COS-511: Learning Theory **Spring 2017**

Lecture 6: Deep Networks (take 1)

Lecturer: Roi Livni

6.1 Boosting

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".

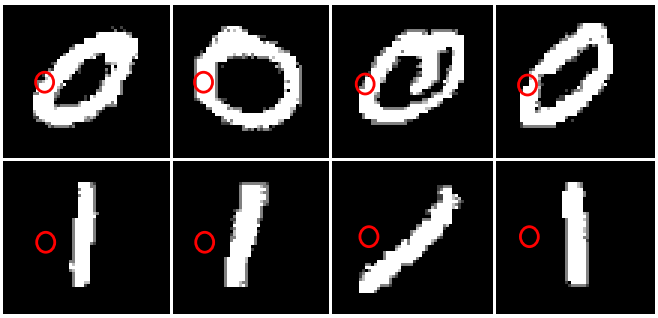

Figure 6.1: Distinguishing zero vs. one from a single pixle.

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.

6.2 The problem of Boosting

We focus on learnability in the realizable model 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 $\text{err}(h^*) = 0$.

Definition 6.1 (Weak learnability)**.** *The binary classification problem $(\mathcal{H}, \chi, \ell_{0,1})$ is said to be γ-weakly-learnable if the following holds. The exists a function $m : (0,1) \to \mathbb{N}$, and an algorithm A that accepts $S = \{(\mathbf{x}^{(i)}, y)\}_{i=1}^m$ and returns an hypothesis in $h_S^A \in \mathcal{H}$ that satisfies:*
if $m > m(\delta)$ and S is an IID sequence from some arbitray distribution, the with probability $1 - \delta$,

$$err(h_S^A(S)) \leq \frac{1}{2} - \gamma$$

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.

6.2.1 AdaBoost

The AdaBoost algorithm Due to R.Schapire & Y.Freund [1], is one of the most useful and successful algorithms in Machine Learning at large. We note that AdaBoost doesn't have to know in advance the parameter γ of the weak learners. Pseudo code for the AdaBoost algorithm is given in 1.

Algorithm 1 AdaBoost

Input: $\mathcal{H}, \epsilon, \delta$, γ-weak-learner algorithm: WL, sample $S_m \sim_D$.
Set $\mathbf{p}_1 \in \delta_m$ be the uniform distribution over S.
for $t = 1, 2 \dots T$ **do**
Find hypothesis $h_t \leftarrow h_S^{\mathrm{WL}}(\mathbf{p}_t, \frac{\delta}{T})$
Calculate $\epsilon_t = \mathrm{err}_{S,\mathbf{p}_t}(h_t) := \sum_{i=1}^m \mathbf{p}_t \ell_{0,1}(h_t(\mathbf{x}_i), y_i)$,
Set $\alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t})$
Update,

$$\mathbf{p}_{t+1}(i) = \frac{\mathbf{p}_t(i)e^{-\alpha_t y_i h_t(i)}}{\sum_{j=1}^m \mathbf{p}_t(j)e^{-\alpha_t y_j h_t(j)}}$$

end for
Return: $\bar{h}(x) = \mathrm{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

We next show that the training error of adaboost decreases exponentially fast with number of iterations:

Theorem 6.2. *Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis with $\epsilon_t < 1/2 - \gamma$. Then the training error of the output hypothesis of AdaBoost is at most*

$$err_S(h_S^A) \leq \exp(-2\gamma^2 T).$$

Proof. For each t denote $f_t = \sum_{i=1}^t \alpha_i h_i$. The output of AdaBoost is f_T, then. Denote:

$$Z_t = \frac{1}{m}\sum_{i=1}^m e^{-y_i f_t(\mathbf{x}_i)}$$

For any target function f we have that $\ell_{0,1}(\mathrm{sgn}(f(\mathbf{x})), y) \leq e^{-yf(\mathbf{x})}$, therefore

$$\mathrm{err}_S(\mathrm{sign}(f_T)) \leq Z_T$$

We will then show that $Z_T \leq e^{-2\gamma^2 T}$. Since $Z_0 = 1$ we can write

$$Z_T = \frac{Z_T}{Z_0} = \prod_{t=0}^{t-1} \frac{Z_{t+1}}{Z_t}$$

Therefore it suffices to show that for every t we have $\frac{Z_{t+1}}{Z_t} \leq e^{-2\gamma^2}$. Using a simple inductive argument, one can show that:

$$\mathbf{p}_t(i) = \frac{e^{-y_i f_t(\mathbf{x}_i)}}{\sum e^{-y_j f_t(\mathbf{x}_j)}}$$

