Superpolynomial Lower Bounds for Learning Monotone Classes

Nader H. Bshouty ✉
Department of Computer Science, Technion, Haifa, Israel

Abstract

Koch, Strassle, and Tan [SODA 2023], show that, under the randomized exponential time hypothesis, there is no distribution-free PAC-learning algorithm that runs in time $n^{\tilde{O}(\log\log s)}$ for the classes of n-variable size-s DNF, size-s Decision Tree, and $\log s$-Junta by DNF (that returns a DNF hypothesis). Assuming a natural conjecture on the hardness of set cover, they give the lower bound $n^{\Omega(\log s)}$. This matches the best known upper bound for n-variable size-s Decision Tree, and $\log s$-Junta.

In this paper, we give the same lower bounds for PAC-learning of n-variable size-s Monotone DNF, size-s Monotone Decision Tree, and Monotone $\log s$-Junta by DNF. This solves the open problem proposed by Koch, Strassle, and Tan and subsumes the above results.

The lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time, and can compute the target function on all the points of the support of the distribution in polynomial time.

2012 ACM Subject Classification Theory of computation

Keywords and phrases PAC Learning, Monotone DNF, Monotone Decision Tree, Monotone Junta, Lower Bound

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.34

Category RANDOM

Acknowledgements I would like to express my sincere gratitude to the reviewers of RANDOM for their valuable comments on this research. Their feedback was greatly appreciated and helped improve the quality of this work.

1 Introduction

In the distribution-free PAC learning model [13], the learning algorithm of a class of functions C has access to an unknown target function $f \in C$ through labeled examples $(x, f(x))$ where x are drawn according to an unknown but fixed probability distribution $\mathcal{D}$. For a class of hypothesis $H \supseteq C$, we say that the learning algorithm $\mathcal{A}$ *PAC-learns* C *by* H in time T and error ϵ if for every target $f \in C$ and distribution $\mathcal{D}$, $\mathcal{A}$ runs in time T and outputs a hypothesis $h \in H$ which, with probability at least $2/3$, is ϵ-close to f with respect to $\mathcal{D}$. That is, satisfies $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[f(\boldsymbol{x}) \neq h(\boldsymbol{x})] \leq \epsilon$.

Koch et al., [10], show that, under the randomized exponential time hypothesis (ETH), there is no PAC-learning algorithm that runs in time $n^{\tilde{O}(\log\log s)}$ for the classes of n-variable size-s DNF, size-s Decision Tree and $\log s$-Junta by DNF. Assuming a natural conjecture on the hardness of set cover, they give the lower bound $n^{\Omega(\log s)}$. Their lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and can compute the target function on all the points of the support of the distribution in polynomial time.

In this paper, we give the same lower bounds for PAC-learning of the classes n-variable size-s Monotone DNF, size-s Monotone Decision Tree and Monotone $\log s$-Junta by DNF. This solves the open problem proposed by Koch, Strassle, and Tan [10].

In various learning models, it is widely recognized that the task of learning classes of monotone Boolean functions is significantly simpler compared to learning classes of non-monotone Boolean functions. An illustrative example is the distribution-free PAC learning of monotone DNF within a model that permits membership queries, which can be accomplished efficiently in polynomial time [2]. Conversely, the challenge of learning DNF in the same model remains an unsolved problem, emphasizing the inherent difficulty associated with non-monotone Boolean functions. In this paper, we demonstrate that for specific classes, the task of PAC-learning monotone Boolean functions under the uniform distribution is equally challenging as learning non-monotone Boolean functions.

1.1 Our Results

In this paper, we prove the following three Theorems.

▶ **Theorem 1.** *Assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable* Monotone $(\log s)$-Junta*, size-s* Monotone DT *and size-s* Monotone DNF[1] *by DNF with $\epsilon = 1/(16n)$ must take at least*

$$n^{c\frac{\log\log s}{\log\log\log s}}$$

time.

▶ **Theorem 2.** *Assuming a plausible conjecture on the hardness of* Set-Cover [2]*, there is a constant c such that any PAC learning algorithm for n-variable* Monotone $(\log s)$-Junta*, size-s* Monotone DT *and size-s* Monotone DNF *by DNF with $\epsilon = 1/(16n)$ must take at least*

$$n^{c\log s}$$

time.

▶ **Theorem 3.** *Assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable* Monotone $(\log s)$-Junta*, size-s* Monotone DT *and size-s* Monotone DNF *by size-s* DNF *with $\epsilon = 1/(16n)$ must take at least*

$$n^{c\log s}$$

time.

All the above lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and can compute the target on all the points of the support of the distribution in polynomial time.

In the following two subsections, we give the technique used in [10] to prove Theorem 1 for $(\log s)$-Junta, and the technique we use here to extend the result to Monotone $(\log s)$-Junta.

---

[1] The results concerning *size-s* Monotone DT and *size-s* Monotone DNF stem from the result on Monotone $(\log s)$-Junta. This relationship also holds for the other theorems. Here, we present a comprehensive list of all these classes to highlight their significance.

[2] See Conjecture 1.

1.2 Previous Technique

In [10], Koch, Strassle, and Tan show that under the randomized exponential time hypothesis, there is no PAC-learning algorithm that runs in time $n^{\tilde{O}(\log\log n)}$ for the class of $\log n$-Junta[3] by DNF. The results for the other classes follow immediately from this result, since all other classes contain $\log n$-Junta. All prior works [1, 6] ruled out only $poly(n)$ time algorithms.

The result in [10] uses the hardness result of (k,k')-Set-Cover where one needs to distinguish between instances that have set cover of size at most k from instances that have minimum-size set cover greater than k':

1. For some parameters k and k' that depends on N, assuming randomized ETH, there is a constant $\lambda<1$ such that (k,k')-Set-Cover on N vertices cannot be solved in time $N^{\lambda k}$.

First, for each set cover instance $\mathcal{S}$, they identify each element in the universe with an assignment in $\{0,1\}^n$ and construct in polynomial time a target function $\Gamma^{\mathcal{S}}:\{0,1\}^n\to\{0,1\}$ and a distribution $\mathcal{D}^{\mathcal{S}}$ that satisfies:

2. The instance $\mathcal{S}$ has minimum-size set cover $\mathrm{opt}(\mathcal{S})$ if and only if the function $\Gamma^{\mathcal{S}}$ is a conjunction of $\mathrm{opt}(\mathcal{S})$ unnegated variables[4] over the distribution $\mathcal{D}^{\mathcal{S}}$.[5]

For a DNF F and $x\in\{0,1\}^n$, they define $\mathrm{width}_F(x)$ to be the size of the smallest term T in F that satisfies $T(x)=1$. They then show that

3. Any DNF F with expected width $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{D}^{\mathcal{S}}}[\mathrm{width}_F(\boldsymbol{x})]\le\mathrm{opt}(\mathcal{S})/2$ is $(1/(2N))$-far from $\Gamma^{\mathcal{S}}$ with respect to $\mathcal{D}^{\mathcal{S}}$ where N is the size[6] of $\mathcal{S}$. That is, $\Pr_{\boldsymbol{x}\sim\mathcal{D}^{\mathcal{S}}}[F(\boldsymbol{x})\neq\Gamma^{\mathcal{S}}(\boldsymbol{x})]\ge 1/(2N)$.

They then use the following gap amplification technique. They define the function $\Gamma^{\mathcal{S}}_{\oplus\ell}:(\{0,1\}^\ell)^n\to\{0,1\}$ where for $y=(y_1,\ldots,y_n)$, $y_i=(y_{i,1},\ldots,y_{i,\ell})\in\{0,1\}^\ell$, $i\in[n]$, we have $\Gamma^{\mathcal{S}}_{\oplus\ell}(y)=\Gamma^{\mathcal{S}}(\oplus y_1,\ldots,\oplus y_n)$ and $\oplus y_i=y_{i,1}+\cdots+y_{i,\ell}$. They also extend the distribution $\mathcal{D}^{\mathcal{S}}$ to a distribution $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$ over domain $(\{0,1\}^\ell)^n$ and prove that

4. $\Gamma^{\mathcal{S}}_{\oplus\ell}(y)$ is a $(\mathrm{opt}(\mathcal{S})\ell)$-Junta over $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$.
5. Any DNF formula F with expected width $\mathbf{E}_{\boldsymbol{y}\sim\mathcal{D}^{\mathcal{S}}_{\oplus\ell}}[\mathrm{width}_F(\boldsymbol{y})]\le\mathrm{opt}(\mathcal{S})\ell/4$ is $(1/(4N))$-far from $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$.

Item 4 follows from the definition of $\Gamma^{\mathcal{S}}_{\oplus\ell}$ and item 2. To prove Item 5, they show that if, to the contrary, there is a DNF F of expected width at most $\mathrm{opt}(\mathcal{S})\ell/4$ that is $1/(4N)$-close to $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$, then there is $j\in[\ell]$ and a projection of all the variables that are not of the form $y_{i,j}$ that gives a DNF F^* of expected width at most $\mathrm{opt}(\mathcal{S})/2$ that is $1/(2N)$-close to $\Gamma^{\mathcal{S}}$ with respect to $\mathcal{D}^{\mathcal{S}}$. Then, by item 3, we get a contradiction.

They then show that

6. Any size-s DNF that is $(1/(4N))$-close to $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$ has average width $\mathbf{E}_{\boldsymbol{y}\sim\mathcal{D}^{\mathcal{S}}_{\oplus\ell}}[\mathrm{width}_F(\boldsymbol{y})]\le 4\log s$.

If F is $(1/(4N))$-close to $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$, then, by items 5 and 6, $4\log s\ge\mathbf{E}_{\boldsymbol{y}\sim\mathcal{D}^{\mathcal{S}}_{\oplus\ell}}[\mathrm{width}_F(\boldsymbol{y})]\ge\mathrm{opt}(\mathcal{S})\ell/4$ and then $s\ge 2^{\mathrm{opt}(\mathcal{S})\ell/16}$. Therefore,

7. Any DNF of size less than $2^{\mathrm{opt}(\mathcal{S})\ell/16}$ is $(1/(4N))$-far from $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$.

