Hardness of learning boolean thresholds

ComComX • 17 May 2025

This post contains my solution to an exercise from the *Computational Learning Theory* course taught by Professor Varun Kanade at Oxford University. The problem reads as follows:

PROBLEM (Learning boolean threshold functions). Let $X_n = \{0,1\}^n$ and for $\mathbf{w} \in \{0,1\}^n$ and $k \in \mathbb{N}$, $f_{\mathbf{w},k} : X_n \longrightarrow \{0,1\}$ is a boolean threshold function defined by

$$f_{\mathbf{w},k} = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i \geq k \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Define the concept class of the threshold functions as

$$\mathsf{THRESHOLDS}_n = \{f_{\mathbf{w},k} | \mathbf{w} \in \{0,1\}^n, 0 \leq k \leq n\}, \tag{2}$$

$$\mathsf{THRESHOLDS} = \bigcup_{n \geq 1} \mathsf{THRESHOLDS}_n. \tag{3}$$

Show that unless $\mathsf{RP} = \mathsf{NP}$, there is no *efficient proper* PAC learning algorithm for $\mathsf{THRESHOLDS}$. □

This problem took me the whole Saturday in library, and after I came up with my solution, I found that my proof was different from the hint given by Kanade (and that's not bad for me :-)).

Let recall the definition of PAC model introduced by Leslie Valiant in his seminal paper in 1984, which was among the key contributions that led to his 2010 Turing Award:

Definition (PAC model). *For $n \geq 1$, let C_n be a concept class and H_n be a polynomially evaluable hypothesis class over instance space $X_n \subseteq \mathbb{R}^n$. Let $C = \bigcup_{n \geq 1} C_n$, $H = \bigcup_{n \geq 1} H_n$, and $X = \bigcup_{n \geq 1} X_n$. For any concept $c \in C$, let $size(c)$ be the representation size of c. Then we say that C is PAC learnable using the hypothesis class H if there exist an algorithm L satisfying that, for every $n \in \mathbb{N}$, for every concept $c \in C$, for every probability distribution D over X_n, for arbitrary small constants $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, if L is given access to an oracle $\mathsf{EX}(c, D)$, n, $size(c)$, ϵ, and δ, L outputs $h \in H_n$ that with*

probability of at least $1-\delta$ *(over the choice of instance* $\mathbf{x}$*), the prediction error* $\mathbb{P}_{\mathbf{x}\sim D}[c(\mathbf{x}) \neq h(\mathbf{x})] \leq \epsilon$*, and that the number of queries that* L *makes to* $\mathsf{EX}(c, D)$ *is bounded by a polynomial in* $n, size(c), 1/\epsilon$*, and* $1/\delta$*.* $\square$

In the above definition, the probability is over the randomness from the oracle $\mathsf{EX}(c, D)$ as well as any internal randomness of algorithm L. By *efficient*, we mean that the running time of L is polynomial in $n, size(c), 1/\epsilon$, and $1/\delta$, and by *proper*, we requires L to output a hypothesis h that belongs to C, meaning that the hypothesis class H is the same as the target concept class C, or in other words, L learns C using C.

PROOF IDEA: Now, I will show that there is no efficient algorithm for proper PAC learning $\mathsf{THRESHOLDS}$, unless $\mathsf{RP} = \mathsf{NP}$. The proof idea is just similar to any other hardness proof: by reducing a known NP-complete language to PAC learning $\mathsf{THRESHOLDS}$. Particularly, suppose A is a known NP-complete language and we wish to decide whether a given string/instance α is a member of A. Given a string α, what we need to do are to construct an instance of the PAC learning $\mathsf{THRESHOLDS}$ and then show that we can determine the membership of α in A via solving the constructed learning instance. More precisely, we need to construct a labeled training data set $S = \{\langle \mathbf{x}, f_{\mathbf{w},k}(\mathbf{x})\rangle | \mathbf{x} \in \{0,1\}^n, f_{\mathbf{w},k}(\mathbf{x}) \in \{0,1\}\}$, and show that there exists a hypothesis $\langle \mathbf{w}, k\rangle$ that is *consistent* with S if and only if $\alpha \in \mathsf{A}$. By *consistent*, we mean all predictions induced by $\langle \mathbf{w}, k\rangle$ match the labels in S. In addition, $|S|$ needs to be bounded by a polynomial in the length of α to guarantee that the reduction can be done in polynomial time.

