Data Mining
PAC learning and the VC-dimension

Ad Feelders

Universiteit Utrecht

Introduction

Questions we are going to address today in the context of binary classification:

- How much training data is enough to guarantee a specified accuracy with a specified confidence?
- What property of a hypothesis class controls this requirement? Is it
  - the number of parameters?
  - the number of hypotheses?
  - or something else?
- Are there hypothesis classes for which no amount of data gives a distribution-free guarantee?

Overview

1. Finite hypothesis classes.
   - Realizable case.
   - Agnostic case.
2. Infinite hypothesis classes.

Realizable Case

Consider the following (simplified) learning setup:

1. Feature vectors x are drawn i.i.d. from an arbitrary unknown distribution $P(x)$.
2. The feature vectors are labeled by an unknown target function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{-1, +1\}$.
3. The learning algorithm processes the training data $D = \{(x_i, y_i)\}_{i=1}^{m}$ and returns a hypothesis h_D from a finite hypothesis set $\mathcal{H}$.
4. The hypothesis set $\mathcal{H}$ contains the target function f.
5. The learning algorithm returns a hypothesis that has zero error on the training data:

$$E_D(h_D) = \frac{1}{m} \sum_{i=1}^{m} I(h_D(x_i) \neq y_i) = 0.$$

This is called a *consistent* learner.

The Question

How many training examples are sufficient to guarantee that with high probability, the hypothesis returned has small true error, where true error is defined as

$$E_P(h) = \Pr_{x \sim P(x)} \left[h(x) \neq f(x) \right]$$

What sample size m is sufficient to guarantee that h_D is probably approximately correct (PAC)?

$$\Pr_{D \sim P(x)^m} \left[E_P(h_D) \leq \epsilon \right] \geq 1 - \delta,$$

where $\delta, \epsilon \in (0, 1)$. ϵ controls the accuracy, δ the confidence.

The Answer

What is the probability that a hypothesis with true error $> \epsilon$ ("bad" hypothesis h_b) has zero error on the training data?

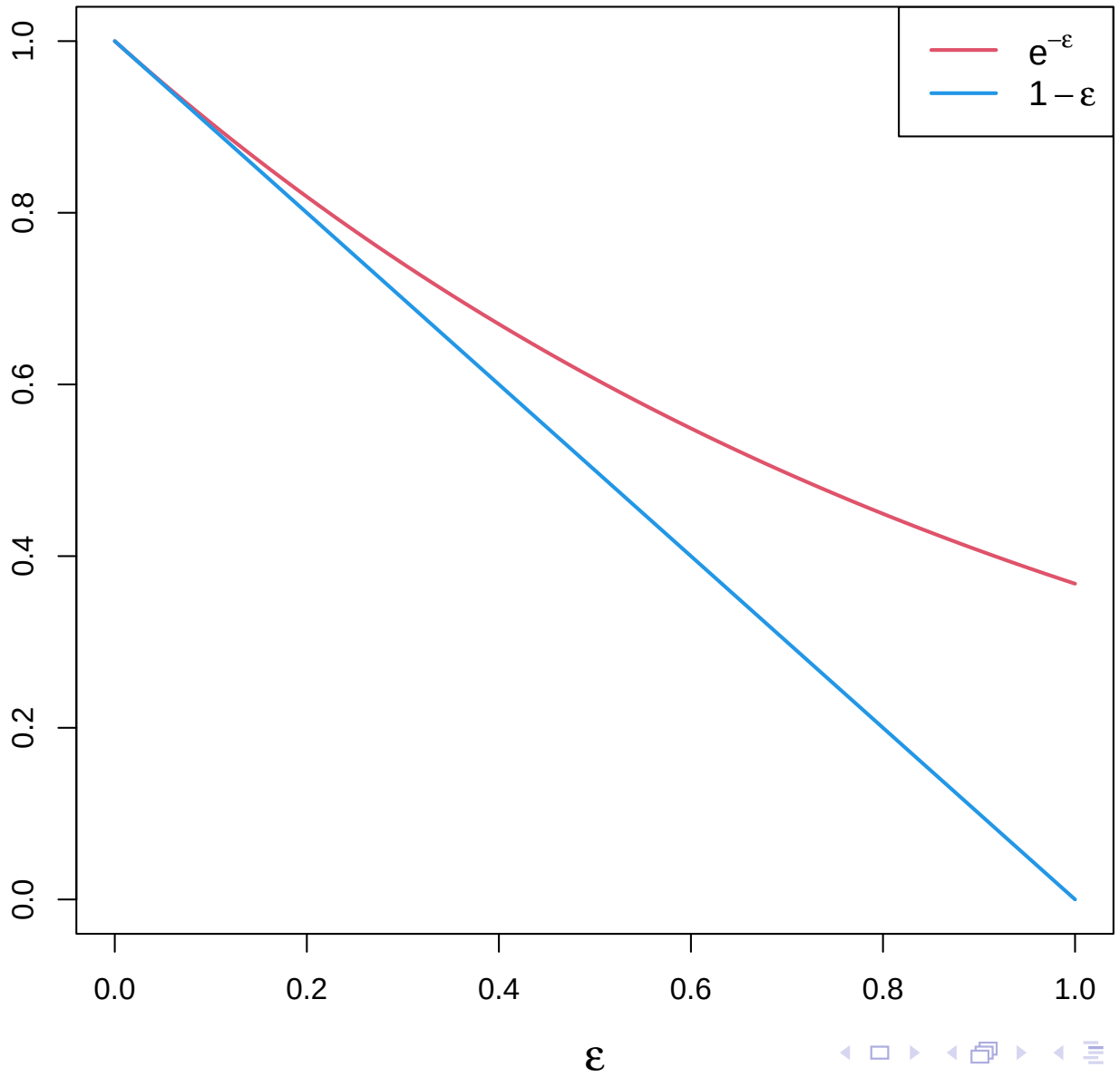
For any individual example (x_i, y_i), we have

$$\Pr[h_b(x_i) = y_i] \leq 1 - \epsilon$$

The probability that such a bad hypothesis classifies all training examples correctly is therefore bounded by:

$$\Pr[h_b(x_1) = y_1 \wedge \ldots \wedge h_b(x_m) = y_m] \leq (1 - \epsilon)^m \leq e^{-\epsilon m}$$

