Machine Learning I

Lecture 6: The Vapnik-Chervonenkis Dimension

Nathaniel Bade

Northeastern University Department of Mathematics

Table of contents

Review of Results

State of the Union

The PAC and APAC formalize has provided us a lot to say about the possibility of learning intractable problems.

PAC Learning: For a labeling task, a hypothesis class $\mathcal{H}$ is **PAC** learnable if the probability of an ERM learning algorithm being approximately correct can be bounded by the size of the training set, *provided* **realizability** holds, i.e. some element of $\mathcal{H}$ provides the labeling, up to a set of measure 0.

APAC Learning: For learning task defined by a loss function and a domain, a hypothesis class $\mathcal{H}$ is **APAC** learnable if the probability an ERM learning algorithm returns a classifier with error close to the best error in the hypothesis class can be bounded by the size of the training set.

State of the Union

Over the last lectures, we proved that several hypothesis classes were, respectively, PAC and APAC learnable:

PAC Learnable: The (infinite) class of interval indicator functions is PAC learnable, with sample complexity

$$N_{\mathcal{H}}(\epsilon, \delta) \leq \text{ceil}\left(\frac{2\log(2/\delta)}{\epsilon}\right).$$

APAC Learnable: Any finite hypothesis class $\mathcal{H}$ is APAC learnable, with sample complexity

$$N_{\mathcal{H}}(\epsilon, \delta) \leq \text{ceil}\left(\frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2}\right).$$

State of the Union

The examples before show that there are infinite classes of functions that are PAC learnable and finite classes of functions that APAC learnable. This raises a few questions:

> Are PAC learnable classes APAC learnable?
>
> Are *all* classes PAC learnable?
>
> Is there an efficient way to tell if a class is PAC/APAC learnable?

We will answer all these questions this lecture. The most surprising might be that the we can answer the first question in the affirmative. But what is an example of an unlearnable class?

State of the Union

The examples before show that there are infinite classes of functions that are PAC learnable and finite classes of functions that APAC learnable. This raises a few questions:

> Are PAC learnable classes APAC learnable? **Yes, surprisingly.**
>
> Are *all* classes PAC learnable? **No.**
>
> Is there a easy way to tell if a class is PAC/APAC learnable? **Yes.**

We will answer all these questions this lecture. The most surprising might be that the we can answer the second question in the affirmative. But what is an example of an unlearnable class?

The class of all functions is unlearnable.

No Free Lunch

Recall that we also proved

No Free Lunch Theorem: For any learning algorithm for the task of binary classification with respect to the 0-1 loss function, provided the training size $\mathcal{T}$ is less than $|\mathcal{X}|/2$, there exists a learnable distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$ such that

$$L_{\mathcal{D}}(A(\mathcal{T})) \geq 1/8$$

with probability $1/7$. That is any learning algorithm A and any class $\mathcal{H}$ has a learnable task at which they do poorly.

This gives us a direct proof of the fact that the class of all functions is unlearnable.

PAC Unlearnable

(SB - Corollary 5.2) Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}_{all}$ be the set of all functions from $\mathcal{X}$ to $\{0, 1\}$. Then $\mathcal{H}$ is not PAC learnable.

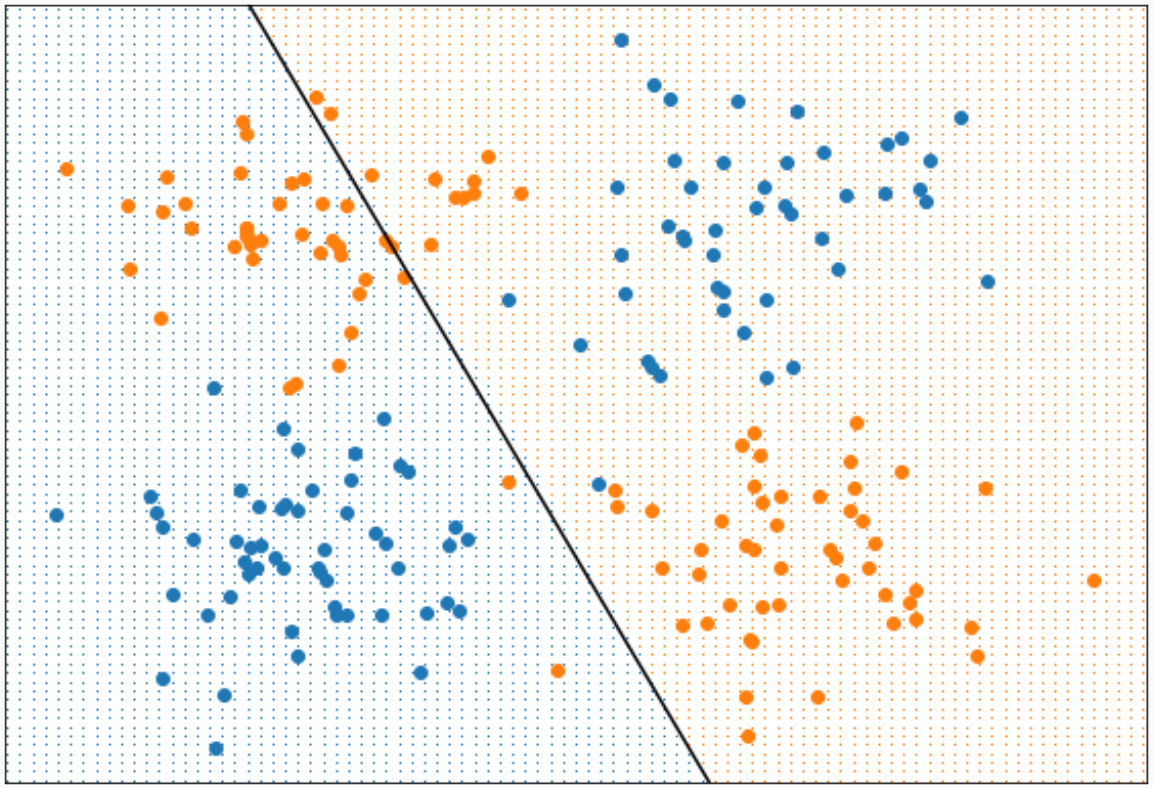
Proof: Assume $\mathcal{H}_{all}$ is PAC learnable by a algorithm A. By the No Free Lunch theorem, there is some distribution $\mathcal{D}$ on $\mathcal{X} \times \{0, 1\}$ and some function $f : \mathcal{X} \to \{0, 1\}$ such that $L_{\mathcal{D}}(f) = 0$, and $L_{\mathcal{D}}(A(\mathcal{T})) > 1/8$ with probability greater than $1/7$.

However, the existence of an f such that $L_{\mathcal{D}}(f) = 0$ means $\mathcal{H}_{all}$ is realizable. That implies that for any $\delta < 1/7$ and $\epsilon < 1/8$, there exists an N such that if $|\mathcal{T}| > N$, $L_{\mathcal{D}}(A(f)) < \epsilon$.

This is a contradiction, so $\mathcal{H}_{all}$ is not PAC learnable.

□

PAC Unlearnable



Question: Why doesn't this proof imply that **no** function is PAC learnable?

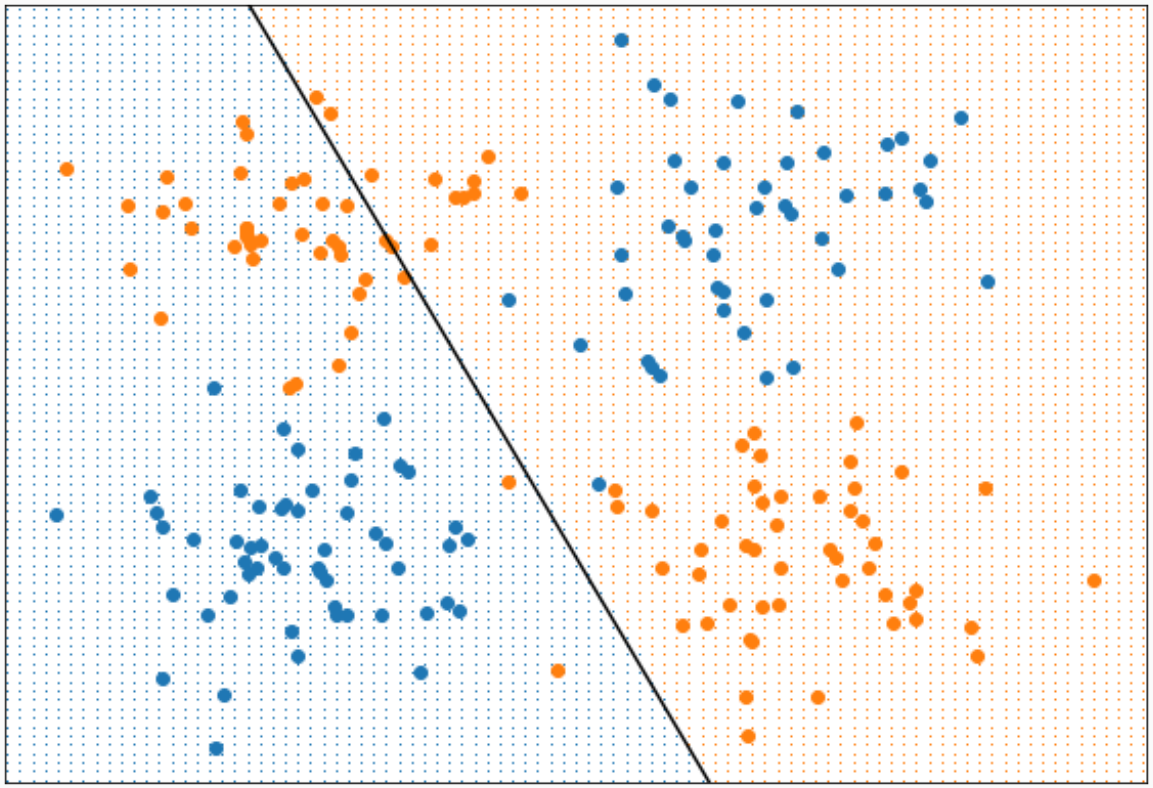
The VC Dimension

VC Dimension

We have seen that finiteness is a sufficient criteria for learnability. We have also seen that while some infinite classes are PAC learnable some infinite classes are not. In other words, the cardinality of $\mathcal{H}$ isn't the proper decider of learnability.

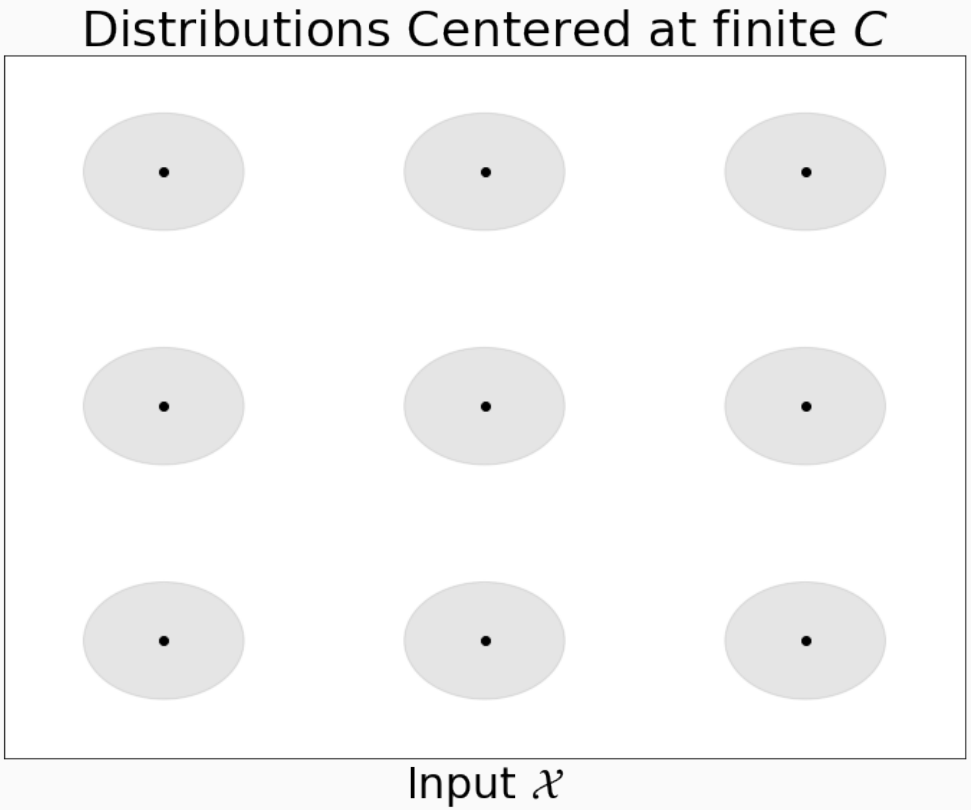
We will show that a property called the VC-dimension of a hypothesis class gives the correct characterization of (A)PAC learnability. To motivate VC-dimension, lets recall a bit about how we proved the No Free Lunch Theorem.

No Free Lunch Arguments



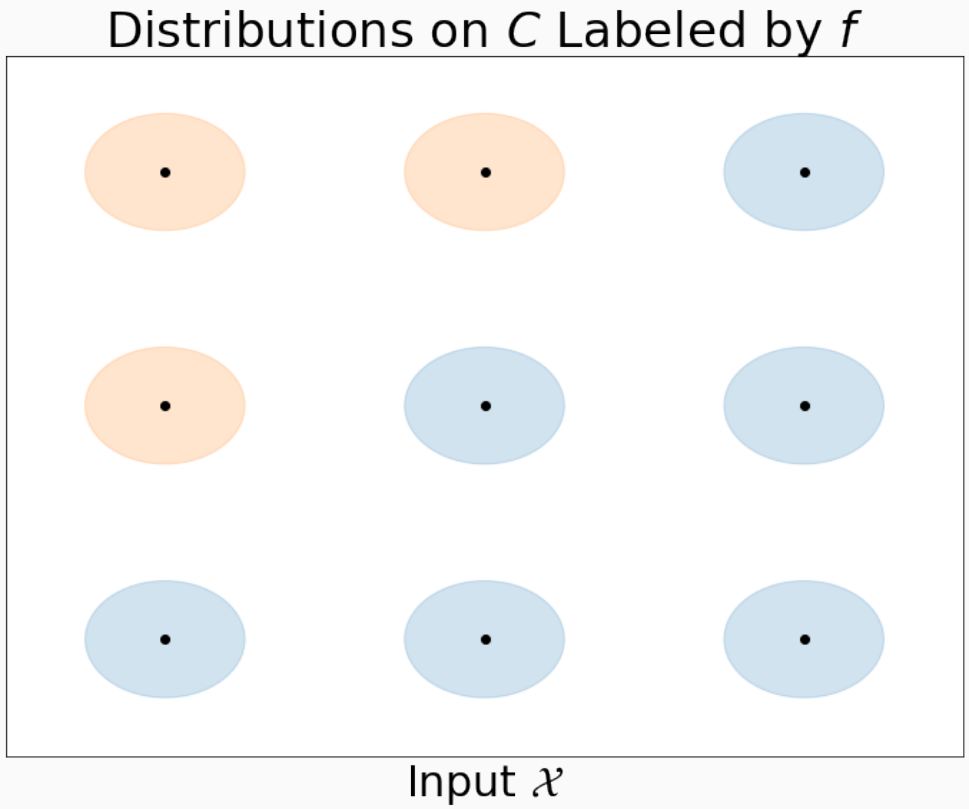
The idea of the proof was to construct a distribution $\mathcal{D}$ on which the learning algorithm would perform poorly.