Now, let $k=\tilde{O}(\log\log n)$. Suppose, to the contrary, that there is a PAC-learning algorithm for $\log n$-Junta by DNF with error $\epsilon=1/(8N)$ that runs in time $t=n^{\lambda k/2}=n^{\tilde{O}(\log\log n)}$, where λ is the constant in item 1. Given a (k,k')-Set-Cover instance, we run the learning

[3] k-Junta are Boolean functions that depend on at most k variables
[4] Their reduction gives a conjunction of negated variable. So here, we are referring to the dual function.
[5] That is, there is a term T with $\mathrm{opt}(\mathcal{S})$ variables such that for every x in the support of $\mathcal{D}^{\mathcal{S}}$, $\Gamma^{\mathcal{S}}(x)=T(x)$.
[6] N is the number of sets plus the size of the universe in $\mathcal{S}$.

algorithm for $\Gamma^{\mathcal{S}}_{\oplus\ell}$ for $\ell = \log n/k$. If the instance has set cover at most k, then by item 4, $\Gamma^{\mathcal{S}}_{\oplus\ell}$ is $\log n$-Junta. Then the algorithm learns the target and outputs a hypothesis that is $(1/(8N))$-close to $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$.

On the other hand, if the instance has a minimum-size set cover of at least k', then any learning algorithm that runs in time $t = n^{\lambda k/2} = n^{\tilde{O}(\log\log n)}$ cannot output a DNF of size more than t terms. By item 7, any DNF of size less than $2^{k'\log n/(16k)} \leq 2^{\mathrm{opt}(\mathcal{S})\ell/16}$ is $(1/(4N))$-far from $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$. By choosing the right parameters k and k', we have $2^{k'\log n/(16k)} > t$, and therefore, any DNF that the algorithm outputs has error of at least $1/(4N)$.

Therefore, by estimating the distance of the output of the learning algorithm from $\Gamma^{\mathcal{S}}_{\oplus\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\oplus\ell}$, we can distinguish between instances that have set cover of size less than or equal to k from instances that have a minimum-size set cover greater than k' in time $t = n^{\lambda k/2}$. Thus, we got an algorithm for (k,k')-Set-Cover that runs in time $n^{\lambda k/2} < n^{\lambda k}$. This contradicts item 1 and finishes the proof of the first lower bound.

Assuming a natural conjecture on the hardness of set cover, they give the lower bound $n^{\Omega(\log s)}$. We will discuss this in Section 5.

1.3 Our Technique

In this paper, we also use the hardness result of (k,k')-Set-Cover . As in [10], we identify each element in the universe with an assignment in $\{0,1\}^n$ and use the function $\Gamma^{\mathcal{S}}$ and the distribution $\mathcal{D}^{\mathcal{S}}$ that satisfies:

1. The instance $\mathcal{S}$ has minimum-size set cover $\mathrm{opt}(\mathcal{S})$ if and only if the function $\Gamma^{\mathcal{S}}$ is a conjunction of $\mathrm{opt}(\mathcal{S})$ variables over the distribution $\mathcal{D}^{\mathcal{S}}$.

We then build a *monotone* target function $\Gamma^{\mathcal{S}}_{\ell}$ and a distribution $\mathcal{D}^{\mathcal{S}}_{\ell}$ and use a different approach to show that any DNF of size less than $2^{\mathrm{opt}(\mathcal{S})\ell/20}$ is $(1/(8N) - 2^{-\mathrm{opt}(\mathcal{S})\ell/20})$-far from $\Gamma^{\mathcal{S}}_{\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\ell}$.

We define, for any odd ℓ, the monotone function $\Gamma^{\mathcal{S}}_{\ell} : (\{0,1\}^{\ell})^n \to \{0,1\}$ where for $y = (y_1,\ldots,y_n)$, $y_i = (y_{i,1},\ldots,y_{i,\ell})$, $i \in [n]$, we have $\Gamma^{\mathcal{S}}_{\ell}(y) = \Gamma^{\mathcal{S}}(\textsc{Majority}(y_1),\ldots,\textsc{Majority}(y_n))$ where Majority is the majority function. A distribution $\mathcal{D}^{\mathcal{S}}_{\ell}$ is also defined such that

2. $\Pr_{\boldsymbol{y}\sim\mathcal{D}^{\mathcal{S}}_{\ell}}[\Gamma^{\mathcal{S}}_{\ell}(\boldsymbol{y}) = 0] = \Pr_{\boldsymbol{y}\sim\mathcal{D}^{\mathcal{S}}_{\ell}}[\Gamma^{\mathcal{S}}_{\ell}(\boldsymbol{y}) = 1] = 1/2$.

In the paper, $\mathcal{D}^{\mathcal{S}}_{\ell}$ is denoted by $\mathcal{D}_{\ell}$. To see the definition, refer to Definitions 3 and 4. Roughly speaking, the distribution is defined in such a way that removing a few coordinates results in the distribution becoming almost uniform. It is clear from the definition of $\Gamma^{\mathcal{S}}_{\ell}$ and item 1 that

3. $\Gamma^{\mathcal{S}}_{\ell}(y)$ is a monotone $(\mathrm{opt}(\mathcal{S})\ell)$-Junta over $\mathcal{D}^{\mathcal{S}}_{\ell}$.

We then define the *monotone size* of a term T to be the number of unnegated variables that appear in T. The intuition for this definition is that negated variables do not contribute to reducing the size of the DNF of monotone functions, and therefore they may as well be ignored. We first show that

4. For every DNF $F : (\{0,1\}^{\ell})^n \to \{0,1\}$ of size $|F| \leq 2^{\mathrm{opt}(\mathcal{S})\ell/5}$ that is ϵ-far from $\Gamma^{\mathcal{S}}_{\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\ell}$, there is another DNF F' of size $|F'| \leq 2^{\mathrm{opt}(\mathcal{S})\ell/5}$ with terms of monotone size at most $\mathrm{opt}(\mathcal{S})\ell/5$ that is $(\epsilon - 2^{-\mathrm{opt}(\mathcal{S})\ell/20})$-far from $\Gamma^{\mathcal{S}}_{\ell}$ with respect to $\mathcal{D}^{\mathcal{S}}_{\ell}$.

This is done by simply showing that terms of large monotone size in the DNF F have a small weight according to the distribution $\mathcal{D}^{\mathcal{S}}_{\ell}$ and, therefore, can be removed from F with the cost of $-2^{-\mathrm{opt}(\mathcal{S})\ell/20}$ in the error.

We then, roughly speaking, show that

5. Let F' be a DNF of size $|F'| \leq 2^{\text{opt}(\mathcal{S})\ell/5}$ with terms of monotone size at most $\text{opt}(\mathcal{S})\ell/5$. For every $y \in (\{0,1\}^\ell)^n$ in the support of $\mathcal{D}_\ell^{\mathcal{S}}$ that satisfies $\Gamma_\ell^{\mathcal{S}}(y) = 1$, either
   - $F'(y) = 0$ or
   - $F'(y) = 1$, and at least $1/(2N)$ fraction of the points z below y in the lattice $(\{0,1\}^\ell)^n$ that are in the support of $\mathcal{D}_\ell^{\mathcal{S}}$ satisfies $F'(z) = 1$ and $\Gamma_\ell^{\mathcal{S}}(z) = 0$.

Roughly speaking, the latter sub-item follows from the fact that when the monotone size of the terms of F is small and $F'(y) = 1$ then y satisfies a small term T in F'. Now, if z with $\Gamma_\ell^{\mathcal{S}}(z) = 0$ is a randomly chosen point that is almost uniformly distributed below y, but not significantly distant from y, then there exists a sufficiently large probability for $T(z) = 1$ and then $F'(z) = 1$.

By item 5, either $1/(4N)$ fraction of the vectors y that satisfy $\Gamma_\ell^{\mathcal{S}}(y) = 1$ satisfy $F'(y) = 0$ or $(1 - 1/(4N))/(2N) > 1/(4N)$ fraction of the points z that satisfy $\Gamma_\ell^{\mathcal{S}}(z) = 0$ satisfy $F'(z) = 1$. Therefore, with item 2, we get that F' is $1/(8N)$-far from $\Gamma_\ell^{\mathcal{S}}$ with respect to $\mathcal{D}_\ell^{\mathcal{S}}$. This, with item 4, implies that

6. If $F : (\{0,1\}^\ell)^n \to \{0,1\}$ is a DNF of size $|F| < 2^{\text{opt}(\mathcal{S})\ell/20}$, then F is $(1/(8N) - 2^{-\text{opt}(\mathcal{S})\ell/20})$-far from $\Gamma_\ell^{\mathcal{S}}$ with respect to $\mathcal{D}_\ell^{\mathcal{S}}$.

The rest of the proof is almost the same as in [10]. See the discussion in subsection 1.2 after item 7.

Assuming a natural conjecture on the hardness of set cover, we establish a lower bound of $n^{\Omega(\log s)}$. The details and proof of this result will be discussed in Section 5, where we utilize a stronger version of item 4.

1.4 Upper Bounds

The only known distribution-free algorithm for $\log s$-Junta is the trivial algorithm that, for every set of $m = \log s$ variables $S = \{x_{i_1}, \ldots, x_{i_m}\}$, checks if there is a function that depends on S and is consistent with the examples. This algorithm takes $n^{O(\log s)}$ time.

For size-s decision tree and monotone size-s decision tree, the classic result of Ehrenfeucht and Haussler [5] gives a distribution-free time algorithm that runs in time $n^{O(\log s)}$ and outputs a decision tree of size $n^{O(\log s)}$.

The learning algorithm is as follows: Let T be the target decision tree of size s. First, the algorithm guesses the variable at the root of the tree T and then guesses which subtree of the root has size at most $s/2$. Then, it recursively constructs the tree of size $s/2$. When it succeeds, it continues to construct the other subtree.

For size-s DNF and monotone size-s DNF, Hellerstein et al. [7] gave a distribution-free proper learning algorithm that runs in time $2^{\tilde{O}(\sqrt{n \log s})}$.

To the best of our knowledge, all the other results in the literature for learning the above classes are either restricted to the uniform distribution or, in addition, use black box queries or return hypotheses that are not DNF.

We recommend that readers who are not experts in the field refer to the first two sections in [10].