For ϵ between 0 and 1: $(1 - \epsilon) \leq e^{-\epsilon}$



The Answer (continued)

The learning algorithm is allowed to pick *any* hypothesis from $\mathcal{H}$ with zero training error. The probability that it picks a bad hypothesis is bounded by (union bound):

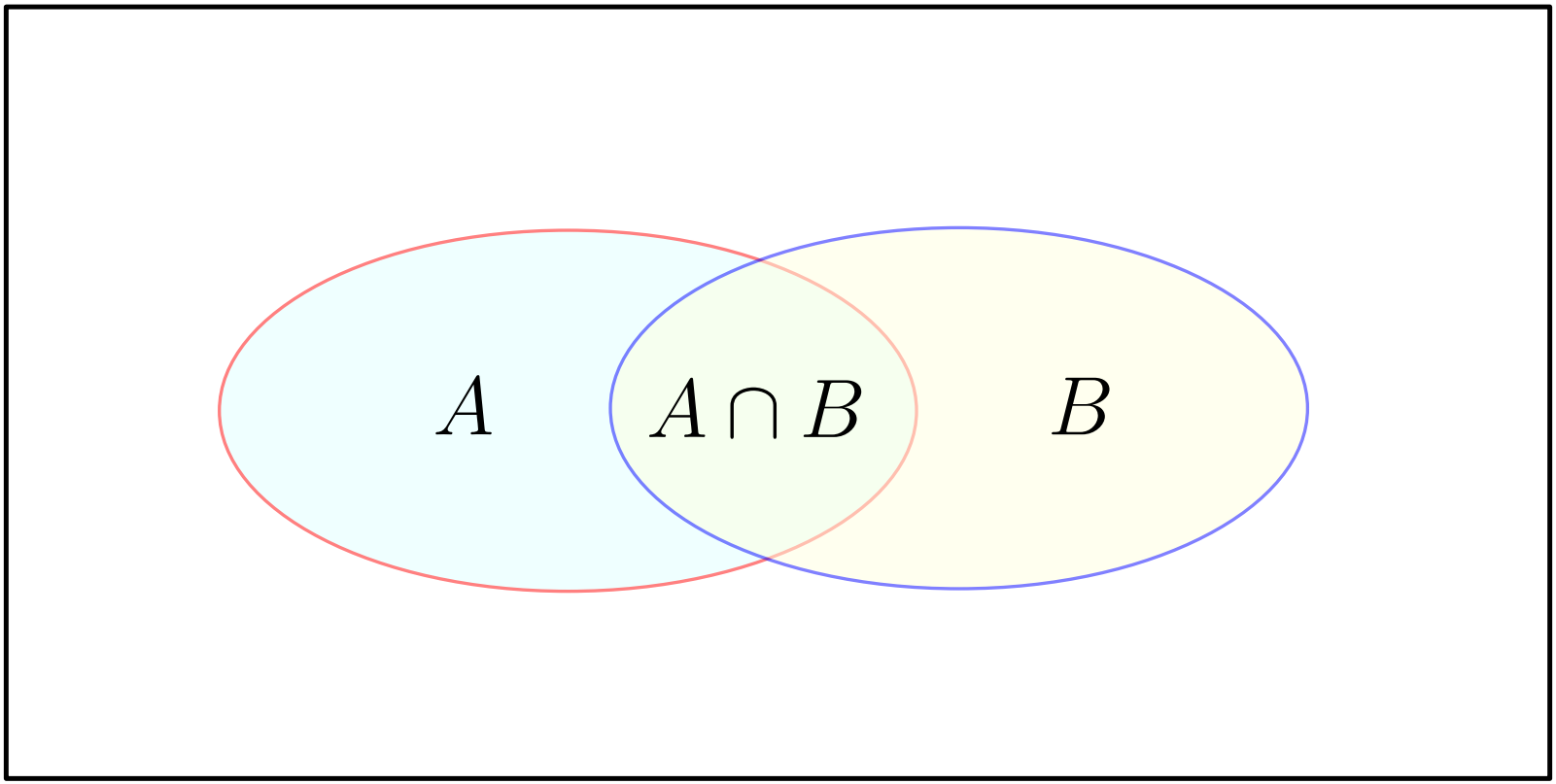
$$\Pr_{D \sim P(X)^m}[\exists h \in \mathcal{H} : E_D(h) = 0 \wedge E_P(h) > \epsilon] \leq |\mathcal{H}| e^{-\epsilon m}$$

We want

$$\Pr_{D \sim P(X)^m}[\exists h \in \mathcal{H} : E_D(h) = 0 \wedge E_P(h) > \epsilon] \leq \delta$$

For $\delta \geq |\mathcal{H}| e^{-\epsilon m}$ this is true.

Union Bound



$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \leq \Pr(A) + \Pr(B).$$

Sufficient sample size

So how many training examples will suffice?

$$\delta \geq |\mathcal{H}| e^{-\epsilon m} \qquad \text{(Divide by } |\mathcal{H}|\text{)}$$

$$\frac{\delta}{|\mathcal{H}|} \geq e^{-\epsilon m} \qquad \text{(Take natural log)}$$

$$\ln\left(\frac{\delta}{|\mathcal{H}|}\right) \geq -\epsilon m \qquad \left(\ln \tfrac{a}{b} = \ln a - \ln b\right)$$

$$\ln \delta - \ln |\mathcal{H}| \geq -\epsilon m \qquad \text{(Multiply by } -1\text{)}$$

$$\ln |\mathcal{H}| - \ln \delta \leq \epsilon m \qquad \text{(Divide by } \epsilon\text{)}$$

$$m \geq \frac{1}{\epsilon}\left(\ln |\mathcal{H}| - \ln \delta\right) \qquad \left(-\ln a = \ln \tfrac{1}{a}\right)$$

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

Sufficient sample size for realizable case

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

The bound grows:

- Linearly in $1/\epsilon$: halving the tolerated error doubles the bound.
- Logarithmically in $1/\delta$: requiring greater confidence has a relatively mild cost.
- Logarithmically in $|\mathcal{H}|$: even a very large finite class can have a modest sample requirement.

