

Introduction to Statistical Learning Theory

Duke Course Notes Cynthia Rudin

Credit: A large part of this lecture was taken from an introduction to learning theory of Bousquet, Boucheron, Lugosi.

Now we are going to study, in a probabilistic framework, the properties of learning algorithms. At the beginning of the course, I told you that it was important for our models to be “simple” in order to be able to *generalize*, or *learn* from data. I didn’t really say that precisely before, but in this lecture I will.

$$\boxed{\text{Generalization} = \text{Data} + \text{Knowledge}}$$

Finite data can only go so far in producing a good model. It cannot replace knowledge. Knowledge allows you to restrict to a simpler set of models to work with.

Perhaps surprisingly, there is no one universal right way to measure simplicity or complexity of a set of models - simplicity is not an absolute notion. But we’ll give at least one precise way to measure this. And we’ll show how our ability to learn depends on the simplicity of the models. So we’ll make concrete (via proof) this philosophical argument that learning somehow needs simplicity.

In classical statistics, the number of parameters in the model is the usual measure of complexity. Here we’ll use other complexity measures, namely the Growth Function and VC dimension (which is a beautiful combinatorial quantity).

Assumptions

Training and test data are drawn iid from the same distribution. If there’s no relationship between training and test, there’s no way to learn of course. (That’s like trying to predict rain in Africa next week using data about horse-kicks in the Prussian war, so we have to make some assumption.)

Each learning algorithm encodes specific knowledge (or a specific assumption, perhaps about what the optimal classifier must look like) and works best when this assumption is satisfied by the problem to which it is applied.

Notation

Input space \mathcal{X} , output space $\mathcal{Y} = \{-1, 1\}$, unknown distribution D on $\mathcal{X} \times \mathcal{Y}$. We observe n iid pairs $\{(x_i, y_i)\}_{i=1}^n$ drawn iid from D . The goal is to construct a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that predicts y from x .

We would like the true risk to be as small as possible, where the *true risk* is:

$$R^{\text{true}}(f) := \mathbb{P}_{(X,Y) \sim D}(f(X) \neq Y) = \mathbb{E}_{(X,Y) \sim D}[\mathbf{1}_{f(X) \neq Y}].$$

Did you recognize this nice thing that comes from the definition of expectation and probability? We can flip freely between notation for probability and expectation.

$$\begin{aligned} \mathbb{E}_{Z \sim D} \mathbf{1}_{[Z=\text{blah}]} &= 1 \times P(\text{outcome 1, when } Z=\text{blah}) \\ &\quad + 0 \times P(\text{outcome 0, when } Z \neq \text{blah}) \\ &= \mathbb{P}_{Z \sim D}(Z = \text{blah}). \end{aligned}$$

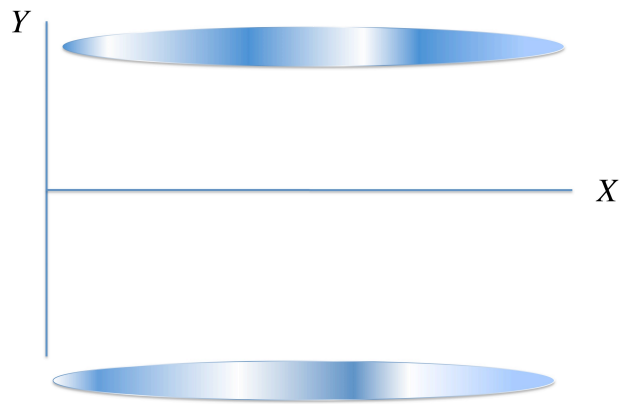
We introduce the *regression function*

$$\eta(x) = \mathbb{E}_{(X,Y) \sim D}(Y|X = x)$$

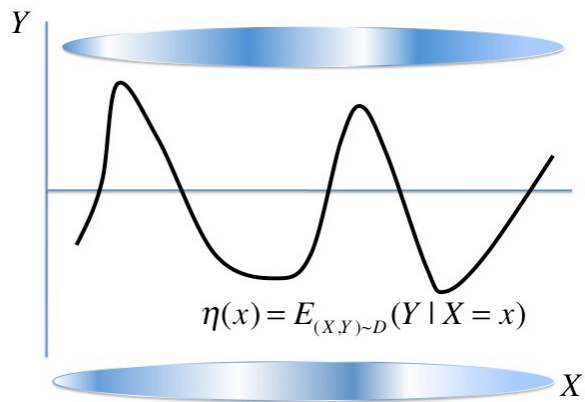
and the *target function* (or Bayes classifier)

$$t(x) = \text{sign } \eta(x).$$

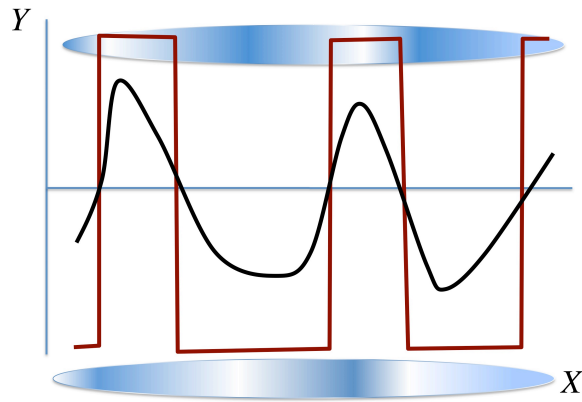
Think of the distribution D , which looks sort of like this:



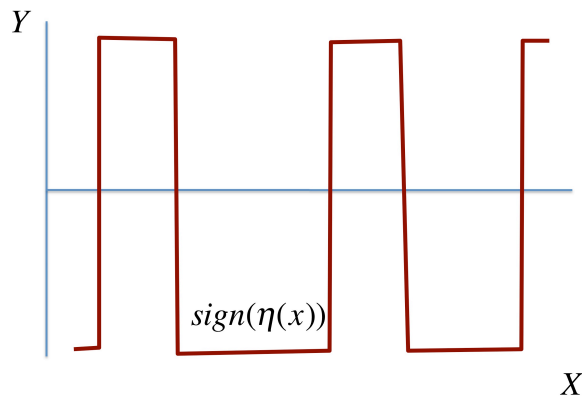
Here's the function η , which is $\mathbb{E}_{(X,Y) \sim D}(Y|X = x)$:



Now take the sign of it:



And that's t :



The target function achieves the minimum risk over all possible measurable functions:

$$R^{\text{true}}(t) = \inf_f R^{\text{true}}(f).$$

We denote the value $R^{\text{true}}(t)$ by R^* , called the *Bayes Risk*.

Our goal is to identify this function t but since D is unknown, we cannot evaluate t at any x .