2 Definitions and Preliminaries

In this section, we give the definitions and preliminary results that are needed to prove our results.

2.1 Set Cover

Let $\mathcal{S} = (S, U, E)$ be a bipartite graph on $N = n + |U|$ vertices where $S = [n]$, and for every $u \in U$, $\deg(u) > 0$. We say that $C \subseteq S$ is a set cover of $\mathcal{S}$ if every vertex in U is adjacent to some vertex in C. The Set-Cover problem is to find a minimum-size set cover. We denote by $\mathrm{opt}(\mathcal{S})$ the size of a minimum-size set cover for $\mathcal{S}$.

We identify each element $u \in U$ with the vector $(u_1, \ldots, u_n) \in \{0,1\}^n$ where $u_i = 0$ if and only if $(i, u) \in E$. We will assume that those vectors are distinct. If there are two distinct elements $u, u' \in U$ that have the same vector, then you can remove one of them from the graph. This is because every set cover that covers one of them covers the other.

▶ **Definition 1.** *The (k, k')-Set-Cover problem is the following: Given as input a set cover instance $\mathcal{S} = (S, U, E)$, and parameters k and k'. Output Yes if $\mathrm{opt}(\mathcal{S}) \leq k$ and No if $\mathrm{opt}(\mathcal{S}) > k'$.*

2.2 Hardness of Set-Cover

Our results are conditioned on the following randomized exponential time hypothesis (ETH) **Hypothesis:** [3, 4, 8, 9, 12]. There exists a constant $c \in (0,1)$ such that 3-SAT on n variables cannot be solved by a randomized algorithm in $O(2^{cn})$ time with success probability at least 2/3.

The following is proved in [11]. See also Theorem 7 in [10]

▶ **Lemma 2** ([11]). *Let $k \leq \frac{1}{2}\frac{\log\log N}{\log\log\log N}$ and $k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k}$ be two integers. Assuming randomized ETH, there is a constant $\lambda \in (0,1)$ such that there is no randomized $N^{\lambda k}$ time algorithm that can solve (k, k')-Set-Cover on N vertices with high probability.*

2.3 Concept Classes

In Appendix A we give the definition of *monotone* function, *literal*, *term*,*clause*, *monotone term*, *DNF*, *monotone DNF*. Then the classes, size-s Monotone DNF, size s-Monotone DT and Monotone k-Junta.

The *size* of a term T, $|T|$, is the number of literals in the term T. The *size* $|F|$ of a DNF (resp. CNF) F is the number of terms (resp. clauses) in F.

It is well known that

$$\text{Monotone } (\log s)\text{-Junta} \subset \text{size-}s \text{ Monotone DT} \subset \text{size-}s \text{ Monotone DNF}. \quad (1)$$

2.4 Functions and Distributions

For any set R, we define $\mathcal{U}(R)$ to be the uniform distribution over R. For a distribution $\mathcal{D}$ over $\{0,1\}^n$ and two Boolean functions f and g, we define $\mathrm{dist}_{\mathcal{D}}(f,g) = \Pr_{\boldsymbol{x}\sim\mathcal{D}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$. Here, bold letters denote random variables. If $\mathrm{dist}_{\mathcal{D}}(f,g) = 0$, then we say that $f = g$ *over* $\mathcal{D}$. For a class of functions C, we say that f is in C over $\mathcal{D}$ (or just, is C over $\mathcal{D}$) if there is a function $g \in C$ such that $f = g$ over $\mathcal{D}$.

▶ **Definition 3** ($\Gamma^{\mathcal{S}}$ and $\mathcal{D}^{\mathcal{S}}$). *Let $\mathcal{S} = (S, U, E)$ be a set cover instance with $S = [n]$. Recall that we identify each element $u \in U$ with the vector $(u_1, \ldots, u_n) \in \{0,1\}^n$ where $u_i = 0$ if and only if $(i, u) \in E$. We define the partial function $\Gamma^{\mathcal{S}} : \{0,1\}^n \to \{0,1\}$ where $\Gamma^{\mathcal{S}}(x) = 0$ if $x \in U$ and $\Gamma^{\mathcal{S}}(1^n) = 1$. We define the distribution $\mathcal{D}^{\mathcal{S}}$ over $\{0,1\}^n$ where $\mathcal{D}^{\mathcal{S}}(x) = 1/2$ if $x = 1^n$, $\mathcal{D}^{\mathcal{S}}(x) = 1/(2|U|)$ if $x \in U$, and $\mathcal{D}^{\mathcal{S}}(x) = 0$ otherwise. We will remove the superscript $\mathcal{S}$ when it is clear from the context and write Γ and $\mathcal{D}$.*

▶ **Fact 1.** *We have*

1. *$C \subseteq S$ is a set cover of $\mathcal{S} = (S, U, E)$, if and only if $\Gamma(x) = \bigwedge_{i\in C} x_i$ over $\mathcal{D}$.*
2. *In particular, If T is a monotone term of size $|T| < \text{opt}(\mathcal{S})$, then there is $u \in U$ such that $T(u) = 1$.*

Proof. Let C be a set cover of $\mathcal{S}$. First, we have $\Gamma(1^n) = 1$. Now, since C is a set cover, every vertex $u \in U$ is adjacent to some vertex in C. This is equivalent to: for every assignment $u \in U$, there is $i \in C$ such that $u_i = 0$. Therefore, $\wedge_{i\in C} u_i = 0$ for all $u \in U$. Thus, $\Gamma(x) = \bigwedge_{i\in C} x_i$ over $\mathcal{D}$.

The other direction can be easily seen by tracing backward in the above proof. ◀

For an odd ℓ, define $\Delta^0 = \{a \in \{0,1\}^\ell | \text{wt}(a) = \lfloor \ell/2 \rfloor\}$ and $\Delta^1 = \{a \in \{0,1\}^\ell | \text{wt}(a) = \lceil \ell/2 \rceil\}$, where $\text{wt}(a)$ is the Hamming weight of a. Notice that $|\Delta^0| = |\Delta^1| = \binom{\ell}{\lfloor \ell/2 \rfloor}$.

▶ **Definition 4** (Γ_ℓ, $\mathcal{D}_\ell$, Δ^0_n and Δ^1_n)**.** *For an odd ℓ, define $\Delta^1_n = (\Delta^1)^n$ and[7] $\Delta^0_n := \cup_{u\in U} \prod_{i=1}^n \Delta^{u_i} = \cup_{u\in U}(\Delta^{u_1} \times \Delta^{u_2} \times \cdots \times \Delta^{u_n})$. Define the distribution $\mathcal{D}_\ell : (\{0,1\}^\ell)^n \to [0,1]$ to be $\mathcal{D}_\ell(y) = 1/(2|\Delta^1_n|) = 1/(2|\Delta^1|^n)$ if $y \in \Delta^1_n$, $\mathcal{D}_\ell(y) = 1/(2|\Delta^0_n|) = 1/(2|U| \cdot |\Delta^0|^n)$ if $y \in \Delta^0_n$, and $\mathcal{D}_\ell(y) = 0$ otherwise. We define the partial function Γ_ℓ over the support $\Delta^0_n \cup \Delta^1_n$ of $\mathcal{D}_\ell$ to be 1 if $y \in \Delta^1_n$ and 0 if $y \in \Delta^0_n$.*

We note here that the distribution $\mathcal{D}_\ell$ is well-defined. This is because: First, the sum of the distribution of the points in Δ^1_n is $1/2$. Second, for two different $u, u' \in U$, we have that $\prod_{i=1}^n \Delta^{u_i}$ and $\prod_{i=1}^n \Delta^{u'_i}$ are disjoint sets. Therefore, $|\Delta^0_n| = |U| \cdot |\Delta^0|^n$, and therefore, the sum of the distribution of all the points in Δ^0_n is half. In particular,

▶ **Fact 2.** *We have* $\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) = 1] = \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) = 0] = \Pr_{\mathcal{D}_\ell}[\Delta^1_n] = \Pr_{\mathcal{D}_\ell}[\Delta^0_n] = \frac{1}{2}$.

For $y \in (\{0,1\}^\ell)^n$, we write $y = (y_1, \ldots, y_n)$, where $y_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,\ell}) \in \{0,1\}^\ell$. Let $(\textsc{Majority}(y_i))_{i\in[n]} = (\textsc{Majority}(y_1), \ldots, \textsc{Majority}(y_n))$ where $\textsc{Majority}$ is the majority function.

▶ **Fact 3.** *If $C \subseteq S$ is a set cover of $\mathcal{S}$, then $\Gamma_\ell(y) = \Gamma((\textsc{Majority}(y_i))_{i\in[n]}) = \bigwedge_{i\in C} \textsc{Majority}(y_i)$ over $\mathcal{D}$. In particular, Γ_ℓ is Monotone $\text{opt}(\mathcal{S})\ell$-Junta over $\mathcal{D}$.*

Proof. First notice that $\textsc{Majority}(x) = 1$ if $x \in \Delta^1$ and $\textsc{Majority}(x) = 0$ if $x \in \Delta^0$. Therefore, for $x \in \Delta^\xi$, $\xi \in \{0,1\}$ we have $\textsc{Majority}(x) = \xi$.

For $y \in \Delta^1_n = (\Delta^1)^n$, $(\textsc{Majority}(y_i))_{i\in[n]} = 1^n$ and $\Gamma_\ell(y) = 1 = \Gamma(1^n)$.

For $y \in \Delta^0_n = \cup_{u\in U}(\Delta^{u_1} \times \Delta^{u_2} \times \cdots \times \Delta^{u_n})$, there is u such that $y \in \Delta^{u_1} \times \Delta^{u_2} \times \cdots \times \Delta^{u_n}$. Then, $(\textsc{Majority}(y_i))_{i\in[n]} = u$ and $\Gamma_\ell(y) = \Gamma((\textsc{Majority}(y_i))_{i\in[n]}) = \Gamma(u) = 0$. ◀

For $t \in [\ell], \xi \in \{0,1\}$ and $u \in \{0,1\}^\ell$, we define $u^{t\leftarrow\xi} \in \{0,1\}^\ell$ the vector that satisfies

$$u_i^{t\leftarrow\xi} = \begin{cases} u_i & i \neq t \\ \xi & i = t \end{cases}.$$

Let $z \in (\{0,1\}^\ell)^n$. For $j \in [\ell]^n$ and $a \in \{0,1\}^n$, define $z^{j\leftarrow a} = (z_1^{j_1\leftarrow a_1}, \ldots, z_n^{j_n\leftarrow a_n})$. For a set $V \subseteq \{0,1\}^n$, we define $z^{j\leftarrow V} = \{z^{j\leftarrow v} | v \in V\}$.

