Lecture 10: Deep Learning Multiclass Classification

Fall 2022

Kai-Wei Chang

CS @ UCLA

kw+cm146@kwchang.net

The instructor gratefully acknowledges Dan Roth, Vivek Srikuar, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

Announcements

- Quiz 4 will be released tomorrow

Multi-class Logistic Regression

Reduction v.s. single classifier

- Reduction
  - Future-proof: if we improve the binary classification model ⇒ improve muti-class classifier
  - Easy to implement
- Single classifier
  - Global optimization: directly minimize the empirical loss; easier for joint prediction

Lecture 11: Computational Learning Theory

Fall 2022

Kai-Wei Chang

CS @ UCLA

kw+cm146@kwchang.net

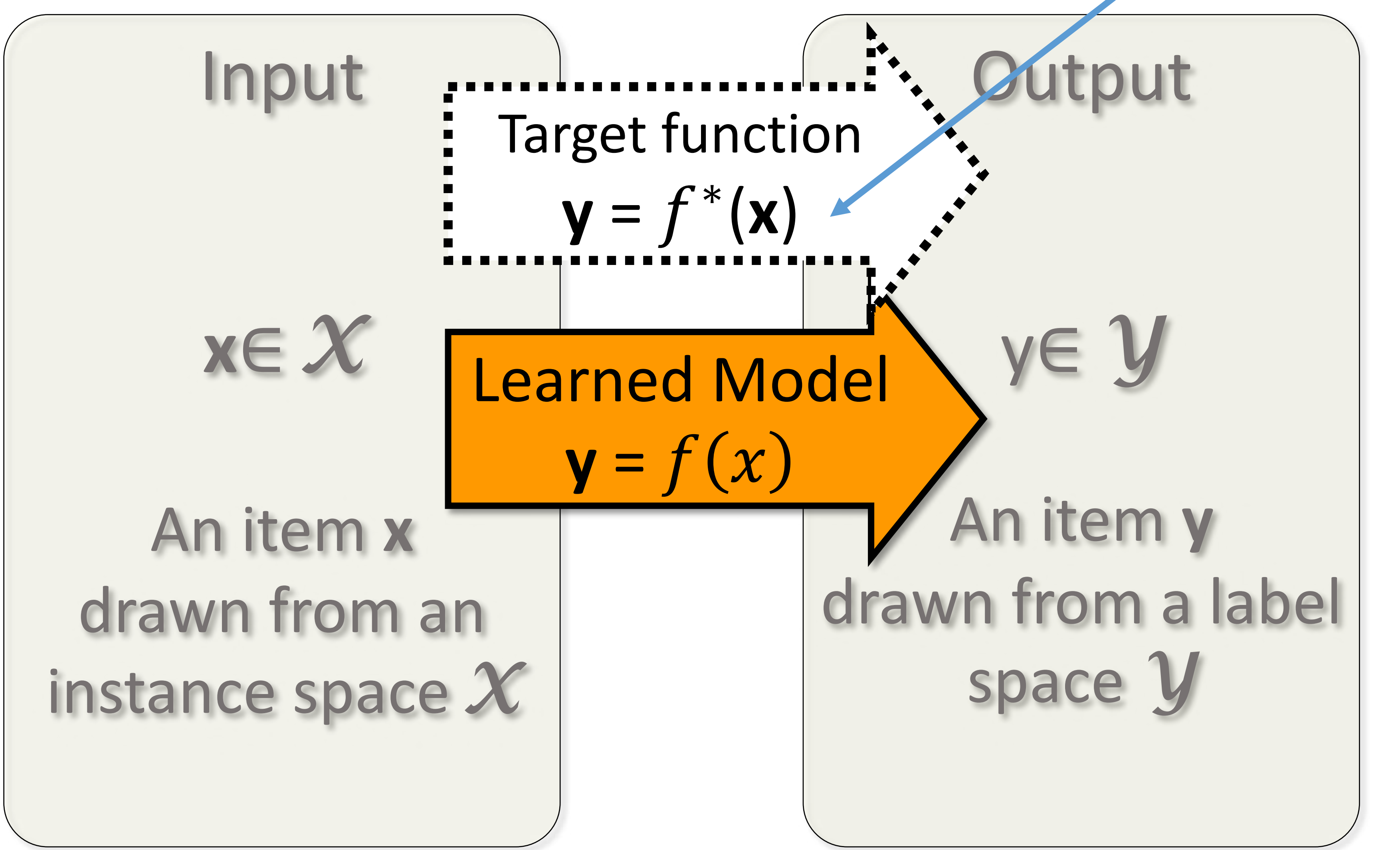
The instructor gratefully acknowledges Dan Roth, Vivek Srikuar, Sriram Sankararaman, Fei Sha, Ameet Talwalkar, Eric Eaton, and Jessica Wu whose slides are heavily used, and the many others who made their course material freely available online.

This lecture: Computational Learning Theory

- The Theory of Generalization
  - Difference between learning and memorizing
- Probably Approximately Correct (PAC) learning
  - How many #training samples you need to get a good classifier
    - Good classifier = with high probability, the error is low
  - We will use the monotone conjunction function class as example

Learning the Mapping

What target function class
Is learnable?



Learning Monotone Conjunctions

- Hypothesis class:

$f = x_1?$ $\quad$ $f = x_2?$ $\quad$ $f = x_1 \wedge x_2 \wedge x_3\ ?$

$f = x_1 \wedge x_2?$ $\quad$ $f = x_2 \wedge x_3\ ?$

Learning Monotone Conjunctions

- Hypothesis class:

$f = x_1?$ $\quad f = x_2?$ $\quad f = x_1 \wedge x_2 \wedge x_3\,?$

$f = x_1 \wedge x_2?$ $\quad f = x_2 \wedge x_3\,?$ $\quad$ ....

- Target function in the hindsight

$$f = x_2 \wedge x_3$$

Exercise

- Hypothesis class: Monotone Conjunctions

$f = x_1?$ $\quad$ $f = x_2?$ $\quad$ $f = x_1 \wedge x_2 \wedge x_3\ ?$

$f = x_1 \wedge x_2?$ $\quad$ $f = x_2 \wedge x_3\ ?$

- Given the following data
  - <(1,1,1), 1>
  - <(1,0,1), 0>
  - <(0,1,1), 1>
  - <(1,1,0), 0>
- Predict <(0,1,0), ? >

Exercise

- Hypothesis class: All Boolean functions

$f = x_1?$ $f = x_2?$ $f = (x_1 \wedge x_2) \vee x_3$?

$f = \neg x_1?$ $f = x_1 \wedge x_2?$ $f = \neg x_1 \vee x_2?$ $f = x_2 \wedge \neg x_3$?