Next,

$$\frac{Z_{t+1}}{Z_t} = \frac{\sum e^{-y_i f_{t+1}(\mathbf{x}_i)}}{\sum e^{-y_j f_t(\mathbf{x}_j)}} = \frac{\sum e^{-y_i f_t(\mathbf{x}_i) - y_i \alpha_{t+1} h_{t+1}(\mathbf{x}_i)}}{\sum e^{-y_j f_t(\mathbf{x}_j)}} = \sum \mathbf{p}_t(i) e^{-y_i \alpha_{t+1} h_{t+1}(\mathbf{x}_i)} =$$

$$e^{\alpha_{t+1}} \cdot \left(\sum_{y_i \neq h_{t+1}(\mathbf{x}_i)} \mathbf{p}_t(i) \right) + e^{-\alpha_{t+1}} \cdot \left(\sum_{y_i = h_{t+1}(\mathbf{x}_i)} \mathbf{p}_t(i) \right) =$$

$$\sqrt{\frac{1}{\epsilon_{t+1}} - 1} \cdot \epsilon_{t+1} + \frac{1}{\sqrt{\frac{1}{\epsilon_{t+1}} - 1}} \epsilon_{t+1} = 2\sqrt{\epsilon_{t+1}(1 - \epsilon_{t+1})}.$$

By assumption, w.p $1-\frac{\delta}{T}$: $\epsilon_{t+1} \leq 1/2-\gamma$, using monotinicity of the function $x(1-x)$ in the interval $[0,1/2]$ one can show that

$$2\sqrt{\epsilon_{t+1}(1-\epsilon_{t+1})} \leq 2\sqrt{(1/2-\gamma)(1/2+\gamma)} \leq \sqrt{1-4\gamma^2}$$

Finally, using the inequality $1-a \leq e^{-a}$ we have that $\sqrt{1-4\gamma^2} \leq e^{-(4\gamma^2)/2} = e^{-2\gamma^2}$. Taking union bound, we have that with probability $1-\delta$ $Z_T \leq e^{-2\gamma^2 T}$. □

6.2.2 Generalization Bound

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 the hypothesis class of the Adaboost algorithm generalizes. Given a concept class $\mathcal{C}$, let us denote by $\mathcal{C}(T)$, the hypothesis:

$$\mathcal{C}(T) = \{\text{sign}(\sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})) : h_i \in \mathcal{C}, i = 1\ldots,T\}$$

Lemma 6.3. *Let $\mathcal{C}$ be a class of VC-dim$(C) \leq d$ then VC-dim$(C(T)) = \tilde{O}(T \cdot$ VC-dim$(C))$[1].*

Proof. Again, we will bound the growh function of $\mathcal{C}(T)$. We can write each element $f \in \mathcal{C}(T)$ as $f_{\mathbf{w}} \circ g_{h_{1:T}}$ were $f_{\mathbf{w}} \in \mathcal{H}_T$ where $\mathcal{H}_T$ is the class of half-spaces and $g_{h_{1:T}} \in \mathcal{G}$ where

$$\mathcal{G} = \{g_{h_{1:T}} : \ g_{h1:T}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})), \ h_i \in \mathcal{C}\}$$

Similar to the proof of Theorem 6.3 (and following Ex.2) we can show that

$$\tau_{\mathcal{C}(T)}(m) \leq \tau_{\mathcal{H}}(m) \cdot \tau_{\mathcal{G}}(m) \leq \tau_{\mathcal{H}}(m)(\tau_{\mathcal{C}})^T \leq m^{T+(T\cdot \text{VC-dim}(\mathcal{C}))}$$

The result now follows from Lemma 6.4 □

Corollary 6.4. *Let S be a sample drawn IID according to some arbitrary distribution D. Assume that at each iteration of AdaBoost, the weak learner returns a hypothesis with $\epsilon_t < 1/2-\gamma$. Then if $m = |S|$ and $m \geq \Omega(\frac{VC\text{-}dim(\mathcal{C})\log m}{\gamma\epsilon^2}\log 1/\delta)$, then Adaboost outputs a hypothesis h_S^A such that w.p $1-\delta$*

$$err(h_S^A) \leq \epsilon$$

Proof. If $T > \frac{1}{\gamma^2}\log m$ we obtain that

$$\text{err}_S(h_S^A) \leq \exp(-\gamma^2 T) < \frac{1}{m}.$$

Since there are m examples this means $\text{err}_S(h_S^A) = 0$. Since $h_S^A \in \mathcal{C}(T)$ and VC-dim$(\mathcal{C}(T)) \leq \tilde{O}(T$VC-dim$(\mathcal{C}))$, the result follows from the fundemental theorem (Thm 3.5). □

References

[1] Yoav Freund and Robert E. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of Computer and System Sciences, 55(1):119–139, August 1997.

[1] Here the $\tilde{O}$ notation, means we neglect logarithmic factors