The *empirical risk* that we can measure is:

$$R^{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{[f(x_i) \neq y_i]}.$$

Algorithm

Most of the calculations don't depend on a specific algorithm, but you can think of using regularized empirical risk minimization.

$$f_n \in \operatorname{argmin}_{f \in \mathcal{F}} R^{\text{emp}}(f) + C \|f\|^2$$

for some norm. The regularization term will control the complexity of the model to prevent overfitting. The class of functions that we're working with is \mathcal{F} .

Bounds

Remember, we can compute f_n and $R^{\text{emp}}(f_n)$, but we cannot compute things like $R^{\text{true}}(f_n)$.

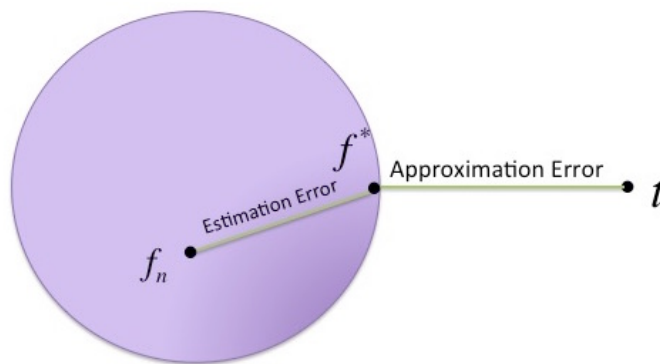
The algorithm chooses f_n from the class of functions \mathcal{F} . Let us call the best function in the class f^* , so that

$$R^{\text{true}}(f^*) = \inf_{f \in \mathcal{F}} R^{\text{true}}(f).$$

Then, I would like to know how far $R^{\text{true}}(f_n)$ is from R^* . How bad is the function we chose, compared to the best one, the Bayes Risk?

$$\begin{aligned} R^{\text{true}}(f_n) - R^* &= [R^{\text{true}}(f^*) - R^*] + [R^{\text{true}}(f_n) - R^{\text{true}}(f^*)] \\ &= \text{Approximation Error} + \text{Estimation Error} . \end{aligned}$$

The Approximation Error measures how well functions in \mathcal{F} can approach the target (it would be zero if $t \in \mathcal{F}$). The Estimation Error is a random quantity (it depends on data) and measures how close is f_n to the best possible choice in \mathcal{F} . Here is the picture for that.



Figuring out the Approximation Error is usually difficult because it requires knowledge about the target, that is, you need to know something about the distribution D . In Statistical Learning Theory, generally there is no assumption made about the target (such as its belonging to some class), and we do not study approximation error. By the way, this is probably the main reason why this theory is so important - it does not require any knowledge of the distribution D .

Also, even if the empirical risk converges to the Bayes risk as n gets large (the algorithm is *consistent*), it turns out that the convergence can be arbitrarily slow if there is no assumption made about the target. On the other hand, the rate of convergence of the Estimation Error can be computed without any such assumption. We'll focus on one of the terms in the estimation error. In particular, we would like to understand how bad the true risk of our algorithm's output, $R^{\text{true}}(f_n)$, could possibly be. We want this to be as small as possible of course, but we can't compute it.

We'll look more carefully at $R^{\text{true}}(f_n)$:

$$R^{\text{true}}(f_n) = R^{\text{emp}}(f_n) + [R^{\text{true}}(f_n) - R^{\text{emp}}(f_n)], \quad (1)$$

where remember we can measure $R^{\text{emp}}(f_n)$.

We could upper bound the term $R^{\text{true}}(f_n) - R^{\text{emp}}(f_n)$, to make something like this:

$$R^{\text{true}}(f_n) \leq R^{\text{emp}}(f_n) + \text{Stuff}(n, \mathcal{F}).$$

The "Stuff" will get more interesting as this lecture continues.

A Bound for One Function f

Let's define the loss g corresponding to a function f . The loss at point (x, y) is:

$$g(x, y) = \mathbf{1}_{f(x) \neq y}.$$

Given \mathcal{F} , define the *loss class*, which contains all of the loss functions coming from \mathcal{F} .

$$\mathcal{G} = \{g : (x, y) \rightarrow \mathbf{1}_{f(x) \neq y} : f \in \mathcal{F}\}.$$

So g doesn't look at predictions f , instead it looks at whether the predictions were correct. Notice that \mathcal{F} contains functions with range in $\{-1, 1\}$ while \mathcal{G} contains functions with range $\{0, 1\}$.

There's a bijection between \mathcal{F} and \mathcal{G} . You can go from an f to its g by $g(x, y) = \mathbf{1}_{f(x) \neq y}$. You can go from a g to its f by saying that if $g(x, y) = 1$ then set $f(x) = -y$, otherwise set $f(x) = y$. We'll use the g notation whenever we're bounding the difference between an empirical average and its mean because the notation is slightly simpler.

Define this notation:

$$P^{\text{true}}g = \mathbb{E}_{(X,Y) \sim D}[g(X, Y)] \quad (\text{true risk again})$$

$$P^{\text{emp}}g = \frac{1}{n} \sum_{i=1}^n g(X_i, Y_i) \quad (\text{empirical risk again})$$

so that we have another way to write the true risk and empirical risk directly in terms of the loss. P^{emp} is called the *empirical measure* associated to the training sample. It just computes the average of a function at the training points. Remember, we are interested in the difference between the true risk and empirical risk, same thing as in the right side of (1), which we're going to upper bound:

$$P^{\text{true}}g_n - P^{\text{emp}}g_n. \tag{2}$$

(g_n is the loss version of f_n .)

Hoeffding's Inequality

For convenience we'll define $Z_i = (X_i, Y_i)$ and $Z = (X, Y)$, and probabilities will be taken with respect to $Z_1 \sim D, \dots, Z_n \sim D$ which we'll write $\mathbf{Z} \sim D^n$.

Let's rewrite the quantity we're interested in, for a general g this time:

$$P^{\text{true}}g - P^{\text{emp}}g = \mathbb{E}_{\mathbf{Z} \sim D^n}[g(Z)] - \frac{1}{n} \sum_{i=1}^n g(Z_i).$$

It's a difference between an empirical mean and its expectation. By the law of large numbers we know asymptotically that the mean converges to the expectation in probability. So with probability 1, with respect to $\mathbf{Z} \sim D^n$,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n g(Z_i) = \mathbb{E}_{\mathbf{Z} \sim D^n}[g(Z)].$$

So with enough data, the empirical risk is a good approximation to its true risk.