- Given the following data
  - <(1,1,1), 1>
  - <(1,0,1), 0>
  - <(0,1,1), 1>
  - <(1,1,0), 0>
- Predict <(0,1,0), ? >

Exercise

- Hypothesis class (3 variables):

$f = x_1?$ $f = x_2?$ $f = x_1 \land x_2 \land x_3\ ?$

$f = x_1 \land x_2?$ $f = x_2 \land x_3\ ?$

- Given the following data, what is the right function
- <(1,1,1), 1>
- <(1,0,1), 0>
- <(0,1,1), 1>
- <(1,1,0), 0>

This lecture: Computational Learning Theory

- When can we say a concept is learnable?
  - Learning v.s. memorization
  - Don’t need to see all samples to make a good prediction
  - Can you compute 1234+2332 = ?

This lecture: Computational Learning Theory

- When can we say a concept is learnable?
  - Learning v.s. memorization
  - Don’t need to see all samples to make a good prediction
  - Can you compute 1234+2332 = ?

- How many training data do we need to train a good classifier? (sample complexity)

This lecture: Computational Learning Theory

- When can we say a concept is learnable?
  - Learning v.s. memorization
  - Don’t need to see all samples to make a good prediction
  - Can you compute 1234+2332 = ?

- How many training data do we need to train a good classifier? (sample complexity)

- PAC learnable – if #examples we need to see is polynomial to the parameters defining the concept (details will discuss later)

$$f = x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$$

Learning Monotone Conjunctions

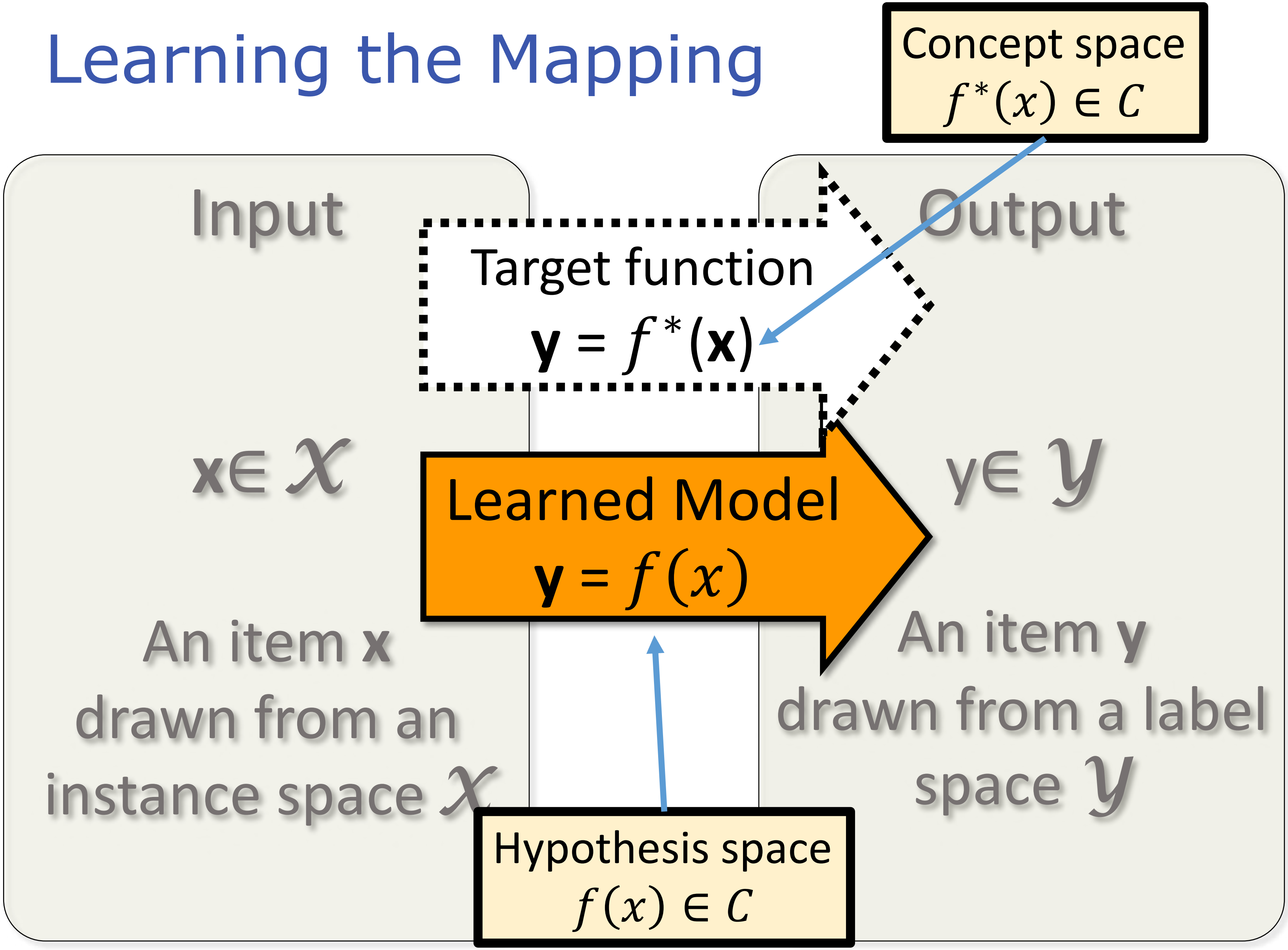
- Supervised Learning

  Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,1,1,1,1,0,...0,1,1), 1>
  - <(1,0,1,1,1,0,...0,1,1), 0>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

The setup

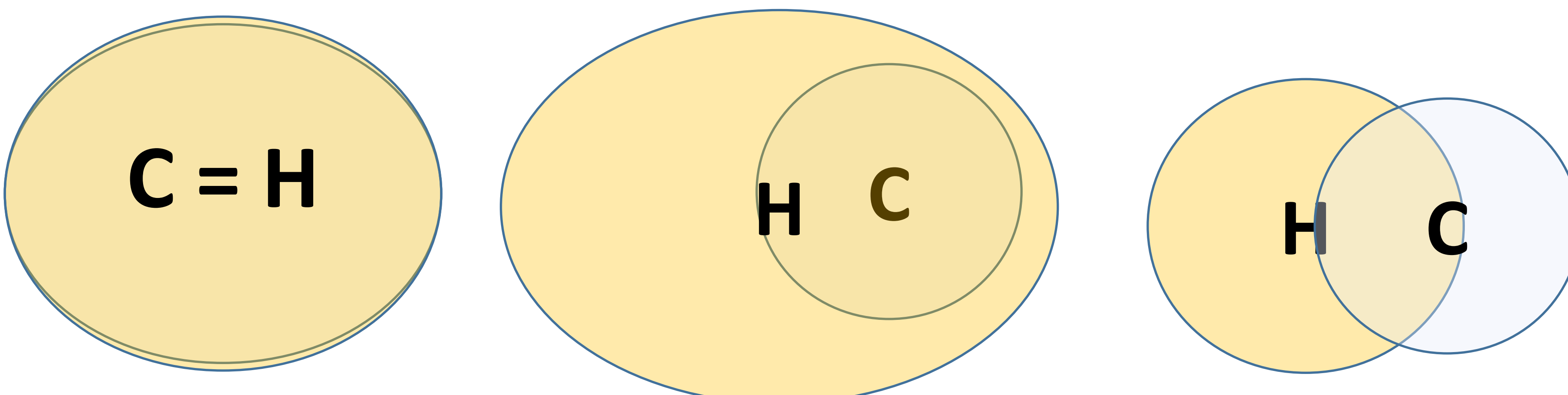
- Instance Space: X, the set of examples
- Concept Space: C, the set of possible target functions: $f \in C$ is the hidden target function
  - E.g.: all n-conjunctions; all n-dimensional linear functions, …
- Hypothesis Space: H, the set of possible hypotheses
  - This is the set that the learning algorithm explores

