THEORETICAL MACHINE LEARNING COS 511

LECTURE #9

MARCH 1, 2016

LECTURER: ELAD HAZAN SCRIBE: ELAD HAZAN

In this lecture we consider a fundamental property of learning theory: it is amenable to *boosting*. Roughly speaking, boosting refers to the process of taking a set of rough "rules of thumb" and combining them into a more accurate predictor.

Consider for example the problem of Optical Character Recognition (OCR) in its simplest form: given a set of bitmap images depicting hand-written postal-code digits, classify those that contain the digit "1" from those of "0".

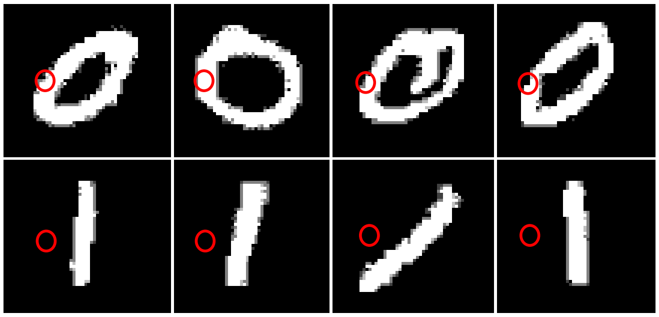

Figure 1: Distinguishing zero vs. one from a single pixle.

Seemingly, discerning the two digits seems a formidable task taking into account the different styles of handwriting, errors, etc. However, an inaccurate rule of thumb is rather easy to produce: in the bottom-left area of the picture we'd expect many more dark bits for "1"s than if the image depicts a "0". This is, of course, a rather inaccurate statement. It does not consider the alignment of the digit, thickness of the handwriting etc. Nevertheless, as a rule of thumb - we'd expect better-than-random performance, or some correlation with the ground truth.

The inaccuracy of the simplistic single-bit predictor is compensated by its simplicity. It is a rather simple task to code up a classifier based upon this rule which is very efficient indeed. The natural and fundamental question which arises now is: can several of these rules of thumb be combined into a single, accurate and efficient classifier?

In the rest of this note we shall formalize this question in the statistical learning theory framework. We then proceed to use the technology developed earlier in the course, namely regret minimization algorithms for OCO, to answer this question on the affirmative.

1 The problem of Boosting

We focus on statistical learnability rather than agnostic learnability. More formally, we assume the so called "realizability assumption", which states that for a learning problem over hypothesis class $\mathcal{H}$ there exists some $h^* \in \mathcal{H}$ such that $\text{error}(h^*) = 0$.

Definition 1.1 (Weak learnability)**.** The concept class $\mathcal{H} : \mathcal{X} \mapsto \mathcal{Y}$ is said to be γ-weakly-learnable if the following holds. There exists an algorithm $\mathcal{A}$ that accepts $S_T = \{(\mathbf{x}, y)\}$ and returns an hypothesis in $\mathcal{A}(S_T) \in \mathcal{H}$ that satisfies:
for any $\delta > 0$ there exists $T = T(\mathcal{H}, \delta, \gamma)$ large enough such that for any distribution $\mathcal{D}$ over pairs $(\mathbf{x}, y)$ and T samples from this distribution, it holds that with probability $1 - \delta$,

$$\text{error}(\mathcal{A}(S_t)) \leq \frac{1}{2} - \gamma$$

This is an apparent weakening of the definition of statistical learnability that we have described earlier: the error is not required to approach zero. The standard case of statistical learning in the context of boosting is called "strong learnability". An algorithm that achieves weak learning is referred to as a weak learner, and respectively we can refer to a strong learner as an algorithm that attains statistical learning for a certain concept class.

The central question of boosting can now be formalized: are weak learning and strong learning equivalent? In other words, is there a (hopefully efficient) procedure that has access to a weak oracle for a concept class, and returns a strong learner for the class?

Miraculously, the answer is affirmative, and gives rise to one of the most effective paradigms in machine learning, as we see next.

2 Boosting by OCO

In this section we describe a *reduction* from regret minimization to boosting.

2.1 Learning a finite sample

Our derivation focuses on simplicity rather than generality. As such, we make the following assumptions:

1. We restrict ourselves to the classical setting of binary classification. Boosting to real-valued losses is also possible, but outside our scope. Thus, we assume the loss function to be the zero-one loss, that is:

$$\ell(\hat{y}, y) = \begin{cases} 0 & y = \hat{y} \\ \\ 1 & 0/w \end{cases}$$

2. We assume that the concept class is realizable, i.e. there exists an $h \in \mathcal{H}$ such that $\text{error}(h) = 0$. There are results on boosting in the agnostic learning setting, but these are beyond our scope.

