Machine Learning Theory (CSC 482A/581A) - Lectures 19 & 20

Nishant Mehta

1 Empirical risk / consistency analysis for AdaBoost

Recall that our goal is to prove that weak learnability implies strong learnability. A first natural step is to ensure that we have obtained low empirical risk on the sample. This guarantee is captured by the following theorem:

Theorem 1. *If AdaBoost is run for T rounds on a training sample of size n, then*

$$\frac{1}{n} \sum_{j=1}^{n} \mathbf{1} \left[\hat{f}(X_j) \neq Y_j \right] \leq \exp \left(-2 \sum_{t=1}^{T} \gamma_t^2 \right).$$

Recall that

$$\varepsilon_t = \Pr_{j \sim D_t} \left(h_t(X_j) \neq Y_j \right) = \frac{1}{2} - \gamma_t.$$

A proof of Theorem 1 *(which we did in class)* can be found in Schapire's lecture notes[1].

Now, suppose that each $\gamma_t \geq \gamma$; this can be arranged with high probability as explained below. Then consistency must hold as soon as $e^{-2T\gamma^2} < \frac{1}{n}$, i.e. as soon as $T \geq \frac{\log n}{2\gamma^2}$.

Now, let's see how to ensure, with high probability over the internal randomization of the algorithm, that each $\gamma_t \geq \gamma$. Assume that $\mathcal{A}$ is a weak learner for $\mathcal{C}$ with edge $\gamma > 0$ and success probability $\delta_0 > 0$, and consider running AdaBoost with boosting-by-resampling. That is, in obtaining the weak hypothesis h_t for each $t \in [T]$, we repeat the following procedure enough times until either Step 3 succeeds *or* until we have repeated the procedure enough times where the probability of every Step 3 failing is as small as desired:

Step 1. Resample from D_t: obtain a sample of size n_0 from D_t.

Step 2. Call the weak learner on this sample, obtaining candidate hypothesis h_t.

Step 3. Verify that $\gamma_t \geq \gamma$.

Taking the union bound over $[T]$, we can also ensure that with high probability every $\gamma_t \geq \gamma$. Thus, by allowing for $\Omega\left(\log \frac{T}{\delta}\right)$ rounds of resampling for each t, we can ensure with high probability at least $1 - \delta$ that each $\gamma_t \geq \gamma$, which, by Theorem 1 with $T \geq \frac{\log n}{2\gamma^2}$ implies (with the same probability) that $\hat{f}$ is consistent with c on the training sample.

[1] http://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0303.pdf.

2 ~~Risk guarantees for AdaBoost~~ Compression bounds

Of course, ensuring consistency of AdaBoost's weighted majority hypothesis $\hat{f}$ on the training sample is only a first step. Our real goal is to ensure that $\Pr_{X \sim P}\left(\hat{f}(X) \neq c(X)\right) \leq \varepsilon$ with probability at least $1 - \delta$ (with sample complexity polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$).

It turns out that the boosting-by-resampling variant of AdaBoost is well-matched to a risk analysis based on compression bounds: these are risk bounds based on compression schemes, which we will now study. This is a bit of a detour, but in the next lecture, we will reconnect compression schemes to our analysis of AdaBoost and complete the proof that weak learnability implies strong learnability.

2.1 Compression schemes

For any concept $c \in \mathcal{C}$ and any $x \in \mathcal{X}$, let $L_c(x) = (x, c(x))$, so that L_c takes an example and returns a labeled example. We extend L_c to sequences of examples $\mathbf{x}^n = (x_1, \ldots, x_n) \in \mathcal{X}^n$ via $L_c(\mathbf{x}^n) = ((x_1, c(x_1)), \ldots, (x_n, c(x_n)))$. With this extension, L_c sends unlabeled samples to labeled samples.