Learning the Mapping



Concept Space v.s. Hypothesis Space

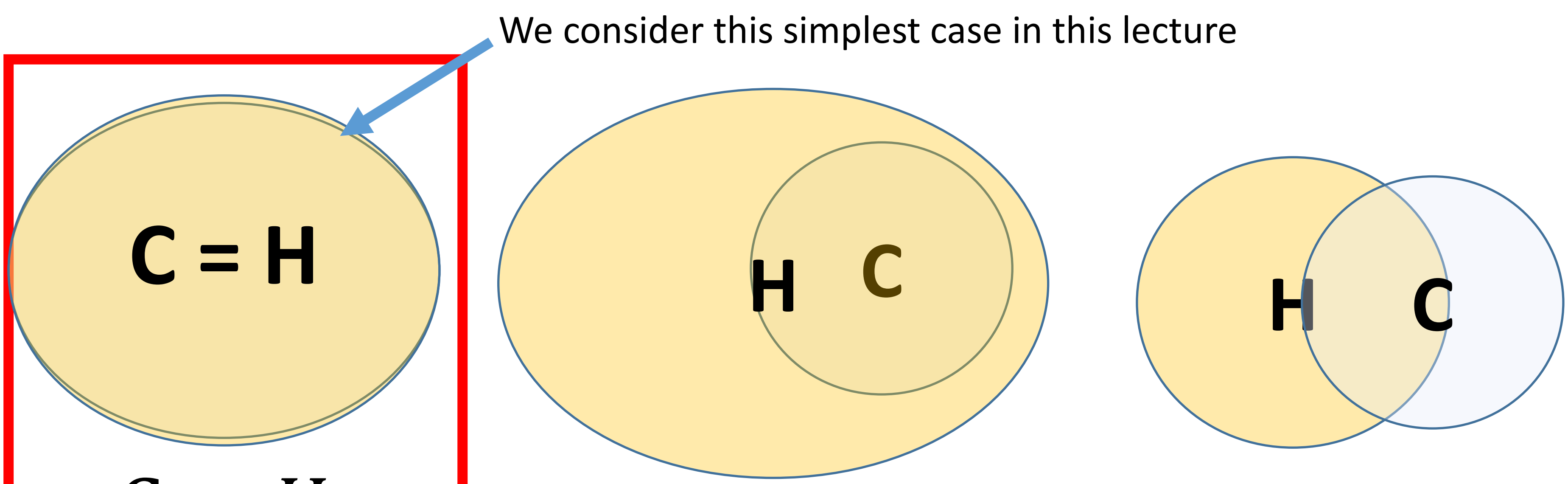
- Concept space = Hypothesis space
  - We will work on this setting in this lecture
- Concept space ⊂ Hypothesis space
- Concept space ⊄ Hypothesis space



$C = H$ $\quad$ $C \subset H$ $\quad$ $C \not\subseteq H$

Concept Space v.s. Hypothesis Space

- Concept space = Hypothesis space
  - We will work on this setting in this lecture
- Concept space ⊂ Hypothesis space
- Concept space ⊄ Hypothesis space



$C = H$ $\quad C \subset H$ $\quad C \not\subseteq H$

The setup

- Instance Space: X, the set of examples
- Concept Space: C, the set of possible target functions: $f \in C$ is the hidden target function
  - E.g.: all n-conjunctions; all n-dimensional linear functions, …
- Hypothesis Space: H, the set of possible hypotheses
  - This is the set that the learning algorithm explores
- Training instances: S x {-1,1}: positive and negative examples of the target concept drawn from distribution D $(S \subseteq X)$

$$< x_1, f(x_1) >, < x_2, f(x_2) >, \ldots < x_n, f(x_n) >$$

- What we want: A hypothesis h $\in$ H such that h(x) = f(x)
- Assumption: Training and test data are both drawn i.i.d. from X

PAC learning

- A framework for *batch learning*
  - Train on a fixed training set
  - Then deploy it in the wild
- How well will your learning algorithm do in *future* instances?
- We will first analyze an algorithm for learning conjunctions
- Then we will define PAC learning

Learning monotone Conjunctions

- Assume both C, H are monotone conjunctions
- Supervised Learning
  Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would be the f?

- Assumption: data are sample from a fixed distribution

Learning monotone Conjunctions

- Teacher provides a set of examples (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: can the function to be $f = x_1 \wedge x_2$?

Learning monotone Conjunctions

- Teacher provides a set of examples (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: can the function to be $f = x_1 \land x_2$?

No, 3rd instance is an violation

Learning monotone Conjunctions

- Teacher provides a set of examples (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: does x_2 in the formulation of f ?

Learning monotone Conjunctions

- Teacher provides a set of examples (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>
- Question: does x_2 in the formulation of f ?

Guess what would f look like?

No, again, 3rd instance is an violation

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: does x_1 in the conjunction formulation?

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>
- Question: does x_1 in the conjunction formulation?

Guess what would f look like?

Maybe, why? Whenever the output is 1, x_1 is present

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>
- Question: can we ensure that x_1 is in the conjunction formulation?

Guess what would f look like?

No, why? It is possible there is a counter-example we've never seen <(0,1,1,1,0,1,…,0,1), 1>

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: can we ensure that x_1 is in the conjunction formulation?

No, but if we have seen many positive examples where $x_1 = 1$
it is likely x_1 is in the formulation of f

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>
- How to learn a monotone conjunction consistent with these training samples

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>
- How to learn a monotone conjunction consistent with these training samples

Start with having all literals in the monotone conjunctions.
Removing literal j if $x_j = 0$ in some positive instances.

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: Based on the algorithm we get $x_1 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$
  Can we guarantee this is f ?