3. We assume that the distribution $\mathcal{D}$ is represented by a small finite sample $S = \{(\mathbf{x}_i, y_i) \ , \ i \in [m]\}$ of size $m = m(\epsilon)$ which depends on the final desired accuracy ϵ. The definition of learnability implies that indeed such a sample size m depends on the required approximation guarantee, probability of success and other properties of the hypothesis class itself such as the VC dimension. Thus, there is no loss of generality with this assumption.

4. For a sample $S \sim \mathcal{D}$, let $\text{error}_S(h)$ be the empirical error of an hypothesis $h \in \mathcal{H}$ on the sample, i.e.

$$\underset{S}{\text{error}}(h) \overset{\text{def}}{=} \mathbb{E}_{(\mathbf{x},y)\sim S}[h(\mathbf{x}) \neq y].$$

   We assume that any hypothesis $h \in \mathcal{H}$ that attains zero error on the sample S_m for $m = m(\epsilon)$ is guaranteed at most ϵ generalization error.

   This assumption is well justified in statistical learning theory, and its central theorems address exactly this scenario: taking a large enough sample and finding a hypothesis which is consistent with the sample (zero error on it), implies ϵ generalization error. However, the conditions in which the above holds are beyond our scope.

With these assumptions and definitions we are ready to prove our main result: a reduction from weak learning to strong learning using an OCO low-regret algorithm. Essentially, our task would be to find a hypothesis which attains zero error on a given sample.

2.2 Algorithm and analysis

Pseudocode for the boosting algorithm is given in Algorithm 1. The reduction above accepts as input a γ-weak learner and treats it as a black box, returning a function which we'll prove is a strong learner.

The reduction also accepts as input an online convex optimization algorithm denoted $\mathcal{A}^{OCO}$. The underlying decision set for the OCO algorithm is the m-dimensional simplex, where m is the sample size. Thus, its decisions are distributions over examples. The cost functions are linear, and assign a value of zero or one, depending on whether the current hypothesis errs on a particular example. Hence, the cost at a certain iteration is the expected error of the current hypothesis (chosen by the weak learner) over the distribution chosen by the low-regret algorithm.

Algorithm 1 Reduction from Boosting to OCO

```
Input: $\mathcal{H}, \epsilon, \delta$, OCO algorithm $\mathcal{A}^{OCO}$, $\gamma$-weak learning algorithm $\mathcal{A}^{WL}$, sample $S_m \sim \mathcal{D}$.
Set $T$ such that $\frac{1}{T}\text{Regret}_T(A^{OCO}) \leq \frac{\gamma}{2}$
Set disribution $\mathbf{p}_1 = \frac{1}{m}\mathbf{1} \in \Delta_m$ to be the uniform distribution.
for $t = 1, 2 \ldots T$ do
  Find hypothesis $h_t \leftarrow \mathcal{A}^{WL}(\mathbf{p}_t, \frac{\delta}{T})$
  Define the loss function $f_t(\mathbf{p}) = \mathbf{r}_t^\top \mathbf{p}$, where the vector $\mathbf{r}_t \in \mathbb{R}^m$ is defined as
      $\mathbf{r}_t(i) = \begin{cases} 1 & h_t(\mathbf{x}_i) = y_i \\ 0 & o/w \end{cases}$
  Update $\mathbf{p}_{t+1} \leftarrow \mathcal{A}^{OCO}(f_1, ..., f_t)$
end for
return $\bar{h}(x) = \text{sign}(\sum_{t=1}^{T} h_t(x))$
```

It is important to note that the final hypothesis $\bar{h}$ which the algorithm outputs does not necessarily belong to $\mathcal{H}$ - the initial hypothesis class we started off with.