No Free Lunch Arguments



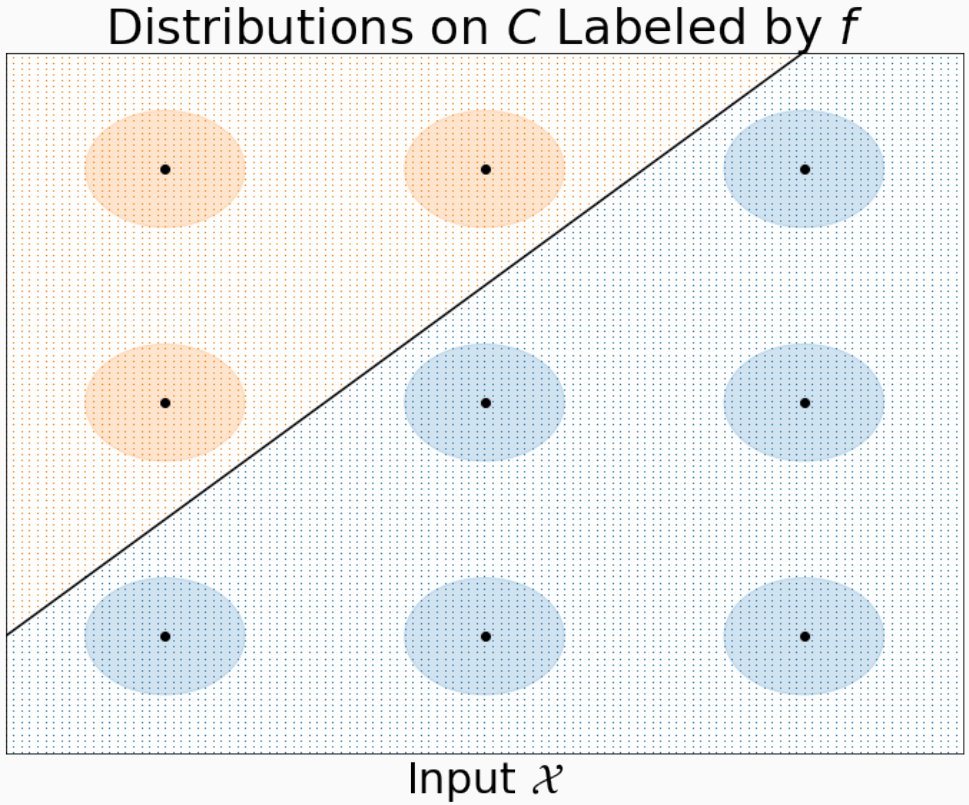
The idea of the proof was to construct an distribution $\mathcal{D}$ on which the learning algorithm would perform poorly. We defined a class of distributions concentrated on a finite set $C \subseteq \mathcal{X}$ with a deterministic "true" labeling function f chosen from the set of all functions.

No Free Lunch Arguments



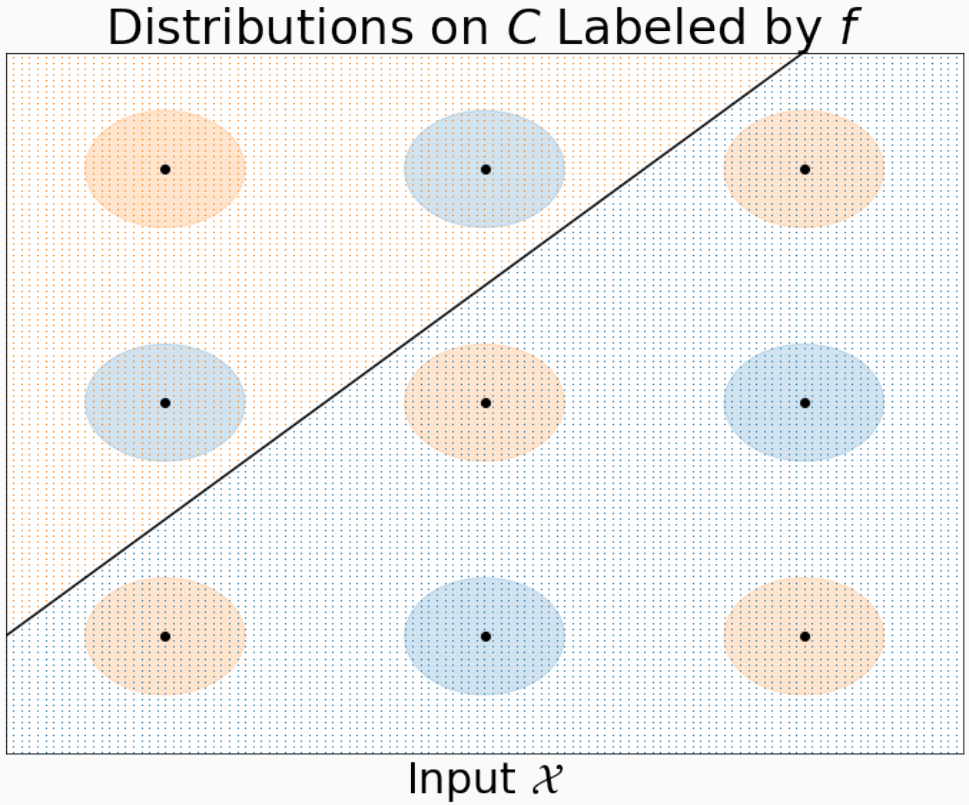
The idea of the proof was to construct an distribution $\mathcal{D}$ on which the learning algorithm would perform poorly. We defined a class of distributions concentrated on a finite set $C \subseteq \mathcal{X}$ with a deterministic "true" labeling function f chosen from the set of all functions.

No Free Lunch Arguments



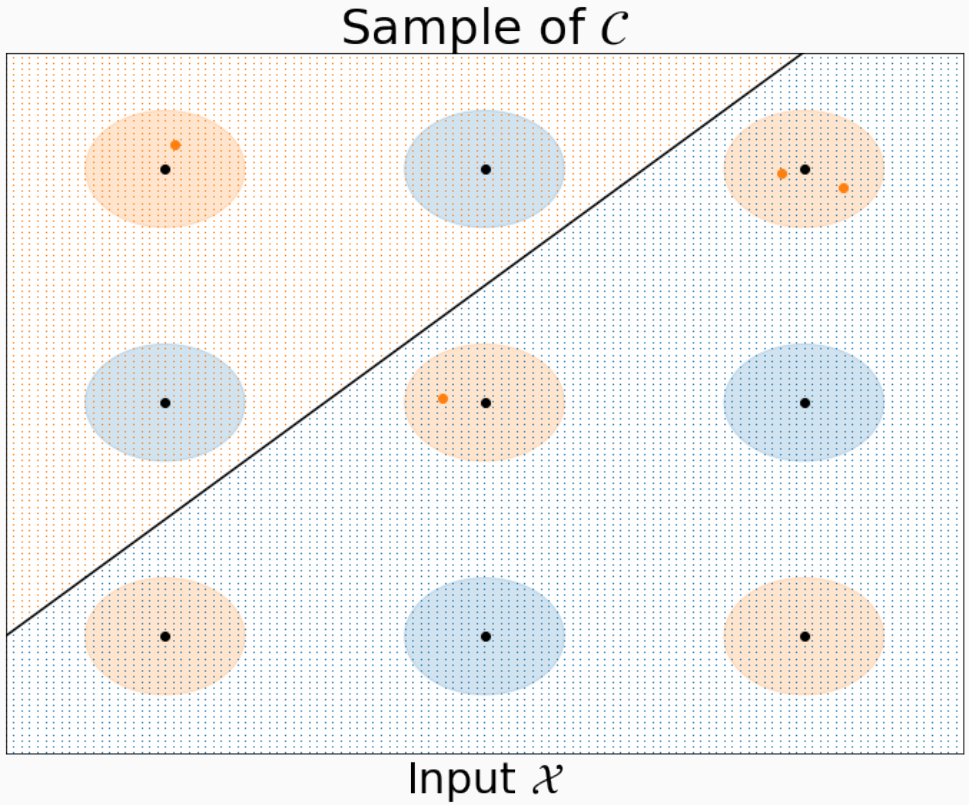
The idea of the proof was to construct an distribution $\mathcal{D}$ on which the learning algorithm would perform poorly. We defined a class of distributions concentrated on a finite set $C \subseteq \mathcal{X}$ with a deterministic "true" labeling function f chosen from the set of all functions.

No Free Lunch Arguments



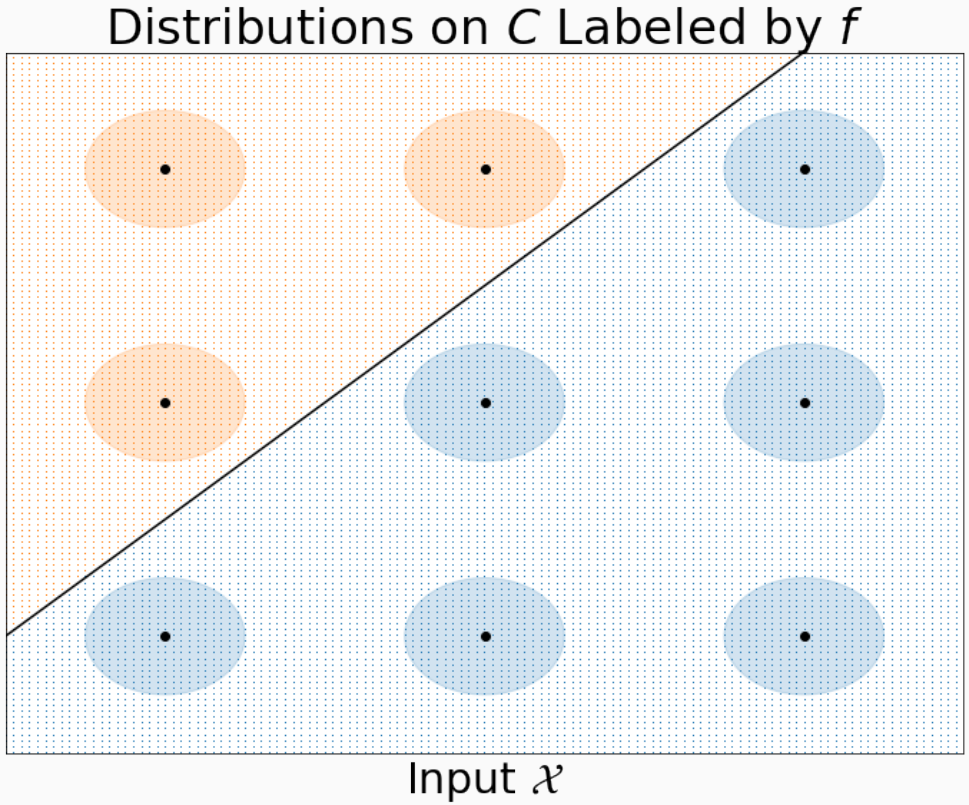
To make an algorithm fail, we used the fact that we could pick a labeling from the set of all possible functions from $C \to \{0, 1\}$.

No Free Lunch Arguments



To make the algorithm fail, we used the fact that we could pick a labeling from the set of all possible functions from $C \rightarrow \{0, 1\}$. We saw there was then a high probability that a random training sample will produce a "bad" classifier.

No Free Lunch Arguments



When considering PAC learnability, realizability restricts the distribution to only the ones for which some hypothesis in $\mathcal{H}$ achieves zero risk. So if $\mathcal{H}$ restricted to C is the set of all functions, we can construct a **realizable** distribution that $\mathcal{H}$ must fail to learn. If $\mathcal{H}$ restricted to C is not the set of all functions, we have a hope $\mathcal{H}$ is learnable for all realizable tasks.

Restricting

The restriction of $\mathcal{H}$ to C. Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X} \to \{0, 1\}$ and let

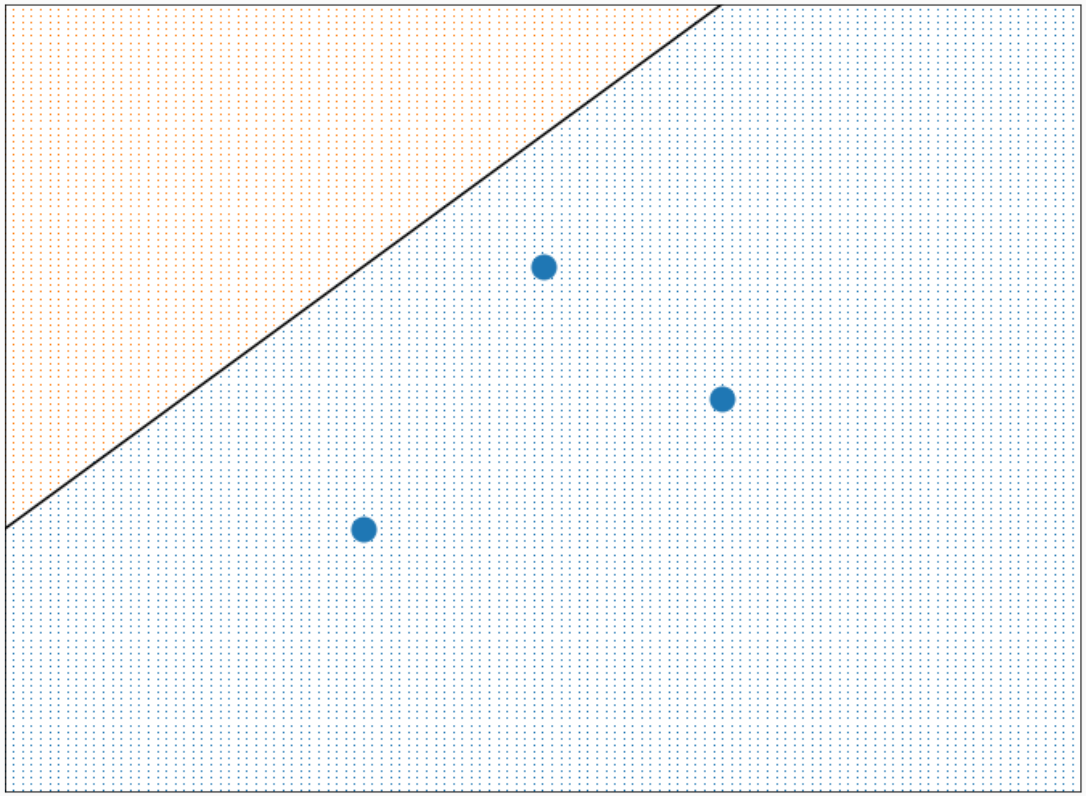
$$C = \{c_1, \ldots, c_m\} \subset \mathcal{X}$$

be a finite subset of $\mathcal{X}$. The **restriction** of $\mathcal{H}$ to C is the set of all function from C to $\{0, 1\}$ that can be derived from $\mathcal{H}$:

$$\mathcal{H}_C = \left\{ \big(h(c_1), \ldots, h(c_m) \big) : h \in \mathcal{H} \right\}.$$

If the restriction of $\mathcal{H}$ to C is the set of all function, we say $\mathcal{H}$ **shatters** C.

