Computational Learning Theory: Shattering and VC Dimensions

Machine Learning

THE UNIVERSITY OF UTAH

Slides based on material from Dan Roth, Avrim Blum, Tom Mitchell and others

This lecture: Computational Learning Theory

- The Theory of Generalization
- Probably Approximately Correct (PAC) learning
- Positive and negative learnability results
- Agnostic Learning
- Shattering and the VC dimension

This lecture: Computational Learning Theory

- The Theory of Generalization
- Probably Approximately Correct (PAC) learning
- Positive and negative learnability results
- Agnostic Learning
- Shattering and the VC dimension

What have we seen so far

If a learner explores a finite hypothesis space and…

1. …guarantees a hypothesis that is consistent with a training set: Occam’s razor for a consistent learner
2. …does not guarantee a consistent hypothesis: Agnostic learning and an Occam’s razor

In both cases, the sample complexity depends on the size of the hypothesis space

What if the hypothesis space is infinite?

Infinite Hypothesis Space

- The previous analysis was restricted to finite hypothesis spaces
- Some infinite hypothesis spaces are more expressive than others
  - E.g., Rectangles, vs. 17- sides convex polygons vs. general convex polygons
  - Linear threshold function vs. a combination of LTUs

Infinite Hypothesis Space

- The previous analysis was restricted to finite hypothesis spaces
- Some infinite hypothesis spaces are more expressive than others
  - E.g., Rectangles, vs. 17- sides convex polygons vs. general convex polygons
  - Linear threshold function vs. a combination of LTUs
- Need a measure of the expressiveness of an infinite hypothesis space other than its size

Infinite Hypothesis Space

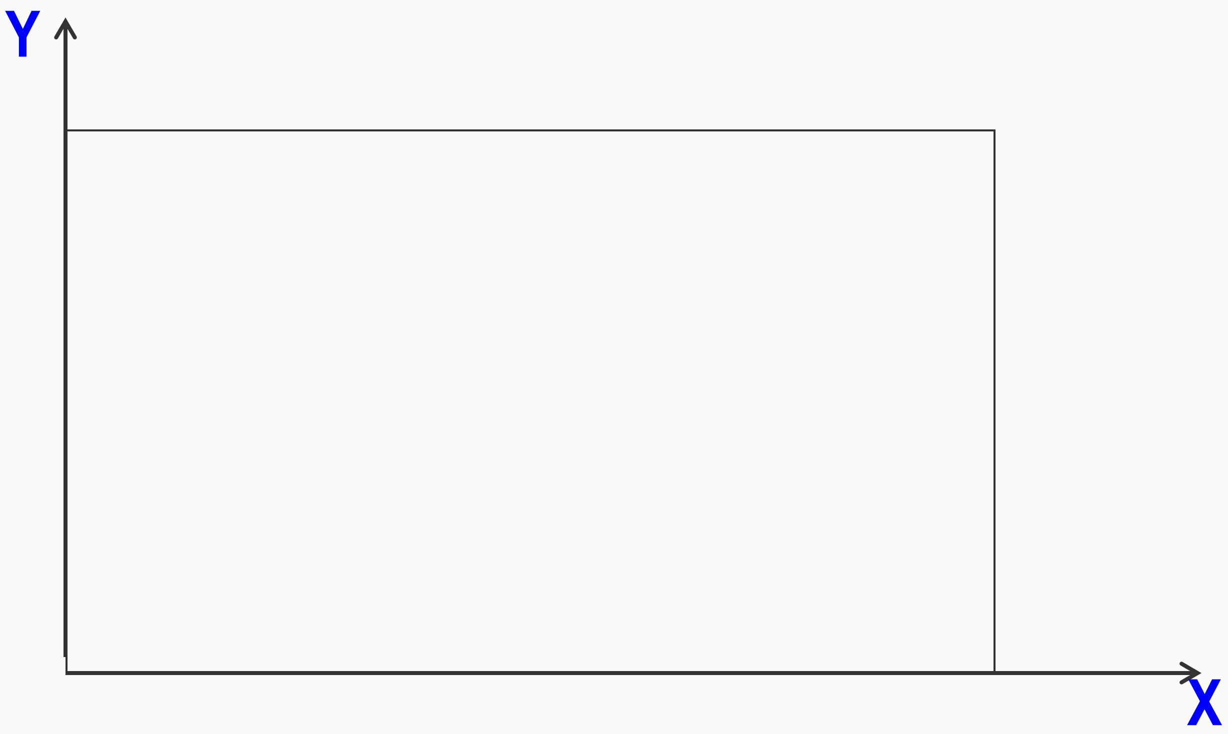
- The previous analysis was restricted to finite hypothesis spaces
- Some infinite hypothesis spaces are more expressive than others
  - E.g., Rectangles, vs. 17- sides convex polygons vs. general convex polygons
  - Linear threshold function vs. a combination of LTUs
- Need a measure of the expressiveness of an infinite hypothesis space other than its size
- The Vapnik-Chervonenkis dimension (VC dimension) provides such a measure
  - "What is the expressive *capacity* of a set of functions?"

Infinite Hypothesis Space

- The previous analysis was restricted to finite hypothesis spaces
- Some infinite hypothesis spaces are more expressive than others
  - E.g., Rectangles, vs. 17- sides convex polygons vs. general convex polygons
  - Linear threshold function vs. a combination of LTUs
- Need a measure of the expressiveness of an infinite hypothesis space other than its size
- The Vapnik-Chervonenkis dimension (VC dimension) provides such a measure
  - "What is the expressive *capacity* of a set of functions?"
- Analogous to $|H|$, there are bounds for sample complexity using $VC(H)$

