COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Suqi Liu

Lecture #10
March 07, 2018

Last time we started proving this very general result about how quickly the empirical average converges to its expected value, not just for a single function or single hypothesis but for an entire family of functions or hypotheses. Our goal is to bound the generalization error in such a way that the training error converges to the generalization error. But we are proving it in a more general form. And we are deriving bounds in terms of the Rademacher complexity that we talked about last time. Today, we are going to finish proving the general theorem and then go back to look at how to actually bound the Rademacher complexity and apply it to the setting which we care about. We will look at various techniques to do that and relate it to all other complexity measures we talked about in the class. And then we will move on to something completely different. That will be the end of the first part of this course.

1 Generalization bounds based on Rademacher complexity

Theorem 1. *Let $\mathcal{F}$ be family of functions $f : \mathbb{Z} \to [0,1]$. $S = \langle z_1, \dots, z_n \rangle$ is a random sample with i.i.d. $z_i \sim \mathcal{D}$. With probability at least $1-\delta$, for all $f \in \mathcal{F}$,*

$$\mathbb{E}[f] \le \hat{\mathbb{E}}_S[f] + \begin{Bmatrix} 2\hat{\mathcal{R}}_S(\mathcal{F}) \\ 2\mathcal{R}_m(\mathcal{F}) \end{Bmatrix} + O\left(\sqrt{\frac{\ln 1/\delta}{m}}\right),$$

where $\hat{\mathcal{R}}_S(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i f(z_i)\right]$ is the empirical Rademacher complexity on a sample S for a function class f and $\mathcal{R}_m(\mathcal{F}) = \mathbb{E}_S\left[\hat{\mathcal{R}}_S(\mathcal{F})\right]$ is its expected value.

Proof. We first look at the function

$$\Phi(S) = \sup_{f\in\mathcal{F}} \left(\mathbb{E}[f] - \widehat{\mathbb{E}}_S[f]\right).$$

Step 1: $\Phi(S) \le \mathbb{E}_S[\Phi(S)] + O\left(\sqrt{\frac{\ln 1/\delta}{m}}\right)$ with probability at least $1-\delta$.

Then, we introduce the double sample trick. Let $S = \langle z_1, \dots, z_m \rangle$ be the "real" sample and $S' = \langle z'_1, \dots, z'_m \rangle$ be the "ghost" sample which has the same distribution with the original sample.

Step 2: $\mathbb{E}_S[\Phi(S)] \le \mathbb{E}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left(\widehat{\mathbb{E}}_{S'}[f] - \widehat{\mathbb{E}}_S[f]\right)\right]$.

We replace the expected value with the empirical average on another new imaginary sample and end up with this nice symmetric expression on the difference between empirical averages.

The other trick that we used before is this idea of taking the data and randomly permuting it. We go through with each pair, flip a coin. If the coin is head, we swap them, otherwise we leave them alone. Writing it in an algorithmic way,

```
for $i = 1 \dots, m$,
    with probability 1/2, swap $z_i \leftrightarrow z'_i$,
    else do nothing.
```

Denote T, T' the resulting samples after this random permuting, where T is the new real sample and T' is the new ghost sample. As before, we can argue that T and T' have exactly the same distributions as the original samples S and S'. They just have the additional randomness that comes from swapping. But the additional randomness does not change the distributions at all. We can take the expression

$$\widehat{\mathbb{E}}_{S'}[f] - \widehat{\mathbb{E}}_S[f] = \frac{1}{m}\sum_{i=1}^m f(z_i') - \frac{1}{m}\sum_{i=1}^m f(z_i) = \frac{1}{m}\sum_{i=1}^m \left(f(z_i') - f(z_i)\right)$$

and replace S and S' with T and T', and get exactly the same expression. Writing it out explicitly, we have

$$\begin{aligned}\widehat{\mathbb{E}}_{T'}[f] - \widehat{\mathbb{E}}_T[f] &= \frac{1}{m}\sum_{i=1}^m \begin{cases} f(z_i) - f(z_i') & \text{if } z_i \leftrightarrow z_i' \text{ swapped} \\ f(z_i') - f(z_i) & \text{if no swap}\end{cases} \\ &= \frac{1}{m}\sum_{i=1}^m \sigma_i \left(f(z_i') - f(z_i)\right),\end{aligned}$$

where $\sigma_i = \begin{cases} -1 & \text{if swap} \\ +1 & \text{if no swap}\end{cases}$ indicates whether they are swapped or not. We wrap this up to the next step.