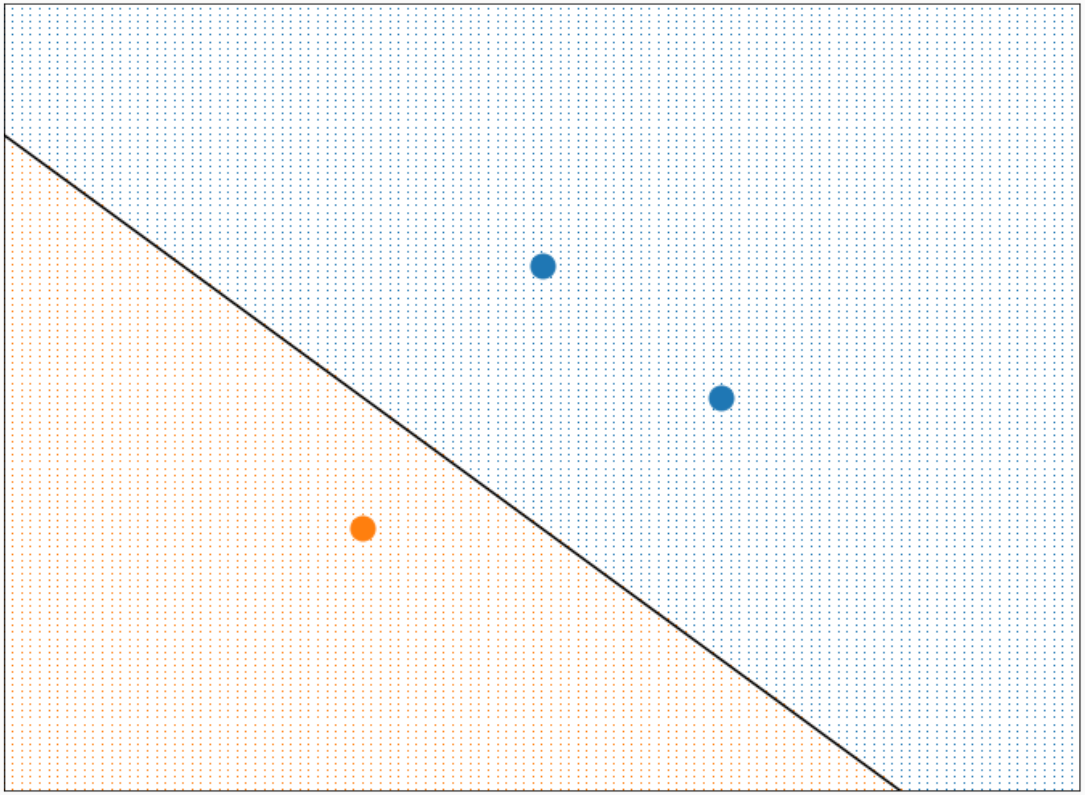
Shattering for Linear Classifier



A hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

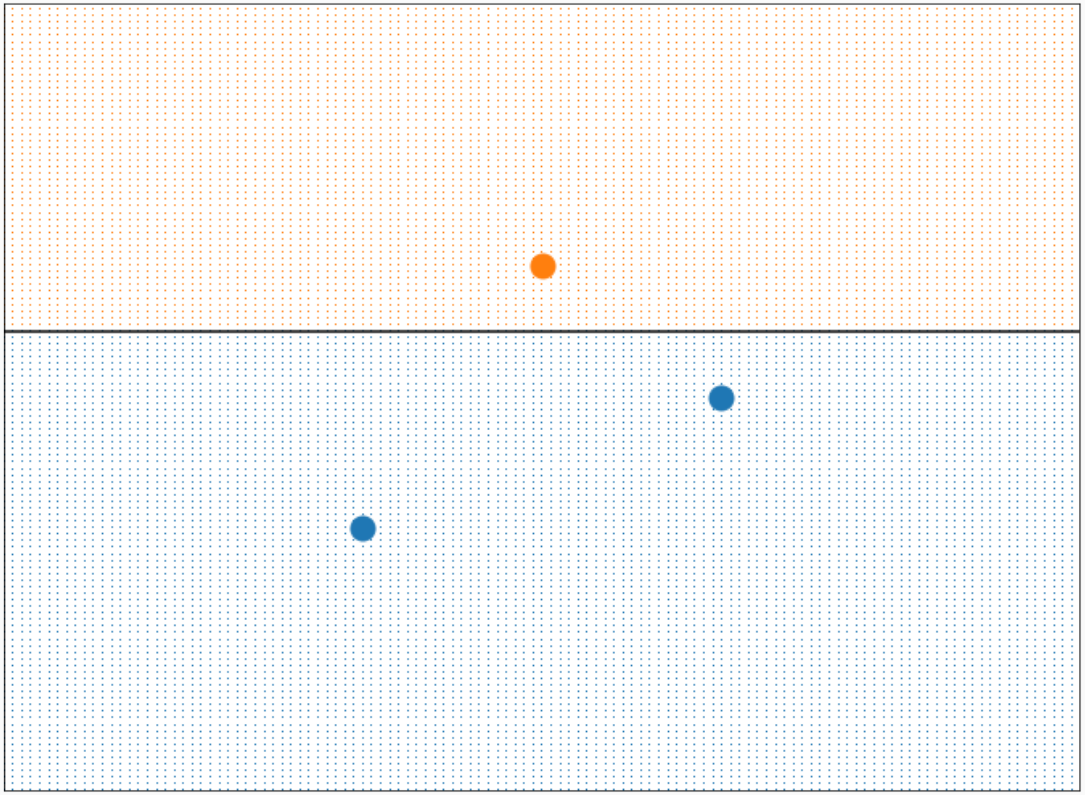
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

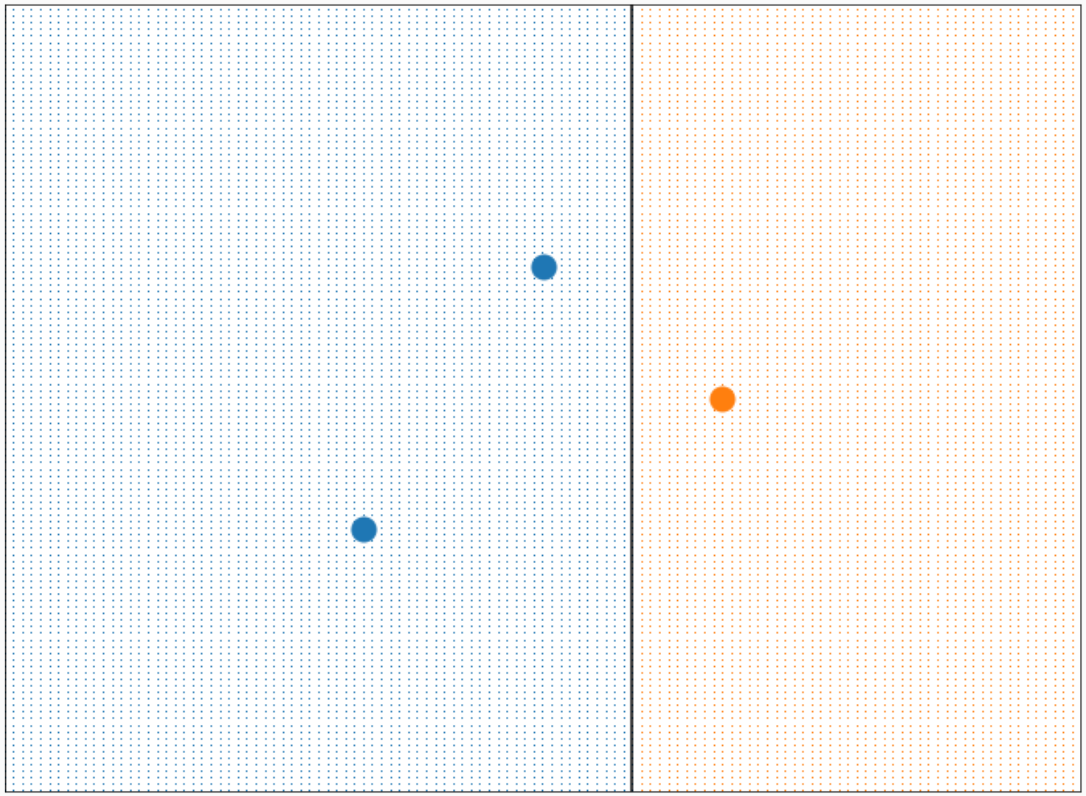
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

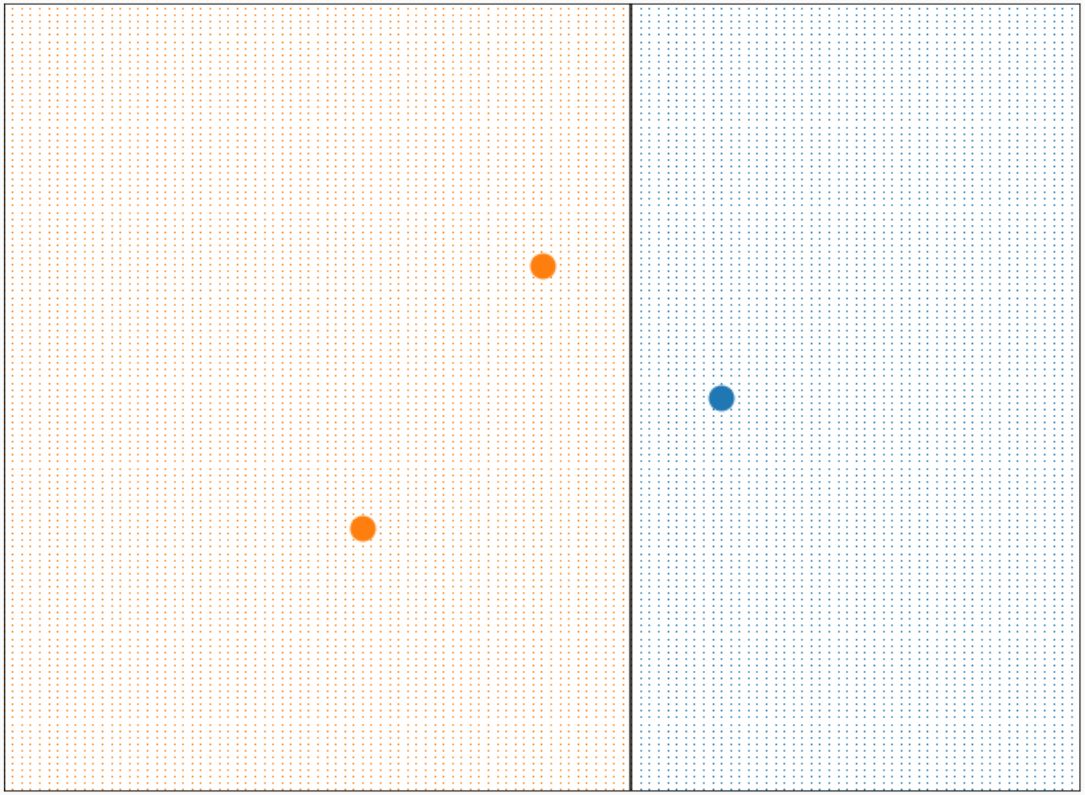
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

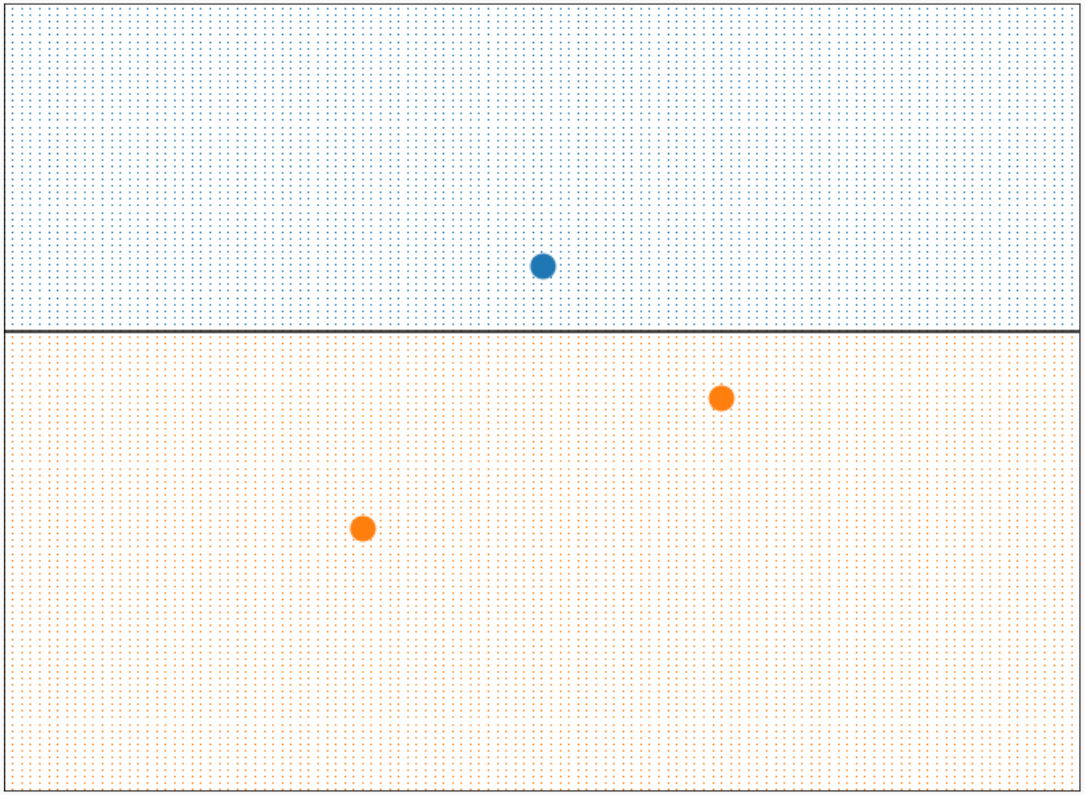
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

