Statistical Learning Theory Assignment 2

Zimeng Yu

October 20, 2025

1 Problem 1

Step 1 By definition, if $\mathcal{H}_n$ is efficiently agnostically properly PAC learnable, then exists efficient learning Algorithm A, satisfy: $L_D(\hat{h}) \leq \inf_{h\in\mathcal{H}_n} L_D(h)+\epsilon$. And, hypothesis ERM $\hat{h}_{ERM} = \min_{h\in\mathcal{H}_n} L_S(h) \to$ intractable. Time A is multi-polynomial time.

Step 2 Build a random algorithm R for $(S,k) \in \text{AGREEMENT}_{\mathcal{H}_n}$. let $|S| = m$. The input is (S,k), error rate is $\epsilon_S = 1 - k/m$. $\text{AGREEMENT}_{\mathcal{H}_n}$ is true, $\iff$ i.i.f. $h \in \mathcal{H}_n$ makes $L_S(h) \leq \epsilon_S$. for every $(x,y) \in S$, probability is $1/m$, ϵ', δ'.the empirical $L_S(\hat{h}) = \frac{|\{(x,y)\in S|\hat{h}(x)\neq y\}|}{|S|}$, output is, if:

$$L_S(\hat{h}) \leq \epsilon_S + \epsilon'$$

output 1; else, output 0.

Due to $\mathcal{H}_n$ is efficiently agnostically properly PAC learnable, A can finish step 2 and 3, so R is a RP Algorithm. (where $S' \neq m$, $S' = \frac{1}{\epsilon'}$)

Step 3 prove R satisfy RP:

(3.1) Situation, (S,k) is not "NO" answer
if $\text{AGREEMENT}_{\mathcal{H}_n}$ is False, then $\forall h \in \mathcal{H}_n$, $|\{(x,y) \in S \mid h(x) = y\}| < k$, that is $\forall h \in \mathcal{H}_n, L_S(h) > 1 - k/m = \epsilon_S$. so $\min_{h\in\mathcal{H}_n} L_S(h) > \epsilon_S$. And, because $\epsilon' > 0$, so $\min_{h\in\mathcal{H}_n} L_S(h) \geq \epsilon_S + \epsilon_{\min}$. where $\epsilon_{\min}$ is minimum error rate, $\epsilon_{\min} > 1/m$.
for Algorithm R, the output $\hat{h}$ satisfy $L_S(\hat{h}) > \epsilon_S$. $L_S(\hat{h})$ can even satisfy $L_S(\hat{h}) \geq \min_{h\in\mathcal{H}_n} L_S(h) \geq \epsilon_S + 1/m > \epsilon_S + \epsilon'$, so R always output 0.
The condition 2: $P(R(S,k) = 0 \mid (S,k) \notin \text{AGREEMENT}_{\mathcal{H}_n}) = 1$ satisfy.

(3.2) Situation, (S,k) is "YES" answer
if $\text{AGREEMENT}_{\mathcal{H}_n}$ is true, then exist $h^* \in \mathcal{H}_n$, let $L_S(h^*) \leq \epsilon_S$, so $\min_{h\in\mathcal{H}_n} L_S(h) \leq \epsilon_S$.
Since $\mathcal{H}_n$ is efficiently agnostically properly PAC learnable, and has limited Sample set S, we can use Hoeffding-like bound or Uniform Convergence to connect $L_S(h)$ and true $L_D(h)$. But for limited Sample set S, the Uniform distribution D_S. $L_S(h)$ equals to $L_{D_S}(h)$:

$$L_{D_S}(h) = \sum_{(x,y)\in S} D_S(x,y) \cdot \mathbb{I}(\hat{h}(x) \neq y) = \sum \frac{1}{m} \cdot \mathbb{I}(\hat{h}(x) \neq y)$$

$$= \frac{1}{m} \sum \mathbb{I}(\hat{h}(x) \neq y) = L_S(\hat{h})$$

That means: $\inf_{h\in\mathcal{H}_n} L_{D_S}(h) = \inf_{h\in\mathcal{H}_n} L_S(h) \le \epsilon_S$

