Electronic Colloquium on Computational Complexity, Report No. 44 (2022)

An Optimal Algorithm for Certifying Monotone Functions

Meghal Gupta *
Microsoft Research

Naren Sarayu Manoj†
Toyota Technological Institute Chicago

April 4, 2022

Abstract

Given query access to a monotone function $f\colon \{0,1\}^n \to \{0,1\}$ with certificate complexity $C(f)$ and an input $x^\star$, we design an algorithm that outputs a size-$C(f)$ subset of $x^\star$ certifying the value of $f(x^\star)$. Our algorithm makes $O\left(C(f) \cdot \log n\right)$ queries to f, which matches the information-theoretic lower bound for this problem and resolves the concrete open question posed in the STOC '22 paper of Blanc, Koch, Lange, and Tan [BKLT22].

We extend this result to an algorithm that finds a size-$2C(f)$ certificate for a real-valued monotone function with $O\left(C(f) \cdot \log n\right)$ queries. We also complement our algorithms with a hardness result, in which we show that finding the shortest possible certificate in $x^\star$ may require $\Omega\left(\binom{n}{C(f)}\right)$ queries in the worst case.

1 Introduction

Given a function $f\colon \{0,1\}^n \to \mathbb{D}$ for some output domain $\mathbb{D}$ and an input $x^\star$, is there a short proof for why $f(x^\star)$ takes on the value it does? This natural question motivates the notion of *certificate complexity* in complexity theory. Loosely speaking, a certificate for $f(x^\star) = y$ is a subset of the bits of $x^\star$ that "fixes" the value of $f(x^\star)$. In other words, every input x that agrees with $x^\star$ on the bits in the certificate will satisfy $f(x) = f(x^\star)$. Besides being a quantity of interest in complexity theory and in the analysis of Boolean functions, certificate complexity has a natural interpretation in the context of explainable AI. Here, the practitioner aims to find simple properties of a given input that explain a classifier's prediction on the input. We formalize the notion of a certificate in Definition 1.1.

Definition 1.1 (Certificate (see, e.g., [AB09])). *Let $x|_S$ denote the substring of x in the coordinates of S.*

For a function $f : \{0,1\}^n \to \mathbb{D}$ and an input $x^\star \in \{0,1\}^n$, we say a set $S \subseteq [n]$ is a certificate if for all $y \in \{0,1\}^n$ such that $x^\star|_S = y|_S$, we have $f(x^\star) = f(y)$.

We use Definition 1.1 to define the certificate complexity of a function f.

Definition 1.2 (Certificate complexity (see, e.g., [AB09])). *For any function $f\colon \{0,1\}^n \to \mathbb{D}$ and $x \in \{0,1\}^n$, we let $C(f,x)$ be the smallest integer such that there exists a $C(f,x)$-sized certificate for $f(x) = j$. We now let the certificate complexity of f be $\max_{x\in\{0,1\}^n} C(f,x)$.*

*E-mail: meghal@mit.edu
†E-mail: nsm@ttic.edu.

ISSN 1433-8092

A natural follow-up question from Definition 1.2 is whether a short certificate can be found in a given input if we know that all inputs have a short certificate. The following problem, posed and studied in the STOC '22 paper of [BKLT22], formalizes this question.

Problem. *Given queries to a function $f : \{0,1\}^n \to \{0,1\}$ with certificate complexity $C(f)$ and an input $x^\star$, output a size-$C(f)$ certificate for f's value on $x^\star$.*

The main result of [BKLT22] is an algorithm for the case where f is monotone and the output range $\mathbb{D} = \{0,1\}$. The authors design a randomized algorithm that makes at most $O\left(C(f)^8 \cdot \log n\right)$ queries using a novel connection to threshold phenomena. Furthermore, [BKLT22] show that $\Omega(C(f) \cdot \log n)$ queries for the certification problem are necessary in the worst-case. The authors identify closing this gap as a concrete direction for future work.

1.1 Our Results

Our main result is a simple, deterministic algorithm that makes $O(C(f) \cdot \log n)$ queries to find a size-$C(f)$ certificate for any monotone binary-valued function f and input $x^\star$. This completely resolves the aforementioned open question from [BKLT22]. Formally, we have Theorem 1.3.

Theorem 1.3. *Given query access to a monotone function $f\colon \{0,1\}^n \to \{0,1\}$ and an input $x^\star$, there exists an algorithm that makes $O(C(f) \cdot \log n)$ queries to f and outputs a size-$C(f)$ subset S corresponding to a subset of indices of $x^\star$ certifying the value of $f(x^\star)$.*

We can extend our result to obtain as a simple corollary an algorithm that finds a size-$2C(f)$ certificate for any monotone **real**-valued function f and input $x^\star$. Specifically, we have Theorem 1.4.

