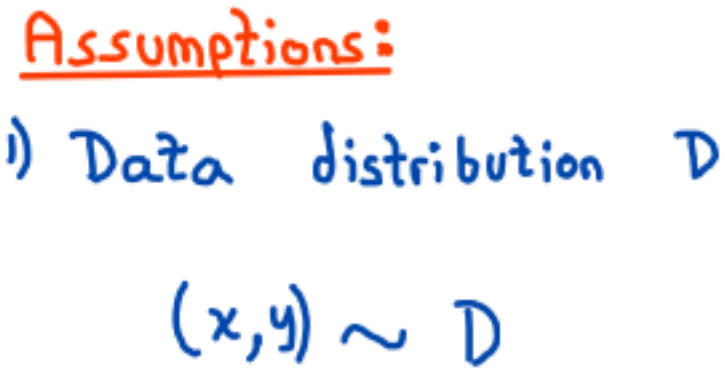
Lecture 9 [Approx/Estimation Error & ERM]

Learning Theory:

Setup / Assumptions:

The assumptions under which we're going to be operating are:

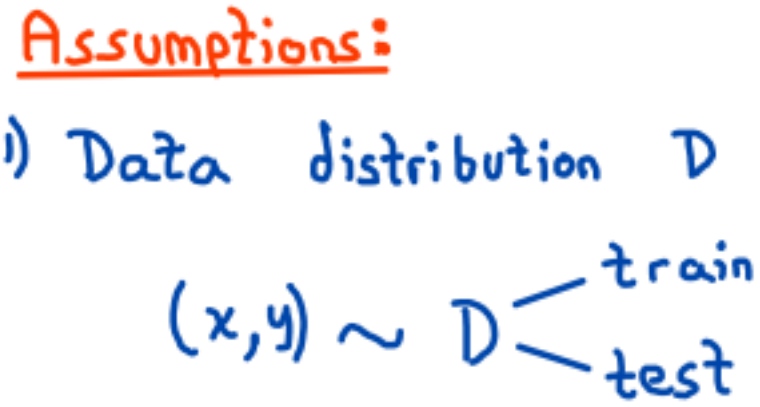
One is that there exists a data distribution **D** from which **(x,y)** pairs are sampled



This makes sense in the supervised learning setting where you're expected to learn a mapping from **x** to **y**, but the assumption also actually holds more generally, even in the unsupervised setting case

The main assumption is that there is a data-generating distribution and the examples that we have in our ***training set*** and the ones we will be encountering when we test it, are all coming from the same distribution

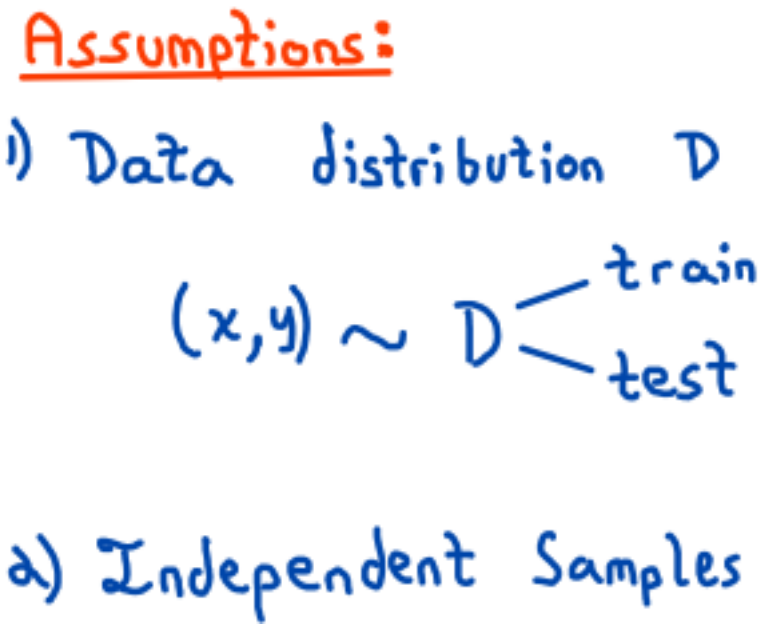
Without this assumption, coming up with any theory is going to be much harder



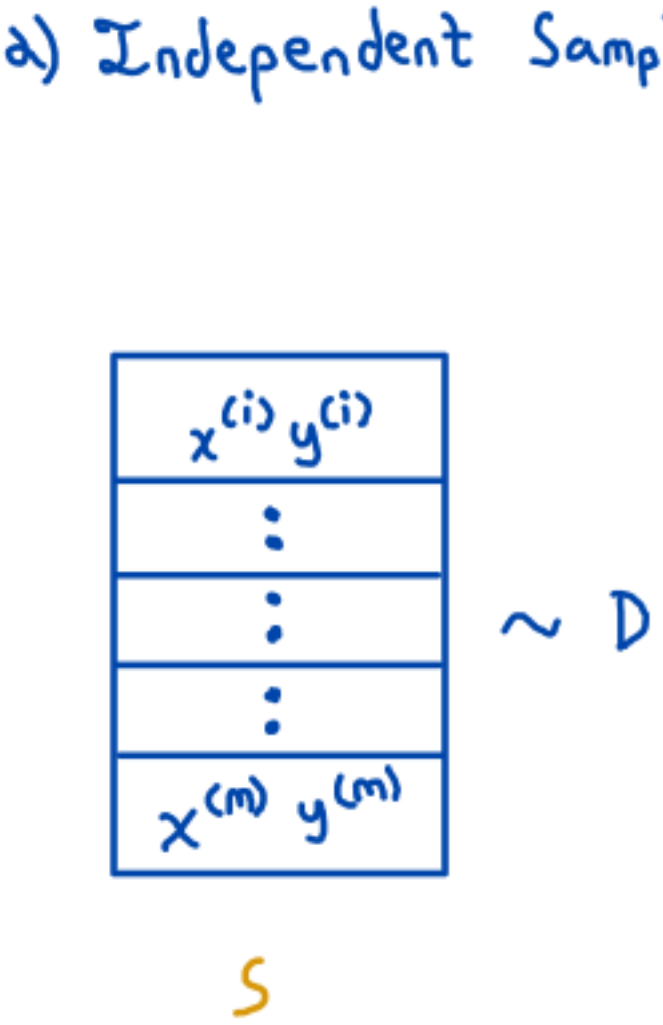
So the assumption here is that there is some kind of a data-generating process, and we have a few samples from that data-generating process that becomes our ***training set*** and that's a finite number

You can get an infinite number of samples from this data generating process, and the examples that we're going to encounter at test time are also samples from the same process

There is a second assumption, which is that all these samples are sampled independently:

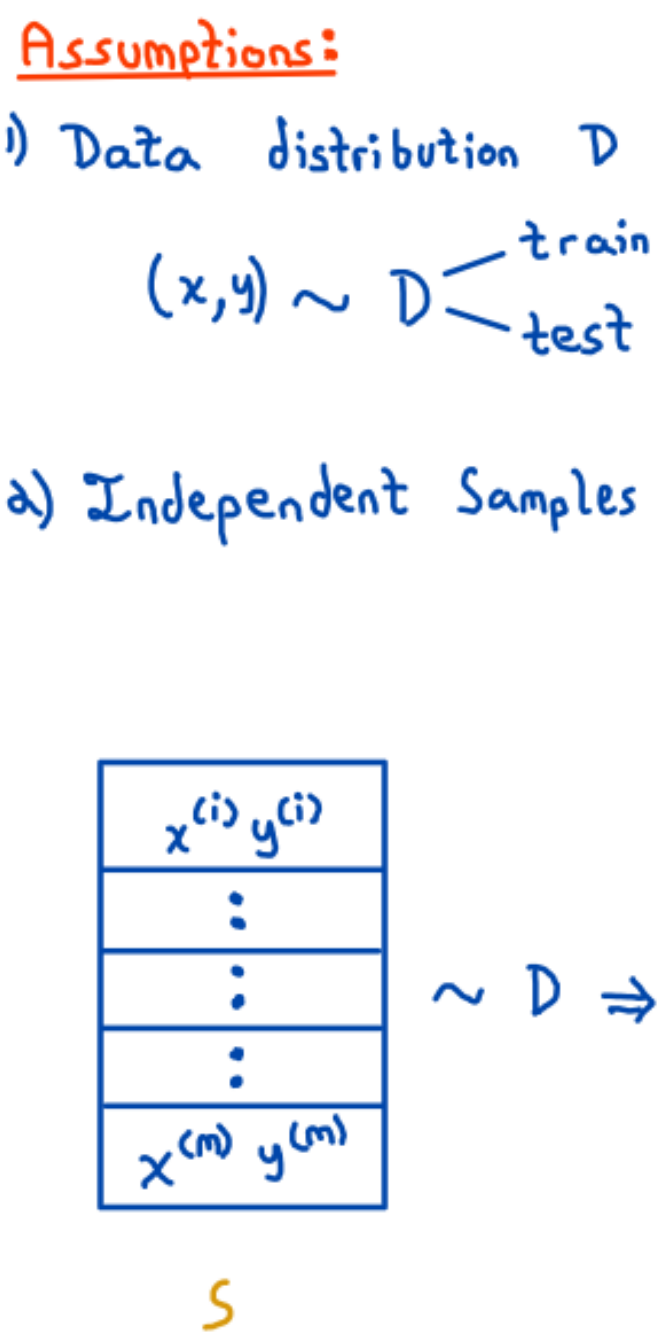


With these two assumptions, we can imagine the process of learning to look something like this:



So we have **m** samples from the data-generating process and we feed this into a learning algorithm and the output of the learning

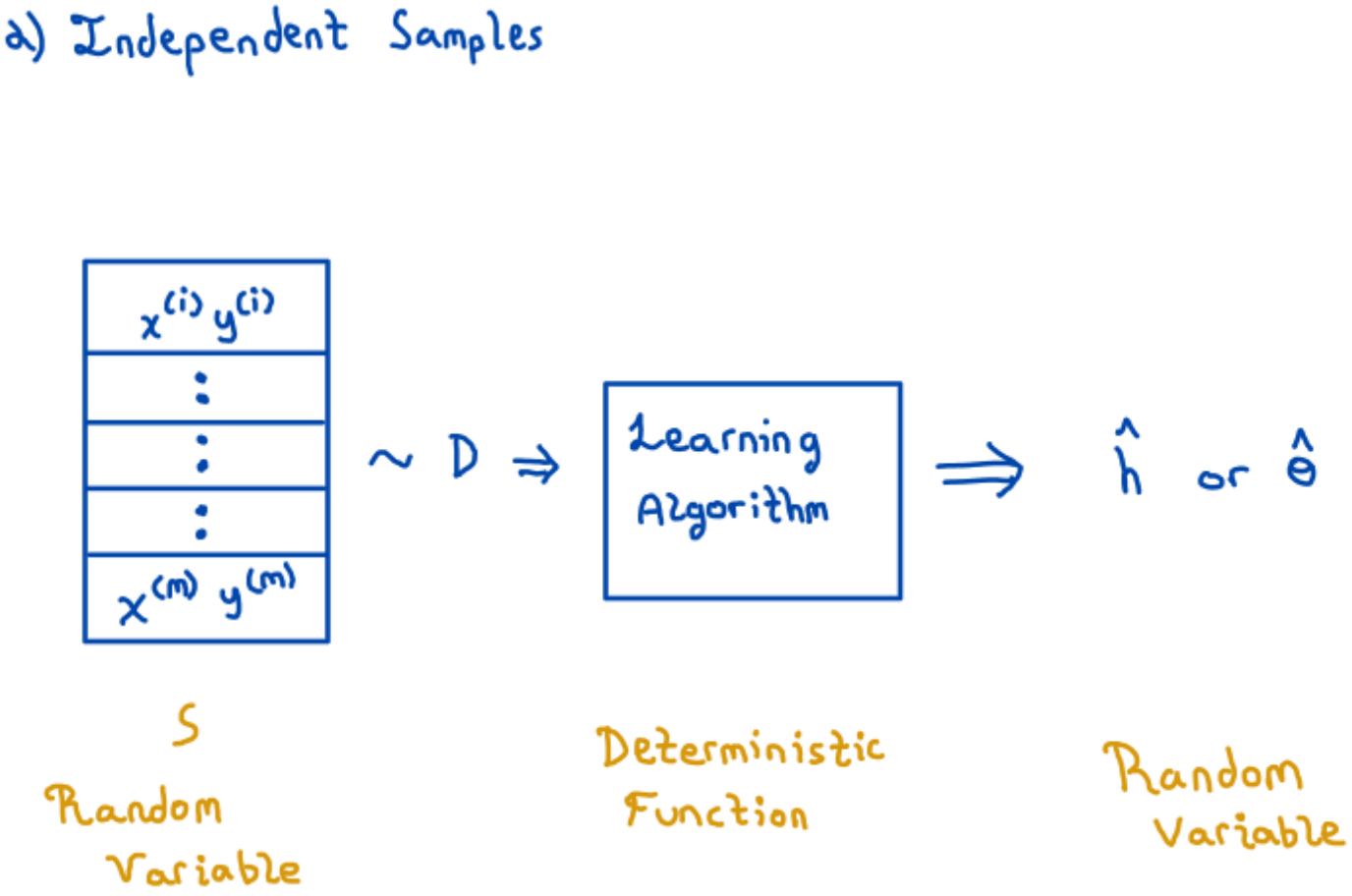
algorithm is what we call a ***hypothesis***