We can do better though, we can say how far apart they are without considering the asymptotic regime:

Theorem (Hoeffding's Inequality). *Let $Z_1 \dots Z_n$ be n iid random variables, and h is a bounded function, $h(Z) \in [a, b]$. Then for all $\epsilon > 0$ we have:*

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[\left| \frac{1}{n} \sum_{i=1}^n h(Z_i) - \mathbb{E}_{\mathbf{Z} \sim D^n}[h(Z)] \right| > \epsilon \right] \leq 2 \exp \left(-\frac{2n\epsilon^2}{(b-a)^2} \right).$$

The probability that the empirical average and expectation are far from each other is small. Let us reparameterize the formula to better understand its consequences. Let the right hand side be δ , so

$$\delta = 2 \exp \left(-\frac{2n\epsilon^2}{(b-a)^2} \right).$$

Then if I solve for ϵ , I get:

$$\epsilon = (b-a) \sqrt{\frac{\log \frac{2}{\delta}}{2n}}$$

So Hoeffding's inequality, applied to the function g becomes:

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[|P^{\text{emp}}g - P^{\text{true}}g| > (b-a) \sqrt{\frac{\log \frac{2}{\delta}}{2n}} \right] \leq \delta.$$

So by “inversion” we get that with probability at least $1 - \delta$:

$$|P^{\text{emp}}g - P^{\text{true}}g| \leq (b - a)\sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$

The expression above is “2-sided” in that there’s an absolute value on the left. This considers whether $P^{\text{emp}}g$ is larger than $P^{\text{true}}g$ or smaller than it. There’s also 1-sided versions of Hoeffding’s inequality where we look at deviations in one direction or the other, for instance here is a 1-sided version of Hoeffding’s:

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[\mathbb{E}_{\mathbf{Z} \sim D^n} [h(Z)] - \frac{1}{n} \sum_{i=1}^n h(Z_i) > \epsilon \right] \leq \exp \left(-\frac{2n\epsilon^2}{(b-a)^2} \right).$$

If set the right side to δ and solve for ϵ , and do “inversion”, we get that with probability at least $1 - \delta$,

$$\mathbb{E}_{\mathbf{Z} \sim D^n} [h(Z)] - \frac{1}{n} \sum_{i=1}^n h(Z_i) \leq (b - a)\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

Here is the inverted one applied to g : with probability at least $1 - \delta$,

$$P^{\text{true}}g - P^{\text{emp}}g \leq (b - a)\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

Moving the empirical term to the right, we have that with probability at least $1 - \delta$,

$$P^{\text{true}}g \leq P^{\text{emp}}g + (b - a)\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

Remember that g is the loss, so $g(Z) = \mathbf{1}_{f(X) \neq Y}$, and so $b - a = 1 - 0 = 1$, and that way we have an upper bound for the true risk, which we want to be small. Namely, with probability at least $1 - \delta$ with respect to the random draw of data,

$$R^{\text{true}}(f) \leq R^{\text{emp}}(f) + \sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

This expression seems very nice, but guess what? It doesn’t apply when f (i.e., g) comes from any reasonable learning algorithm!

Why not?

The answer is in very small font. Try to figure it out before looking at the answer.

It’s for a fixed f . It doesn’t apply when f is chosen based on the data, like in our learning algorithms.

Limitations

In a nutshell, the reason is because we did not pick f_n before seeing the data. f_n actually depends on the data since it minimizes some type of empirical risk. So f_n is a random variable. This means we didn't account for that randomness when we used Hoeffding's inequality.

In other words, what we **showed** was that for f chosen in advance (before seeing the data), with probability at least $1 - \delta$,

$$R^{\text{true}}(f) \leq R^{\text{emp}}(f) + \sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

What we actually **wanted** is that with probability at least $1 - \delta$, for f_n chosen after seeing the data,

$$R^{\text{true}}(f_n) \leq R^{\text{emp}}(f_n) + \sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

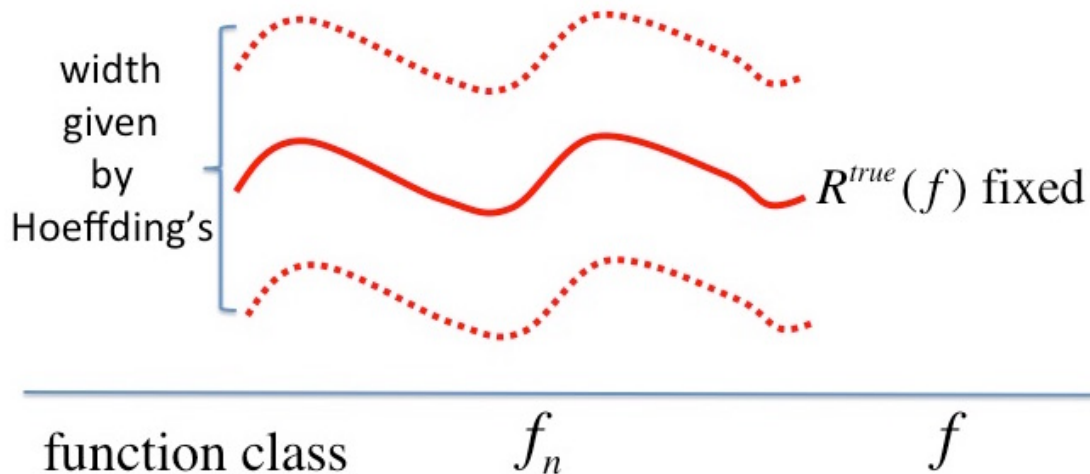
And we do not have this yet.

Let's go into more detail. The result above says that for each fixed function $g \in \mathcal{G}$, there is a large set S of "good" datasets for which

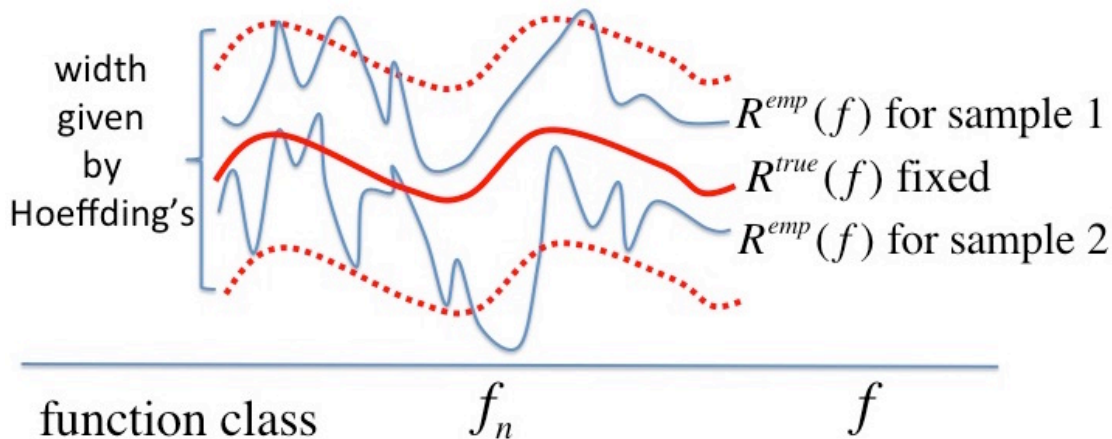
$$P^{\text{true}}g - P^{\text{emp}}g \leq \sqrt{\frac{\log \frac{1}{\delta}}{2n}},$$

and this set of datasets has measure $\mathbb{P}_{\mathbf{Z} \sim D^n}[\mathbf{Z} \in S] \geq 1 - \delta$. However, these sets may be different for different functions g . In other words, for the sample \mathbf{Z} we actually observe, there's no telling how many of the functions in \mathcal{G} will actually satisfy this inequality!