Example: Threshold Functions

Consider the class of threshold functions $\mathcal{H} = \left\{ \frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10} \right\}$, and let x be a real number in the interval $[0, 1]$. For example, one of the members of $\mathcal{H}$ is the function:

$$h(x) = \begin{cases} +1 & \text{if } x \geq \frac{1}{10} \\ -1 & \text{otherwise} \end{cases}$$

(a) Give an algorithm that returns a hypothesis with zero training error.

(b) How many examples suffice for $\epsilon = 0.1$ and $\delta = 0.05$?

(c) What if $\epsilon = 0.01$?

(d) What if $\delta = 0.01$?

Example: Conjunctions of Binary Conditions

Let $x_1, \ldots, x_n$ be binary variables. The hypothesis class $\mathcal{H}_n$ consists of conjunctions of upto n conditions.

For example, for $n = 4$, $(x_1 = 1 \wedge x_2 = 0 \wedge x_4 = 1) \in \mathcal{H}_4$

- $(1, 0, 0, 1)$ is a positive example
- $(0, 0, 0, 1)$ is a negative example

Sample complexity of $\mathcal{H}_n$

Clearly, $\mathcal{H}_n$ is finite. In fact

$$|\mathcal{H}_n| = 3^n$$

(either $x_i = 1$ is in the expression or $x_i = 0$ is in, or neither is in).

Recall that we require a sample size

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

Filling in $|\mathcal{H}| = 3^n$ gives:

$$m \geq \frac{\ln (3^n) + \ln(1/\delta)}{\epsilon} = \frac{n \ln (3) + \ln(1/\delta)}{\epsilon}$$

For example, for $\delta = 0.05$, $\epsilon = 0.1$ and $n = 10$, we need at least 140 examples to guarantee (at least) 90% accuracy with probability (at least) 95%.

Learning $\mathcal{H}_n$

Let $x = (x_1, \ldots, x_n)$ be an example to learn an expression in $\mathcal{H}_n$. If

- x is a positive example and $x_i = 1$, then $x_i = 0$ is ruled out
- x is a positive example and $x_i = 0$, then $x_i = 1$ is ruled out
- if x is a negative example, we change nothing (the current hypothesis already classifies it correctly)

A simple algorithm is, thus,

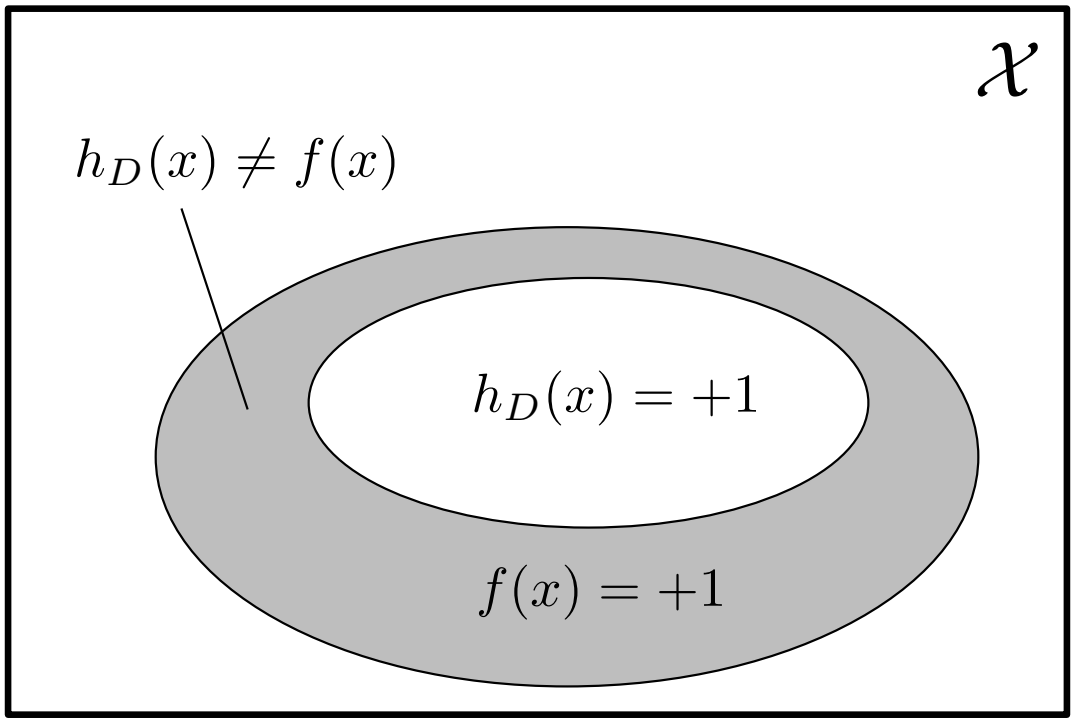
- start with the conjunction of all possible conditions
- with each positive example delete the conditions per above
- when all positive examples have been processed return the conjunction of all remaining conditions
- if all examples were negative, you have learned nothing

Properties of Algorithm

1. The conditions of f are a subset of the conditions of h_D.
2. Therefore, the part of the instance space that is labeled positive by h_D is a subset of the part that is labeled positive by f.
3. The algorithm is a consistent learner, that is, $E_D(h_D) = 0$.

Hypothesis returned

The relation between h_D and f for the given algorithm is depicted schematically below.