The ***hypothesis*** is a function which accepts an input, a new input **x**, and makes a prediction about **y** for that **x**. This hypothesis is also sometimes in the form of:

So if we restrict ourselves to a class of ***hypotheses***, for example, all possible ***logistic regression*** models of dimension **n**, then its obtaining those parameters is equivalent to obtaining the ***hypothesis function*** itself

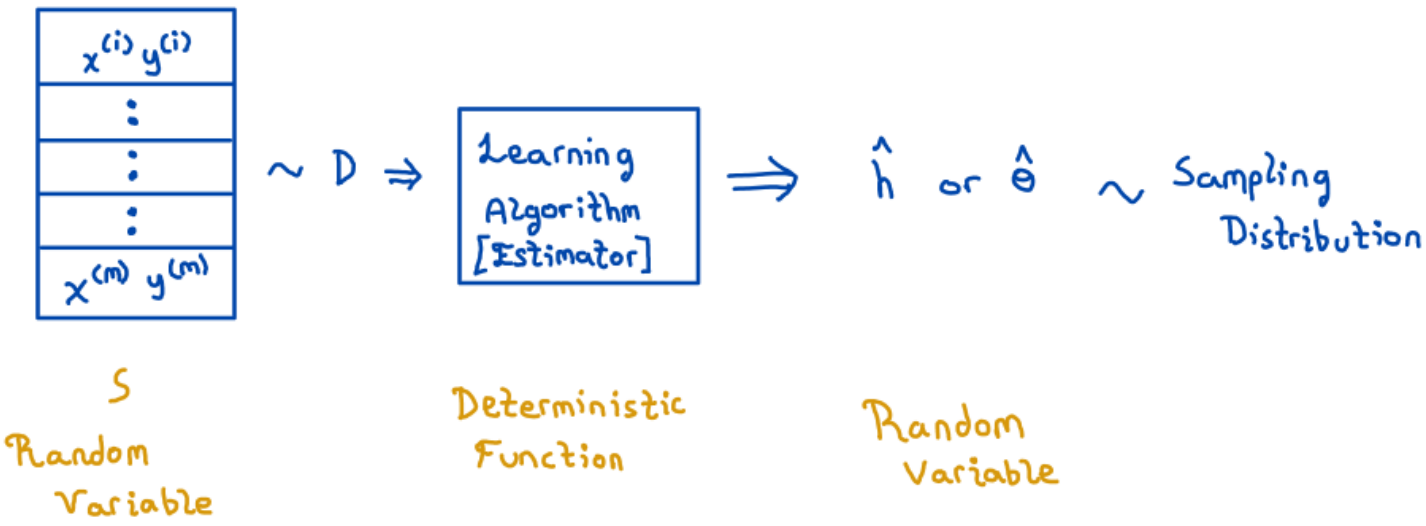
A key thing to note here is that:



All ***random variables*** have a distribution associated with them. The distribution associated with the data is the distribution of **D**, the learning algorithm is a ***fixed deterministic function***, and there is a distribution associated with the parameters we obtain

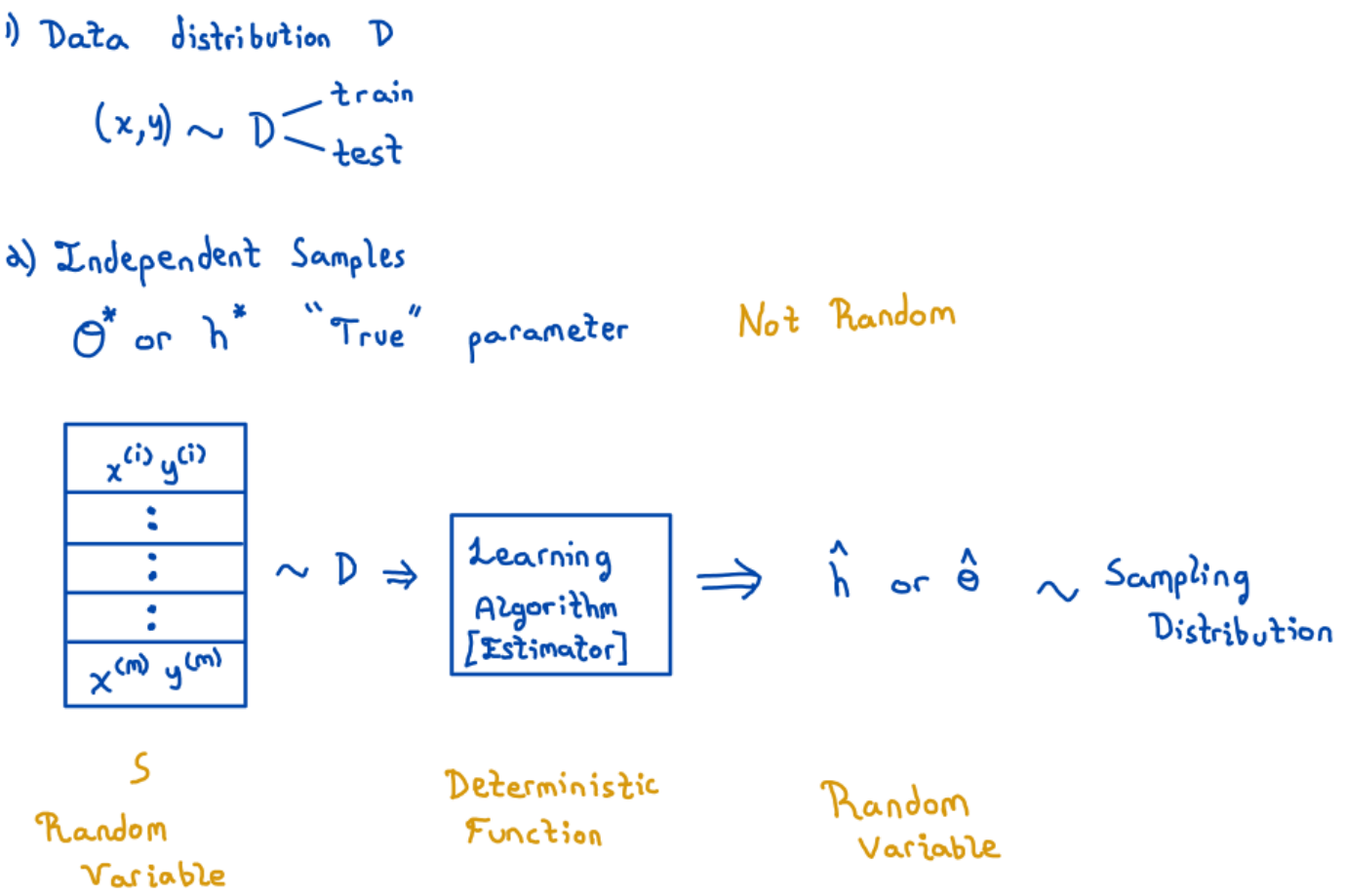
In a more statistical setting, we call this learning algorithm an **Estimator**, and the distribution of **$\hat{\theta}$** is also called the ***sampling distribution***:

2) Independent Samples



What's implied in this process is that there exists some **Θ*** or **h*** which is, in a sense, a ***true parameter*** that we wish to be the output of the learning algorithm. But we never know what **Θ*** or **h*** is and what we get out of the learning algorithm is going to be just a sample from a ***random variable***

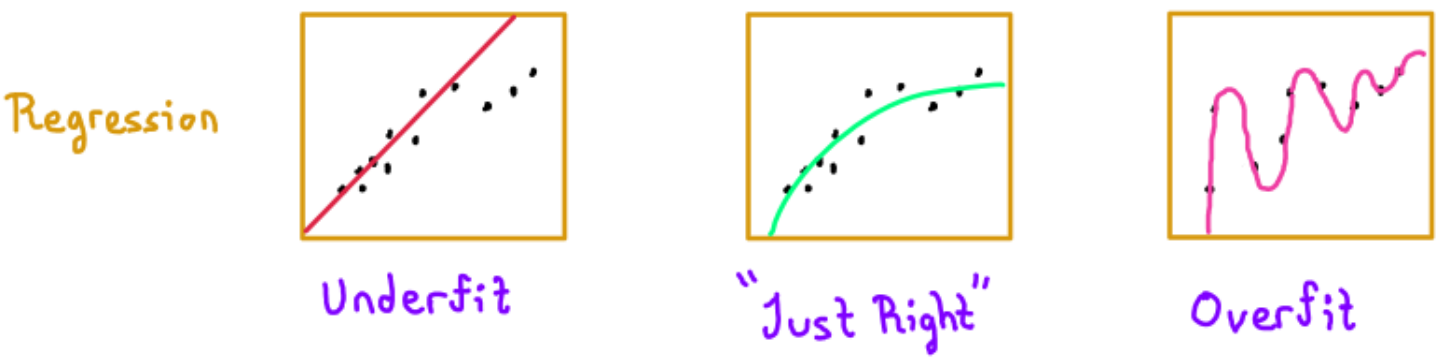
Another thing to note is that the **Θ*** or **h*** is not random, it's just an unknown constant. When we say it's not random it means there is no ***probability distribution*** associated with it. It's just a constant which we don't know. That's the assumption under which you operate



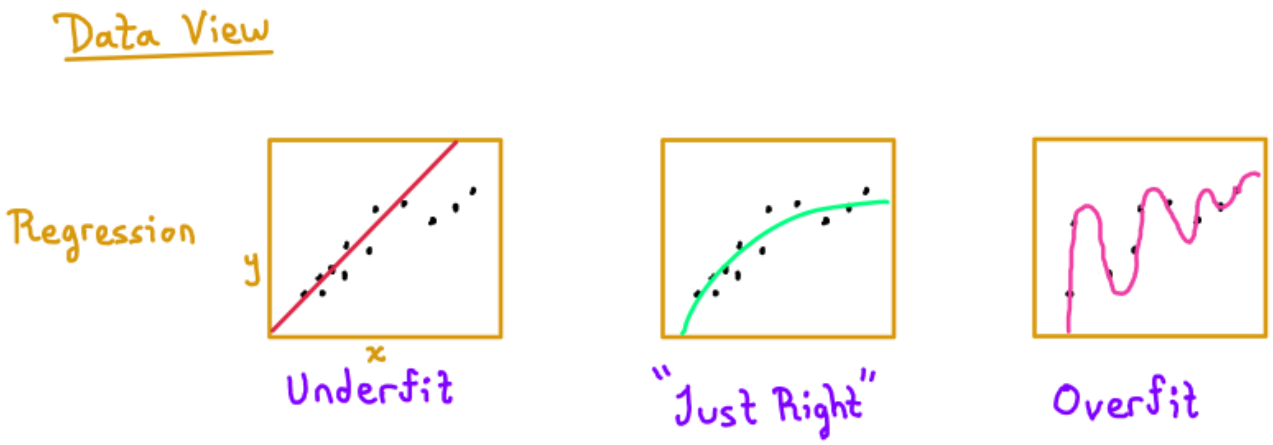
All the entities that we estimate are generally decorated with a ^ on top, which indicates that it's something that we estimated, and anything with a * is the ***true*** or ***right*** answer, which we generally don't have access to

Bias Variance Tradeoff:

In the case of ***Regression***, we saw:



This is how you would view it from the data

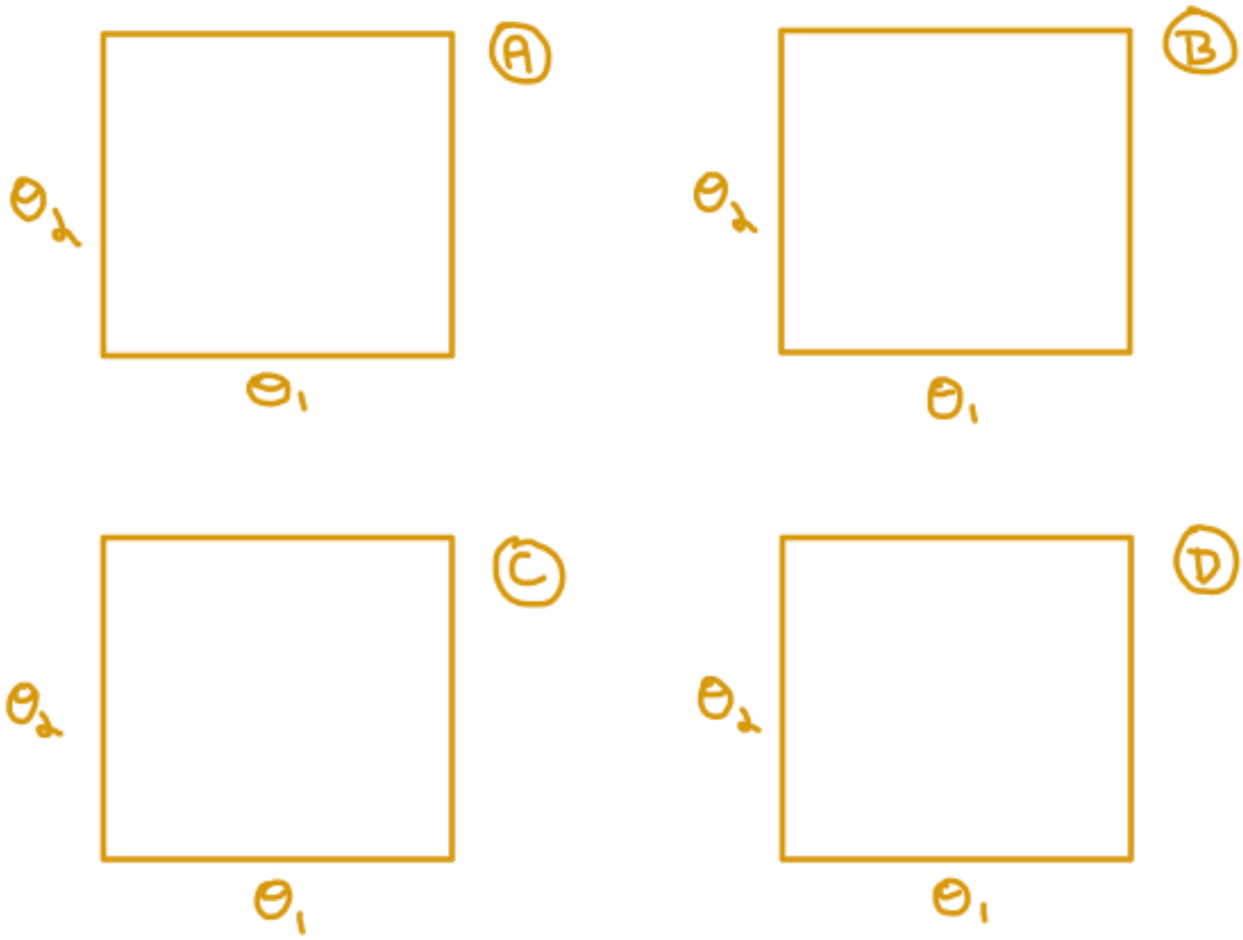


So the concept of ***underfitting*** and ***overfitting*** are, kind of, closely related to ***bias*** and ***variance***

