10 Boosting

Boosting is a general machine learning paradigm that is based on improving a weak learner. It approaches the bias-complexity tradeoff (appoximation vs. estimation error) in a different way. It starts with a strong bias (large approximation error), but progressing works on expanding the class and decreasing the approximation error. Boosting also addresses complexity of learning. When a strong learner is computationally complex, it might be easier to use a weak-slightly-better-than-random efficient learner to approximate gradually good predictors for larger classes. The AdaBoost algorithm outputs a linear combination of weak learners.

Robert Schapire and Yoav Freund are the authors. Currently, they are professors at Princeton and UCSD. Teaching theoretical machine learning. They published a book on boosting.

10.1 WEAK LEARNABILITY

We start by defining the notion of a weak learner. The motivation is that computing an ERM might be very difficult, and we might be satisfied with a learner that is easy to compute and slightly better than random by a factor γ.

Definition 10.1 (γ-weak learnability)

A learning algorithm, A, is a γ-weak learner for hypothesis class $\mathcal{H}$ if there exist a function $m_{\mathcal{H}} : (0,1) \to \mathbb{N}$ such that for every $\delta \in (0,1)$ for every distribution D over $\mathcal{X}$ for every labeling function $f : \mathcal{X} \to \{0,1\}$ if the realizable assumption holds and one uses $m > m_{\mathcal{H}}$ samples when running the learning algorithm has loss $L_{D,f}(h) \leq 1/2 - \gamma$.

The difference between γ-weak learnability and PAC learnablity is in PAC learnability you want a strong learner of at most ϵ error, whereas in weak learners you want to be γ better than random chance (1/2), with the hope that weak learners are easier to acquire.

If one cannot have a weak learner inside a hypothesis class $\mathcal{H}$, then an alternative is to use a base hypothesis class B where the ERM_B is efficiently implementable and every sample labeled from $\mathcal{H}$ has error at most $1/2 - \gamma$.

- Do $\mathcal{H}$ and B need to be related somehow? Is it only about the error they make?

The remaining question is on boosting a weak learner to make it a strong learner. But first, example of using a base class B

Example 10.1 Working with an example of the 3-piece problem (+-+). $\mathcal{X} = \mathbb{R}$ and $\mathcal{H}$ is the class of 3-piece classifiers parametrized with two thetas and b.

We use a basic hypothesis class B of Decision stumps which will have $L_D(h) \leq 1/3$ for $h \in B$. We wish to show that ERM_B is a weak learner for $\mathcal{H}$ with $\gamma = 1/12$. So the error of ERM_B has error $\epsilon = 1/3$ and the random error $1/2$, so solving equation $1/3 + \gamma = 1/2 - \gamma$, $\gamma = 1/12$.

10.1.1 Efficient implementation of ERM for Decision Stumps

Now, we define decision stumps over d dimensions as a class

$$\mathcal{H}_{DS} = \{x \to sign(\theta - x_i)\dot{b} : \theta \in \mathbb{R}, i \in [d], b \in \{+1, -1\}\}$$

Now, we wish to find this class' ERM. To do so, we will define the distribution vector of a training sample S as $\mathbb{D}$. The weak learner receives S and D and outputs a weak learner with loss:

$$L_D(h) = \sum_{i=1}^{m} D_i 1_{[h(x_i) \neq y_i]}$$

If D is uniform, then the loss is empirical loss ($L_S = L_D$)

We need to minimize the decision stump algorithm, which shall be done by minimizing the objective function:

$$\min_{j \in [d]} \min_{\theta \in \mathbb{R}} (\sum_{i:y_i=1}^{m} D_i 1_{[x_{i,j} > \theta]} + \sum_{i:y_i=-1}^{m} D_i 1_{[x_{i,j} \leq \theta]})$$

This can be done in $O(dm^2)$ if tried out all combos from the training set across all d dimensions. A more efficient way is to presort all the samples, define $\Theta = \{\frac{x_{i,j} + x_{i+1,j}}{2} : i \in [m-1]\}$ This interpolates between all the data points, so that you go in sorted order and try out each interpolated value as a θ boundary, and simply take the one with minimal loss. The saving in computation is that you don't need to try all combinations, but can recursively compute the loss as a function of the previous loss (on the data point to the left/right in the sorted order) and the output of the current datapoint.