Now considering A algorithm:

if $\mathcal{H}_n$ is efficiently agnostically properly PAC learnable, A is related algorithm, then A must satisfy: A receive enough samples, it will return a hypothesis approach ERM in high probability.

according EAP-PAC learnability, S' is sample, D is real distribution D_S. then A is (in probability $1-\delta'$):

$$L_{D_S}(\hat{h}) \le \inf_{h\in\mathcal{H}_n} L_{D_S}(h) + \epsilon'$$

replace $L_{D_S}(\hat{h}) = L_S(\hat{h})$:

$$L_S(\hat{h}) \le \min_{h\in\mathcal{H}_n} L_S(h) + \epsilon'$$

As the condition is $\text{AGREEMENT}_{\mathcal{H}_n}$ is true, so $\min_{h\in\mathcal{H}_n} L_S(h) \le \epsilon_S$. so, $L_S(\hat{h}) \le \epsilon_S + \epsilon'$ the inequation is in probability $1-\delta'$, and $\delta' = \frac{1}{4}$, so: $\min_{h\in\mathcal{H}_n} L_S(h) \le \epsilon_S$.

$$P(R(S,k)=1 \mid (S,k)\in \text{AGREEMENT}_{\mathcal{H}_n}) = P(L_S(\hat{h}) \le \epsilon_S + \epsilon') \ge 1-\delta' = \frac{3}{4}$$

then, the RP condition 1: $P(R(S,k)=1 \mid (S,k)\in \text{AGREEMENT}_{\mathcal{H}_n}) \ge \frac{3}{4} > \frac{1}{2}$ satisfy. In Summary: Algorithm satisfy RP's two conditions for NO answer, it always output 0, for "YES", it output 1 in high probability (at least $\frac{3}{4}$), and R run is polynomial time.

2 Problem 2

(a) show that $vc(\mathcal{H}_n^k) \le O(k\log(\binom{n}{k}))$

the $\mathcal{H}_n^k$ is a union, and hypothesis is choosing k features to use. And the number of ways to choose k features n is $\sum_{i=0}^{k}\binom{n}{i} \le \left(\frac{en}{k}\right)^k$.

Smaller class $\mathcal{H}_J$, using features J with $|J| \le k$ is a linear separator in $\mathbb{R}^{|J|}$. The VC dimension of a perceptron in $\mathbb{R}^d$ is $d+1$, $vc(\mathcal{H}_J) \le k+1$.

By Sauer's Lemma, the growth function is: $\Pi_{\mathcal{H}_J}(m) \le \left(\frac{em}{k+1}\right)^{k+1}$

The growth function of the union is bounded by Number of classes $\times$ Max growth funct(ion) of a class, which is: $\Pi_{\mathcal{H}_n^k}(m) \le \left(\frac{en}{k}\right)^k \left(\frac{em}{k+1}\right)^{k+1}$

To find VC dim: let $d = vc(\mathcal{H}_n^k)$, we must have $2^d \le \Pi_{\mathcal{H}_n^k}(d)$.

$$2^d \le \left(\frac{en}{k}\right)^k \left(\frac{ed}{k+1}\right)^{k+1}$$

taking $\log_2$ of both sides:

$$d < k\log_2\left(\frac{en}{k}\right) + (k+1)\log_2\left(\frac{ed}{k+1}\right)$$

d grows as $O(k\log(n/k))$, which is $O(k\log n)$. The requested bound $O(k\log(n^k)) = O(k^2\log n)$ is a looser bound. $O(k\log n) \le O(k^2\log n)$ is true.

(b) Provide an explicit learning rule A and show the bound. the bound is classic result for Structural Risk Minimization (SRM): learn rule A (SRM): the algorithm A finds the hypothesis $h_{w,\theta}$ that minimizer a penalized empirical risk:

$$A(S) = \arg\min_{w,\theta}\{L_S(h_{w,\theta}) + \text{Penalty}(\|w\|_0, m, \delta)\}$$

where the penalty term is taken directly from the bound we want to prove:

$$\text{penalty}(k,m,\delta) = C\sqrt{\frac{k\log n + \log n + \log(1/\delta)}{m}}$$