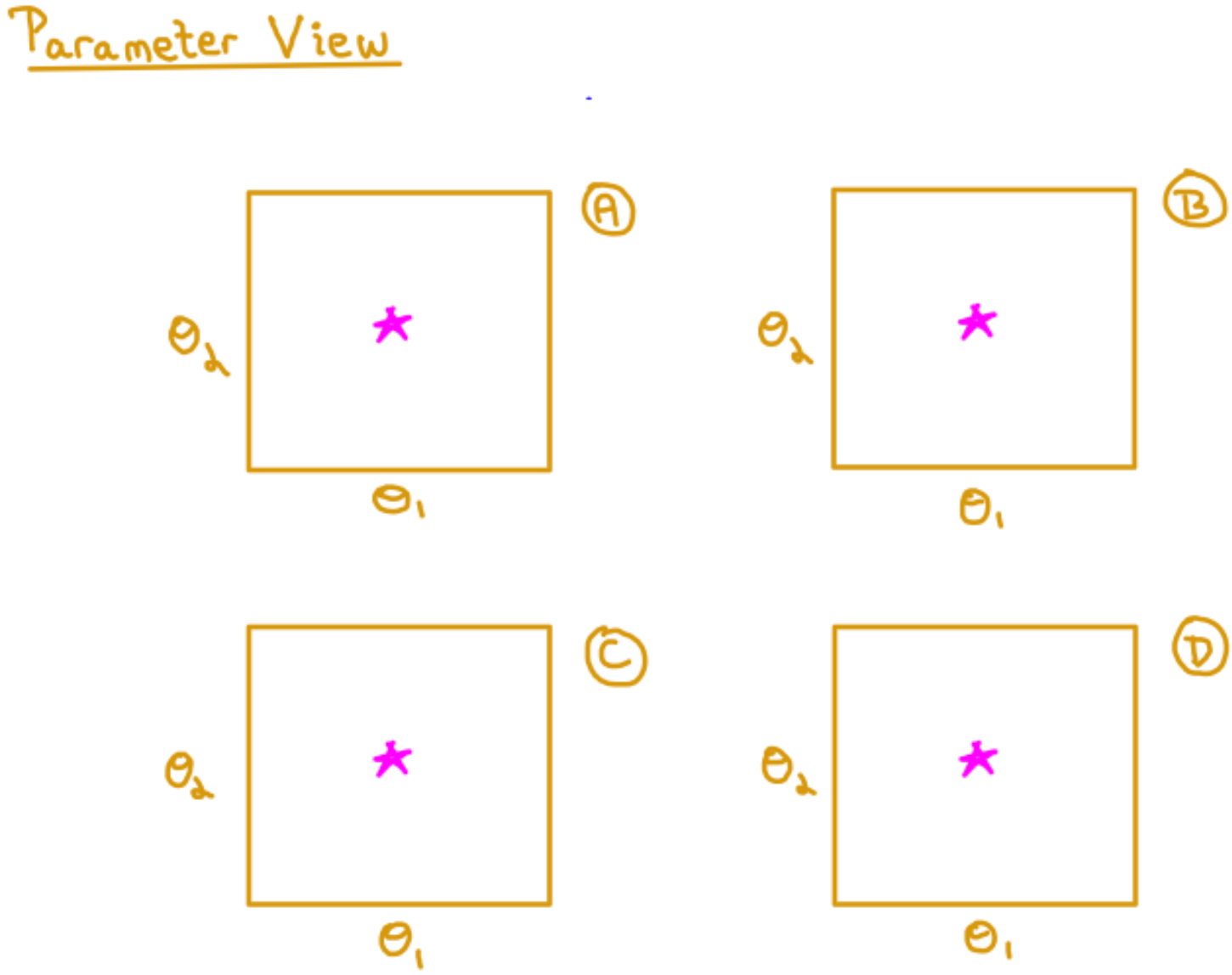
If you look at it from a data point of view, these are the kind of different algorithms you might get. However, to get a more formal view into what's ***bias*** and ***variance***, it's more useful to see it from the parameter view

Let's imagine we have four different learning algorithms with the parameter space Θ_1 , Θ_2

Parameter View



There is also a true **Θ***, which is unknown:

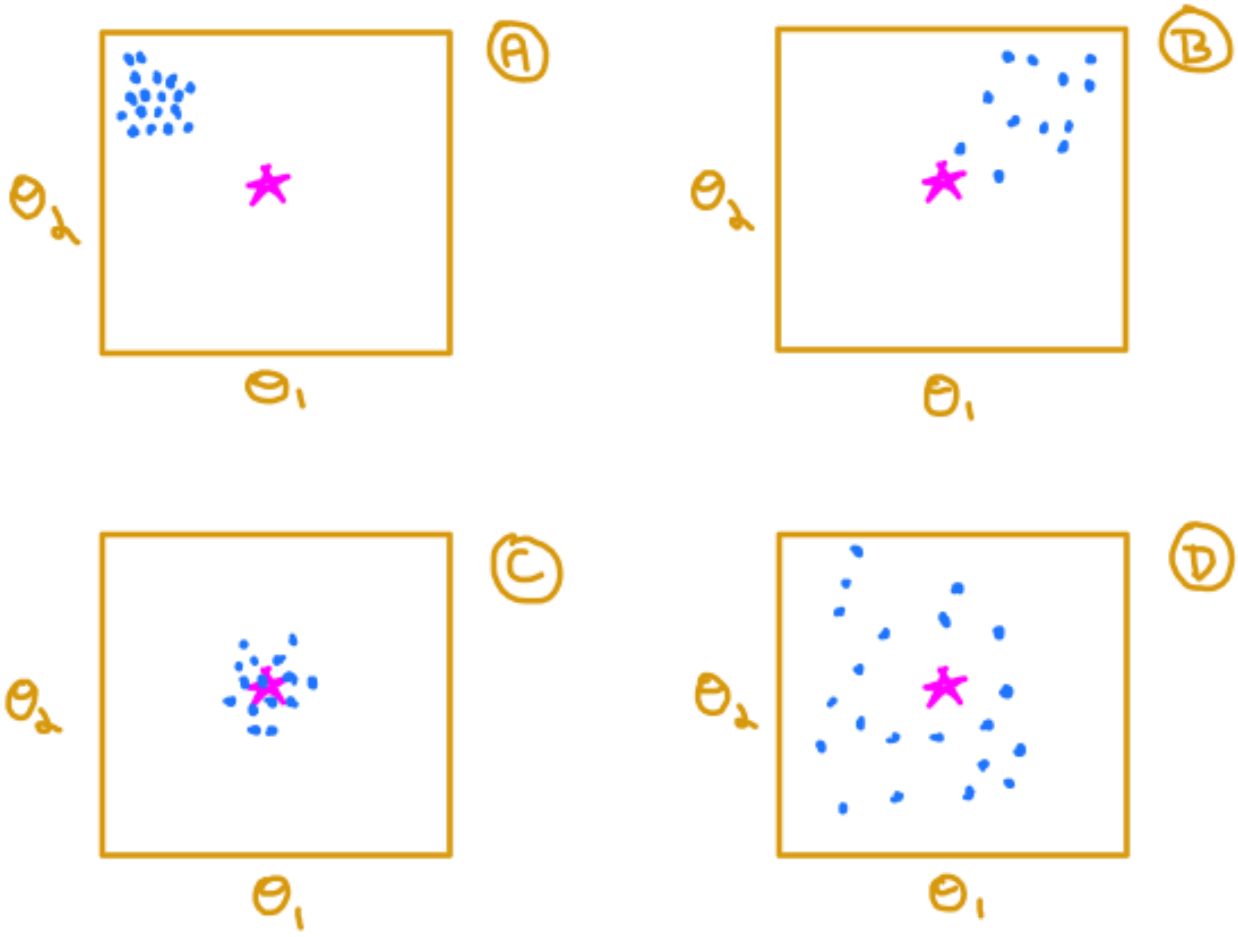


Now, let's imagine we run through this process of sampling **m** examples, running it through the algorithm, and obtain a **Θ^**. Then we start with a new sample from **D** (***Data distribution D***), run it through the algorithm, and we get a different **Θ^**

Θ^ is going to be different for different learning algorithms, so let's imagine we first sample some data (that's our ***training set***), run it through algorithms ***A***, ***B***, ***C***, and ***D***, and repeat this process over and over. The key is that the number of samples per input is **m**

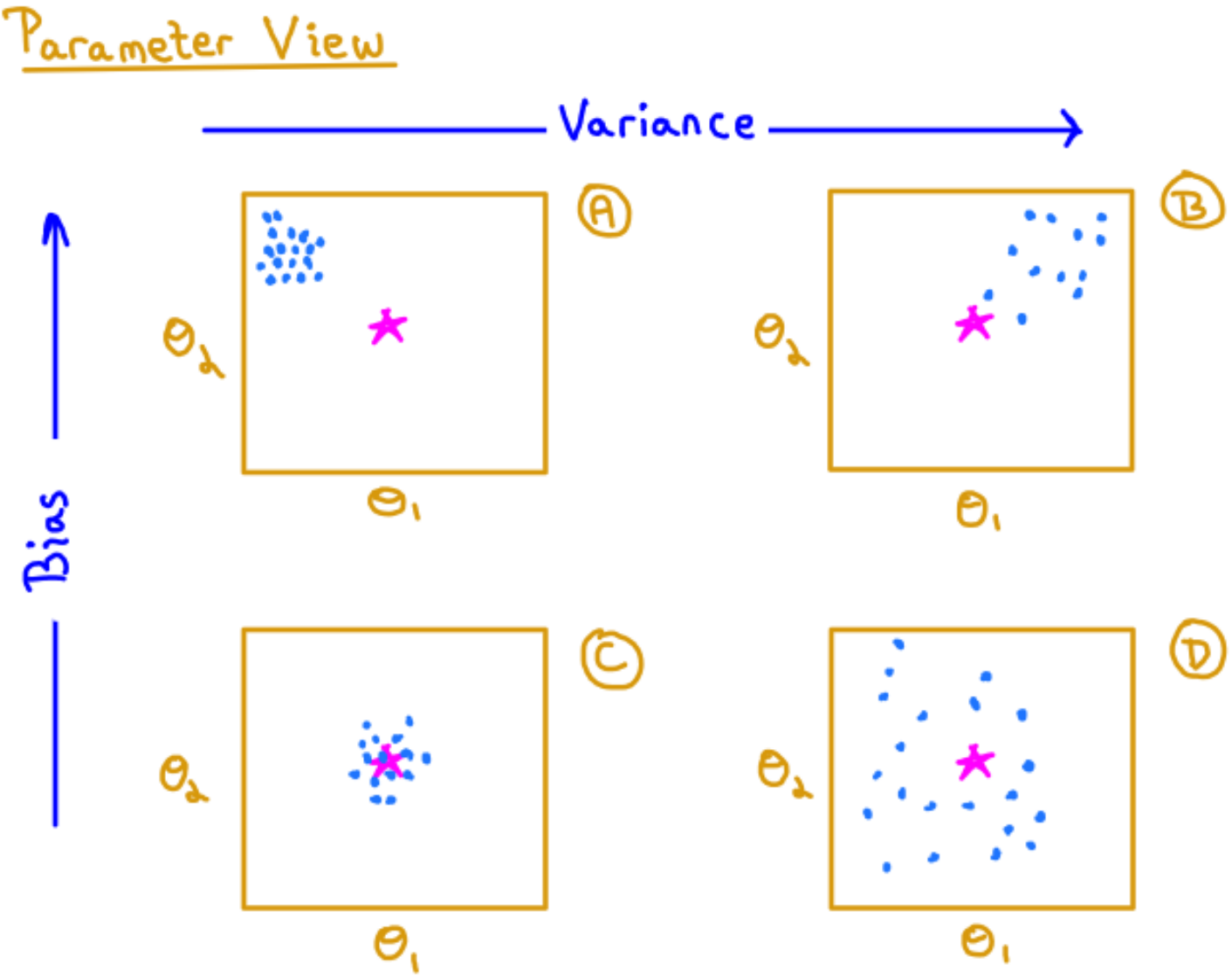
So we're going to repeat this process over and over, and for every time we repeat it, we get a different point:

Parameter View



So each point (each dot) corresponds to a sample of size **m**. The number of points is basically the number of times we repeated the experiment, and what we see is that these dots are basically samples from the ***sampling distribution***

If we were to classify this in terms of ***bias*** and ***variance***:



Bias is basically checking " *Is the* ***sampling distribution*** *kind of centered around the true unknown parameter? "*. ***Variance*** is basically measuring how dispersed the ***sampling distribution*** is. Essentially, ***bias*** and ***variance*** are basically just properties of the first and second moments of your sampling distribution. The first moment (that's the ***mean***) you're asking if it's centered around the ***true parameter***, and the second moment (that's the ***variance***)

Also, different algorithms can have different ***bias*** and ***variance*** even though they have the same number of parameters, for example, if you had regularization, the ***variance*** would come down

A few observations that we want to make is that as we increase the size of the data (if you take a bigger sample for every time we learn), the ***variance*** of **Θ^** would become small. So if we repeat the same thing but with a larger number of examples, all of these points would be more tightly concentrated. So the spread is a function of how many examples we have in each iteration

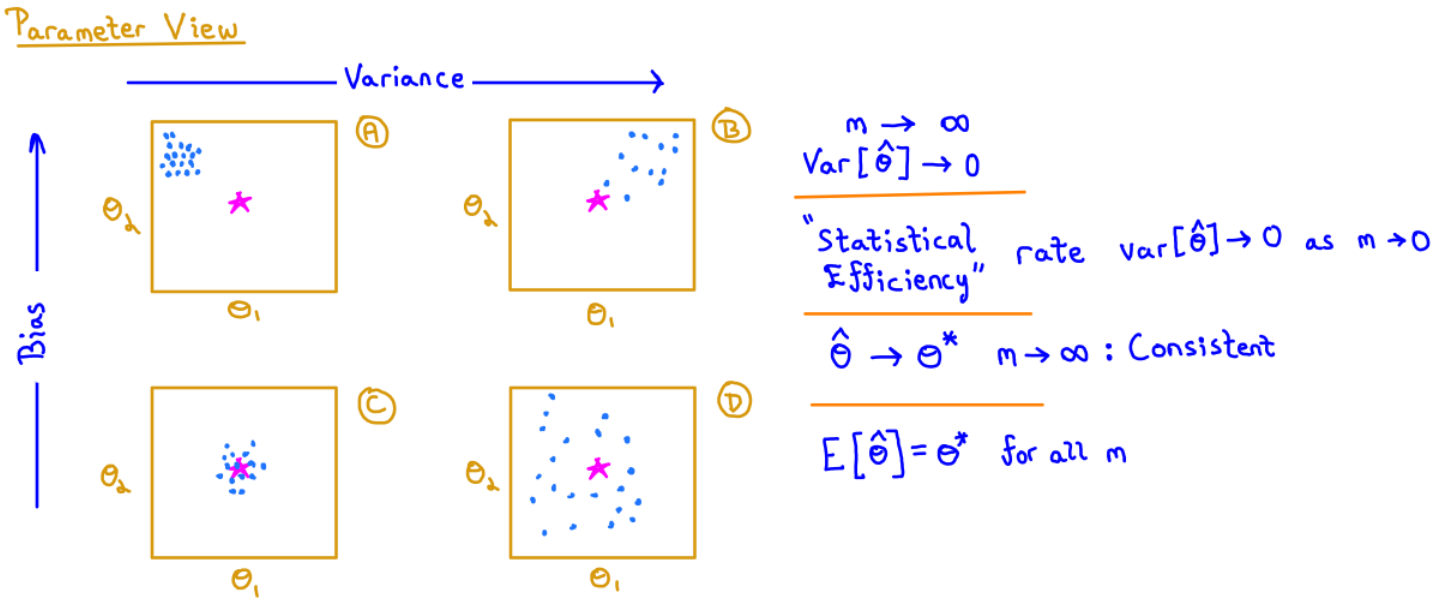
So as **m** → ∞ , **variance** → **0**

If you were to collect an infinite number of samples, run it through the algorithm, you would get some particular **Θ^**, and if you were to repeat that with an infinite number of examples, you'll always keep getting the same **Θ^**

The rate at which the ***variance*** goes to **0** as you increase **m**, can be thought of as what's called "***statistical efficiency***" which is basically a measure of how efficient your algorithm is in squeezing out information from a given amount of data. So ***efficiency*** is basically the rate at which the ***variance*** approaches **0** as **m** approaches **0**

And if **Θ^ → Θ*** as **m → ∞**, you call such algorithms as ***consistent***

And if the ***Expected value*** (**E**) of your **Θ^** is equal to **Θ*** for all **m** (so no matter how big your sample size is), if you always end up with a ***sampling distribution*** that's centered around the ***true parameter***, then your estimator is called an ***unbiased estimator***



Informally speaking, if your algorithm has ***high bias***, it essentially means that no matter how much data or evidence you provided, it kind of always keeps away from **Θ***. You cannot change its mind, no matter how much data you feed it, it's never going to center itself around **Θ***. That's a ***high biased*** algorithm, it's biased away from the ***true parameter***

Variance can be thought of as your algorithm that's kind of highly distracted by the noise in the data and kind of easily swayed far away depending on the noise in your data. So you would call these algorithms as those having ***high variance***

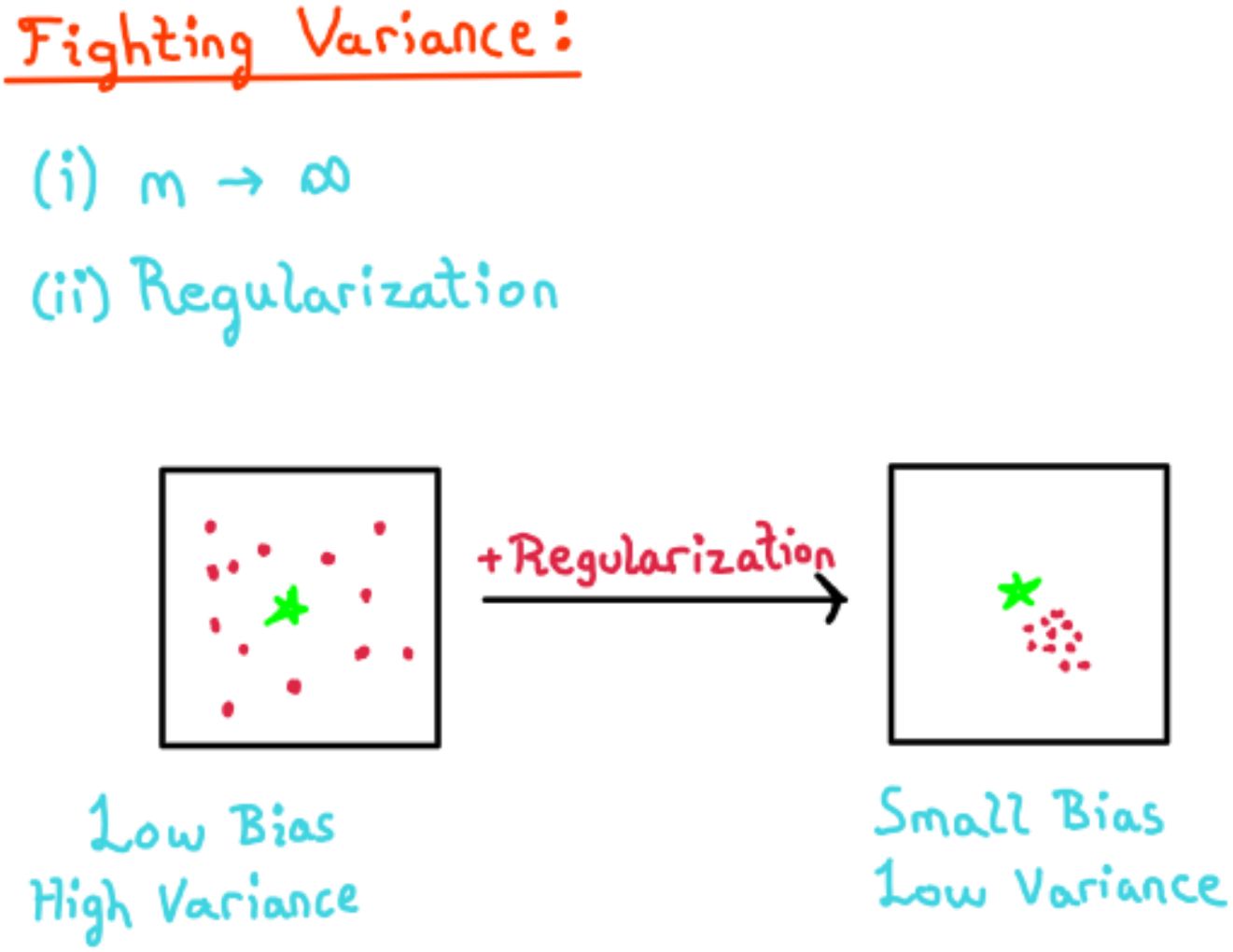
As we are seeing here, ***bias*** and ***variance*** are kind of independent of each other. You can have algorithms that have an independent amount of ***bias*** and ***variance*** in them. There is no correlation between ***bias*** and ***variance***

Bias and ***variance*** are properties of the algorithm at a given size **m**

One way to kind of address if you're in a ***high variance*** situation, is to just increase the amount of data that you have, and that would naturally just reduce the ***variance*** in your algorithm. You don't know upfront whether you're in a ***high bias*** or ***high variance*** scenario. One way to kind of test that is by looking at your ***training performance*** versus ***test performance***

So one way to address ***variance*** is you just get more data. As you get more data, your ***sampling distributions*** tend to get more concentrated

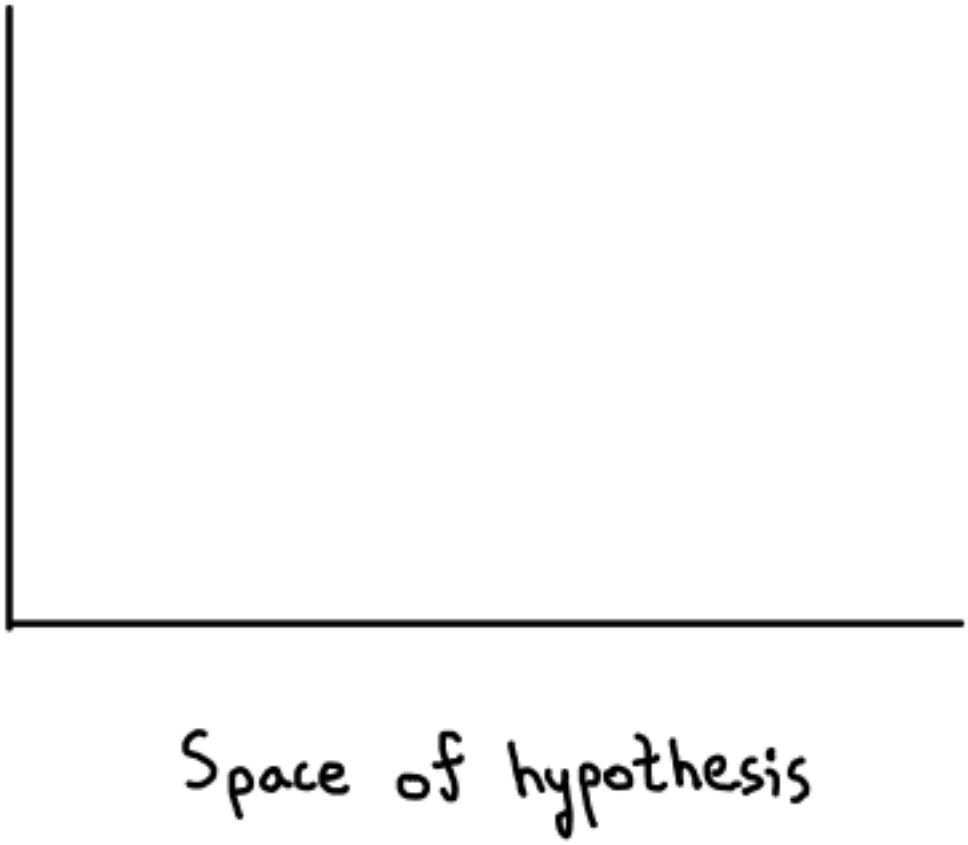
The other was is what's called as ***Regularization***. When you add ***regularization***, like **L_2 regularization** or **L_1 regularization**, what we're effectively doing is you're increasing the ***bias*** and lowering the ***variance***



So if what you care about is your ***predictive accuracy***, you're probably better off trading off ***high variance*** to some ***bias*** and reducing your ***variance*** to a large extent

