# CS 446/ECE 449: Machine Learning

## Lecture 10: PAC Learning Theory (II)

Han Zhao

02/15/2024

ILLINOIS
Computer Science
GRAINGER COLLEGE OF ENGINEERING

# Recap: Bayes Error

Bayes error rate:

$$\text{Bayes error: } \varepsilon_\mu^* := \inf_{f:\mathcal{X}\to\mathcal{Y}} \varepsilon_\mu(f)$$

Binary classification:

Bayes error rate: $\varepsilon_\mu^* = \mathbb{E}\min\left\{\Pr(Y=1\,|\,X), \Pr(Y=0\,|\,X)\right\}$

Bayes optimal classifier: $f_{\text{Bayes}}(X) := \begin{cases} 1 & \text{if } \Pr(Y=1|X) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$

Regression with squared loss:

Bayes error rate: $\varepsilon_\mu^* = \mathbb{E}\mathrm{Var}[Y\,|\,X]$

Bayes optimal regressor: $f_{\text{Bayes}}(X) = \mathbb{E}[Y\,|\,X]$

# Recap: Error Decomposition

## Error decomposition: $\forall f \in \mathscr{F}$:

$$\varepsilon_\mu(f) = \boxed{\varepsilon_\mu(f) - \inf_{f \in \mathscr{F}} \varepsilon_\mu(f)} + \boxed{\inf_{f \in \mathscr{F}} \varepsilon_\mu(f) - \varepsilon_\mu^*} + \varepsilon_\mu^*$$

**Estimation error**
(depending on the size of our data and $\mathcal{F}$)

**Approximation error**
(depending on the expressiveness of $\mathcal{F}$)

**Bayes error**
(depending on the inherent noise in the data)

- Often the case, there is a trade-off between the estimation error and the approximation error
- If $\mathscr{F}$ is more expressive, then the approximation error gets smaller but the estimation error gets larger
- If $\mathscr{F}$ is more restricted, then the approximation error gets larger but the estimation error gets smaller (assume the size of training data is fixed)

# Lecture Today

- Probably Approximately Correct (PAC) framework
- Generalization analysis
- Vapnik–Chervonenkis dimension (VC dim)

# Probably Approximately Correct (PAC)

The learning process:

- We can choose a predictor $f$ from some pre-defined class of functions $\mathscr{F}$, e.g., the class of linear predictors, decision trees, kernel machines, neural networks, etc.

We also have our training data $\mathscr{D} := \{(x^{(i)}, y^{(i)})\}_{i=1}^{n} \sim \mu$ sampled independently and identically (iid) from the underlying distribution $\mu$ over $\mathscr{X} \times \mathscr{Y}$

We can then talk about two error measures (classification):

$$\text{Training error: } \hat{\varepsilon}_{\mathscr{D}}(f) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(f(x^{(i)}) \neq y^{(i)})$$

$$\text{Test error: } \varepsilon_{\mu}(f) := \mathbb{E}_{\mu}\left[\mathbb{I}(f(X) \neq Y)\right] = \Pr_{\mu}(f(X) \neq Y)$$

We are interested in finding $f$ that minimizes the test error but we can only observe the training error

# Probably Approximately Correct (PAC)

For a given hypothesis class $\mathscr{F}$, can we relate the training and test errors?

Generalization error/gap: $|\hat{\varepsilon}_{\mathscr{D}}(f) - \varepsilon_{\mu}(f)|$

Note:

- The generalization error is a random variable due to the randomness in $\mathscr{D} \sim \mu$
- For any fixed $f$, we would expect the generalization error to be small:

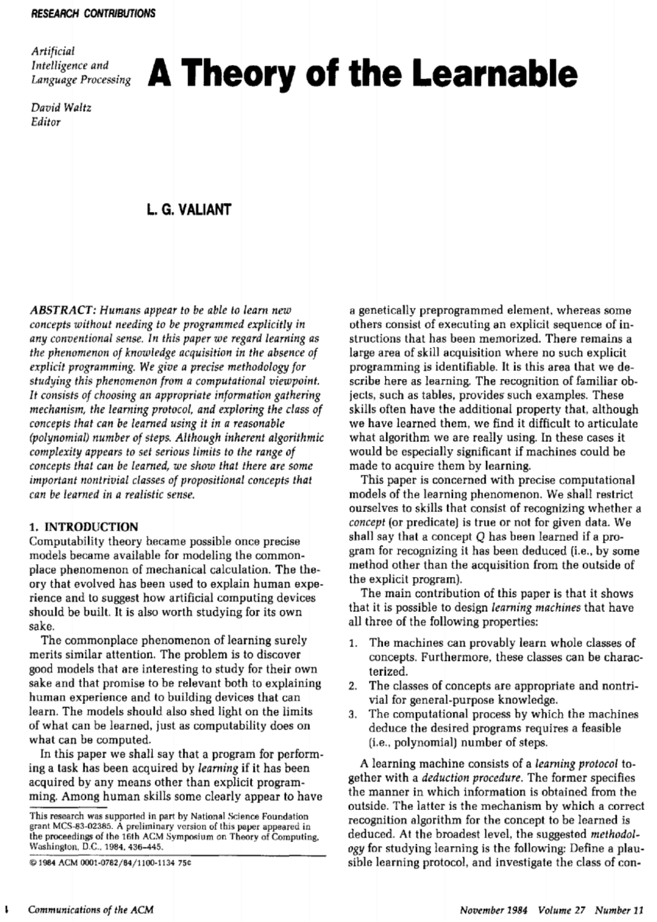
$$\mathbb{E}\left[\hat{\varepsilon}_{\mathscr{D}}(f)\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(f(x^{(i)}) \neq y^{(i)})\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbb{I}(f(x^{(i)}) \neq y^{(i)})\right] = \varepsilon_{\mu}(f)$$

The argument above is in expectation, and it does not necessarily apply to our specific training data $\mathscr{D}$. How about we consider a high-probability guarantee instead?