This figure might help you understand. Each point on the x-axis is a different function. The curve marked $R^{\text{true}}(f)$ is the true risk, which is a constant for each f (not a distribution) since it is calculated from the whole distribution (and not a sample).



If you give me a dataset and a function f , I can calculate $R^{\text{emp}}(f)$ for that dataset, which gives me a dot on the plot. So, for each dataset we get a different curve on the figure. For each f , Hoeffding's inequality makes sure that most of the time, the $R^{\text{emp}}(f)$ curves lie within a small distance of $R^{\text{true}}(f)$, though we don't know which ones. In other words, for an observed dataset, only some of the functions in \mathcal{F} will satisfy the inequality, not all of them.



But remember, our algorithms choose f_n *knowing* the data. They generally try to minimize the regularized $R^{\text{emp}}(f)$. Consider drawing a dataset S , which corresponds to a curve on the figure. Our algorithm could (on purpose) meander along that curve until it chooses a f_n that gives a small value of R^{emp} . This value could be very far from $R^{\text{true}}(f_n)$. This could definitely happen, and if there are more f 's to choose from (if the function class is larger), then this happens more easily – uh oh! In other words, if \mathcal{F} is large enough, one can find, somewhere along the axis, a function f for which the difference between the two curves $R^{\text{emp}}(f)$ and $R^{\text{true}}(f)$ will be very large.

We don't want this to happen!

Uniform Bounds

We really need to make sure our algorithm doesn't do this – otherwise it will never generalize. That's why we're going to look at *uniform deviations* in order to upper bound (1) or (2):

$$R^{\text{true}}(f_n) - R^{\text{emp}}(f_n) \leq \sup_{f \in \mathcal{F}} (R^{\text{true}}(f) - R^{\text{emp}}(f))$$

where we look at the worst deviation over all functions in the class.

Let us construct a first uniform bound, using Hoeffding's inequality and the union bound. [This bound applies only when the number of functions in \$\mathcal{F}\$ \(or equivalently \$\mathcal{G}\$ \) is finite.](#) Let's assume that the total number of classifiers in the set is M . You might think that it is unrealistic to have a finite set of classifiers, but it is actually not unrealistic at all. The set of decision trees over categorical/binary data is finite. The set of linear models with integer coefficients is finite. The

set of neural networks with integer coefficients (which people purposely develop because they can program hardware more easily with binary/integer-valued weights) is finite. The bound we will develop next (the Ockham's Razor bound) are nice and tight for such function classes.

We'll index the functions by j , so we write that g_j is the j th function in \mathcal{G} .

Define the following for loss function g_j :

$$C_j^{(\text{bad})} = \{\text{"bad" datasets } \mathbf{Z} \text{ for } g_j \text{ where } P^{\text{true}}g_j - P^{\text{emp}}g_j \geq \epsilon\}.$$

This set contains all the "bad" datasets, those for which the bound fails for function g_j . From Hoeffding's Inequality, for each j ,

$$\mathbb{P}_{\mathbf{Z} \sim D^n}[\mathbf{Z} \in C_j^{(\text{bad})}] \leq \delta.$$

Consider two functions g_1 and g_2 . Say we want to measure how many datasets are "bad" for either one of these functions or the other. We're going to use the *union bound* to do this. Shortening notation a little bit, the union bound says:

$$\mathbb{P}_{\mathbf{Z} \sim D^n}[\mathbf{Z} \in C_1^{(\text{bad})} \cup C_2^{(\text{bad})}] \leq \mathbb{P}_{\mathbf{Z} \sim D^n}[\mathbf{Z} \in C_1^{(\text{bad})}] + \mathbb{P}_{\mathbf{Z} \sim D^n}[\mathbf{Z} \in C_2^{(\text{bad})}] \leq 2\delta,$$

the probability that we hit a bad dataset for either g_1 or g_2 is

$$\leq \text{prob to hit a bad dataset in } C_1^{(\text{bad})} + \text{prob to hit a bad dataset in } C_2^{(\text{bad})}.$$

More generally, the union bound says (where here I've shortened the notation to make $C_j^{(\text{bad})}$ look like an event rather than a set):

$$\mathbb{P}[C_1^{(\text{bad})} \cup \dots \cup C_M^{(\text{bad})}] \leq \sum_{j=1}^M \mathbb{P}[C_j^{(\text{bad})}] \leq M\delta.$$

So this is a bound on the probability that our chosen dataset will be bad for any of the functions g_1, \dots, g_M . So we get:

$$\begin{aligned} & \mathbb{P}_{\mathbf{Z} \sim D^n}[\exists g \in \{g_1, \dots, g_M\} : P^{\text{true}}g - P^{\text{emp}}g \geq \epsilon] \\ & \leq \sum_{j=1}^M \mathbb{P}_{\mathbf{Z} \sim D^n}[P^{\text{true}}g_j - P^{\text{emp}}g_j \geq \epsilon] \\ & \leq \sum_{j=1}^M \exp(-2n\epsilon^2) \\ & = M \exp(-2n\epsilon^2). \end{aligned}$$

If we define a new δ so we can invert:

$$\delta := M \exp(-2n\epsilon^2)$$

and solve for ϵ , we get:

$$\epsilon = \sqrt{\frac{\log M + \log \frac{1}{\delta}}{2n}}.$$

Plugging that in and inverting, we find that with probability at least $1 - \delta$,

$$\forall g \in \{g_1, \dots, g_M\} : P^{\text{true}}g - P^{\text{emp}}g \leq \sqrt{\frac{\log M + \log \frac{1}{\delta}}{2n}}.$$

Changing g 's back to f 's, we've proved the following:

Theorem. (Ockham's Razor = Hoeffding + Union Bound)

This bound applies for finite \mathcal{F} , so $\mathcal{F} = \{f_1 \dots f_M\}$. For all $\delta > 0$ with probability at least $1 - \delta$,

$$\forall f \in \mathcal{F}, \quad R^{\text{true}}(f) \leq R^{\text{emp}}(f) + \sqrt{\frac{\log M + \log \frac{1}{\delta}}{2n}}.$$

Just to recap the reason why this bound is better than the last one, if we know our algorithm only picks functions from a finite function class \mathcal{F} , we now have a bound that can be applied to f_n , even though it depends on the data.

