22. AdaBoost

Thursday, February 29, 2024 3:16 PM

§10.2 in [SS] or §7 in Mohri → a bit more complete, less vague, and see nice Fig 7.2

AdaBoost (= Adaptive Boosting) Binary classification, $Y = \{\pm 1\}$ ← vague in [SS]

Given dataset $S = ((x_1, y_1), \ldots, (x_m, y_m))$

and a γ-weak-learner algorithm, proceeds in rounds t

Initialize $D_i^{(0)} = \frac{1}{m}$ // in general, $D^{(t)} \in \Delta$ = m-dim. probability simplex $= \{ D \in \mathbb{R}_+^m : \sum D_i = 1 \}$

Iterate $t = 0, 1, \ldots, T$

- The weak learner solves $\underbrace{\text{weighted ERM}}_{\text{using } D_i^{(t)}}$ and returns h_t

  We (for now) treat the "learning problem" to be finding h w/ low "true risk", $L_{D^{(t)}}(h) := \sum_{i=1}^{m} D_i^{(t)} \cdot \mathbb{1}_{h(x_i) \neq y_i}$

  → conceptual leap: $D^{(t)}$ is empirical and discrete but treat it as any other underlying distribution

  our weak learner either explicitly solves weighted ERM* (w/ guarantee), or is more generic (i.e. bootstrap resample S according to $D^{(t)}$ possibly ?)

  * in fact, this is directly solving true risk (over D^t) exactly

- Weak learning guarantee is that

  $$\varepsilon_t := L_{D^{(t)}}(h_t) \leq \frac{1}{2} - \gamma$$

  with probability at least $1-\delta$

  Save h_t and give it a weight $\omega_t = \frac{1}{2} \log\left(\frac{1}{\varepsilon_t} - 1\right)$ (α_t in Mohri)

  ($\omega_t > 0$ if $\varepsilon_t < \frac{1}{2}$) so lower error $\varepsilon_t \Rightarrow$ higher weight

- adjust $D^{(t+1)}$ to give more weight to those samples we misclassified: in fact, give equal mass to sets of correct and incorrectly identified pts

  $$\tilde{D}_i^{(t+1)} = D_i^{(t)} \cdot \exp(-\omega_t \overbrace{y_i h_t(x_i)}^{+1 \text{ if we got it right, } -1 \text{ if we got it wrong}})$$

  $$D_i^{(t+1)} = \frac{\tilde{D}_i^{(t+1)}}{\underbrace{\sum_{j=1}^{m} \tilde{D}_j^{(t+1)}}_{\text{call this normalization } Z^t}} \quad \text{so that } D^{(t+1)} \in \Delta \quad \text{i.e. is a probability}$$

22a. AdaBoost

Thursday, February 29, 2024 6:03 PM

After iterating, return $h(x) = \mathrm{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right)$ as strong learner
i.e. weighted majority vote

We need to assume realizability for weak learners (i.e. wrt $\mathcal{H}$, not $\mathcal{B}$). We don't need to know the edge γ to run the algorithm (only for analysis / guarantees)

RECALL:
$\varepsilon_t = L_{D^{(t)}}(h_t) \le \frac{1}{2} - \gamma$
$w_t = \frac{1}{2}\log\left(\frac{1}{\varepsilon_t} - 1\right) = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

Before analysis, some lemmas

- **Lemma** $Z^{(t)} = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$

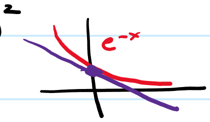
proof: $Z^{(t)} := \sum_{i=1}^{m} D_i^{(t)} \exp\Big(-\overbrace{\tfrac{1}{2}\log\left(\tfrac{1-\varepsilon_t}{\varepsilon_t}\right)}^{w_t} \overbrace{y_i h_t(x_i)}^{\in \pm 1} \Big)$ +1 if correct, −1 if incorrect

$$= \sum_{i=1}^{m} D_i^{(t)} \left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)^{-\frac{1}{2}\cdot \pm 1}$$

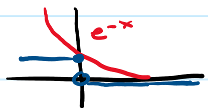
$$= \Big(\underbrace{\sum_{i \text{ correct}} D_i^{(t)}}_{:=1-\varepsilon}\Big)\cdot\sqrt{\frac{\varepsilon}{1-\varepsilon}} + \Big(\underbrace{\sum_{i \text{ incorrect}} D_i^{(t)}}_{=\varepsilon}\Big)\cdot\sqrt{\frac{1-\varepsilon}{\varepsilon}}$$

$$= 2\sqrt{\varepsilon_t(1-\varepsilon_t)} \quad \square$$

- **Fact** $4\varepsilon(1-\varepsilon) = 1 - 4\left(\frac{1}{2}-\varepsilon\right)^2$
- **Recall** $1 - x \le e^{-x}$