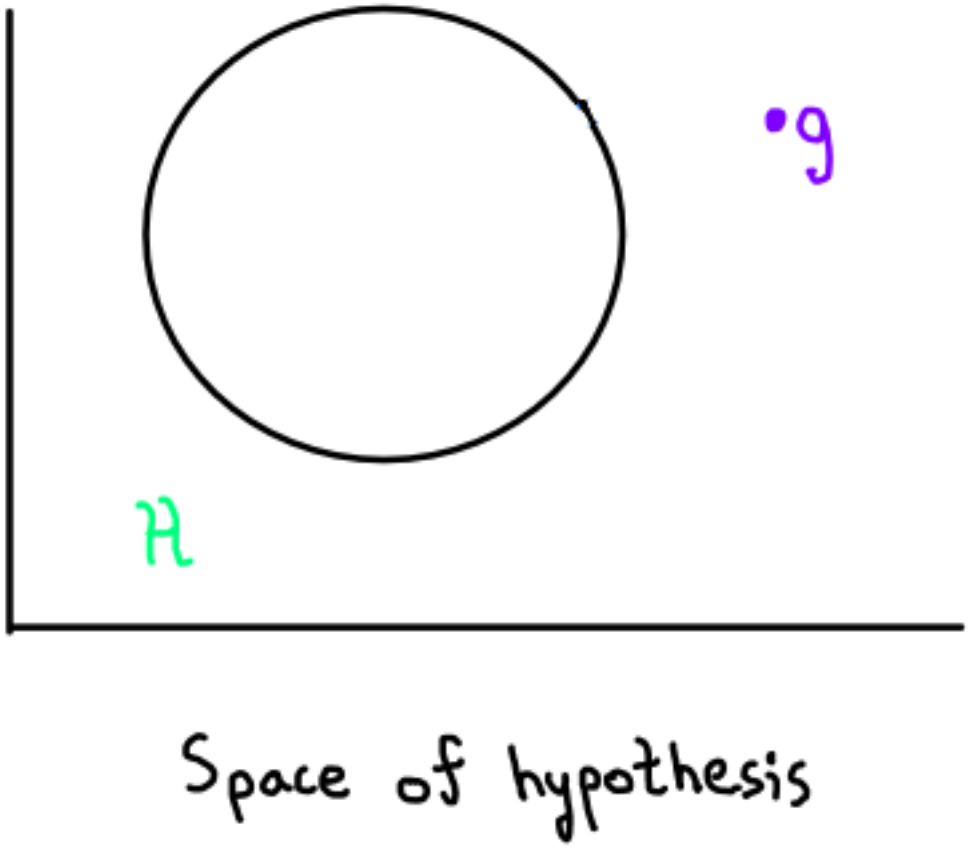
Approximation Error and Estimation Error:

In order to get a better understanding of this, think of this as the ***space of hypothesis***:



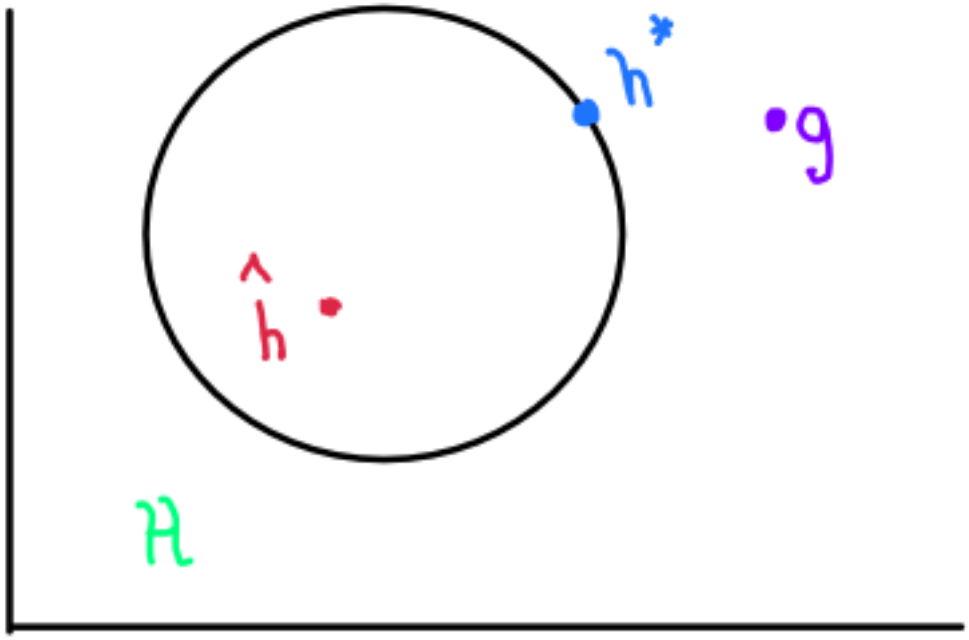
Let's assume there exists a hypothesis **g**, which is the best possible hypothesis you can think of, meaning if you were to take this hypothesis and take the expected value of the loss with respect to the data-generating distribution across an infinite amount of data, you kind of have the lowest error with this hypothesis **g**

Then, there is this class of hypotheses, called **H**. This, for example, can be the set of all ***logistic regression*** hypotheses, or the set of all ***SVM***s

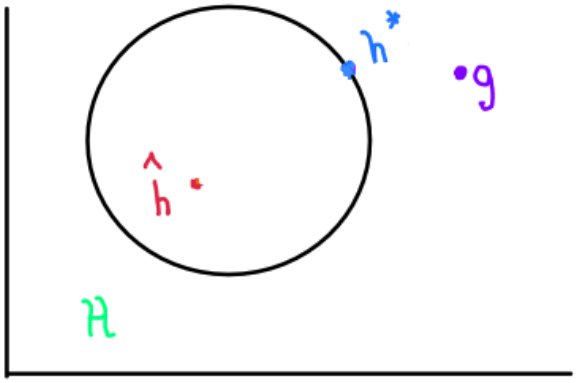


So this is a class of hypotheses and what we end up with when we take a finite amount of data is some member called **h^**. There is also some hypothesis in this class called **h*** which is the best in-class hypothesis. So within the set of all ***logistic regression*** functions, there exists some model which would give you the lowest error if you were to test it on the full data distribution

The best possible hypothesis may not be inside your hypothesis class, it's just some hypothesis that's conceptually something outside the class



Space of hypothesis



Space of hypothesis

g - Best possible hypothesis

h^* - Best in class $\mathcal{H}$

$\hat{h}$ - Learnt from finite data

"epsilon"

↳ $\mathcal{E}(h)$: Risk/Generalization Error

"Expectation" →

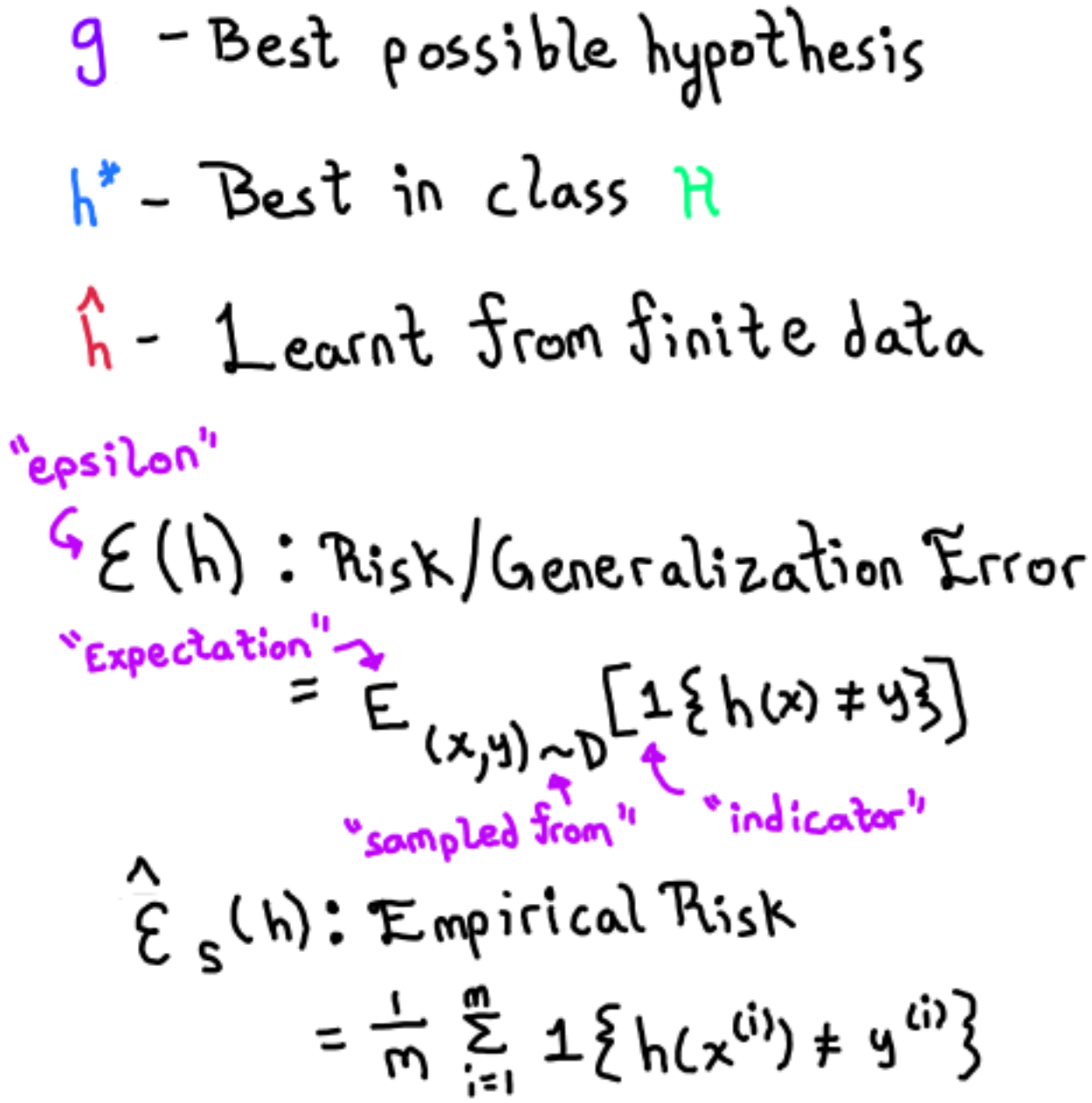
$$= E_{(x,y)\sim D}[1\{h(x) \neq y\}]$$

"sampled from" ↑ "indicator" ↑

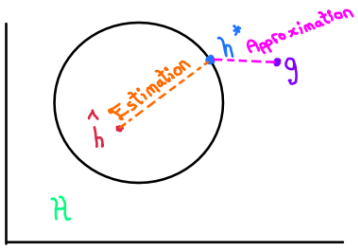
So for the ***Risk/Generalization Error***, you sample examples from the data-generating process, run it through the hypothesis, check whether it matches with your output, and if it doesn't match, you get a **1**, and if it does match, you get a **0**. So on average, roughly speaking, this is the fraction of all examples on which you make a mistake

And here we are kind of thinking about this from a classification point of view to check if the class of your output matches the true class or not, but you can also extend this to the ***regression*** setting, but that's a little harder to analyze

The ***Empirical Risk/Empirical Error***:



The difference between the ***Risk/Generalization Error*** and the ***Empirical Risk/Empirical Error*** is that with the former, it is like an infinite process where you're sampling from **D** forever and calculating the long-term average. Whereas with the latter, you have a finite number that's given to you and "*what's the fraction of examples on which you make an error?*"



g - Best possible hypothesis

h^* - Best in class $\mathcal{H}$

$\hat{h}$ - Learnt from finite data

"epsilon"

$\varepsilon(h)$: Risk/Generalization Error

"Expectation"

$= E_{(x,y)\sim D}[1\{h(x) \neq y\}]$

"sampled from" "indicator"

$\hat{\varepsilon}_s(h)$: Empirical Risk

$= \frac{1}{m}\sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$

$\varepsilon(g)$ = Bayes Error / Irreducible Error

$\varepsilon(h^*) - \varepsilon(g)$ = Approximation Error } class

$\varepsilon(\hat{h}) - \varepsilon(h^*)$ = Estimation Error } data

The ***Bayes Error*** essentially means, if you take the best possible hypothesis, what's the rate at which you make errors?. It can be non-zero. Even if you take the best possible hypothesis ever, that can still make some mistakes, and this is also called the ***irreducible error***

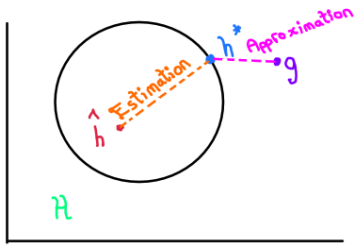
For example, if your data-generating process spits out examples where, for the same **x**, you have different **y**'s in two different examples, then no learning algorithm can do well in such cases. That's just one kind of ***irreducible error***, there can be other kinds of irreducible errors as well

The ***Approximation Error*** essentially means, what is the price that we are paying for limiting ourselves to some class? It's the difference between the best possible error that you can get and the best possible error you can get from **h***. It is an attribute of the class

The ***Estimation Error*** essentially means, what's the error due to estimation

The error on **g** is the ***Bayes error***
The gap between the ***Bayes error*** and the best in class is the ***Approximation error***
The gap between the best in class and the hypothesis that you end up with is called the ***Estimation error***

If you just add them up, all of these cancel out and you're just left with **ε(h^)**



g - Best possible hypothesis

h^* - Best in class $\mathcal{H}$

$\hat{h}$ - Learnt from finite data

"epsilon"

$\varepsilon(h)$: Risk/Generalization Error

"Expectation"

$= E_{(x,y)\sim D}[1\{h(x) \neq y\}]$

"sampled from" "indicator"