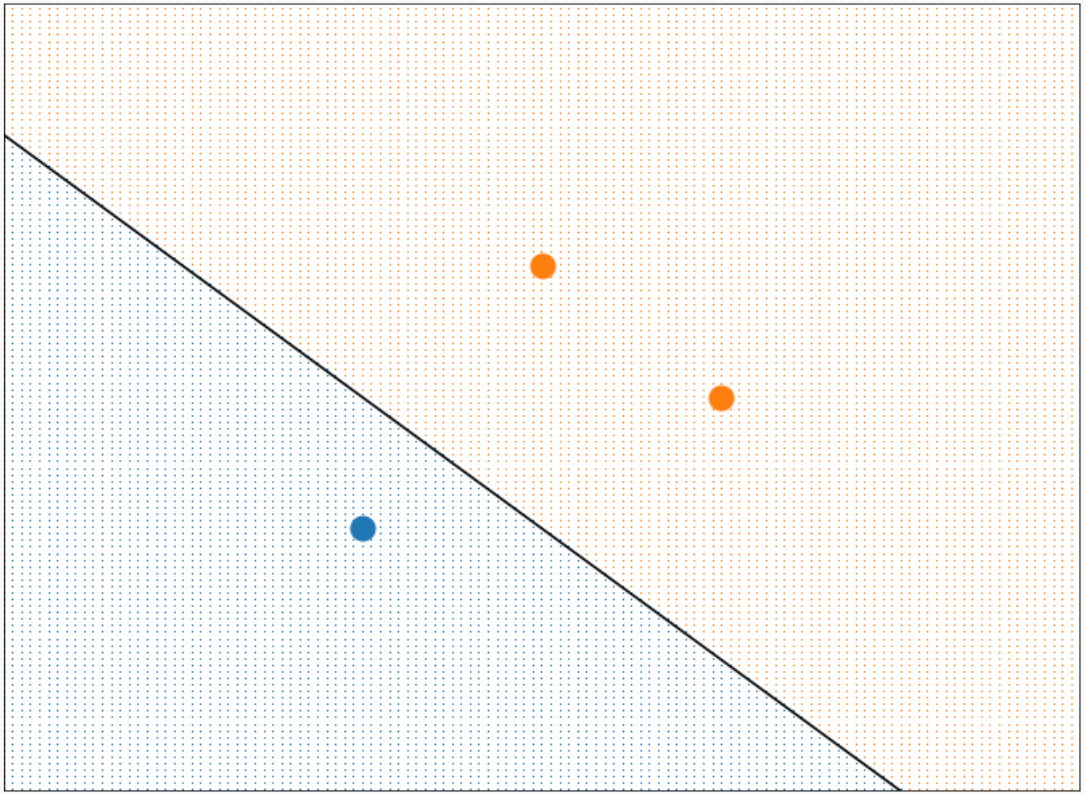
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

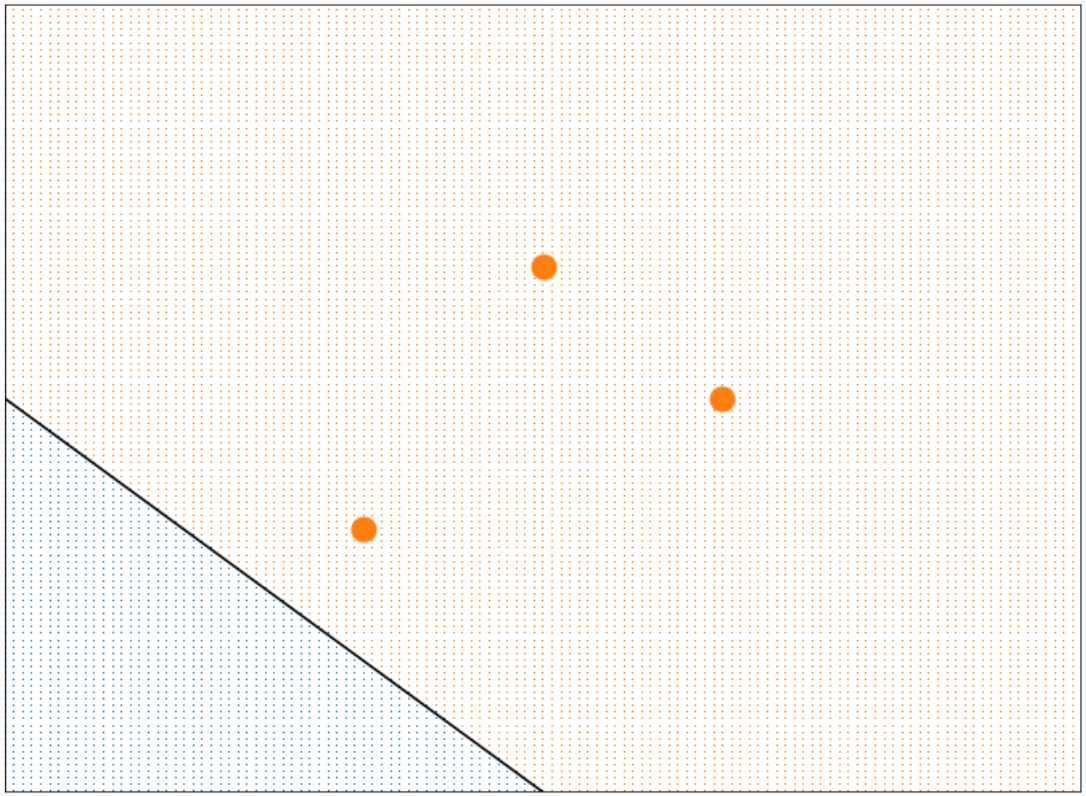
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

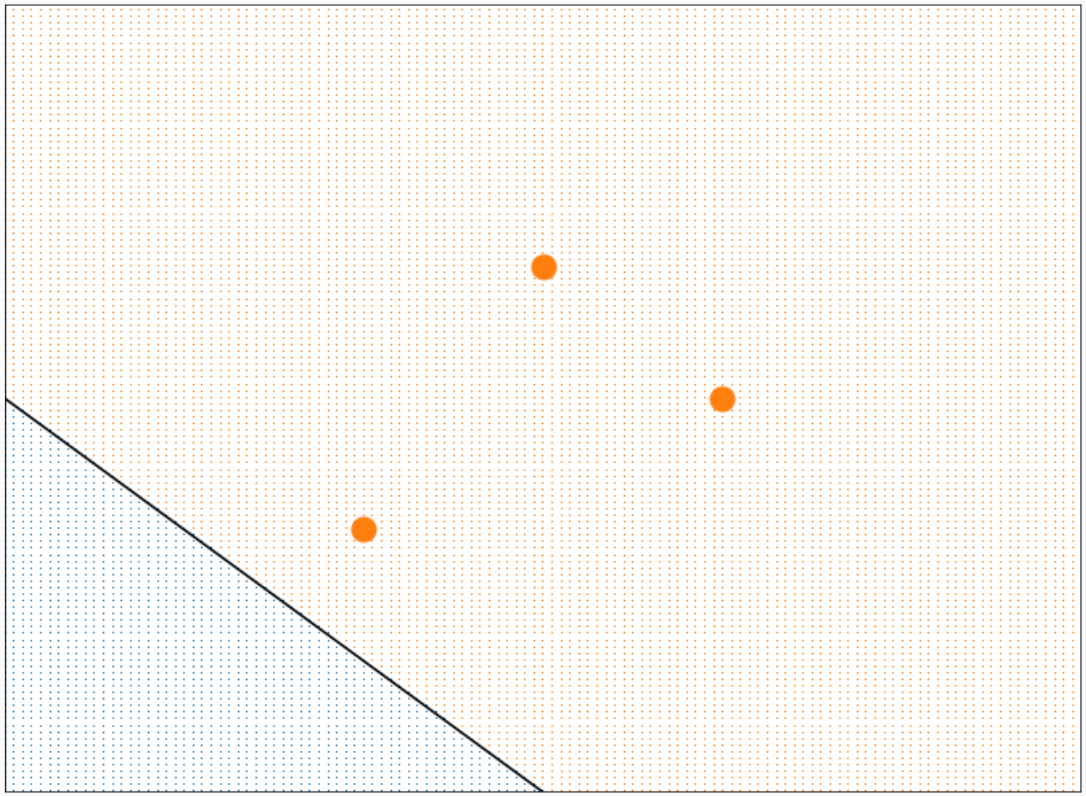
Shattering for Linear Classifier



Formally, a hypothesis class **shatters** a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all functions.

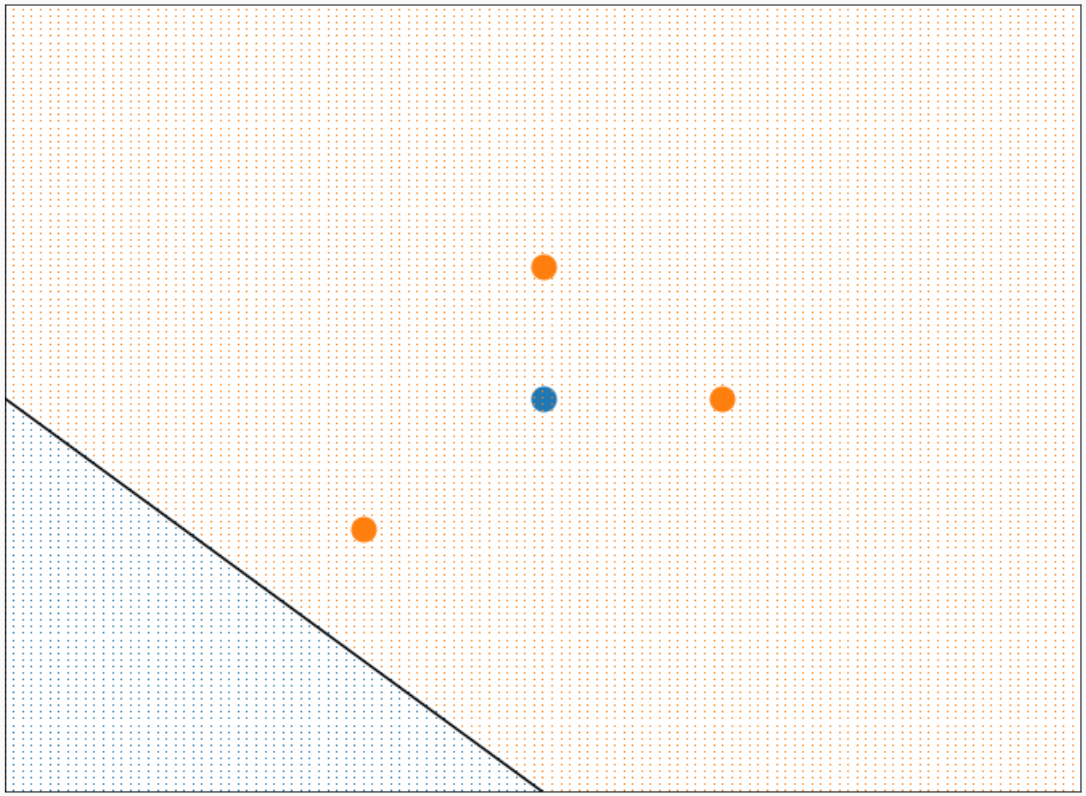
For example, the 3 point set can be shattered by the halfplane classifier, since any labeling on the three points can be generated by a half plane.

Shattering for Linear Classifier



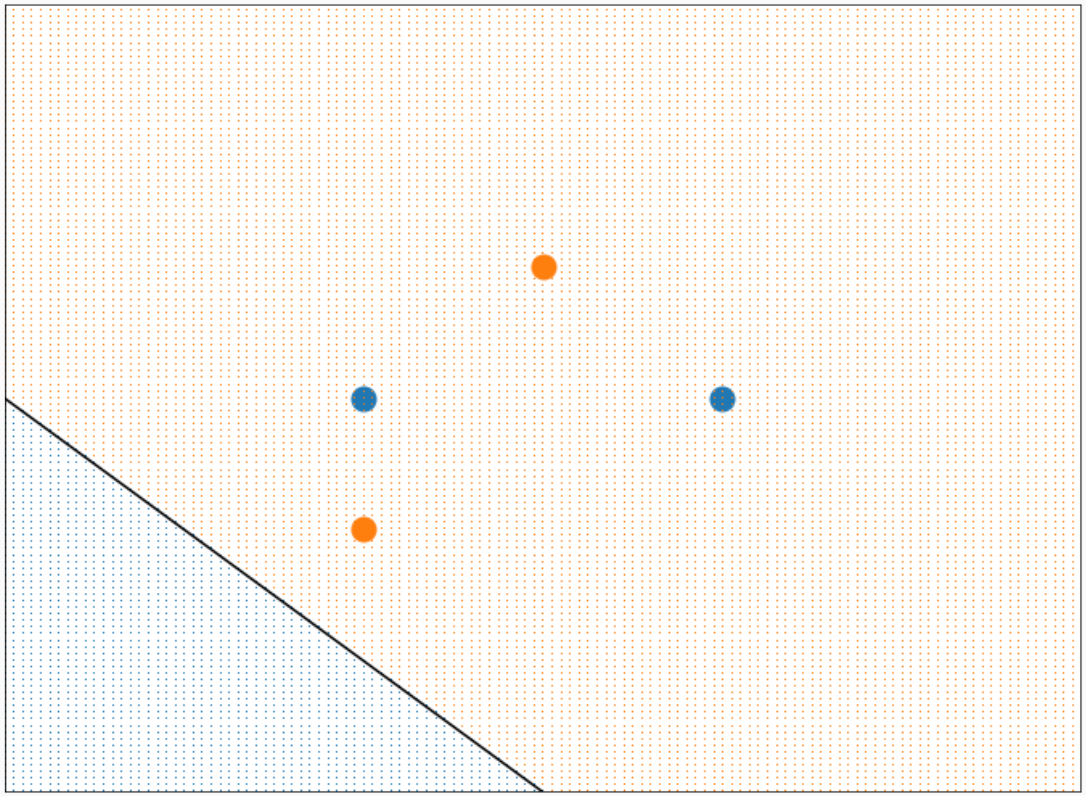
This means that for a training set of size $N = \text{floor}(\frac{3}{2})$, we can construct a distribution on 3 points that is unlearnable by the set of half plane classifiers.