# Probably Approximately Correct (PAC)

Probably Approximately Correct (PAC, Valiant, CACM 1984)

- (Informal) A framework to quantify the meaning of learning a concept from samples
- With high probability (P), the learned predictor will have low generalization error (AC)
- No distributional assumption



RESEARCH CONTRIBUTIONS

*Artificial*
*Intelligence and*
*Language Processing*

*David Waltz*
*Editor*

## A Theory of the Learnable

L. G. VALIANT

**ABSTRACT:** *Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming. We give a precise methodology for studying this phenomenon from a computational viewpoint. It consists of choosing an appropriate information gathering mechanism, the learning protocol, and exploring the class of concepts that can be learned using it in a reasonable (polynomial) number of steps. Although inherent algorithmic complexity appears to set serious limits to the range of concepts that can be learned, we show that there are some important nontrivial classes of propositional concepts that can be learned in a realistic sense.*

### 1. INTRODUCTION

Computability theory became possible once precise models became available for modeling the commonplace phenomenon of mechanical calculation. The theory that evolved has been used to explain human experience and to suggest how artificial computing devices should be built. It is also worth studying for its own sake.

The commonplace phenomenon of learning surely merits similar attention. The problem is to discover good models that are interesting to study for their own sake and that promise to be relevant both to explaining human experience and to building devices that can learn. The models should also shed light on the limits of what can be learned, just as computability does on what can be computed.

In this paper we shall say that a program for performing a task has been acquired by *learning* if it has been acquired by any means other than explicit programming. Among human skills some clearly appear to have a genetically preprogrammed element, whereas some others consist of executing an explicit sequence of instructions that has been memorized. There remains a large area of skill acquisition where no such explicit programming is identifiable. It is this area that we describe here as learning. The recognition of familiar objects, such as tables, provides such examples. These skills often have the additional property that, although we have learned them, we find it difficult to articulate what algorithm we are really using. In these cases it would be especially significant if machines could be made to acquire them by learning.

This paper is concerned with precise computational models of the learning phenomenon. We shall restrict ourselves to skills that consist of recognizing whether a *concept* (or predicate) is true or not for given data. We shall say that a concept $Q$ has been learned if a program for recognizing it has been deduced (i.e., by some method other than the acquisition from the outside of the explicit program).

The main contribution of this paper is that it shows that it is possible to design *learning machines* that have all three of the following properties:

1. The machines can provably learn whole classes of concepts. Furthermore, these classes can be characterized.
2. The classes of concepts are appropriate and nontrivial for general-purpose knowledge.
3. The computational process by which the machines deduce the desired programs requires a feasible (i.e., polynomial) number of steps.

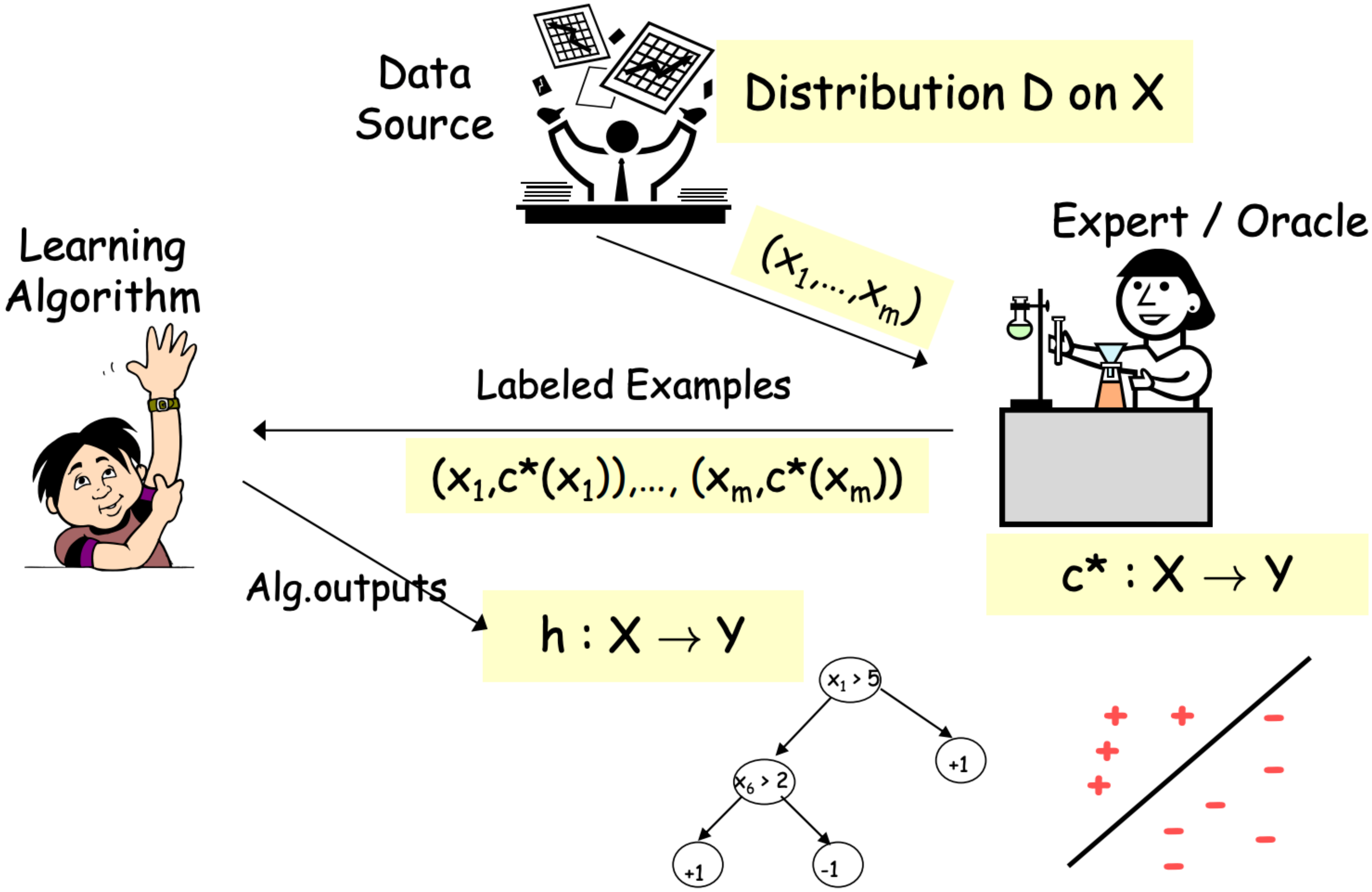
A learning machine consists of a *learning protocol* together with a *deduction procedure*. The former specifies the manner in which information is obtained from the outside. The latter is the mechanism by which a correct recognition algorithm for the concept to be learned is deduced. At the broadest level, the suggested *methodology* for studying learning is the following: Define a plausible learning protocol, and investigate the class of con-

