Sampling Algorithms: Lower Bounds and Applications *

Ziv Bar-Yossef†
Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
zivi@cs.berkeley.edu

Ravi Kumar
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
ravi@almaden.ibm.com

D. Sivakumar
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
siva@almaden.ibm.com

February 10, 2002

Abstract

We develop a framework to study probabilistic sampling algorithms that approximate general functions of the form $f : A^n \to B$, where A and B are arbitrary sets. Our goal is to obtain lower bounds on the query complexity of functions, namely the number of input variables x_i that any sampling algorithm needs to query to approximate $f(x_1, \ldots, x_n)$.

We define two quantitative properties of functions — the *block sensitivity* and the *minimum Hellinger distance* — that give us techniques to prove lower bounds on the query complexity. These techniques are quite general, easy to use, yet powerful enough to yield tight results. Our applications include the mean and higher statistical moments, the median and other selection functions, and the frequency moments, where we obtain lower bounds that are close to the corresponding upper bounds.

*An extended abstract of this paper appeared in the Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC), pages 266–275, 2001.

†Part of this work was done while the author was visiting IBM Almaden Research Center. Supported by NSF Grant CCR-9820897.

1 Introduction

The need for computing with massive data sets has sparked off much interest in novel computational paradigms. These include, but are not restricted to, algorithms that probe only small (random) portions of the data; algorithms that work by making a few passes over the data [HRR99]; algorithms that operate on a stream of data with limited space and stringent constraints on time per data item [HRR99, AMS99, FKSV99]. Algorithms of this nature are typically intended to compute sufficiently good approximate solutions to the problem at hand. Several algorithmic and complexity questions arise in the context of these computational paradigms.

In this paper, we focus on one such paradigm: algorithms that work by randomly sampling a few entries of a large input. We develop the first systematic complexity theory of probabilistic sampling algorithms that approximate general functions of the form $f : A^n \to B$, where A and B are arbitrary sets. A highlight of our work is the level of generality we are able to achieve toward our goal: we consider functions where the domain and range are not necessarily metric spaces, which makes the issue of "approximate" computation rather subtle; even when the domain and range have metric properties, there are many standard notions of approximation, e.g., relative vs. additive, that need to be encompassed. Also, we include algorithms that may perform non-oblivious (adaptive) sampling of input entries.

Main contributions.

From the standpoint of modeling, we present a single unified view of what it means to approximate a general function on a possibly non-metric domain and range. As our computational model, we show how the standard decision tree model may be suitably adapted. Our main results are two techniques for obtaining lower bounds on the query complexity for approximating broad classes of functions. A primary advantage of these techniques is that the lower bounds are stated in terms of certain quantitative properties of functions, which make them quite general and very easy to use. We also present (general and specific) upper bounds that establish the tightness of our lower bounds.

Lower bounds.

We present two techniques for obtaining lower bounds on the query complexity of randomized sampling algorithms that approximately compute functions. Our first lower bound technique is based on an adaptation of the notion of *block sensitivity* [Nis91]. The advantage of this technique is that it applies to *any* function, even though the lower bounds may be somewhat weak. Moreover, it applies to the *expected* query complexity, not just the worst case query complexity. Our second lower bound technique is based on the notion of *Hellinger distance* [LY90] between probability distributions, and yields much stronger bounds for symmetric functions. These results are described in Sections 3 and 4.

Using the method based on the Hellinger distance, we obtain (in many cases, optimal) lower bounds on the query complexity for several problems. These include approximate computations of the median, minimum, maximum, and other selection functions, the mean and higher statistical moments, and frequency moments F_k for $k \neq 1$. For the case of the mean, our lower bound matches the lower bound of Canetti, Even, and Goldreich [CEG95], and (modulo the machinery) is substantially simpler. This lower bound has another powerful consequence: it implies (a restricted version of) the main technical result of Radhakrishnan and Ta-Shma [RT00], which they use to obtain lower bounds on extractor, disperser, and superconcentrator parameters.