Theorem 1.4. *Given query access to a monotone function $f\colon \{0,1\}^n \to \mathbb{R}$ and an input $x^\star$,there exists an algorithm that makes $O(C(f) \cdot \log n)$ queries to f and outputs a size-$2C(f)$ subset S corresponding to a subset of indices of $x^\star$ certifying the value of $f(x^\star)$.*

The careful reader might also wonder why we are only looking for a certificate of size-$C(f)$ on the input $x^\star$ – by definition, the shortest certificate on a fixed input $x^\star$ is size-$C(f, x^\star)$. We show that finding a certificate of length $C(f, x^\star)$ may require far more queries than simply finding one of length $C(f)$. In particular, in the case where $C(f, x^\star) = n/2$, it may require exponentially many queries. Moreover, our result matches the trivial upper bound provided by an algorithm which simply queries all $C(f, x^\star)$-size certificates. See Theorem 1.5.

Theorem 1.5. *For any k, for any (randomized) algorithm that queries a given function, there exists a function f and input $x^\star$ such that $k = C(f, x^\star)$, and the algorithm must make at least $\frac{1}{2}\binom{n}{k}$ queries to determine a certificate with probability $> 1/2$.*

1.2 Related Work

We derive our setting and problem statements from the work of [BKLT22]. The authors of [BKLT22] formally propose the problem of certifying a monotone function f on an input $x^\star$ and provide an algorithm for doing so, as mentioned earlier. They also look at the certification question for a general (non-monotone) function f. Here, they show $\Omega\left(2^{C(f)} + C(f) \cdot \log n\right)$ queries are necessary, and

$O\left(2^{C(f)} \cdot C(f) \cdot \log n\right)$ queries suffice with high probability. Closing this gap remains an interesting open direction.

Before the work of [BKLT22], Angluin (see [Ang88]) gave a local search algorithm that can be used to certify a monotone function f on an input $x^\star$ with query complexity $O(n)$. See Appendix C in [BKLT22] for a detailed exposition and proof of correctness of Angluin's algorithm.

2 Preliminaries

Notation In this work, we use the following notation.

- We denote the set $\{x \in \mathbb{Z}_{\geq 0} : 1 \leq x \leq n\}$ as $[n]$. In an abuse of notation, let $[0] = \emptyset$.
- For a set $S \subseteq [n]$, we write x_S to be the indicator vector for S; i.e., x_S is such that $x_i = \mathbb{1}\{i \in S\}$, for all $i \in [n]$. Additionally, we write $x|_S$ to be the substring of x in the coordinates of S. Specifically, we have $x|_S = \{(i, x_i) \text{ for all } i \in S\}$.
- Let $\mathbb{1}^n$ denote the all-1s vector in n dimensions.
- For a vector $x \in \{0,1\}^n$, we denote S_x to be $\{i : x_i = 1\}$.

In our work, it is helpful to distinguish a *minimal* certificate from a general certificate.

Definition 2.1 (Minimal Certificate)**.** *For a given function f, we say a certificate $S \subseteq [n]$ is* minimal *if for all $a \in S$, we have that $S \setminus a$ is not a certificate for $f(x)$.*

If f is monotone, this is equivalent to requiring that for all $A \subset S$, we have $f(x|_A) \neq f(x|_S)$.

Finally, we note the information-theoretic lower bound from [BKLT22] on the query complexity of any algorithm used to certify $f(x)$ for a monotone function f.

Lemma 2.2 (Claim 1.2 in [BKLT22])**.** *For any $c < 1$ and any $k \leq l \leq n^c$, let $\mathcal{A}$ be an algorithm which, given query access to a monotone function $f\colon \{0,1\}^n \to \{0,1\}$ with certificate complexity $\leq k$ and an input $x^\star$, returns a size-l certificate for f's value on $x^\star$ with high probability. The query complexity of $\mathcal{A}$ must be $\Omega(k \log n)$.*

3 Our Algorithm to Certify a Binary Monotone Function

We first restate the problem.

Problem. *Given query access to a monotone function $f\colon \{0,1\}^n \to \{0,1\}$ with certificate complexity $C(f)$ and an input $x^\star$, output a size-$C(f)$ certificate for f's value on $x^\star$.*

3.1 Overview of Our Algorithm

We informally describe our algorithm. Without loss of generality, we let $x^\star$ be such that $f(x^\star) = 1$. A valid certificate is any subset $S \subset S_{x^\star}$ of indices such that $f(x_S) = 1$.