Step 3: $\mathbb{E}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left(\widehat{\mathbb{E}}_{S'}[f] - \widehat{\mathbb{E}}_S[f]\right)\right] = \mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(z_i') - f(z_i)\right)\right]$.

The expected value is taken with respect to T and T', which is generated by the original samples S, S', and then the sequence of σ_i's indicating swap or no-swap for every pair.

Step 4: $\mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(z_i') - f(z_i)\right)\right] \le 2\mathcal{R}_m(\mathcal{F})$.

We can take the average and break it apart.

$$\begin{aligned}\mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(z_i') - f(z_i)\right)\right] &= \mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^m \sigma_i f(z_i') + \frac{1}{m}\sum_{i=1}^m (-\sigma_i) f(z_i)\right)\right] \\ &\le \mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i f(z_i') + \sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m (-\sigma_i) f(z_i)\right] \\ &= \mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i f(z_i')\right] \\ &\quad + \mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m (-\sigma_i) f(z_i)\right].\end{aligned}$$

The inequality holds because taking the suprema of two expressions separately, we can only get a larger number. The second term in the last line is also the Rademacher complexity since the $(-\sigma_i)$'s have exactly the same distribution as σ_i's. Therefore,

$$\mathbb{E}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i\left(f(z_i') - f(z_i)\right)\right] \le 2\mathcal{R}_m(\mathcal{F}).$$

What we have done is actually a set of straight-line inequalities all the way down. Putting these steps together, we end up with the first bound with respect to the expected Rademacher complexity. To go from expected Rademcher complexity to the empirical

Rademacher complexity which is actually its value on the sample, we can use McDiarmid's inequality, which we already used in the first step to show that the complicated random variable is close to its expected value. We just have to check that the conditions of McDiarmid's are satisfied. □

Here, this is the uniform convergence result. The key part is that with high probability on a single sample, for every function within the family, the empirical average will be close to its true expected value. That is what is so hard to get but it is what we need for learning because we want to be able to show that what happens on a sample is a good proxy for what is happening in general on the entire universe.

We proved this very general result. But we are interested in a more specific case of classification error. We want to be able to show that in the usual learning setting we have been studying, with probability at least $1-\delta$, $\forall h \in \mathcal{H}$,

$$\mathrm{err}(h) \leq \widehat{\mathrm{err}}(h) + \text{something small}.$$

We already talked about how to express the true error and the training error in terms of the indicator variables,

$$\mathrm{err}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}\{h(x) \neq y\}\right],$$
$$\widehat{\mathrm{err}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\{h(x_i) \neq y_i\}.$$

We want to take the theorem and apply it to these indicator variables. Starting with the hypothesis space, we can set up functions which behave like the indicator variables. First, let the space $Z = X \times \{+1, -1\}$ be the Cartesian product between the set of all examples X in our domain and the set of possible labels $\{+1, -1\}$. Then, for each $h \in \mathcal{H}$, define $f_h(x, y) = \mathbb{1}\{h(x) \neq y\}$ which maps the input (x, y) to 0 or 1, depending on whether h incorrectly classifies example (x, y) or not. If we use this notation, the training error can be rewritten simply as the empirical average of f_h on a sample S, $\widehat{\mathrm{err}}(h) = \widehat{\mathbb{E}}_S[f_h]$, and likewise the expected value can be rewritten as the expected value of f_h, $\mathrm{err}(h) = \mathbb{E}[f_h]$. Then, the family of functions we are working with is $\mathcal{F}_{\mathcal{H}} = \{f_h : h \in \mathcal{H}\}$, which is the set of all these functions f_h where h ranges over the hypothesis space $\mathcal{H}$. Basically we are just making some definitions so that what we are calling generation error and training error match what we were calling empirical average and true expectation. In particular, we take the theorem and plug in the definitions. The result we get is that with high probability, $\forall h \in \mathcal{H}$ (equivalently $\forall f_h \in \mathcal{F}_{\mathcal{H}}$),

