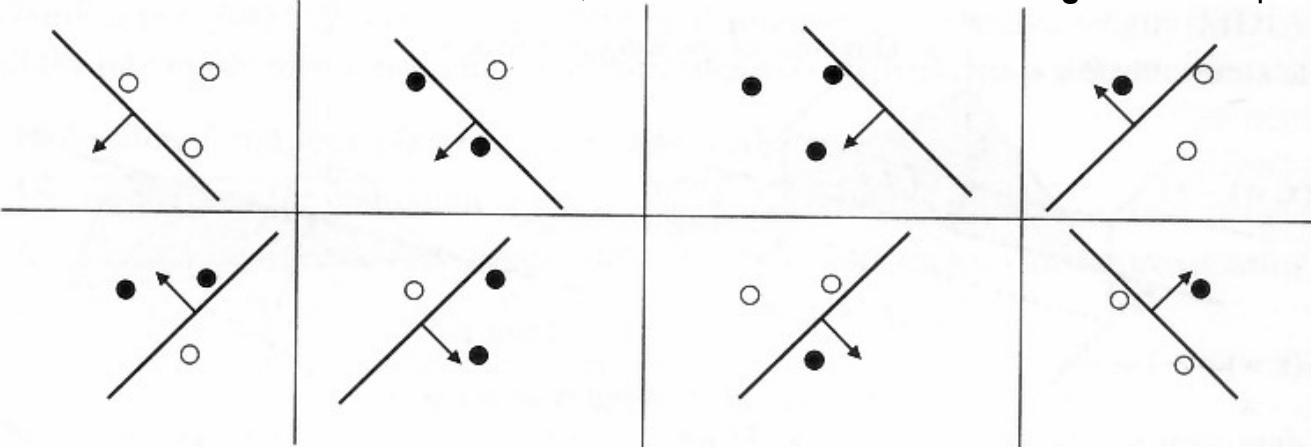
a linear classifier in 2 dimensions can shatter
any labeling of this set of three points



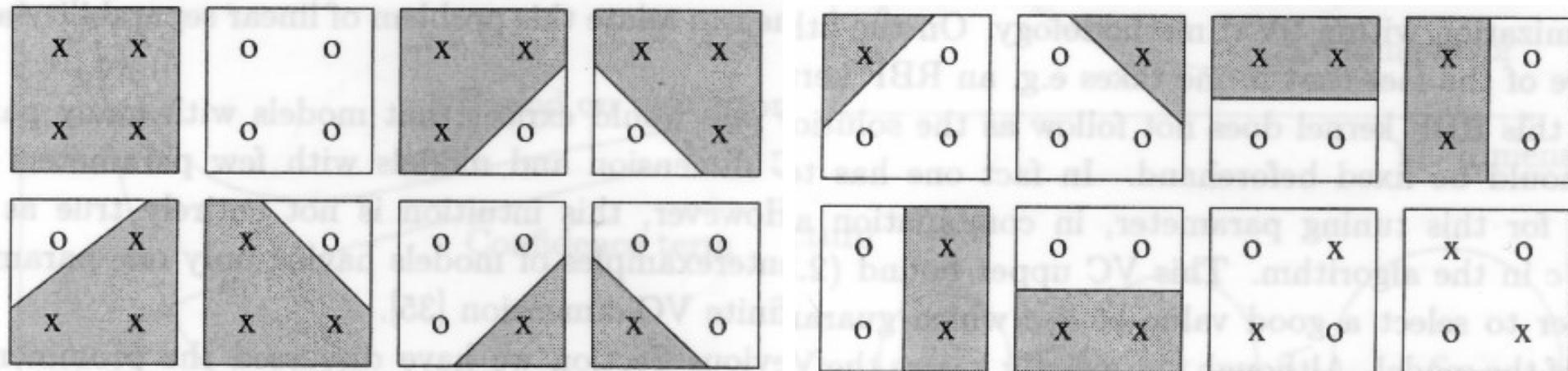
The Vapnik-Chervonenkis Dimension

The **VC dimension** is defined as the **maximum number of points** h that can be arranged so that $f(x)$ can shatter them.

for a **linear classifier** in 2 dimensions, there exists at least one arrangement of 3 points that can be shattered. So $h_{VC} \geq 3$.



for a **linear classifier** in 2 dimensions, there exists no **one arrangement** of 4 points that can be shattered. So $h_{VC} < 4$

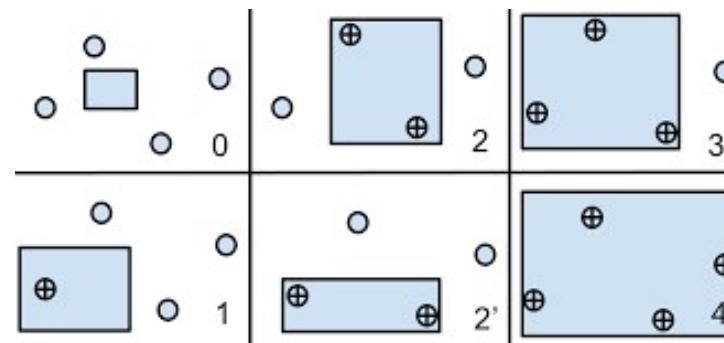


for a **linear classifier** in 2 dimensions, $h_{VC} = 3$.