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: Based on the algorithm we get $x_1 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$
  Can we guarantee this is f ?

No, examples can be generated by the following function and we just haven't seen the counter-example <(0,1,1,1,0,1,…,0,1), 1>
$f = x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: If the algorithm eliminates x_2 (e.g,. we get $x_1 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$) is it possible x_2 is in f?

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Guess what would f look like?

- Question: If the algorithm eliminates x_2 (e.g,. we get $x_1 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$) is it possible x_2 is in f?

No, because f is a conjunction function

Learning monotone Conjunctions

- Teacher provides a set of example (x, f(x))
  - <(1,1,1,1,1,1,…,1,1), 1>
  - <(1,1,1,0,0,0,…,0,0), 0>
  - <(1,0,1,1,1,0,...0,1,1), 1>
  - <(1,1,1,1,1,0,...0,0,1), 1>
  - <(1,0,1,0,0,0,...0,1,1), 0>
  - <(1,1,1,1,1,1,…,0,1), 1>
  - <(0,1,0,1,0,0,...0,1,1), 0>

Intuitively, with more training data, it's unlikely we never see $x_1 = 0$ in positive training examples but see it in the test time if data are sampled from the same distribution

- Question: When the target function is $x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$, but our algorithm returns $x_1 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_{100}$? How likely this would happen when I already see N examples?

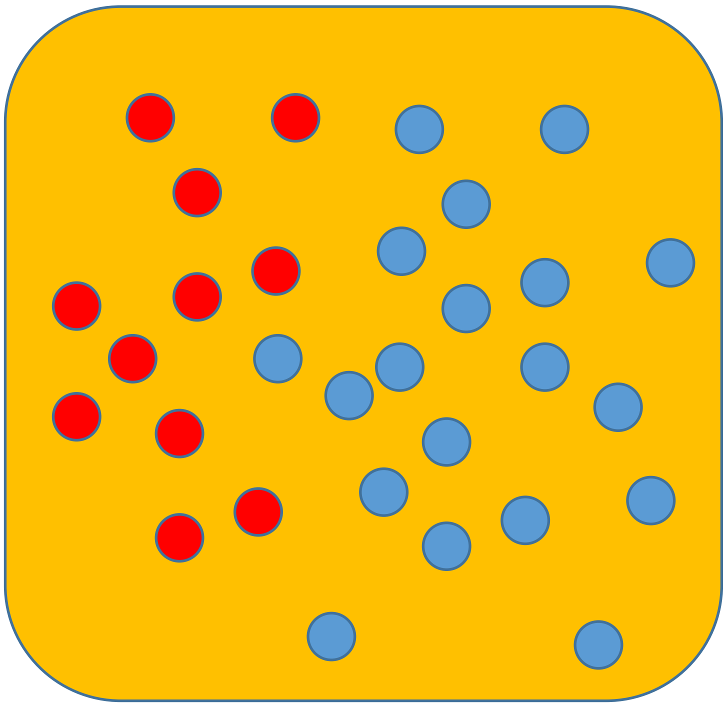
"The future will be like the past":

- We have seen many examples (drawn according to the distribution D)
  - Since in all the positive examples x_1 was active, it is very likely that it will be active in future positive examples
  - Otherwise, x_1 is active only in a small percentage of the examples so our error will be small

How likely we never see the examples like <(0,1,1,1,1,1,…,0,1), 1> to filter out x_1 but have such cases in the test time

Illustrative Example

- Scenario 1: 10 red balls with 20 blue balls



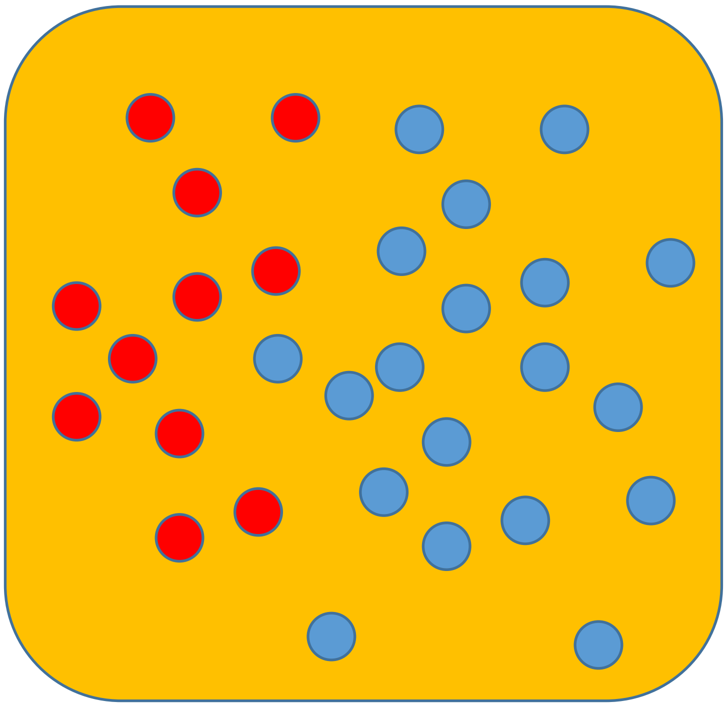
Training set: 100 points

Test set: 1 point

When $x_1 = 0, y = 1$ otherwise

Illustrative Example

❖ Scenario 1: 10 red balls with 20 blue balls



Training set: 100 points

Test set: 1 point

Never see the red ball in the training:

$$\left(\frac{2}{3}\right)^{100} \cong 2.45\text{E-}18$$

See a red ball in the test time:

$$\frac{1}{3} \cong 0.3$$

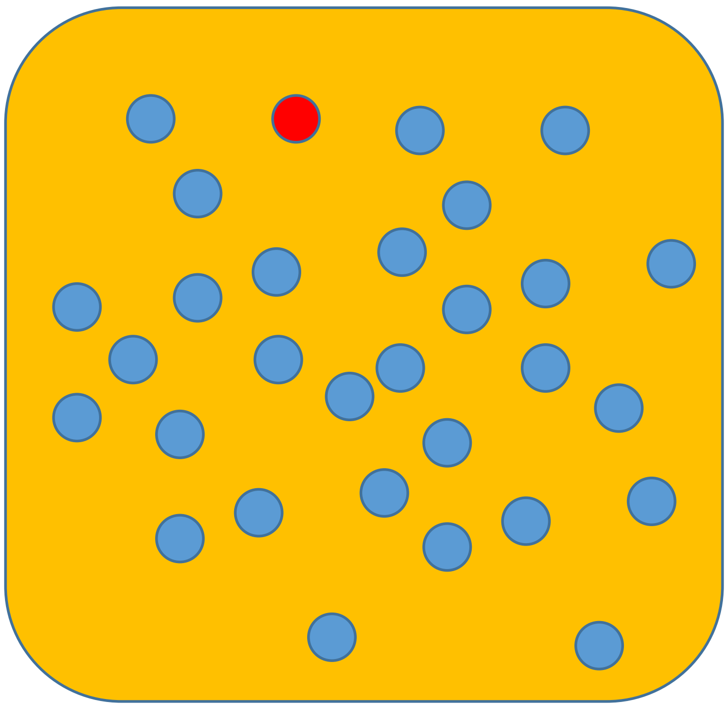
Both events happen: ~ 0%

When #(training points) is large, it's unlikely we never see a red ball

Illustrative Example

❖ Scenario 2: 1 red ball with 29 blue balls

How likely we see a red point in the test time but not in training time?