$\hat{\varepsilon}_s(h)$: Empirical Risk

$= \frac{1}{m}\sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$

$\varepsilon(g)$ = Bayes Error / Irreducible Error

$\varepsilon(h^*) - \varepsilon(g)$ = Approximation Error } class

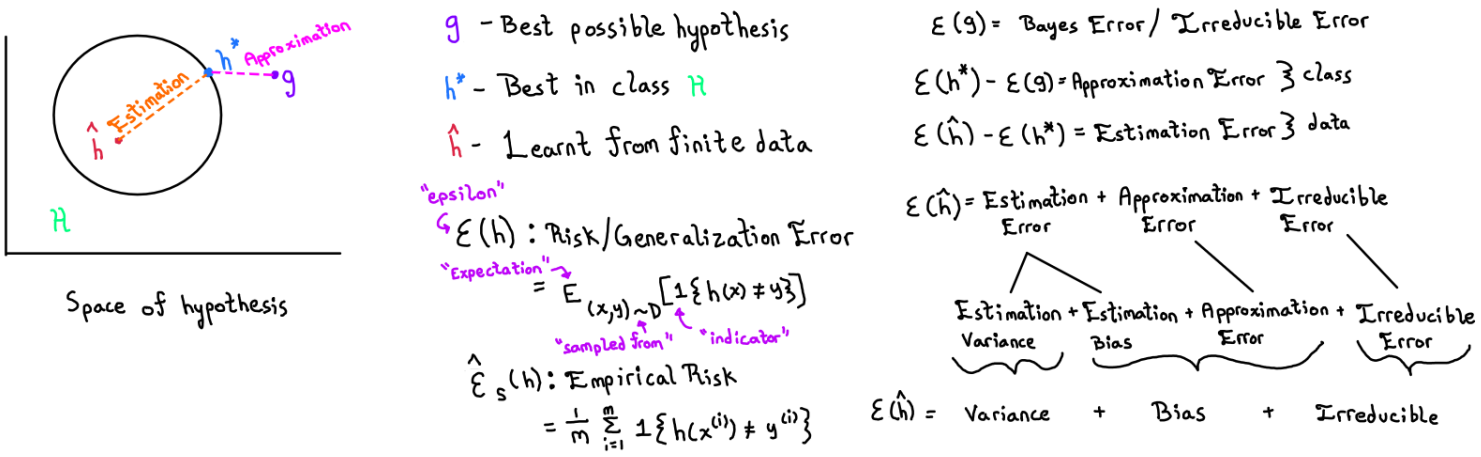
$\varepsilon(\hat{h}) - \varepsilon(h^*)$ = Estimation Error } data

$\varepsilon(\hat{h})$ = Estimation Error + Approximation Error + Irreducible Error

It's kind of useful to think about your ***generalization error*** as different components. Some error which you just cannot reduce no matter what hypothesis you pick, no matter how much training data you have, there's no way you can get rid of the ***irreducible error***. Then you make some decisions about limiting yourself to ***neural networks*** or ***logistic regression*** or whatever, and thereby you're defining a class of all possible models and that has a cost itself, and that's your ***approximation error***. And then you are working with limited data, and with the limited data that you have, and possibly due to some nuances of your algorithm, you also have an ***estimation error***

We can further see that the ***estimation error*** can be broken down into ***estimation variance*** and the ***estimation bias***

And what we commonly call as ***bias*** and ***variance*** are:



So sometimes you see the ***bias-variance decomposition*** and sometimes you see the ***estimation-approximation error decomposition***. They are somewhat related, they're not exactly the same

The ***bias*** is basically trying to capture “why is **h^** far from **g**?” ("why did our hypothesis stay away from the true hypothesis?"). And that could be because your class is too small or it could be due to other reasons, such as regularization

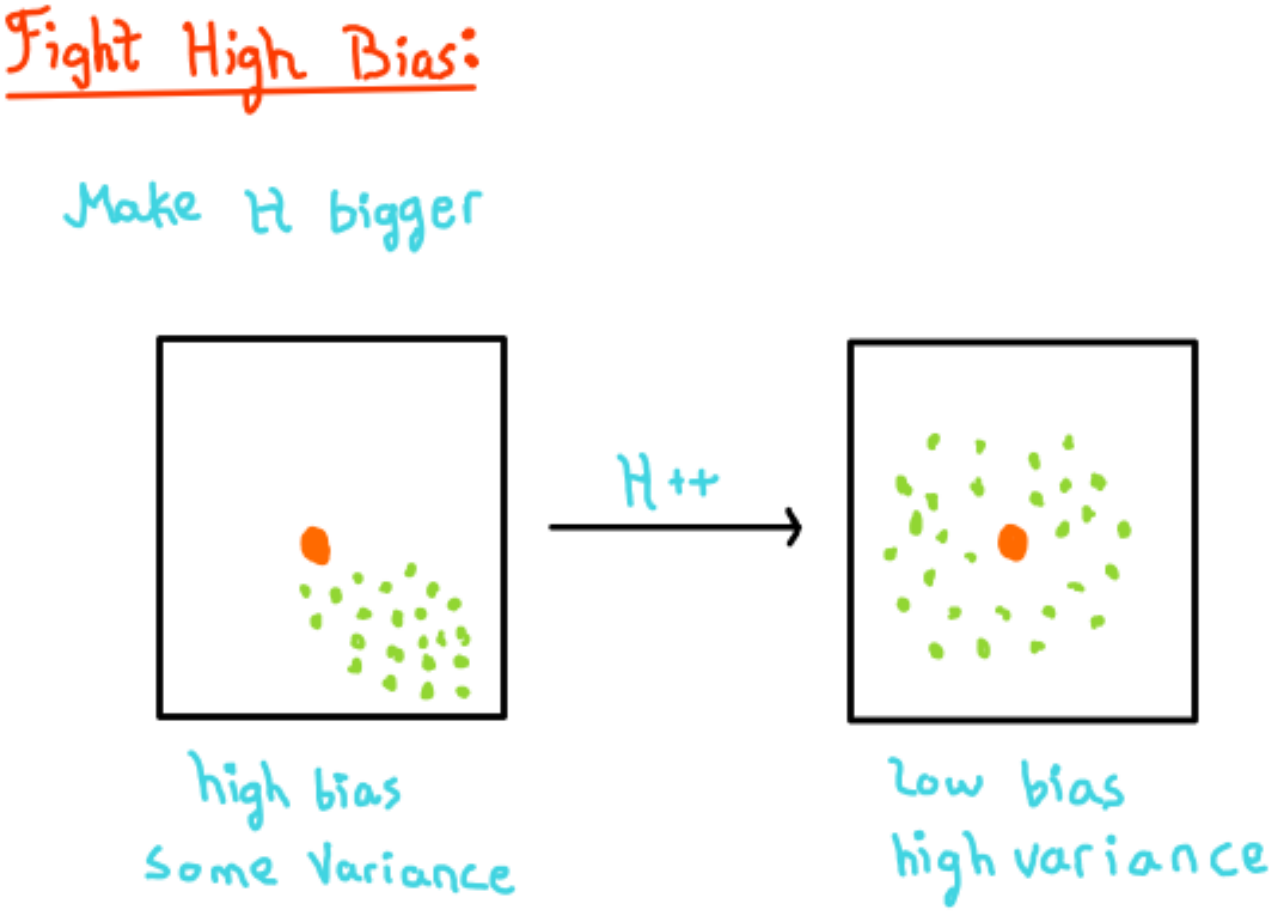
The ***variance*** is generally almost always due to having small data. It could be due to other reasons as well

But these are two different ways of decomposing your error

How do you fight ***high bias***?

One way is to just make **H** bigger. You can also try different algorithms after making your **H** bigger

What this generally means is just by having a bigger class, there is a higher probability that the hypothesis that you estimate can vary a lot



If you reduce the ***space of hypothesis***, you may be increasing your ***bias*** because you may be moving away from **g**, but you're also effectively reducing your ***variance***

So that's one of the trade offs that you observe, that a step that you take, for example, in reducing ***bias*** by making the ***space of hypothesis*** bigger, also makes it possible for your **h^** to land at a wider space and increases your ***variance***. And if you take a step to reduce your ***variance*** by maybe making your class smaller, you may end up making it smaller by being away from the end, thereby increasing your ***bias***

So when you add ***regularization***, you're effectively kind of shrinking the class of hypothesis that you have. You start penalizing those hypotheses whose **Θ** is very large, and in a way you're kind of shrinking the class of hypothesis that you have. So if you shrink the class of hypothesis, your ***variance*** is kind of reduced because there's much smaller wiggle room for your estimator to place your **h^**. And if you shrink it by going away from **g**, you also introduce ***bias***

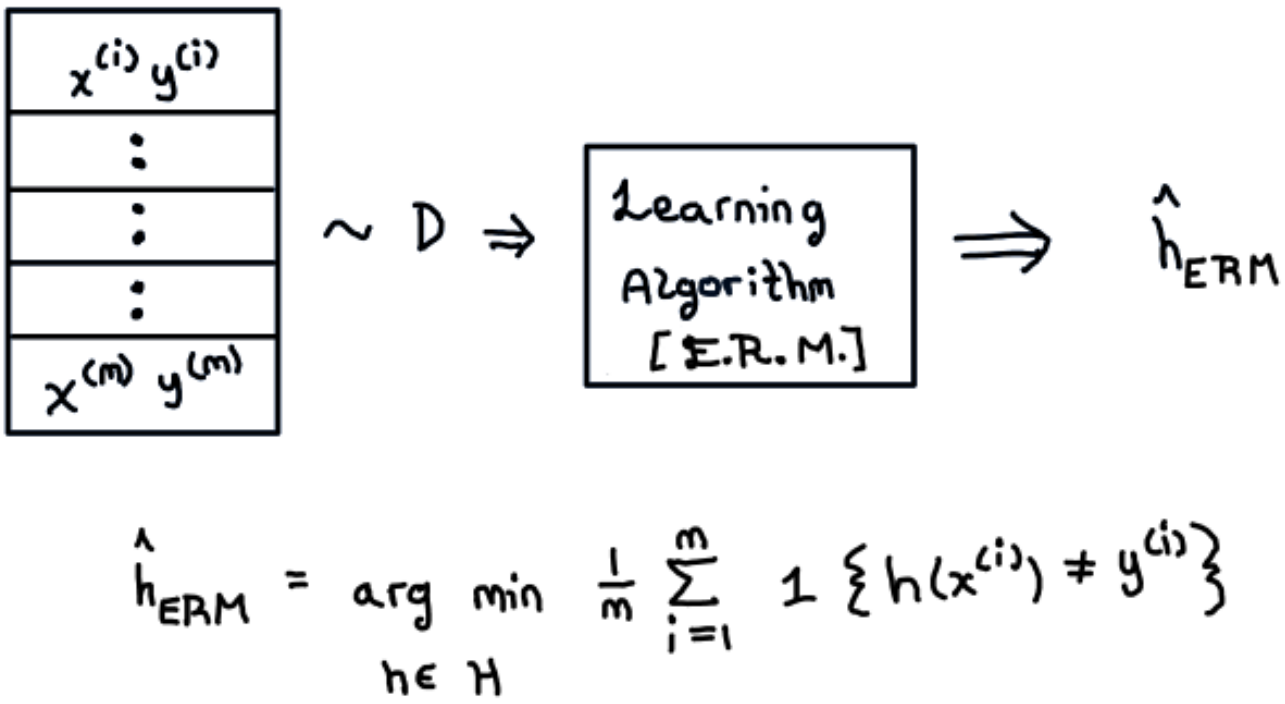
That's the ***bias-variance*** tradeoff

Empirical Risk Minimization:

The ***Empirical Risk Minimizer*** is a learning algorithm

We've been doing this so far. We try to find a minimizer in a class of hypotheses that minimizes the average ***training error***

Empirical Risk Minimization:



$$\hat{h}_{ERM} = \arg\min_{h \in H} \frac{1}{m} \sum_{i=1}^{m} 1\{h(x^{(i)}) \neq y^{(i)}\}$$

So for example, this is trying to minimize the ***training error***. From a classification perspective, this is kind of increasing the training accuracy which is different from what ***logistic regression*** did, where we were doing the maximum likelihood or minimizing the negative log likelihood

It can be shown that losses like the logistic loss can be well approximated by the ***ERM*** and this theory should hold nonetheless

If we are limiting ourselves to that class of algorithms which work by minimizing the training loss, as opposed to something that, say, returns a constant all the time or does something else, if we limit ourselves to ***empirical risk minimizers***, then we can come up with more theoretical results, for example, ***uniform convergence***

Uniform Convergence:

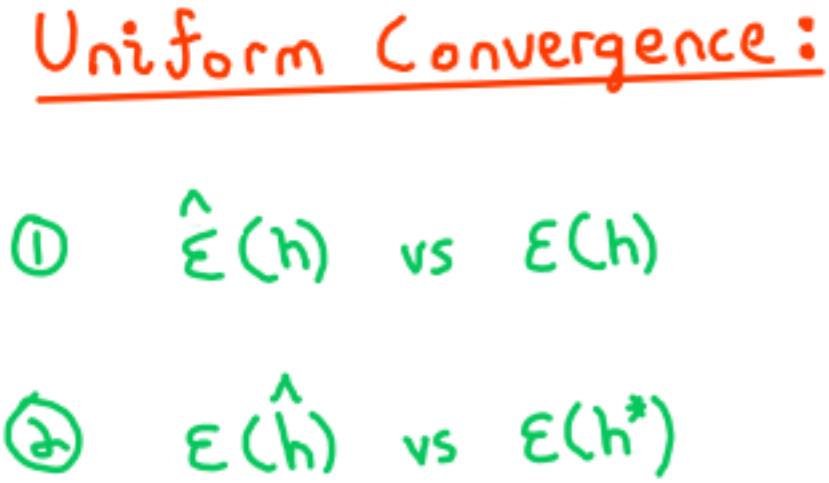
There are two central questions that we are kind of interested in. One question is “If we do ***empirical risk minimization***, that is, if we just reduce the training loss, what does that say about the ***generalization error***?”