This research was supported in part by National Science Foundation grant MCS-83-02385. A preliminary version of this paper appeared in the proceedings of the 16th ACM Symposium on Theory of Computing, Washington, D.C., 1984, 436–445.

A Theory of the Learnable, Communications of the ACM, 1984

# Probably Approximately Correct (PAC)

Probably Approximately Correct (PAC, Valiant, CACM 1984)



Data Source

Distribution D on X

Expert / Oracle

$(x_1,\dots,x_m)$

Learning Algorithm

Labeled Examples

$(x_1,c^*(x_1)),\dots,(x_m,c^*(x_m))$

$c^* : X \to Y$

Alg.outputs

$h : X \to Y$

[Figure credit: Nina Balcan]

# Probably Approximately Correct (PAC)

Probably Approximately Correct (PAC, Valiant, CACM 1984)

Definition (PAC-learnable): A concept class $\mathscr{F}$ is said to be PAC-learnable if there exists an algorithm $\mathscr{A}$ such that for any $0 < \epsilon, \delta < 1$, for any distribution $\mu$ over $\mathscr{X}$ and for any target concept $c \in \mathscr{F}$, the following holds for any sample size $n \geq \mathrm{poly}(1/\epsilon, 1/\delta, d)$:

$$\Pr_{\mathscr{D}}(\varepsilon_\mu(f) \leq \epsilon) \geq 1 - \delta$$

where $f$ is the output of the algorithm $\mathscr{A}$.

- $\epsilon$ is called the accuracy parameter
- $\delta$ is called the confidence parameter

# Probably Approximately Correct (PAC)

Probably Approximately Correct (PAC, Valiant, CACM 1984)

Definition (PAC-learnable): A concept class $\mathscr{F}$ is said to be PAC-learnable if there exists an algorithm $\mathscr{A}$ such that for any $0 < \epsilon, \delta < 1$, for any distribution $\mu$ over $\mathscr{X}$ and for any target concept $c \in \mathscr{F}$, the following holds for any sample size $n \geq \mathrm{poly}(1/\epsilon, 1/\delta, d)$:

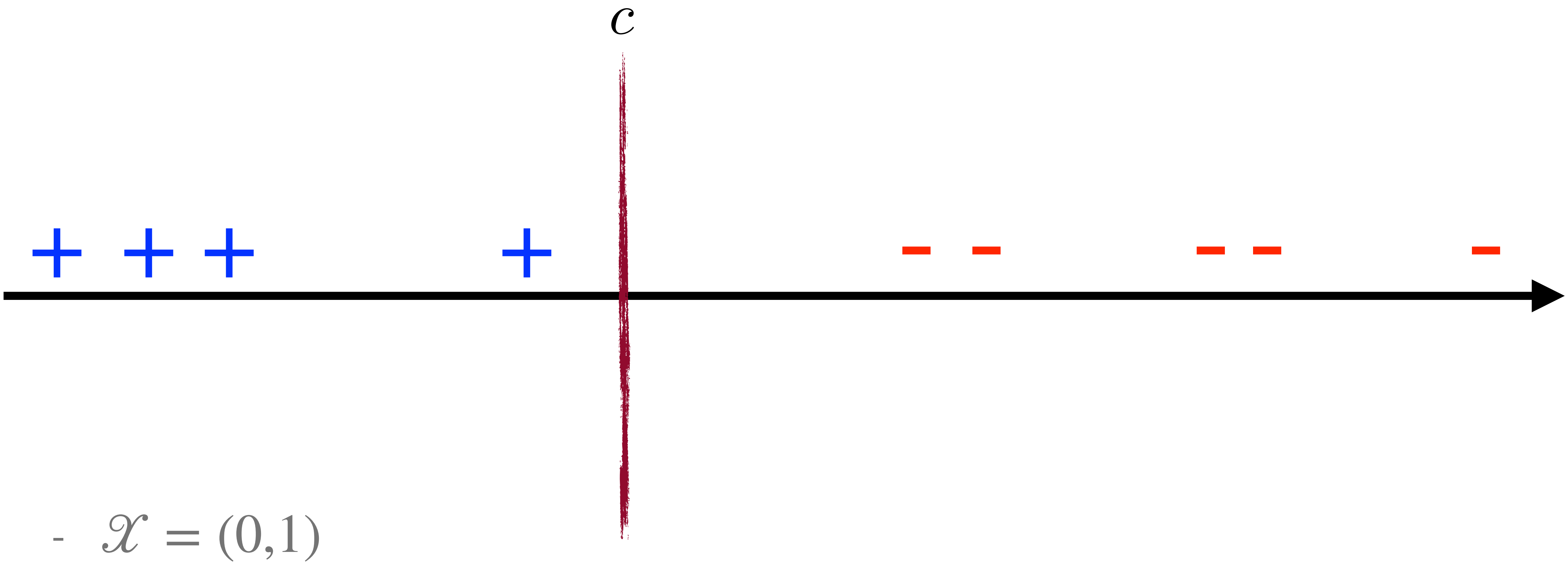
$$\Pr_{\mathscr{D}}(\varepsilon_\mu(f) \leq \epsilon) \geq 1 - \delta$$

where $f$ is the output of the algorithm $\mathscr{A}$.