Definition 1. For a concept class $\mathcal{C}$, a compression scheme with kernel size k is a pair of mappings, the *compression map*

$$\kappa \colon \bigcup_{n=k}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to (\mathcal{X} \times \mathcal{Y})^k$$

and the *reconstruction map*

$$\rho \colon (\mathcal{X} \times \mathcal{Y})^k \times \mathcal{X} \to \mathcal{Y},$$

which satisfy the following properties:

- for all $c \in \mathcal{C}$ and all $n \geq k$, the compression set $\kappa(L_c(\mathbf{x}^n))$ is a subsequence of $L_c(\mathbf{x}^n)$ for all $\mathbf{x}^n \in \mathcal{X}^n$;
- for all $c \in \mathcal{C}$ and all $n \geq k$, for any $x_j \in \mathbf{x}^n$, $\rho(\kappa(L_c(\mathbf{x}^n)), x_j) = c(x_j)$.

In words, a compression map sends every finite labeled sample to a compression set that is a size-k subset of that sample. Given the compression set of a labeled sample, the reconstruction map can be viewed as a classifier $\hat{f}_{\rho,\kappa}$ which consistently labels each example within the unlabeled sample; that is, the hypothesis $\hat{f}_{\rho,\kappa} \colon x \mapsto \rho(\kappa(L_c(\mathbf{x}^n)), x)$ is consistent with c on the sample $\mathbf{x}^n$.

Compression schemes are interesting in their own right, and there are numerous examples of compression schemes. We will study them in more detail later in this course. For now, let's look at one particularly simple example.

Example 1 (Axis-aligned rectangles)**.** Take $\mathcal{C}$ to be the class of axis-aligned rectangles in $\mathbb{R}^2$. Given any labeled sample, clearly the tightest rectangle that contains all the positively labeled examples is consistent with the training sample. Moreover, we can reconstruct this rectangle using only 4 points (and in some cases even less): just choose the leftmost, rightmost, topmost, and bottommost positive examples. Therefore, there is a compression scheme of size 4.

The existence of a finite-kernel compression scheme for a class $\mathcal{C}$ implies a remarkably-simple-to-derive PAC-style risk bound.

Theorem 2. *Let (κ, ρ) be a compression scheme for $\mathcal{C}$ with kernel size k. Suppose that $X_1, \ldots, X_n$ are drawn independently from P and labeled according to $c \in \mathcal{C}$, and denote by $\hat{f}_{\kappa,\rho}$ the hypothesis defined as $\hat{f}_{\kappa,\rho} \colon x \mapsto \rho(\kappa(L_c(\mathbf{X}^n)), x)$. With probability at least $1 - \delta$,*

$$\Pr_{X \sim P}\left(\hat{f}_{\kappa,\rho}(X) \neq c(X)\right) \leq \frac{k \log n + \log \frac{1}{\delta}}{n - k}.$$

Proof. Let $\mathcal{T}$ be the set of all cardinality-k subsets of $\{1, \ldots, n\}$. Consider a fixed subset $S \in \mathcal{T}$, and let $\mathbf{X}_S$ denote the subsequence of examples $(X_j)_{j \in S}$. Corresponding to S is a hypothesis

$$\hat{f}_{S,\rho} \colon x \mapsto \rho(L_c(\mathbf{X}_S), x).$$

Observe that $\hat{f}_{S,\rho}$ depends only on $\mathbf{X}_S$. Since $\mathbf{X}_S$ is independent of $\mathbf{X}_{[n] \setminus S}$, it also is true that $\hat{f}_{S,\rho}$ is independent of $\mathbf{X}_{[n] \setminus S}$. Now, for each fixed S, the probability that $\hat{f}_{S,\rho}$ has risk more than ε and yet is consistent with c on $\mathbf{X}_{[n] \setminus S}$ is at most

$$(1 - \varepsilon)^{n-k}. \tag{1}$$

Applying a union bound over all $\binom{n}{k} \leq n^k$ elements of $\mathcal{T}$, the probability that $\hat{f}_{\kappa,\rho}(x)$ has risk more than ε and yet is consistent with c on $\mathbf{X}^n$ is at most