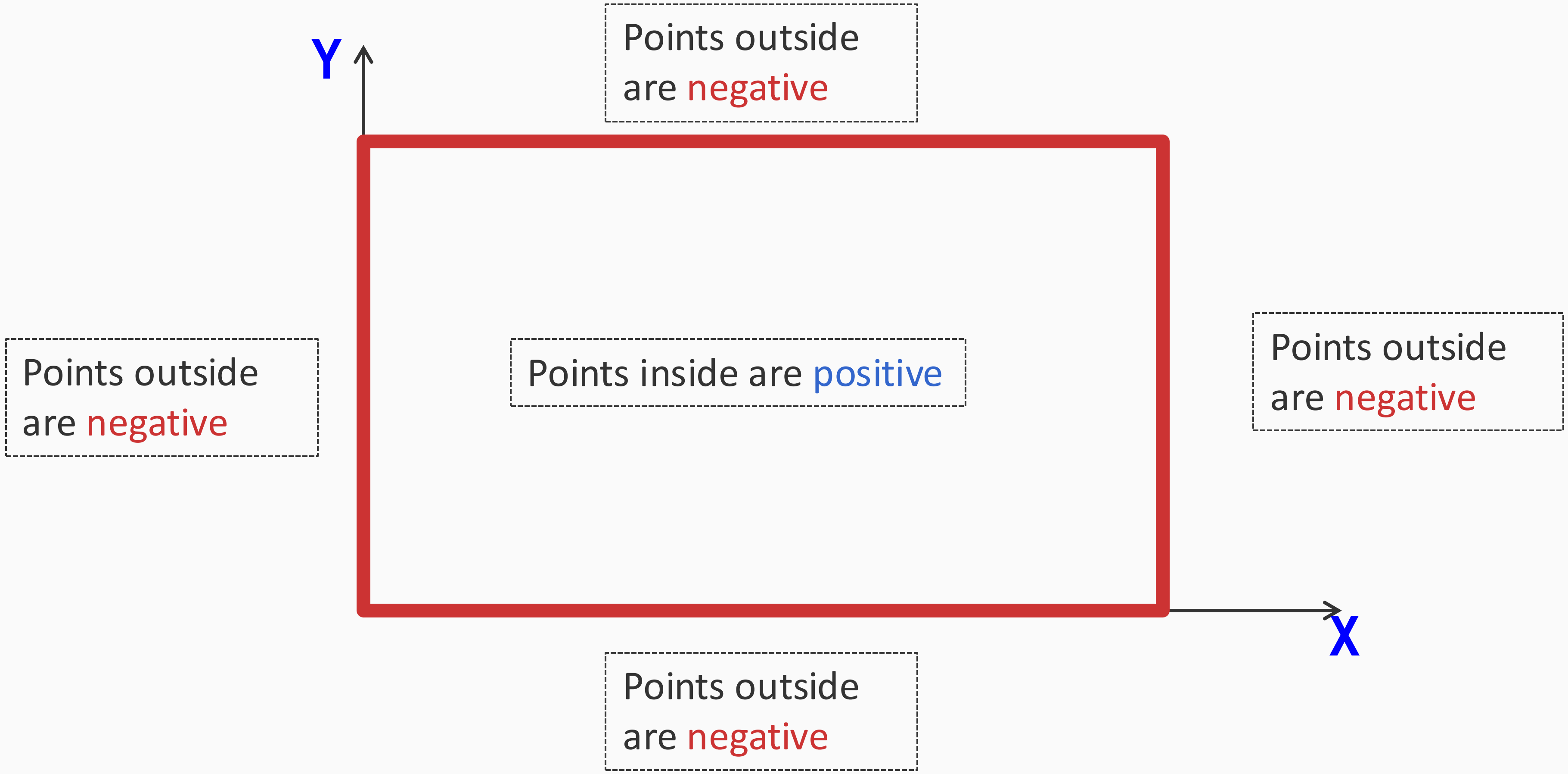
Learning Rectangles

Assume the target concept is an axis parallel rectangle



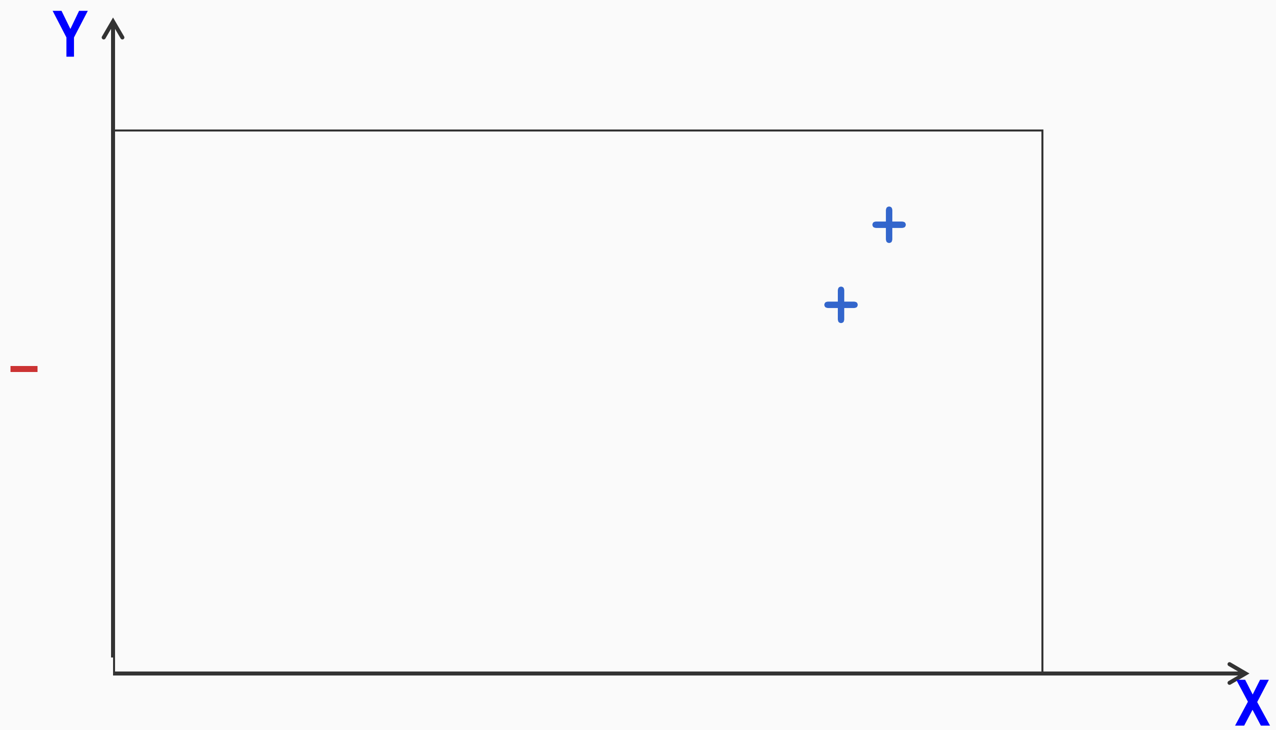
Learning Rectangles

Assume the target concept is an axis parallel rectangle



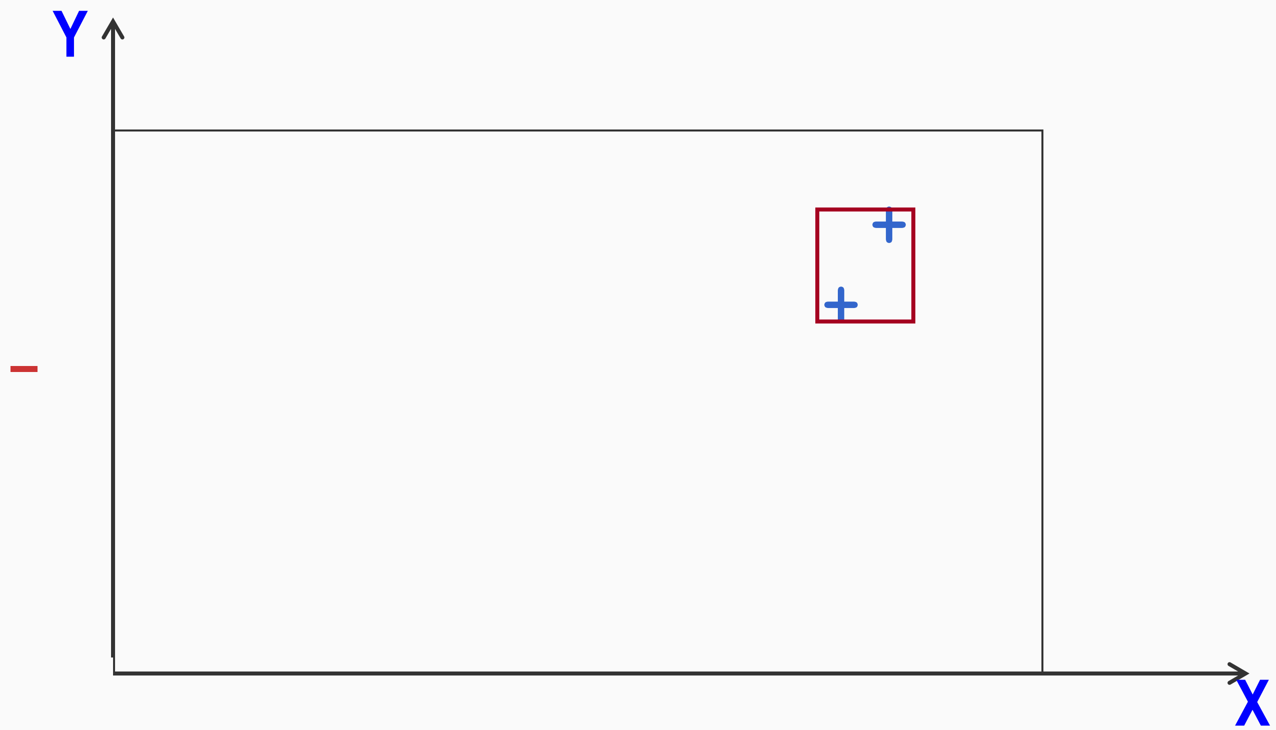
Learning Rectangles

Assume the target concept is an axis parallel rectangle



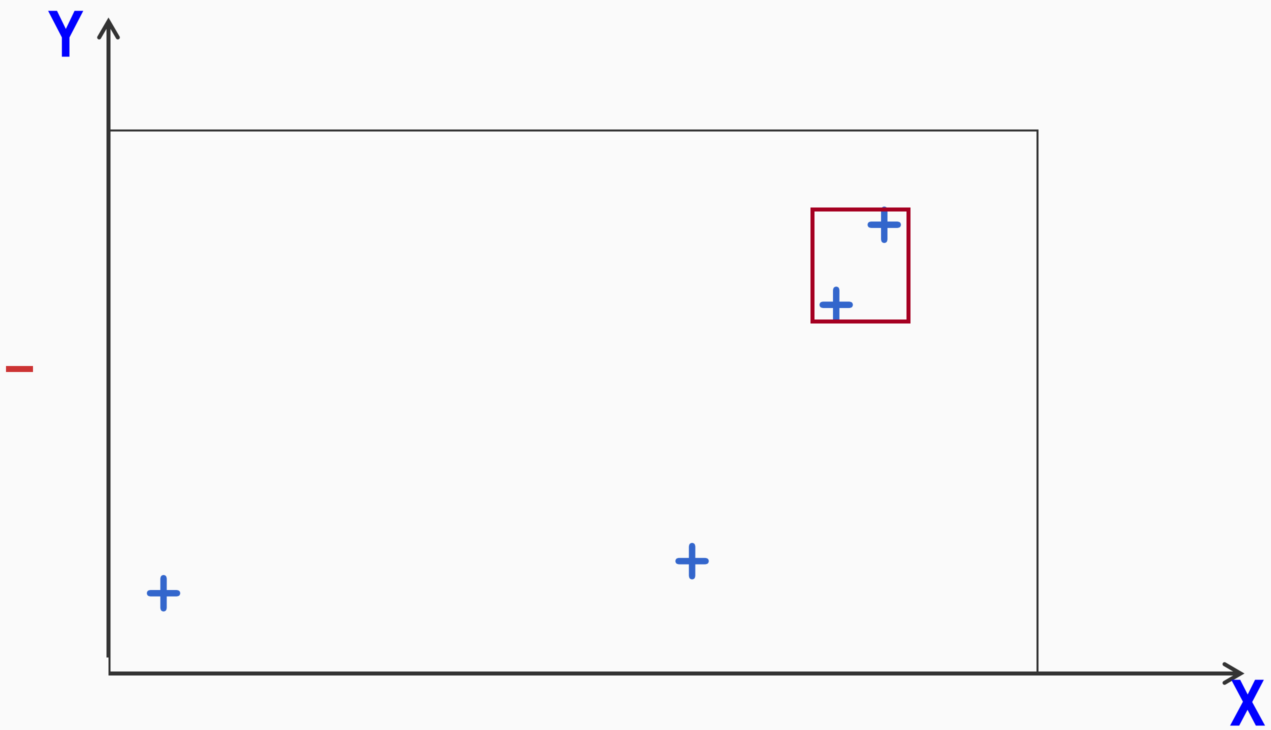
Learning Rectangles

Assume the target concept is an axis parallel rectangle



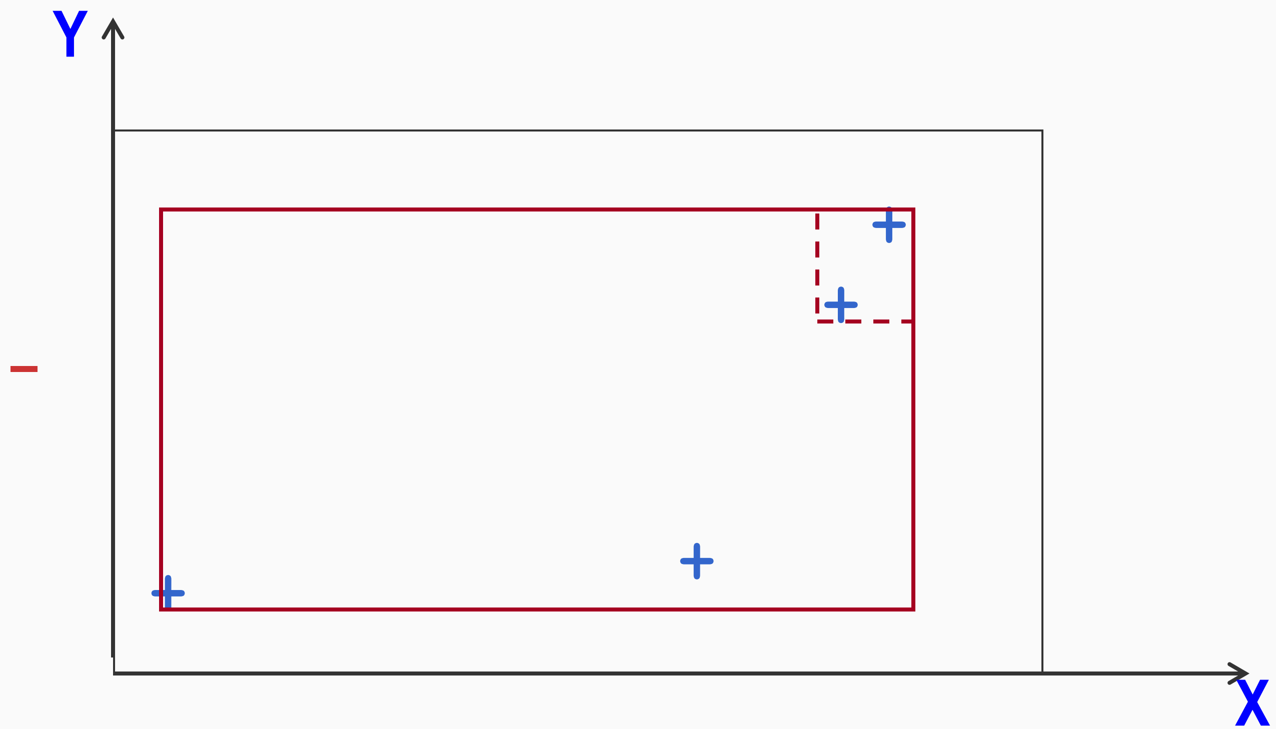
Learning Rectangles

Assume the target concept is an axis parallel rectangle



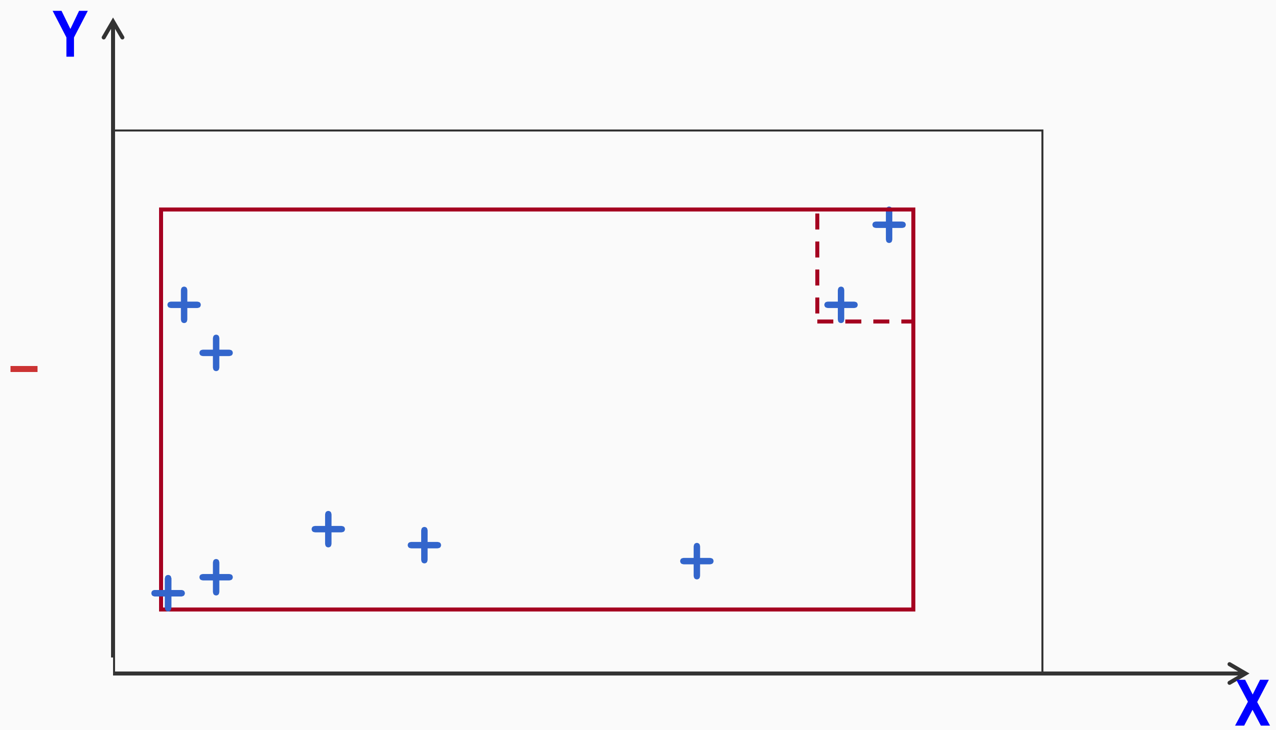