the $k \log n$ is from $vc(\mathcal{H}_n^k)$, and $\log n$ term is from union bound.
To prove: standard SRM analysis shows $\hat{h} = A(S)$ chosen by this rule and the true optimum h_{opt}:

$$L_D(\hat{h}) \leq L_S(\hat{h}) + \text{Penalty}(\hat{h})$$

$$L_S(\hat{h}) + \text{Penalty}(\hat{h}) \leq L_S(h_{opt}) + \text{Penalty}(h_{opt})$$

$$L_S(h_{opt}) \leq L_D(h_{opt}) + \text{Penalty}(h_{opt})$$

chaining these gives $L_D(\hat{h}) \leq L_D(h_{opt}) + 2 \cdot \text{Penalty}(h_{opt})$, which is the desired bound.
(c) Prove the bound using the validation rule. let $m_1 = m_2 = m/2$. The proof relies on two separate uniform convergence bounds.
(1) Training Bound (S_1): a bound for all n classes $\mathcal{H}_n^k$ simultaneously, with prob $\geq 1 - \delta/2$: $\forall k, \forall h \in \mathcal{H}_n^k, |L_D(h) - L_{S_1}(h)| \leq \epsilon_k = O\left(\sqrt{\frac{k \log n + \log(n/\delta)}{m}}\right)$; this bound is to ensure $w_{k_{opt}}$ (the ERM on S_1) is close to h_{opt}. $L_D(w_{k_{opt}}) \leq L_{S_1}(w_{k_{opt}}) + \epsilon_{k_{opt}} \leq L_{S_1}(h_{opt}) + \epsilon_{k_{opt}} \leq L_D(h_{opt}) + 2\epsilon_{k_{opt}}$.
(2) Validation Bound (S_2): need a bound for n specific hypotheses $\{w_1, w_2 ... w_n\}$ that were chosen by S_1. This is a union bound over a finite class of size n.

By Hoeffding's inequality, w.p. $\geq 1 - \delta/2$: $\forall k, |L_D(w_k) - L_{S_2}(w_k)| \leq \epsilon' = O\left(\sqrt{\frac{\log(n/\delta)}{m}}\right)$

let $\hat{h} = w_{\hat{k}}$, be the returned hypothesis and h_{opt} be the true optimum with k_{opt}:

$$\begin{aligned}
L_D(\hat{h}) &\leq L_{S_2}(\hat{h}) + \epsilon' && \text{(by bound 2)} \\
&\leq L_{S_2}(w_{k_{opt}}) + \epsilon' && \text{(by def. of } \hat{k} \text{ as minimizer on } S_2) \\
&\leq L_D(w_{k_{opt}}) + \epsilon' + \epsilon' && \text{(by bound 2)} \\
&\leq (L_D(h_{opt}) + 2\epsilon_{k_{opt}}) + 2\epsilon' && \text{(from step 1)}
\end{aligned}$$

$$L_D(\hat{h}) \leq L_D(h_{opt}) + 2\epsilon_{k_{opt}} + 2\epsilon'$$

since $k_{opt} \geq 1$, $L_D(\hat{h}) \leq L_D(h_{opt}) + O(\epsilon_{k_{opt}})$, matches the requirement.

3 Problem 3

(a) The key idea is to find the probability that all N independent runs of the learner A fail, and set this failure probability very small ($\leq \delta/2$).

The learner A is $(\epsilon/2, 1/2)$-PAC. So for a single run i, it is successful with $P(L_D(h_i) \leq \epsilon/2) \geq 1/2$. Therefore, the probability that it "fails" is $P(L_D(h_i) > \epsilon/2) \leq 1 - 1/2 = 1/2$.