We define $\text{one}(z) = \prod_{i=1}^n \{m_i | z_{i,m_i} = 1\} = \{m_1 | z_{1,m_1} = 1\} \times \cdots \times \{m_n | z_{n,m_n} = 1\}$.

[7] Here $\Delta^\xi = \Delta^0$ if $\xi = 0$ and Δ^1 if $\xi = 1$.

▶ **Fact 4.** *Let $w \in \Delta_n^1$, $j \in \text{one}(w)$, and T be a term that satisfies $T(w) = 1$. Then*
1. $w^{j \leftarrow U} \subseteq \Delta_n^0$.
2. $|w^{j \leftarrow U}| = |U|$.
3. *If $T^j(y_{1,j_1}, \ldots, y_{n,j_n})$ is the conjunction of all the variables that appear in T of the form y_{i,j_i}, then $T(w^{j \leftarrow a}) = T^j(a)$.*

Proof. We first prove item 1. Let $u \in U$ and i be any integer in $[n]$. Since $w \in \Delta_n^1$, we have $w_i \in \Delta^1$. Since $j \in one(w)$, we have $w_{i,j_i} = 1$. Therefore, $w_i^{j_i \leftarrow u_i} \in \Delta^{u_i}$ for all $i \in [n]$ and $w^{j \leftarrow u} \in \prod_{i=1}^n \Delta^{u_i}$. Thus, $w^{j \leftarrow u} \in \Delta_n^0$ for all $u \in U$.

To prove item 2, let u, u' be two distinct elements of U. There is i such that $u_i \neq u'_i$. Therefore $w_i^{j_i \leftarrow u_i} \neq w_i^{j_i \leftarrow u'_i}$ and $w^{j \leftarrow u} \neq w^{j \leftarrow u'}$.

We now prove item 3. Let T' be the conjunction of all the variables that appear in T that are not of the form y_{i,j_i}. Then $T = T' \wedge T^j$. Since $T(w) = 1$, we have $T'(w) = 1$. Since the entries of $w^{j \leftarrow a}$ are equal to those in w on all the variables that are not of the form y_{i,j_i}, we have $T'(w^{j \leftarrow a}) = 1$. Therefore, $T(w^{j \leftarrow a}) = T'(w^{j \leftarrow a}) \wedge T^j(w_{1,j_1}^{j \leftarrow a}, \ldots, w_{n,j_n}^{j \leftarrow a}) = T^j(a)$. ◀

We now give a different way of sampling according to the distribution $\mathcal{D}_\ell$. The proof is in Appendix B.

▶ **Fact 5.** *Let $\mathcal{S}$ be a Set-Cover instance. The following is an equivalent way of sampling from $\mathcal{D}_\ell$.*
1. *Draw $\xi \in \{0,1\}$ u.a.r.*[8]
2. *Draw $w \in \Delta_n^1$ u.a.r.*
3. *If $\xi = 1$ then output $y = w$.*
4. *If $\xi = 0$ then*
   a. *draw $j \in \text{one}(w)$ u.a.r.*
   b. *draw $v \in w^{j \leftarrow U}$ u.a.r.*
   c. *output $y = v$.*

In particular, for any event X,

$$\Pr_{\boldsymbol{y} \sim \mathcal{U}(\Delta_n^0)}[X] = \Pr_{\boldsymbol{w} \sim \mathcal{U}(\Delta_n^1), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{w})), \boldsymbol{y} \sim \mathcal{U}(\boldsymbol{w}^{\boldsymbol{j} \leftarrow U})}[X].$$

3 Main Lemma

In this section, we prove

▶ **Lemma 5.** *Let $\mathcal{S} = (S, U, E)$ be a set cover instance. If $F : (\{0,1\}^\ell)^n \to \{0,1\}$ is a DNF of size $|F| < 2^{\text{opt}(\mathcal{S})\ell/20}$, then $\text{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) \geq 1/(8|U|) - 2^{-\text{opt}(\mathcal{S})\ell/20}$.*

Note that Lemma 5 is used to prove Theorem 1 and 2. To prove Theorem 3, we will need Lemma 11, a stronger version of Lemma 5.

To prove the lemma, we first establish some results.

For a term T, let $T_{\mathcal{M}}$ be the conjunction of all the unnegated variables in T. We define the *monotone size* of T to be $|T_{\mathcal{M}}|$. The proof of the following Claim is in Appendix B.

▷ Claim 6. Let $\mathcal{S} = (S, U, E)$ be a set cover instance and $\ell \geq 5$. If $F : (\{0,1\}^\ell)^n \to \{0,1\}$ is a DNF of size $|F| < 2^{\text{opt}(\mathcal{S})\ell/20}$, then there is a DNF, F', of size $|F'| \leq 2^{\text{opt}(\mathcal{S})\ell/20}$ with terms of monotone size at most $\text{opt}(\mathcal{S})\ell/5$ such that $\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') \leq \text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F) + 2^{-\text{opt}(\mathcal{S})\ell/20}$.

[8] Uniformly at random.

We now prove

▷ Claim 7. Let $z \in \Delta_n^1$. Let F be a DNF with terms of monotone size at most $\lceil \ell/2 \rceil (\text{opt}(\mathcal{S}) - 1)/2$ that satisfies $F(z) = 1$. Then

$$\Pr_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z)), \boldsymbol{y} \sim \mathcal{U}(z^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1] \geq \frac{1}{2|U|}.$$

Proof. Since $F(z) = 1$, there is a term T in F that satisfies $T(z) = 1$. Let $Y_0 = \{y_{i,m} | z_{i,m} = 0\}$ and $Y_1 = \{y_{i,m} | z_{i,m} = 1\}$. Since $T(z) = 1$, every variable in Y_0 that appears in T must be negated, and every variable in Y_1 that appears in T must be unnegated. For $j \in \text{one}(z)$, define $q(j)$ to be the number of variables in $\{y_{1,j_1}, \ldots, y_{n,j_n}\}$ that appear in $T(y)$. All those variables appear unnegated in T because $j \in \text{one}(z)$. Recall that $T_{\mathcal{M}}$ is the conjunction of all unnegated variables in T. Then $|T_{\mathcal{M}}| \leq \lceil \ell/2 \rceil (\text{opt}(\mathcal{S}) - 1)/2$. Each variable in $T_{\mathcal{M}}$ contributes $\lceil \ell/2 \rceil^{n-1}$ to the sum $\sum_{j \in one(z)} q(j)$ and $|\text{one}(z)| = \lceil \ell/2 \rceil^n$. Therefore,

$$\mathop{\mathbb{E}}_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z))}[q(\boldsymbol{j})] = \frac{|T_{\mathcal{M}}|}{\lceil \ell/2 \rceil} \leq \frac{\text{opt}(\mathcal{S}) - 1}{2}.$$

By Markov's bound, at least half the elements $j \in \text{one}(z)$ satisfies $q(j) \leq \text{opt}(\mathcal{S}) - 1$. Let $J = \{j \in \text{one}(z) | q(j) < \text{opt}(\mathcal{S})\}$. Then $\Pr_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z))}[\boldsymbol{j} \in J] \geq 1/2$. Consider $j \in J$ and let T^j be the conjunction of all the variables that appear in T of the form y_{i,j_i}. Then $|T^j| = q(j) \leq \text{opt}(\mathcal{S}) - 1$. By Fact 1, there is $u \in U$ such that $T^j(u) = 1$. By Fact 4, we have $T(z^{j \leftarrow u}) = T^j(u) = 1$. Then $F(z^{j \leftarrow u}) = 1$. Since by item 1 in Fact 4, $|z^{j \leftarrow U}| = |U|$, we have

$$\Pr_{\boldsymbol{j} \sim \mathcal{U}(one(z)), \boldsymbol{y} \sim \mathcal{U}(z^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1 | \boldsymbol{j} \in J] \geq \frac{1}{|U|}.$$

Therefore,

$$\Pr_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z)), \boldsymbol{y} \sim \mathcal{U}(z^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1] \geq \Pr_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z))}[\boldsymbol{j} \in J] \times \Pr_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z)), \boldsymbol{y} \sim \mathcal{U}(z^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1 | \boldsymbol{j} \in J] \geq \frac{1}{2|U|}.$$ ◁

We are now ready to prove Lemma 5

Proof. By Claim 6, there is a DNF, F', of size $|F'| \leq 2^{\text{opt}(\mathcal{S})\ell/20}$ with terms of monotone size at most $\text{opt}(\mathcal{S})\ell/5$ such that $\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') \leq \text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F) + 2^{-\text{opt}(\mathcal{S})\ell/20}$. Therefore, it is enough to prove that $\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') \geq 1/(8|U|)$.