Training set: 100 points

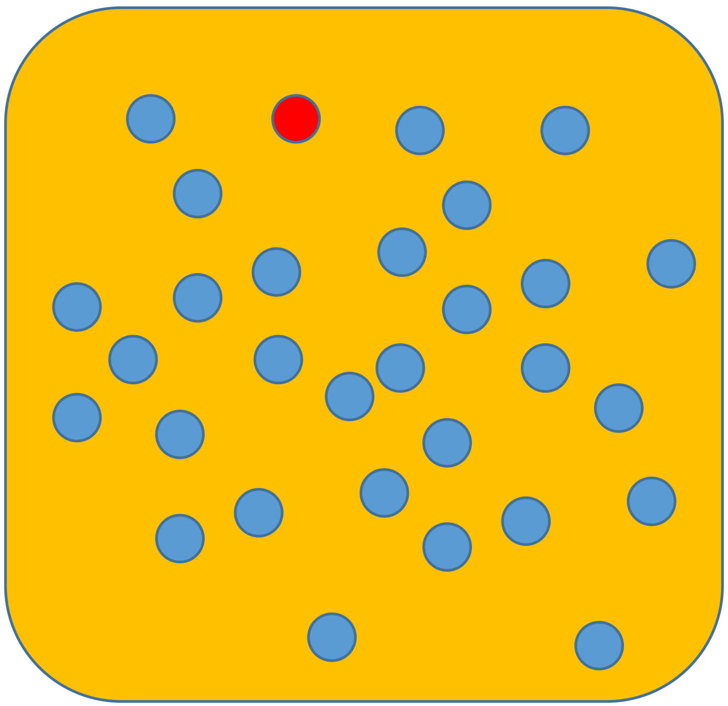
Test set: 1 point

When $x_1 = 0, y = 1$ otherwise

Illustrative Example

- Scenario 2: 1 red ball with 29 blue balls

How likely we see a red point in the test time but not in training time?



Training set: 100 points

Test set: 1 point

Never see the red ball in the training:

$$\left(\frac{29}{30}\right)^{100} \cong 0.0337$$

See a red ball in the test time:

$$\frac{1}{30} \cong 0.0333$$

Both events happen: ~ 0.11%

If there is only very few red ball, we may miss them in training, but the probability we see them in test is also low

Error of a hypothesis

Definition

Given a distribution D over examples, the *error* of a hypothesis h with respect to a target concept f is

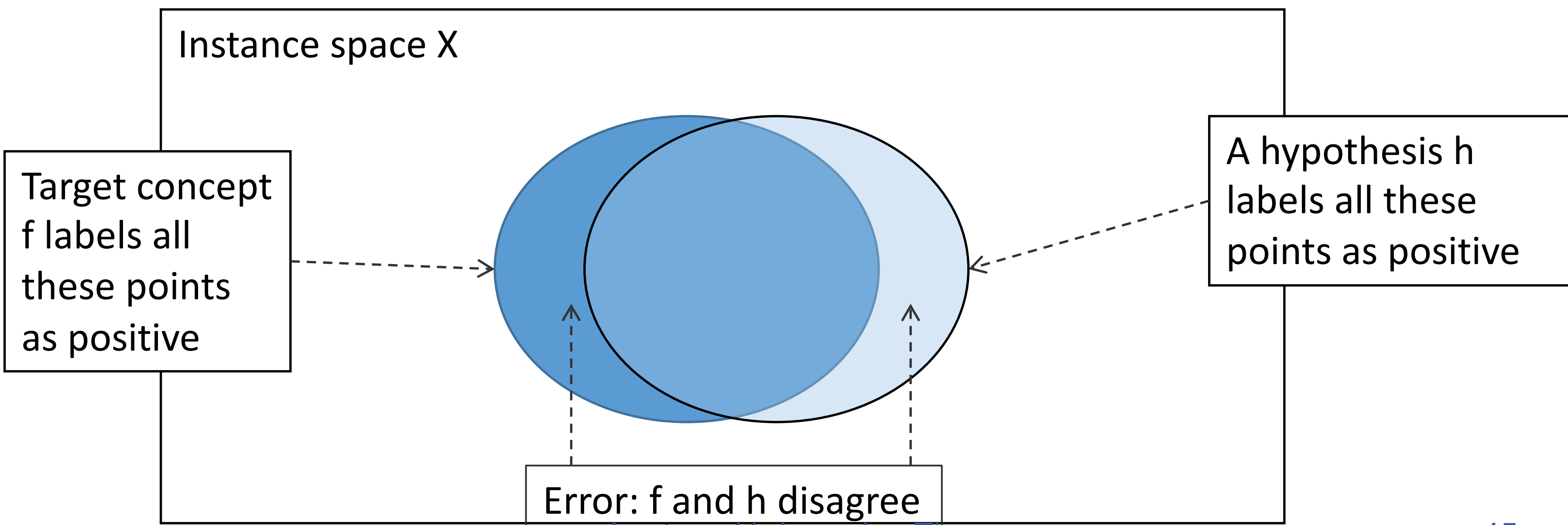
$$err_D(h) = Pr_{x \sim D}[h(x) \neq f(x)]$$

Error of a hypothesis

Definition

Given a distribution D over examples, the *error* of a hypothesis h with respect to a target concept f is