Shattering for Linear Classifier



On the other hand, the restriction of the halfplane classifiers to any 4 point set C is not the set of all functions from $C \to \{0, 1\}$. Four point sets come in two configurations, a point inside the convex hull of three points,

Shattering for Linear Classifier



On the other hand, the restriction of the halfplane classifiers to any 4 point set C is not the set of the functions from $C \to \{0, 1\}$. Four point sets come in two configurations, a point inside the convex hull of three points, and a convex four polygone. In both cases, there are labelings that cannot be determined by a halfplane classifier.

Shattering and No Free Lunch

Putting this reasoning together, we have

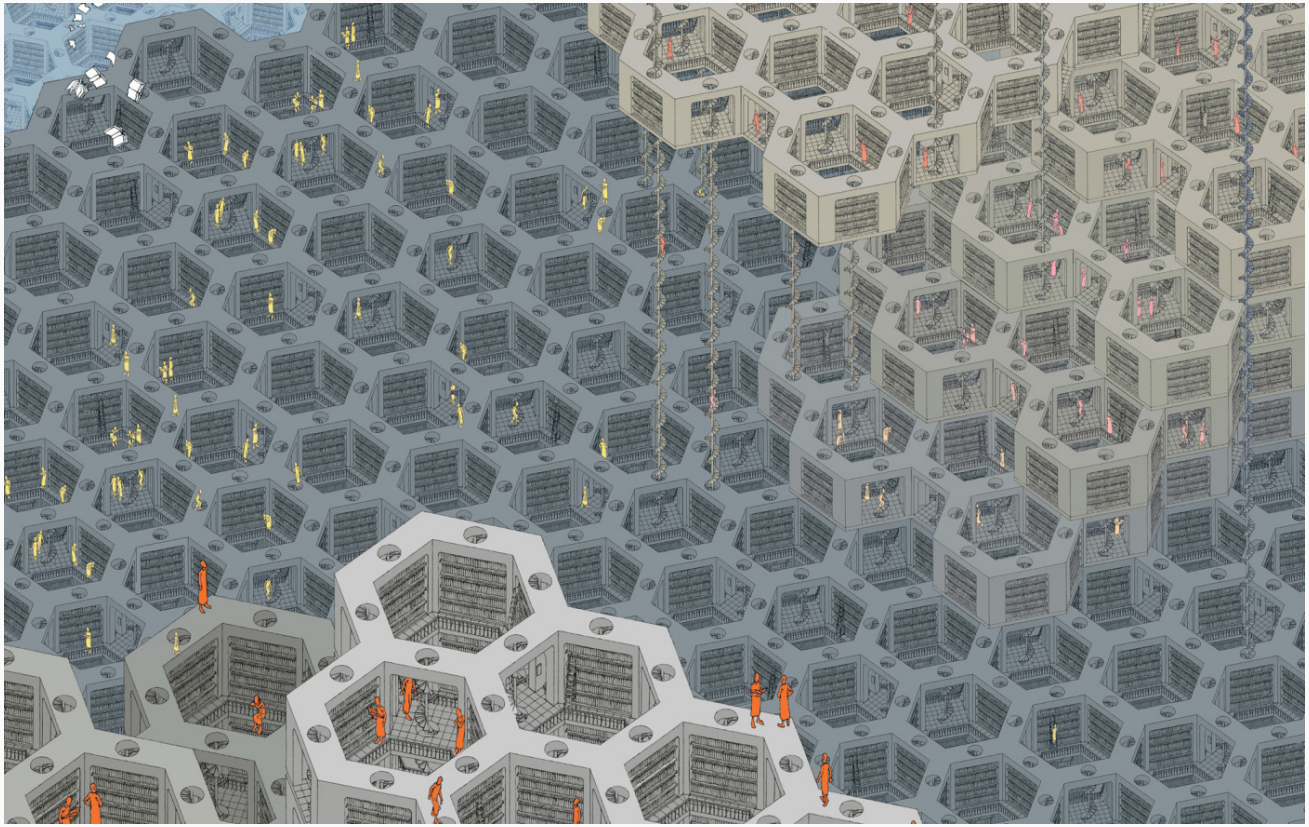
(SB - Theorem 6.6) Let $\mathcal{H}$ be a hypothesis class of functions $\mathcal{X} \to \{0, 1\}$ and N be a training set size. Assume there exists a set $C \subset \mathcal{X}$ of size $2N$ that is shattered by $\mathcal{H}$.

For any learning algorithm A, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$ and a predictor $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) = 0$, but with probability at least $1/7$ over the choice of training sets $\mathcal{T}$,

$$L_{\mathcal{D}}(A(\mathcal{T})) \geq 1/8\,.$$

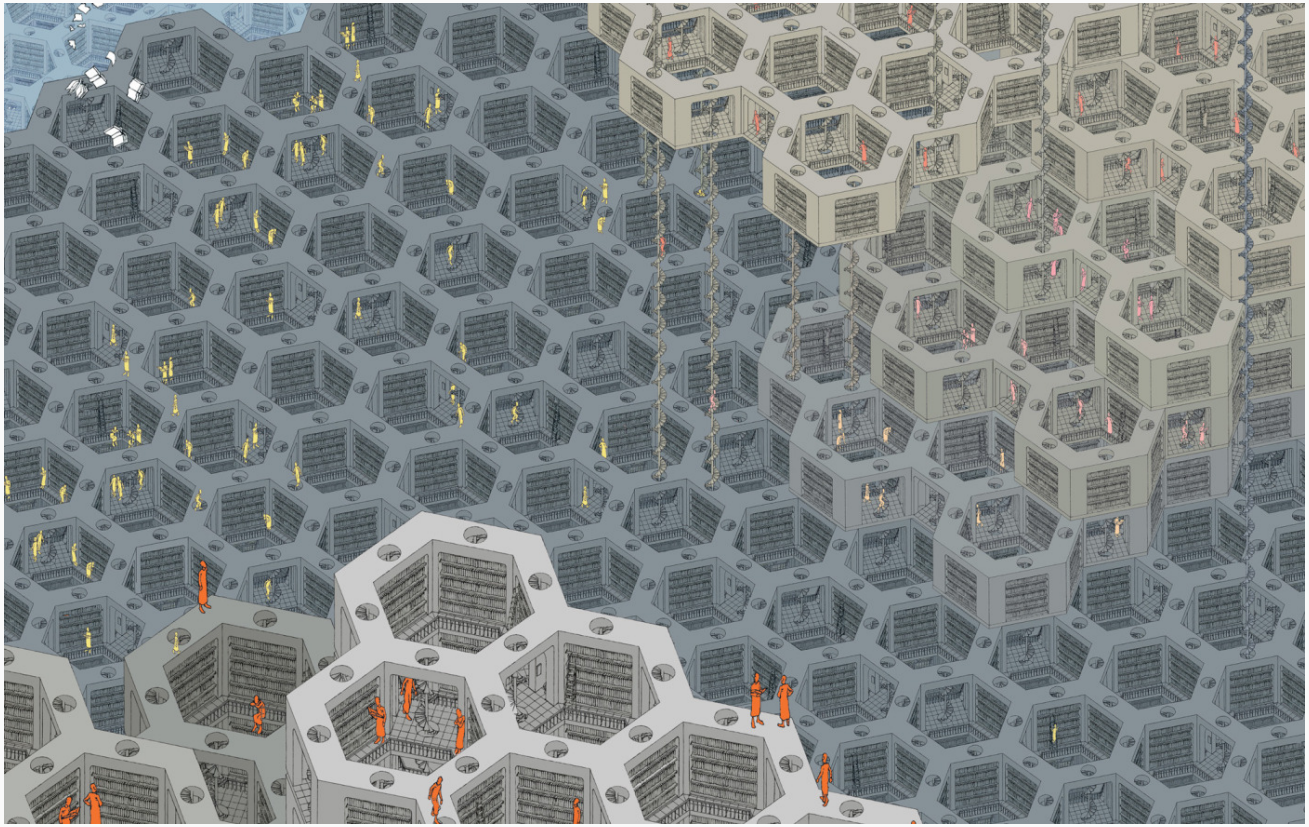
In other words, if $\mathcal{H}$ shatters a set C of size $2N$, we cant learn learn $\mathcal{H}$ using N examples.

Shattering and No Free Lunch



The author Borges has a story about an infinite library which contains a book with ever possible combination of letters. Somewhere in the library, books every truth must be lie but they are impossible to find, and the librarians spend their lives contending with gibberish.

Shattering and No Free Lunch



Philosophically, if a hypothesis class knows every labeling then, like Borges' library, any truth must be contained somewhere in it but will be impossible to find.

VC-Dimension

This lead us directly to the following definition: The **VC-dimension** of a hypothesis class, denoted VCDim($\mathcal{H}$) is the maximum size of a set that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrary size, we say it has infinite VC-dimension.

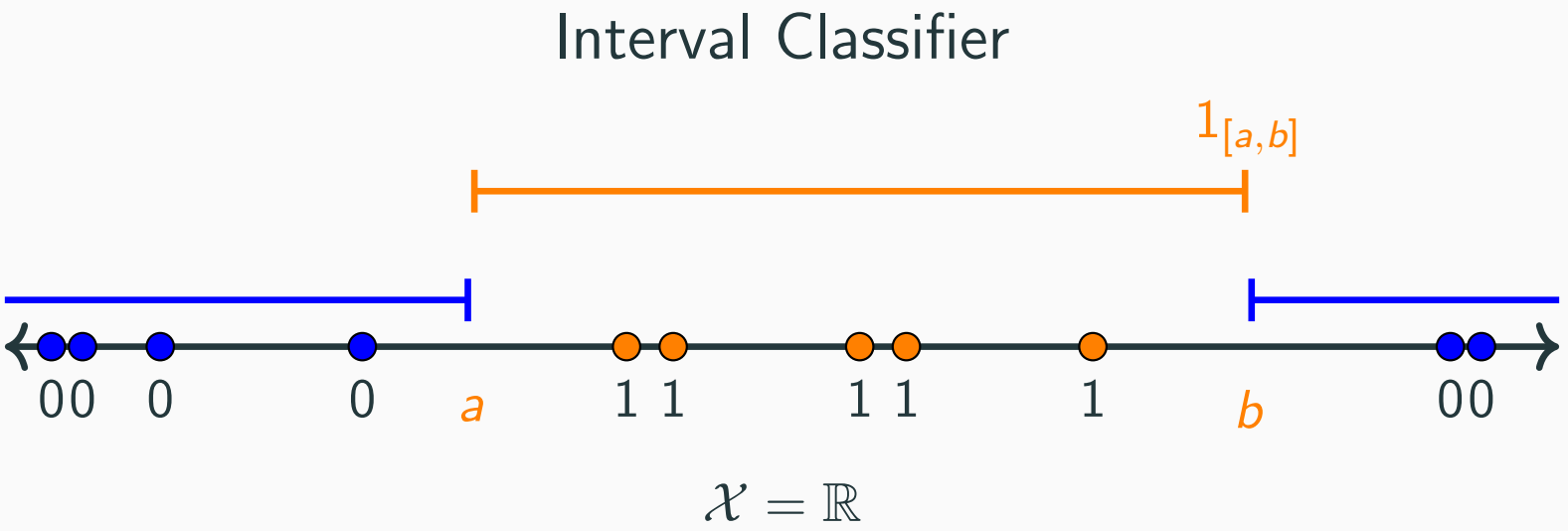
SB Theorem 6.6: If $\mathcal{H}$ is a class of infinite VC-dimension, then $\mathcal{H}$ is not PAC learnable.

Proof: Since $\mathcal{H}$ has infinite VC-dimension, for any N there is a set of size $2N$ shattered by $\mathcal{H}$. $\square$

We shall see later that if $\mathcal{H}$ has finite VC-dimension it is PAC learnable.

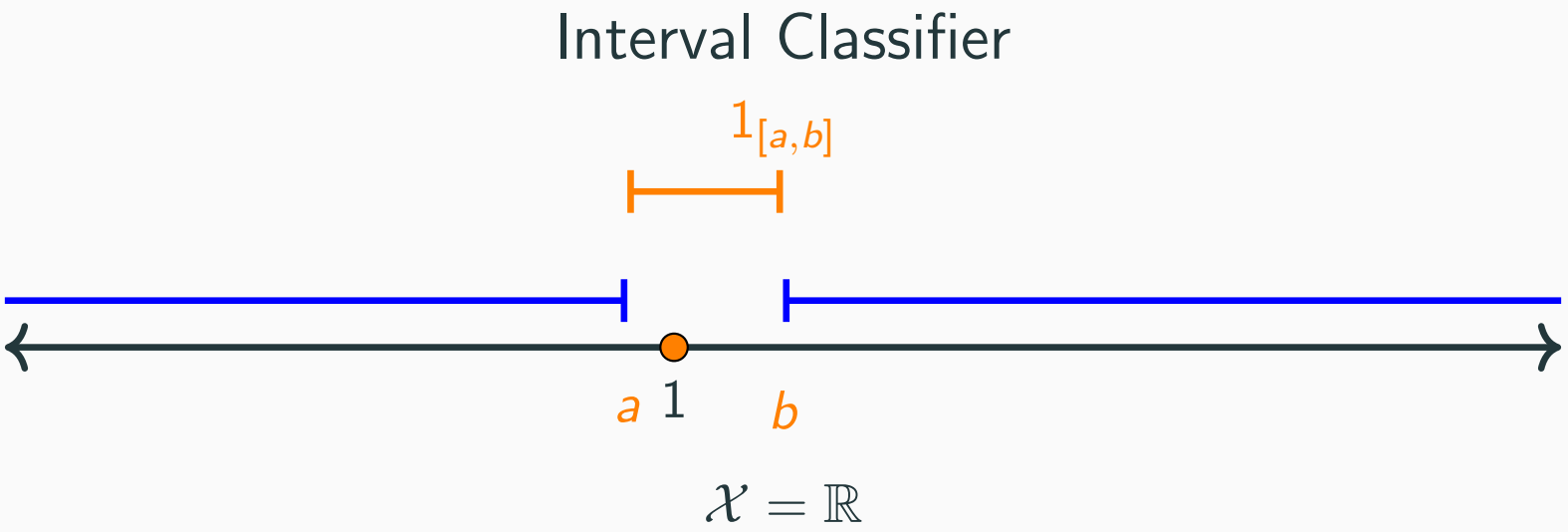
Examples of VC Dimension

VC-Dimension



Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

VC-Dimension

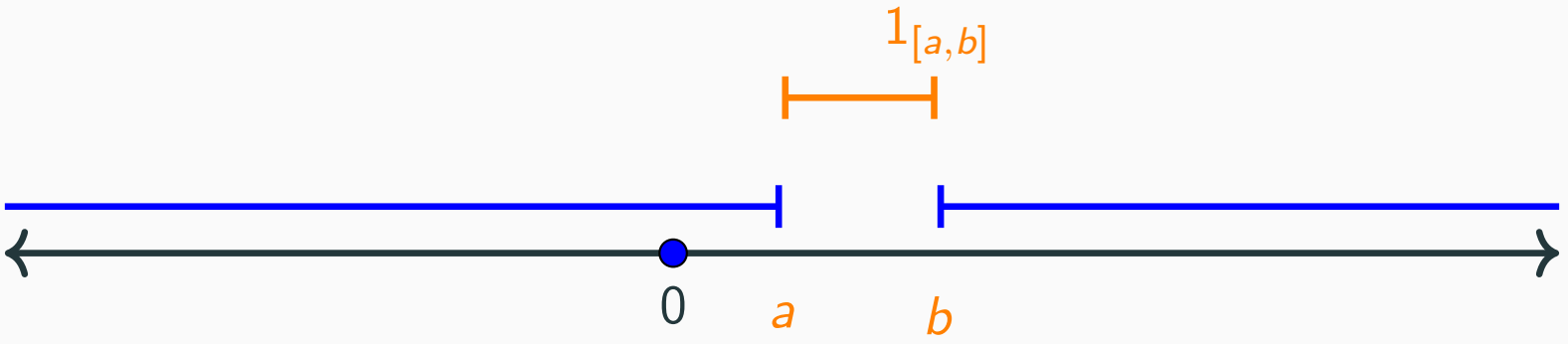


Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

The class $\mathcal{H}$ can clearly shatter a single point.