Our lower and upper bounds for the frequency moments have some interesting implications. (Recall that given $x \in [m]^n$, for $k \geq 0$, the k-th frequency moment $F_k(x)$ is defined as $\sum_{i=1}^{m} q_i^k$, where q_i is the number of times i appears in the sequence x.) While any oblivious sampling algorithm that uses s samples can be simulated by a streaming algorithm that uses space (roughly) s, we show that the converse is not true for F_2: the streaming algorithm of Alon, Matias, and Szegedy [AMS99] uses $O(\log m)$ space for approximating F_2, while we prove an $\Omega(\sqrt{m})$ lower bound for sampling algorithms. (This style of "separation" of the two models is also demonstrated in [FKSV00].) We then show that for $k \geq 2$, the k-th frequency moment can be approximated using $O(m^{1-\frac{1}{k}})$ samples, which immediately implies the space upper bound of [AMS99] for all $k > 2$. Finally, we provide a simple proof for the sampling lower bound of Charikar *et al.* [CCMN00] for F_0. These results are described in Section 5.

We also investigate the question of how tight our lower bound methodologies are. We obtain a general theorem, which shows that at least in some special cases (symmetric functions on bounded domains), the lower and upper bounds are quadratically related. In addition, we point out some limitations of our techniques. These results are described in Section 6.

In Section 7 we discuss related prior work and in particular the relationship of our model to the areas of Boolean decision tree complexity [Bd99], PAC and statistical learning Theory [Val84, KV94, Vap98], statistical decision theory [Ber85], statistical estimation theory [Van68], and statistical sequential analysis [Sie85]. Section 8 concludes with some open problems.

2 Preliminaries

In this section we introduce a notion of approximation for functions $f : A^n \to B$, where A and B are arbitrary sets. We then generalize Boolean decision trees to decision trees that approximately compute such functions. These decision trees will be our model for sampling algorithms.

2.1 Approximation Notions

An approximation for a function $f : A^n \to B$ is a function that maps inputs of f to subsets of its range B. This function assigns to every input x a set of values in B that are considered a good approximation for $f(x)$. It has two natural requirements, as specified in the following definition:

Definition 2.1 (Approximation) *An* approximation *for a function $f : A^n \to B$ is a family of functions $\{C_{f,\epsilon} : A^n \to 2^B\}_{\epsilon \geq 0}$, parameterized by an error parameter ϵ, which satisfies the following two conditions:*

(1) Containment of the truth: $\forall x \in A^n,\ C_{f,0}(x) = \{f(x)\}$

(2) Nesting: $\forall x \in A^n \; \forall \epsilon > \epsilon' \geq 0, \; C_{f,\epsilon}(x) \supseteq C_{f,\epsilon'}(x)$

We call $C_{f,\epsilon}(x)$ the ϵ-*approximation set* of the input x, and each value $z \in C_{f,\epsilon}(x)$ an ϵ-*approximation of* $f(x)$.

One might wonder why we define $C_{f,\epsilon}$ as a function of the inputs rather than as a function of the output values, i.e., why wouldn't $C_{f,\epsilon}$ determine for each value in the range what other values in the range are considered a good approximation for it? The rank and property testing approximations described below show that in some cases the approximation is indeed a function of the inputs: there can be two different inputs with the same output value but with different approximation sets.

Throughout this paper, for simplicity of notation, we implicitly assume that every function f is associated with a single approximation $C_{f,\epsilon}$ (the "natural" approximation for f).

Following are a few examples of popular approximation notions interpreted in light of the above definition. In all of them we define $C_{f,0}(x) \stackrel{\text{def}}{=} \{f(x)\}$.

1. *Additive approximation:* (B, d_B) is a metric space.
   $C_{f,\epsilon}(x) = \{y \in B \mid d_B(y, f(x)) < \epsilon\}$ for $\epsilon > 0$.

