Computational and Statistical Learning Theory

TTIC 31120

Prof. Nati Srebro

Lecture 8:

Boosting

Compression Schemes

“Weak” vs “Strong” Learning

- Recall definition of (realizable) PAC learning of $\mathcal{H}$ using rule $A(\cdot)$:

For any $\mathcal{D}$ s.t. $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) = 0$, and **any** $\epsilon, \delta > 0$, using $m(\epsilon,\delta)$ sample,

$$\forall^{\delta}_{S\sim\mathcal{D}^{m(\epsilon,\delta)}} \;\; L_{\mathcal{D}}\big(A(S)\big) < \epsilon$$

- $A(\cdot)$ is a **weak learner** for $\mathcal{H}$ if:

There **exists** $\epsilon < {}^{1}/_{2}$, $\delta < 1$, m, s.t. for any $\mathcal{D}$ with $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) = 0$,

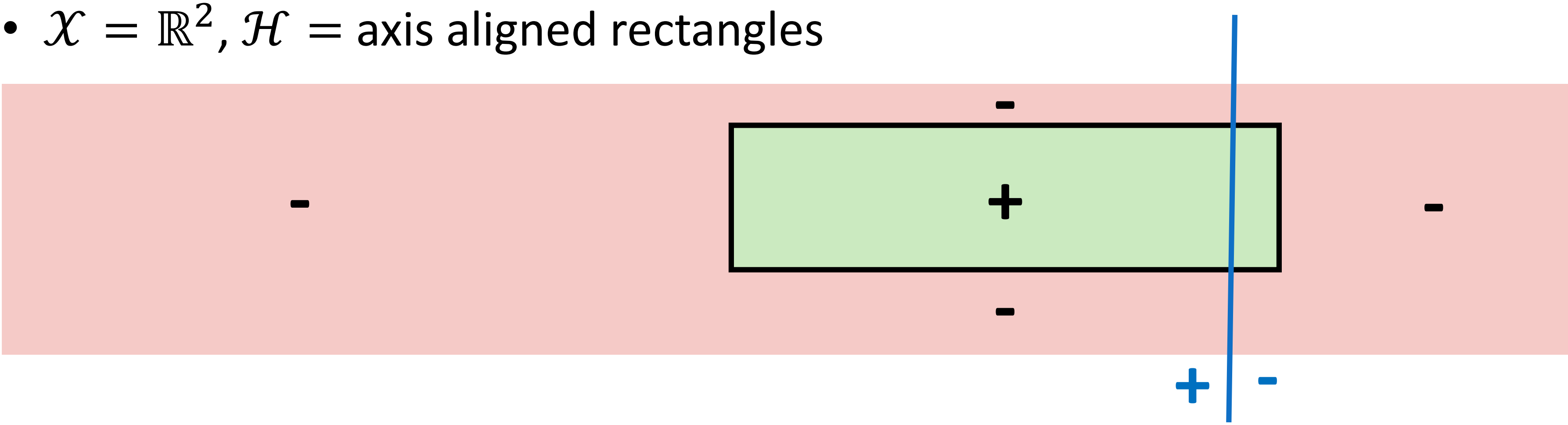
$$\forall^{\delta}_{S\sim\mathcal{D}^{m}} \;\; L_{\mathcal{D}}\big(A(S)\big) < \epsilon$$

(e.g. $\epsilon = 0.49$ and $1-\delta = 0.01$)

- If $\mathcal{H}$ is weakly learnable, is it also strongly learnable?
  - Yes: $\mathcal{H}$ is weakly learnable ➔ VCdim($\mathcal{H}$)<∞ ➔ $\mathcal{H}$ is (strongly) learnable
- If we have access to an (efficient) weak learner $A(\cdot)$, can we use it to build an (efficient) strong learner?