- This holds for arbitrary target concept
- No assumption on the distribution $\mu$
- PAC-learnability does not mention about the time complexity of running $\mathscr{A}$
- The polynomial $\mathrm{poly}(1/\epsilon, 1/\delta, d)$ is called the sample complexity of $\mathscr{A}$

# Generalization Analysis

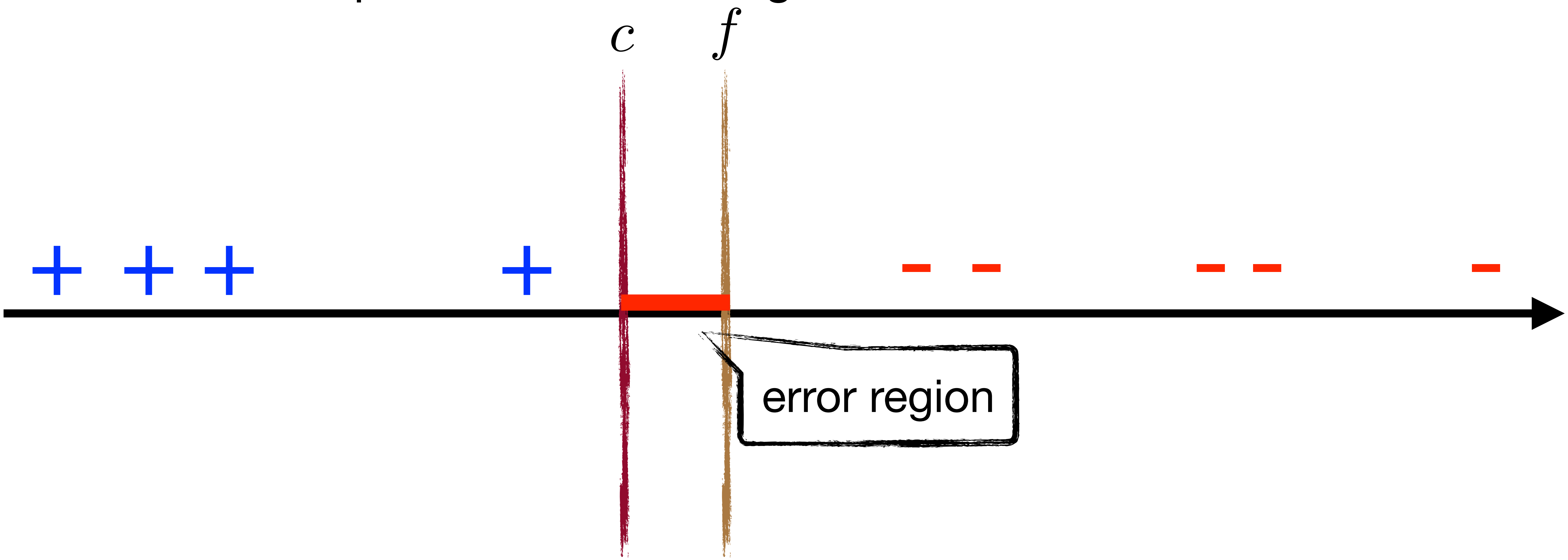
A running example: learning with initial-segment



- $\mathscr{X} = (0,1)$
- $\mathscr{F} = \{c_a \in 2^{(0,1)} \mid c_a(x) = 1 \iff x \leq a\}$
- Let's consider a simple algorithm: return $f = (\max_{x:x\in+} x + \min_{x:x\in-} x)/2$
- Assume the distribution over $\mathscr{X}$ to be uniform

