21. Boosting

Tuesday, February 27, 2024 2:48 PM

(Throughout, we're assuming binary classification and 0-1 loss)

An example of an ensemble method

① We know there's a tradeoff (bias v. variance) based on the complexity of $\mathcal{H}$, but it's not always clear how to change the complexity

② Finding the ERM for most $\mathcal{H}$ is difficult

Boosting helps w/ both the above issues

- pick a small, simple class $\mathcal{H}$ (or $\mathcal{B}$ for "base" class) for which we can solve ERM. It should be "good enough" of a learner (aka a "weak learner")
- we'll boost $\mathcal{B}$ (solving $ERM_{\mathcal{B}}$ several times) to get a "strong" learner.

History:

1990 MIT grad. student Robert Schapire, turned practical in '95 Schapire + Yoav Freund's AdaBoost

PAC-learner aka strong learner (for $\mathcal{H}$): $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$

an algo A s.t. if m is sufficiently large, $|S| = m$ has iid samples, then $L_{D,f}(A(S)) \leq \varepsilon$ with probability at least $1-\delta$

(we're in the realizable (non-agnostic) setting)

Def An algo A is a γ-weak learner if $\exists\, m_{\mathcal{H}} : (0,1) \to \mathbb{N}$ s.t. ($\gamma > 0$)
$\forall$ dist. D, $\forall f$, if S has $m \geq m_{\mathcal{H}}(\delta)$ iid samples then
$L_{D,f}(A(S)) \leq \frac{1}{2} - \gamma$ with prob. at least $1-\delta$

(γ: the "edge", like the Casino's "edge")

($L_{D,f}(A(S)) = P[\text{misclassification}]$ since 0-1 loss)

... and $\mathcal{H}$ is γ-weak-learnable if $\exists$ a γ-weak-learner for $\mathcal{H}$.

Theoretically, for binary classif., $\mathcal{H}$ is PAC learnable iff $VCdim(\mathcal{H}) < \infty$ iff $\mathcal{H}$ is γ-weak-learnable.

But a particular algo. A might be a weak (but not strong) learner.

And there may be computational issues w/ ERM (iff ERM is a PAC learner)

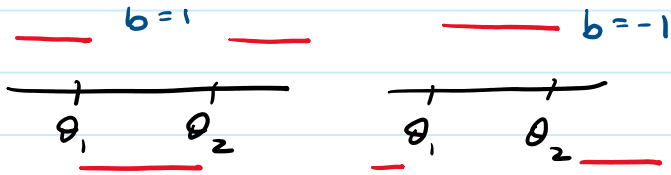
21a. Boosting (example)

Tuesday, February 27, 2024 3:05 PM

Ex. 10.1 $\mathcal{H}$ = "3-piece classifiers", algo A is "decision stump" (tiny decision trees)

Setup: $X = \mathbb{R}$, $Y = \{\pm 1\}$, $\mathcal{H} = \{ h_{\theta_1,\theta_2,b} : \theta_1, \theta_2 \in \mathbb{R}$ w/ $\theta_1 < \theta_2$, $b \in \{\pm 1\} \}$ (bit flip)

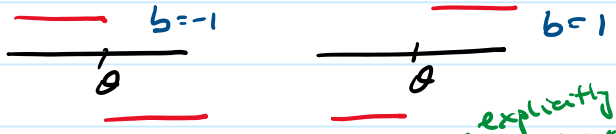
$$h_{\theta_1,\theta_2,b}(x) = \begin{cases} +b & x \notin [\theta_1, \theta_2] \\ -b & x \in [\theta_1, \theta_2] \end{cases}$$



VCdim($\mathcal{H}$) = 3, so it's PAC learnable

Let $\mathcal{B} = \{ x \mapsto \text{sign}(x-\theta)\cdot b, \ \theta \in \mathbb{R}, b \in \{\pm 1\} \}$

↑ decision stumps aka depth-1 decision trees (A common choice in boosting)