Learning Rectangles

Assume the target concept is an axis parallel rectangle



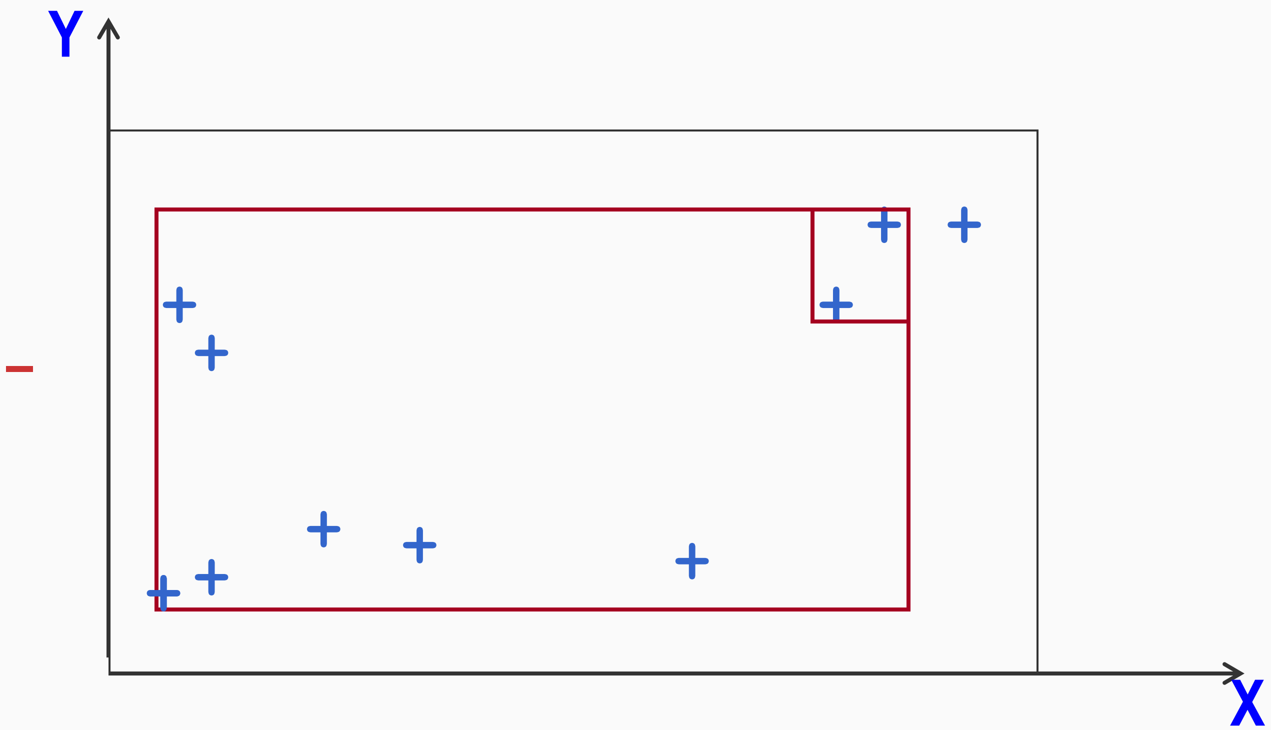
Learning Rectangles

Assume the target concept is an axis parallel rectangle



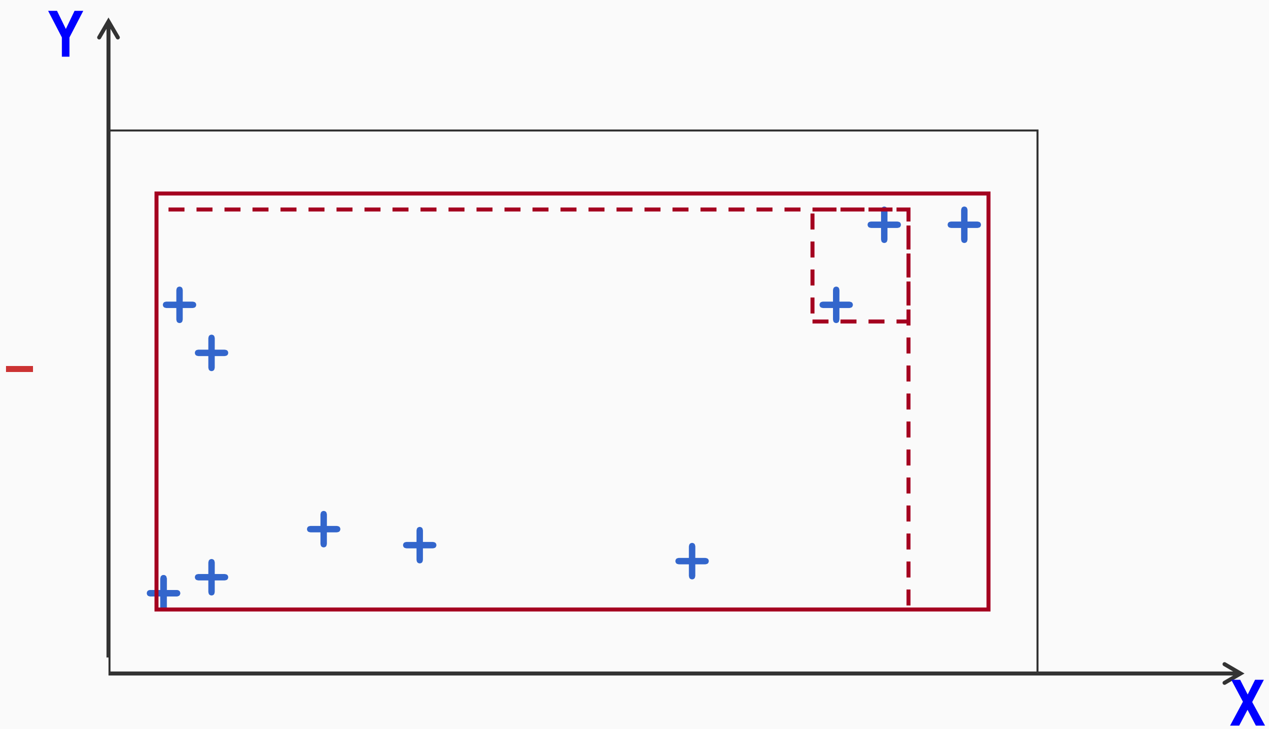
Learning Rectangles

Assume the target concept is an axis parallel rectangle



Learning Rectangles

Assume the target concept is an axis parallel rectangle



Will we be able to learn the target rectangle?

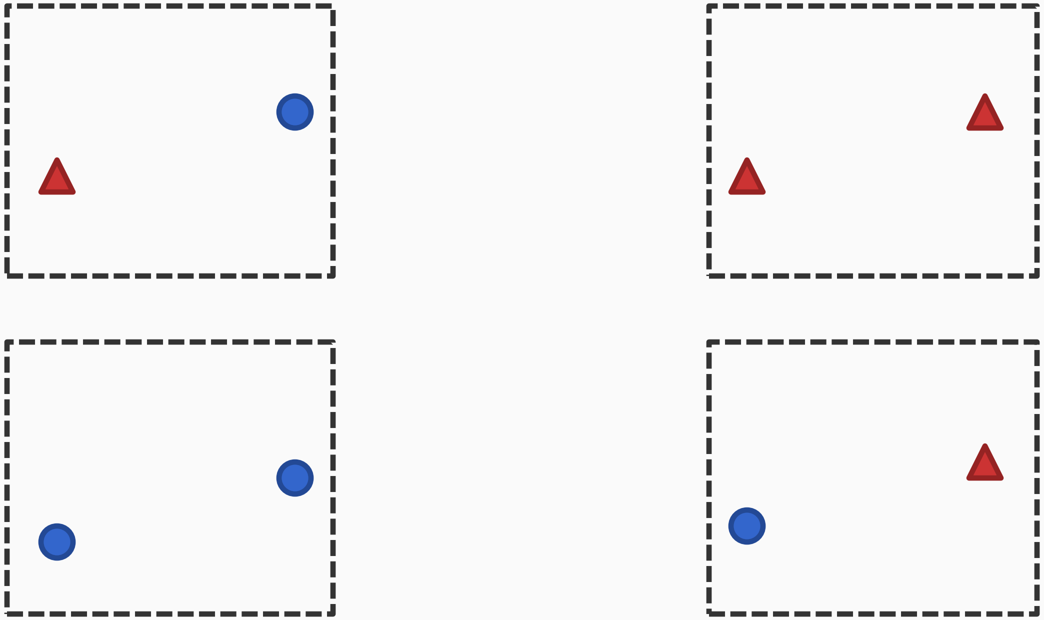
Can we come close?

Let’s think about expressivity of functions

Suppose we have two points.

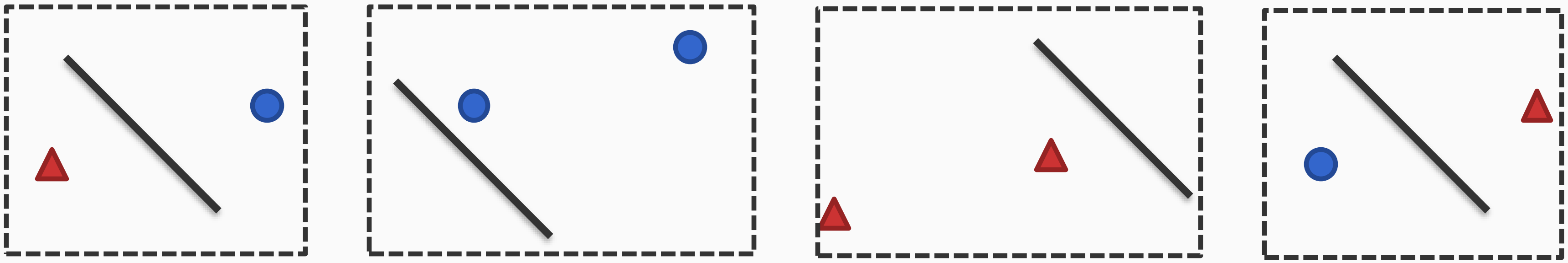