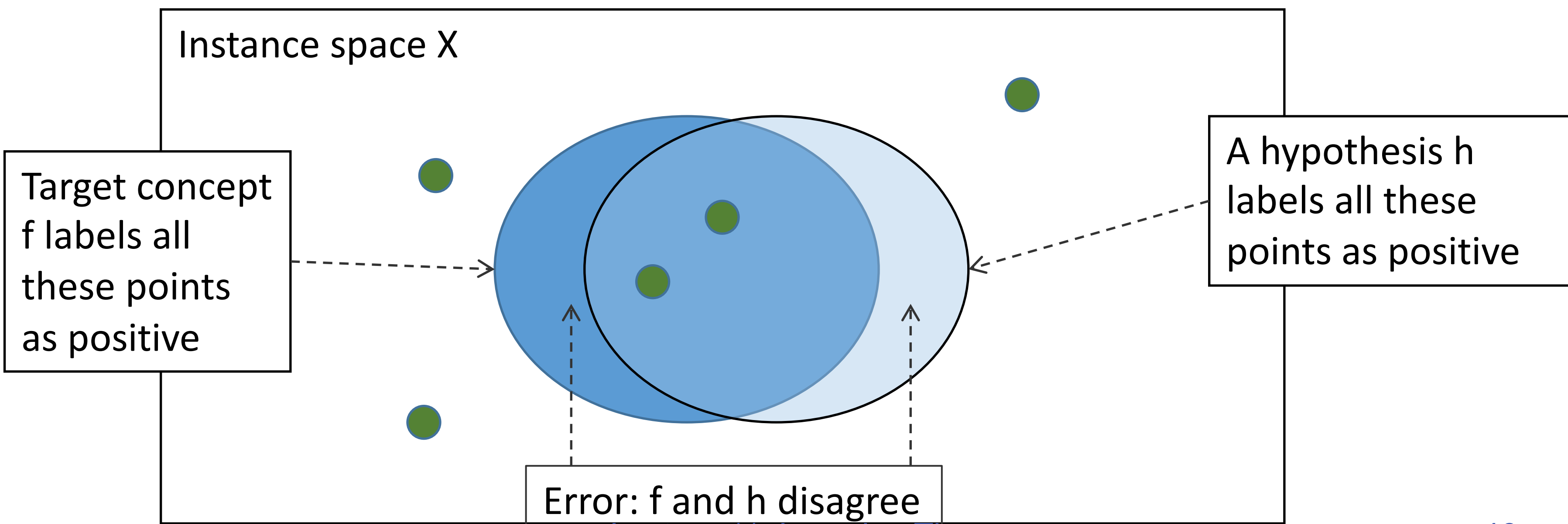
$$err_D(h) = Pr_{x \sim D}[h(x) \neq f(x)]$$



Error of a hypothesis

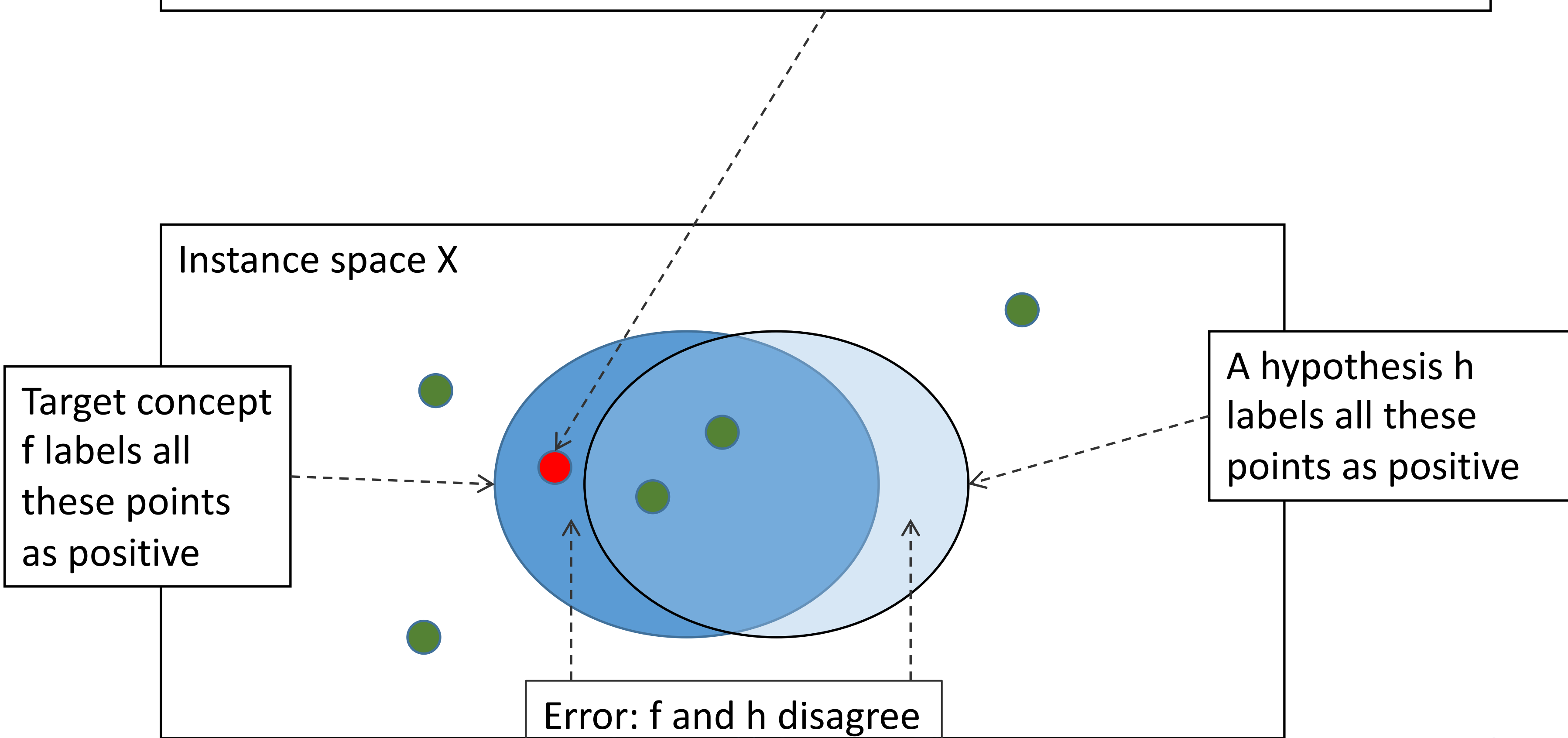
You may have a learned model that is consistent with the training data but still makes mistakes.