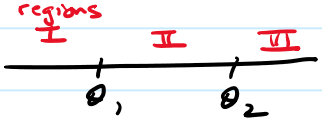


$\mathcal{B}$ not as expressive as $\mathcal{H}$, and VCdim($\mathcal{B}$) = 2 (easy to show) — explicitly or bound as union of left/right thresholding

Claim: $A = ERM_{\mathcal{B}}$ is a $\gamma = 1/2$ weak learner for $\mathcal{H}$ (any $\gamma < 1/6$ works in fact)

proof Realizability: arbitrary distr. D over x, and $\exists f \in \mathcal{H}$ s.t. $y = f(x)$.

We claim $\exists h \in \mathcal{B}$ s.t. $L_{D,f}(h) \le 1/3$.



"true" θ_1, θ_2

$1 = D(\mathbb{R}) = D(\text{I} \cup \text{II} \cup \text{III})$

$\Rightarrow \exists$ some region (I, II or III) s.t. $D(\text{region}) \le 1/3$

Then ignore this region!

eg. $D(\text{I}) \le 1/3$, so set $\theta = \theta_2$, get 100% correct on regions II & III.

Next, since VCdim($\mathcal{B}$) = 2 < ∞, $\mathcal{B}$ is PAC learnable (AGNOSTIC VERSIONS! subtle!)

so apply (quantitative) Fund. Thm. of ML (Thm 6.8) to $\mathcal{B}$,

if $m \ge m_{\mathcal{B}}(\varepsilon, \delta)$, $m_{\mathcal{B}}(\varepsilon,\delta) \le \text{const} \cdot \frac{2 + \log(1/\delta)}{\varepsilon^2}$ then

$ERM_{\mathcal{B}}$ returns $h \in \mathcal{B}$ s.t. $L_{D,f}(h) \le \min_{h' \in \mathcal{B}} L_{D,f}(h') + \varepsilon$

$\le 1/3 + \varepsilon$

Choose $\varepsilon = 1/2$

so loss $\le 5/12 = \frac{1}{2} - \frac{1}{12}$ ✓

21b. Boosting (example)

Tuesday, February 27, 2024 3:22 PM

(example continued)

Claim: $ERM_{\mathcal{B}}$ is easy to implement

Let $\mathcal{H}_{ht} = \{ x \mapsto \text{sign}(x-\theta) : \theta \in \mathbb{R} \}$ hard-thresholding (aka 1D hyperplanes!)

$\mathcal{H}_{-ht} = \{ x \mapsto -\text{sign}(x-\theta) : \theta \in \mathbb{R} \}$

$\mathcal{B} = \mathcal{H}_{ht} \cup \mathcal{H}_{-ht}$ so 1) solve $h_1 \in ERM_{ht}$

$\min_{x \in A \cup B} f(x) = \min\left(\min_{x \in A} f(x), \min_{x \in B} f(x) \right)$

2) solve $h_2 \in ERM_{-ht}$

3) choose whichever is better, h_1 or h_2

So $\min_{h \in \mathcal{H}_{ht}} \hat{L}_S(h) := \sum_{i=1}^m \frac{1}{m} \ell(h, (x_i, y_i))$ $= \begin{cases} 0 & \text{if } h(x)=y \\ 1 & \text{else} \end{cases}$

or, since it'll be useful later, consider a slight generalization

$\min_{h \in \mathcal{H}_{ht}} \sum_{i=1}^m D_i \, \ell(h, (x_i, y_i))$ $:= \hat{L}_D(h)$ for non-neg. weights D_i eg $D_i = \frac{1}{m}$

don't confuse with $\mathcal{D}$

$\vec{D} = (D_i)_{i=1}^m$

while we're at it, let's also generalize to $x \in \mathbb{R}^d$, not just $d=1$

$\mathcal{H}_{ht} = \{ x \in \mathbb{R}^d \mapsto \text{sign}(\theta - x^{(j)}), \ \theta \in \mathbb{R}, \ j \in [d] \}$ (j^{th} coordinate)