Example: Weak Learning with a Weak Class

- $\mathcal{X} = \mathbb{R}^2$, $\mathcal{H}$ = axis aligned rectangles



- Decision stumps: $\mathcal{B} = \left\{ \left[\left[\, s \cdot x[i] < \theta \right]\right] \mid i = 1{,}2,\; s = \pm 1,\; \theta \in \mathbb{R} \right\}$
- Claim: For any $\mathcal{D}$, if $\exists_{h_\blacksquare \in \mathcal{H}} L_{\mathcal{D}}(h_\blacksquare) = 0$ ➔ $\exists_{h \in \mathcal{B}} L_{\mathcal{D}}(h) \leq \frac{3}{7} < 0.429$
- Since VCdim($\mathcal{B}$)=3, with $m = m_{VC}(D = 3, \epsilon = 0.001, \delta = 0.9)$:

$$\text{w.p.} \geq 0.1 \text{ over } S \sim \mathcal{D}^m: \quad L_{\mathcal{D}}\big(ERM_{\mathcal{B}}(S)\big) < 0.43$$

- Conclusion:
  $ERM_{\mathcal{B}}(\cdot)$ is a weak learner for $\mathcal{H}$ with $\epsilon = 0.43 < 0.5$ and $\delta = 0.9 < 1$

The Boosting Problem

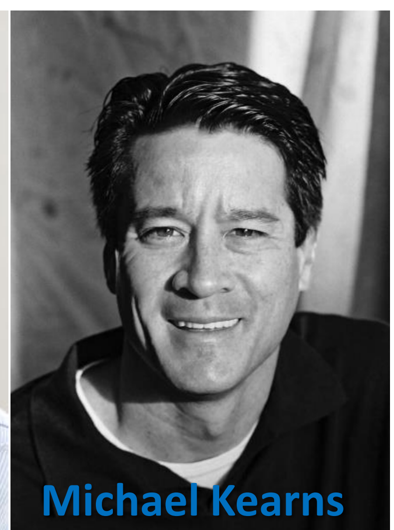
- Boosting the Confidence:
  If the learning algorithm works only with some very small fixed probability $1 - \delta_0$ (e.g. $1 - \delta_0 = 0.01$), can we construct a new algorithm that works with arbitrarily high probability $1 - \delta$ (for any $\delta > 0$) ?

- Boosting the error:
  If the learning algorithm only returns a predictor that is guaranteed to be slightly better then chance, i.e. has error $\epsilon_0 = \frac{1}{2} - \gamma < \frac{1}{2}$ (for some fixed $\gamma > 0$), can we construct a new algorithm that achieves arbitrarily low error ϵ?

Boosting the Error

If a learning algorithm only returns a predictor that is guaranteed to be slightly better then chance, i.e. has error $\epsilon_0 = \frac{1}{2} - \gamma < \frac{1}{2}$ (for some $\gamma > 0$), can we construct a new algorithm that achieves arbitrarily low error ϵ?

- Posed (as a theoretical question) by Valiant and Kearns, Harvard 1988
- Solved by MIT student Robert Schapire, 1990
- AdaBoost Algorithm by Schapire and Yoav Fruend, AT&T 1995



AdaBoost

- Input: Training set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
- Weak Learner A, which will be applied to *distributions D over S*
  - If thinking of $A(S')$ as accepting a sample S':
    each $(x, y) \in S'$ is set to (x_i, y_i) w.p. D_i (independently and with replacements)
  - Can often think of A as operating on a weighted sample, with weights D_i
- Output: hypothesis h

Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = A(D^{(t)})$$

$$\epsilon_t = L_{D^{(t)}}(h_t) = \frac{1}{m}\sum_i D_i^{(t)} \cdot [[h_t(x_i) \neq y_i]]$$

$$\alpha_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$$

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{\sum_j D_i^{(t)} \exp\left(-\alpha_t y_j h_t(x_j)\right)}$$

Output: $\overline{h}_T(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

AdaBoost: Weight Update

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t} = \frac{1}{Z_t} \cdot \begin{cases} D_i^{(t)} \cdot \sqrt{\dfrac{1-\epsilon_t}{\epsilon_t}} & \text{if } h_t(x_i) \neq y_i \\ D_i^{(t)} \cdot \sqrt{\dfrac{\epsilon_t}{1-\epsilon_t}} & \text{if } h_t(x_i) = y_i \end{cases}$$