- Samples correctly predicted by h
- Samples incorrectly predicted by h



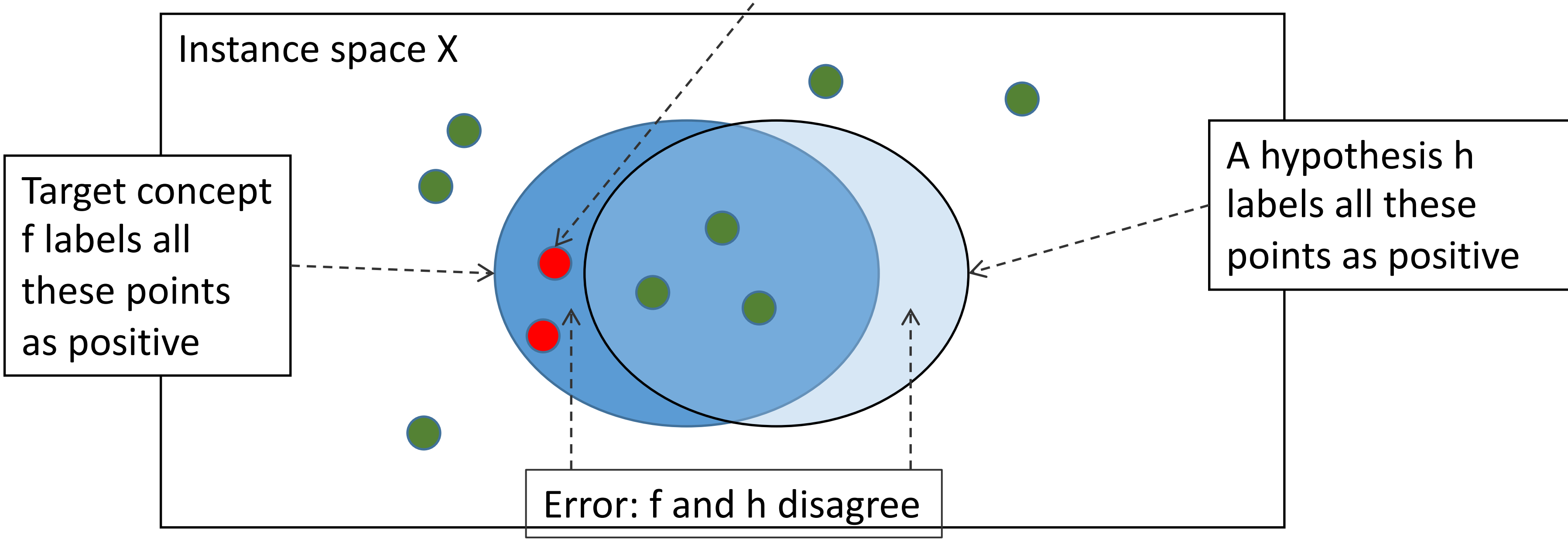
Error of a hypothesis

With the IID sampling assumption, we either have seen this example in the training phase, or it is unlikely to see it in the test time.



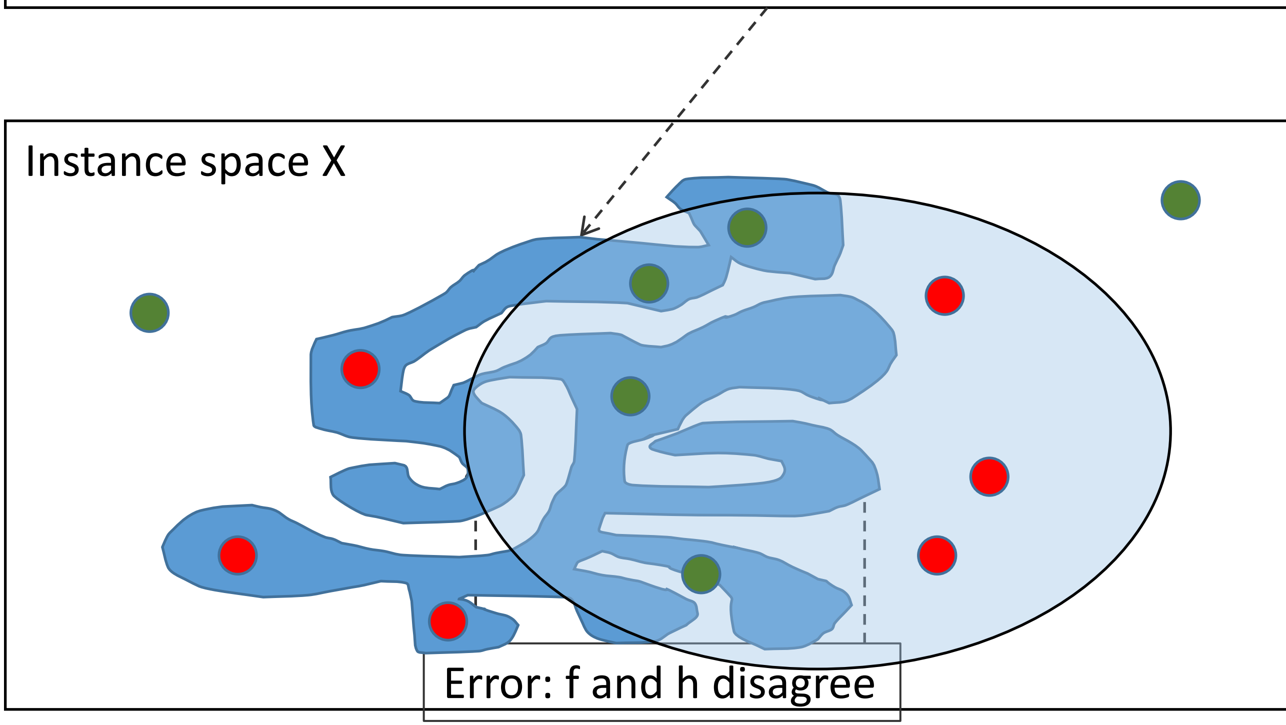
Intuition of PAC Learnability

With the IID sampling assumption, if a concept is reasonable. After, we saw enough samples, it is unlikely to have many these red points



Intuition of PAC Learnability

With the IID sampling assumption, if a concept is too complicated. We need to see exponential number of samples, such that we can rule out those red points



PAC Learning for Monotone Conjunctions

- Consider the concept space and hypothesis space are both monotone conjunctions with n variables
- Algorithm: Start with having all literals in the monotone conjunctions. Removing literal j if $x_j = 0$ in some positive instances.
- With probability $(1 - \delta)$, the above algorithm requires ?? examples to achieve an error rate $< \epsilon$
  - E.g., how may examples we need to ensure the error <5% with 99% probability δ=1%, ϵ=5%)
- Let's consider the case n=10, δ=1%, ϵ=5%

How likely is the learned h wrong

How many examples do we need to learn 10-variable monotone conjunctions such that with a probability 99%, the algorithm achieves an error rate $< 5\%$

- When we will make a mistake?
  - h includes some ”bad literal” z, where we never see (x_z = 0, $y = 1$) in training but see it in test time.