A question might arise at this point: **given a PAC learning algorithm for $\mathsf{THRESHOLDS}$, how can we decide A (in polynomial time with high probability)?** It can be done by a general method as follows. Given a string α, suppose that we know how to construct S so that there is a consistent hypothesis $\langle \mathbf{w}, k\rangle$ with S if and only if $\alpha \in \mathsf{A}$. Let D be a uniform distribution over S. Let select the target error $\epsilon = \frac{1}{2|S|}$ and the confidence parameter $\delta = 1/2$. With D and S, we can simulate an oracle $\mathsf{EX}(c, D)$ and use it to learn $\langle \mathbf{w}, k\rangle$ with an error less than ϵ and confidence $1-\delta$ in polynomial time (the concept class c here is established by using the labels in S). We then simply check whether the output $\langle \mathbf{w}, k\rangle$ of the learning algorithm is consistent with S. If it is, we accept α, otherwise, we reject it.

Why does the above simulation successfully decide A (with probability of at least $1-\delta$)?

- In the case $\alpha \in \mathsf{A}$: by our construction of S, there is a concept class that is consistent with S. By the PAC learning guarantee, the output hypothesis $\langle \mathbf{w}, k\rangle$ of the learning algorithm is guaranteed with the prediction error less than ϵ and confidence $1-\delta$. Moreover, by the choice of ϵ it must be

true that this output hypothesis is consistent with S (for if there is even a single incorrect prediction, the error probability is $\frac{1}{|S|} > \frac{1}{2|S|} = \epsilon$ which contradicts the PAC learning guarantee). That is, the learning algorithm output a consistent hypothesis with confidence of $1 - \delta$.

- If $\alpha \notin \mathsf{A}$: there is no concept that is consistent with S and, obviously, the learning algorithm will return an inconsistent output. This can be easily checked in polynomial time.

Therefore, we can decide A (with probability of at least $1 - \delta$) by checking the output of the learning algorithm.

PROOF: the most challenging part of any hardness proof is choosing which NP-complete language to reduce from. For this proof, I have tried several problems before succeeding with $\mathsf{SET\_COVER}$. In fact, in the problem set on the course webpage, Kanade gave a hint to reduce from the Binary Integer Programming problem (which I haven't tried yet).

The SET-COVER problem is defined as follows: given a finite collection Q of n finite subsets and an integer u, is there a subset of Q with cardinality at most u whose union equals the union of Q? The corresponding language is given by:

$$\mathsf{SET\_COVER} = \left\{ \langle Q, u \rangle \middle| \exists I \subseteq [n], |I| \leq u, \bigcup_{i \in I} Q_i = \bigcup_{i \in [n]} Q_i \right\}, \tag{4}$$

where $[n] := \{1, 2, \ldots, n\}$ denotes the index set of subsets in Q. $\mathsf{SET\_COVER}$ is well-known to be NP-complete. Let $G = \bigcup_{i \in [n]} Q_i$ denote the ground set and $m = |G|$ be the total number of elements. There are two properties of this problem that motivated me to choose it: *i*) each element of G is covered by at most u subsets, and *ii*) each element of G is covered by at least one subset. However, to match the condition in the definition of $f_{\mathbf{w},k}$, these two "thresholds" must be the same value (which is k in (1)). I will show shortly that this hurdle can be overcome with a simple padding trick.