The true loss $E_P(h_D)$ is equal to the probability that when we draw a point at random from $P(X)$, we will hit the grey region.

Example

Consider three binary features:

- S: sunny
- W: warm
- V: windy

We want to learn when someone chooses to go for a walk. The unknown target is

$$f(S, W, V) = (S = 1) \wedge (V = 0) \qquad \text{(sunny and not windy)}.$$

Initially,

$$h = (S = 1 \wedge S = 0) \wedge (W = 1 \wedge W = 0) \wedge (V = 1 \wedge V = 0).$$

This contradictory conjunction is just the initialization: all conditions are still candidates.

Example

Step	Example (S, W, V)	Walk?	Conditions deleted	Conjunction
1	$(1, 1, 0)$	Yes	$S = 0, W = 0, V = 1$	$S = 1 \wedge W = 1 \wedge V = 0$
2	$(0, 1, 0)$	No	None	$S = 1 \wedge W = 1 \wedge V = 0$
3	$(1, 0, 0)$	Yes	$W = 1$	$S = 1 \wedge V = 0$
4	$(0, 1, 1)$	No	None	$S = 1 \wedge V = 0$

The returned hypothesis is

$$h_D = (S = 1) \wedge (V = 0)$$

Are there other consistent hypotheses?

Overview

1. Finite hypothesis classes.
   - Realizable case.
   - Agnostic case.
2. Infinite hypothesis classes.

Agnostic Learning

The assumption that the hypothesis set contains a hypothesis with zero true error is rather unrealistic, because:

- The target function f may not be contained in $\mathcal{H}$.
- The labels may be noisy: we don't have $y = f(x)$, but $y \sim P(y \mid x)$.

How does this change the analysis?

Empirical Risk Minimization (ERM)

An empirical risk minimization (ERM) algorithm chooses a hypothesis in $\mathcal{H}$ that makes the fewest mistakes on the training sample.
Given $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, define the empirical error

$$E_D(h) = \frac{1}{m} \sum_{i=1}^{m} I\{h(x_i) \neq y_i\}.$$

An ERM algorithm returns

$$h_D \in \arg\min_{h \in \mathcal{H}} E_D(h).$$

If several hypotheses achieve the minimum, it may choose any of them.

- Realizable case: the minimum training error is zero, so ERM returns a consistent hypothesis.
- Agnostic case: the minimum may be positive; ERM still returns a hypothesis with the smallest training error.

ϵ-Representative Sample

A sample D is ϵ-representative with respect to hypothesis class $\mathcal{H}$ and distribution P if

$$\forall h \in \mathcal{H} : |E_D(h) - E_P(h)| \leq \epsilon$$

In words: the empirical error approximates the true error within ϵ, simultaneously for every hypothesis in the class.

Hoeffding's Inequality

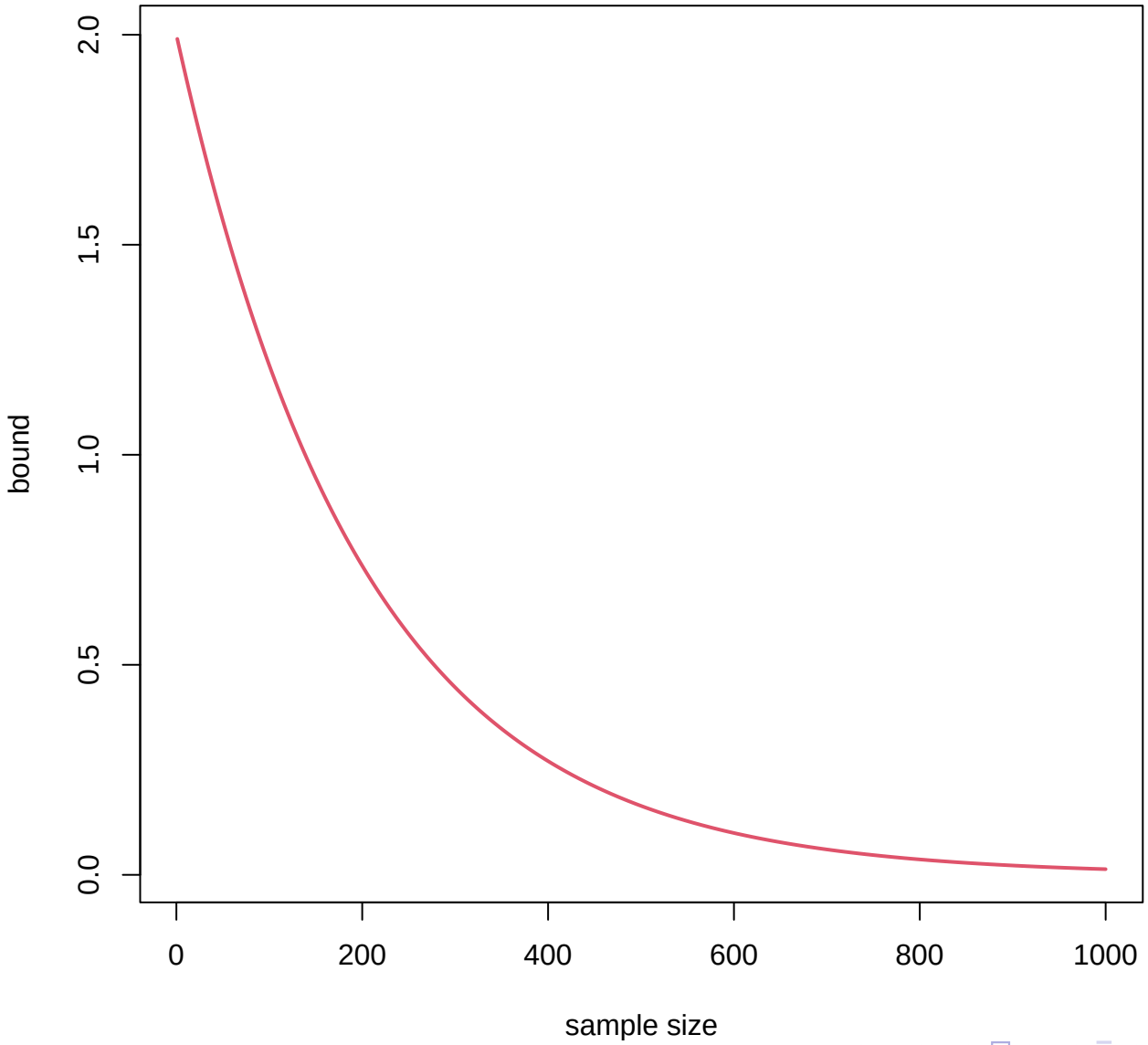
For a fixed hypothesis h and i.i.d. sample $D \sim P(X, Y)^m$, Hoeffding's inequality states