2. *Relative approximation:* $B = \mathbf{R}$.
   $C_{f,\epsilon}(x) = \{y \in B \mid (1-\epsilon)f(x) < y < (1+\epsilon)f(x)\}$ for $0 < \epsilon \leq 1$.

3. *Ratio approximation:* $B = \mathbf{R}$.
   $C_{f,\epsilon}(x) = \{y \in B \mid (1-\epsilon)f(x) < y < (1/(1-\epsilon))f(x)\}$ for $0 < \epsilon < 1$.

4. *Rank approximation:* An approximation notion for selection functions, like the median. The input for Select_q, the selection function of order q ($0 \leq q \leq 1$), is a set of n real numbers $a_1, \ldots, a_n$. If we order these numbers from smallest to largest: $a_{i_0} \leq \cdots \leq a_{i_{n-1}}$, then $\text{Select}_q(a_1, \ldots, a_n) = a_{i_{\lfloor q(n-1) \rfloor}}$. For example, the median is $\text{Select}_{\frac{1}{2}}$, minimum is Select_0, and maximum is Select_1. The approximation for $0 < \epsilon \leq 1$ is: $C_{\text{Select}_q,\epsilon}(a_1, \ldots, a_n) = \{a_{i_j} \mid j \in (\lfloor q'(n-1) \rfloor, \lfloor q''(n-1) \rfloor), q' = \max(q-\epsilon, 0), q'' = \min(q+\epsilon, 1)\}$.

5. *Property testing approximation:* An approximation notion for functions with a Boolean range. We assume some distance measure D on the input domain. The approximation for $\epsilon > 0$ is: $C_{f,\epsilon}(x) = \{1\}$, if $D(x,y) < \epsilon/2$ for some input y with $f(y) = 1$; $C_{f,\epsilon}(x) = \{0\}$, if $D(x,y) > \epsilon$ for all inputs y with $f(y) = 1$; and $C_{f,\epsilon}(x) = \{0, 1\}$ otherwise.

A special kind of approximation is one that satisfies the "sunflower property," defined below. It is easy to check that approximations (2)–(5) above satisfy the sunflower property. Approximation (1) satisfies it, if $B = \mathbf{R}$ and $d_B(x,y) = |x - y|$.

Definition 2.2 (Sunflower property) *An approximation $C_{f,\epsilon}$ is said to satisfy the* sunflower property*, if for every $D \subseteq A^n$, the following condition holds:*

$$(\forall x, x' \in D)[C_{f,\epsilon}(x) \cap C_{f,\epsilon}(x') \neq \emptyset] \Rightarrow \bigcap_{x \in D} C_{f,\epsilon}(x) \neq \emptyset.$$

A notion that will play an important role in our discussion is the following:

Definition 2.3 (Disjoint inputs) *Two inputs $x, y \in A^n$ are said to be ϵ-disjoint with respect to approximation $C_{f,\epsilon}$, if $C_{f,\epsilon}(x) \cap C_{f,\epsilon}(y) = \emptyset$.*

For ϵ-disjoint inputs x, y, any value $z \in B$ which is an ϵ-approximation of $f(x)$ is not an ϵ-approximation of $f(y)$ and vice versa.

2.2 Decision Trees for General Functions

A *randomized decision tree* for a function $f : A^n \to B$ is a rooted labeled tree, not necessarily binary. Similar to Boolean decision trees, internal nodes are associated with either queries of input variables or random coin tosses, and leaves are associated with outputs. Given an input x, we use the queries and the random coin tosses to determine a path from the root to one of the leaves. This leaf specifies the output on x.