VC-Dimension

Interval Classifier

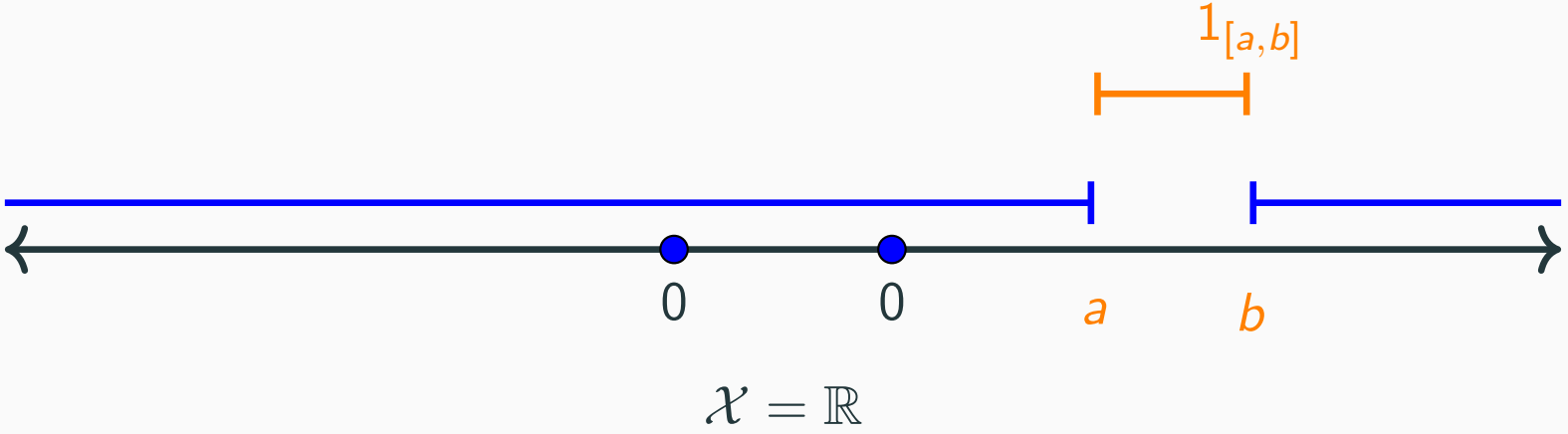


$$\mathcal{X} = \mathbb{R}$$

Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

The class $\mathcal{H}$ can clearly shatter a single point.

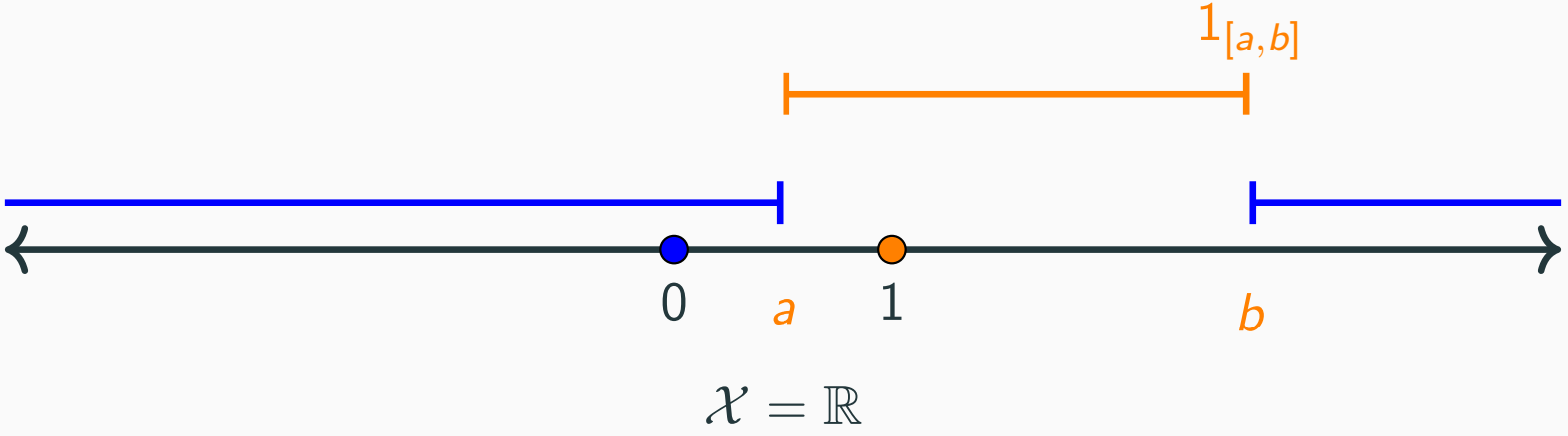
Interval Classifier



Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

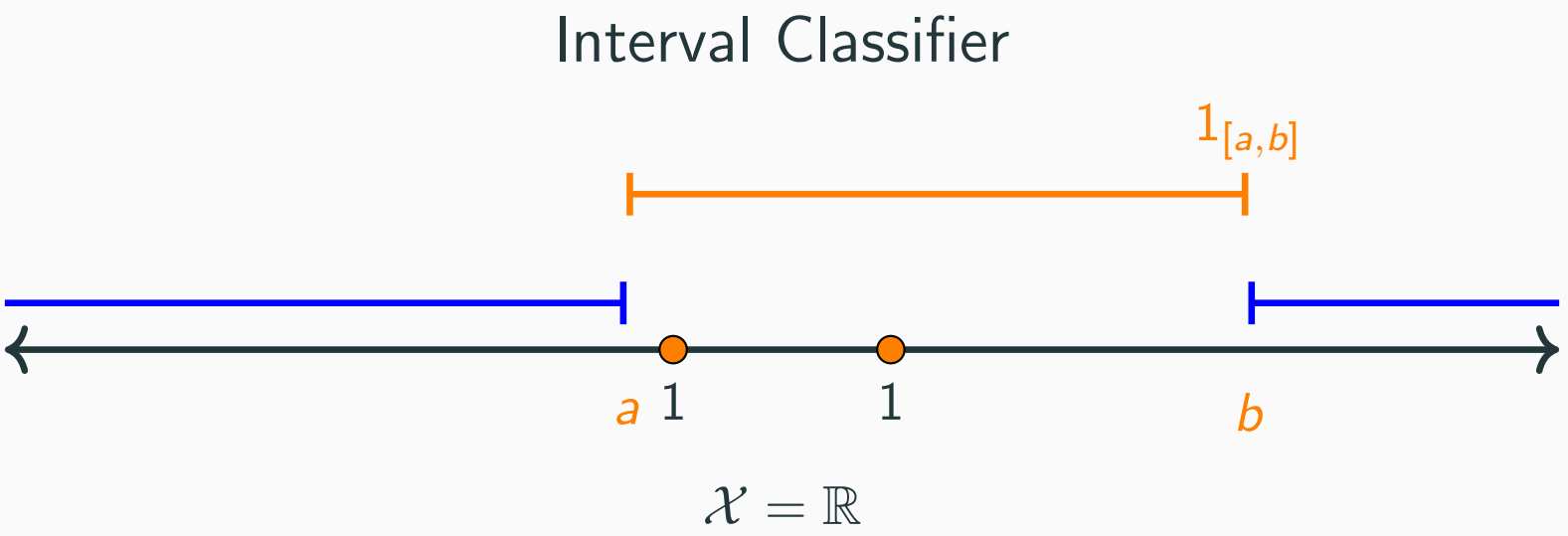
The class $\mathcal{H}$ can clearly shatter a single point. It can also shatter a two point set.

Interval Classifier



Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

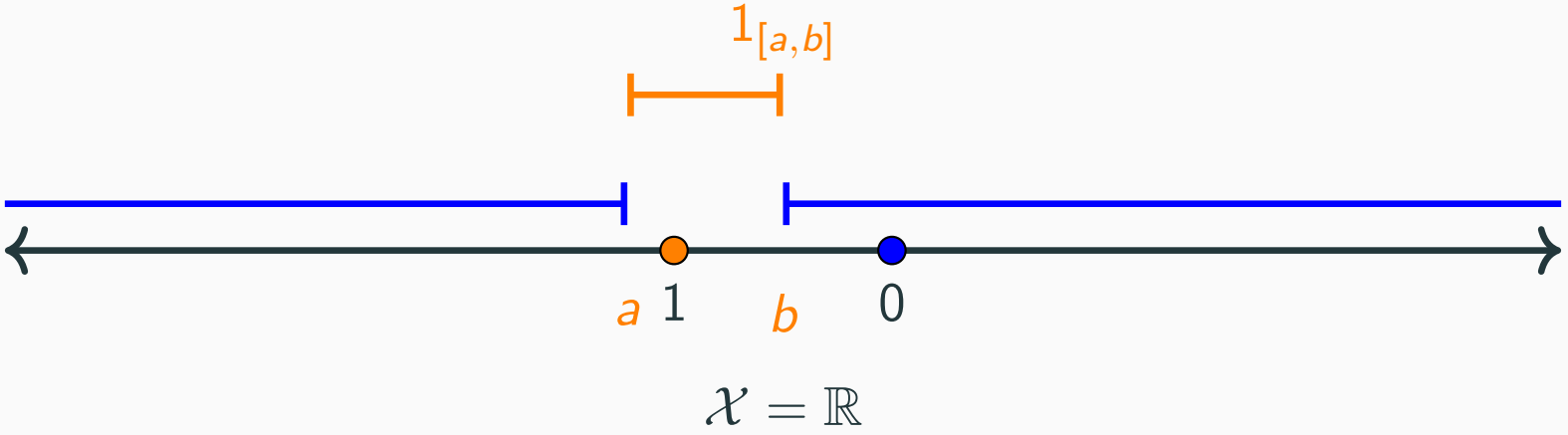
The class $\mathcal{H}$ can clearly shatter a single point. It can also shatter a two point set.



Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

The class $\mathcal{H}$ can clearly shatter a single point. It can also shatter a two point set.

Interval Classifier

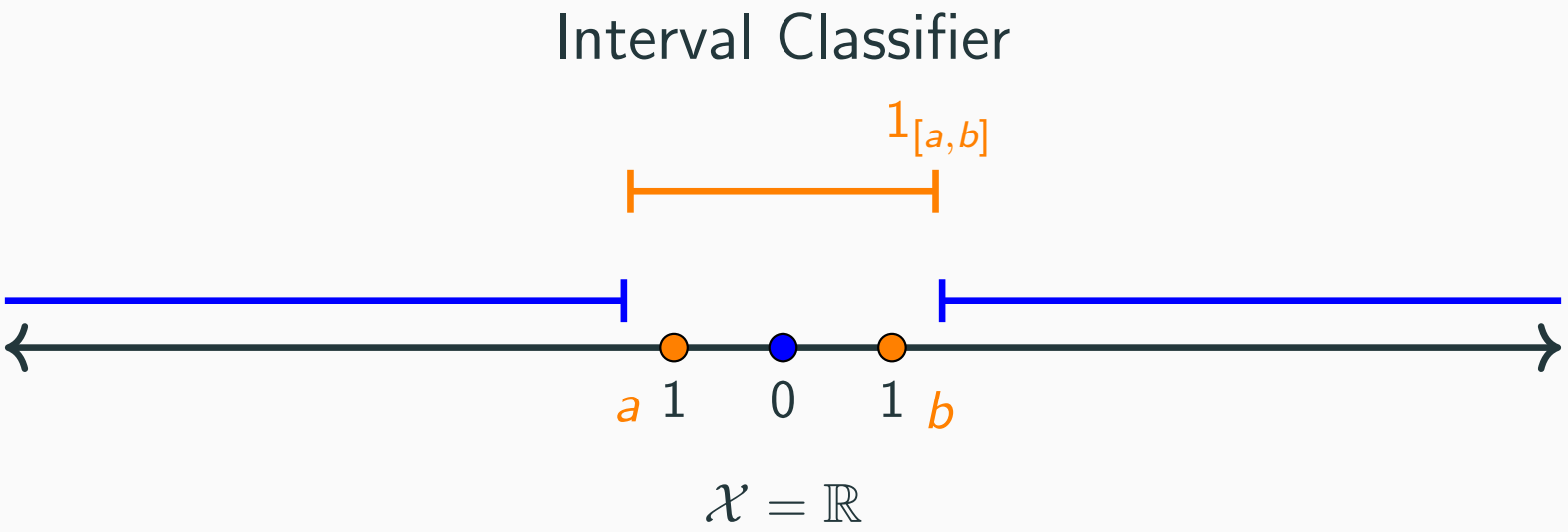


$\mathcal{X} = \mathbb{R}$

Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

The class $\mathcal{H}$ can clearly shatter a single point. It can also shatter a two point set.