# Generalization Analysis

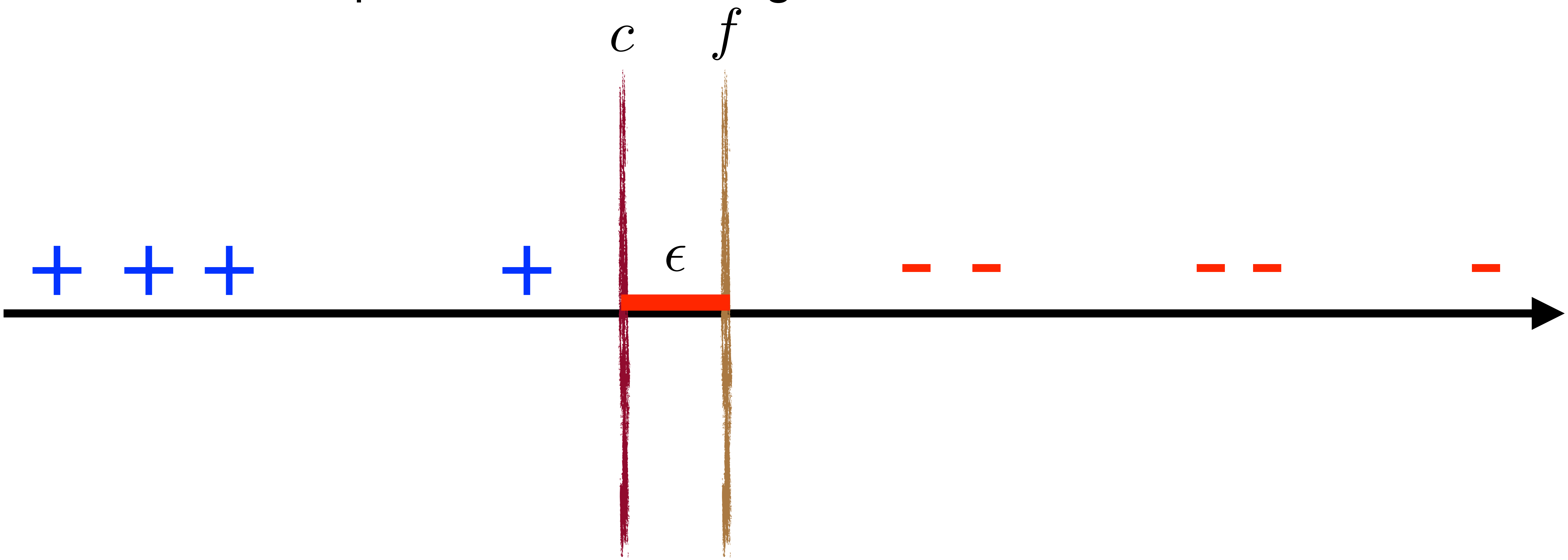
If the returned position is on the right of $c$:



- Error only happens at the interval between $c$ and $f$
- We want to upper bound the error probability: $\Pr(\varepsilon_\mu(f) \geq \epsilon)$

# Generalization Analysis

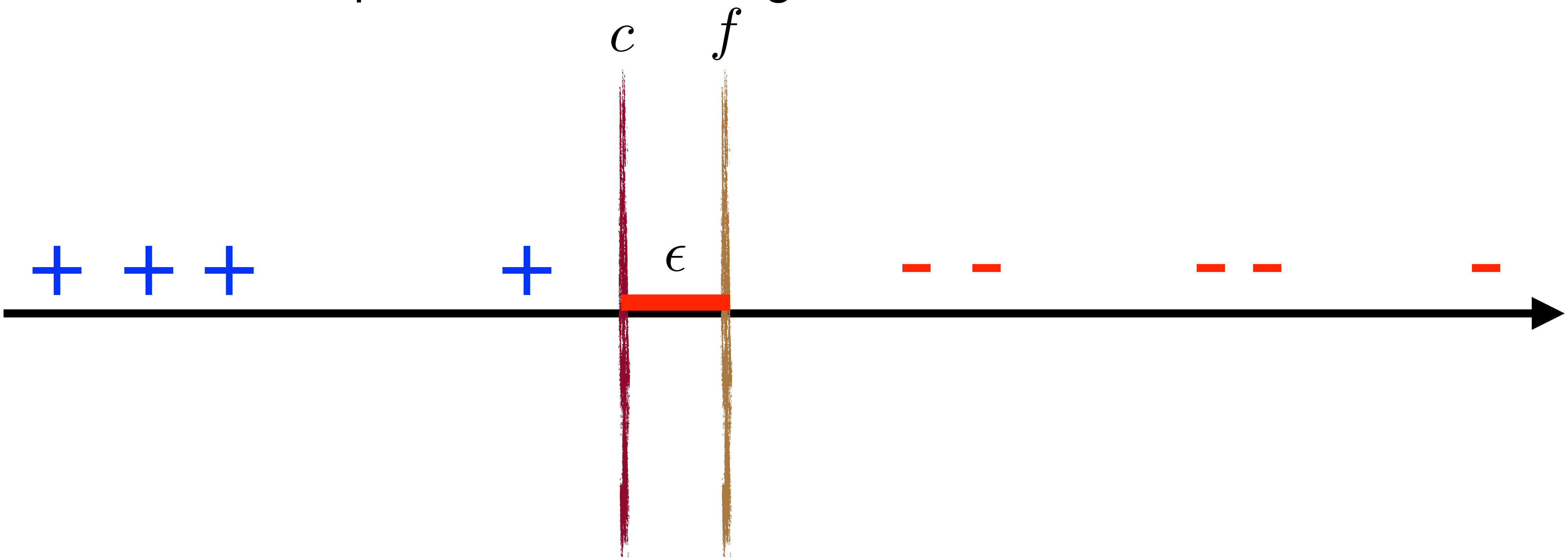
If the returned position is on the right of $c$:



- Claim: there is no point in the training data from $\mu$ that lies in this interval (?)

# Generalization Analysis

If the returned position is on the right of $c$:



$$\Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, f \text{ on the right of } c)$$
$$\leq \Pr(\text{none of the training data lies in the interval})$$
$$\leq (1-\epsilon)^n$$
$$\leq \exp(-n\epsilon)$$

iid assumption

$\forall x, 1 - x \leq \exp(-x)$

# Generalization Analysis

Similarly, if the returned position is on the left of $c$:

$$\Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, f \text{ on the left of } c) \leq \exp(-n\epsilon)$$

Now, by a union bound ($\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$), and let $L = f$ on the left of $c$ and $R = f$ on the right of $c$

$$\begin{aligned}\Pr(\varepsilon_\mu(f) \geq \epsilon) &= \Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, L)\Pr(L) + \Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, R)\Pr(R) \\ &\leq \Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, L) + \Pr(\varepsilon_\mu(f) \geq \epsilon \,|\, R) \\ &\leq 2\exp(-n\epsilon) \\ &\leq \delta\end{aligned}$$

Solving for $n$, we get: it suffices if

$$n \geq \frac{1}{\epsilon} \log \frac{1}{\delta}$$

This shows that $\mathscr{F}$ is PAC-learnable.