If $\Pr_{\boldsymbol{y} \sim \mathcal{U}(\Delta_n^1)}[F'(\boldsymbol{y}) \neq 1] \geq 1/(4|U|)$, then by Fact 2, for the event $Y(\boldsymbol{y}) = [\Gamma_\ell(\boldsymbol{y}) = 1]$, we have

$$\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') \geq \Pr_{\boldsymbol{y} \sim \mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) \neq F'(\boldsymbol{y}) | Y(\boldsymbol{y})] \Pr_{\boldsymbol{y} \sim \mathcal{D}_\ell}[Y(\boldsymbol{y})] = \frac{1}{2} \Pr_{\boldsymbol{y} \sim \mathcal{U}(\Delta_n^1)}[F'(\boldsymbol{y}) \neq 1] \geq \frac{1}{8|U|}.$$

If $\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[F'(\boldsymbol{y}) \neq 1] < 1/(4|U|)$, then by Fact 2 and 5, and Claim 7,

$$\begin{aligned}
\mathrm{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') &\geq \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) \neq F'(\boldsymbol{y}) | \Gamma_\ell(\boldsymbol{y}) = 0] \cdot \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) = 0] \\
&= \frac{1}{2} \Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^0)}[F'(\boldsymbol{y}) = 1] \\
&= \frac{1}{2} \Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1), \boldsymbol{j}\sim\mathcal{U}(\mathrm{one}(\boldsymbol{z})), \boldsymbol{y}\sim\mathcal{U}(\boldsymbol{z}^{\boldsymbol{j}\leftarrow U})}[F'(\boldsymbol{y}) = 1] \\
&\geq \frac{1}{2} \Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1), \boldsymbol{j}\sim\mathcal{U}(\mathrm{one}(\boldsymbol{z})), \boldsymbol{y}\sim\mathcal{U}(\boldsymbol{z}^{\boldsymbol{j}\leftarrow U})}[F'(\boldsymbol{y}) = 1 | F'(\boldsymbol{z}) = 1] \times \\
&\quad \Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1)}[F'(\boldsymbol{z}) = 1] \\
&\geq \frac{1}{2}\frac{1}{2|U|}\left(1 - \frac{1}{4|U|}\right) \geq \frac{1}{8|U|}.
\end{aligned}$$

◀

4 Superpolynomial Lower Bound

In this section, we prove the first results of the paper. First, we prove the following result for Monotone $(\log n)$-Junta.

▶ **Lemma 8.** *Assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable Monotone $(\log n)$-Junta by DNF with $\epsilon = 1/(16n)$ must take at least $n^{c\frac{\log\log n}{\log\log\log n}}$ time.*

The lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and can compute the target on all the points of the support of the distribution in polynomial time.

Proof. Consider the constant λ in Lemma 2. Let $c = \min(1/40, \lambda/4)$. Suppose there is a PAC learning algorithm $\mathcal{A}$ for Monotone $(\log n)$-Junta by DNF with $\epsilon = 1/(16n)$ that runs in time $n^{c\frac{\log\log n}{\log\log\log n}}$. We show that there is k such that for

$$k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k},$$

(k, k')-Set-Cover can be solved in time $N^{4ck} \leq N^{\lambda k}$. By Lemma 2, the result then follows.

Let $\mathcal{S} = (S, U, E)$ be an N-vertex (k, k')-Set-Cover instance where

$$k = \frac{1}{2}\frac{\log\log N}{\log\log\log N} \text{ and } k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k}.$$

Let $\ell = \frac{\log N}{k}$ and consider Γ_ℓ and $\mathcal{D}_\ell$.

Consider the following algorithm $\mathcal{B}$

1. Input $\mathcal{S} = (S, U, E)$ an instance for (k, k')-Set-Cover .
2. Construct Γ_ℓ and $\mathcal{D}_\ell$.
3. Run $\mathcal{A}$ using Γ_ℓ and $\mathcal{D}_\ell$. If it runs more than N^{4ck} steps, then output No .
4. Let F be the output DNF.
5. Estimate $\eta = \mathrm{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell)$.
6. If $\eta \leq \frac{1}{16N}$, output Yes , otherwise output No .

The running time of this algorithm is $N^{4ck} \leq N^{\lambda k}$. Therefore, it is enough to prove the following

▷ Claim 9. Algorithm $\mathcal{B}$ solves (k,k')-Set-Cover .

Proof. Yes case: Let $\mathcal{S}=(S,U,E)$ be a (k,k')-Set-Cover instance and $\text{opt}(\mathcal{S})\le k$. Then, $\text{opt}(\mathcal{S})\cdot\ell\le k\ell=\log N$, and by Fact 3, Γ_ℓ is Monotone $\log N$-Junta. Therefore, w.h.p., algorithm $\mathcal{A}$ learns Γ_ℓ and outputs a DNF that is $\eta=1/(16N)$ close to the target with respect to $\mathcal{D}_\ell$. Since $\mathcal{B}$ terminates $\mathcal{A}$ after N^{4ck} time, we only need to prove that $\mathcal{A}$ runs at most N^{4ck} time.

The running time of $\mathcal{A}$ is $N^{c\frac{\log\log N}{\log\log\log N}}<N^{4ck}$.

No Case: Let $\mathcal{S}=(S,U,E)$ be a (k,k')-Set-Cover instance and $\text{opt}(\mathcal{S})>k'$. By Lemma 5, any DNF, F, of size $|F|<2^{k'\ell/20}$ satisfies $\text{dist}_{\mathcal{D}_\ell}(F,\Gamma_\ell)\ge 1/(8|U|)-2^{-k'\ell/20}$. First, we have

$$(2k)^{2k}=\left(\frac{\log\log N}{\log\log\log N}\right)^{\frac{\log\log N}{\log\log\log N}}<\frac{\log N}{\log\log N}.$$

Therefore, since $c\le 1/40$,

$$k'=\frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k}>\frac{1}{2}(2k)^2>80ck^2.$$

So $k'\ell/20>(k\ell)(4ck)$ and $2^{k'\ell/20}>(2^{k\ell})^{4ck}=N^{4ck}$. Now since the algorithm runs in time N^{4ck}, it cannot output a DNF F of size more than $N^{4ck}<2^{k'\ell/20}$, and by Lemma 5,

$$\text{dist}_{\mathcal{D}_\ell}(F,\Gamma_\ell)\ge\frac{1}{8|U|}-\frac{1}{N^{4ck}}\ge\frac{1}{9N}.$$

So it either runs more than N^{4ck} steps and then outputs No in step 3 or outputs a DNF with an error greater than $1/(9N)>1/(16N)$ and outputs No in step 6. ◁

Notice that the learning algorithm knows Γ_ℓ and $\mathcal{D}_\ell$. It is also clear from the definition of Γ_ℓ and $\mathcal{D}_\ell$ that the learning algorithm can draw a sample according to the distribution $\mathcal{D}_\ell$ in polynomial time and can compute the target Γ_ℓ on all the points of the support of the distribution in polynomial time. ◀

We now prove

▶ **Theorem 1.** *Assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable size-s Monotone DT and size-s Monotone DNF by DNF with $\epsilon=1/(16n)$ must take at least*

$$n^{c\frac{\log\log s}{\log\log\log s}}$$

time.

The lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and can compute the target on all the points of the support of the distribution in polynomial time.

Proof. By Lemma 8, assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable Monotone $(\log n)$-Junta by DNF with $\epsilon=1/(16n)$ runs in time

$$n^{c\frac{\log\log n}{\log\log\log n}}.$$

Now by (1) and since $s=n$, the result follows. ◀

5 Tight Bound Assuming some Conjecture

A plausible conjecture on the hardness of Set-Cover is the following.

▶ **Conjecture 1** ([10]). *There are constants $\alpha, \beta, \lambda \in (0,1)$ such that, for $k < N^{\alpha}$, there is no randomized $N^{\lambda k}$ time algorithm that can solve $(k, (1-\beta)\cdot k \ln N)$-Set-Cover on N vertices with high probability.*

We now prove

▶ **Theorem 2.** *Assuming Conjecture 1, there is a constant c such that any PAC learning algorithm for n-variable Monotone $(\log s)$-Junta, size-s Monotone DT and size-s Monotone DNF by DNF with $\epsilon = 1/(16n)$ must take at least*

$$n^{c \log s}$$

time.

The lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and can compute the target on all the points of the support of the distribution in polynomial time.

Proof. We give the proof for Monotone $(\log s)$-Junta. As in the proof of Theorem 1, the result then follows for the other classes.

Consider the constants α, β and λ in Conjecture 1. Let $c = \min(\lambda/10, (1-\beta)/(20 \log e))$. Suppose there is a PAC learning algorithm $\mathcal{A}$ for Monotone $(\log s)$-Junta by DNF with $\epsilon = 1/(16n)$ that runs in time $n^{c \log s}$. We show that there is $k < N^{\alpha}$, $k = \omega(1)$, such that (k, k')-Set-Cover can be solved in time $N^{\lambda k}$ where $k' = (1-\beta)k \ln N$. By Conjecture 1, the result then follows.

Consider the following algorithm $\mathcal{B}$

1. Input $\mathcal{S} = (S, U, E)$ an instance for (k, k')-Set-Cover .
2. Construct Γ_5 and $\mathcal{D}_5$.
3. Run $\mathcal{A}$ using Γ_5 and $\mathcal{D}_5$ with $s = 2^{5k}$. If it runs more than N^{5ck} steps, then output No .
4. Let F be the output DNF.
5. Estimate $\eta = \mathrm{dist}_{\mathcal{D}_5}(F, \Gamma_5)$.
6. If $\eta \leq \frac{1}{16N}$, output Yes , otherwise output No .

Since $c < \lambda/10$, the running time of this algorithm is $N^{5ck} < N^{\lambda k}$. Therefore, it is enough to prove the following

▷ Claim 10. Algorithm $\mathcal{B}$ solves (k, k')-Set-Cover .

Proof. Yes case: Let $\mathcal{S} = (S, U, E)$ be a (k, k')-Set-Cover instance and $\mathrm{opt}(\mathcal{S}) \leq k$. Then, $5 \cdot \mathrm{opt}(\mathcal{S}) \leq 5k = \log s$, and by Fact 3, Γ_5 is Monotone $\log s$-Junta. Therefore, w.h.p., algorithm $\mathcal{A}$ learns Γ_5 and outputs a DNF that is $\eta = 1/(16N)$ close to the target with respect to $\mathcal{D}_5$. Since $\mathcal{B}$ terminates $\mathcal{A}$ after N^{5ck} time, we only need to prove that $\mathcal{A}$ runs at most N^{5ck} time.