$$\Pr_{D \sim P(X,Y)^m} \left[\left| E_D(h) - E_P(h) \right| > \epsilon \right] \leq 2e^{-2m\epsilon^2}.$$

For a fixed hypothesis, the probability that its empirical error differs from its true error by more than ϵ is at most $2e^{-2m\epsilon^2}$.

Larger samples make large deviations between empirical and true error exponentially less likely.

Hoeffding's Inequality ($\epsilon = 0.05$)



Agnostic Learning

If we require

$$\Pr_{D\sim P(X,Y)^m}\left[E_P(h_D) > \min_{h\in\mathcal{H}}\{E_P(h)\} + \epsilon\right] \leq \delta,$$

where h_D is any hypothesis output by an ERM learner, then it suffices to obtain a sample that is $\frac{\epsilon}{2}$ representative with probability at least $1-\delta$. That is, we need

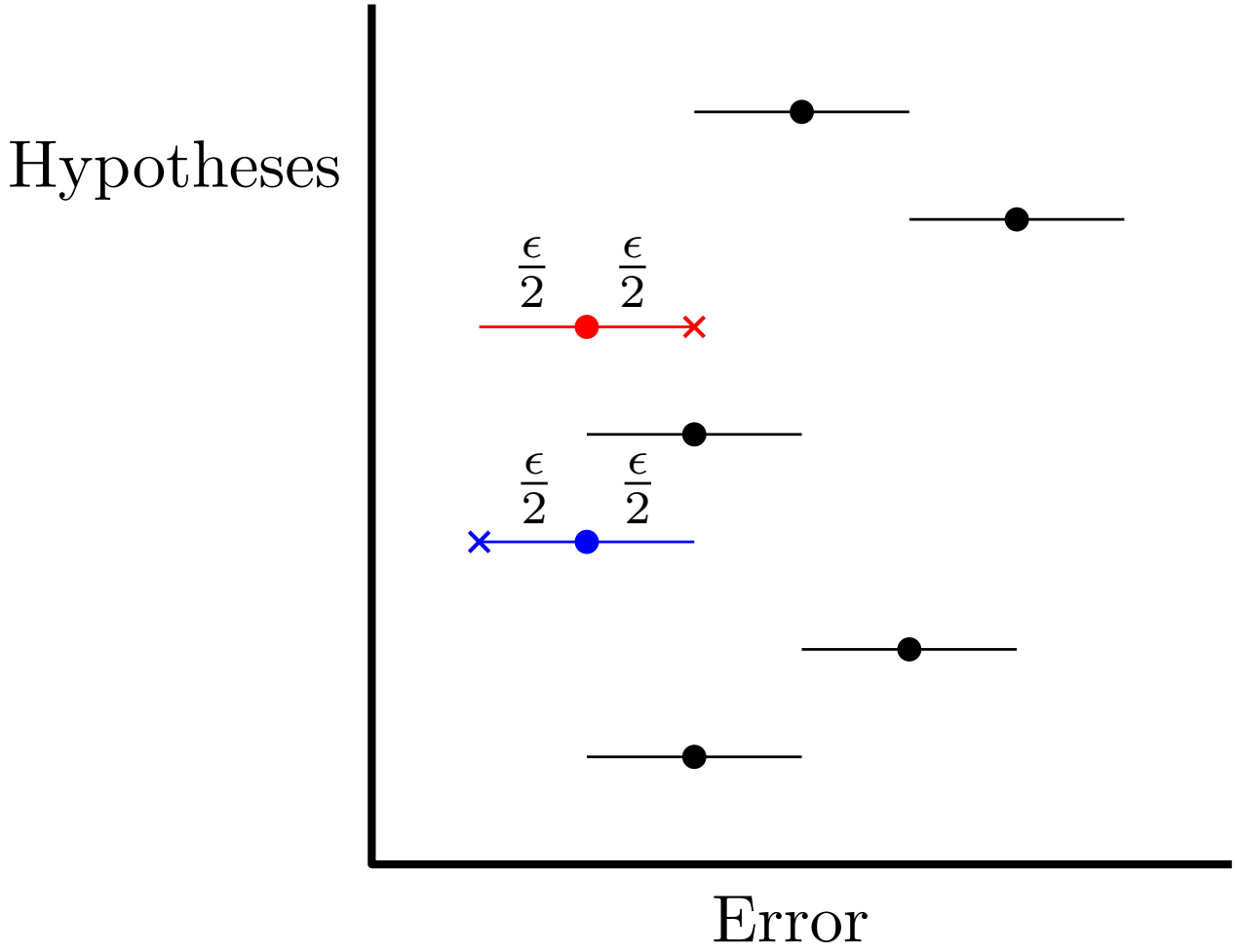
$$\Pr_{D\sim P(X,Y)^m}\left[\exists h \in \mathcal{H} : \left|E_P(h) - E_D(h)\right| > \frac{\epsilon}{2}\right] \leq \delta.$$

By the union bound and Hoeffding's inequality we have that

$$\Pr_{D\sim P(X,Y)^m}\left[\exists h \in \mathcal{H} : \left|E_P(h) - E_D(h)\right| > \frac{\epsilon}{2}\right] \leq 2|\mathcal{H}|e^{-\frac{1}{2}m\epsilon^2}.$$

Hence, for $\delta \geq 2|\mathcal{H}|e^{-\frac{1}{2}m\epsilon^2}$ we're good.