Note the main difference with plain Hoeffding's inequality is the extra $\log M$ term on the right hand side. This term is the one saying we want M bounds to hold simultaneously.

Estimation Error

Let's say we're doing empirical risk minimization, that is, f_n is the minimizer of the empirical risk R^{emp} . In other words, $f_n \in \arg \min_{f \in \mathcal{F}} R^{\text{emp}}(f)$.

We can use the theorem above (combined with the general idea we used near Equation (1)) to get an upper bound on the Estimation Error. Start with this:

$$R^{\text{true}}(f_n) = R^{\text{true}}(f_n) - R^{\text{true}}(f^*) + R^{\text{true}}(f^*)$$

Then we'll use the fact that $R^{\text{emp}}(f^*) - R^{\text{emp}}(f_n) \geq 0$. (From the definition of f_n being the minimizer of R^{emp} .)

We'll add that to the expression above:

$$\begin{aligned} R^{\text{true}}(f_n) &\leq [R^{\text{emp}}(f^*) - R^{\text{emp}}(f_n)] + R^{\text{true}}(f_n) - R^{\text{true}}(f^*) + R^{\text{true}}(f^*) \\ &= R^{\text{emp}}(f^*) - R^{\text{true}}(f^*) - R^{\text{emp}}(f_n) + R^{\text{true}}(f_n) + R^{\text{true}}(f^*) \\ &\leq |R^{\text{true}}(f^*) - R^{\text{emp}}(f^*)| + |R^{\text{true}}(f_n) - R^{\text{emp}}(f_n)| + R^{\text{true}}(f^*) \\ &\leq 2 \sup_{f \in \mathcal{F}} |R^{\text{true}}(f) - R^{\text{emp}}(f)| + R^{\text{true}}(f^*). \end{aligned}$$

We could use a 2-sided version of the theorem (leading to an extra factor of 2 somewhere) that with probability $1 - \delta$, that first term is bounded by the square root term in the theorem. Specifically, we know that with probability at least $1 - \delta$:

$$R^{\text{true}}(f_n) \leq 2 \sqrt{\frac{\log M + \log \frac{2}{\delta}}{2n}} + R^{\text{true}}(f^*).$$

Actually, if you think about it, both terms in the right hand side depend on the size of the class \mathcal{F} . If this size increases, the first term will increase, and the second term will decrease. (The first term will increase because the number of functions in the class, M , increases, while the second term becomes smaller because $R^{\text{true}}(f^*)$ has the possibility to become smaller as the number of functions in \mathcal{F} increases.)

Summary and Perspective

- Generalization requires knowledge (like restricting f to lie in a restricted class \mathcal{F}).
- The error bounds are valid with respect to the repeated sampling of training sets.
- For a fixed function f , for most datasets,

$$R^{\text{true}}(f) - R^{\text{emp}}(f) \approx 1/\sqrt{n}.$$

- For most datasets, if the function class is finite, $|\mathcal{F}| = M$,

$$\sup_{f \in \mathcal{F}} [R^{\text{true}}(f) - R^{\text{emp}}(f)] \approx \sqrt{\log M/n}.$$

The extra term is because we choose f_n in a way that changes with the data.

There are several things that could be improved. For instance Hoeffding's inequality only uses the boundedness of the functions, not their variance, which is something we won't deal with here. The supremum over \mathcal{F} of $R^{\text{true}}(f) - R^{\text{emp}}(f)$ is not necessarily what the algorithm would choose, so the upper bound could be loose. The union bound is fairly loose, because it still holds even if all the f_j 's had "bad" datasets that do not overlap, when in fact a bad dataset for one function could easily be a bad dataset for another function.

The bound is vacuous when \mathcal{F} is infinite, even if \mathcal{F} is very boring, like all lines that are very close to each other. Those lines don't have much expressive power, but the bound is so loose that it can't tell us anything important. We need another way to say that \mathcal{F} is limited in its expressiveness.

Infinite Case: VC Dimension

Here we'll show how to extend the previous results to the case where the class \mathcal{F} is infinite.

We'll start with a simple refinement of the union bound that allows to extend the previous results to the (countably) infinite case.

Recall that by Hoeffding's inequality for a single function g , for each $\delta > 0$, where possibly we could choose δ depending on g , which we write $\delta(g)$, we have:

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[P^{\text{true}}g - P^{\text{emp}}g \geq \sqrt{\frac{\log \frac{1}{\delta(g)}}{2n}} \right] \leq \delta(g).$$

Hence if we have a countable set \mathcal{G} , the union bound gives:

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[\exists g \in \mathcal{G} : P^{\text{true}} g - P^{\text{emp}} g \geq \sqrt{\frac{\log \frac{1}{\delta(g)}}{2n}} \right] \leq \sum_{g \in \mathcal{G}} \delta(g).$$

If we choose the $\delta(g)$'s so that they add up to a constant total value δ , that is, $\delta(g) = \delta p(g)$ where $\sum_{g \in \mathcal{G}} p(g) = 1$, then the right hand side is just δ and we get the following with inversion: with probability at least $1 - \delta$,

$$\forall g \in \mathcal{G}, P^{\text{true}} g \leq P^{\text{emp}} g + \sqrt{\frac{\log \frac{1}{p(g)} + \log \frac{1}{\delta}}{2n}}.$$

This bound is nice - it handles a countably infinite hypothesis space.

Note that if \mathcal{G} is finite with size M , and we take a uniform $p(g) = \frac{1}{M}$, we get the $\log M$ term as before.

General Case

When the set \mathcal{G} is uncountable, the previous approach doesn't work because $p(g)$ is a density, so it's 0 for a given g and the bound will be vacuous. We'll switch back to the original class \mathcal{F} rather than the loss class for now, and we'll have only binary f . (We can easily reduce real-valued f to binary by taking $\text{sign}(f)$). The general idea is to look at the function class's behavior on the dataset. Given z_1, \dots, z_n , we consider

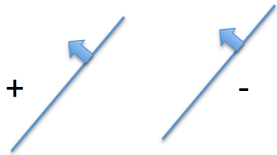
$$\mathcal{F}_{z_1, \dots, z_n} = \{f(z_1), \dots, f(z_n) : f \in \mathcal{F}\}.$$

$\mathcal{F}_{z_1, \dots, z_n}$ is the set of ways the data z_1, \dots, z_n are classified by functions from \mathcal{F} . Since the functions f can only take two values, this set will always be finite, no matter how big \mathcal{F} is.