Can linear classifiers correctly classify any labeling of these points?

Let’s think about expressivity of functions



There are four ways to label two points

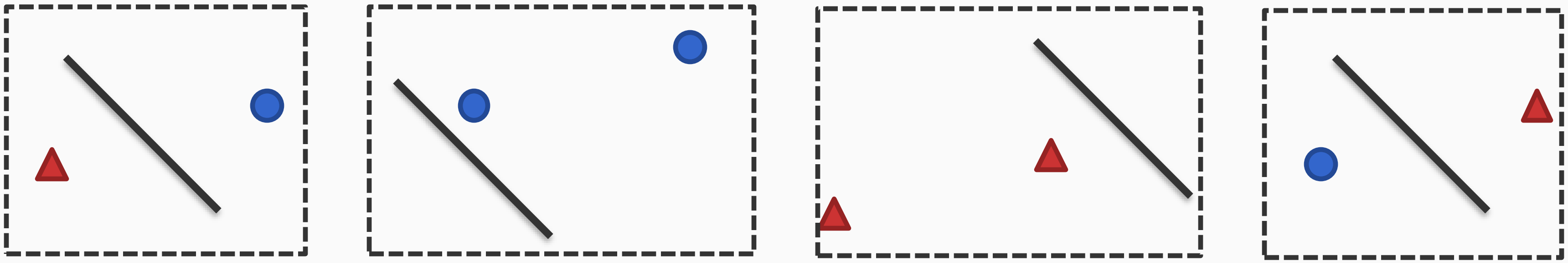
Let's think about expressivity of functions



There are four ways to label two points

And it is possible to draw a line that separates positive and negative points in all four cases

Let’s think about expressivity of functions

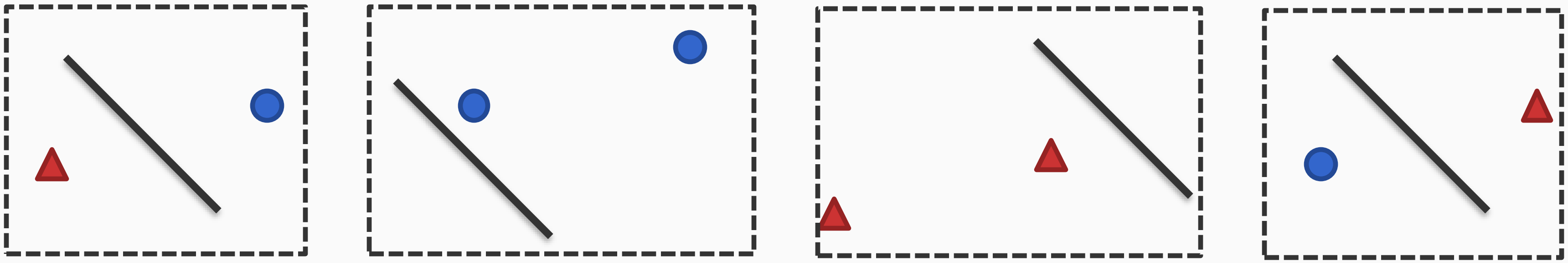


There are four ways to label two points

And it is possible to draw a line that separates positive and negative points in all four cases