Given $\langle Q, u \rangle$, the training set S for the PAC learning $\mathsf{THRESHOLDS}$ is constructed as follows. S will consist of $m + 1$ labeled samples where the first m samples are labeled as 1 and the last sample is labeled as 0. Each of the first m samples corresponds to one element of the ground set G: for $1 \leq i \leq m$, we have $\mathbf{x}_i \in \{0, 1\}^{n+u}$ given by

$$x_{ij} = \begin{cases} 1 & \text{if } (j \leq n \text{ and } i \in Q_j) \text{ or } j > n \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Precisely, the first n entries of the vector indicate which subsets in Q that contain the ith element of G, and the last u entries are padded with 1. For the last sample which is labeled as 0, we have $\mathbf{x}_{m+1} \in \{0, 1\}^{n+u}$ given by

$$x_{m+1,j} = \begin{cases} 1 & \text{if } j \leq n \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

That is, $\mathbf{x}_{m+1}$ is a vector with value 1 in the first n entries and padding value 0 in the last u entries. The problem of determining the membership of $\langle Q, u\rangle$ in SET_COVER is now reduced to check the existence of a hypothesis $\langle \mathbf{w}, u+1\rangle$ that is consistent with S by PAC learning for the concept class $\mathsf{THRESHOLDS}_{n+u}$ given by:

$$\mathsf{THRESHOLDS}_{n+u} = \{f_{\mathbf{w},k} | \mathbf{w} \in \{0,1\}^{n+u}, 0 \leq k \leq u+1\}. \tag{7}$$

What remains now is to prove the **correctness of the reduction**, that is, to show that $\langle Q, u\rangle$ is in SET_COVER if and only if S is consistent with some output $\langle \mathbf{w}, u+1\rangle$ of the learning algorithm:

- First, suppose the ground set G can be covered no more than u subsets. Let I be the index set of this cover. We define the vector $\mathbf{w} \in \{0,1\}^{n+u}$ as follows

$$w_j = \begin{cases} 1 & \text{if } j \in I \text{ or } j > n \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

  Then the hypothesis $\langle \mathbf{w}, u+1\rangle$ is in fact consistent with S. Specifically, consider the vector $\mathbf{x}_i$ corresponding to the ith element of G, since every element is covered by at least one subset, we have $\sum_{j=1}^{n} w_j x_{ij} \geq 1$. Moreover, since the last u entries of $\mathbf{x}_i$ and those of $\mathbf{w}$ are all 1, we have $\sum_{j=n+1}^{n+u} w_j x_{ij} = u$. This implies that $\sum_{j=1}^{n+u} w_j x_{ij} \geq u+1$, which is consistent with the label of $\mathbf{x}_i$. For the last sample, we have that

$$\begin{aligned} \sum_{j=1}^{n+u} w_j x_{m+1,j} &= \sum_{j=1}^{n} w_j x_{m+1,j} + \sum_{j=n+1}^{n+u} w_j x_{m+1,j} \\ &\leq u + 0 \\ &< u + 1, \end{aligned}$$

  which is also consistent with the label 0. Here the inequality follows from the fact that there are at most u positions with value 1 in the first entries of $\mathbf{w}$ and that all the last u entries of $\mathbf{x}_{m+1}$ are zero by their definition given in (6).

- For the other direction, suppose that the hypothesis $\langle \mathbf{w}, u+1\rangle$ is consistent with S, the cover of G is simply defined as $I = \{j | 1 \leq j \leq n \text{ and } w_j = 1\}$. It is straightforward to verify that this is a valid cover. Specifically, by the consistency, we have that $\sum_{j=1}^{n+u} w_j x_{m+1,j} < u+1$. By the definition of $\mathbf{x}_{m+1}$, it must be that $\sum_{j=1}^{n} w_j x_{m+1,j} \leq u$, which implies $|I| \leq u$. For the ith element of the

ground set, also by the consistency, we have that $\sum_{j=1}^{n+u} w_j x_{ij} \geq u + 1$. It follows that $\sum_{j=1}^{n} w_j x_{ij} \geq 1$, or there is at least one chosen subset that covers this element due to the definition of $\mathbf{x}_i$ as given in (5).

Now we know how to reduce SET_COVER to PAC learning THRESHOLDS. If there is an efficient proper PAC learning algorithm for THRESHOLDS, then we would have a randomized algorithm that decides an NP-complete language A in polynomial time with error probability less than δ. This would imply that RP $=$ NP and, since (it is widely believed that) RP $\neq$ NP, it must be the case that such a learning algorithm, if it exists, is inefficient. This completes the proof. $\square$