Machine Learning Theory
Tübingen University, WS 2016/2017
Lecture 9

Tolstikhin Ilya

Abstract

In this lecture we start with very shortly discussing questions behind AdaBoost. Next we recap the basic setting of the prediction problems, this time with *real-valued* loss functions. We will introduce the *Rademacher Complexity* and show that it upper bounds the uniform deviations of expected loss from the empirical one over the class of predictors. Blue colour will be used to highlight parts appearing in the upcoming homework assignments.

1 AdaBoost: recap

In previous lectures we introduced the AdaBoost algorithm. Consider a class B of *base predictors* mapping $\mathcal{X}$ to $\{-1, +1\}$, and a *weak learner*, which, provided with a weighted training sample $\{S_n, W\}$, returns an element $\hat{h}_n$ of B with a relatively low weighted empirical loss. AdaBoost is an iterative procedure, which, starting from the uniform weights over S_n, on every t-th step (a) runs the weak learner on the weighted training sample, resulting in $h_t \in B$, (b) computes the corresponding weight α_t, and (c) updates the weights of the training examples. After T iterations AdaBoost outputs the composition, which is a weighted majority vote:

$$\hat{h}_n^T(x) := \operatorname{sgn}\left(\sum_{i=1}^{T} \alpha_i h_i(X)\right).$$

This classifier belongs to the following set:

$$L(B,T) := \left\{ f\colon f(x) = \operatorname{sgn}\left(\sum_{i=1}^{T} \beta_i g_i(x)\right), g_i \in B, \beta_i \in \mathbb{R} \right\}.$$

It is clear that T controls the bias-variance (or estimation-approximation) tradeoff of the learning problem. Indeed, as T increases, the function class $L(B,T)$ grows, i.e. becomes more flexible and rich. As a result, the best classifier of $L(B,T)$ gets closer and closer to the Bayes optimal classifier h^* as $T \to \infty$. On the other hand, it can be shown that the VC-dimension of $L(B,T)$ also grows with T (approximately as $T \cdot \mathrm{VC}(B)$). This shows that we may end up overfitting, as we already know that the test performance of the learned predictors is controlled in terms of the VC dimension of the function class, i.e. with probability at least $1-\delta$ over the training sample S_n:

$$L(\hat{h}_n^T) - \inf_{h \in \mathcal{H}} L(h) \le O\left(\frac{\mathrm{VC}\big(L(B,T)\big)\log n + \log(1/\delta)}{n}\right). \qquad \text{(VC-Bound)}$$

The same argument may be used to bound the *generalization error* of the predictors:

$$L(\hat{h}_n^T) \le L_n(\hat{h}_n^T) + O\left(\frac{\mathrm{VC}\big(L(B,T)\big)\log n + \log(1/\delta)}{n}\right). \tag{1}$$

In the previous lectures we showed that as the empirical loss of the AdaBoost composition $L_n(\hat{h}_n^T)$ decreases to zero exponentially fast with a number of iterations T. I.e., the first term of (1) decreases to zero exponentially with $T \to \infty$. Actually, we showed that under certain "weak learnability" conditions, it approaches zero in a *finite number of steps*. Meanwhile, the second term of (1) is expected to grow with increasing T. Indeed, we already saw that VC-dimension of $L(B,T)$ is essentially of the order $\tilde{O}(\sqrt{T \cdot \mathrm{VC}(B)/n})$, where $\tilde{O}$ hides constant and logarithmic factors. In other words, the last term grows sublinearly with T. Together, these facts may be used to partially explain a nice empirical behaviour of AdaBoost algorithm.

However, it was empirically noticed that the test error of AdaBoost composition keeps decreasing (as T increases) even after $L_n(\hat{h}_n^T)$ approaches zero. This is indeed surprising: at some point we would expect that $L(B,T)$ becomes too large and as a result AdaBoost should overfit, leading to the increasing test error. Unfortunately, the VC-bound (1) is not enough to explain this nice behaviour. In the following lectures we will develop a slightly new argument (compared to VC theory), which will help us to partially address this issue.