$\frac{\epsilon}{2}$ Representative Sample



$\frac{\epsilon}{2}$ Representative Sample

The dots indicate the training error $E_D(h)$ of different hypotheses h. Since the sample is $\frac{\epsilon}{2}$ representative, the true error is within $\frac{\epsilon}{2}$ of the training error. This interval is indicated by the horizontal bar. Our ERM-algorithm will return a hypothesis with minimum training error. In the picture, both the red and the blue hypothesis achieve the minimum training error, so an ERM algorithm might return either one of them. Let's say our ERM-algorithm returns the red one. Worst thing that can happen is that its true error (the red cross) is $\frac{\epsilon}{2}$ higher than its training error, whereas for the hypothesis we did not select, the true error (the blue cross) is $\frac{\epsilon}{2}$ smaller than its training error. But even then the true error of the selected hypothesis is still within ϵ of the best true error.

Agnostic Learning

Starting with $\delta \geq 2|\mathcal{H}|e^{-\frac{1}{2}m\epsilon^2}$ and solving for m we find:

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

In the realizable case we got:

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

The most important difference is that in the agnostic case we have the factor $\frac{2}{\epsilon^2}$, instead of $\frac{1}{\epsilon}$ in the realizable case. Since ϵ is between 0 and 1, and typically close to 0, we have that ϵ^2 is smaller than ϵ, and hence $\frac{1}{\epsilon^2}$ is bigger than $\frac{1}{\epsilon}$. Bottom line is we need more data in the agnostic case.

Example: Threshold Functions

Consider the class of threshold functions $\mathcal{H} = \left\{ \frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10} \right\}$, and let x be a real number in the interval $[0, 1]$. For example, one of the members of $\mathcal{H}$ is the function:

$$h(x) = \begin{cases} +1 & \text{if } x \geq \frac{1}{10} \\ -1 & \text{otherwise} \end{cases}$$

(a) Give an algorithm that returns a hypothesis with minimum training error.

(b) How many examples suffice for $\epsilon = 0.1$ and $\delta = 0.05$?

(c) What if $\epsilon = 0.01$?

(d) What if $\delta = 0.01$?

Overview

1. Finite hypothesis classes.
   - Realizable case.
   - Agnostic case.
2. Infinite hypothesis classes.

Growth Function

- An assignment of class labels to a set of data points is called a *dichotomy*.
- For a given set of data points, many different members of $\mathcal{H}$ will produce the same dichotomy.
- The growth function $g_{\mathcal{H}}(m)$ of $\mathcal{H}$ is the maximum number of dichotomies that $\mathcal{H}$ can produce on a set of m points.
- You get to choose the m data points so as to maximize the number of dichotomies!

So $g_{\mathcal{H}}(m)$ is the maximum over all data sets of size m of the number of dichotomies that $\mathcal{H}$ can produce on this set.

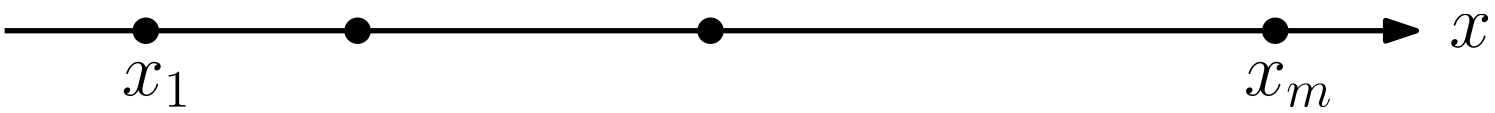
Growth Function: Threshold Functions

Consider the class $\mathcal{H}_t$ of threshold functions:

$$h(x) = \begin{cases} +1 & \text{if } x \geq t \\ -1 & \text{if } x < t, \end{cases}$$

where $x, t \in \mathbb{R}$. Notice that this is an infinite hypothesis class, since t can be any real number.

(a) Consider a set of m distinct data points $x_1, x_2, \ldots, x_m$. How many dichotomies, as a function of m, can $\mathcal{H}_t$ produce on such a set?



(b) Compute the value of the growth function of $\mathcal{H}_t$ for $m = 1, 2, 3$.

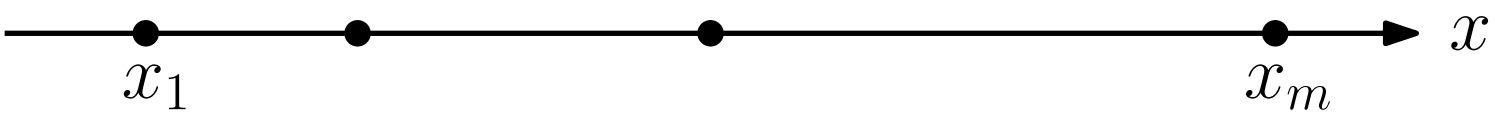
Growth Function: Intervals

Consider the class $\mathcal{H}_{(t_1,t_2)}$ of intervals:

$$h(x) = \begin{cases} +1 & \text{if } t_1 \leq x \leq t_2 \\ -1 & \text{otherwise} , \end{cases}$$

where $x, t_1, t_2 \in \mathbb{R}$.

(a) Consider a set of m distinct data points $x_1, x_2, \ldots, x_m$. How many dichotomies can $\mathcal{H}_{(t_1,t_2)}$ produce on such a set?



(b) Compute the value of the growth function of $\mathcal{H}_{(t_1,t_2)}$ for $m = 1, 2, 3$.