We say that linear functions are expressive enough to ***shatter*** two points

Let's think about expressivity of functions



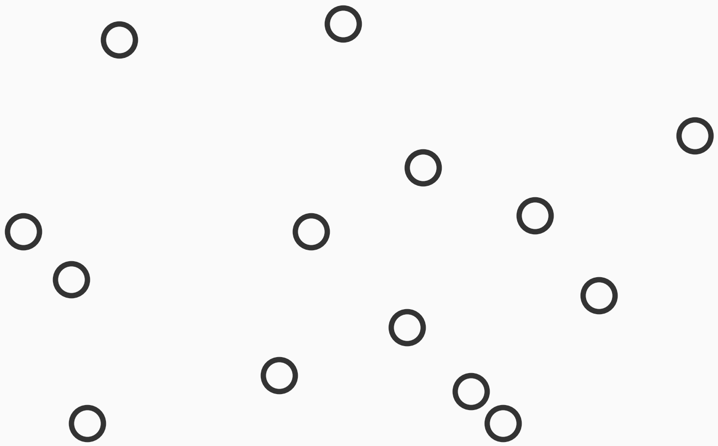
There are four ways to label two points

And it is possible to draw a line that separates positive and negative points in all four cases

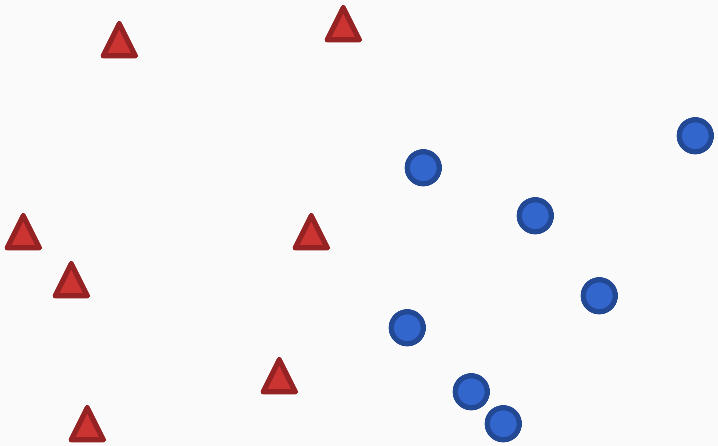
We say that linear functions are expressive enough to ***shatter*** two points

What about fourteen points?

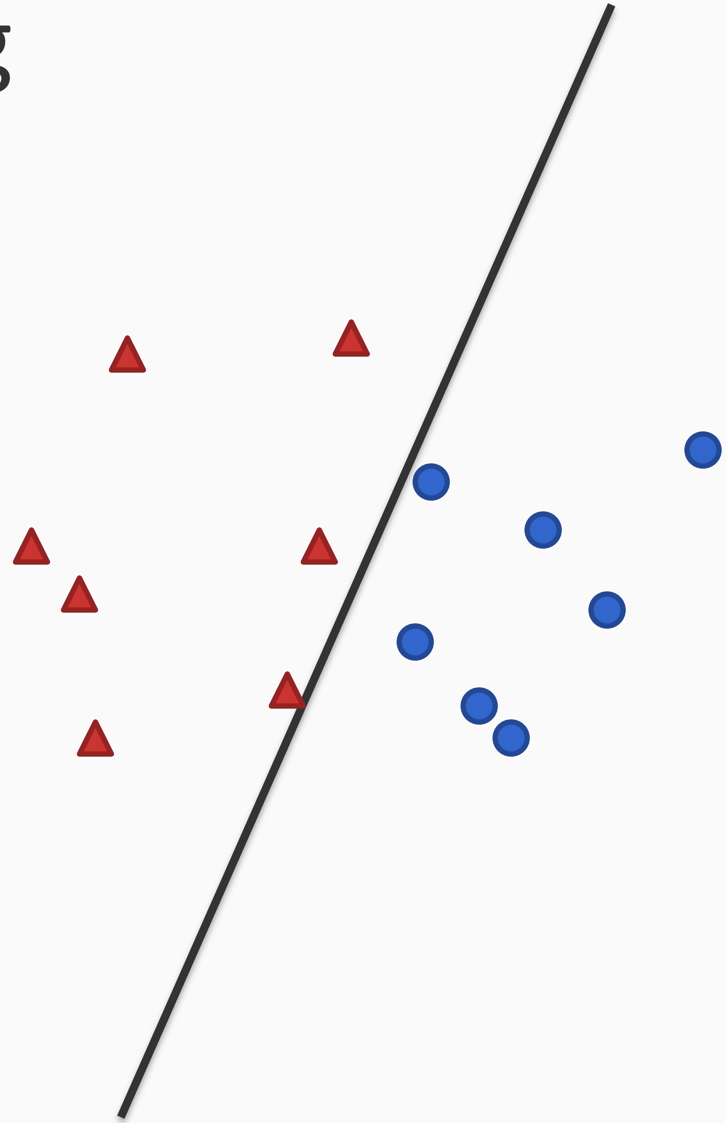
Shattering



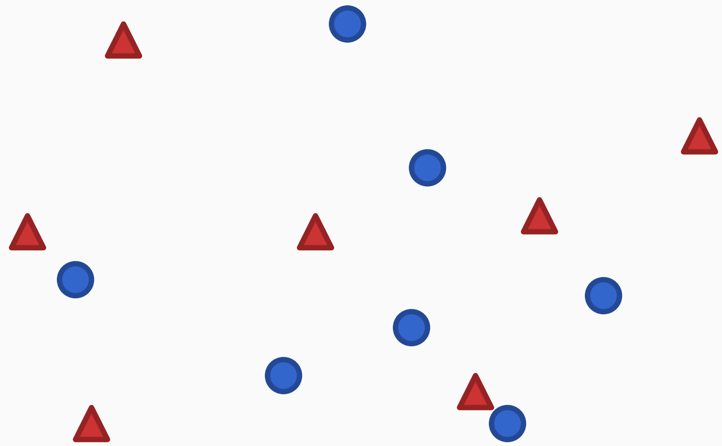
Shattering



Shattering

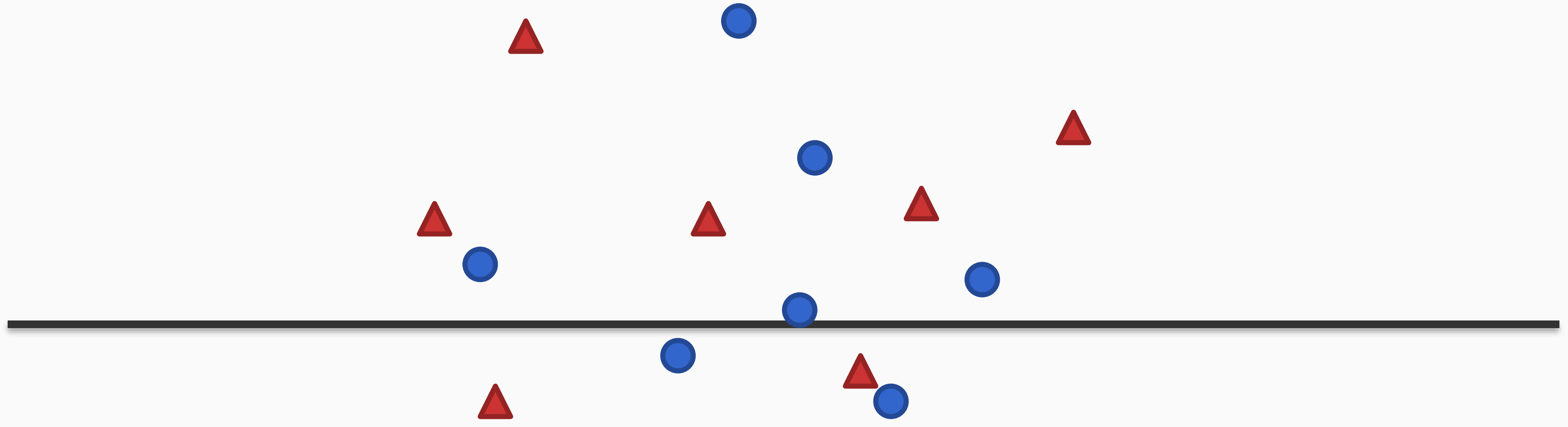


Shattering



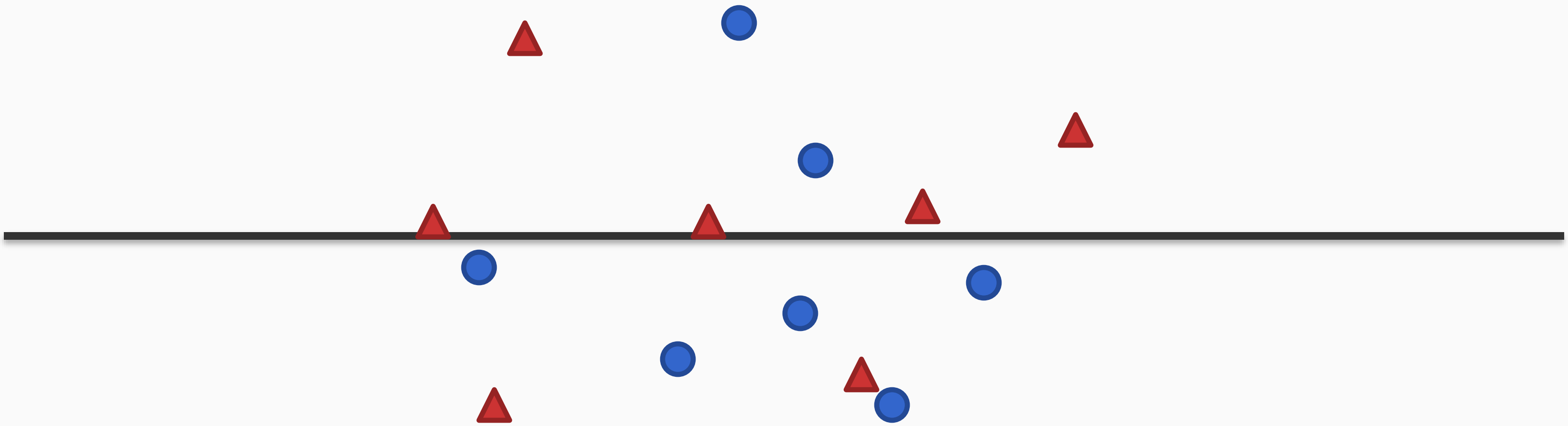
What about this labeling?