$$\underset{\substack{\uparrow \\ \mathrm{err}(h)}}{\mathbb{E}[f_h]} \quad \leq \quad \underset{\substack{\uparrow \\ \widehat{\mathrm{err}}(h)}}{\widehat{\mathbb{E}}_S[f_h]} \; + \left\{ \begin{array}{c} 2\hat{\mathcal{R}}_S(\mathcal{F}_{\mathcal{H}}) \\ 2\mathcal{R}_m(\mathcal{F}_{\mathcal{H}}) \end{array} \right\} + O\left(\sqrt{\frac{\ln 1/\delta}{m}}\right).$$

The point is that the expected value is the true error of h and the empirical average is the training error.

We are left with the additional term of Rademacher complexity. We want to be claiming convergence of empirical average to the true expectation. However, it is not even clear how it depends on m. We have to show in fact it goes to zero when m gets large. Right now we would prefer to simplify the Rademacher complexity for this particular function class.

Start by writing out the empirical Rademacher complexity,

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\mathcal{H}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i f_h(z_i)\right] = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i f_h(x_i, y_i)\right].$$

We can take f_h and plug in what it is, which is an indicator variable. But by using the same trick as last time, we can instead use the algebraic form of the indicator function $f_h(x_i, y_i) = \mathbb{1}\{h(x_i) \neq y_i\} = \frac{1-y_i h(x_i)}{2}$, which gives

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\mathcal{H}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}_\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i \frac{1-y_i h(x_i)}{2}\right].$$

By pulling out the first term, which does not depend h, we have

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\mathcal{H}) = \frac{1}{2}\mathbb{E}_\sigma\left[\frac{1}{m}\sum_{i=1}^m \sigma_i + \sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m (-y_i\sigma_i) h(x_i)\right],$$

where the expectation of the first term is just zero. We are left with the expectation over the second term. It is almost like the Rademacher complexity except for $(-y_i\sigma_i)$'s. Since y_i's which are -1's and $+1$'s are fixed and -1 is fixed, we are taking -1's and $+1$'s and multiplying it by a random variable which is $+1$ and -1 with equal probability. Thus, the whole thing $(-y_i\sigma_i)$ is itself a Rademacher variable that has exactly the same distribution as σ_i. Therefore,

$$\widehat{\mathcal{R}}_S(\mathcal{F}_\mathcal{H}) = \frac{1}{2}\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)\right] = \frac{1}{2}\widehat{\mathcal{R}}_S(\mathcal{H}).$$

Plugging this into the theorem, we have

$$\text{err}(h) \leq \widehat{\text{err}}(h) + \left\{\begin{array}{c} \hat{\mathcal{R}}_S(\mathcal{H}) \\ \mathcal{R}_m(\mathcal{H}) \end{array}\right\} + O\left(\sqrt{\frac{\ln 1/\delta}{m}}\right).$$

What we are left with is the Rademacher complexity over the hypothesis space $\mathcal{H}$. That is nice because rather than working with this more complicated set of functions, we can just be working with the hypotheses themselves. Also, notice that the result really only depends on the x_i's since the y_i's have disappeared. The S is only the unlabeled part x_i's, not the labels y_i's. The upshot is, we just have to be able to compute Rademacher complexity of the hypothesis space itself. If we can do that, we get exactly the kind of bounds we want.

2 Rademacher complexity bounds for binary classification

Let's finally prove some bounds on Randemacher complexity. We get our pick either to work with the empirical version or the expected Rademacher complexity. It is usually nicer to work with the empirical version because it is defined on a particular sample, i.e. a particular set of points. We are going to take advantage of that in a minute. It means we only have to be considering what hypothesis in this class is doing on a particular fixed sample. Starting with the easiest case where the hypothesis space is finite, we have the following theorem.

Theorem 2. *If $|\mathcal{H}| < \infty$, then*

$$\widehat{\mathcal{R}}_S(\mathcal{H}) \le \sqrt{\frac{2\ln|\mathcal{H}|}{m}}$$

where $S = \langle x_1, \dots, x_m\rangle$ consists of m unlabeled points.