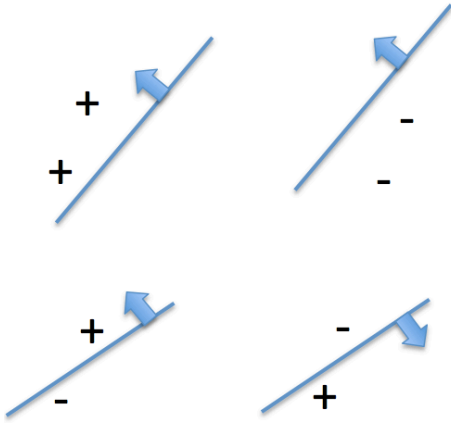
Definition (Growth Function) The growth function of function class \mathcal{F} is the maximum number of ways into which n points can be classified by the function class. The notation is:

$$S_{\mathcal{F}}(n) = \sup_{(z_1, \dots, z_n)} |\mathcal{F}_{z_1, \dots, z_n}|.$$

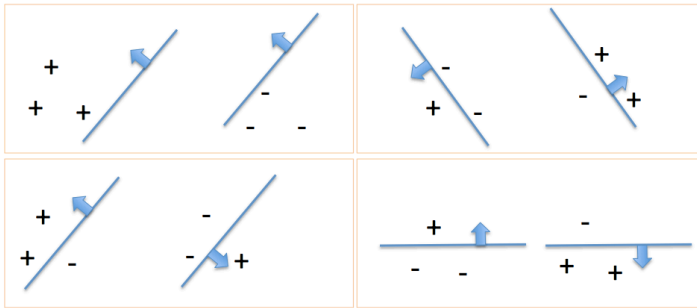
Consider \mathcal{F} containing halfspaces – binary functions whose decision boundary is a line. We'll work in the plane. The growth function for one point is $S_{\mathcal{F}}(1) = 2 = 2^1$.



The growth function for two points is $S_{\mathcal{F}}(2) = 4 = 2^2$.



The growth function for three points is $S_{\mathcal{F}}(3) = 8 = 2^3$.



What is the growth function for 4 points? I'll let you work it out but whatever it is, it will be less than 2^4 . This is because of situations like this:

- +
+ -

where there is no way to separate the points using some combination of labels. No matter how I move the points around, I can't get rid of this situation.

We defined the growth function in terms of the initial class \mathcal{F} but we can do the same with the loss class \mathcal{G} since there's a 1-1 mapping, so we'll get $S_{\mathcal{G}}(n) = S_{\mathcal{F}}(n)$.

This growth function can be used as a measure of the 'size' of a class of functions as demonstrated by the following result:

Theorem-GrowthFunction (Vapnik-Chervonenkis) For any $\delta > 0$, with probability at least $1 - \delta$ with respect to a random draw of the data,

$$\forall f \in \mathcal{F} \quad R^{\text{true}}(f) \leq R^{\text{emp}}(f) + 2\sqrt{2 \frac{\log S_{\mathcal{F}}(2n) + \log \frac{4}{\delta}}{n}}.$$

(proof soon).

This bound shows nicely that simplicity implies generalization. The simpler the function class, the better the guarantee that R^{true} will be small. In the finite case where $|\mathcal{F}| = M$ (we have M possible classifiers), we have $S_{\mathcal{F}}(n) \leq M$ (at worst we use up all the classifiers when we're computing the growth function). So this bound is always better than the one we had before (except for the constants).

But we need to figure out how to compute $S_{\mathcal{F}}(n)$. We'll do that using VC dimension.

VC dimension

Since $f \in \{-1, 1\}$, it is clear that $S_{\mathcal{F}}(n) \leq 2^n$.

If $S_{\mathcal{F}}(n) = 2^n$ there is a data set of size n points such that \mathcal{F} can generate any classification on these points (we say \mathcal{F} *shatters* the set).

The VC dimension of a class \mathcal{F} is the size of the largest set that it can shatter.

Definition. (VC dimension) *The VC dimension of a class \mathcal{F} is the largest n such that*

$$S_{\mathcal{F}}(n) = 2^n.$$

What is the VC dimension of halfspaces in 2 dimensions?

It's three.

Can you guess the VC dimension of halfspaces in p dimensions?

It's $p + 1$.

In the example, the number of parameters needed to define the half space in \mathbf{R}^p is the number of dimensions, p . So a natural question to ask is whether the VC dimension is related to the number of parameters of the function class. In other words, VC dimension is supposed to measure complexity of a function class - does it just basically measure the number of parameters?

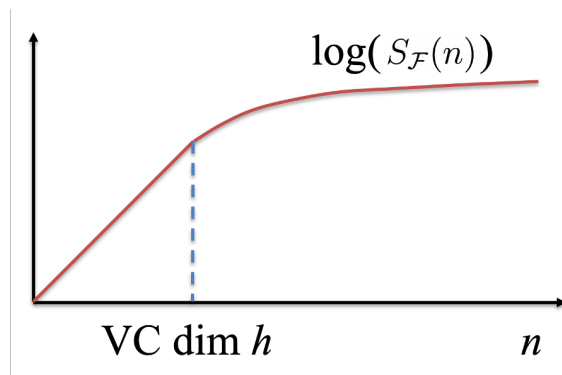
No, there are examples in 1 dimension where the VC dimension is infinite. Think of the set of sin functions with arbitrarily large or small frequency. Then you can distribute the points along the x-axis so that they can be separated with various sin functions. To construct this special class of functions, start with y as some vector of -1's and 1's that you choose yourself. Then choose $x = 2\pi 10^{-i}$ for each point i . Then calculate $t = 1/4 \sum_i [(1 - y_i)10^i + 1]$, which is an enormous number. Then compute $\text{sign}(\sin(tx_i))$ for each i and you will be surprised to find that it equals y_i for each i !

So how can VC dimension help us compute the growth function? Well, if a class of functions has VC dim h , then we know that we can shatter n observations when $n \leq h$, and in that case, $S_{\mathcal{F}}(n) = 2^n$. If $n > h$, then we know we can't shatter the points, so $S_{\mathcal{F}}(n) < 2^n$.

This doesn't seem very helpful perhaps, but actually an intriguing phenomenon occurs for $n \geq h$, shown below.

The plot below shows for $n \geq h$ (where we can't shatter) the number of ways we can classify - that's the growth function. The growth function which is exponential up until the VC dimension,

becomes polynomial afterwards!



Typical behavior of the log growth function.

This is captured by the following lemma.