2 Prediction problems with real-valued losses

In previous lectures we considered only with binary losses, i.e. $\ell(y', y'') \in \{0, 1\}$. This is a natural choice in situations when your outputs Y may take only two values (i.e., binary classification problems). However, there are many applied problems where on needs to predict a real-valued output $Y \in \mathbb{R}$ based on the observed inputs X (features). It is clear that the binary loss does not always make sense in these situations. Indeed, imagine predicting a height of a person based on his/her weight. Consider a case where you predicted 180 cm while the true answer was 190. Binary loss would return value 1. Does it tell us much? Perhaps, not so much. Indeed, apart from whether we guessed the height correctly or not, we also want to know *how close was our prediction from the truth.*

Next we introduce the setting of statistical learning theory for real-valued outputs (these problems are usually called *regression* problems), which is almost identical to the problem of binary classification considered throughout the course.

The setting Consider an input space $\mathcal{X}$ and a space $\mathcal{Y} = \mathbb{R}$, which is a real line. Consider any real-valued loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. There is an unknown distribution P over the product space $\mathcal{X} \times \mathcal{Y}$, which spits the input-output tuples (X, Y). We observe a training sample $S_n := \{(X_i, Y_i)\}_{i=1}^n$ sampled i.i.d. from P. We fix a certain class $\mathcal{H}$ of functions mapping $\mathcal{X}$ to $\mathcal{Y}$. For any predictor $h \in \mathcal{H}$ we define its expected loss as $L(h) := \mathbb{E}_{(X,Y)\sim P}\big[\ell\big(h(X), Y\big)\big]$ and empirical loss $L_n(h) := \frac{1}{n}\sum_{i=1}^n \ell\big(h(X_i), Y_i\big)$. Overall, our goal is to find a predictor $h \in \mathcal{H}$ with the value of $L(h)$ as small as possible. We will denote h^* the predictor having smallest expected loss in $\mathcal{H}$, i.e. $L(h^*) = \inf_{h\in\mathcal{H}} L(h)$. A more ambitious goal would be to find the *Bayes optimal* predictor h^B, which is the function having minimal expected loss among *all the measurable functions mapping* $\mathcal{X}$ *to* $\mathcal{Y}$. We will call the expected loos of Bayes predictor $L(h^B)$ the *Bayes optimal risk.*

Short motivation: squared loss Let us shortly motivate this setting. Consider a *squared loss*: $\ell(y', y'') = (y' - y'')^2$, which is a very popular loss of choice in applied problems. It turns

out that, under certain mild conditions on the P, the function achieving Bayes optimal risk is $h^B(x) = \mathbb{E}[Y|X = x]$, i.e. it holds that $L(\mathbb{E}[Y|X = x]) = \inf_f L(f)$, where infimum is over all the measurable predictors (HW). In other words, if we could indeed minimize $L(h)$ for quadratic loss we would arrive at conditional expected value $\mathbb{E}[Y|X = x]$. This function is indeed a very nice and reasonable predictor for a real-valued prediction problems. When presented with input value x it returns the average (expected) value of outputs observed at the location x.

Error bounds Now that we agreed that this setting makes sense, how do we solve the regression problems? Can we use the same empirical risk minimization (ERM) algorithm again? Recall that for a binary classification nice properties of ERM algorithm were justified by the VC-bound. But does VC bound hold for this real-valued output problems? The answer is "no" and the reason is simple: the VC dimension is not even defined for the real-valued loss functions, as it was heavily based on the combinatorial nature of sets of binary vectors, induced by the losses of classifiers on the training samples. Now that our loss is real-valued, we don't have binary vectors any more. And thus VC analysis is simply not applicable. Then, what do we do?