Formally, for every node v we denote by $\deg(v)$ the number of children it has and by $S(v)$ the number of query nodes on the path from the root to v (including v itself). If v is a query node, it is labeled by an input variable $i \in [n]$ and a children selection function $N_v : A^{S(v)} \to [\deg(v)]$. If the path reaches v, we apply N_v on the values queried so far to determine which child of v should be the next node on the path. If v is a random coin node, it is labeled by the special character \$, instructing the path to pick one of its children uniformly at random. If v is a leaf, it is labeled by an output function $O_v : A^{S(v)} \to B$, which specifies the output of the tree, based on the values queried along the path. Note that any input x may be associated with several possible paths leading from the root to a leaf, depending on the random choices made in the random coin nodes. These random choices induce a distribution over the paths corresponding to x.

A *deterministic decision tree* is one that has no random coin nodes. In such trees each input x is associated with a *single* path from the root to a leaf.

For an input x and a decision tree T, we denote by $T(x)$ the output of T on x and by $S(T, x)$ its *query cost*—the number of query nodes on the path in T corresponding to x. When T is deterministic $T(x)$ and $S(T, x)$ are fixed values, while when T is randomized they are random variables. For randomized decision trees, we discuss two notions of cost: the *expected query cost*: $S^e(T, x) \stackrel{\text{def}}{=} E(S(T, x))$ and the *worst-case query cost*: $S^w(T, x) \stackrel{\text{def}}{=} \max S(T, x)$, where the expectation and the maximum are over the distribution of paths in T corresponding to x.

The *query complexity* of a deterministic decision tree T is $S(T) \stackrel{\text{def}}{=} \max_{x \in A^n} S(T, x)$, and the *expected* and *worst-case query complexities* of a randomized decision tree T are, respectively, $S^e(T) \stackrel{\text{def}}{=} \max_{x \in A^n} S^e(T, x)$ and $S^w(T) \stackrel{\text{def}}{=} \max_{x \in A^n} S^w(T, x)$.

Definition 2.4 ((ϵ, δ)-approximation) *Let $f : A^n \to B$ be a function, $\epsilon \geq 0$ an error parameter, and $0 \leq \delta \leq 1$ a confidence parameter.*

(1) A deterministic decision tree T is said to ϵ-approximate *f, if*

$$\forall x \in A^n, \quad T(x) \in C_{f,\epsilon}(x)$$

(2) A randomized decision tree T is said to (ϵ, δ)-approximate f*, if*

$$\forall x \in A^n, \quad \Pr(T(x) \in C_{f,\epsilon}(x)) > 1 - \delta$$

The ϵ*-(deterministic) query complexity* of f is $D_\epsilon(f) \stackrel{\text{def}}{=} \min\{S(T) \mid T\ \epsilon\text{-approximates } f\}$. Similarly, the (ϵ, δ)*-expected (randomized) query complexity* and the (ϵ, δ)*-worst-case query complexity* of f are, respectively, $R^e_{\epsilon,\delta}(f) \stackrel{\text{def}}{=} \min\{S^e(T) \mid T\ (\epsilon, \delta)\text{-approximates } f\}$ and $R^w_{\epsilon,\delta}(f) \stackrel{\text{def}}{=} \min\{S^w(T) \mid T\ (\epsilon, \delta)\text{-approximates } f\}$.

Notice that this model can simulate, with the same efficiency, various models of decision trees, including Boolean, comparison, and algebraic decision trees.

Every fixing of the random choices made in the random coin nodes of a randomized decision tree induces a deterministic decision tree. Thus, a randomized decision tree T can be viewed as a distribution μ over deterministic decision trees $T_1, \ldots, T_m$. With this view, $S^e(T) = E_{T_j \in \mu}(S(T_j))$ and $S^w(T) = \max_{j \in [m]} S(T_j)$. Also, if T (ϵ, δ)-approximates a function $f : A^n \to B$, then $\forall x \in A^n$, $\Pr_{T_j \in \mu}(T_j(x) \in C_{f,\epsilon}(x)) > 1 - \delta$. This characterization turns out to be very useful in some situations.