How likely is the learned h wrong

How many examples do we need to learn 10-variable monotone conjunctions such that with a probability 99%, the algorithm achieves an error rate $< 5\%$

- When we will make a mistake?
  - h includes some ”bad literal” z, where we never see (x_z = 0, $y = 1$) in training but see it in test time.
  - Let p(z) be the probability that z is a bad literal
- To achieve 5% error rate, it is sufficient to ensure the probability of each ”bad literal” p(z) < 0.5% (i.e., ϵ/n)

h makes a mistake if it contains any bad literal

How likely is the learned h wrong

How many examples do we need to learn 10-variable monotone conjunctions such that with a probability 99%, the algorithm achieves an error rate $< 5\%$

- To achieve 5% error rate, it is sufficient to ensure the probability of each "bad literal" p(z) < 0.5% (i.e., ϵ/n)
- There are two cases:
  - The probability of seeing (x_z = 0, $y = 1$) < 0.5%
    - The probability see it in test time is already < 0.5%
  - The probability of seeing (x_z = 0, $y = 1$) > 0.5% (i.e., > ϵ/n)
    - We can bound #examples we need to ensure p(z) < 0.5%

$$Pr(\text{Any bad literal survives } m \text{ examples})$$

Learning Conjunctions: Analysis

We assume the probability of seeing (x_z = 0, $y = 1$) > ϵ/n

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

Learning Conjunctions: Analysis

We assume the probability of seeing (x_z = 0, $y = 1$) > ϵ/n

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

But say we have m training examples. Then

$$Pr(\text{A bad literal survives } m \text{ examples}) < \left(1 - \frac{\epsilon}{n}\right)^m$$

Learning Conjunctions: Analysis

We assume the probability of seeing ($x_z = 0$, $y = 1$) > ϵ/n

$$Pr(\text{A bad literal is not eliminated by one example}) < 1 - \frac{\epsilon}{n}$$

But say we have m training examples. Then

$$Pr(\text{A bad literal survives } m \text{ examples}) < \left(1 - \frac{\epsilon}{n}\right)^m$$

There are at most n bad literals. So

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

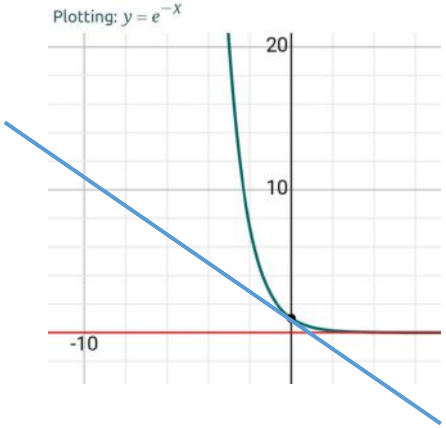
Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

We want this probability to be small than 1% (i.e., δ)

Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than δ

That is, we want $n\left(1 - \frac{\epsilon}{n}\right)^m < \delta$



We know that $1 - x < e^{-x}$. So it is sufficient to require

$$ne^{-\frac{m\epsilon}{n}} < \delta$$

Learning Conjunctions: Analysis

$$Pr(\text{Any bad literal survives } m \text{ examples}) < n\left(1 - \frac{\epsilon}{n}\right)^m$$

We want this probability to be small than 1% (i.e., δ)

Why? So that we can choose enough training examples so that the probability that any z survives all of them is less than δ

That is, we want

$$n\left(1 - \frac{\epsilon}{n}\right)^m < \delta$$

We know that $1 - x < e^{-x}$. So it is sufficient to require $ne^{-\frac{m\epsilon}{n}} < \delta$

Or equivalently,

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

Learning Conjunctions: Analysis

Theorem: Suppose we are learning a monotone conjunctive concept with n-dimensional Boolean features using m training examples. If

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

then, with probability > 1 - δ, the error of the learned hypothesis $err_D(h)$ will be less than ϵ.

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if for all $f \in C$, for all distribution D over X, and fixed $\epsilon > 0$, $\delta < 1$, given m examples sampled i.i.d. according to D, the algorithm L produces, with probability at least (1- δ), a hypothesis h $\in$ H that has error at most ε, where m is *polynomial* in 1/ ε, 1/ δ, n and size(H)

example: conjunction: $m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$

efficiently learnability

- The concept class C is *efficiently learnable* if L can produce the hypothesis in time that is polynomial in $1/\varepsilon$, $1/\delta$, n and size(H)

PAC Learnability

- We impose two limitations
- Polynomial *sample complexity* (information theoretic constraint)
  - Is there enough information in the sample to distinguish a hypothesis h that approximate f ?
- Polynomial *time complexity* (computational complexity)
  - Is there an efficient algorithm that can process the sample and produce a good hypothesis h ?

Example Disjunction

Let H be any hypothesis space.

With probability 1 -δ a hypothesis h → H that is consistent with a training set of size m will have an error < ϵ on future examples if

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

Size of hypothesis class for disjunction class $|H| = 3^n$, so a sufficient number of example to learn the disjunction concept is

$$\mathrm{m} > \frac{1}{\epsilon}\left(n \ln 3 + \ln\frac{1}{\delta}\right)$$

$\delta = \epsilon = 0.05, n = 10 \quad \Rightarrow m > 280$

$\delta = 0.01\ \epsilon = 0.05, n = 10 \quad \Rightarrow m > 312$

$\delta = \epsilon = 0.01, n = 10$ ⇒ m > 1,625

$\delta = \epsilon = 0.01, n = 50$ ⇒ m > 5,954

A general result

Let H be any hypothesis space.

With probability 1 -δ a hypothesis h → H that is consistent with a training set of size m will have an error < ϵ on future examples if

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

1. Expecting lower error increases sample complexity (i.e more examples needed for the guarantee)

2. If we have a larger hypothesis space, then we will make learning harder (i.e higher sample complexity)

3. If we want a higher confidence in the classifier we will produce, sample complexity will be higher.

Example: Learning Monotone Conjunctions

Suppose we are learning a monotone conjunctive concept with n-dimensional Boolean features using m training examples. If

$$m > \frac{n}{\epsilon}\left(\log(n) + \log\left(\frac{1}{\delta}\right)\right)$$

then, with probability > 1 - δ, the error of the learned hypothesis $err_D(h)$ will be less than ϵ.

m is *polynomial* in 1/ ε, 1/ δ, n and size(H)

Example Arbitrary Boolean Function

Let H be any hypothesis space.

With probability $1 - \delta$ a hypothesis $h \rightarrow H$ that is consistent with a training set of size m will have an error < ϵ on future examples if $m > \frac{1}{\epsilon}\left(\ln(|H|) + \ln\frac{1}{\delta}\right)$

Size of hypothesis class for Boolean functions is $|H| = 2^{2^n}$, so a sufficient number of example to learn the Boolean function concept is

$$\mathrm{m} > \frac{1}{\epsilon}\left(2^n \ln 2 + \ln\frac{1}{\delta}\right)$$

$\delta = \epsilon = 0.05, n = 10 \quad \Rightarrow m > 14{,}256$

$\delta = \epsilon = 0.05, n = 50 \quad \Rightarrow m > 1.5\times10^{16}$

Extend to real value functions (not in exam)

- VC(H) quantifies the complexity of the hypothesis space

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