It is saying that the Rademacher complexity is upper bounded by something which is like log of hypothesis space, our usual complexity measure for finite hypothesis spaces. Finally we have a result that says, at least in this case, the Rademacher complexity is going to zero at the rate $\sqrt{1/m}$. Taking this bound and plugging it back into the inequality, we get a bound on the generalization error in terms of training error plus some additional term, which is a bound of the same kind that we had before which we proved using Hoeffding's inequality and union bound. Here we are getting a little bit different constants, but we have seen that it falls out of this more general result. The Mohri et al. book gives a proof of this, which they call Massart's lemma. We are not going to prove this now, but we are going to reach a point later in the course, where it will follow as a corollary from other stuff.

We already have a way of dealing with finite hypothesis spaces. The whole point of doing all this work was to be able to deal with infinite hypothesis spaces. Suppose we have a hypothesis space $\mathcal{H}$, possibly infinite. Remember that we are working on the fixed sample S consisting of the examples x_i's. All that we care about is how the hypotheses are behaving on that fixed sample. So if we have another set of hypotheses which happen to behave exactly the same way on that sample, then we get exactly the same value. Or maybe said differently, all we care about are the behaviors of the hypotheses in this class on this fixed sample. In particular, we can imagine forming a much smaller hypothesis space, defining $\mathcal{H}' \subset \mathcal{H}$ consisting of one representative hypothesis from $\mathcal{H}$ for each behavior on S. And if we now take this definition and replace $\mathcal{H}$ by $\mathcal{H}'$, we will get exactly the same value, because we haven't changed the hypotheses in terms of how they behave on the sample. If we take the sup over $\mathcal{H}$ and replace it with $\mathcal{H}'$, it will work out the same and we get exactly the same thing. The result is the empirical Rademacher complexity on the sample S of this other class $\mathcal{H}'$,

$$\widehat{\mathcal{R}}_S(\mathcal{H}) = \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)\right] = \mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}'} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)\right] = \widehat{\mathcal{R}}_S(\mathcal{H}').$$

Whereas $\mathcal{H}$ is infinite, $\mathcal{H}'$ is finite. The size of $\mathcal{H}'$ is exactly equal to the number of behaviors of $\mathcal{H}'$ on S. Therefore,

$$\left|\mathcal{H}'\right| = |\Pi_{\mathcal{H}}(S)| \le \Pi_{\mathcal{H}}(m),$$

where $\Pi_{\mathcal{H}}(m)$ is the growth function. Here we have the Rademacher complexity of a finite class. And we can apply the theorem to $\mathcal{H}'$ which is finite and get

$$\widehat{\mathcal{R}}_S(\mathcal{H}) \le \sqrt{\frac{2\ln|\Pi_{\mathcal{H}}(S)|}{m}} \le \sqrt{\frac{2\ln \Pi_{\mathcal{H}}(m)}{m}}.$$

This is basically the same kind of bound we have proved before but with the square root, which is kind of inevitable when we are not getting consistency. That's two of the complexity measures we talked about, the log cardinality of the space and the log growth function. And the last one was good old VC dimension, which we can also get easily. By Sauer's lemma, the growth function is upper bounded by

$$\Pi_{\mathcal{H}}(m) \le \left(\frac{em}{d}\right)^d$$

for $m \geq d \geq 1$, where d is the VC dimension of $\mathcal{H}$. Plugging this in, we have

$$\widehat{\mathcal{R}}_S(\mathcal{H}) \leq \sqrt{\frac{2d \ln \frac{em}{d}}{m}}.$$

We get this bound going to zero in terms of VC dimension. This is in fact the VC dimension on the sample, not VC dimension in general. All the things can be measured on the sample S.