3 A General Lower Bound via Block Sensitivity

In this section we generalize the notion of block sensitivity (defined by Nisan [Nis91] for Boolean functions) and obtain a lower bound on the expected query complexity in terms of block sensitivity.

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be sensitive to a subset (a "block") $I \subseteq [n]$ on input x, if f flips its value when flipping all the bits in I. The block sensitivity of f on x, $\text{bs}(f, x)$, is the maximum number t of pairwise disjoint subsets $I_1, \ldots, I_t \subseteq [n]$, to which f is sensitive on x. The block sensitivity of f, $\text{bs}(f)$, is the maximum, over all inputs x, of $\text{bs}(f, x)$.

For general domains and ranges, the basic intuition we carry over from the Boolean case is that a function is sensitive to a block of variables if a change to the corresponding input elements results in a significant change to the function value. In the following definition we denote by $x^{(I \leftarrow Q)}$ the input obtained from x by changing the elements in $I \subseteq [n]$ to the values in $Q \in A^{|I|}$:

Definition 3.1 (Block sensitivity) *f is* ϵ-sensitive to a subset of variables $I \subseteq [n]$ on input x*, if there exists $Q \in A^{|I|}$ such that x and $x^{(I \leftarrow Q)}$ are ϵ-disjoint.* $\text{bs}_\epsilon(f, x)$*, the* ϵ-block sensitivity of f on x*, is the maximum number t of pairwise disjoint subsets $I_1, \ldots, I_t \subseteq [n]$, such that f is ϵ-sensitive to each of them on x.* $\text{bs}_\epsilon(f)$*, the* ϵ-block sensitivity of f*, is* $\max_{x \in A^n} \text{bs}_\epsilon(f, x)$.

Nisan [Nis91] proved that for Boolean functions f, $R^e_{0,\delta}(f) \geq (1 - 2\delta)\text{bs}(f)$. We generalize the proof of this theorem to our case.

Theorem 3.2 *For every $f : A^n \to B$, $\epsilon \geq 0$, and $0 \leq \delta \leq 1/2$,*

$$R^e_{\epsilon,\delta}(f) > (1 - 2\delta) \cdot \text{bs}_\epsilon(f)$$

Proof. Consider any decision tree T that (ϵ, δ)-approximates f. We will use the view of T as a distribution μ over deterministic decision trees $T_1, \ldots, T_m$. We call a tree T_j "bad" for an input x, if $T_j(x) \notin C_{f,\epsilon}(x)$. Note that for every input x, $\Pr_{T_j \in \mu}(T_j \text{ is bad for } x) < \delta$.

Let x be the input that achieves the block sensitivity of f, and let $I_1, \ldots, I_t \subseteq [n]$ be the disjoint variable blocks to which f is ϵ-sensitive on x (note that $t = \mathrm{bs}_\epsilon(f)$). Finally, let $Q_1, \ldots, Q_t$ be the assignments to these blocks, for which x and $x^{(I_k \leftarrow Q_k)}$ are ϵ-disjoint.

Consider a tree T_j in which the path corresponding to x does not query any of the variables in I_k for some $k \in [t]$. This means that the same path will correspond to $x^{(I_k \leftarrow Q_k)}$. Moreover, since all the values queried on this path are the same at x and at $x^{(I_k \leftarrow Q_k)}$, the leaf will output the same value y for both. However, since $C_{f,\epsilon}(x) \cap C_{f,\epsilon}(x^{(I_k \leftarrow Q_k)}) = \emptyset$ either $y \notin C_{f,\epsilon}(x)$ or $y \notin C_{f,\epsilon}(x^{(I_k \leftarrow Q_k)})$. So T_j is bad either for x or for $x^{(I_k \leftarrow Q_k)}$. Therefore,

$$\Pr_{T_j \in \mu}(T_j \text{ does not query } I_k \text{ on } x) \leq \Pr_{\mu}(T_j \text{ is bad for } x) + \Pr_{\mu}(T_j \text{ is bad for } x^{(I_k \leftarrow Q_k)}) < 2\delta$$