Shattering



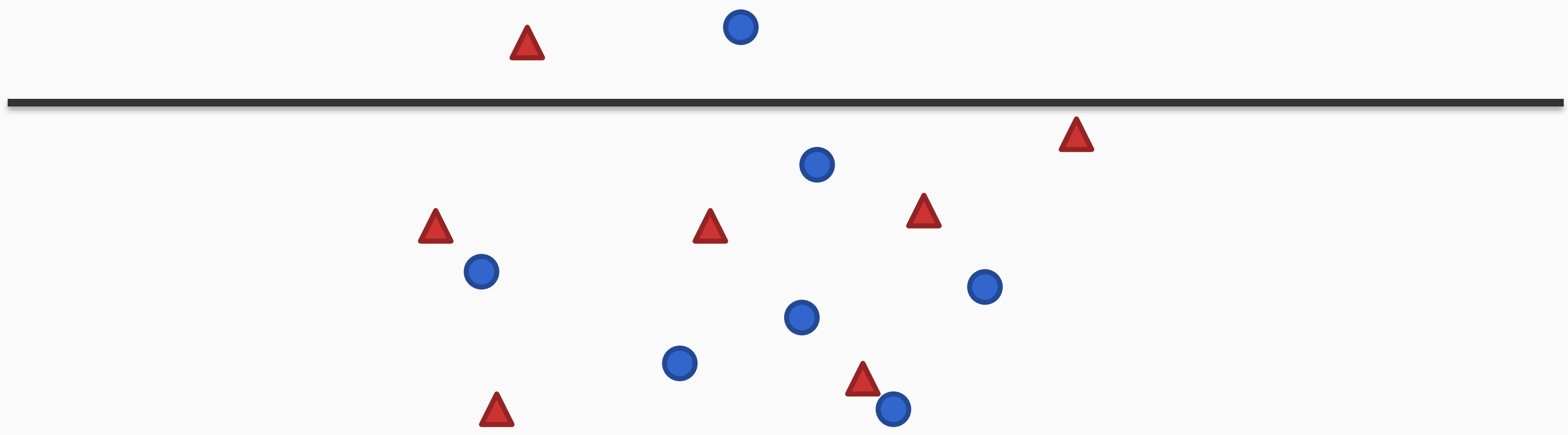
This particular labeling of the points cannot be separated by *any* line

Shattering



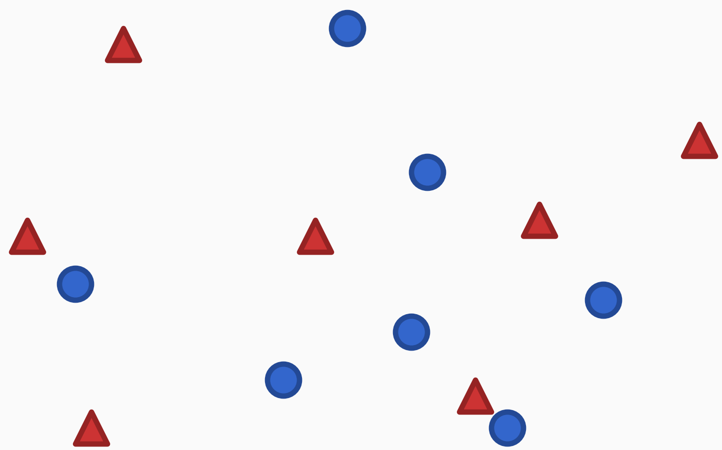
This particular labeling of the points cannot be separated by *any* line

Shattering



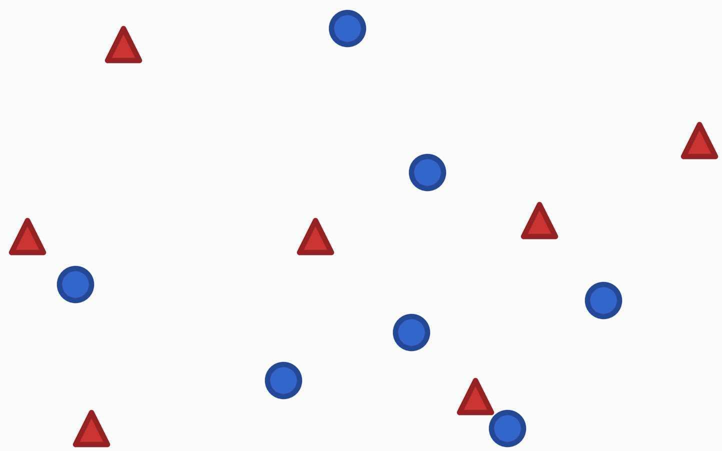
This particular labeling of the points cannot be separated by *any* line

Shattering



This particular labeling of the points cannot be separated by *any* line

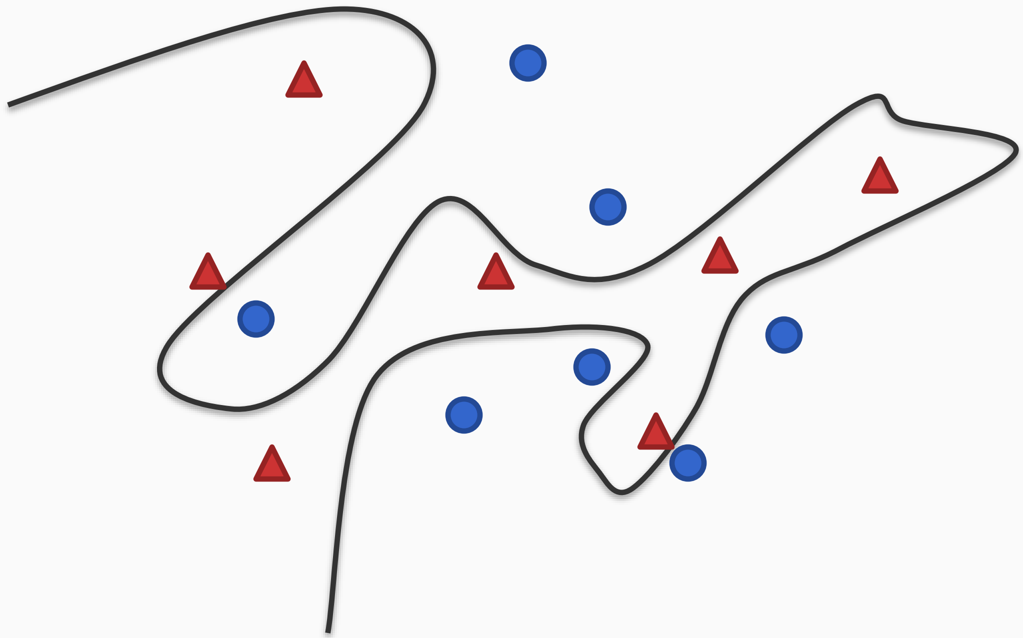
Shattering



Linear functions are not expressive enough to shatter fourteen points

Because there is *at least one labeling* that can not be separated by them

Shattering



Linear functions are not expressive enough to shatter fourteen points

Because there is *at least one labeling* that can not be separated by them

Of course, a more complex function could separate them

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

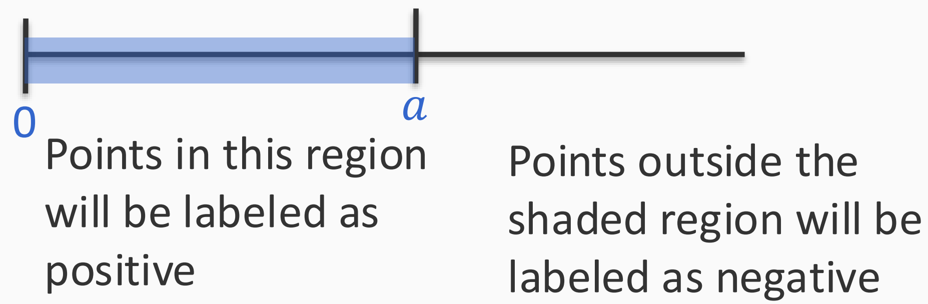
Intuition: A rich set of functions shatters large sets of points

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Intuition: A rich set of functions shatters large sets of points

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

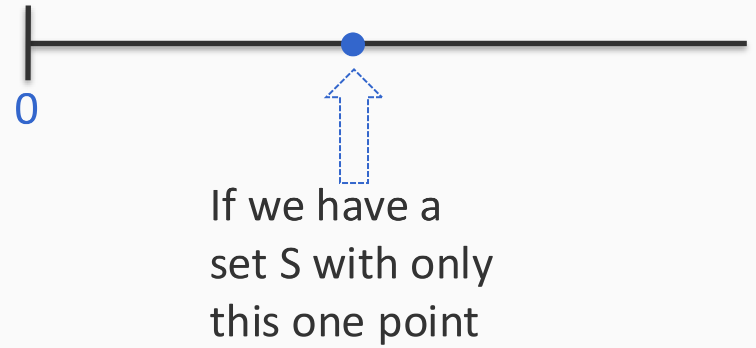


Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

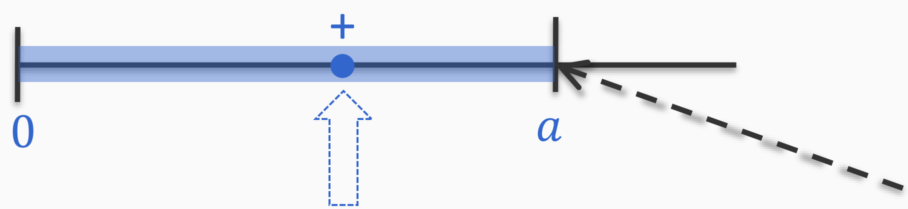
Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0



Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0



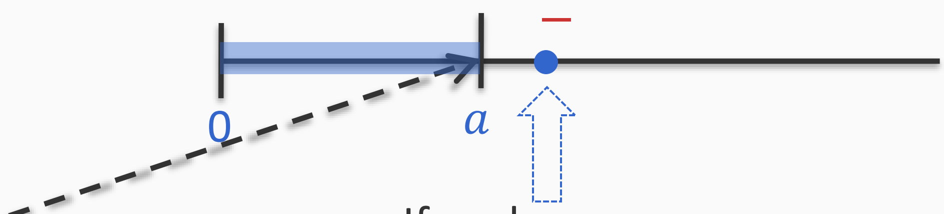
If we have a set S with only this one point

If the point is labeled +, we can find an a that is to the right of that point

This hypothesis correctly labels the point as positive

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0



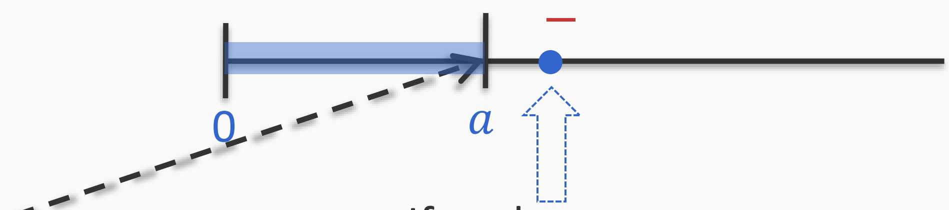
If we have a set S with only this one point

If the point is labeled −, we can find an a that is to the right of that point

This hypothesis correctly labels the point as negative

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0



If we have a set S with only this one point

If the point is labeled −, we can find an a that is to the right of that point

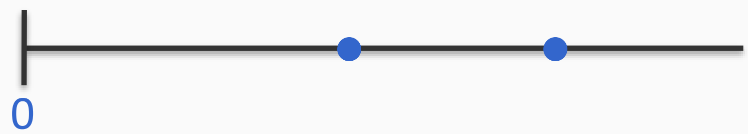
This hypothesis correctly labels the point as negative

Any set of **one** point can be shattered by the hypothesis class of left bounded intervals

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points

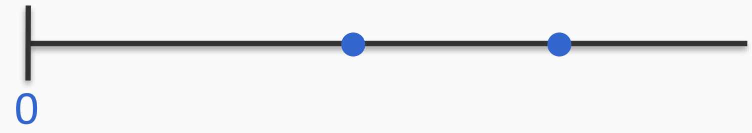


If we have a set S with these two points

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points



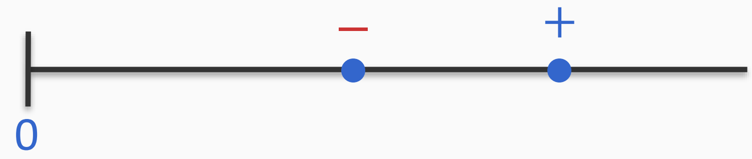
If we have a set S with these two points

We can label the points such that no hypothesis in our class can match the labels

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points



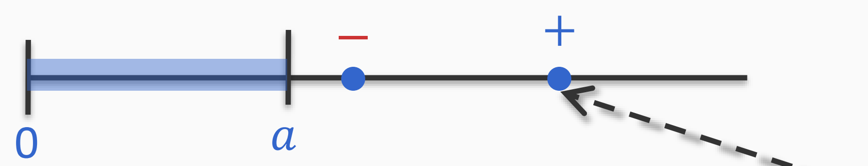
If we have a set S with these two points

We can label the points such that no hypothesis in our class can match the labels

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points



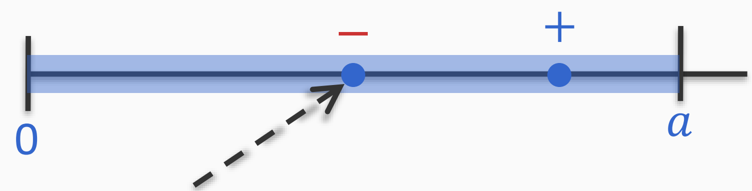
Incorrectly labels this point as negative

We can label the points such that no hypothesis in our class can match the labels

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points



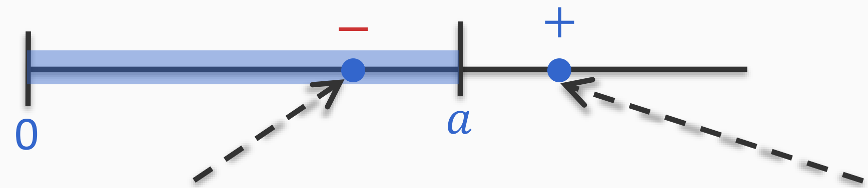
Incorrectly labels this point as positive

We can label the points such that no hypothesis in our class can match the labels

Left bounded intervals

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Let us consider a set with two points



Incorrectly labels this point as positive

Incorrectly labels this point as negative

We can label the points such that no hypothesis in our class can match the labels

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Intuition: A rich set of functions shatters large sets of points

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Intuition: A rich set of functions shatters large sets of points

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Sets with **one** point can be shattered

That is: Given one point, **for any** labeling of the points, we can find a concept in this class that is consistent with it

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Intuition: A rich set of functions shatters large sets of points

Example 1: Hypothesis class of left bounded intervals on the real axis: [0,a) for some real number a>0

Sets with **one** point can be shattered

That is: Given one point, **for any** labeling of the points, we can find a concept in this class that is consistent with it

Sets with **two** points cannot be shattered

That is: given two points, it is **possible** to label them in such a way that no concept in this class will be consistent with their labeling

Shattering

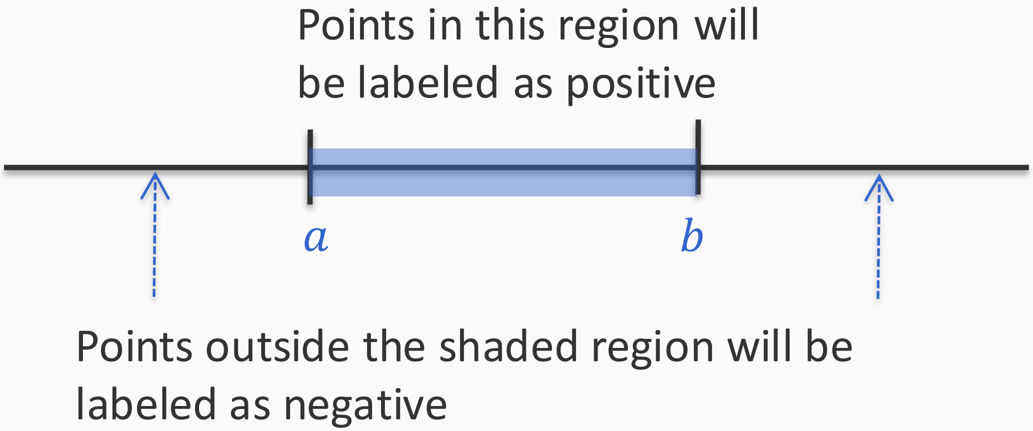
Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Example 2: Hypothesis class is the set of intervals on the real axis: [a,b],for some real numbers b>a

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Example 2: Hypothesis class is the set of intervals on the real axis: [a,b],for some real numbers b>a



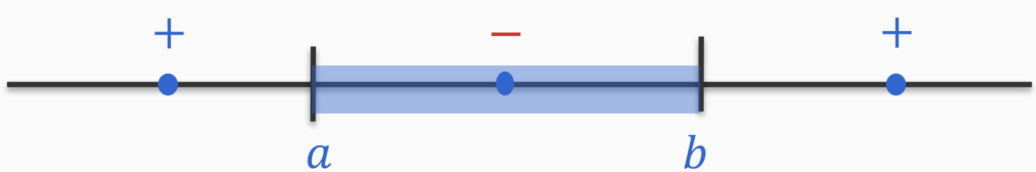
Real intervals

Example 2: Hypothesis class is the set of intervals on the real axis: [a,b],for some real numbers b>a

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

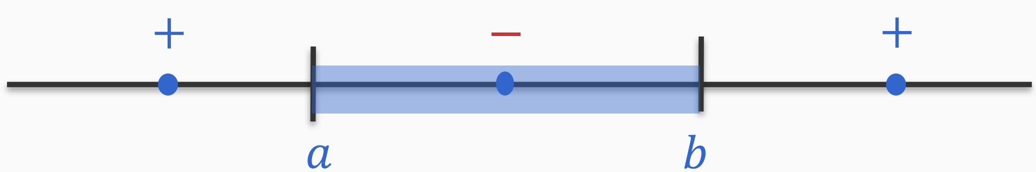
Example 2: Hypothesis class is the set of intervals on the real axis: [a,b],for some real numbers b>a



Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Example 2: Hypothesis class is the set of intervals on the real axis: [a,b],for some real numbers b>a



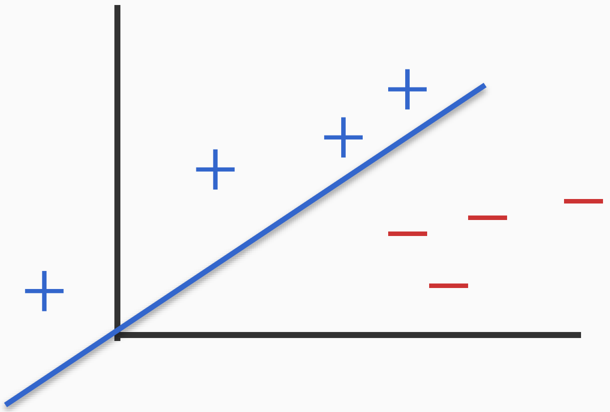
All sets of one or two points can be shattered
But sets of three points cannot be shattered

Proof? Enumerate all possible three points

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

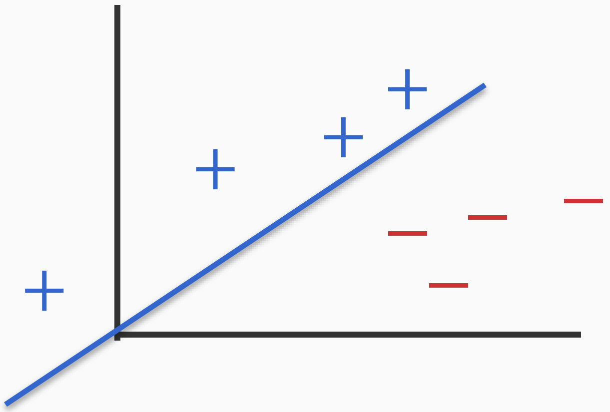
Example 3: Half spaces in a plane



Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Example 3: Half spaces in a plane



Can one point be shattered?