# Generalization Analysis

Could we generalize the previous results?

Realizable case with finite $\mathcal{F}$: $|\mathcal{F}| < \infty, c \in \mathcal{F}$

Theorem: Let $f$ be an empirical risk minimizer on a training data with $n$ examples where

$$n \geq \frac{1}{\epsilon}\left(\log|\mathcal{F}| + \log\frac{1}{\delta}\right)$$

Then $\Pr(\varepsilon_\mu(f) \leq \epsilon) \geq 1 - \delta$. Equivalently, with probability at least $1 - \delta$:

$$\varepsilon_\mu(f) \leq \frac{1}{n}\left(\log|\mathcal{F}| + \log\frac{1}{\delta}\right)$$

Empirical Risk Minimization (ERM):

$$f_{\mathrm{ERM}} = \mathcal{A}_{\mathrm{ERM}}(\mathcal{D}) := \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(f(x^{(i)}) \neq y^{(i)})$$

i.e., the ERM algorithm finds a predictor that minimizes the training loss

# Generalization Analysis

Realizable case with finite $\mathscr{F}$: $|\mathscr{F}| < \infty, c \in \mathscr{F}$

Theorem: Let $f$ be an empirical risk minimizer on a training data with $n$ examples where

$$n \geq \frac{1}{\epsilon}\left(\log|\mathscr{F}| + \log\frac{1}{\delta}\right)$$

Then $\Pr(\varepsilon_\mu(f) \leq \epsilon) \geq 1 - \delta$. Equivalently, with probability at least $1 - \delta$:

$$\varepsilon_\mu(f) \leq \frac{1}{n}\left(\log|\mathscr{F}| + \log\frac{1}{\delta}\right)$$

- Fact: since we are using ERM under realizable case, the training error of the ERM solution will be 0
- Let's fix a classifier $f$ and consider its true error. Instead, let $f_{\mathrm{ERM}}$ be the solution returned by the ERM algorithm

# Generalization Analysis

By definition of conditional probability:

$$\Pr\left(\hat{\varepsilon}_{\mathscr{D}}(f) = 0 \wedge \varepsilon_\mu(f) > \epsilon\right) \leq \Pr\left(\hat{\varepsilon}_{\mathscr{D}}(f) = 0 \mid \varepsilon_\mu(f) > \epsilon\right)$$

But,

$$\Pr\left(\hat{\varepsilon}_{\mathscr{D}}(f) = 0 \mid \varepsilon_\mu(f) > \epsilon\right) \leq (1-\epsilon)^n$$

Hence, by union bound,

$$\Pr\left(\exists f \in \mathscr{F} : \hat{\varepsilon}_{\mathscr{D}}(f) = 0 \wedge \varepsilon_\mu(f) > \epsilon\right) \leq |\mathscr{F}| \cdot (1-\epsilon)^n \leq \delta$$

On the other hand, we have

$$\begin{aligned} \Pr(\varepsilon_\mu(f_{\mathrm{ERM}}) > \epsilon) &= \Pr(\hat{\varepsilon}_{\mathscr{D}}(f_{\mathrm{ERM}}) = 0 \wedge \varepsilon_\mu(f_{\mathrm{ERM}}) > \epsilon) \\ &\leq \Pr\left(\exists f \in \mathscr{F} : \hat{\varepsilon}_{\mathscr{D}}(f) = 0 \wedge \varepsilon_\mu(f) > \epsilon\right) \\ &\leq |\mathscr{F}|(1-\epsilon)^n \leq |\mathscr{F}|\exp(-n\epsilon) \leq \delta \end{aligned}$$

Solving for $n$, we get

$$n \geq \frac{1}{\epsilon}\left(\log|\mathscr{F}| + \log\frac{1}{\delta}\right)$$

# Probably Approximately Correct (Agnostic)

So far we mainly talk about realizable case with finite hypothesis class.

Probably Approximately Correct (agnostic case)

Definition (PAC-learnable): A hypothesis space $\mathscr{H}$ is said to be agnostic PAC-learnable if there exists an algorithm $\mathscr{A}$ such that for any $0 < \epsilon, \delta < 1$, for all distribution $\mu$ over $\mathscr{X} \times \mathscr{Y}$, the following holds for any sample size $n \geq \text{poly}(1/\epsilon, 1/\delta, d)$:

$$\Pr\left(\varepsilon_\mu(f) \leq \min_{f' \in \mathscr{H}} \varepsilon_\mu(f') + \epsilon\right) \geq 1 - \delta$$

where $f$ is the output of the algorithm $\mathscr{A}$.

- $\epsilon$ is called the accuracy parameter
- $\delta$ is called the confidence parameter
- No assumption on $\mu$ has been made
- Agnostic PAC-learnability does not mention about the time complexity of running $\mathscr{A}$
- The polynomial $\text{poly}(1/\epsilon, 1/\delta, d)$ is called the sample complexity of $\mathscr{A}$

# Concentration Inequality

Some useful inequalities regarding the concentration of RVs

Theorem (Hoeffding's inequality): Let $Z_1, \ldots, Z_n$ be independent RVs where $Z_i \in [a, b]$. The for any $\epsilon > 0$, the following inequality hold for the mean $\bar{Z}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$ :

$$\Pr\left(\left|\bar{Z}_n - \mathbb{E}[\bar{Z}_n]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

Equivalent statement: with probability at least $1 - \delta$, we have:

$$\left|\bar{Z}_n - \mathbb{E}[\bar{Z}_n]\right| \leq (b-a)\sqrt{\frac{\log(2/\delta)}{2n}}$$