3 Introduction to boosting

We reached the end of the first part of this course. We started out with a lot of idealized and restrictive conditions on learning by assuming the class of functions that are labeling the examples and there is no noise and so on. And bit by bit, we eliminated those assumptions one by one. So at this point, the data can be anything although we are still assuming it is random and i.i.d. We started out by giving this very tedious and specialized ad hoc arguments to show that an algorithm works, meaning outputs a highly accurate hypothesis. Now we can just get rid of all that and have these very general and powerful tools that say if you have an algorithm and want to show it works, we just need to show that it finds a hypothesis that is either consistent with the data or has low training error. Additionally if the hypothesis has low complexity, then we have various ways of saying what that means in precise quantitative terms with enough data. If we have that, then we just plug into these general theorems and have an automatic immediate proof that the hypothesis will be highly accurate. And also these theorems qualitatively tell us something about learning. They can also tell us something about the cases when learning does not work or where learning is impossible. Now that we have developed these tools, we can start looking at more specific algorithms, not just for finding rectangles and intervals, but the kind of algorithms that actually get used in practice.

We are going to talk about algorithms. But the first point we are going to talk about actually begins with a theoretical question, which eventually leads to an algorithm that turns out to be practical. Go back to PAC model. We have a concept class and assume the examples are labeled according to the concept. We require that it be possible for the learning algorithm to come up with a hypothesis with arbitrarily high accuracy, to be able to drive the error down as close to zero as we want, provided we give the algorithm enough data. But what if we have an algorithm that cannot do that? Say a crumby algorithm can only get the error down to 40%, that is, reliably gets accuracy 60%. Can we somehow use that algorithm to come up with a hypothesis that gives 99% accuracy? We got this thing that is doing something really mediocre, not a whole lot better than just guessing randomly. Somehow we want to use that algorithm to come up with a hypothesis that gives almost perfect accuracy. We can kind of take this to an extreme, rather than 60%, the algorithm only gives accuracy 51%. We know we can always trivially get 50% accuracy just by flipping a coin. At an extreme, we can say what happens if we can get accuracy a little bit better than random guessing? Can we somehow boost the accuracy up to 99%?

We can ask the question in the PAC model. A class $\mathcal{C}$ is learnable if there is an algorithm $\mathcal{A}$ so that for every target concept c in that class and for any distribution, for all $\epsilon > 0$, $\delta > 0$, $\mathcal{A}$ is given some polynomial number in $1/\epsilon$, $1/\delta$ of examples labeled by c, and outputs a hypothesis h such that

$$\Pr\left(\operatorname{err}_{\mathcal{D}}(h) \leq \epsilon\right) \geq 1 - \delta.$$

To make the distinction, this is called **strongly learnable**. One thing to notice is that we are allowing the hypothesis h to come from any class.

The other model is almost the same, except that we get rid of the ϵ part. We say a class $\mathcal{C}$ is **weakly learnable** if there exists $\gamma > 0$ and an algorithm $\mathcal{A}$ such that for all $\delta > 0$, with number of examples polynomial in $1/\delta$,

$$\Pr\left(\mathrm{err}_{\mathcal{D}}(h) \leq \frac{1}{2} - \gamma\right) \geq 1 - \delta.$$

We can summarize the comparison in the following table.

strong learnable	weak learnable
$\exists$ algorithm $\mathcal{A}$	$\exists$ algorithm $\mathcal{A}$
	$\boldsymbol{\exists\gamma > 0}$
$\forall$ concept $c \in \mathcal{C}$	$\forall$ concept $c \in \mathcal{C}$
$\forall$ distribution $\mathcal{D}$	$\forall$ distribution $\mathcal{D}$
$\boldsymbol{\forall\epsilon > 0}$	
$\forall\delta > 0$	$\forall\delta > 0$
given $m = \mathrm{poly}(1/\epsilon, 1/\delta)$	given $m = \mathrm{poly}(1/\delta)$
output $h \in \mathcal{H}$	output $h \in \mathcal{H}$
$\Pr(\mathrm{err}_{\mathcal{D}}(h) \leq \epsilon) \geq 1 - \delta$	$\Pr(\mathrm{err}_{\mathcal{D}}(h) \leq \frac{1}{2} - \gamma) \geq 1 - \delta$

The question now is if we have a weak learning algorithm that does a little bit better than random, can we convert it to a strong learning algorithm? In more abstract terms, are these two notions of learnability equivalent? The answer turns out that we can, that the models are equivalent. It turns out that if we are given a weak learning algorithm that satisfies this criterion, we can convert it into a strong learning algorithm. We will talk about it in detail in the next couple of lectures.