Lemma. (Vapnik and Chervonenkis, Sauer, Shelah) *Let \mathcal{F} be a class of functions with finite VC dimension h . Then for all $n \in \mathbb{N}$,*

$$S_{\mathcal{F}}(n) \leq \sum_{i=0}^h \binom{n}{i}$$

and for all $n \geq h$,

$$S_{\mathcal{F}}(n) \leq \left(\frac{en}{h}\right)^h.$$

Using this lemma for $n \geq h$ along with Theorem-GrowthFunction, (and removing the h term from that denominator to create an upper bound) we get:

Theorem VC-Bound. If \mathcal{F} has VC dim h , and for $n \geq h$, with probability at least $1 - \delta$,

$$\forall f \in \mathcal{F} \quad R^{\text{true}}(f) \leq R^{\text{emp}}(f) + 2\sqrt{2 \frac{h(\ln(2n) + 1) + \log \frac{4}{\delta}}{n}}.$$

What is important to remember from this result is that the difference between the true and empirical risk is at most of order

$$\sqrt{\frac{h \log n}{n}}.$$

Remember, at the beginning of the lecture, the bound was infinite for infinite function classes, i.e., vacuous!

Recap

Why is Theorem VC-Bound important? It shows that limiting the complexity of the class of functions leads to better generalization. An interpretation of VC dim and growth functions is

that they measure the “effective” size of the class, that is, the size of the projection of the class onto finite observations. This measure doesn’t just count the number of functions in the class, but depends on the geometry of the class, that is, the projections onto the possible observations. Also since the VC dimension is finite, our bound shows that the empirical risk will converge uniformly over the class \mathcal{F} to the true risk.

Back to Margins

How is it that SVM’s limit the complexity? Well, the choice of kernel controls the complexity. But also the margin itself controls complexity. There is a set of linear classifiers called “gap-tolerant classifiers” that I won’t define precisely (it gets complicated) that require a margin of at least Δ between points of the two different classes. The points are also forced to live inside a sphere of diameter D . So the class of functions is fairly limited, since they not only need to separate the points with a margin of Δ , but also we aren’t allowed to move the points outside of the sphere.

“Theorem” VC-Margin. (Vapnik) *For data in \mathbf{R}^p , the VC dimension h of (linear) gap-tolerant classifiers with gap Δ belong to a sphere of diameter D is bounded by the inequality:*

$$h \leq \min \left(\left\lceil \frac{D^2}{\Delta^2} \right\rceil, p \right) + 1.$$

So the VC dimension (of the set of functions that separate points with some margin) is less than $1/\text{margin}$. If we have a large margin, we necessarily have a small VC-dimension.

This says something interesting about halfspaces in \mathbf{R}^p . We know that if we are allowed to choose any halfspace, the VC dim is $p + 1$. But what about if we have a large margin?

It shows that the VC dim can be less than $p+1$. It is strictly better if we include the margin than if we don’t!

Symmetrization

We’ll do the proof of Theorem-GrowthFunction. The key ingredient is the *symmetrization lemma*. We’ll use what’s called a “ghost sample” which is an extra (virtual) data set Z'_1, \dots, Z'_n . Denote P'^{emp} the corresponding empirical measure.

(Lemma-Symmetrization) *For any $t > 0$, such that $nt^2 \geq 2$,*

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[\sup_{g \in \mathcal{G}} (P^{\text{true}} - P^{\text{emp}})g > t \right] \leq 2 \mathbb{P}_{\mathbf{Z} \sim D^n, \mathbf{Z}' \sim D^n} \left[\sup_{g \in \mathcal{G}} (P'^{\text{emp}} - P^{\text{emp}})g > t/2 \right].$$

That is, if we can bound the difference between the behavior on one dataset versus another, it gives us a bound on the behavior of a dataset with respect to the true risk.

Proof. Let g_n be the function achieving the supremum in the lhs term, which depends on Z_1, \dots, Z_n . Think about the event that: $(P^{\text{true}} - P^{\text{emp}})g_n > t$ (the dataset’s loss is far from the true loss) and $(P^{\text{true}} - P'^{\text{emp}})g_n < t/2$ (the ghost dataset’s loss is close to the true loss). If

this event were true, it sort of means that things didn't generalize well for Z_1, \dots, Z_n but that they did generalize well for Z'_1, \dots, Z'_n . If we can show that this event happens rarely, then the ghost sample can help us. Again, the event that we want to happen rarely is $(P^{\text{true}} - P^{\text{emp}})g_n > t$ and $(P^{\text{true}} - P^{\text{emp}})g_n < t/2$.

$$\begin{aligned}
& \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t} \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n < t/2} \\
&= \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t \text{ and } (P^{\text{true}} - P^{\text{emp}})g_n < t/2} \\
&= \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t \text{ and } (P^{\text{emp}} - P^{\text{true}})g_n \geq -t/2} \\
&\leq \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}} + P^{\text{emp}} - P^{\text{true}})g_n > t - t/2 = t/2} = \mathbf{1}_{(-P^{\text{emp}} + P^{\text{emp}})g_n > t/2}.
\end{aligned}$$

Taking expectations with respect to the second dataset, and using the trick to change expectation into probability,

$$\begin{aligned}
& \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t} \mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n < t/2] \\
&\leq \mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{emp}} - P^{\text{emp}})g_n > t/2].
\end{aligned} \tag{3}$$

Do you remember Chebyshev's Inequality? It says $\mathbb{P}[|X - \mathbb{E}X| \geq t] \leq \text{Var}X/t^2$. We'll apply it now, to that second term on the left, inverted:

$$\mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n \geq t/2] \leq \frac{4\text{Var}g_n}{nt^2}.$$

I hope you'll believe me when I say that any random variable that has range $[0, 1]$ has variance less than or equal to $1/4$. Hence,

$$\mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n \geq t/2] \leq \frac{1}{nt^2}.$$

Inverting back, so that it looks like the second term on the left of (3) again:

$$\mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n < t/2] \geq 1 - \frac{1}{nt^2}.$$

Multiplying both sides by $\mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t}$ I get back to the left of (3):

$$\begin{aligned}
\mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t} \left(1 - \frac{1}{nt^2}\right) &\leq \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t} \mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n < t/2] \\
&\leq \mathbb{P}_{\mathbf{Z}' \sim D^n}[(P^{\text{emp}} - P^{\text{emp}})g_n > t/2] \quad \text{from (3)}.
\end{aligned}$$

Taking the expectation with respect to the first dataset, the term

$$\mathbb{E}_{\mathbf{Z} \sim D^n} \mathbf{1}_{(P^{\text{true}} - P^{\text{emp}})g_n > t} \quad \text{becomes} \quad \mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n > t].$$