If $Z_1, \ldots, Z_n$ are iid, then $\mathbb{E}[\bar{Z}_n] = \mathbb{E}[Z_i], \forall i \in [n]$ so we have

$$\left|\bar{Z}_n - \mathbb{E}[Z_1]\right| \leq (b-a)\sqrt{\frac{\log(2/\delta)}{2n}}$$

# Concentration Inequality

Some useful inequalities regarding the concentration of RVs

Theorem (Hoeffding's inequality): Let $Z_1, \ldots, Z_n$ be independent RVs where $Z_i \in [a, b]$. The for any $\epsilon > 0$, the following inequality hold for the mean $\bar{Z}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i$ :

$$\Pr\left(\left|\bar{Z}_n - \mathbb{E}[\bar{Z}_n]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)$$

Example: Coin flipping

- Suppose we have a coin with head probability $p$
- We flipped the coin for 1000 times, with an average head frequency $\hat{p}$
- How close will the frequency $\hat{p}$ be to the true $p$?

# Generalization Analysis

For any fixed $f \in \mathscr{F}$, we can use the Hoeffding's inequality to get a generalization bound:

Let $Z_i = \mathbb{I}(f(X^{(i)}) \neq Y^{(i)}) \in \{0,1\} \subseteq [0,1]$, then

Training error: $\bar{Z}_n = \frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(f(X^{(i)}) \neq Y^{(i)}) = \hat{\varepsilon}_{\mathscr{D}}(f)$

Test error: $\mathbb{E}\left[\bar{Z}_n\right] = \mathbb{E}\left[Z_1\right] = \varepsilon_\mu(f)$

By Hoeffding's inequality, with probability at least $1-\delta$, we have

$$\varepsilon_\mu(f) \leq \hat{\varepsilon}_{\mathscr{D}}(f) + \sqrt{\frac{\log(2/\delta)}{2n}}$$

Note: it is important to fix a predictor $f$ in order for the analysis above to hold, i.e., $f$ cannot be the output of an algorithm $\mathscr{A}$ that depends on the data $\mathscr{D}$

# Generalization Analysis

## What if $f = \mathscr{A}(\mathscr{D})$?

Given $\mathscr{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n} \sim \mu$ be a dataset of iid samples. Define our algorithm as follows:

$$f(x) := \begin{cases} y_i & \text{if } x = x_i \\ \text{"unknown"} & \text{otherwise} \end{cases}$$

Then $\hat{\varepsilon}_{\mathscr{D}}(f) = 0$ and $\varepsilon_{\mu}(f) = 1$!

Why?

- Hoeffding's inequality cannot be applied anymore, since $f$ is the outcome of an algorithm $\mathscr{A}$ that depends on the data $\mathscr{D}$. In other words, given $f$, the data $Z_i$ are no longer independent

Fix?

- Use a disjoint validation set to empirically estimate the error
- Pay a model complexity penalty term: with probability at least $1 - \delta$, for all $f \in \mathscr{F}$ simultaneously, we have:

$$\varepsilon_{\mu}(f) \leq \hat{\varepsilon}_{\mathscr{D}}(f) + O\left(\sqrt{\frac{\text{complexity}(\mathscr{F}) + \log(1/\delta)}{n}}\right)$$

# Vapnik–Chervonenkis dimension (VC-dim)

## VC dimension:

Let $\mathscr{F} : \mathbb{R}^d \rightarrow \{0,1\}$ be a set of binary functions. Then the VC dimension of $\mathscr{F}$, denoted by $\mathrm{VCdim}(\mathscr{F})$ is the cardinality of the largest set of points in $\mathbb{R}^d$ that can be shattered by $\mathscr{F}$.

Shattering:

Given a set $\mathscr{D} \subseteq \mathbb{R}^d$ of size $n$, i.e., $|\mathscr{D}| = n$, we say that $\mathscr{D}$ can be shattered by $\mathscr{F}$ iff

$$\forall S \subseteq \mathscr{D}, \exists f \in \mathscr{F} : \forall x \in S, f(x) = 1, \forall x \notin S, f(x) = 0$$

Note:

- By definition, in order to claim the VC-dim of a given hypothesis class $\mathscr{F}$ to be $n$, we need to verify the following two conditions:

  ★ $\mathrm{VCdim}(\mathscr{F}) \geq n$: $\exists \mathscr{D} \subseteq \mathbb{R}^d : |\mathscr{D}| = n, \mathscr{F}$ shatters $\mathscr{D}$