- $Z_t = \sum_{h_t(x_i)\neq y_i} D_i^{(t)} \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + \sum_{h_t(x_i)=y_i} D_i^{(t)} \cdot \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}$

  $= \epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (1-\epsilon_t) \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = \mathbf{2\sqrt{\epsilon_t(1-\epsilon_t)}}$

AdaBoost: Weight Update

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t} = \begin{cases} \dfrac{D_i^{(t)}}{2\epsilon_t} & \text{if } h_t(x_i) \neq y_i \\ \dfrac{D_i^{(t)}}{2(1-\epsilon_t)} & \text{if } h_t(x_i) = y_i \end{cases}$$

- $Z_t = \sum_{h_t(x_i)\neq y_i} D_i^{(t)} \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + \sum_{h_t(x_i)=y_i} D_i^{(t)} \cdot \sqrt{\frac{\epsilon_t}{1-\epsilon_t}}$

  $= \epsilon_t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (1-\epsilon_t) \cdot \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} = \mathbf{2\sqrt{\epsilon_t(1-\epsilon_t)}}$

- $\boldsymbol{L_{D^{(t+1)}}(h_t)} = \sum_{h_t(x_i)\neq y_i} D_i^{(t+1)} = \sum_{h_t(x_i)\neq y_i} D_i^{(t)} \cdot \frac{1}{2\epsilon_t} = \epsilon_t \cdot \frac{1}{2\epsilon_t} = \mathbf{\frac{1}{2}}$

AdaBoost as Learning a Linear Classifier

- Recall: $\overline{h}_T(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
- Let $\mathcal{B} = \{\, all\ hypothesis\ outputed\ by\ A \,\}$
  - "Base Class", e.g. decision stumps

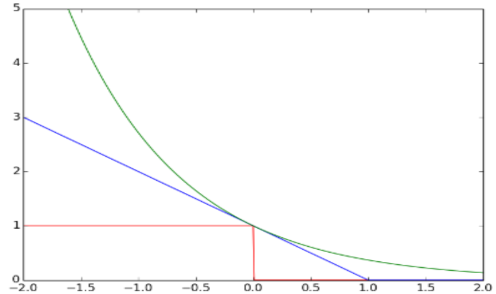
$\phi(x)[h] = h(x)$

$$\underbrace{\overline{h}_T \in \{\, h_w(x) = sign(\langle w, \phi(x) \rangle) \mid w \in \mathbb{R}^{\mathcal{B}} \,\}}_{\text{Class of halfspaces } \mathcal{L}(\mathcal{B})}$$

$w[h] = \sum_{h_t = h} \alpha_t$

$$L_S^{\exp}(w) = \frac{1}{m} \sum \ell^{\exp}(h_w(x_i); y_i) \qquad \ell^{\exp}(z, y) = e^{-yz}$$

- Each step of AdaBoost: Coordinate descent on $L_S^{\exp}(w)$
  - Choose coordinate h of $\phi(x)$ s.t. $\frac{\partial}{\partial w[h]} L_S^{\exp}(\mathrm{w})$ is high
  - Update $w[h] = \arg\min L_S^{\exp}(w)$ s.t. $\forall_{h' \neq h} w[h']$ is unchanged



Coordinate Descent on $L_S^{\exp}(w)$

- $\frac{\partial}{\partial w[h]} L_S^{exp}(w) = \frac{\partial}{\partial w[h]} \frac{1}{m} \sum e^{-y_i h_w(x_i)}$

$$= \frac{1}{m} \sum e^{-y_i h_w(x_i)} \left(-y_i \frac{\partial h_w(x_i)}{\partial w[h]}\right) = \frac{1}{m} \sum e^{-y_i h_w(x_i)} (-y_i h(x_i))$$

$$= \frac{1}{m} \sum \underbrace{e^{-y_i \sum_{t=1}^{T-1} \alpha_t h_t(x_i)}}_{\prod_{t=1}^{T-1} e^{-y_i \alpha_t h_t(x_i)} \propto D_i^{(T)}} (-y_i h(x_i)) \propto \mathbf{1 - 2} L_{D^{(T)}}(\boldsymbol{h})$$