The running time of $\mathcal{A}$ is

$$n^{c \log s} \leq N^{5ck}.$$

No Case: Let $\mathcal{S} = (S, U, E)$ be a (k, k')-Set-Cover instance and $\mathrm{opt}(\mathcal{S}) > k' = (1-\beta)k \ln N$. By Lemma 5, any DNF, F, of size $|F| < 2^{k'/4}$ satisfies $\mathrm{dist}_{\mathcal{D}_5}(F, \Gamma_5) \geq 1/(8|U|) - 2^{-k'/4}$. Since, $c < (1-\beta)/(20 \log e)$,

$$2^{k'/4} = 2^{\frac{(1-\beta)k \ln N}{4}} = N^{\frac{(1-\beta)k}{4 \log e}} > N^{5ck},$$

any DNF, F, that the learning outputs satisfies

$$\text{dist}_{\mathcal{D}_5}(F, \Gamma_5) \geq \frac{1}{8|U|} - 2^{-k'/4} \geq \frac{1}{8N} - \frac{1}{N^{5ck}} \geq \frac{1}{9N}.$$

Therefore, with high probability the algorithm answer NO . ◁

◀

6 Strictly Proper Learning

In this section, we prove

▶ **Theorem 3.** *Assuming randomized ETH, there is a constant c such that any PAC learning algorithm for n-variable* MONOTONE $(\log s)$-JUNTA*, size-s* MONOTONE DT *and size-s* MONOTONE DNF *by size-s DNF with $\epsilon = 1/(16n)$ must take at least*

$$n^{c \log s}$$

time.

The lower bound holds, even if the learner knows the distribution, can draw a sample according to the distribution in polynomial time and compute the target on all the points of the support of the distribution in polynomial time.

We first prove the following stronger version of Lemma 5

▶ **Lemma 11.** *Let $\mathcal{S} = (S, U, E)$ be a set cover instance, and let $\ell \geq 5$. If $F : (\{0,1\}^\ell)^n \to \{0,1\}$ is a DNF of size $|F| < 2^{\text{opt}(\mathcal{S})\ell/16}$, then $\text{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) \geq 1/(8|U|)$.*

To prove this lemma, we will give some more results.

Recall that, for a term T, $T_{\mathcal{M}}$ is the conjunction of all the unnegated variables in T. We define the *monotone size* of T to be $|T_{\mathcal{M}}|$. For a DNF $F = T_1 \vee T_2 \vee \cdots \vee T_s$ and $z \in (\{0,1\}^\ell)^n$, we define the *monotone width* of z in F as

$$\text{mwidth}_F(z) := \begin{cases} \min_{T_i(z)=1} |(T_i)_{\mathcal{M}}| & F(z) = 1 \\ 0 & F(z) = 0 \end{cases}.$$

We define $F^{-1}(1) = \{z | F(z) = 1\}$ and

$$\Omega = \Delta_n^1 \cap F^{-1}(1).$$

▷ Claim 12. Let F be a DNF with

$$\mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{U}(\Omega)} [\text{mwidth}_F(\boldsymbol{z})] \leq \text{opt}(S) \cdot \ell/4.$$

Then

$$\Pr_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{z})), \boldsymbol{y} \sim \mathcal{U}(\boldsymbol{z}^{\boldsymbol{j} \leftarrow U})} [F(\boldsymbol{y}) = 1] \geq \frac{1}{2|U|}.$$

Proof. Let $z \in \Omega$. Then $F(z) = 1$ and $z \in \Delta_n^1$. Let T^z be the term in F with $|T^z_{\mathcal{M}}| = \text{mwidth}_F(z)$ that satisfies $T^z(z) = 1$. Let $Y_0 = \{y_{i,m} | z_{i,m} = 0\}$ and $Y_1 = \{y_{i,m} | z_{i,m} = 1\}$. Since $T^z(z) = 1$, every variable in Y_0 that appears in T^z must be negated, and every variable

in Y_1 that appears in T^z must be unnegated. For $j \in \text{one}(z)$, define $q_z(j)$ to be the number of variables in $\{y_{1,j_1}, \ldots, y_{n,j_n}\}$ that appear in $T^z(y)$. All those variables appear unnegated in T because $j \in \text{one}(z)$. Each variable in $T^z_{\mathcal{M}}$ contributes $\lceil \ell/2 \rceil^{n-1}$ to the sum $\sum_{j \in one(z)} q_z(j)$ and $|\text{one}(z)| = \lceil \ell/2 \rceil^n$. Therefore,

$$\mathop{\mathbb{E}}_{\boldsymbol{j} \sim \mathcal{U}(\text{one}(z))}[q_z(\boldsymbol{j})] = \frac{|T^z_{\mathcal{M}}|}{\lceil \ell/2 \rceil} = \frac{\text{mwidth}_F(z)}{\lceil \ell/2 \rceil}.$$

Now,

$$\begin{aligned} \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{z}))}[q_{\boldsymbol{z}}(\boldsymbol{j})] &= \frac{\mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{U}(\Omega)}[\text{mwidth}_F(\boldsymbol{z})]}{\lceil \ell/2 \rceil} \\ &\leq \frac{\text{opt}(\mathcal{S})}{2}. \end{aligned}$$

By Markov's bound,

$$\Pr_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{z}))}[q_{\boldsymbol{z}}(\boldsymbol{j}) < \text{opt}(\mathcal{S})] \geq \frac{1}{2}.$$

Suppose for some $z \in \Omega$ and $j \in \text{one}(z)$, we have $q_z(j) < \text{opt}(\mathcal{S})$. Let T^j be the conjunction of all the variables that appear in $T^z_{\mathcal{M}}$ of the form y_{i,j_i}. Then $|T^j| = q_z(j) < \text{opt}(\mathcal{S})$. By Fact 1, there is $u \in U$ such that $T^j(u) = 1$. By Fact 4, we have $T^z(z^{j \leftarrow u}) = T^j(u) = 1$. Then $F(z^{j \leftarrow u}) = 1$. Since by item 1 in Fact 4, $|z^{j \leftarrow U}| = |U|$, we have

$$\Pr_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(one(\boldsymbol{z})), \boldsymbol{y} \sim \mathcal{U}(\boldsymbol{z}^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1 | q_{\boldsymbol{z}}(\boldsymbol{j}) < \text{opt}(\mathcal{S})] \geq \frac{1}{|U|}.$$

Therefore, for $\mathcal{D}'(\boldsymbol{y}) = [\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{z})), \boldsymbol{y} \sim \mathcal{U}(\boldsymbol{z}^{\boldsymbol{j} \leftarrow U})]$

$$\begin{aligned} \Pr_{\mathcal{D}'}[F(\boldsymbol{y}) = 1] \geq & \Pr_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(\text{one}(\boldsymbol{z}))}[q_{\boldsymbol{z}}(\boldsymbol{j}) < \text{opt}(\mathcal{S})] \cdot \\ & \Pr_{\boldsymbol{z} \sim \mathcal{U}(\Omega), \boldsymbol{j} \sim \mathcal{U}(one(\boldsymbol{z})), \boldsymbol{y} \sim \mathcal{U}(\boldsymbol{z}^{\boldsymbol{j} \leftarrow U})}[F(\boldsymbol{y}) = 1 | q_{\boldsymbol{z}}(\boldsymbol{j}) < \text{opt}(\mathcal{S})] \\ \geq & \frac{1}{2|U|}. \end{aligned}$$

◁

▷ Claim 13. Let $\mathcal{S} = (S, U, E)$ be a set cover instance, and let $\ell \geq 5$. If $F : (\{0,1\}^\ell)^n \to \{0,1\}$ is a DNF and $\mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{U}(\Omega)}[\text{mwidth}_F(\boldsymbol{z})] \leq \text{opt}(\mathcal{S})\ell/4$, then $\text{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) \geq 1/(8|U|)$.

Proof. If $\Pr_{\boldsymbol{y} \sim \mathcal{U}(\Delta^1_n)}[F(\boldsymbol{y}) \neq 1] \geq 1/(4|U|)$, then by Fact 2, for the event $Y(\boldsymbol{y}) = [\Gamma_\ell(\boldsymbol{y}) = 1]$, we have

$$\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F) \geq \Pr_{\boldsymbol{y} \sim \mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y}) \neq F(\boldsymbol{y}) | Y(\boldsymbol{y})] \Pr_{\boldsymbol{y} \sim \mathcal{D}_\ell}[Y(\boldsymbol{y})] = \frac{1}{2} \Pr_{\boldsymbol{y} \sim \mathcal{U}(\Delta^1_n)}[F(\boldsymbol{y}) \neq 1] \geq \frac{1}{8|U|}.$$

If $\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{y})\neq 1] < 1/(4|U|)$, then by Fact 2 and 5, and Claim 12,

$$\begin{aligned}
\mathrm{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F) &\geq \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y})\neq F(\boldsymbol{y})|\Gamma_\ell(\boldsymbol{y})=0]\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\Gamma_\ell(\boldsymbol{y})=0]\\
&= \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^0)}[F(\boldsymbol{y})=1]\\
&= \frac{1}{2}\Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1),\boldsymbol{j}\sim\mathcal{U}(\mathrm{one}(\boldsymbol{z})),\boldsymbol{y}\sim\mathcal{U}(\boldsymbol{z}^{\boldsymbol{j}\leftarrow U})}[F(\boldsymbol{y})=1]\\
&\geq \frac{1}{2}\Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1),\boldsymbol{j}\sim\mathcal{U}(\mathrm{one}(\boldsymbol{z})),\boldsymbol{y}\sim\mathcal{U}(\boldsymbol{z}^{\boldsymbol{j}\leftarrow U})}[F(\boldsymbol{y})=1|F(\boldsymbol{z})=1]\cdot\Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{z})=1]\\
&= \frac{1}{2}\Pr_{\boldsymbol{z}\sim\mathcal{U}(\Omega),\boldsymbol{j}\sim\mathcal{U}(\mathrm{one}(\boldsymbol{z})),\boldsymbol{y}\sim\mathcal{U}(\boldsymbol{z}^{\boldsymbol{j}\leftarrow U})}[F(\boldsymbol{y})=1]\cdot\Pr_{\boldsymbol{z}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{z})=1]\\
&\geq \frac{1}{2}\frac{1}{2|U|}\left(1-\frac{1}{4|U|}\right)\geq\frac{1}{8|U|}.
\end{aligned}$$

◁

▷ Claim 14. Let F be a size-s DNF formula for $s\geq 2$ such that $\mathrm{dist}_{\mathcal{D}_\ell}(F,\Gamma_\ell)\leq 1/4$, then

$$\mathbb{E}_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[\mathrm{mwidth}_F(\boldsymbol{y})]\leq 4\log s.$$

Proof. First, we have

$$\begin{aligned}
\frac{3}{4} &\leq \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F(\boldsymbol{y})=\Gamma_\ell(\boldsymbol{y})]\\
&= \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F(\boldsymbol{y})=\Gamma_\ell(\boldsymbol{y})|\Gamma_\ell(\boldsymbol{y})=1]+\frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F(\boldsymbol{y})=\Gamma_\ell(\boldsymbol{y})|\Gamma_\ell(\boldsymbol{y})=0]\\
&\leq \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{y})=1]+\frac{1}{2}.
\end{aligned}$$

Therefore, $\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{y})=1]\geq 1/2$.