Since the N runs are independent, the probability that *all* of them fail is:

$$P(\text{all } N \text{ runs fail}) = \prod_{i=1}^{N} P(\text{run } i \text{ fails}) \leq (1/2)^N$$

We want the probability of "at least one success" to be $\geq 1 - \delta/2$. This is the complement of "all fail".

$$P(\text{at least one success}) = 1 - P(\text{all } N \text{ runs fail}) \geq 1 - (1/2)^N$$

We need $1 - (1/2)^N \geq 1 - \delta/2$, which simplifies to $(1/2)^N \leq \delta/2$. The solve for N is:

$$2^N \geq 2/\delta$$

$$N \geq \log_2(2/\delta) \quad \text{or} \quad N \geq 1 + \log_2(1/\delta)$$

N should be at least $N \geq \log_2(2/\delta)$

(b) By hint, the key idea is to use the Chernoff-Hoeffding bound and a union bound to ensure that the test set S' is large enough to "faithfully" estimate the true error $L_D(h)$ for all N hypotheses simultaneously.

From (a), we assume (w.p. $\geq 1-\delta/2$) that we have at least one "good" h_g with $L_D(h_g) \leq \epsilon/2$. We need to ensure that our chosen h_{j^*} is not "bad" ($L_D(h_{j^*}) > \epsilon$).

if we can guaranty $|L_{S'}(h_i) - L_D(h_i)| \leq \epsilon/4$ for all hypotheses N, we succeed:

$$\begin{aligned} L_D(h_{j^*}) &\leq L_{S'}(h_{j^*}) + \epsilon/4 \quad \text{(by guarantee)} \\ &\leq L_{S'}(h_g) + \epsilon/4 \quad \text{(by def. of } h_{j^*}) \\ &\leq (L_D(h_g) + \epsilon/4) + \epsilon/4 \quad \text{(by guarantee)} \\ &\leq (\epsilon/2 + \epsilon/4) + \epsilon/4 = \epsilon \end{aligned}$$

fail this step if *any* h_i is misleading. By Hoeffding's inequality, for a *single* h_i:

$$P(|L_{S'}(h_i) - L_D(h_i)| > \epsilon/4) \leq 2e^{-2m(\epsilon/4)^2} = 2e^{-m\epsilon^2/8}$$

apply a union bound over all N hypotheses:

$$P(\exists i \text{ s.t. } |L_{S'}(h_i) - L_D(h_i)| > \epsilon/4) \leq \sum_{i=1}^{N} P(\dots) \leq N \cdot 2e^{-m\epsilon^2/8}$$

need this total failure probability to be $\leq \delta/2$:

$$2Ne^{-m\epsilon^2/8} \leq \delta/2$$

$$e^{-m\epsilon^2/8} \leq \frac{\delta}{4N}$$

$$-m\epsilon^2/8 \leq \ln\left(\frac{\delta}{4N}\right) = -\ln\left(\frac{4N}{\delta}\right)$$

$$m\epsilon^2/8 \geq \ln\left(\frac{4N}{\delta}\right)$$

m should be at least:$m \geq \frac{8}{\epsilon^2}\ln\left(\frac{4N}{\delta}\right)$

(c) The total complexity of the sample is the sum of all the samples used to train the hypotheses N and the samples used for the final validation test.

For training samples, the learner runs the learner A N times. Each run requires $m_A(\epsilon/2, 1/2)$ samples. That is:

$$Total\ Training\ Samples = N \cdot m_A(\epsilon/2, 1/2)$$

For validation samples: Learner B draws one test set S' of size m:

$$Total\ Validation\ Samples = m$$

The total sample complexity $m_B(\epsilon, \delta)$ is: $m_B(\epsilon, \delta) = N \cdot m_A(\epsilon/2, 1/2) + m$

4 Problem 4

(a) The key idea is to use the probabilistic method. Show that the expected error of a hypothesis chosen randomly from the set of $2k$ available hypotheses $\{\pm h_{w_1}, \dots, \pm h_{w_k}\}$ is at most $1/2 - 1/2k^2$.

it shows that at least one hypothesis must have an error, no larger than the expectation.

Consider the set of $2k$ hypotheses $\mathcal{H}' = \{h_{w_1}, \ldots, h_{w_k}, -h_{w_1}, \ldots, -h_{w_k}\}$. Since $L_D(h) + L_D(-h) = P(h \neq y) + P(h = y) = 1$, the average error is exactly $1/2$. The problem requires that at least one of these hypotheses has an "edge," or is better than random guessing.

The edge h is: $\frac{1}{2} - L_D(h) = \frac{1}{2}E[y \cdot h(x)]$.