We add elements into our certificate A one-by-one. To do this, we simply iterate the following steps until A is a valid certificate:

1. Find the smallest $s \in S_{x^\star}$ such that including all $i \leq s \in S_{x^\star}$ in the certificate, along with elements already in A, yields a valid certificate.

2. Add s to A.

Because the function is monotone, s can be found through binary search at each step. Moreover, observe that removing any one element from A no longer yields a valid certificate; thus, as we will show in Lemma 3.3, the output certificate is length at most $C(f)$. This also implies the algorithm makes a total of $O(C(f) \cdot \log n)$ queries.

3.2 Formal Description of Our Algorithm

We state our algorithm formally. In our algorithm description and analysis, we assume without loss of generality that $f(x^\star) = 1$. We can make this assumption since if $f(x^\star) = 0$, we can instead run the algorithm making queries to $g(x) \coloneqq 1 - f(\mathbb{1}^n - x)$, which is a monotone function with $g(x^\star) = 1$.

Definition 3.1 (search)**.** *The procedure* $\mathsf{search}(f, A, S)$ *acts on a monotone function* $f \in \{0,1\}^n \to \{0,1\}$*, and two sets* $A, S \subseteq [n]$*. If* $f(x_A) = 1$ *or* $f(x_{A \cup S}) = 0$*, it outputs* ERROR*. Else, it outputs the smallest* $s \in S$ *for which* $f(x_{A \cup ([s] \cap S)}) = 1$*. The function proceeds using binary search, which can be done because* f *is monotone.*

Algorithm 1 : Algorithm to Certify a Binary Monotone Function Where $f(x^\star) = 1$

1. **Input:** Query access to a function $f\colon \{0,1\}^n \to \{0,1\}$ and point $x^\star$ for which $f(x^\star) = 1$.
2. Initialize the sets $A \leftarrow \emptyset$ and $S \leftarrow S_{x^\star}$
3. Run the following procedure until $f(x_A) = 1$:
   (a) Set $s \leftarrow \mathsf{search}(f, A, S)$.
   (b) Add s to A.
   (c) Set $S \leftarrow S \cap [s-1]$
4. **Output:** A.

3.3 Analysis

Theorem 3.2. *The Algorithm in 3.2 outputs a certificate of length at most* $C(f)$ *for* f *on* $x^\star$ *and makes at most* $O(C(f) \cdot \log n)$ *queries.*

We break the proof down into a series of lemmas.

Lemma 3.3. *If* S *is a minimal certificate, then* $|S| \le C(f)$*.*

Proof. Consider the shortest certificate C for the input x_S. We must have $f(x_C) = f(x_S)$, and $|C| \le C(f)$. The fact that $|C| < |S|$ and $f(x_C) = f(x_S)$ contradicts that S is minimal. ☐

Lemma 3.4. *The Algorithm in 3.2 never outputs* ERROR.

Proof. If the algorithm outputs ERROR, it must be in Step 3a. By definition, an error occurs if $f(x_A) = 1$ or $f(x_{A \cup S}) = 0$. The former cannot be true because the algorithm checks this exact condition in Step 3. The latter cannot be true because:

- If this is the first iteration of Step 3, $A \cup S = S_{x^\star}$, which means $f(x_{A \cup S_{x^\star}}) = f(x^\star) = 1$.
- Else, in the previous iteration of Step 3a (let the values of A, S, s at that step be A', S', s' respectively), it must have been the case that $f(x_{A' \cup (S' \cap [s'])}) = 1$. Note that $A' \cup (S' \cap [s']) = A \cup (S' \cap [s' - 1]) = A \cup S$, and so $f(x_{A \cup S}) = 1$.

□

Lemma 3.5. *If the Algorithm in 3.2 terminates, it outputs a minimal certificate for f on $x^\star$.*

Proof. It must be the case that $f(A) = 1$; otherwise, we could not have left Step 3. Consider any $s \in A$ and we will show that $f(A \setminus s) = 0$.

At the iteration of Step 3 where s was added to A (let the temporary certificate A at the start of that step be A_s), it must be the case that $f(x_{(S \cap [s]) \cup A_s}) = 1$ but $f(x_{(S \cap [s-1]) \cup A_s}) = 0$. All future elements that are added to create the final certificate A must be a subset of $S \cap [s-1]$ (where S is being referenced from the current iteration of Step 3). Therefore, $A \setminus S \subseteq (S \cap [s-1]) \cup A_s$, and therefore $f(A \setminus s) = 0$. □

Lemma 3.6. *The Algorithm in 3.2 terminates, making at most $O(C(f) \log n)$ queries.*

Proof. Observe that in every iteration of the main loop, we add exactly one element to A. By Lemma 3.3, there are at most $C(f)$ coordinates in the output A. Hence, we run the main loop at most $C(f)$ times.