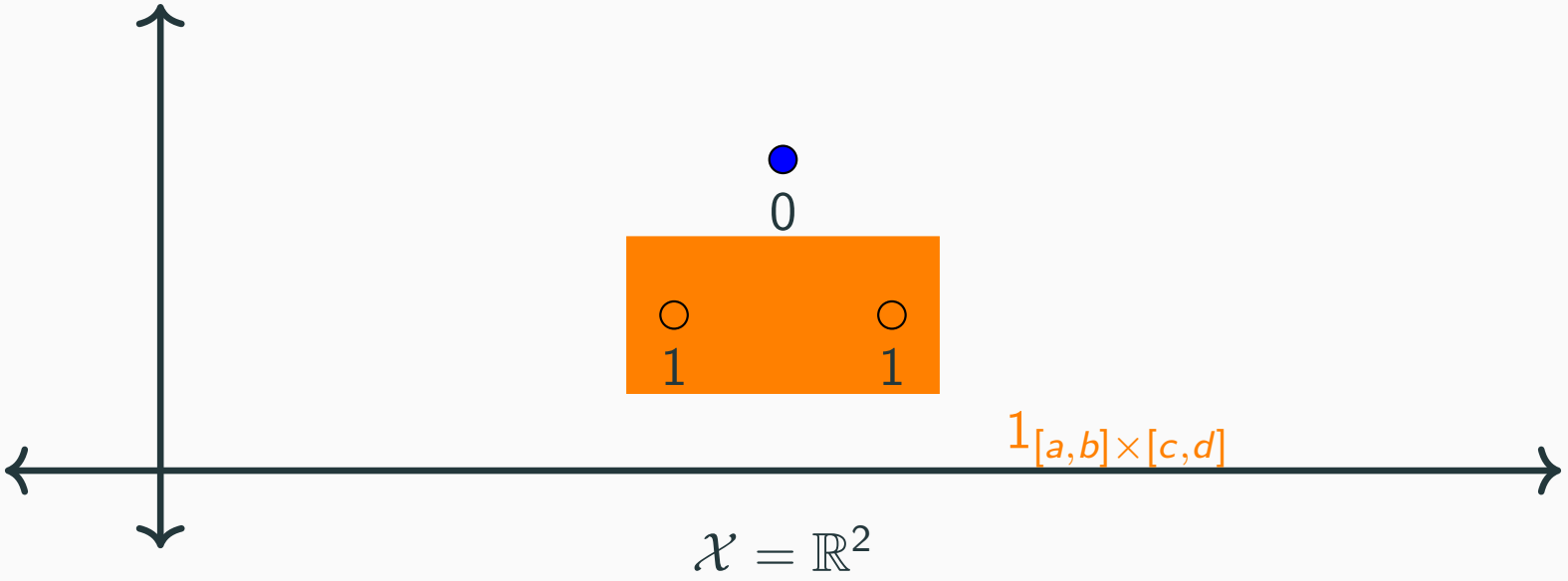
VC-Dimension



Question: What is the VC-dimension of the hypothesis class of interval classifiers $1_{[a,b]}$ on $\mathbb{R}$?

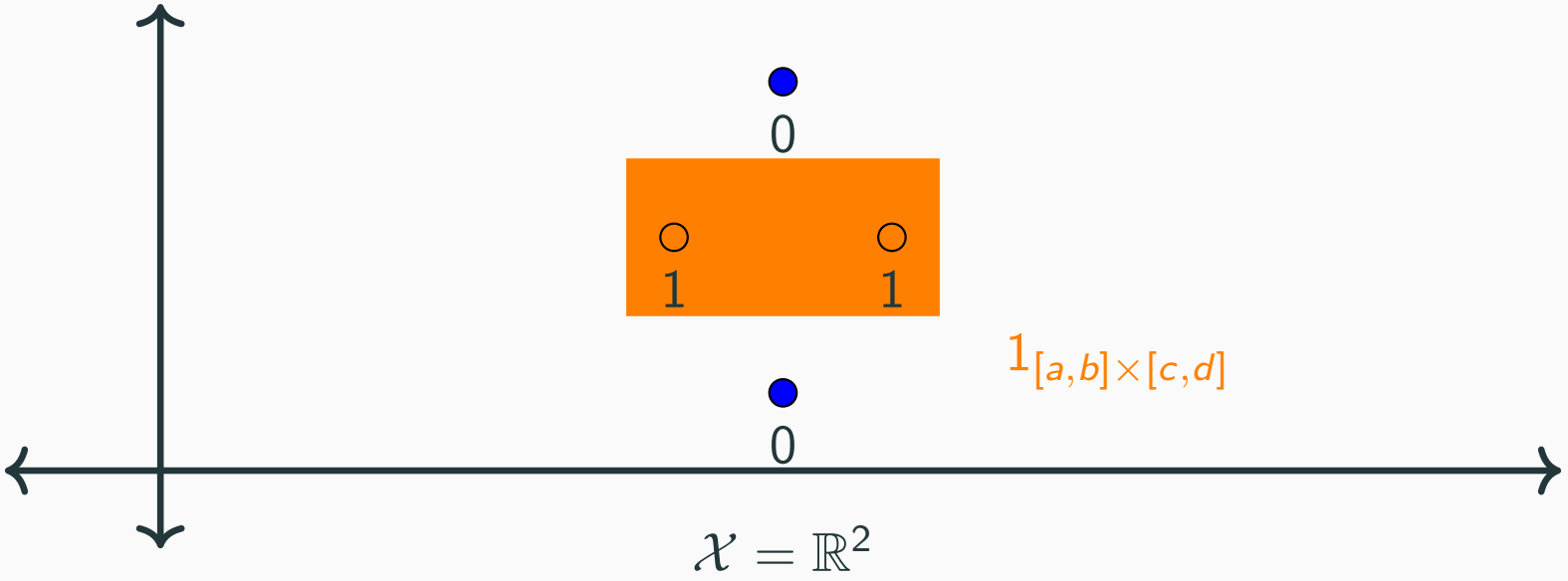
The class $\mathcal{H}$ can clearly shatter a single point. It can also shatter a two point set. But an interval classifier cannot produce every labeling on the three point set. So the VC-dimension is 3.

VC-Dimension



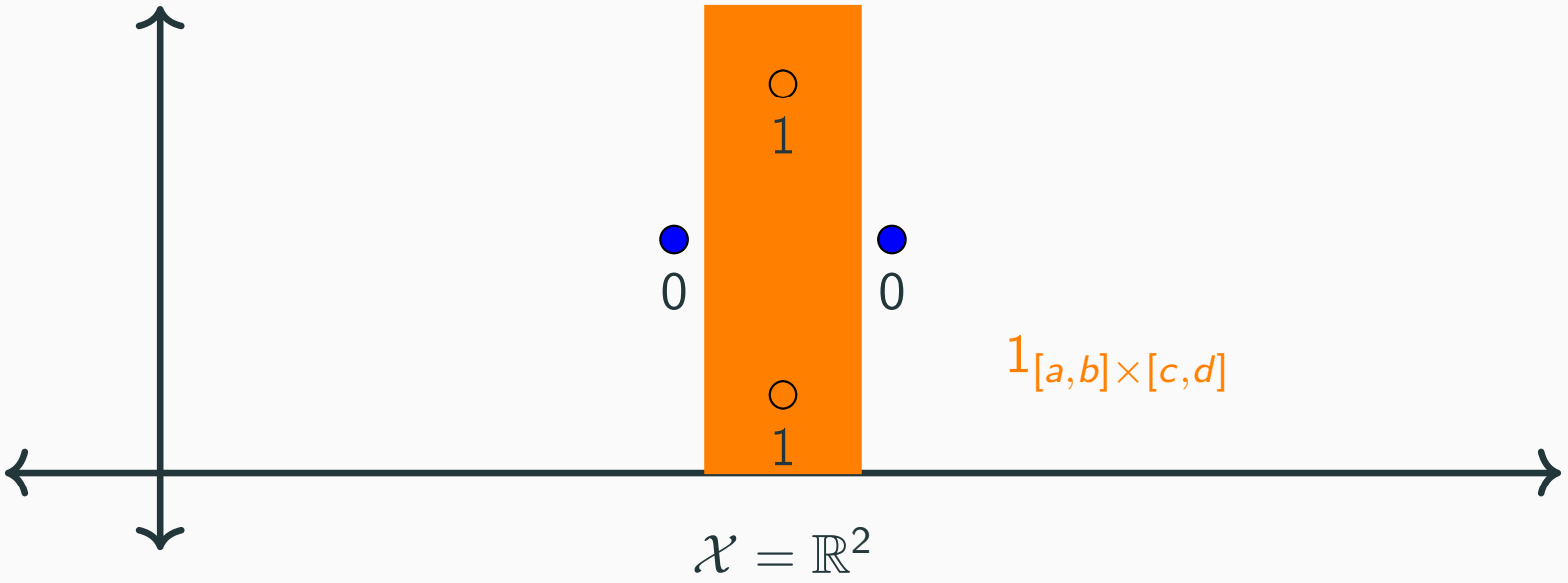
Question: What is the VC-dimension of the hypothesis class of indicator functions on the axis aligned rectangles?

VC-Dimension



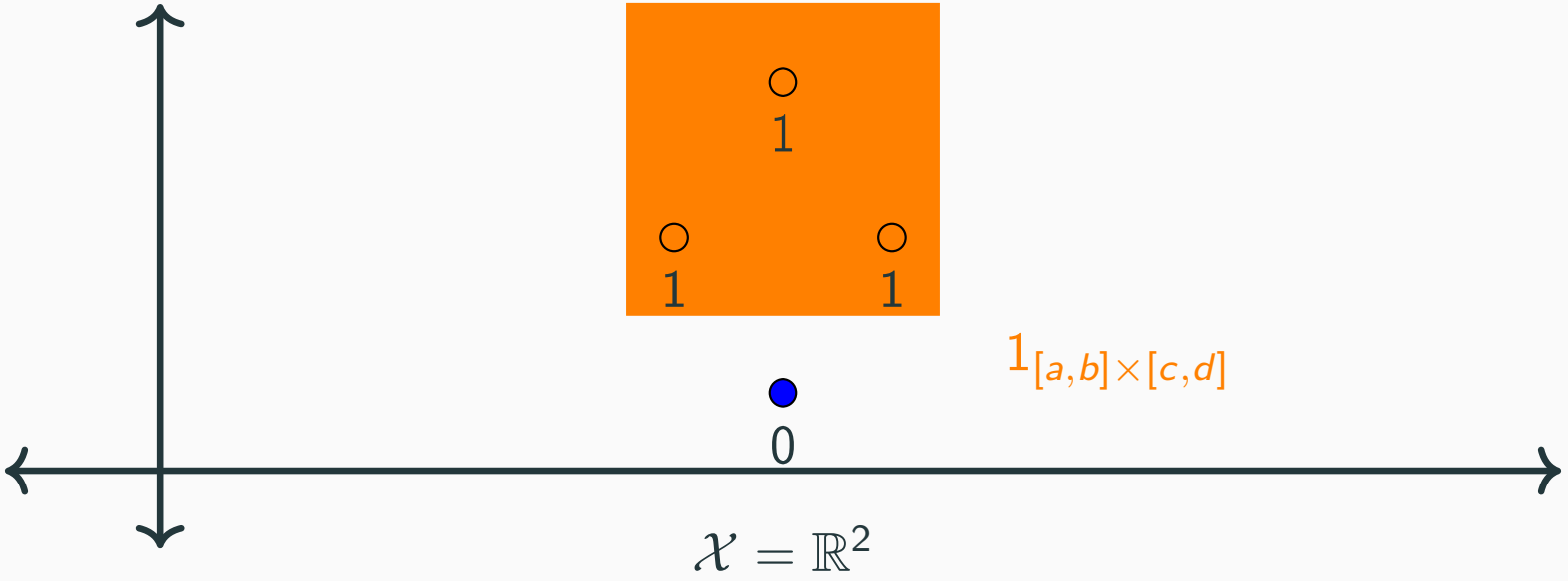
Question: What is the VC-dimension of the hypothesis class of indicator functions on the axis aligned rectangles? Its not hard to see that a 4 point see can be shattered.

VC-Dimension



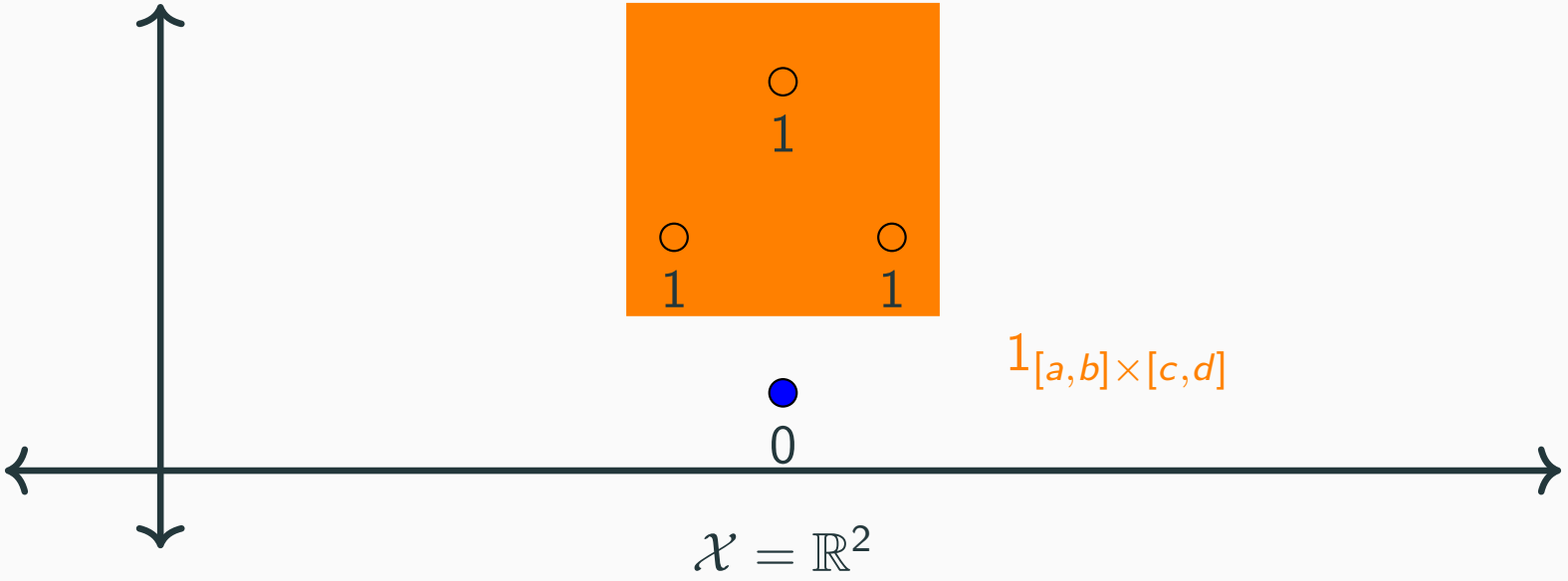
Question: What is the VC-dimension of the hypothesis class of indicator functions on the axis aligned rectangles? Its not hard to see that a 4 point see can be shattered.