Theorem 2.1. *Algorithm 1 returns a hypothesis $\bar{h}$ such that with probability at least $1 - \delta$,*

$$\underset{S}{\text{error}}(\bar{h}) = 0$$

Proof. Given $h \in \mathcal{H}$, we denote its empirical error on the sample S, weighted by the distribution $\mathbf{p} \in \delta_m$, by:

$$\underset{S,\mathbf{p}}{\text{error}}(h) = \sum_{i=1}^{m} \mathbf{p}(i) \cdot \mathbf{1}_{h(\mathbf{x}_i) \neq y_i}$$

Notice that by definition of $\mathbf{r}_t$ we have $\mathbf{r}_t^\top \mathbf{p}_t = 1 - \text{error}_{S,\mathbf{p}_t}(h_t)$. Since h_t is the output of a γ-weak-learner on the distribution $\mathbf{p}_t$, we have for all $t \in [T]$,

$$\begin{aligned}
\Pr[\mathbf{r}_t^\top \mathbf{p}_t \leq \frac{1}{2} + \gamma] &= \Pr[1 - \underset{S,\mathbf{p}_t}{\text{error}}(h_t) \leq \frac{1}{2} + \gamma] \\
&= \Pr[\underset{S,\mathbf{p}_t}{\text{error}}(h_t) \geq \frac{1}{2} - \gamma] \\
&\leq \frac{\delta}{2T}
\end{aligned}$$

This applies for each t separately, and by the union bound we have

$$\Pr[\frac{1}{T} \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{p}_t \leq \frac{1}{2} + \gamma] \leq \delta$$

Denote by $\mathbf{p}^*$ the uniform distribution over the samples from S that $\bar{h}$ misclassifies. Suppose there are N such samples, then:

$$\begin{aligned}
\sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{p}^* &= \sum_{t=1}^{T} \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}_{h_t(\mathbf{x}_j) = y_j} \\
&= \frac{1}{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \mathbf{1}_{h_t(\mathbf{x}_j) = y_j} \\
&\leq \frac{1}{N} \sum_{j=1}^{N} \frac{T}{2} && \bar{h}(\mathbf{x}_j) \neq y_j \\
&= \frac{T}{2}
\end{aligned}$$

Combining the pervious two observations, we have with probability at least $1 - \delta$ that

$$\begin{aligned}
\frac{1}{2} + \gamma &\leq \tfrac{1}{T} \textstyle\sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{p}_t \\
&\leq \tfrac{1}{T} \textstyle\sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{p}^* + \tfrac{1}{T} \text{Regret}_T(\mathcal{A}^{OCO}) && \text{low regret of } \mathcal{A}^{OCO} \\
&\leq \tfrac{1}{2} + \tfrac{1}{T} \text{Regret}_T(\mathcal{A}^{OCO}) \\
&\leq \tfrac{1}{2} + \tfrac{\gamma}{2}
\end{aligned}$$

This is a contradiction. We conclude that a distribution $\mathbf{p}^*$ cannot exist, and thus all examples in S are classified correctly.

□

2.3 AdaBoost

A special case of the template reduction we have described is obtained when the OCO algorithm is taken to be the Multiplicative Updates method we have come to know in the manuscript.

We have a bound of $O(\sqrt{T \log m})$ on the regret of the EG algorithm in our context. This bounds T in Algorithm 1 by $O(\frac{1}{\gamma^2} \log m)$.

Closely related is the AdaBoost algorithm, which is one of the most useful and successful algorithms in Machine Learning at large (see bibliography). Unlike the Boosting algorithm that we have analyzed, AdaBoost doesn't have to know in advance the parameter γ of the weak learners. Pseudo code for the AdaBoost algorithm is given in 2.

Algorithm 2 AdaBoost

```
Input: $\mathcal{H}, \epsilon, \delta$, $\gamma$-weak-learner $\mathcal{A}^{WL}$, sample $S_m \sim \mathcal{D}$.
Set $\mathbf{p}_1 \in \Delta_m$ be the uniform distribution over $S$.
for $t = 1, 2 \ldots T$ do
    Find hypothesis $h_t \leftarrow \mathcal{A}^{WL}(\mathbf{p}_t, \frac{\delta}{T})$
    Calculate $\epsilon_t = \text{error}_{S,\mathbf{p}_t}(h_t)$, $\alpha_t = \frac{1}{2} \log(\frac{1-\epsilon_t}{\epsilon_t})$
    Update,
        $$\mathbf{p}_{t+1}(i) = \frac{\mathbf{p}_t(i) e^{-\alpha_t h_t(i)}}{\sum_{j=1}^m \mathbf{p}_t(j) e^{-\alpha_t h_t(j)}}$$
end for
Return: $\bar{h}(x) = \text{sign}(\sum_{t=1}^T h_t(x))$
```

2.4 Completing the picture

In our discussion so far we have focused only on the empirical error over a sample. To show generalization and complete the Boosting theorem, one must show that

1. zero empirical error on a sample implies ϵ generalization error on the underlying distribution for a large enough sample.

2. Deal with the fact that the hypothesis returned by the Boosting algorithms does not belong to the original concept class.