3 Rademacher Complexity

It turns out that we can introduce the following complexity measure, called *Rademacher Complexity*:

Definition 1. *Consider a class of functions $\mathcal{F}$ defined on some space $\mathcal{Z}$ and mapping to $[-1, 1]$. Consider a sequence $Z_1, \dots, Z_n$ of independent random variables distributed according to P, defined over $\mathcal{Z}$. Let $\sigma_1, \dots, \sigma_n$ be independent random variables taking values $+1$ and -1 with probabilities $1/2$.*

The following quantity is called the **conditional Rademacher complexity***:*

$$\hat{R}_n(\mathcal{F}) := \mathbb{E}_{\sigma_1,\dots,\sigma_n}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(Z_i)\Big| Z_1, \dots, Z_n\right]$$

and the following is called the **Rademacher complexity***:*

$$R_n(\mathcal{F}) := \mathbb{E}_{Z_1,\dots,Z_n}[\hat{R}_n(\mathcal{F})].$$

Note that these quantities indeed measure the complexity of the function class $\mathcal{F}$. For $\mathcal{F} = \{f\}$, i.e. a function class consisting only of one function, we obviously have $\hat{R}_n(\mathcal{F}) = R_n(F) = 0$. For $\mathcal{F}$ containing all the measurable functions we have $\hat{R}_n(\mathcal{F}) = R_n(F) = 1$. It also can be shown that $\hat{R}_n(\mathcal{F})$ and $R_n(\mathcal{F})$ are both bounded in $[0, 1]$ interval (HW). In other words, we get the highest value of Rademacher Complexity (RC) for the largest possible function class and the smallest possible value for the least flexible $\mathcal{F}$, containing only one function. Moreover, it is clear that if $\mathcal{F} \in \mathcal{F}'$ then $R_n(\mathcal{F}) \le R_n(\mathcal{F}')$ and $\hat{R}_n(\mathcal{F}) \le \hat{R}_n(\mathcal{F}')$ (HW), i.e. RC grows monotonically.

It turns out that Rademacher complexity plays an important role in quantifying the performance of ERM. Recall that if h_n^{ERM} is the outcome of ERM on the training sample S_n, i.e.:

$$L_n(h_n^{ERM}) = \inf_{h\in\mathcal{H}} L_n(h),$$

then

$$L(h_n^{ERM}) - \inf_{h\in\mathcal{H}} L(h) \le \sup_{h\in\mathcal{H}}\big(L(h) - L_n(h)\big) + \big(L_n(h^*) - L(h^*)\big). \tag{2}$$

Last term, where h^* is the best predictor in $\mathcal{H}$, can be upper bounded using a simple Hoeffding's inequality to give $\big(L_n(h^*) - L(h^*)\big) \leq \sqrt{\frac{\log(1/\delta)}{2n}}$ with probability at least $1-\delta$. Moreover, for any $h \in \mathcal{H}$ it obviously holds that:

$$L(h) \leq L_n(h) + \sup_{h\in\mathcal{H}}\big(L(h) - L_n(h)\big). \tag{3}$$

We thus see that the quantity $\sup_{h\in\mathcal{H}}\big(L(h) - L_n(h)\big)$ upper bounds the excess risk of ERM and generalization error of the class $\mathcal{H}$. The following result established the relation between these quantities and RC:

Theorem 2. *Consider any class of real-valued predictors $\mathcal{H}$ and any loss function ℓ, such that $\ell\big(h(X), Y\big) \in [0,1]$ for all $X \in \mathcal{X}$, $Y \in \mathcal{Y}$, and $h \in \mathcal{H}$. The for any $\delta \in (0,1)$ with probability at least $1-\delta$ over the training sample S_n we have*

$$\sup_{h\in\mathcal{H}}\big(L(h) - L_n(h)\big) \leq 2 \cdot R_n(\ell \circ \mathcal{H}) + 2\sqrt{\frac{2\log(1/\delta)}{n}}.$$

Remark 1. The function class $\ell \circ \mathcal{H}$, appearing in the theorem, is defined as

$$\big\{\ell_h(x,y) = \ell\big(h(x), y\big),\ h \in \mathcal{H}\big\}.$$

This is *the loss class* associated with the class of predictors $\mathcal{H}$. In order to understand how the definition of RC can be applied to our setting, we set $Z_i = (X_i, Y_i)$ and $\mathcal{F} = \ell \circ \mathcal{H}$. I.e.