$$n^k (1 - \varepsilon)^{n-k} \leq n^k e^{-(n-k)\varepsilon}.$$

The result follows by inversion. □

In our analysis of AdaBoost, we will see that we can encode each weak hypothesis h_t using a size-n_0 subset of the training sample. Therefore, given T such subsets, we can encode all of the weak hypotheses *if* we know which subset should encode which hypothesis.[2] However, in the basic compression scheme defined above, the compression map only returns a subsequence of the training sample, and therefore it is not possible to determine which elements of the compression set should be used to encode which weak hypothesis. If we could also encode a permutation of the compression set, then we could appropriately permute the compression set and then take the first n_0 examples to encode h_1, the second n_0 examples to encode h_2, and so on. A quick argument shows that we can encode a permutation over k elements using $\log(k!) \leq k \log k$ bits. Encoding such a permutation motivates an extended version of compression schemes.

[2] Technically, we cannot use a compression scheme as defined above to analyze AdaBoost for the following reason: when using boosting-by-resampling, it is possible that, for a given $j \in [n]$, the example (x_j, y_j) appears in the compression set for some weak hypothesis h_t multiple times; however, a subsequence of the training sample can only include each point x_j once. Therefore, we cannot necessarily encode h_t using a subsequence of the training sample. Moreover, the problem becomes even worse when we need to encode $h_1, \ldots, h_T$ using a single subsequence of the training sample. There is a simple workaround, which involves instead allowing for repeats. Allowing for repeats simply means that we need to consider n^k possible compression sets, and since we already used the loose upper bound $\binom{n}{k} \leq n^k$ in our proof of Theorem 2, that result holds even for this more general form of compression scheme. The reason we looked at the subsequence-based definition of compression schemes is that this is the standard one that appears in the literature.

Definition 2. For a concept class $\mathcal{C}$, an extended compression scheme with kernel size k and side information set $\mathcal{I}$ is a pair of mappings, the *compression map*

$$\kappa\colon \bigcup_{n=k}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to (\mathcal{X} \times \mathcal{Y})^k \times \mathcal{I}$$

and the *reconstruction map*

$$\rho\colon (\mathcal{X} \times \mathcal{Y})^k \times \mathcal{I} \times \mathcal{X} \to \mathcal{Y},$$

which satisfy the following properties:

- for all $c \in \mathcal{C}$ and all $n \geq k$, the compression set part of $\kappa(L_c(\mathbf{x}^n))$ is a subsequence of $L_c(\mathbf{x}^n)$ for all $\mathbf{x}^n \in \mathcal{X}^n$;
- for all $c \in \mathcal{C}$ and all $n \geq k$, for any $x_j \in \mathbf{x}^n$, $\rho(\kappa(L_c(\mathbf{x}^n)), x_j) = c(x_j)$.

Theorem 3. *Let (κ, ρ) be an extended compression scheme for $\mathcal{C}$ with kernel size k and side information set $\mathcal{I}$. Suppose that $X_1, \ldots, X_n$ are drawn independently from P and labeled according to $c \in \mathcal{C}$, and denote by $\hat{f}_{\kappa,\rho}$ the hypothesis defined as $\hat{f}_{\kappa,\rho}\colon x \mapsto \rho(\kappa(L_c(\mathbf{X}^n)), x)$. With probability at least $1 - \delta$,*

$$\Pr_{X \sim P}\left(\hat{f}_{\kappa,\rho}(X) \neq c(X)\right) \leq \frac{\log|\mathcal{I}| + k \log n + \log \frac{1}{\delta}}{n - k}.$$

The proof is essentially the same as the proof for basic compression schemes. The only difference is that we now run the argument in the proof of Theorem 2 but also for each fixed value of the side information $i \in \mathcal{I}$, yielding hypotheses $\hat{f}_{S,i,\rho}$, and so we also take the union bound over $\mathcal{I}$.