- **Fact** $\mathbb{1}_{x \le 0} \le e^{-x}$



Thm (10.2 [SS] or 7.2 Mohri) "Adaboost solves ERM"

The training error decays exponentially fast: w.p. $\ge 1 - \delta T$, after T rounds, $h_S = \mathrm{sign}\left(\sum_{t=1}^{T} w_t h_t\right)$, $\hat{L}_S(h) := \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{h(x_i)\neq y_i}$, then $\hat{L}_S(h_S) \le e^{-2\gamma^2 T}$ (using a γ-weak learner)

proof

Let $f_T := \sum_{t=1}^{T} w_t h_t$ so $h_S = \mathrm{sign} \circ f_T$. Observe:

$$(*) \quad \hat{L}_S(h_S) := \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{y_i \cdot f_T(x_i) \le 0} \le \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f_T(x_i)}$$ via

22b. AdaBoost

Friday, March 1, 2024 10:10 AM

(proof continued)

Also observe $D_i^{(T+1)} = \frac{D_i^{(T)}}{Z^{(T)}} \cdot e^{-\omega_T y_i h_T(x_i)}$

apply recursively...

$$= \frac{D_i^{(T)} D_i^{(T-1)}}{Z^{(T)} Z^{(T-1)}} \cdot \exp\left(-\omega_T y_i h_T(x_i) - \omega_{T-1} y_i h_{T-1}(x_i) \right)$$

recall $D_i^{(0)} = \frac{1}{m}$

$$= \frac{1}{m} \frac{e^{-y_i \sum_{t=1}^{T} \omega_t h_t(x_i)}}{\prod_{t=1}^{T} Z^{(t)}} = \frac{e^{-y_i f_T(x_i)}}{m \prod_{t=1}^{T} Z^{(t)}}$$

i.e. $e^{-y_i f_T(x_i)} = D_i^{(T+1)} \cdot m \cdot \prod_{t=1}^{T} Z^{(t)}$, plug into (*)

$$(*) \ldots \leq \frac{1}{m} \sum_{i=1}^{m} e^{-y_i f_T(x_i)} = \cancel{\frac{1}{m}} \underbrace{\sum_i D_i^{(T+1)}}_{=1 \text{ since a probability}} \cancel{m} \prod_t Z^{(t)}$$

$$= \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

now simplify...

$$= \prod_{t=1}^{T} \sqrt{1 - 4\left(\tfrac{1}{2} - \varepsilon_t\right)^2}$$

$$\leq \prod_{t=1}^{T} \sqrt{e^{-4(1/2 - \varepsilon_t)^2}}$$

$1 - x \leq e^{-x}$

$$= \exp\Big(-2 \sum_{t=1}^{T} \underbrace{\left(\tfrac{1}{2} - \varepsilon_t\right)^2}_{\leq \gamma} \Big)$$

$\varepsilon_t \leq \frac{1}{2} - \gamma$

$$\leq e^{-2T\gamma^2} \quad \square$$

22c. AdaBoost

Friday, March 1, 2024 10:23 AM

Generalization error of AdaBoost

If each weak learner is $h_t \in \mathcal{B}$, then our final output h_S is in the space

[SS] uses different notation

$$\mathcal{H}_{\mathcal{B},T} = \{ x \mapsto \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : w \in \mathbb{R}^T, h_t \in \mathcal{B} \}$$

Bounds:

good HW problems

(from Mohri) $\text{VCdim}(\mathcal{H}_{\mathcal{B},T}) \leq 2(d+1) \cdot T \cdot \log_2((T+1) \cdot e)$

(from [SS]) " " $\leq (d+1) \cdot T \cdot (3 \ln(T(d+1)) + 2)$ $T, d \geq 3$

$$\approx O(d \, T \log(T))$$

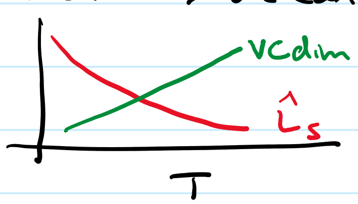
where $d = \text{VCdim}(\mathcal{B})$

For some classes (e.g. $\mathcal{B} = \mathcal{L}_D$, in which case shouldn't depend on T since $\mathcal{L}_D$ is closed under lin. comb.)

this isn't tight, but $\exists$ $\mathcal{B}$ s.t. this is a nearly tight bound.

i.e. $\text{VCdim}(\mathcal{H}_{\mathcal{B},T}) \geq \Omega(d \, T)$

So, finite VCdim $\Rightarrow$ we can generalize



Tune T to find location in bias-variance tradeoff you want to be at

Misc

- Sometimes generalization still improves as $T \to \infty$, even after $\hat{L}_S = 0$
  See Mohri for margin based analysis
  (and coordinate descent interpretation, game theory interpretation, + connections to regularization)
- Doesn't work well w/ noise (i.e. $\nexists$ oracle labeling)
- Another way to tune bias-variance is early stopping of SGD