- Minimize $L_{D^{(T)}}(h)$ ➔ Maximize $\frac{\partial}{\partial w[h]} L_S^{\exp}(w)$

- Updating $w[h]$: set $w^{(t)}[h_t] = w^{(t-1)}[h_t] + \alpha$

$$\alpha = \arg\min L_S^{\exp}(w^{(t)})$$

➔ $0 = \frac{\partial}{\partial \alpha} L_S^{\exp}(w^{(t)}) = \frac{\partial}{\partial w[h_t]} L_S^{\exp}(w^{(t)}) \propto 1 - 2L_{D^{(t+1)}}(h_t)$

➔ choose α s.t. $L_{D^{(t+1)}}(h_t) = \frac{1}{2}$

AdaBoost: Minimizing $L_S(h)$

- Theorem: If $\forall_t\ \epsilon_t \le \frac{1}{2} - \gamma$, then $L_S^{01}(\overline{h}_T) \le L_S^{\exp}(\overline{h}_T) \le e^{-2\gamma^2 T}$

$D_i^{T+1} = \frac{1}{m}\prod_{t=1}^{T}\frac{e^{-y_i\alpha_t h_t(x_i)}}{z_t}$

$\sum_i D_i^{T+1} = 1$

Proof: $L_S^{\exp}(\overline{h}_T) = \frac{1}{m}\sum_i e^{-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)} = \frac{1}{m}\sum_i\left(D_i^{(T+1)} m\prod_{t=1}^{T} z_t\right) = \prod_{t=1}^{T} z_t$

$= \prod_{t-1}^{T}\left(2\sqrt{\epsilon_t(1-\epsilon_t)}\right) \le \left((1-2\gamma)(1+2\gamma)\right)^{T/2} = (1-4\gamma^2)^{T/2} \le e^{-2\gamma^2 T}$

- If $A(\cdot)$ is a weak learner with $\delta_0, \epsilon_0 = \frac{1}{2} - \gamma$, and if $L_{\mathcal{D}}(h) = 0$:
  - ➔ $L_S(h) = 0$ ➔ $\mathrm{L}_{D^{(t)}}(h) = 0$ ➔ w.p. $1-\delta$, $L_{D^{(t)}}(h) \le \frac{1}{2} - \gamma$
  - ➔ w.p. $1 - \delta T$, $L_S(h_s) \le e^{-2\gamma^2 T}$
- To get any $\epsilon > 0$, run AdaBoost for $T = \frac{\log\left(\frac{1}{\epsilon}\right)}{2\gamma^2}$ rounds
- Setting $\epsilon = \frac{1}{2m}$, after $T = \frac{\log(2m)}{2\gamma^2}$ rounds: $L_S(h_s) = 0$!
- What about $L_{\mathcal{D}}(h)$?

Sparse Linear Classifiers

- Recall: $h_s(x) = sign(\sum_{t=1}^{T} w_t h_t(x))$
- Let $\mathcal{B} = \{\, all\ hypothesis\ outputed\ by\ A \,\}$
  - "Base Class", e.g. decision stumps

$$h_T \in \underbrace{\{\, h_w(x) = sign(\langle w, \phi(x) \rangle) \mid w \in \mathbb{R}^{\mathcal{B}}, \|w\|_0 \leq T \}}_{\text{Class of **sparse** halfspaces } \mathcal{L}(\mathcal{B}, T)}$$

- We already saw: $\text{VCdim}(\mathcal{L}(\mathcal{B}, T)) \leq O(T \log|\mathcal{B}|)$
- Even if $\mathcal{B}$ is infinite (e.g. in the case of decision stumps):
  $$\text{VCdim}\big(\mathcal{L}(\mathcal{B}, T)\big) \leq \tilde{O}\big(T \cdot \text{VCdim}(\mathcal{B})\big)$$
- Sample complexity: $m = \tilde{O}\left(\frac{\log(m)}{\gamma^2} \cdot \frac{\text{VCdim}(\mathcal{B})}{\epsilon}\right) = \tilde{O}\left(\frac{\text{VCdim}(\mathcal{B})}{\gamma^2 \epsilon}\right)$
- But if weak learner is improper and $\text{VCdim}(\mathcal{B}) = \infty$?