Next, $\mathsf{search}(f, A, S)$ is a binary search over a domain of size $|S| \leq |S_{x^\star}| \leq n$. Therefore, $\mathsf{search}(f, A, S)$ uses at most $\log n$ queries.

Finally, the check $f(x^\star_A) = 1$ costs 1 query, and this runs at the beginning of every iteration of the loop. In total, we make at most $C(f) \cdot (\log n + 1)$ queries, as desired. □

Combining these lemmas finishes the proof of Theorem 3.2.

4 Extension to Real-Valued Functions

In this section, we prove the following corollary of our main result wherein the output domain is $\mathbb{R}$ instead of $\{0, 1\}$.

Corollary 4.1. *There exists an algorithm that, given an input $x^\star$ and query access to a monotone $f\colon \{0,1\}^n \to \mathbb{R}$, makes $O(C(f) \cdot \log n)$ queries to f and outputs a size-$2 \cdot C(f)$ certificate for $f(x^\star)$.*

4.1 Our Algorithm

We begin with two necessary definitions.

Definition 4.2 ($\mathsf{binary\_cert}(f, x^\star)$)**.** *The procedure $\mathsf{binary\_cert}(f, x^\star)$ is given query access to function $f : \{0,1\}^n \to \{0,1\}$ and an input $x^\star$, runs our algorithm from Section 3, and outputs a size-$C(f)$ certificate for $f(x^\star)$.*

Definition 4.3 ($g_{0,f,x^\star}(x)$, $g_{1,f,x^\star}(x)$)**.** *Let $b \in \{0,1\}$, $f : \{0,1\}^n \to \mathbb{R}$ and $x^\star \in \{0,1\}^n$. The function $g_{b,f,x^\star} : \{0,1\}^n \to \{0,1\}$ is defined as follows:*

$$g_b(x) = \begin{cases} 0 & f(x) < f(x^\star) \\ 1 & f(x) > f(x^\star) \\ b & f(x) = f(x^\star) \end{cases}$$

We will abbreviate $g_{b,f,x^\star} : \{0,1\}^n \to \{0,1\}$ as g_b when f and $x^\star$ are clear.

Algorithm 2 : Algorithm to Certify a Real-Valued Monotone Function

1. **Input:** $f, x^\star$.
2. Set $C_0 \leftarrow \mathsf{binary\_cert}(g_0, x^\star)$.
3. Set $C_1 \leftarrow \mathsf{binary\_cert}(g_1, x^\star)$.
4. **Output:** $C_0 \cup C_1$.

4.2 Analysis

Theorem 4.4. *The Algorithm in 4.1 outputs a certificate for f of length at most $2C(f)$ and makes at most $O(C(f) \log n)$ queries.*

We break the proof into a series of lemmas. Call the output A.

Lemma 4.5. *A is a valid certificate for f on $x^\star$.*

Proof. For any input y such that $y|_A = x^\star|_A$, we must have $g_b(y) = g_b(x^\star)$ for both $b = 0, 1$. Notice that both of the following hold:

$$g_0(y) = g_0(x^\star) \text{ implying } f(y) \leq f(x^\star)$$
$$g_1(y) = g_1(x^\star) \text{ implying } f(y) \geq f(x^\star)$$

Hence, we have $f(y) = f(x^\star)$. □

Lemma 4.6. *$|A| \leq 2C(f)$.*

Proof. It suffices to show that $C(f) \geq C(g_b)$ for $b \in \{0,1\}$. We will show that any certificate B for f on x is also a certificate for g_b. For all y, y' with $y|_C = y'|_C$, we have $f(y) = f(y')$, but this implies by definition that $g_b(y) = g_b(y')$. Hence, B is also a certificate for g_b. □

Combining these lemmas concludes the proof of Theorem 4.4.

5 Finding the Shortest Certificate for a Monotone Function

In this section, we show that there exists a family of instances on which the problem of finding the shortest certificate for a binary-valued f on an input $x^\star$ (denoted $k := C(f, x^\star)$) requires at least $\Omega\left(\binom{n}{k}\right)$ queries.

Notice that this result is essentially optimal: for any function f, any input $x^\star$ and $k = C(f, x^\star)$, $O\left(\binom{n}{k}\right)$ suffice to find a size-k certificate. Assuming $f(x^\star) = 1$, the algorithm can simply query $f(x_S)$ for all subsets S of size k and check if each one of them is a certificate.