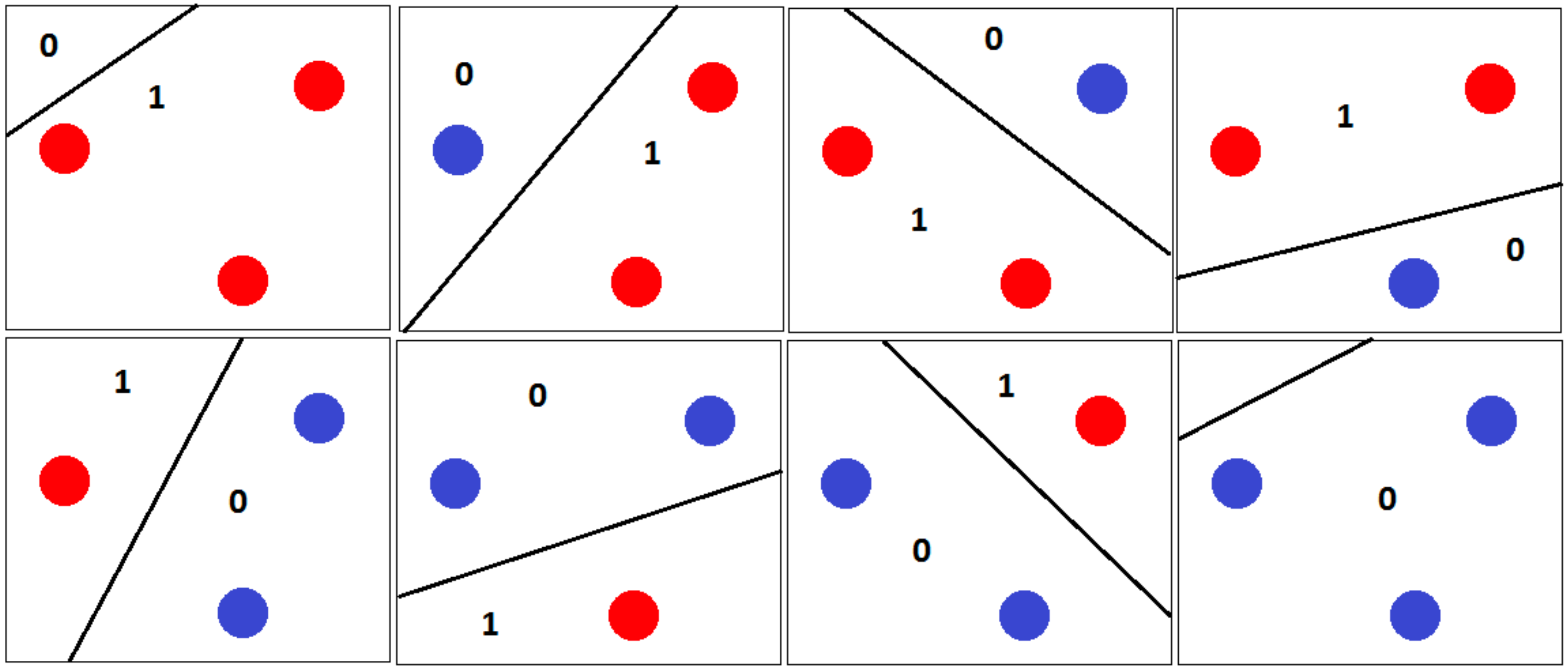
  ★ $\mathrm{VCdim}(\mathscr{F}) \leq n$: $\forall \mathscr{D} \subseteq \mathbb{R}^d : |\mathscr{D}| = n + 1, \mathscr{D}$ cannot be shattered by $\mathscr{F}$

# Vapnik–Chervonenkis dimension (VC-dim)

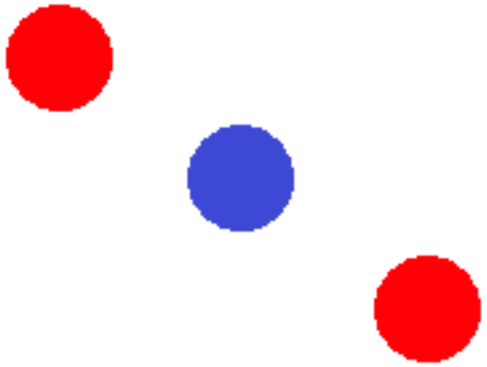
Example: $d = 2, \mathscr{F} = \{\text{linear classifiers in } \mathbb{R}^2\}$

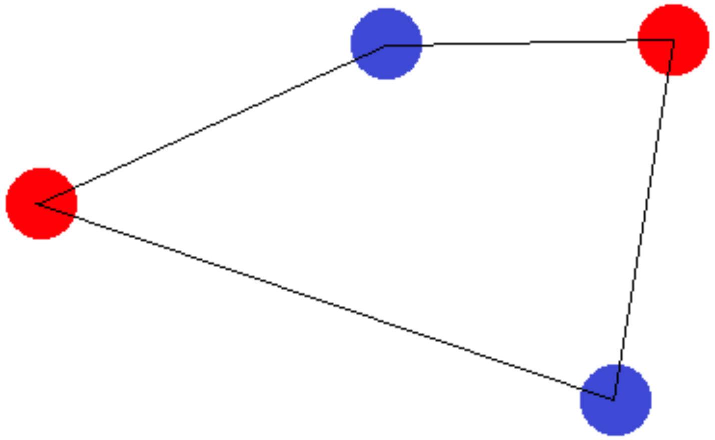
Claim: $\mathrm{VCdim}(\mathscr{F}) = 3$

Proof that $\mathrm{VCdim}(\mathscr{F}) \geq 3$:



Proof that $\mathrm{VCdim}(\mathscr{F}) < 4$: XOR





# Generalization Analysis

With VC dim as the complexity measure, we have the following uniform generalization bound:

Given $\mathscr{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n \sim \mu$ be a dataset of iid samples. Let $\mathscr{F}$ be a hypothesis class of finite VC-dim, i.e., $\mathrm{VCdim}(\mathscr{F}) < \infty$, then for $0 < \delta < 1$, with probability at least $1 - \delta$, for all $f \in \mathscr{F}$:

$$\varepsilon_\mu(f) \leq \hat{\varepsilon}_{\mathscr{D}}(f) + O\left(\sqrt{\frac{\mathrm{VCdim}(\mathscr{F}) + \log(1/\delta)}{n}}\right)$$

Note:

- As long as $\mathrm{VCdim}(\mathscr{F}) < \infty$, as $n \to \infty$, we know that the training error converges to the test error
- The bound above gives the generalization error, and we can use the generalization error bound to provide an upper bound on the estimation error, i.e., $\varepsilon_\mu(f) - \inf_{f' \in \mathscr{F}} \varepsilon_\mu(f')$
- There are other forms of complexity measures to characterize the expressiveness/richness/powerfulness of a given hypothesis class, but it is beyond the scope of this course
- The bound above could be loose, i.e., the generalization error could be larger than 1 for classification problems

# Next Time

- Perceptron Algorithm
- Deep Learning