Compression Bounds

- Focus on realizable case, and learning rules s.t. $L_S(A(S)) = 0$
- Suppose $A(S)$ only dependent on first $r < m$ examples, $A\big((x_1, y_1), \ldots, (x_m, y_m)\big) = \tilde{A}\big((x_1, y_1), \ldots, (x_r, y_r)\big)$:

$$L_{S[r+1:m]}\left(\tilde{A}(S[1:r])\right) = 0 \quad \Rightarrow \quad \forall^{\delta}_{S \sim \mathcal{D}^m}\, L_{\mathcal{D}}\big(A(S)\big) \le \frac{\log\left({}^{1}/_{\delta}\right)}{m - r}$$

- In fact, same holds for any **predetermined** $i_1, \ldots, i_r$, if $A(S)$ only depends on $\left(x_{i_1} y_{i_1}\right), \ldots, \left(x_{i_r}, y_{i_r}\right)$
- Now consider $A(S) = \tilde{A}\left(S_{I(S)}\right)$ with $I: (\mathcal{X} \times \mathcal{Y})^m \to \{1..m\}^r$. That is, can represent $A(S)$ using r training points, but need to choose which ones.
- Taking a union bound over m^r choices of indices:

$$L_{\mathcal{D}}\big(A(S)\big) \le \frac{r \log m + \log\left({}^{1}/_{\delta}\right)}{m - r}$$

Compression Schemes

- $A(S)$ is "r-compressing" if $A(S) = \tilde{A}\left(S_{I(S)}\right)$ for some $I: (\mathcal{X} \times \mathcal{Y})^m \to \{1..m\}^r$
- Axis Aligned Rectangles
  - $I(S)$ = { leftmost positive, rightmost positive, top positive, bottom positive}
  - $r = 4$
- Halfspaces in $\mathbb{R}^d$
  - A bit trickier, but can be done with $r = d + 1$ (for non-homogenous)
- $A(\cdot)$ is r-compressing and $L_S\left(A(S)\right) = 0$ ➔ for $m > 2r$, $\forall^{\delta}_{S\sim\mathcal{D}^m}$

$$L_{\mathcal{D}}\left(A(S)\right) \leq 2\,\frac{r\log m + \log\left({}^{1}/_{\delta}\right)}{m}$$

- By VC lower bound: $FINDCONS_{\mathcal{H}}$ is r-compressing ➔ $VCdim(\mathcal{H}) \leq O(r)$
- In fact: $VCdim(\mathcal{H}) \leq r$
- Conjecture: every $\mathcal{H}$ has a $VCdim(\mathcal{H})$-compressing $FINDCONS_{\mathcal{H}}$

Back to Boosting…

- $A(S)$ is an $(\epsilon_0 = \frac{1}{2} - \gamma, \delta_0)$ weak learner that uses m_0 samples.
- Boost the confidence to get a $(\frac{1}{2} - \frac{\gamma}{2}, \delta')$ learner that uses $m_1(\delta') = O\left(m_0 \cdot \frac{\log 1/\delta'}{\log 1/\delta_0} + \frac{\log 1/\delta' - \log\log 1/\delta_0}{\gamma^2}\right)$ samples
- Run AdaBoost on m samples for $T = \frac{2\log m}{\gamma^2}$ iterations, each time using $m_1\left(\frac{\delta}{T}\right)$ samples for the weak learner to get $L_S(\overline{h}_T) = 0$

$$\overline{h}_T = \sum_{t=1}^{T} \alpha_t h_t$$

$h_t = A(sample\ of\ size\ m_1)$

- $(h_1, \ldots, h_T)$ has a compression scheme with $r = T \cdot m_1$ points
- What about α_t???