$$R_n(\ell \circ \mathcal{H}) = \mathbb{E}_{S_n}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left[\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\ell\big(h(X_i), Y_i\big)\right].$$

Next we prove this theorem. First we need the following concentration inequality:

Theorem 3 (McDiarmid's inequality)**.** *Consider a function f defined over $\mathcal{Z}^n$ and mapping to $\mathbb{R}$. Assume it satisfies the following bounded difference condition:*

$$\sup_{z_1,\ldots,z_n,z'} |f(z_1,\ldots,z_n) - f(z_1,\ldots,z_{i-1},z',z_{i+1},\ldots,z_n)| \leq c_i,$$

where $c_1,\ldots,c_n$ are nonnegative constants. Consider a sequence of random variables $Z_1,\ldots,Z_n$ sampled from distribution P defined over $\mathcal{Z}$. Introduce a random variable $G = f(Z_1,\ldots,Z_n)$. Then with probability larger than $1-\delta$ it holds that:

$$G \leq \mathbb{E}[G] + \sqrt{\frac{1}{2}\left(\sum_{i=1}^{n} c_i^2\right)\log(1/\delta)}.$$

This is a rather powerful and useful result, which allows us to bound the random variable using its expectation. If $c_1,\ldots,c_n$ are relatively small, the bounded difference condition tells us that the function f does not depend too much on any of its arguments. It also turns out that McDiarmid's inequality generalizes Hoeffding's inequality for sums of random variables (HW).

Now set $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, $Z_i := (X_i, Y_i)$, and consider the following function:

$$f(Z_1,\ldots,Z_n) = \sup_{h\in\mathcal{H}}\big(L(h) - L_n(h)\big).$$

First of all, convince yourself that it is indeed a function mapping $\mathcal{Z}^n$ to $\mathbb{R}$. Next, you can show that this function satisfies the bounded difference condition of McDiarmid's inequality with $c_i = 1/n$ (HW). We may conclude that with probability at least $1 - \delta$ it holds that

$$\sup_{h \in \mathcal{H}} \big(L(h) - L_n(h) \big) \leq \mathbb{E}_{S_n} \left[\sup_{h \in \mathcal{H}} \big(L(h) - L_n(h) \big) \right] + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

We may conclude the proof of Theorem 2 using the following result:

Theorem 4 (Rademacher Symmetrization)**.** *Consider a class of functions $\mathcal{F}$ defined on some space $\mathcal{Z}$ and mapping to $\mathbb{R}$. Consider a sequence $Z_1, \ldots, Z_n$ of independent random variables distributed according to P, defined over $\mathcal{Z}$. Then*

$$\mathbb{E}_{Z_1, \ldots, Z_n} \left[\sup_{f \in \mathcal{F}} \left(\mathbb{E}_{Z \sim P}[f(Z)] - \frac{1}{n} \sum_{i=1}^{n} f(Z_i) \right) \right] \leq 2 \cdot R_n(\mathcal{F}).$$

This theorem is extremely easy to prove by (a) introducing *the ghost* sample $Z'_1, \ldots, Z'_n$, (b) writing $\mathbb{E}_{Z \sim P}[f(Z)] = \mathbb{E}_{Z'_1, \ldots, Z'_n} \left[\frac{1}{n} \sum_{i=1}^{n} f(Z'_i) \right]$, and (c) using Jensen's inequality (HW).

4 Discussion

Theorem 2 tells us that the uniform deviations $\sup_{h \in \mathcal{H}} \big(L(h) - L_n(h) \big)$ are essentially upper bounded by the Rademacher complexity of the loss class associated with $\mathcal{H}$. In particular, in view of (2), if $R_n(\ell \circ \mathcal{H}) \to 0$ as $n \to \infty$ then the performance of ERM achieves the best in the class. In the following lectures and homeworks we will see how to bound Rademacher Complexities of function classes. For instance, it happens that for a binary classification problems $R_n(\mathcal{H})$ may be upper bounded by $O(\sqrt{\mathrm{VC}(\mathcal{H})/n})$, leading to the VC-bound. Also we will see that RC is very useful when it comes to AdaBoost and Support Vector Machines and kernel methods in general.