Uniform Convergence:

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

So consider some hypothesis that gives you some amount of training error. What does that say about its ***generalization error***?

The second question is “How does the ***generalization error*** of our learned hypothesis compare to the best possible ***generalization error*** in that class”



Note we're only talking about **h*** and not **g**, **h*** is the best in class

These are two central questions that we want to explore, and for this we're going to use our two tools

One is called the ***Union Bound***. If we have **k** different events, then these need not be independent

Uniform Convergence:

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

② $\varepsilon(\hat{h})$ vs $\varepsilon(h^*)$

Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$$

The probability of any one of these events happening is less than or equal to the sum of the probabilities of each of them happening

Then we have a second tool called ***Hoeffding's Inequality***

Uniform Convergence:

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

② $\varepsilon(\hat{h})$ vs $\varepsilon(h^*)$

Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$$

(2) Hoeffding's Inequality

Let $Z_1, Z_2, \ldots, Z_m \sim \text{Bernoulli}(\phi)$

$$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$$

"Gamma" → Let $\gamma > 0$ [margin]

$$P\left[|\hat{\phi} - \phi| > \gamma\right] \leq 2 \exp(-2\gamma^2 m)$$

(error, margin)

Each of the **Z**'s is either a **0** or **1**

The ***Hoeffding's Inequality*** basically says, the probability that the absolute difference between the **estimated ϕ** parameter and the **true ϕ** parameter is greater than some margin **γ**, can be bounded by **2 exp(-2γ^{2}m)**

What it's basically saying is, there is some parameter between **0** and **1** of a ***Bernoulli distribution*** (the fact that it is between **0** and **1** means it's bounded, and that's a key requirement for the ***Hoeffding's Inequality***), and now we take samples from this ***Bernoulli distribution*** and the estimator for this is basically just the averages of your samples

The absolute difference between the ***estimated value*** and the ***true value***, the probability that this difference becomes greater than some margin **γ**, is bounded by the above expression

You'd like the *'error'* (the absolute value of how far away your ***estimated values*** are from the ***true values***) to be small (closer). So you'd probably want your **ϕ^** and **ϕ** to be not more than **0.001**. So in which case, if the absolute value between the ***estimated*** and the ***true parameter*** is greater than **0.001**, if that's the margin that you're interested in, then the ***Hoeffding's Inequality*** proves that if you were to repeat this process over and over and over, the number of times **ϕ^** is going to be farther than **0.001** from the ***true parameter***, is going to be less than the above expression which is a function of **m**

As **m** increases, the above expression becomes smaller, which means the probability of your estimate deviating more than a certain margin, only reduces as you increase **m**

Is **h*** the limit of **h^** as **m → ∞** ?

h* in the limit as **m → ∞**, if it is a consistent estimator. If your estimator is not consistent, then it need not be. So in general, **h^** need not converge to **h*** as you get an infinite amount of data

Now we want to use these tools to answer the central questions

We have hypotheses **h** and error **E**

E(h) is the ***generalization risk*** (or the ***generalization error***) of every possible hypothesis in our class

Pick one hypothesis that's going to be somewhere on the **h** axis, calculate the ***generalization error***, and that's the height of the curve. The dotted line now corresponds to the ***Empirical Error*** for the given **S**

Now, let's sample a set of **m** examples and calculate the ***empirical error*** of all our hypotheses in our class and plot it as a curve

In order to apply Hoeffding's Inequality here, let's consider some $\mathbf{h_i}$. This is some hypothesis (that we don't now. We start with some random hypothesis)

So by starting with some hypothesis, the height of its line, up to the ***generalization error*** curve (the ***generalization error*** of $\mathbf{h_i}$ is the height to the **E(h)** curve), and the height to the dotted curve is $\boldsymbol{\varepsilon\hat{}(h_i)}$ (we're going to ignore the **S** for now)

The $\boldsymbol{\varepsilon\hat{}(h_i)}$ corresponds to the sample that we obtain

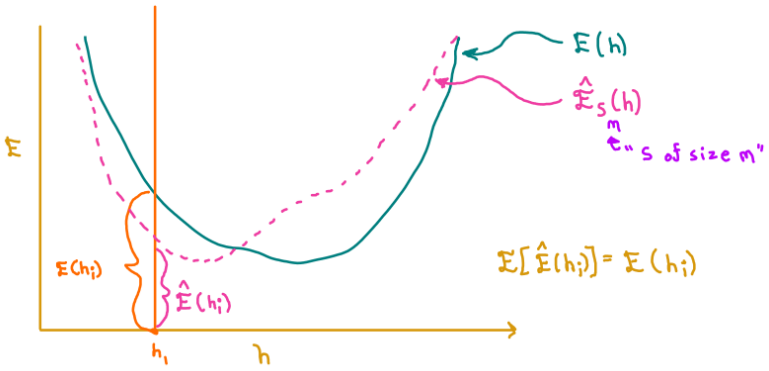
One thing that you can check is that the expected value of $\boldsymbol{\varepsilon\hat{}(h_i)}$ is equal to $\boldsymbol{\varepsilon(h_i)}$, where the expectation is with respect to the data's sample

#!# I messed up, the E is supposed to be Epsilon ε #!#

Uniform Convergence:

① $\hat{E}(h)$ vs $E(h)$

② $E(\hat{h})$ vs $E(h^*)$



Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$

(2) Hoeffding's Inequality

Let $Z_1, Z_2, \ldots, Z_m \sim \text{Bernoulli}(\phi)$

$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$

"Gamma" → Let $\gamma > 0$ [margin]

$P[|\hat{\phi} - \phi| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

error, margin

So what this means is that, for one particular sample, this is the generalization error you got

In general, on average, if you sum the average across all possible training samples that you can get, the expected value of the height to the dotted line is going to be equal to the height to the solid line

Now if you apply ***Hoeffding's Inequality***, you basically get the probability:

#!# I messed up, the E is supposed to be Epsilon ε #!#

Uniform Convergence:

① $\hat{E}(h)$ vs $E(h)$

② $E(\hat{h})$ vs $E(h^*)$

Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$

(2) Hoeffding's Inequality

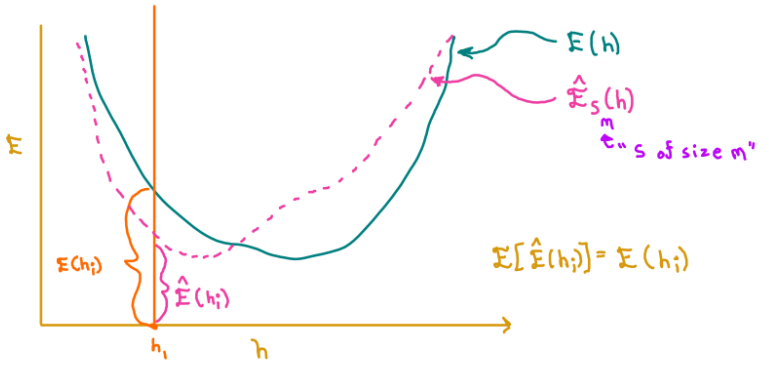
Let $Z_1, Z_2, \ldots, Z_m \sim \text{Bernoulli}(\phi)$

$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$

"Gamma" → Let $\gamma > 0$ [margin]

$P[|\hat{\phi} - \phi| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

error, margin



$P[|\hat{E}(h_i) - E(h_i)| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

So what we are saying is essentially, the gap between the ***generalization error*** and the ***empirical error*** being greater than some margin **γ**, is going to be bounded by this expression. Loosely speaking, what this means is, as we increase the size **m**, if we plot the set of all dotted lines for a larger **m**, they are going to be more concentrated around the solid line

This dotted line corresponds to **S** of some particular size **m**. We could take another sample of a fixed set of examples and that might look something like this:

Uniform Convergence:

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

② $\varepsilon(\hat{h})$ vs $\varepsilon(h^*)$

Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$

(2) Hoeffding's Inequality

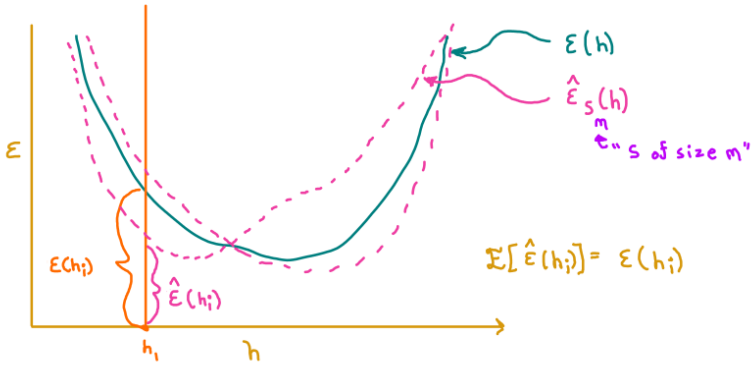
Let $Z_1, Z_2, \ldots, Z_m \sim \text{Bernoulli}(\phi)$

$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$

"Gamma" → Let $\gamma > 0$ [margin]

$P[|\hat{\phi} - \phi| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

error, margin



$P[|\hat{\varepsilon}(h_i) - \varepsilon(h_i)| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

Now, consider the set of all deviations from the solid line to every possible dotted line along the vertical line of $\mathbf{h_i}$. This gap is greater than some margin $\boldsymbol{\gamma}$ with probability less than $\mathbf{2\ exp(-2\gamma^2 m)}$. So it essentially means that if you start plotting dotted lines with a bigger **m**, where the set of all those dotted lines correspond to a bigger **m**, they are going to be much more tightly concentrated around the ***true generalization*** of that edge

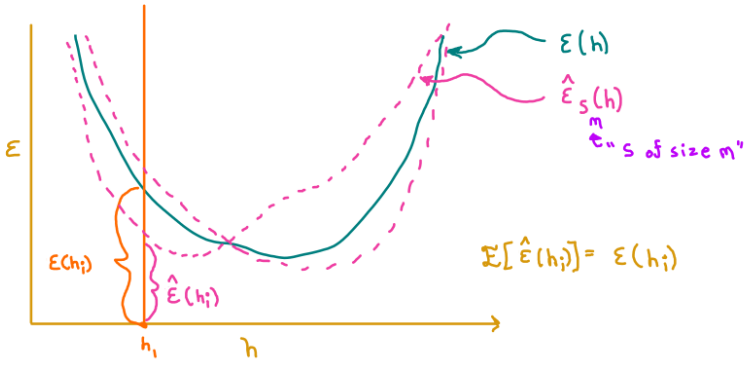
That's good, but there's a problem here. The problem is that we started with some hypotheses and then averaged across all possible data that you could sample, but in practice this is useless, because in practice, we start with some data and run the ***empirical risk minimizer*** to find the lowest **h** for that particular data, which means that **h** and the data that you have are not really independent. You chose the **h** to minimize the risk for the ***empirical risk*** for the particular data that you are given in the first place

To fix this, what we want to do is basically extend this result we got to account for all **h**. Now if we want to get a bound on the gap between the ***probabilistic bound*** and the gap between the ***generalization error*** and the ***empirical error*** for all **h**. This is basically called ***uniform convergence*** because we are trying to see how the ***empirical risk*** curve converges uniformly to the ***generalization risk*** curve

Uniform Convergence:

① $\hat{\varepsilon}(h)$ vs $\varepsilon(h)$

② $\varepsilon(\hat{h})$ vs $\varepsilon(h^*)$



Tools

(1) Union Bound

$A_1, A_2, \ldots, A_k$ (need not be independent)

$P(A_1 \cup A_2 \cup \ldots \cup A_k) \leq P(A_1) + P(A_2) + \ldots + P(A_k)$

(2) Hoeffding's Inequality

Let $Z_1, Z_2, \ldots, Z_m \sim \text{Bernoulli}(\phi)$

$\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} z_i$ ("Gamma")

Let $\gamma > 0$ [margin]

$P[|\hat{\phi} - \phi| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

error, margin

$P[|\hat{\varepsilon}(h_i) - \varepsilon(h_i)| > \gamma] \leq 2 \exp(-2\gamma^2 m)$

$\longleftrightarrow$

"extend to account for all h"

This we showed using ***Hoeffing's Inequality*** and we got this bound for a fixed **h**, but we are interested in getting the bound for any possible **h**. So that's our next step and the way we're going to extend this pointwise result to across all of them is going to look different for two possible cases. One is a case of a finite hypothesis class and the other is gonna be the case for infinite hypothesis classes

First we're gonna assume that the class of **H** has a finite number of hypotheses. The result by itself is not very useful but it's going to be like a building block for the other cases. So let's assume that the number of hypotheses in this class is some number **k**

Basically what we do is we apply the union bound for all **k** hypotheses and we end up just multiplying that by a factor of **k**. So what we get is:

Finite Hypothesis Class $\mathcal{H}$:

$$|\mathcal{H}| = k$$

$$P[\exists h \in \mathcal{H} \mid \hat{\varepsilon}_S(h) - \varepsilon(h) | > \gamma] \leq k \cdot 2 \exp(-2\gamma^2 m)$$

$$\Rightarrow P[\forall h \in \mathcal{H} \mid \hat{\varepsilon}(h) - \varepsilon(h) | < \gamma] \geq 1 - \underbrace{2k \exp(-2\gamma^2 m)}_{\delta \text{ "delta"}}$$

Let $\delta = 2k \exp(-2\gamma^2 m)$

δ - Probability of error

γ - Margin of error

m - Sample size

This is just ***Hoeffding's Inequality*** plus ***Union Bound*** and just negate the two sides

What we basically have now is a relation between **δ**, which is like the ***probability of error*** (the ***empirical risk*** and the ***generalization risk*** are farther than some margin)

What this basically tells us is, if your algorithm is the ***empirical risk minimizer***, it could have been any kind of algorithm, but if it is the kind that minimizes the ***training error***, then you can get, by just changing the sample size, a relation between the ***margin of error*** and the ***probability of error*** and relate it to the sample size

What we can do with this relation is basically fix any two and solve for the third, and that gives us some actionable results. What that could mean is, for example:

Fix $\gamma, \delta > 0$

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

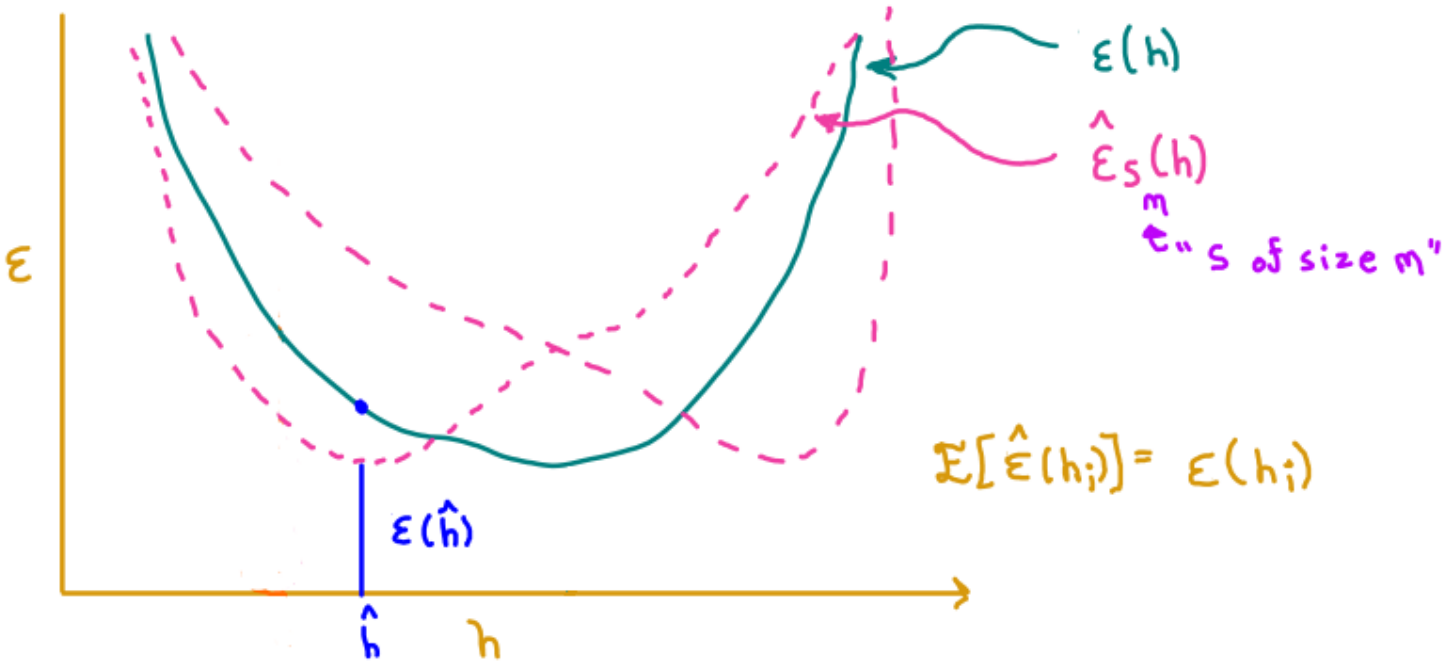
Sample Complexity

So what this means is, with probability at least **1 - δ**, which means probably at least **99%** or **99.9%**, the ***margin of error*** between the ***empirical risk*** and the ***true generalization risk*** is going to be less than **γ** as long as your training size is bigger than this expression

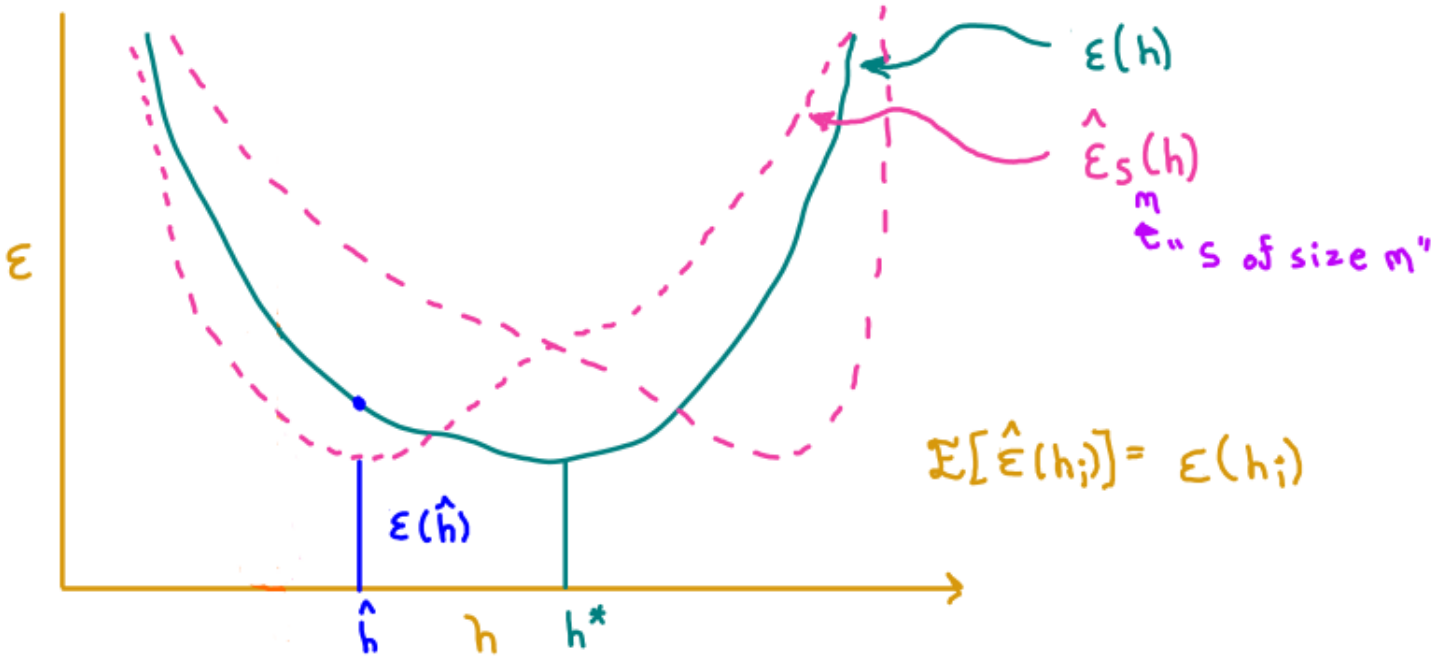
This is also called the ***Sample Complexity*** and basically what this means is, as you increase **m** and you sample different sets of datasets, your dotted lines are going to get closer and closer to the solid line which means minimizing on the dotted line will also get you closer to the ***generalization error***. So this is basically telling you how minimizing on the empirical risk gets you closer to generalization

So we started off with two questions, relating the ***empirical risk*** to ***generalization risk***, now we'll explore the second question. What about the ***generalization error*** of our ***minimizer*** with the best possible in class?

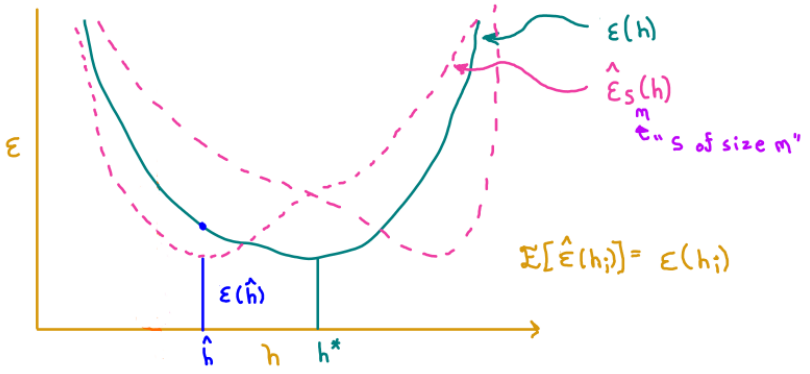
Let's say we started with this dotted curve and the minimizer of that would be **h^**, and this has a particular ***generalization error***:



Let's assume we got this dataset, we ran the empirical risk minimizer, and we obtained this hypothesis. Now how does this compare to the performance of the minimizer of the best in class **h*** ?



Now, we want to get a relation between the **h*** error level and the **h^** error level, and it's pretty straightforward to do that



$$\varepsilon(\hat{h}) \le \hat{\varepsilon}(\hat{h}) + \gamma$$
$$\le \hat{\varepsilon}(h^*) + \gamma$$
$$\le \varepsilon(h^*) + 2\gamma$$

$\Rightarrow$ with probability $1-\delta$, and for training size m,

$$\varepsilon(\hat{h}) \le \varepsilon(h^*) + 2\sqrt{\frac{1}{2m} + \log\frac{2k}{\delta}}$$

The ***empirical error*** of **h^**, by definition, is less than or equal to the ***empirical error*** on any other hypotheses, including the best in class, because this is the ***training error***, not the ***generalization error***

So we wanted the relation between our ***hypothesis generalization error*** to the ***generalization error of the best in class hypotheses***, so we dropped from the ***generalization error*** to the ***empirical error*** of our hypotheses, related that to the ***empirical error*** of the ***best in class***, and again bounded by the gap between these two. So we got a gap between the ***generalization error*** of our hypothesis to ***best in class generalization***

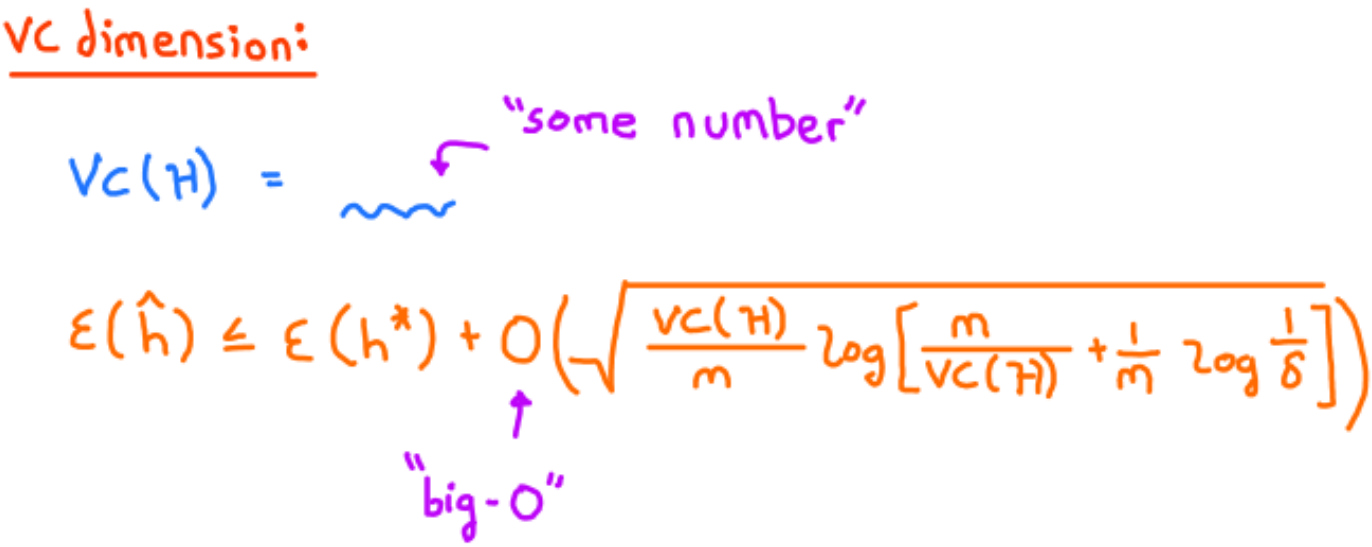
The case for infinite classes is an extension to this

VC dimension:

There is a concept called ***VC dimension*** which you can think of it as trying to assign a size to an infinite-size hypothesis class. For a fixed size hypothesis class, we had **k** to be the size of the hypothesis class

So **VC(H)** is going to be some number which is like the size of the hypothesis class. It's basically telling you how expressive it is

On using the ***VC dimension***, there are very nice geometrical meanings of ***VC dimension***. You can get a similar bound, but now it's not for finite classes anymore



The key takeaway from this is that the number of data examples, that the sample complexity that you want is generally an order of the ***VC dimension*** to get good results. That's basically the main result from that