Partial Compression

- Instead of r training points specifying $A(S)$ exactly, suppose they only specify a limited set of hypothesis in which $A(S)$ lies.
  - $I: (\mathcal{X} \times \mathcal{Y})^m \to \{1..m\}^r$
  - $F: (\mathcal{X} \times \mathcal{Y})^r \to$ hypothesis classes, each with $\mathrm{VCdim}\big(F(S)\big) \leq D$
  - $A(S) \in F\big(I(S)\big)$

- Theorem: If $A(S)$ has a compression scheme as above and $L_S\big(A(S)\big) = 0$, then for $m \geq 2r + D$, $\forall^{\delta}_{S\sim\mathcal{D}^m}$

$$L_{\mathcal{D}}\big(A(S)\big) \leq O\left(\frac{(D+r)\log m + \log {}^{2}\!/_{\delta}}{m}\right)$$

Proof outline: take union bound over choice of indices $I(S)$, of the VC-based uniform convergence bounds, each time using just the points outside $I(S)$.

Back to Boosting…

- $A(S)$ is an $(\epsilon_0 = \frac{1}{2} - \gamma, \delta_0)$ weak learner that uses m_0 samples.
- Boost the confidence to get a $(\frac{1}{2} - \frac{\gamma}{2}, \delta')$ learner that uses

$$m_1(\delta') = O\left(m_0 \cdot \frac{\log^1/_{\delta'}}{\log^1/_{\delta_0}} + \frac{\log^1/_{\delta'} - \log\log^1/_{\delta_0}}{\gamma^2}\right) \text{ samples}$$

- Run AdaBoost on m samples for $T = \frac{2\log m}{\gamma^2}$ iterations, each time using $m_1\left(\frac{\delta}{T}\right)$ samples for the weak learner to get $L_S(\overline{h}_T) = 0$

$$\overline{h}_T = \sum_{t=1}^{T} \alpha_t h_t \in \mathcal{L}(\{h_1, \dots, h_T\}) = F(I(S))$$

- Conclusion:

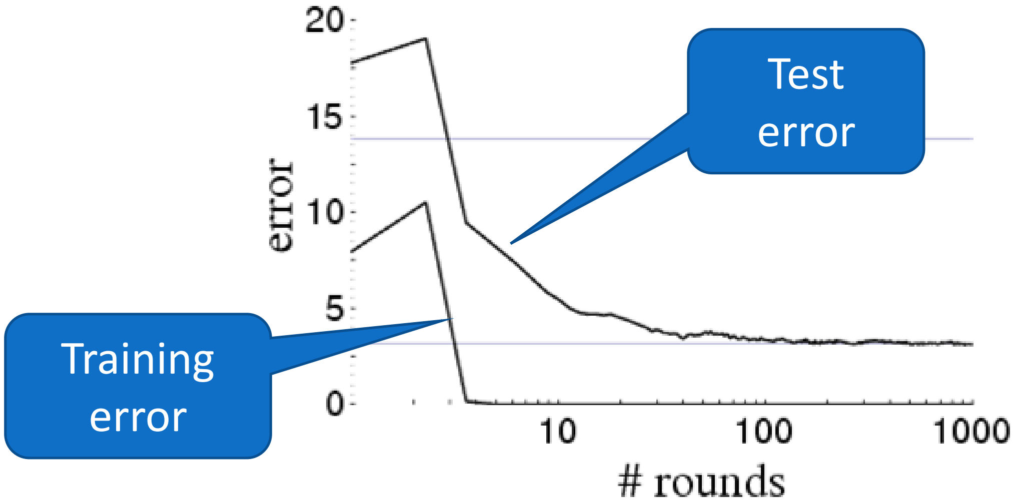
For fixed ϵ_0, δ_0

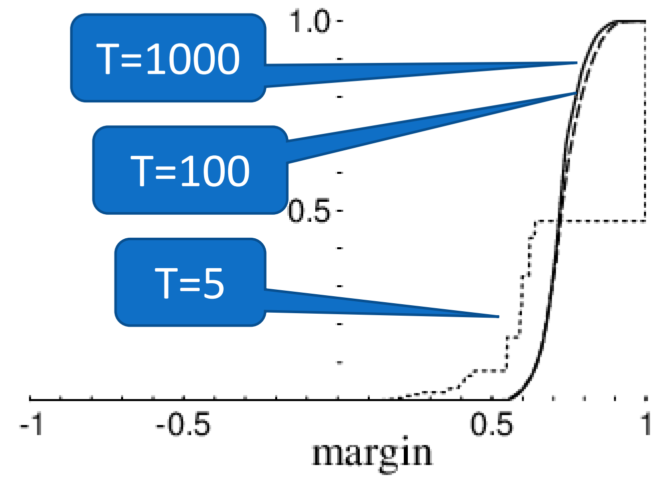
$$L_{\mathcal{D}}(\overline{h}_T) \le O\left(\frac{(T + Tm_1)\log m + \log\frac{1}{\delta}}{m}\right) = O\left(\frac{m_0 \cdot \log^2 m \cdot \log\frac{1}{\delta}}{m}\right)$$

$$\Rightarrow m(\epsilon, \delta) = O\left(\frac{m_0 \log^2 {}^1/_\epsilon \log^1/_\delta}{\epsilon} \cdot \frac{1}{\gamma^2 \log\frac{1}{\delta_0}}\right)$$

AdaBoost In Practice

- Complexity control is in terms of sparsity (#iterations) T
- Realizable case (MDL): use first T s.t. $L_S(\overline{h}_T) = 0$
- More realistically (SRM): Use validation/cross-validation to select T





- Even after $L_S(\overline{h}_T) = 0$, AdaBoost keeps improving the ℓ_1 *margin*

Interpretations of AdaBoost

- "Boosting" weak learning to get arbitrary small error
  - Theory is for realizable case
  - Shows efficient weak and strong learning equivalent
- Ensemble method for combining many simpler predictors
  - E.g. combining decision stumps or decision trees
  - Other ensemble methods: bagging, averaging, gating networks
- Method for learning using ***sparse*** linear predictors with large (infinite?) dimensional feature space
  - Sparsity controls complexity
  - Number of iterations controls sparsity
- Coordinate-wise optimization of $L_S^{\exp}(w)$
  - We'll get back to this when we talk about real-valued loss
- Learning (in high dimensions) with large ℓ_1 margin
  - Learning guarantee in terms of ℓ_1 margin
  - We'll get back to this when we talk about ℓ_1 margin

Just one more thing…

Back to Hardness of Agnostic Learning

$$\mathcal{H} = \{x \mapsto [[\langle w, x\rangle > 0]] \mid w \in \mathbb{R}^n\}$$
$$\mathcal{H}_{k(n)} = \{h_1 \wedge h_2 \wedge \cdots \wedge h_k \mid h_i \in \mathcal{H}\}$$

- Lemma: $\exists_{h \in \mathcal{H}_k} L_{\mathcal{D}}(h) = 0 \Rightarrow \exists_{h \in \mathcal{H}} L_{\mathcal{D}}(h) < \frac{1}{2} - \frac{1}{2k^2}$

$\mathcal{H}$ is efficiently agnostically learnable

$\Downarrow$

Efficient weak learner for $\mathcal{H}_{k(n)}$ with $\gamma = \frac{1}{2k^2}$

$\Downarrow$

$\mathcal{H}_{k(n)}$ is efficiently learnable (in realizable case) for, e.g. $k(n) = n$

- Conclusion: assuming $\tilde{O}(n^{1.5}) - uSVP \notin RP$, halfspaces are not efficiently agnostically learnable (even improperly)