Last time: • confidence boosting (easy)
• start accuracy boosting:
- setup, boosting framework
- start proof of concept: 3-stage boosting a 40% error learner to a 35.2% error learner (three stage process)

Today: • finish
• discuss boosting-by-filtering vs boosting over a fixed sample
• start discussion + analysis of AdaBoost (✓)

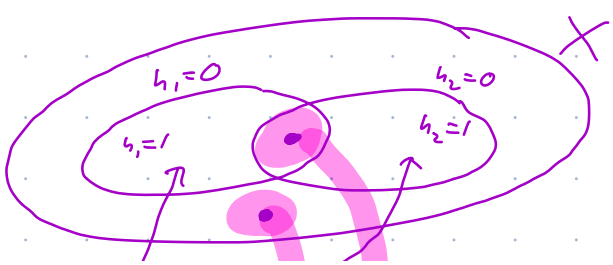
Questions?

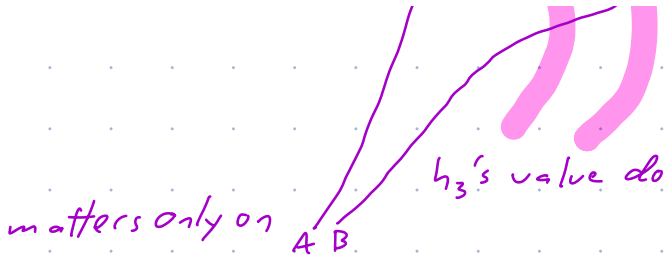
(additive) CB: toss coin w/ $\Pr[\$ = H] = p$
$m = \frac{O(\log(1/\delta))}{\tau^2}$ times:

observed freq. of H is $\pm\tau$ close to p

with prob. $\geq 1-\delta$.

At this point, have h_1 & h_2.
Need to construct D_3, & then our final h will be $h = MAJ(h_1, h_2, h_3)$.





So ϑ_3 should put all its wt. on regions A + B.

We take ϑ_3 to be ϑ with 0 wt on x s.t. $h_1(x)=h_2(x)$ + wt on $A \cup B$ "scaled up" so total weight = 1.

Run A on $EX(c, \vartheta_3)$ to get h_3;

$h = MAJ(h_1, h_2, h_3)$.

How to sample?

Draw x from $EX(c, \vartheta)$; eval. $h_1(x)$, $h_2(x)$
Keep x as (draw from $EX(c, \vartheta_3)$) iff $h_1(x) \neq h_2(x)$,
else discard + try again.

Let $\eta := \vartheta[h_1(x) \neq h_2(x)]$.

Takes, on avg, $\frac{1}{\eta}$ repetit. to get x.

If η tiny: sad (inefficient!)...

but good news is, h_3 only affects h on η frac. of ϑ. So

if n tiny (as witnessed by many rep's of draw from $EX(c,\vartheta)$ without getting $h_1 \neq h_2$), just take $h_3 \equiv$ anything., $h' = MAJ(h_1, h_2, h_3)$

"factor-1000 overhead"

So can efficiently construct h' s.t. $|err_\vartheta[h', c] - err_\vartheta[h, c]| \leq .001$

We'll ignore this & pretend have bona fide h_1, h_2, h_3 from above procedure.

To show: $\vartheta[h(x) \neq c(x)] = 35.2\%$

Divide X into 4 regions: $X = R_1 \cup \ldots \cup R_4$

R_1: x s.t. $h_1(x) = h_2(x) = c(x)$	R_2: $h_1(x) = c(x)$, $h_2(x) \neq c(x)$	R_3: $h_1(x) = h_2(x) \neq c(x)$	R_4: $h_1(x) \neq c(x)$, $h_2(x) = c(x)$
$\vartheta_2(R_1) = .5 - \rho$	$\vartheta_2(R_2) = \rho$	$\vartheta_2(R_3) = .4 - \rho$	$\vartheta_2(R_4) = .1 + \rho$
$\vartheta_1(R_1) = \frac{6}{5}(.5 - \rho)$	$\vartheta_1(R_2) = \frac{6}{5}\rho$	$\vartheta_1(R_3) = \frac{4}{5}(.4 - \rho)$	$\vartheta_1(R_4) = \frac{4}{5}(.1 + \rho)$

Let's define $\rho := \vartheta_2[R_2]$. We set things up s.t. $\vartheta_2[h_2(x) \neq c(x)] = .4$,

so means $\vartheta_2(R_3) = 0.4 - \rho$

we set things up s.t. $\vartheta_2[h_1(x) \neq c(x)] = .5$,

$\vartheta_2[h_1(x) = c(x)] = .5$, so

$\vartheta_2(R_1) = .5 - \rho$ & $\vartheta_2(R_4) = .1 + \rho$

Recalling how ϑ_2 was obtained from ϑ_1, get green values

(ϑ_3: all wt is on R_2, R_4)

h correct on $R_2 \cup R_4$ exactly when h_3 correct

h is correct : 100% on R_1
0% on R_3
60% on $R_2 \cup R_4$.