2.2 Weak learnability implies strong learnability

As argued above, we can encode the T weak hypotheses learned by AdaBoost using $T \cdot n_0$ examples; however, the final hypothesis $\hat{f}$ of AdaBoost also requires the weight variables $\alpha_1, \ldots, \alpha_T$, since $\hat{f}$ is the weighted majority predictor

$$x \mapsto \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right);$$

therefore, we cannot immediately apply Theorem 3 to get a risk bound for AdaBoost. Suppose for the time being that we instead predicted according to a simple majority

$$x \mapsto \operatorname{sgn}\left(\sum_{t=1}^{T} h_t(x)\right).$$

In this case, we *can* directly apply Theorem 3 to obtain a risk bound. To see this, we define an extended compression scheme as follows. For each $t \in [T]$, let $\mathbf{z}^{(t)} := \Big((x_{j_{t,1}}, y_{j_{t,1}}), \ldots, (x_{j_{t,n_0}}, y_{j_{t,n_0}})\Big)$ be a subsequence of $((x_1, y_1), \ldots, (x_n, y_n))$ such that

$$\mathcal{A}\left(\mathbf{z}^{(t)}\right) = h_t. \tag{2}$$

Concatenate the subsequences $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ and sort the resulting sequence so that the examples appear in the same order as they appear in the training sample. The sorted sequence is then a subsequence of the training sample. Let Π be the permutation which puts the examples back into their original order (before this sorting).

We define the compression map κ to be the function which takes in the training sample and outputs the sorted version of $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ as the compression set and Π as the side information. The side information set is thus the set of all permutations over $T \cdot n_0$ elements.

Next, we define the reconstruction map ρ. Observe that, given the compression set and Π, we can reconstruct the concatenation of subsequences $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$. In light of (2), we also can reconstruct $h_1, \ldots, h_T$. Define the reconstruction map ρ to be the function which takes in the compression set, Π, and an input x and outputs

$$\operatorname{sgn}\left(\sum_{t=1}^{T} h_t(x) \right).$$

Define $k = T \cdot n_0 = \frac{n_0 \log n}{2\gamma^2}$; clearly, $k = O(\log n)$. Then, since γ and n_0 are constants, we have with probability at least $1 - \delta$, the risk of the simple majority classifier is at most

$$\frac{k \log k + k \log n + \log \frac{1}{\delta}}{n - k} = O\left(\frac{(\log n)^2 + \log \frac{1}{\delta}}{n} \right).$$

Thus, if we did not have to encode $\alpha_1, \ldots, \alpha_T$, by choosing n to grow roughly as $\frac{1}{\varepsilon}$ (ignoring logarithmic factors), we can ensure that the risk of the simple majority classifier is as at most ε.

2.3 Upgraded version of analysis for AdaBoost

To obtain a risk bound for AdaBoost, we will use a hybrid approach that mixes compression schemes with a VC dimension-type analysis. For the time being, suppose that $S \in \mathcal{T}$ (corresponding to a compression set) and $i \in \mathcal{I}$ (corresponding to the side information) are fixed, inducing a sequence of weak hypotheses $h_1, \ldots, h_T$. Let's consider the restricted class of hypotheses used by AdaBoost for such fixed $h_1, \ldots, h_T$.

Define the feature map φ as

$$\varphi(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_T(x) \end{pmatrix}.$$

Letting $\alpha = (\alpha_1, \ldots, \alpha_T)$, observe that the hypothesis of AdaBoost is equal to $x \mapsto \operatorname{sgn}(\alpha \cdot \varphi(x))$ for some $\alpha \in \mathbb{R}_+^T$. Thus, if $h_1, \ldots, h_T$ are fixed, then this hypothesis is a homogeneous linear separator (also called a linear threshold function) for points in the new representation $\varphi(x)$. In the second problem set, you proved that the VC dimension of the class of general (i.e. not necessarily homogeneous) halfspaces in T dimensions is $T + 1$. It turns out that the VC dimension of the class of linear threshold functions in T dimensions is *exactly* T. Proving this latter VC dimension result is considerably easier than the case of general halfspaces, but since the difference is just 1, we will not cover a proof here. Thus, we have the following upper bound on the VC dimension of the restricted class:

$$\begin{aligned} &\operatorname{VCdim}\left(\left\{ \operatorname{sgn}(\alpha \cdot \varphi(x) : \alpha \in \mathbb{R}_+^T \right\} \right) \\ &\leq \operatorname{VCdim}\left(\left\{ \operatorname{sgn}(\alpha \cdot \varphi(x) : \alpha \in \mathbb{R}^T \right\} \right) \\ &= T. \end{aligned}$$