VC-Dimension



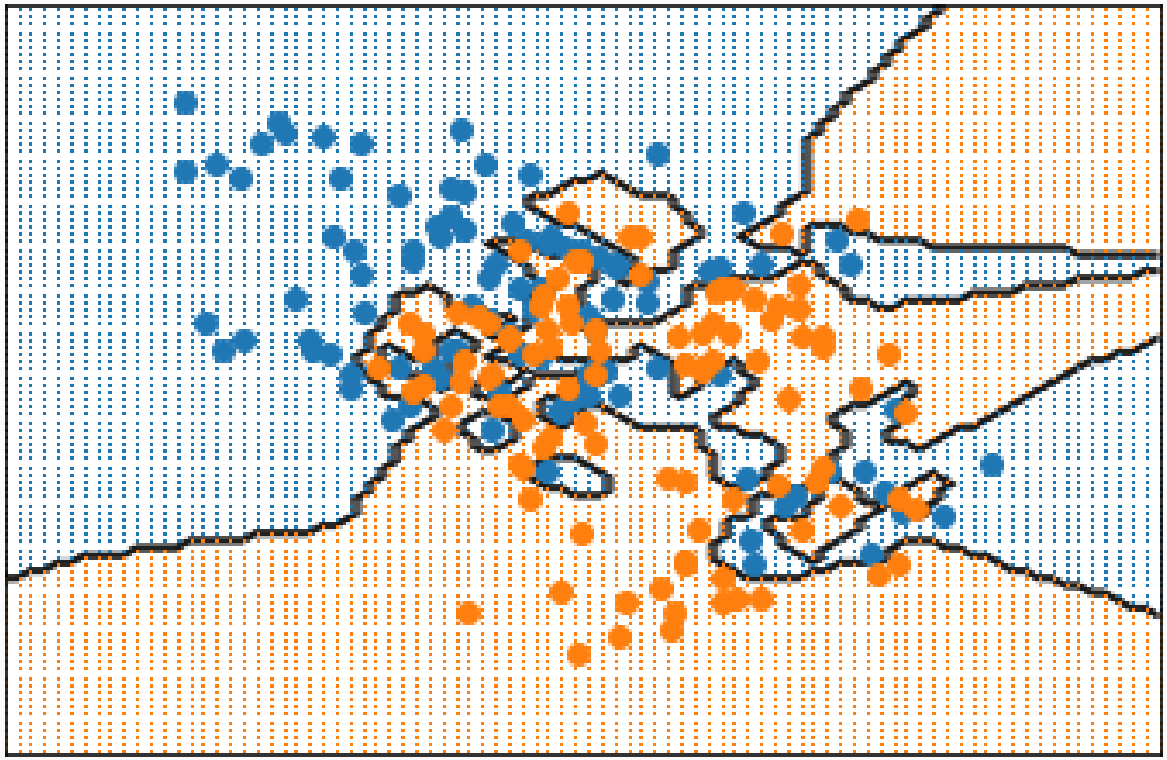
Question: What is the VC-dimension of the hypothesis class of indicator functions on the axis aligned rectangles? Its not hard to see that a 4 point see can be shattered.

VC-Dimension



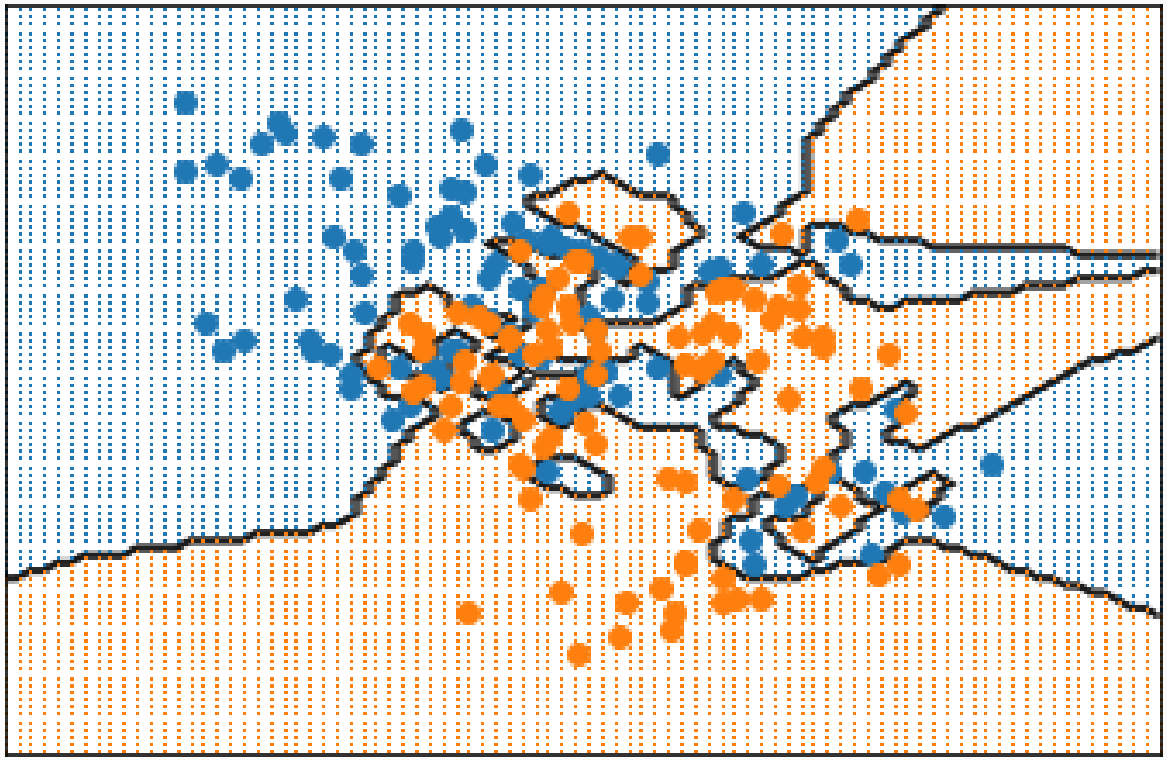
Question: What is the VC-dimension of the hypothesis class of indicator functions on the axis aligned rectangles? Its not hard to see that a 4 point see can be shattered. And it turns out there is no way to shatter any 5 point set. (**exercise**).

VC-Dimension



Question: What is the VC-Dimension of k-nearest neighbors?

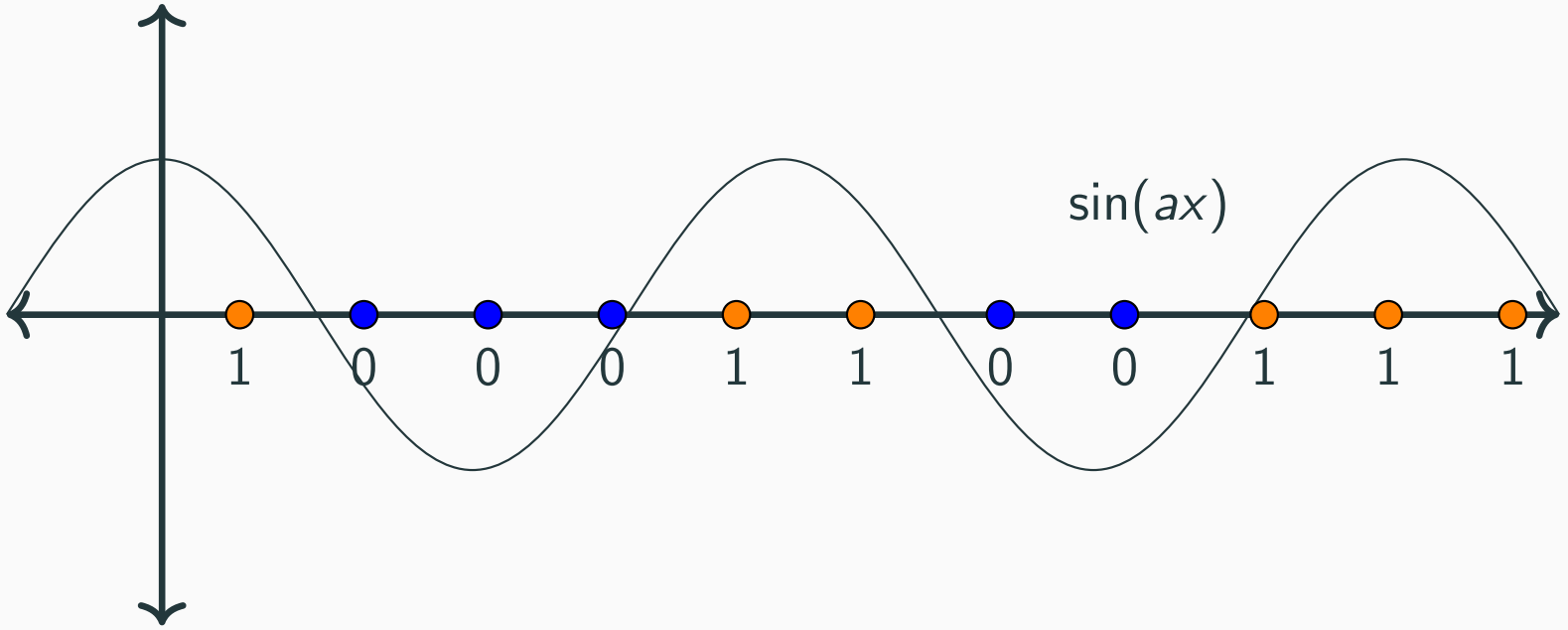
VC-Dimension



Question: What is the VC-Dimension of k-nearest neighbors?

With a little bit of thought, its clear that 1 nearest neighbors has infinite VC dimension, since when restricted to any finite set of size N, it is clearly the set of all functions. What about for higher k?

VC-Dimension and the number of parameters



Be aware: VCDim doesn't directly measure the number of parameters, but instead some "effective" number of parameters. For example, the classifier

$$\text{ceil}\big[\ \sin(ax)\ \big]$$

depends on only 1 parameter, but has infinite VC dimension.

The Fundamental Theorem of PAC Learning

Fundamental Theorem

With the concept of VC dimension, we are finally in the position to state the theorem which will tie all of PAC learning together.

Fundamental Theorem

Fundamental Theorem of Statistical Learning: Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be 0-1 loss. Then the following are equivalent:

a) $\mathcal{H}$ has the uniform convergence property.

b) Any ERM rule is a successful APAC learner for $\mathcal{H}$.

c) $\mathcal{H}$ is APAC learnable.

d) $\mathcal{H}$ is PAC learnable.

e) Any ERM rule is a successful PAC learner for $\mathcal{H}$.

f) The VC-dimension of $\mathcal{H}$ is finite.

We have already shown $a \to b \to c \to d \to e \to f$, it only remains to show $f \to a$.

Fundamental Theorem: Proof Outline

The proof is based on two main claims:

(1) First, if $\text{VCdim}(\mathcal{H}) = d$, then when restricting to finite sets the "effective size" of $\mathcal{H}$ is $O(|C|^d)$. That is, the size of $\mathcal{H}$ grows polynomially, not the exponential $2^{|C|}$ we might expect. This claim is known as Sauers Lemma.

(2) The second step is to show that if $|\mathcal{H}_C|$ has a small "effective size" then it enjoys the uniform convergence property. This is a very instructive but straightforward application of the standard probability inequity theorems like Markov and Hoeffding's. It it well worth studying, but is perhaps a better example of probability theory then machine learning.

Sauers Lemma

Growth Function

Let $\mathcal{H}$ be a hypothesis class. Then, the growth function of $\mathcal{H}$, denoted $\tau_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ is

$$\tau_{\mathcal{H}}(N) = \max_{C \subset \mathcal{X} : |C| = N} |\mathcal{H}|,$$

that is the maximum number of different functions from sets C of size N to $\{0, 1\}$.

First, if $N \leq d$, than clearly $\tau_N = 2^N$. The surprising thing is that when $N > d$, τ settles down and becomes polynomial.

Growth Function

Sauer (-Shelah-Perles) Lemma: Let $\mathcal{H}$ be a hypothesis class with $\text{VCdim}(\mathcal{H}) \leq d < \infty$. Then, for all N,

$$\tau_{\mathcal{H}}(N) \leq \sum_{i=0}^{f} \binom{m}{i}.$$

In particular, if $N > d + 1$, then

$$\tau_{\mathcal{H}}(N) \leq (eN/d)^d,$$

where $e = \exp(1)$.

So for N large enough, the effective growth of the hypothesis class over a test set is polynomial in N

Fundamental Theorem

Fundamental Theorem of Statistical Learning: Quantitative Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ with VC-dim d and let the loss function be 0-1 loss. Then there are absolute constants c_1 and c_2 such that

(1) The class $\mathcal{H}$ has the uniform convergence property and is APAC learnable, with

$$c_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq N_{\mathcal{H}}^{UC}(\epsilon, \delta), N_{\mathcal{H}}^{APAC}(\epsilon, \delta) \leq c_2 \frac{d + \log(1/\delta)}{\epsilon^2}.$$

(2) The class $\mathcal{H}$ is PAC learnable with

$$c_1 \frac{d + \log(1/\delta)}{\epsilon} \leq N_{\mathcal{H}}(\epsilon, \delta) \leq c_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}.$$

So the VC dimension, if it can be computed, tells us exactly how the sample complexity depends on the accuracy and precision.

Fundamental Theorem

Fundamental Theorem of Statistical Learning: Quantitative Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ with VC-dim d and let the loss function be 0-1 loss. Then there are absolute constants c_1 and c_2 such that

(1) The class $\mathcal{H}$ has the uniform convergence property and is APAC learnable, with

$$c_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq N_{\mathcal{H}}^{UC}(\epsilon, \delta), N_{\mathcal{H}}^{APAC}(\epsilon, \delta) \leq c_2 \frac{d + \log(1/\delta)}{\epsilon^2}.$$

(2) The class $\mathcal{H}$ is PAC learnable with

$$c_1 \frac{d + \log(1/\delta)}{\epsilon} \leq N_{\mathcal{H}}(\epsilon, \delta) \leq c_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}.$$

Note: If $\mathcal{H}$ is realizable accuracy is much cheaper, bu still more expensive than percistion.