VC-dimension

- The VC-dimension of a hypothesis class $\mathcal{H}$ is the size of the largest data set for which $\mathcal{H}$ can produce all 2^m possible dichotomies.
- When $\mathcal{H}$ can produce all possible 2^m dichotomies on a set of m data points, we say that $\mathcal{H}$ *shatters* this set of data points.
- If there is *any* set of m data points that $\mathcal{H}$ can shatter, then the VC-dimension of $\mathcal{H}$ is at least m.

(a) What is the VC-dimension of the class of threshold functions $\mathcal{H}_t$?

(b) What is the VC-dimension of the class of intervals $\mathcal{H}_{(t_1,t_2)}$?

VC-dimension of linear classifiers

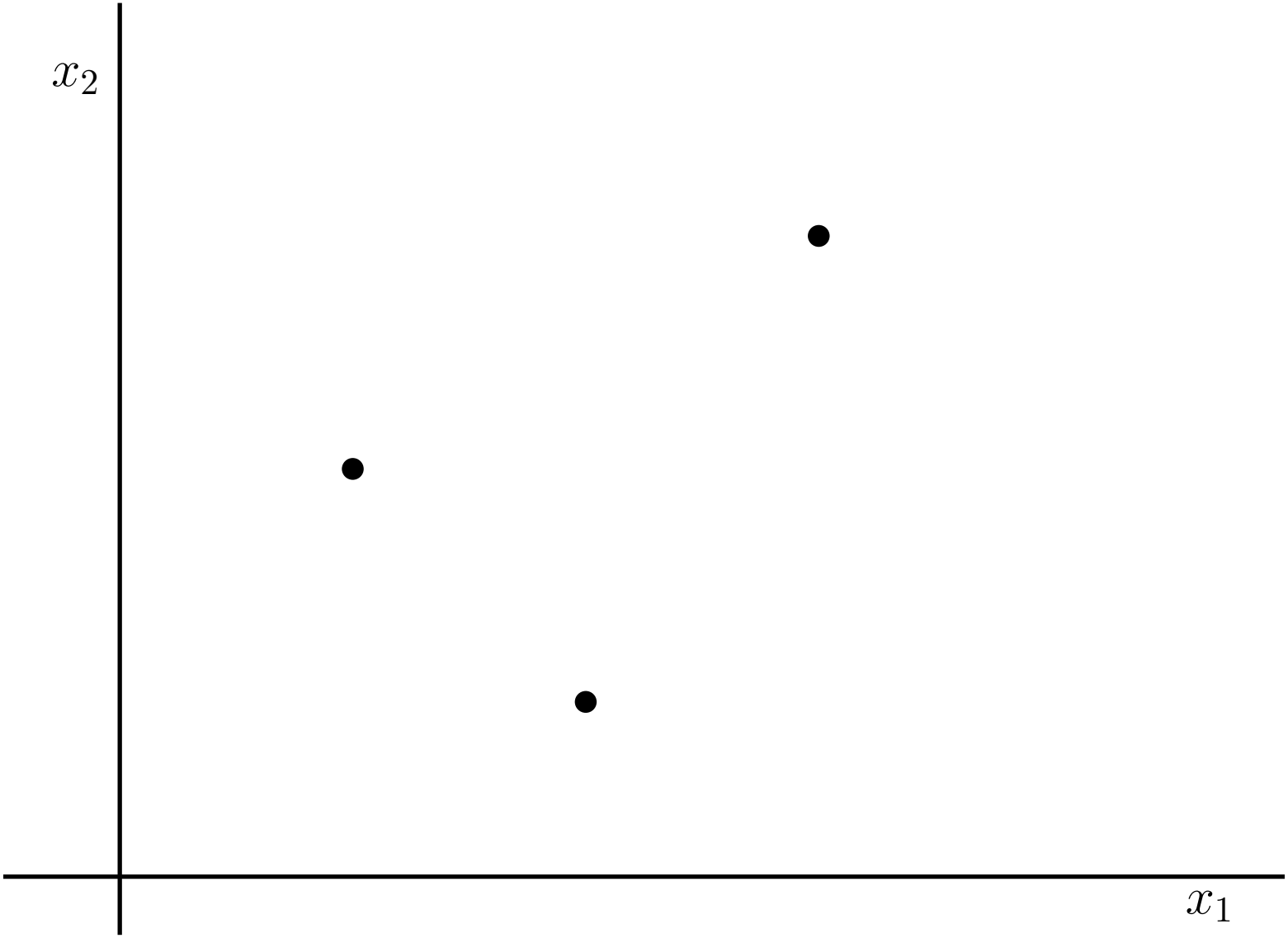
A linear classifier in $\mathbb{R}^d$ is a classifier of the form:

$$h(x_1, x_2, \ldots, x_d) = \begin{cases} +1 & \text{if } w_0 + \sum_{j=1}^{d} w_j x_j \geq 0 \\ -1 & \text{if } w_0 + \sum_{j=1}^{d} w_j x_j < 0 \end{cases}$$