Now, for every fixed S and i, let $\mathcal{F}_{S,i}$ be the corresponding restricted class of linear threshold functions. We can thus decompose the class of hypotheses from which AdaBoost predicts into

$$\bigcup_{S \in \mathcal{T}, i \in \mathcal{I}} \mathcal{F}_{S,i},$$

where, we have $\text{VCdim}(\mathcal{F}_{S,i}) \leq T$ for all S and i.

Retracing through the proof of Theorem 3, we can replace our bound (1) on the probability that a *single* hypothesis $\hat{f}_{S,i,\rho}$ is consistent with the examples not in the compression set and yet has risk exceeding ε with the probability that *any* hypothesis in $\mathcal{F}_{S,i}$ is consistent with the examples not in the compression set and yet has risk exceeding ε. For this, we use the following result from Theorem 1 of Lecture 11:

Theorem 4. *Let $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ be a VC class with $\text{VCdim}(\mathcal{F}) = V$, and let $\hat{f}$ be an ERM classifier (which, given a training sample, outputs a hypothesis in $\mathcal{F}$ that minimizes the empirical risk), and let P be an arbitrary probability distribution P over $\mathcal{X} \times Y$ that satisfies $Y = c(X)$ for some $c \in \mathcal{F}$.*

Then for any $n \geq V$, and any $\varepsilon > 0$.

$$\Pr\left(R(\hat{f}) > \varepsilon\right) \leq 2\left(\frac{2en}{V}\right)^V e^{-n\varepsilon/2}.$$

Replacing (1) with $2\left(\frac{2en}{T}\right)^T e^{-n\varepsilon/2}$ and setting the failure probability to $\frac{\delta}{2}$ (since we also need to allow for boosting-by-resampling to fail with some probability $\frac{\delta}{2}$), it remains to solve for ε in the following expression to obtain a risk bound for AdaBoost (with $k = T \cdot n_0 = \frac{n_0 \log n}{2\gamma^2}$ as before):

$$k! \cdot n^k \cdot 2\left(\frac{2e(n-k)}{T}\right)^T e^{-(n-k)\varepsilon/2} = \frac{\delta}{2}.$$

We thus have the following strong learning guarantee for AdaBoost:

Theorem 5. *Assume that $n \geq \frac{n_0 \log n}{\gamma^2}$, so that $n - k \geq \frac{n}{2}$. Let AdaBoost be run for $T = \frac{\log n}{\gamma^2}$ rounds with a weak learner $\mathcal{A}$ for concept class $\mathcal{C}$ with edge γ and success probability δ_0, and further assume that to obtain each weak hypothesis h_t, we make $\Omega(\frac{\log T}{\delta})$ resampling attempts.*

Then with probability at least $1 - \delta$, the risk of the hypothesis $\hat{f}$ returned by AdaBoost is at most

$$\Pr_{X \sim P}\left(\hat{f}(X) \neq c(X)\right) = O\left(\frac{(\log n)^2 + \log \frac{1}{\delta}}{n}\right),$$

where the $O(\cdot)$ notation hides the constants n_0 and γ.

By taking the sample size n to grow only slightly superlinearly (and hence polynomially) in $\frac{1}{\varepsilon}$ (it suffices to let n grow as $\frac{1}{\varepsilon}$, ignoring log factors), and by observing that the amount of computation used also is polynomial in $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$ (in fact, it only grows as $\log \frac{1}{\delta}$, we see that weak learnability implies strong learnability.