And now we get:

$$\begin{aligned}
\mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n > t] \left(1 - \frac{1}{nt^2}\right) &\leq \mathbb{P}_{\mathbf{Z}' \sim D^n} \mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{emp}} - P^{\text{emp}})g_n > t/2] \\
\mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n > t] &\leq \left(\frac{1}{1 - \frac{1}{nt^2}}\right) \mathbb{P}_{\mathbf{Z}' \sim D^n} \mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{emp}} - P^{\text{emp}})g_n > t/2].
\end{aligned}$$

Only one more step, which uses our assumption $nt^2 \geq 2$.

$$\begin{aligned} nt^2 &\geq 2 \\ \frac{1}{nt^2} &\leq \frac{1}{2} \\ 1 - \frac{1}{nt^2} &\geq 1 - \frac{1}{2} = \frac{1}{2} \\ \left(\frac{1}{1 - \frac{1}{nt^2}} \right) &\leq 2 \end{aligned}$$

Plug:

$$\begin{aligned} \mathbb{P}_{\mathbf{Z} \sim D^n}[(P^{\text{true}} - P^{\text{emp}})g_n > t] &\leq 2\mathbb{P}_{\mathbf{Z}' \sim D^n, \mathbf{Z} \sim D^n}[(P'^{\text{emp}} - P^{\text{emp}})g_n > t/2] \\ &\leq 2\mathbb{P}_{\mathbf{Z} \sim D^n, \mathbf{Z}' \sim D^n} \left[\sup_{g \in \mathcal{G}} (P'^{\text{emp}} - P^{\text{emp}})g > t/2 \right]. \end{aligned}$$

■

Remember, we're still in the middle of proving Theorem-GrowthFunction. The symmetrization is just a step in that proof. This symmetrization lemma allows us to replace the expectation $P^{\text{true}}g$ by an empirical average over the ghost sample. As a result, the proof will only depend on the *projection* of the class \mathcal{G} on the double dataset

$$\mathcal{G}_{Z_1 \dots Z_n, Z'_1 \dots Z'_n},$$

which contains finitely many different vectors. In other words, an element of this set is just the vector $[g(x_1), \dots, g(x_n), g(x'_1), \dots, g(x'_n)]$, and there are finitely many possibilities for vectors like this. So we can use the union bound that we used for the finite case. The other ingredient that we need to prove Theorem-GrowthFunction is this one:

$$\mathbb{P}_{\mathbf{Z} \sim D^n, \mathbf{Z}' \sim D^n} [P^{\text{emp}}g - P'^{\text{emp}}g > t] \leq 2e^{-nt^2/2}. \quad (4)$$

This one comes itself from a mix of Hoeffding's with the union bound:

$$\begin{aligned} &\mathbb{P}_{\mathbf{Z} \sim D^n, \mathbf{Z}' \sim D^n} [P^{\text{emp}}g - P'^{\text{emp}}g > t] \\ &= \mathbb{P}_{\mathbf{Z} \sim D^n, \mathbf{Z}' \sim D^n} [P^{\text{emp}}g - P^{\text{true}}g + P^{\text{true}}g - P'^{\text{emp}}g > t] \\ &\leq \mathbb{P}_{\mathbf{Z} \sim D^n} [P^{\text{emp}}g - P^{\text{true}}g > t/2] + \mathbb{P}_{\mathbf{Z}' \sim D^n} [P^{\text{true}}g - P'^{\text{emp}}g > t/2] \\ &\leq e^{-2n(t/2)^2} + e^{-2n(t/2)^2} \\ &= 2e^{-nt^2/2}. \end{aligned}$$

We just have to put the pieces together now:

$$\begin{aligned}
& \mathbb{P}_{\mathbf{Z} \sim D^n} [\sup_{g \in \mathcal{G}} (P^{\text{true}} - P^{\text{emp}})g > t] \\
& \leq 2 \mathbb{P}_{\mathbf{Z} \sim D^n \mathbf{Z}' \sim D^n} [\sup_{g \in \mathcal{G}} (P'^{\text{emp}} - P^{\text{emp}})g > t/2] \quad \text{Lemma-Symmetrization} \\
& = 2 \mathbb{P}_{\mathbf{Z} \sim D^n \mathbf{Z}' \sim D^n} [\sup_{g \in \mathcal{G}_{Z_1, \dots, Z_n, Z'_1, \dots, Z'_n}} (P'^{\text{emp}} - P^{\text{emp}})g > t/2] \quad (\text{restrict to data}) \\
& \leq 2 \sum_{g \in \mathcal{G}_{Z_1, \dots, Z_n, Z'_1, \dots, Z'_n}} \mathbb{P}_{\mathbf{Z} \sim D^n \mathbf{Z}' \sim D^n} [(P'^{\text{emp}} - P^{\text{emp}})g > t/2] \quad (\text{union bound}) \\
& \leq 2 \sum_{g \in \mathcal{G}_{Z_1, \dots, Z_n, Z'_1, \dots, Z'_n}} 2e^{-n(t/2)^2/2} \quad (\text{Hoeffding}) \\
& = 4e^{-nt^2/8} \sum_{g \in \mathcal{G}_{Z_1, \dots, Z_n, Z'_1, \dots, Z'_n}} 1 \\
& = 4S_{\mathcal{G}}(2n) e^{-nt^2/8}.
\end{aligned}$$

And using inversion,

$$\mathbb{P}_{\mathbf{Z} \sim D^n} [\sup_{g \in \mathcal{G}} (P^{\text{true}} - P^{\text{emp}})g \leq t] \geq 1 - 4S_{\mathcal{G}}(2n) e^{-nt^2/8}.$$

Letting $\delta = 4S_{\mathcal{G}}(2n) e^{-nt^2/8}$, solving for t yields:

$$t = \sqrt{\frac{8}{n} \log \frac{4S_{\mathcal{G}}(2n)}{\delta}}.$$

Plug:

$$\mathbb{P}_{\mathbf{Z} \sim D^n} \left[\sup_{g \in \mathcal{G}} (P^{\text{true}} - P^{\text{emp}})g \leq 2 \sqrt{2 \frac{\log S_{\mathcal{G}}(2n) + \log \frac{4}{\delta}}{n}} \right] \geq 1 - \delta.$$

So, with probability at least $1 - \delta$,

$$\forall g \in \mathcal{G} \quad (P^{\text{true}} - P^{\text{emp}})g \leq 2 \sqrt{2 \frac{\log S_{\mathcal{G}}(2n) + \log \frac{4}{\delta}}{n}}.$$

That's the result of Theorem-GrowthFunction. ■

We have done quite a lot in this lecture. We have shown that in order to generalize, we should have a low empirical error and a low capacity function class.

I will end the lecture by saying that VC dimension is not the only way to measure complexity of a class of functions. There are many other ways, including *covering numbers* and *Rademacher complexity*. As long as you can find some way in which the model class is limited, you should be able to create a generalization bound from it.