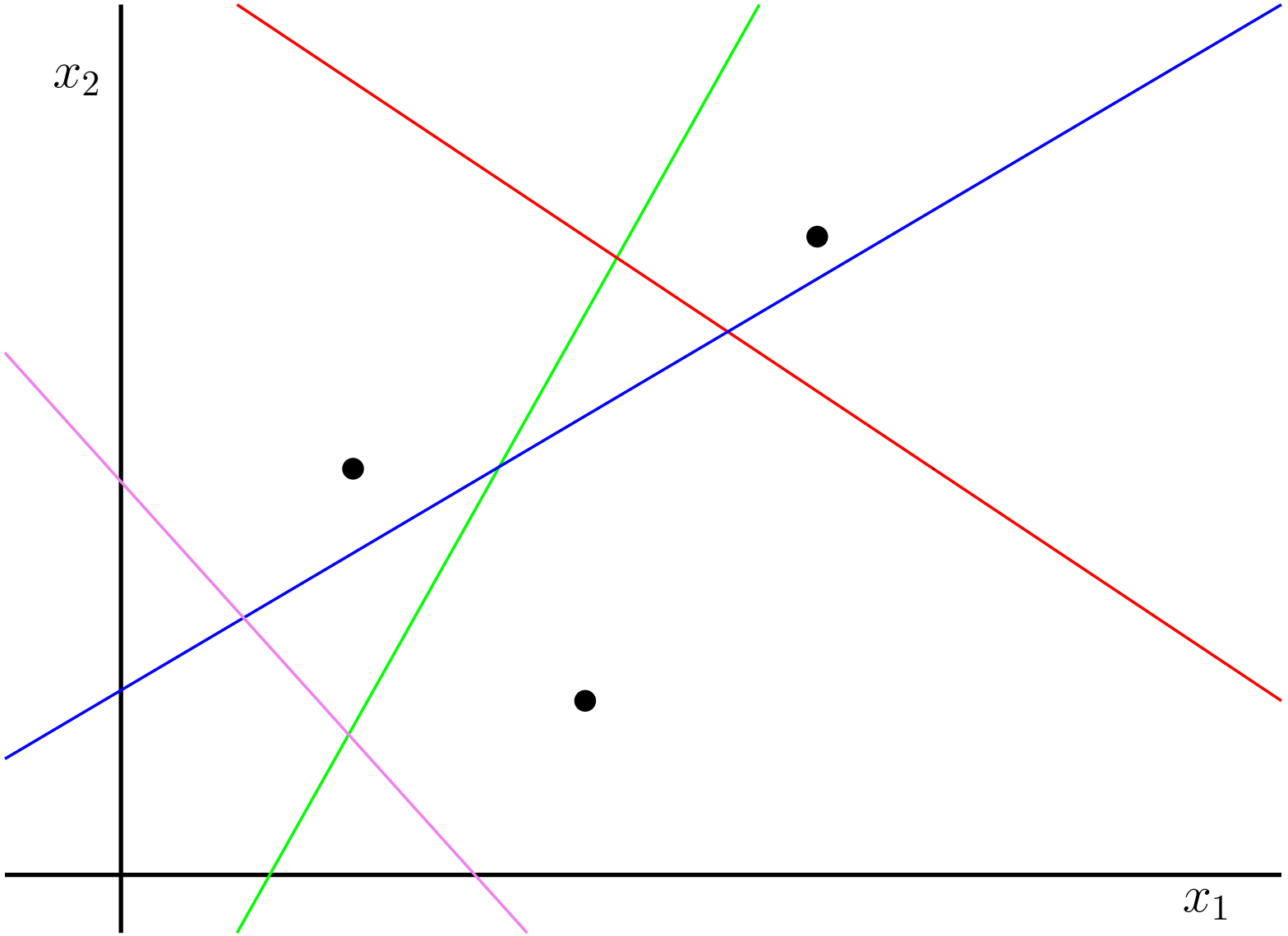
- The $d + 1$ weights w_j can have any real number as a value.
- In $\mathbb{R}^2$ the decision boundary $w_0 + w_1 x_1 + w_2 x_2 = 0$ is a line.
- In $\mathbb{R}^3$ the decision boundary $w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 = 0$ is a plane.

A logistic regression model is a linear classifier.

VC-dimension of linear classifiers



VC-dimension of linear classifiers



VC-dimension of linear classifiers

- If $d = 2$, a linear classifier can shatter a set of 3 points.
- If $d = 2$, a linear classifier can not shatter *any* set of 4 points.
- Hence, for $d = 2$, the VC-dimension of linear classifiers is 3.
- In general, the VC-dimension of linear classifiers is $d + 1$.

VC-dimension and sufficient sample size

If we require

$$\Pr_{D \sim P(X,Y)^m} \left[E_P(h_D) > \min_{h \in \mathcal{H}} \{ E_P(h) \} + \epsilon \right] \leq \delta,$$

where h_D is any hypothesis output by an ERM learner, then it holds for the sufficient sample size that

$$m = O\left(\frac{d_{vc} \ln(1/\epsilon) + \ln(1/\delta)}{\epsilon^2} \right),$$

where d_{vc} is the VC-dimension of $\mathcal{H}$.

This bound is linear in the VC-dimension.

VC-dimension and PAC learning

For binary classification, $\mathcal{H}$ has finite VC dimension if and only if: for every $\epsilon, \delta \in (0,1)$, there exists a finite sample size $m_{\mathcal{H}}(\epsilon, \delta)$ such that, for every distribution $P(X,Y)$ and every $m \geq m_{\mathcal{H}}(\epsilon, \delta)$,

$$\Pr_{D \sim P(X,Y)^m} \left[\forall h \in \mathcal{H} : \left| E_P(h) - E_D(h) \right| \leq \epsilon \right] \geq 1 - \delta.$$

If the VC dimension is infinite, no finite sample-size function can provide these guarantees uniformly over all distributions.

VC-dimension and PAC learning

If $\mathcal{H}$ has infinite VC dimension, then for some fixed accuracy and confidence requirements, whatever finite sample size m you choose, there is a distribution for which the guarantee fails.
The intuition is that the class can fit arbitrarily many points in arbitrary ways. For example:

- Choose $2m$ points that $\mathcal{H}$ shatters, and distribute probability uniformly over them.
- Label all those points $+1$.
- A sample of size m observes at most half the points.
- Because the points are shattered, some hypothesis labels all observed points $+1$ and all unobserved points -1.
- That hypothesis has zero training error but true error at least $1/2$. Thus, for this distribution, uniform convergence fails for every sample of size m whenever $\epsilon < 1/2$.

VC-dimension and PAC learning

Practically, this means small training error alone cannot certify small true error without further restrictions. For example, restricting the hypothesis class or making assumptions about the distribution. It does not mean learning is impossible on every particular problem, or that every learner will choose the bad hypothesis!

Consider for example the class of monotone classifiers in $\mathbb{R}^2$.