Definition 5.1 (F_k)**.** *We define the set of k-indicator functions, denoted F_k as follows.*
Let $f_P\colon \{0,1\}^n \to \{0,1\}$ for some $P \subset [n]$ be defined as follows:

$$f_P(x) = \begin{cases} 0 & |S_x| < k \\ 1 & |S_x| > k \\ 0 & |S_x| = k, P \neq S_x \\ 1 & |S_x| = k, P = S_x \end{cases}$$

Finally, let $F_k = \{f_P \ : \ |P| = k\}$.

Lemma 5.2. *Every function $f \in F_k$ has $C(f, \mathbb{1}^n) = k$.*

Proof. It is easy to see that every function in F_k is monotone for all k.

Let $P \subset [n]$ be such that $f_P = f$. Observe that $|P| = k$. Next, notice that $f(\mathbb{1}^n_P) = f(\mathbb{1}^n) = 1$. This implies that $C(f, \mathbb{1}^n) \leq |P| = k$. Finally, consider any $S \subset [n]$ such that $|S| < k$. Note that for $x = \mathbb{1}^n_S$, we have $|S_x| < k$, so $f_P(x) = 0$. Thus, we have $C(f, \mathbb{1}^n) \geq k$, and we're done. □

Theorem 5.3. *For any $k \in [n-1]$ and (randomized) algorithm $\mathcal{A}$, there exists a function $f\colon \{0,1\}^n \to \{0,1\}$ and input $x^\star$ with $C(f, x^\star) = k$ such that $\mathcal{A}$ must make at least $1/2 \cdot \binom{n}{k}$ queries to f to find the size-k certificate with probability $\geq 1/2$.*

Proof. Fix an arbitrary $k \in [n-1]$. We will show that some function $f \in F_k$ takes $\geq 1/2 \cdot \binom{n}{k}$ queries to certify on the input $\mathbb{1}^n$. Note that $C(f, \mathbb{1}^n) = k$. Additionally, observe that any randomized algorithm to find a certificate for $f(\mathbb{1}^n) = 1$ can be converted to one that only makes queries x satisfying $|S_x| = k$.

Let $X = \{x \in \{0,1\}^n \ : \ |S_x| = k\}$. Notice that $|X| = \binom{n}{k}$. Any randomized algorithm for finding the single $x \in X$ such that $f(x) = 1$ can be viewed as one that samples a permutation from some distribution over permutations of X and makes queries to f in the order determined by the permutation until the algorithm encounters the $x \in X$ for which $f(x) = 1$. This is because query i only depends on the values of the queries $1, \ldots, i-1$, and not their responses – in particular, the responses to queries $1, \ldots, i-1$ are all 0 if the algorithm has not terminated prior to issuing query i. With this interpretation in mind, fix some distribution of permutations of X; call this distribution $\mathcal{P}$.

For each $x \in X$, consider $\Pr_{P \in \mathcal{P}}\left[P^{-1}(x) \leq |X|/2\right]$ where $P^{-1}(x)$ is the index of element X. Let $\mu(P)$ denote the probability that a random permutation drawn from $\mathcal{P}$ is P, and observe the following manipulations:

$$\begin{aligned}\sum_{x\in X}\Pr_{P\in\mathcal{P}}\left[P^{-1}(x)\leq |X|/2\right] &= \sum_{x\in X}\sum_{P}\mu(P)\cdot\mathbb{1}\left\{P^{-1}(x)\leq |X|/2\right\} \\ &= \sum_{P}\mu(P)\cdot\sum_{x\in X}\mathbb{1}\left\{P^{-1}(x)\leq |X|/2\right\} \\ &= \sum_{P}\mu(P)\cdot\frac{1}{2}\cdot|X| = \frac{1}{2}\cdot|X|\end{aligned}$$

Thus, there exists at least one $x \in X$ for which $\Pr_{P\in\mathcal{P}}\left[P^{-1}(x)\leq |X|/2\right] \leq 1/2$. It follows that the algorithm does not find a sized-k subset of $\mathbb{1}^n$ certifying $f(\mathbb{1}^n) = 1$ with probability $> \frac{1}{2}$ without making at least $\frac{1}{2}\binom{n}{k}$ queries. □

References

[Ang88] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009. ISBN: 0521424267.

[BKLT22] Guy Blanc, Caleb Koch, Jane Lange, and Li-Yang Tan. The query complexity of certification, 2022. DOI: 10.48550/ARXIV.2201.07736. URL: https://arxiv.org/abs/2201.07736.

[ODo21] Ryan O'Donnell. Analysis of boolean functions. *arXiv preprint arXiv:2105.10386*, 2021.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il