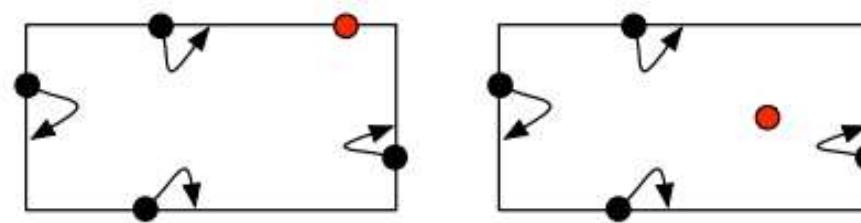
The Vapnik-Chervonenkis Dimension

The **VC dimension** is defined as the **maximum number of points** h that can be arranged so that $f(x)$ can shatter them.

for **axis-aligned rectangles**, there exists **at least one arrangement** of 4 points that can be shattered. So $h_{VC} \geq 4$.



for **axis-aligned rectangles**, there exists **at least one arrangement** of 4 points that can be shattered. So $h_{VC} < 5$.

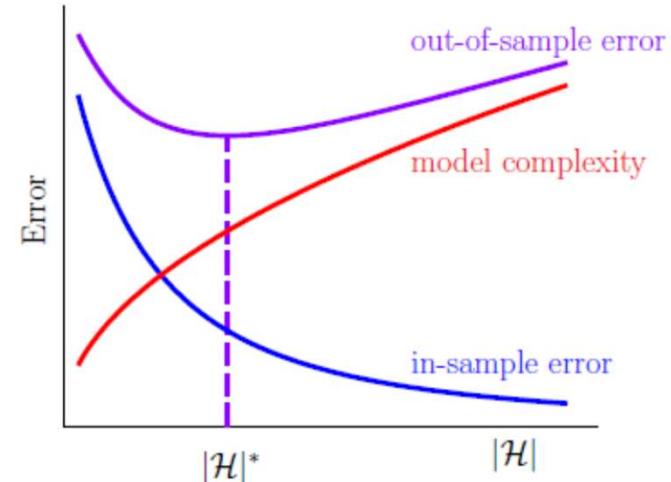


Our minimum enclosing rectangle that allows us to select all five points is defined by only four points – one for each edge. So, it is clear that the fifth point must lie either on an edge or on the inside of the rectangle. This prevents us from selecting four points without the fifth.

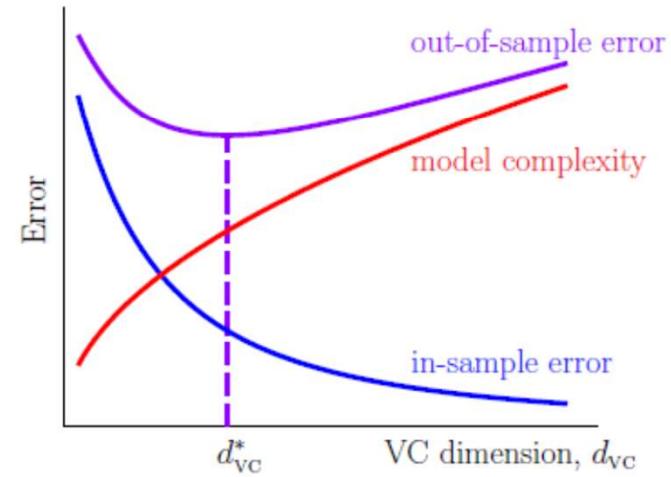
for **axis-aligned rectangles**, $h_{VC} = 4$.

Bounding Learning with Infinite Hypothesis Spaces

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \sqrt{\frac{1}{2N} \log \frac{2|\mathcal{H}|}{\delta}}$$



$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \underbrace{\sqrt{\frac{8}{N} \log \frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta}}}_{\text{penalty for model complexity } \Omega(d_{\text{VC}})}$$



Sample Complexity

How Many Data Points Do We Need?

Set the error bar at ϵ .

$$\epsilon = \sqrt{\frac{8}{N} \ln \frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta}}$$

Solve for N :

$$N = \frac{8}{\epsilon^2} \ln \frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta} = O(d_{\text{VC}} \ln N)$$

Example. $d_{\text{VC}} = 3$; error bar $\epsilon = 0.1$; confidence 90% ($\delta = 0.1$). A simple iterative method works well. Trying $N = 1000$ we get

$$N \approx \frac{1}{0.1^2} \log \left(\frac{4(2000)^3 + 4}{0.1} \right) \approx 21192.$$

We continue iteratively, and converge to $N \approx 30000$. If $d_{\text{VC}} = 4$, $N \approx 40000$; for $d_{\text{VC}} = 5$, $N \approx 50000$.

$(N \propto d_{\text{VC}}$, but gross overestimates)

Practical Rule of Thumb: $N = 10 \times d_{\text{VC}}$

The Test Set

- Another way to estimate $E_{\text{out}}(g)$ is using a *test set* to obtain $E_{\text{test}}(g)$.
- E_{test} is better than E_{in} : you don't pay the price for fitting.
You can use $|\mathcal{H}| = 1$ in the Hoeffding bound with E_{test} .
- Both a test and training set have variance.
The training set has *optimistic bias* due to selection – fitting the data.
A test set has no bias.
- The price for a test set is fewer training examples. (why is this bad?)
 $E_{\text{test}} \approx E_{\text{out}}$ but now E_{test} may be bad.