Two points?

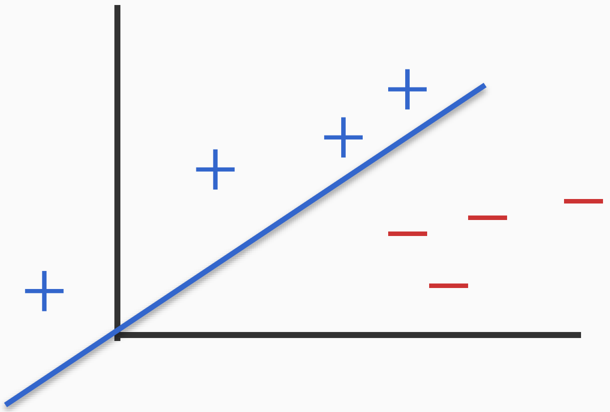
Three points? Can any three points be shattered?

Half spaces on a plane: 3 points

Shattering

Definition: A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Example 3: Half spaces in a plane



Can four points be shattered?

Suppose three of them lie on the same line, label the outside points + and the inner one –

Otherwise, make a convex hull. Label points outside + and the inner one –

Four points cannot be shattered!

Half spaces on a plane: 4 points

Shattering: The adversarial game

You

An adversary

Shattering: The adversarial game

You **An adversary**

You: Hypothesis class H can shatter ***these*** d points

Shattering: The adversarial game

You

An adversary

You: Hypothesis class H can shatter ***these*** d points

Adversary: That's what you think! Here is a labeling that will defeat you.

Shattering: The adversarial game

You

An adversary

You: Hypothesis class H can shatter ***these*** d points

Adversary: That's what you think! Here is a labeling that will defeat you.

You: Aha! There is a function $h \in H$ that correctly predicts your evil labeling

Shattering: The adversarial game

You

An adversary

You: Hypothesis class H can shatter ***these*** d points

Adversary: That's what you think! Here is a labeling that will defeat you.

You: Aha! There is a function $h \in H$ that correctly predicts your evil labeling

Adversary: Argh! You win this round. But I'll be back…..

Some functions can shatter infinite points!

If arbitrarily large finite subsets of the instance space X can be shattered by a hypothesis space H.

Intuition: A rich set of functions shatters large sets of points

Some functions can shatter infinite points!

If arbitrarily large finite subsets of the instance space X can be shattered by a hypothesis space H.

An unbiased hypothesis space H shatters the entire instance space X, i.e, it can induce every possible partition on the set of all possible instances

The larger the subset X that can be shattered, the more expressive a hypothesis space is, i.e., the less biased it is

Intuition: A rich set of functions shatters large sets of points

Vapnik-Chervonenkis Dimension

A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Vapnik-Chervonenkis Dimension

A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Definition: The VC dimension of hypothesis space H over instance space X is the size of the largest *finite* subset of X that is shattered by H

Vapnik-Chervonenkis Dimension

A *set S of examples* is shattered by a *set of functions* H if for *every* partition of the examples in S into positive and negative examples there is a function in H that gives exactly these labels to the examples

Definition: The VC dimension of hypothesis space H over instance space X is the size of the largest *finite* subset of X that is shattered by H

- If there exists any subset of size d that can be shattered, VC(H) >= d
  - Even one subset will do
- If no subset of size d can be shattered, then VC(H) < d

What we have managed to prove

Concept class	VC Dimension	Why?
Half intervals	1	There is a dataset of size 1 that can be shattered No dataset of size 2 can be shattered
Intervals	2	There is a dataset of size 2 that can be shattered No dataset of size 3 can be shattered
Half-spaces in the plane	3	There is a dataset of size 3 that can be shattered No dataset of size 4 can be shattered

More VC dimensions

Concept class	VC Dimension
Linear threshold unit in d dimensions	d + 1
Neural networks	Number of parameters
1 nearest neighbors	infinite

Intuition: A rich set of functions shatters large sets of points

More VC dimensions

Concept class	VC Dimension
Linear threshold unit in d dimensions	d + 1
Neural networks	Number of parameters
1 nearest neighbors	infinite

What is the number of parameters needed to specify a linear threshold unit in d dimensions?

Intuition: A rich set of functions shatters large sets of points

More VC dimensions

Concept class	VC Dimension
Linear threshold unit in d dimensions	d + 1
Neural networks	Number of parameters
1 nearest neighbors	infinite

What is the number of parameters needed to specify a linear threshold unit in d dimensions? d + 1

Intuition: A rich set of functions shatters large sets of points

More VC dimensions

Concept class	VC Dimension
Linear threshold unit in d dimensions	d + 1
Neural networks	Number of parameters
1 nearest neighbors	infinite

What is the number of parameters needed to specify a linear threshold unit in d dimensions? d + 1

Local minima in learning means neural networks may not find the best parameters

Intuition: A rich set of functions shatters large sets of points

More VC dimensions

Concept class	VC Dimension
Linear threshold unit in d dimensions	d + 1
Neural networks	Number of parameters
1 nearest neighbors	infinite

What is the number of parameters needed to specify a linear threshold unit in d dimensions? d + 1

Local minima in learning means neural networks may not find the best parameters

Exercise: Try to prove this after we see nearest neighbors

Intuition: A rich set of functions shatters large sets of points

Why VC dimension?

- Remember sample complexity
  - Occam's razor
  - Agnostic learning

- Sample complexity in both cases depends on the log of the size of the hypothesis space

- For infinite hypothesis spaces, its VC dimension behaves like $\log(|H|)$

VC dimension and Occam's razor for consistent learners

- Using VC(H) as a measure of expressiveness, we have an Occam theorem for infinite hypothesis spaces

- Given a sample D with m examples, find some $h \in H$ is consistent with all m examples. If

$$m > \frac{1}{\epsilon}\left(8\mathrm{VC}(H)\log\frac{13}{\epsilon} + 4\log\frac{2}{\delta}\right)$$

  Then with probability at least $1 - \delta$, the hypothesis h has error less than ϵ.

That is, if m is polynomial we have a PAC learning algorithm;
To be efficient, we need to produce the hypothesis h efficiently

VC dimension and Agnostic Learning

Similar statement for the agnostic setting as well

If we have m examples, then with probability $1 - \delta$, a the true error of a hypothesis h with training error $err_S(h)$ is bounded by

$$err_D(h) \leq err_S(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

(Phew!)

Exercises

What is the VC dimension axis parallel rectangles (which we saw at the beginning of this lecture)?

Your homework asks you to compute the VC dimension of different classes of functions

PAC learning: What you need to know

- What is PAC learning?
  - Remember: We care about generalization error, not training error
- Finite hypothesis spaces
  - Connection between size of hypothesis space and sample complexity
  - Derive and understand the sample complexity bounds
  - Count number of hypotheses in a hypothesis class
- Infinite hypotheses classes
  - What is shattering and VC dimension?
  - How to find VC dimension of simple concept classes?
  - Higher VC dimensions $\Rightarrow$ more sample complexity

Computational Learning Theory

- Probably Approximately Correct (PAC) learning
  - A general definition that assumes fixed, but perhaps unknown distribution
- Occam’s razor for consistent learners in finite hypothesis spaces
  - Positive and negative learnability results in this setting
- Agnostic Learning and the associated Occam razor
- Shattering and the VC dimension
- Many extensions to the theory exist
  - Noisy data, known data distributions, probabilistic models
  - One important extension: PAC-Bayes theory that makes assumptions about the the prior distribution over hypothesis spaces

COLT still doesn't explain why learning works in all cases

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs].

Why computational learning theory

- Answers questions such as
  - What is learnability? How good is my class of functions?
  - Is a concept learnable? How many examples do I need?

- Mistake bounds imply PAC-learnability

- Raises interesting theoretical questions
  - If a concept class is weakly learnable (i.e there is a learning algorithm that can produce a classifier that does slightly better than chance), does this mean that the concept class is strongly learnable?

  - We have seen bounds of the form

    true error < training error + (a term with ϵ, δ *and VC dimension)*

    Can we use this to define a learning algorithm?

Why computational learning theory

- Answers questions such as
  - What is learnability? How good is my class of functions?
  - Is a concept learnable? How many examples do I need?

- Mistake bounds imply PAC-learnability

- Raises interesting theoretical questions
  - If a concept class is weakly learnable (i.e there is a learning algorithm that can produce a classifier that does slightly better than chance), does this mean that the concept class is strongly learnable?

    Boosting
  - We have seen bounds of the form

    true error < training error + (a term with ϵ, δ *and VC dimension)*

    Can we use this to define a learning algorithm?

    Structural Risk Minimization principle
    Support Vector Machine