The hint's probabilistic argument (involves analyzing the cases $y = 1$ and $y = -1$) shows that the average advantage over random guessing is non-zero.

Specifically, $\max_{h \in \mathcal{H}'} |E[y \cdot h(x)]| \geq 1/k^2$.

Since the error is $L_D(h) = 1/2 - \frac{1}{2}E[y \cdot h(x)]$, the best hypothesis h^* in this set must have an error of $L_D(h^*) \leq 1/2 - \frac{1}{2}(1/k^2)$, which is $L_D(h^*) \leq 1/2 - 1/2k^2$.

(b) The key idea is that we don't need to learn $\mathcal{H}_n^k$ directly. Instead, we use the given agnostic learner A (for the base class $\mathcal{H}_n$) as the weak-learner W for Adaboost. Part (a) guaranties that even on a re-weighted distribution D_t, there is always a good base hypothesis $h \in \mathcal{H}_n$ for A.

The weak-learner W works as follows:

- Input: A distribution D_t from Adaboost and a confidence δ_W.
- Procedure: Its internal parameters for the agnostic learner A: $\epsilon_A = 1/4k^2$ and $\delta_A = \delta_W$. It draws $m_A(\epsilon_A, \delta_A)$ samples from D_t and runs A on them, returning the resulting hypothesis $h_A \in \mathcal{H}_n$.

Part (a) guaranties that the best possible hypothesis in $\mathcal{H}_n$ has an error $\min_{h \in \mathcal{H}_n} L_{D_t}(h) \leq 1/2 - 1/2k^2$. The agnostic learner A guaranties (with probability $1 - \delta_A$) that its output h_A satisfies $L_{D_t}(h_A) \leq \min_{h \in \mathcal{H}_n} L_{D_t}(h) + \epsilon_A$.

Plug it in to our values:

$L_{D_t}(h_A) \leq (1/2 - 1/2k^2) + 1/4k^2 = \mathbf{1/2 - 1/4k^2}$.

This is a valid weak-learner with an edge $\boldsymbol{\gamma = 1/4k^2}$, which is $1/\text{poly}(n)$ as required.

(c) The key idea is to run Adaboost by W from part (b) as our weak-learner. The total complexity is the number of boosting rounds T multiplied by the complexity of W.

Adaboost guaranties a final error ϵ in the realizable case after $T = O(\frac{1}{\gamma^2}\log(1/\epsilon))$ rounds. Substituting our edge $\gamma = 1/4k^2$, the number of rounds is:

$$T = O(k^4 \log(1/\epsilon))$$

the weak-learner W must succeed in all T rounds, so we set its confidence as $\delta_W = \delta/T$.

The samples and runtime per round are:

Samples per round (m_W): W calls the agnostic learner A with $\epsilon_A = 1/4k^2$ and $\delta_A = \delta/T$. The sample complexity of an agnostic learner is $m_A \approx \text{poly}(n, 1/\epsilon_A, 1/\delta_A)$. Thus,

$$m_W = \text{poly}(n, k^2, T/\delta)$$

Runtime per round (t_W): The runtime of A is also

$$t_W = \text{poly}(n, 1/\epsilon_A, 1/\delta_A) = \text{poly}(n, k^2, T/\delta)$$

The total samples complexity is: $m_{total} = T \times m_W$, and the total runtime is:

$$t_{total} = T \times t_W$$

then, the total samples and runtime are:

Total Samples: $m_{total} = O(k^4 \log(1/\epsilon)) \times \text{poly}(n, k, T/\delta)$.

Total Runtime: $t_{total} = O(k^4 \log(1/\epsilon)) \times \text{poly}(n, k, T/\delta)$.

Substituting by $T = O(k^4 \log(1/\epsilon))$, both total complexities simplify. Since $k(n)$ is a polynomial in n, the final complexities are:

Total Sample Complexity: $\text{poly}\boldsymbol{(n, k, 1/\epsilon, 1/\delta)}$

Total Runtime: $\text{poly}\boldsymbol{(n, k, 1/\epsilon, 1/\delta)}$

5 Appendix

I asked Gemini for following questions as a supplementary, check this link: Gemini Prompt.