Let $F=T_1\vee T_2\vee\cdots\vee T_s$. For $y\in\Omega$, let $\omega(y)\in[s]$ be the minimum integer such that $\mathrm{mwidth}_F(y)=|(T_{\omega(y)})_{\mathcal{M}}|$ and $T_{\omega(y)}(y)=1$.

Then, by (6),

$$\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[T_i(\boldsymbol{y})=1]=\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[T_i(\boldsymbol{y})=1|F(\boldsymbol{y})=1]=\frac{\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[T_i(\boldsymbol{y})=1]}{\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[F(\boldsymbol{y})=1]}\leq 2^{-|(T_i)_{\mathcal{M}}|/2+1}.$$

Now, by the concavity of log,

$$\begin{aligned}
\frac{1}{2}\mathbb{E}_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[\mathrm{mwidth}_F(\boldsymbol{y})]-1 &= \mathbb{E}_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}\left[\log\left(2^{\mathrm{mwidth}_F(\boldsymbol{y})/2-1}\right)\right]\\
&\leq \log\left(\mathbb{E}_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}\left[2^{\mathrm{mwidth}_F(\boldsymbol{y})/2-1}\right]\right)\\
&= \log\left(\sum_{i\in[s]}2^{|(T_i)_{\mathcal{M}}|/2-1}\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[\omega(\boldsymbol{y})=i]\right)\\
&\leq \log\left(\sum_{i\in[s]}2^{|(T_i)_{\mathcal{M}}|/2-1}\Pr_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[T_i(\boldsymbol{y})=1]\right)\\
&\leq \log\left(\sum_{i\in[s]}2^{|(T_i)_{\mathcal{M}}|/2-1}2^{-|(T_i)_{\mathcal{M}}|/2+1}\right)\\
&= \log s.
\end{aligned}$$

Therefore, $\mathbb{E}_{\boldsymbol{y}\sim\mathcal{U}(\Omega)}[\mathrm{mwidth}_F(\boldsymbol{y})]\leq 4\log s$. ◁

We are now ready to prove Lemma 11

Proof. If $\mathrm{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) > 1/4$, then the result follows. Now suppose $\mathrm{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) \leq 1/4$. If $s = |F| < 2^{\mathrm{opt}(\mathcal{S})\ell/16}$, then by Claim 14, $\mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mathcal{U}(\Omega)}[\mathrm{mwidth}_F(\boldsymbol{y})] \leq 4 \log s = \mathrm{opt}(\mathcal{S})\ell/4$. Then by Claim 13, $\mathrm{dist}_{\mathcal{D}_\ell}(F, \Gamma_\ell) \geq 1/(8|U|)$. ◀

The proof of Theorem 3 is the same as the proof of Theorem 14 in [10]. We give the proof in Appendix C for completeness.

References

1 Michael Alekhnovich, Mark Braverman, Vitaly Feldman, Adam R. Klivans, and Toniann Pitassi. The complexity of properly learning simple concept classes. *J. Comput. Syst. Sci.*, 74(1):16–34, 2008. doi:10.1016/j.jcss.2007.04.011.

2 Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.

3 Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-sat: An isolation lemma for k-cnfs. *J. Comput. Syst. Sci.*, 74(3):386–393, 2008. doi:10.1016/j.jcss.2007.06.015.

4 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the tutte polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014. doi:10.1145/2635812.

5 Andrzej Ehrenfeucht and David Haussler. Learning decision trees from random examples. *Inf. Comput.*, 82(3):231–246, 1989. doi:10.1016/0890-5401(89)90001-1.

6 Thomas R. Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Inf. Comput.*, 126(2):114–122, 1996. doi:10.1006/inco.1996.0040.

7 Lisa Hellerstein, Devorah Kletenik, Linda Sellie, and Rocco A. Servedio. Tight bounds on proper equivalence query learning of DNF. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 31.1–31.18, 2012. URL: http://proceedings.mlr.press/v23/hellerstein12/hellerstein12.pdf.

8 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

9 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

10 Caleb Koch, Carmen Strassle, and Li-Yang Tan. Superpolynomial lower bounds for decision tree learning and testing. *CoRR*, abs/2210.06375, 2022. doi:10.48550/arXiv.2210.06375.

11 Bingkai Lin. A simple gap-producing reduction for the parameterized set cover problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 81:1–81:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.81.

12 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

13 Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

A Definitions

A.1 Concept Classes

For the lattice $\{0,1\}^n$, and $x, y \in \{0,1\}^n$, we define the partial order $x \leq y$ if $x_i \leq y_i$ for every i. When $x \leq y$ and $x \neq y$, we write $x < y$. If $x < y$, we say that x is *below* y, or y is *above* x. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if, for every $x \leq y$, we

have $f(x) \leq f(y)$. A *literal* is a variable or negated variable. A *term* is a conjunction ($\wedge$) of literals. A *clause* is a disjunction ($\vee$) of literals. A *monotone term* (resp. clause) is a conjunction (resp. disjunction) of unnegated variables. The *size* of a term T, $|T|$, is the number of literals in the term T. A DNF (resp. CNF) is a disjunction (resp. conjunction) of terms (resp. clauses). The *size* $|F|$ of a DNF (resp. CNF) F is the number of terms (resp. clauses) in F. A *monotone DNF* (resp. monotone CNF) is a DNF (resp. CNF) with monotone terms (resp. clauses).

We define the following classes

1. size-s DNF and size-s MONOTONE DNF are the classes of DNF and monotone DNF, respectively, of size at most s.
2. size s-DT and size-s MONOTONE DT are the classes of decision trees and monotone decision trees, respectively, with at most s leaves.
3. k-JUNTA and MONOTONE k-JUNTA are the classes of Boolean functions and monotone Boolean functions that depend on at most k variables.

It is well known that

$$\text{MONOTONE } (\log s)\text{-JUNTA} \subset \text{size-}s \text{ MONOTONE DT} \subset \text{size-}s \text{ MONOTONE DNF}.$$

B Proofs

B.1 Proof of Fact 5

Proof. Denote the above distribution by $\mathcal{D}'$. By Item 1 in Fact 4, if $w \in \Delta_n^1$ and $j \in \text{one}(w)$, then $w^{j\leftarrow U} \subseteq \Delta_n^0$. Therefore, for $z \in \Delta_n^1$, $\Pr_{\boldsymbol{y}\sim\mathcal{D}'}[\boldsymbol{y} = z|\boldsymbol{\xi} = 0] = 0$ and then

$$\Pr_{\boldsymbol{y}\sim\mathcal{D}'}[\boldsymbol{y} = z] = \Pr_{\boldsymbol{\xi}\sim\mathcal{U}(\{0,1\})}[\boldsymbol{\xi} = 1] \cdot \Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[\boldsymbol{y} = z] = \frac{1}{2|\Delta_n^1|} = \frac{1}{2|\Delta^1|^n}.$$

For $z \in \Delta_n^0$, suppose $z \in \Delta^{u_1} \times \cdots \times \Delta^{u_n}$ where $u \in U$. In the sampling according to $\mathcal{D}'$ and when $\xi = 0$, since for $j \in \text{one}(w)$, the elements of $w^{j\leftarrow U}$ are below w, we have $\Pr_{\boldsymbol{y}\sim\mathcal{D}'}[\boldsymbol{y} = z|\boldsymbol{w} \not> z] = 0$. Therefore,

$$\begin{aligned}\Pr_{\boldsymbol{y}\sim\mathcal{D}'}[\boldsymbol{y} = z] \quad = \quad & \Pr_{\boldsymbol{\xi}\sim\mathcal{U}(\{0,1\})}[\boldsymbol{\xi} = 0] \cdot \Pr_{\boldsymbol{w}\sim\mathcal{U}(\Delta_n^1)}[\boldsymbol{w} > z] \times \\ & \Pr_{\boldsymbol{j}\sim\mathcal{U}(one(\boldsymbol{w}))}[z \in \boldsymbol{w}^{\boldsymbol{j}\leftarrow U}|\boldsymbol{w} > z, \boldsymbol{w} \in \Delta_n^1] \times \\ & \Pr_{\boldsymbol{v}\sim\mathcal{U}(\boldsymbol{w}^{\boldsymbol{j}\leftarrow U})}[\boldsymbol{v} = z|z \in \boldsymbol{w}^{\boldsymbol{j}\leftarrow U}]. \end{aligned} \tag{2}$$

Now, since, for $x \in \Delta^0$, the number of elements in Δ^1 that are above x is $\lceil \ell/2 \rceil$, we have that the number of $w \in \Delta_n^1 = (\Delta^1)^n$ that are above $z \in \Delta^{u_1} \times \cdots \times \Delta^{u_n}$ is $\lceil \ell/2 \rceil^{n-\text{wt}(u)}$. Therefore,

$$\Pr_{\boldsymbol{w}\sim\mathcal{U}(\Delta_n^1)}[\boldsymbol{w} > z] = \frac{\lceil \ell/2 \rceil^{n-\text{wt}(u)}}{|\Delta_n^1|}. \tag{3}$$