10.2 ADABOOST

AdaBoost combines weak learner to create a strong learner. It receives a training set of examples $S = (x_1, y_1, ..., x_m, y_m)$ and proceeds with boosting in rounds. At round t, booster has a distribution D^t over S, both of which are passed on to the weak learner. The weak learner then outputs a hypothesis in round t h_t whose error is defined as $\epsilon_t = \sum_{i=1}^{m} D_i^t 1_{[h_t(x_i) \neq y_i]}$ and is upper bounded by

$1/2-\gamma$. AdaBoost then assigns weights $w_t = \frac{1}{2}log(\frac{1}{\epsilon_t}-1)$ to h_t such that higher weights will correspond to those functions with a smaller error. Finally, the distribution D_i is updated for the next round by adding probability mass for examples on which the learner failed, deducting for correct ones. The output of the algorithm is a weighted sum of all the weak hypotheses.

Good AdaBoost video

Theorem 10.2 If we weak learner returns a hypothesis upper bounded by $\epsilon \leq 1/2-\gamma$ then the error of the hypothesis outputted after T steps by AdaBoost is at most $exp(-2\gamma^2 T)$

Proof. We start by looking at the output of AdaBoost at round t as f_t, defined as $f_t = \sum_{p \leq t} w_p h_p$. Define $Z_t = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f_i(x_i)}$.

- TODO: why is $1_{[h(x)\neq y]} \leq e^{-yh(x)}$. It is unclear what h(x) here stands for (I'm guessing this applies on all rounds) => theory is that this is just "randomly" introduced.

Because of this, $L_S(f_T) \leq Z_T$, the proof is to show $Z_T \leq e^{-2\gamma^2 T}$ When we split by T rounds to break down Z_t we show that every round ratio is less than $e^{-2\gamma^2}$. If this works for every round, it works when summing up T rounds. First, we set the definition of Z_t as a ratio (page 107). The rest is simple math, using the definiton of ϵ and w expressed as ϵ.

- TODO fully clarify why is monotonically increasing a sufficient condition to carry on the inequality.
- $a(1-a)$ is a parabolic function with the peak at 0.5, after which it falls down.

Each iteration of AdaBoost is $O(m)$ mostly because the error summation and distribution update.

Remark 10.2 The probability that the weak learner will fail is δ, and by the union bound makes AdaBoost probability not failure $1-\delta T$.

- TODO: why is the probability of failure zero in decision stumps?

10.3 Linear Combinations of Base Hypotheses

AdaBoost is esentially a linear combination of simpler hypotheses. The linear combination is parametrized by T, number of iterations, and a weight vector w. Increasing T one allows for more hypotheses, therefore decreases the approximation risk, but potentially increases the estimation risk. Thus, T is a parameter to control the bias-complexity tradeoff. An example is shown where decision stumps can be expressed via a single linear function with T thresholds and an alpha factor. This function has $T+1$ VC-dimension.

- TODO: why is there a subset $\mathcal{G}_{\mathcal{T}} \subset L(H_{DS1}, T)$.

10.3.1 The VC-Dimension of $L(B, T)$

VC-dimension of $L(B, T)$ is upper bounded by O(VCdim(B)T).

Lemma 10.3 Let B be a base class and let L(B,T) be an AdaBoost-like class. Assume that T and VCDim(B) are at least 3. Then

$$VCdim(L(B, T)) \leq T(VCdim(B) + 1)(3log(T(VCdim(B) + 1)) + 2)$$

Proof. Set $d = VCdim(B)$. Let C be the set of size m shattered by $L(B, T)$. By Sauer's lemma there are at most $(em/d)^d$ different labelings induced from B over C. So, we need to choose T hypotheses out of $(em/d)^d$ and there are $(em/d)^{dT}$ ways to do so.