$\mathcal{H}_{ds} = \mathcal{H}_{ht} \cup \mathcal{H}_{-ht}$ ds = decision tree

need to find $\theta \in \mathbb{R}$ and $j \in [d]$

$$\min_{j \in [d]} \ \min_{\theta \in \mathbb{R}} \ \sum_{i=1}^m D_i \cdot \mathbb{1}_{h_{j,\theta}(x_i) \neq y_i}$$

loop over j

$x_{ij} = j^{th}$ coordinate of (i^{th} example $\vec{x}_i$)

$$\sum_{i: y_i = +1} D_i \cdot \mathbb{1}_{x_{ij} > \theta} + \sum_{i: y_i = -1} D_i \cdot \mathbb{1}_{x_{ij} \leq \theta}$$

Fixing $j \in [d]$: back to 1D problem with $\{x_{i,j}\}$. Let's sort them,

$\tilde{x}_1 \leq \dots \leq \tilde{x}_m$ (all in 1D now)

Objective is piecewise constant in θ, look for "break points" / "turning pts"

eg. restrict search to $\theta \in \oplus := \{ \tilde{x}_1 - 1, \ \frac{1}{2}(\tilde{x}_1 + \tilde{x}_2), \ \frac{1}{2}(\tilde{x}_2 + \tilde{x}_3), \dots \frac{1}{2}(\tilde{x}_{m-1} + \tilde{x}_m), \ \tilde{x}_m + 1 \}$

$m+1$ things to try, $O(m)$ cost per try, but can reduce that if clever (cumulative sum), so total $O(m)$ cost plus cost of sort

— so altogether $O(d \cdot m \cdot \log(m))$ complexity

21c. Boosting / sorting / shuffling / bagging

Tuesday, February 27, 2024 3:42 PM

Still on decision stump example:

$ERM_{\mathcal{B}}$ costs $O(dm\log(cm))$ flops

How much would $ERM_{\mathcal{H}}$ cost?

Can still do similar tricks w/ bitflips b and coordinate j

But now $\theta \in \mathbb{R}^2$ not $\mathbb{R}^1$, much trickier. It's still

piecewise constant in regions but not that helpful

(analogy: in 1D rootfinding, we can do the bisection method

w/ $\log(1/\varepsilon)$ steps... in 2D, no equivalent "ellipsoid method" is closest equiv.)

At best it'd be $O(m^2)$,

so $ERM_{\mathcal{H}}$ at least $\Omega(dm^2)$ cost.

Aside on complexity of sorting, shuffling, etc.

Given a list of d elements:

- Sorting is $O(d\log d)$ for generic comparison sorts,
  or $O(d)$ for radix sorts (eg. integers) but complicated due
- finding max or min is $O(d)$
- finding top-k is $O(\log(k)\,d)$, maybe surprising
  (naively, it's $\min(k, \log(d))\cdot d$ — via max / via sorting)
- finding median is $O(d)$, surprising! (naively it's $\log(d)\cdot d$ — via sorting)
  (but complicated... often better to use a well-implemented sort in practice)
- Shuffling data (applying random permutation) is $O(d)$, also surprising
  (naively it's done via sorting in $O(d\log d)$)
  use Fisher-Yates / Knuth shuffle

Aside on other well-known ensemble methods

Bootstrap resampling (Bradley Efron '79), extends jackknife (like GCV...), a bit like cross-validation

Resample as much as you want from a fixed sample (with replacement) ... to estimate errors, confidence intervals

Refs: Larry Wasserman's "All of Statistics", or work of Victor Chernozhukov (econ at MIT)

Bagging = Bootstrap Aggregating reduces variance, helps w/ overfitting, often used on decision trees.

Resample dataset S into k new datasets (via bootstrap), train k learners, combine learners via averaging (regression) or voting (classification)