Now let $w > z$ and $w \in \Delta_n^1$. Since for two different $u, u' \in U$, we have $\prod_{i=1}^n \Delta^{u_i}$ and $\prod_{i=1}^n \Delta^{u'_i}$ are disjoint sets, and since $z \in \Delta^{u_1} \times \cdots \times \Delta^{u_n}$, we have $z \in w^{j\leftarrow U}$ if and only if $z = w^{j\leftarrow u}$. Therefore, the number of elements $j \in \text{one}(w)$ that satisfy $z \in w^{j\leftarrow U}$ is the number of elements $j \in \text{one}(w)$ that satisfy $z = w^{j\leftarrow u}$. This is the number of elements $j \in \text{one}(w)$ that satisfies for every $u_i = 0$, $z_{i,j_i} = 0$. For a j u.a.r. and a fixed i where $u_i = 0$, the probability that z_i and w_i differ only in entry j_i is $1/\lceil \ell/2 \rceil$. Therefore,

$$\Pr_{\boldsymbol{j}\sim\mathcal{U}(one(\boldsymbol{w}))}[z \in \boldsymbol{w}^{\boldsymbol{j}\leftarrow U}|\boldsymbol{w} > z, \boldsymbol{w} \in \Delta_n^1] = \frac{1}{\lceil \ell/2 \rceil^{n-\text{wt}(u)}}. \tag{4}$$

Finally, by item 2 in Fact 4, since $|w^{j\leftarrow U}| = |U|$, we have

$$\Pr_{\boldsymbol{v}\sim\mathcal{U}(\boldsymbol{w}^{j\leftarrow U})}[\boldsymbol{v} = z | z \in \boldsymbol{w}^{j\leftarrow U}] = \frac{1}{|\boldsymbol{w}^{j\leftarrow U}|} = \frac{1}{|U|}. \tag{5}$$

By (2), (3), (4), and (5), we have

$$\Pr_{\boldsymbol{y}\sim\mathcal{D}'}[\boldsymbol{y} = z] = \frac{1}{2}\cdot\frac{\lceil \ell/2\rceil^{n-\mathrm{wt}(u)}}{|\Delta_n^1|}\cdot\frac{1}{\lceil \ell/2\rceil^{n-\mathrm{wt}(u)}}\cdot\frac{1}{|U|} = \frac{1}{2|U|\cdot|\Delta_n^1|} = \frac{1}{2|U|\cdot|\Delta^0|^n}. \qquad \blacktriangleleft$$

B.2 Proof of Claim 6

Proof. Let T be a term of monotone size at least $\mathrm{opt}(\mathcal{S})\ell/5$. Let b_i denote the number of unnegated variables of T of the form $y_{i,j}$ and let T_i be their conjunction. Then $T_{\mathcal{M}} = \wedge_{i=1}^n T_i$ and $\sum_{i=1}^n b_i = |T_{\mathcal{M}}| \geq \mathrm{opt}(\mathcal{S})\ell/5$. If, for some i, $b_i > \lceil \ell/2\rceil$, then the term T_i is zero on all $\Delta^0 \cup \Delta^1$, and therefore, T is zero on all $\Delta_n^0 \cup \Delta_n^1$. Thus, it can be just removed from F. So, we may assume that $b_i \leq \lceil \ell/2\rceil$ for all i. First,

$$\begin{aligned}
\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[T(\boldsymbol{y}) = 1 | \Gamma_\ell(\boldsymbol{y}) = 1] &= \Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[T(\boldsymbol{y}) = 1] \leq \Pr_{\boldsymbol{y}\sim\mathcal{U}(\Delta_n^1)}[T_{\mathcal{M}}(\boldsymbol{y}) = 1] \\
&= \prod_{i=1}^n \Pr_{\boldsymbol{y}_i\sim\mathcal{U}(\Delta^1)}[T_i(\boldsymbol{y}_i) = 1] \\
&= \prod_{i=1}^n \frac{\binom{\ell-b_i}{\lceil \ell/2\rceil - b_i}}{\binom{\ell}{\lceil \ell/2\rceil}} \\
&= \prod_{i=1}^n \left(1 - \frac{b_i}{\ell}\right)\left(1 - \frac{b_i}{\ell-1}\right)\cdots\left(1 - \frac{b_i}{\lceil \ell/2\rceil + 1}\right) \\
&\leq \prod_{i=1}^n \prod_{j=\lceil \ell/2\rceil+1}^{\ell} \exp(-b_i/j) = \prod_{i=1}^n \exp\left(-b_i \sum_{j=\lceil \ell/2\rceil+1}^{\ell} 1/j\right) \\
&= \exp\left(-|T_{\mathcal{M}}| \sum_{j=\lceil \ell/2\rceil+1}^{\ell} 1/j\right) \\
&\leq 2^{-|T_{\mathcal{M}}|/2} \leq 2^{-\mathrm{opt}(\mathcal{S})\ell/10}.
\end{aligned} \tag{6}$$

Let F' be the disjunction of all the terms in F of monotone size at most $\mathrm{opt}(\mathcal{S})\ell/5$. Let $T^{(1)}, \ldots, T^{(m)}$ be all the terms of monotone size greater than $\mathrm{opt}(\mathcal{S})\ell/5$ in F. Then, by (6) and the union bound,

$$\begin{aligned}
\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F(\boldsymbol{y}) \neq F'(\boldsymbol{y}) | \Gamma_\ell(\boldsymbol{y}) = 1] &\leq \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[\vee_{i=1}^m T^{(i)}(\boldsymbol{y}) = 1 | \Gamma_\ell(\boldsymbol{y}) = 1] \\
&\leq 2^{-\mathrm{opt}(\mathcal{S})\ell/10} m \leq 2^{-\mathrm{opt}(\mathcal{S})\ell/20}.
\end{aligned} \tag{7}$$

and (Here we abbreviate $F'(\boldsymbol{y}), F(\boldsymbol{y})$ and $\Gamma_\ell(\boldsymbol{y})$ by F', F and Γ_ℓ)

$$\begin{aligned}
\text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F') &= \Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F' \neq \Gamma_\ell] \\
&= \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F' \neq \Gamma_\ell|\Gamma_\ell = 1] + \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F' \neq \Gamma_\ell|\Gamma_\ell = 0] \qquad (8)\\
&= \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F' \neq F|\Gamma_\ell = 1] + \\
&\quad \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F \neq \Gamma_\ell|\Gamma_\ell = 1] + \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F' \neq \Gamma_\ell|\Gamma_\ell = 0] \qquad (9)\\
&\leq 2^{-\text{opt}(\mathcal{S})\ell/20} + \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F \neq \Gamma_\ell|\Gamma_\ell = 1] + \frac{1}{2}\Pr_{\boldsymbol{y}\sim\mathcal{D}_\ell}[F \neq \Gamma_\ell|\Gamma_\ell = 0] \qquad (10)\\
&= 2^{-\text{opt}(\mathcal{S})\ell/20} + \text{dist}_{\mathcal{D}_\ell}(\Gamma_\ell, F).
\end{aligned}$$

In (8), we used Fact 2. In (9), we used the probability triangle inequality. In (10), we used (7) and the fact that if $F'(\boldsymbol{y}) \neq 0$, then $F(\boldsymbol{y}) \neq 0$. ◀

C The proof of Theorem 3

Proof. Consider the constant λ in Lemma 2. Let $c = \lambda/6$. Suppose there is a PAC learning algorithm $\mathcal{A}$ for MONOTONE $(\log s)$-JUNTA by size-s DNF with $\epsilon = 1/(16n)$ that runs in time $n^{\text{clog}\, s}$. We show that there is k such that for

$$k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k},$$

(k, k')-SET-COVER can be solved in time $N^{5ck} \leq N^{\lambda k}$. By Lemma 2, the result then follows.

Let $\mathcal{S} = (S, U, E)$ be an N-vertex (k, k')-SET-COVER instance where

$$k = \frac{1}{2}\frac{\log\log N}{\log\log\log N} \text{ and } k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k}.$$

Consider the following algorithm $\mathcal{B}$

1. Input $\mathcal{S} = (S, U, E)$ an instance for (k, k')-SET-COVER .
2. Construct Γ_5 and $\mathcal{D}_5$.
3. Run $\mathcal{A}$ using Γ_5 and $\mathcal{D}_5$ with $s = 2^{5k}$ and $n = N$. If it runs more than N^{5ck} steps, then output NO .
4. Let F be the output DNF.
5. If $|F| > s$ then output NO .
6. Estimate $\eta = \text{dist}_{\mathcal{D}_5}(F, \Gamma_5)$.
7. If $\eta \leq \frac{1}{16N}$, output YES , otherwise output NO .

The running time of this algorithm is $N^{5ck} \leq N^{\lambda k}$. Therefore, it is enough to prove the following

▷ Claim 15. Algorithm $\mathcal{B}$ solves (k, k')-SET-COVER .

Proof. YES case: Let $\mathcal{S} = (S, U, E)$ be a (k, k')-SET-COVER instance and $\text{opt}(\mathcal{S}) \leq k$. Then, $size(\Gamma_5) \leq 2^{5\cdot\text{opt}(\mathcal{S})} \leq 2^{5k} = s$, and by Fact 3, Γ_5 is MONOTONE $\log s$-JUNTA. Therefore, w.h.p., algorithm $\mathcal{A}$ learns Γ_5 and outputs a DNF that is $\eta = 1/(16N)$ close to the target with respect to $\mathcal{D}_5$. Since $\mathcal{B}$ terminates $\mathcal{A}$ after N^{5ck} time, we only need to prove that $\mathcal{A}$ runs at most N^{5ck} time.

The running time of $\mathcal{A}$ is

$n^{c \log s} = N^{c \log s} \leq N^{5ck}$.

No Case: Let $\mathcal{S} = (S, U, E)$ be a (k, k')-Set-Cover instance and $\text{opt}(\mathcal{S}) > k'$. By Lemma 11, any DNF, F, of size $|F| < 2^{5k'/16}$ satisfies $\text{dist}_{\mathcal{D}_5}(F, \Gamma_5) \geq 1/(8|U|)$. First, we have, for large N

$$k' = \frac{1}{2}\left(\frac{\log N}{\log\log N}\right)^{1/k} > 32k.$$

Therefore, any DNF, F, of size $|F| < 2^{10k}$ satisfies $\text{dist}_{\mathcal{D}_5}(F, \Gamma_5) \geq 1/(8|U|)$.

We have $2^{10k} > s$. So, $\mathcal{B}$ either runs more than N^{5ck} steps and then outputs No in step 3 or outputs a DNF of size more than s and then outputs No in step 4 or outputs a DNF of size at most s with $\text{dist}_{\mathcal{D}_5}(F, \Gamma_5) \geq 1/(8|U|) > 1/(8N) > 1/(16N)$ and outputs No in step 6. ◁

◀