Rewrite:

h's error $= \vartheta_1(R_3) + \frac{4}{10}(\vartheta_1(R_2) + \vartheta_1(R_4))$

So $= \Pr_{x \sim \vartheta_1 = \vartheta}[h(x) \neq c(x)] =$

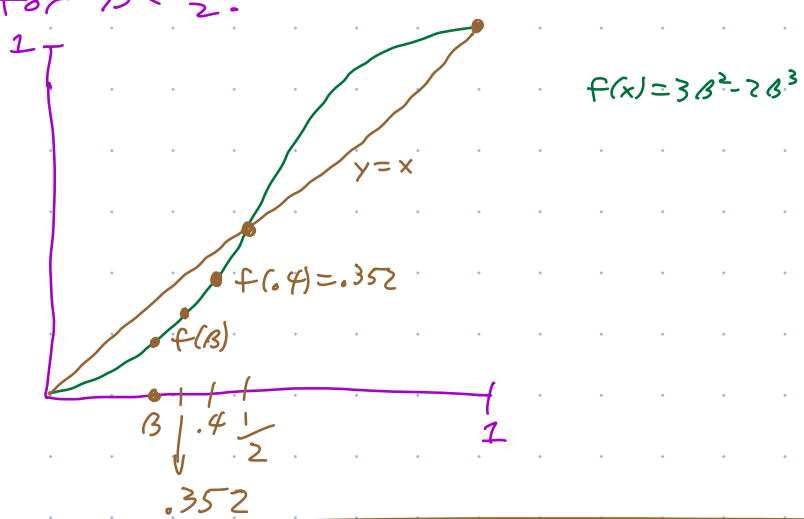
$= \frac{4}{5}(.4-\rho) + \frac{4}{10} \cdot \left(\frac{6}{5}\rho + \frac{4}{5}(.1+\rho)\right)$

$= \frac{1.6}{5} + \frac{1.6}{50} + \left(-\frac{40}{50}\rho + \frac{24}{50}\rho + \frac{16}{50}\rho\right)$

$= \frac{32}{100} + \frac{3.2}{100} + 0 = 35.2\%.$

More generally: suppose weak learner only promised to have error $\beta < \frac{1}{2}$.

Above analysis $\Rightarrow$ hyp h has error $\leq 3\beta^2 - 2\beta^3$;
$< \beta$ for $\beta < \frac{1}{2}$.



Boosting-by-Filtering (what we did) vs Boosting over a Fixed Sample.

→ Inefficient: must discard precious data to do it !!

→ In practice: $y^i = c(x^i)$

- Draw $S = (x^1, y^1), \ldots, (x^m, y^m)$

Only ever use these.

- The dist's we use: all over

- Initial dist. put wt $\frac{1}{m}$ on each data pt (ϑ).
  New dist. ϑ': new assignment of wt to points.

$$\sum_{i=1}^{m} \vartheta'(x^i) = 1$$

- weak learning guarantee: for any (say $w_i = \vartheta'(x^i)$),
  if run weak learner using $w_1, \ldots, w_m$ as wts,
  get h' s.t.

$$\sum_{i : h'(x^i) \neq c(x^i)} w_i \leq \frac{1}{2} - \gamma.$$

- Goal of booster: construct a final h s.t.

$$\sum_{i : h(x^i) \neq c(x^i)} \frac{1}{m} \leq \varepsilon.$$

(typ. go for 0 error on

data set.)

AdaBoost: simple, powerful, adaptive alg. to boost over a sample.

Idea of AdaBoost:

- given h_t, $\mathcal{D}_t$: construct $\mathcal{D}_{t+1}$ by reweighting $\mathcal{D}_t$ s.t. h_t has 50% acc. under $\mathcal{D}_{t+1}$. (scale regions ↑ ↓ like $\mathcal{D}_2$ did).

- At end: combines $h_1, h_2, \ldots$ using weighted maj. vote. Hyp's that did better: more weight. ("adaptivity")

Let's see it... :)

Next time: analyze AdaBoost.