Recall that $S(T, x)$ is a random variable specifying the number of queries T performs on x. Define for every $k \in [t]$, a random variable X_k which counts the number of these queries that belong to I_k. Since $I_1, \ldots, I_t$ are pairwise disjoint, $S(T, x) \geq \sum_{k=1}^{t} X_k$. What we proved above implies that $\Pr(X_k \geq 1) > 1 - 2\delta$, implying that also $E(X_k) > 1 - 2\delta$. Therefore, by linearity of expectation,

$$R^e_{\epsilon,\delta}(f) \geq E(S(T, x)) \geq E(\sum_{k=1}^{t} X_k) = \sum_{k=1}^{t} E(X_k) > t \cdot (1 - 2\delta) = (1 - 2\delta) \cdot \mathrm{bs}_\epsilon(f)$$

□

Let us consider an application of this lower bound for additive approximation of the mean: $\mu_1 : [0,1]^n \to [0,1]$. The best upper bound is $R^w_{\epsilon,\delta}(\mu_1) = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$. Using Theorem 3.2 we prove the following:

Proposition 3.3 $R^e_{\epsilon,\delta}(\mu_1) \geq \Omega(\frac{1}{\epsilon}(1 - 2\delta))$

Proof. Let $x = 1^n$, and consider any subset $I \subseteq [n]$ of size $2\epsilon n$. Define an assignment $Q = 0^{|I|}$. Since $\mu_1(x) = 1$ and $\mu_1(x^{(I \leftarrow Q)}) = 1 - 2\epsilon$, x and $x^{(I \leftarrow Q)}$ are ϵ-disjoint, implying μ_1 is ϵ-sensitive to I on x. This argument works for any subset I of size $2\epsilon n$; thus, the maximum number of pairwise disjoint subsets to which μ_1 is ϵ-sensitive on x is $n/|I| = 1/2\epsilon$. Applying Theorem 3.2 completes the proof. □

Proposition 3.3 shows that the block sensitivity lower bound does not yield tight bounds in general. In the case of the mean, the bound produced is not tight not in terms of the error ϵ, and not in terms of the confidence δ. In Section 5 we show how our second lower bound technique gives a tight lower bound for the mean.

Nisan [Nis91] proved that for Boolean functions, block sensitivity and (deterministic/randomized) decision tree complexity are polynomially related. A particular corollary of this is that for Boolean functions, deterministic and randomized decision tree complexities are polynomially related. It is natural to ask if block sensitivity also gives a (polynomial) characterization of randomized decision tree complexity for approximating general functions, but it turns out not to be the case. While the ϵ-approximate median problem has a randomized sampling algorithm that makes $O(\epsilon^{-2})$ probes (independent of n), it is easy to show that any deterministic algorithm needs at least $\Omega(n(1-\epsilon))$ probes (see Section 4.7).

4 Lower Bounds for Symmetric Functions

The query complexity lower bound via block sensitivity is very general but is also weak. In this section we obtain stronger lower bounds that work only for *symmetric functions* — functions that are invariant under permutations of the input elements:

Definition 4.1 (Symmetric functions) *A function $f : A^n \to B$ is* ϵ-symmetric *if for every input $x \in A^n$ and every permutation $\pi \in S_n$, $C_{f,\epsilon}(x) = C_{f,\epsilon}(\pi(x))$ (where $\pi(x)_i = x_{\pi^{-1}(i)}$).*

The basic idea for the lower bounds is to show that for symmetric functions we can show that the "computer science sampling setting", in which different queries are not necessarily independent and do not have the same query distribution, is almost equivalent to the "statistics setting", in which all queries are independent and identically distributed ("the i.i.d. case"). We show that with almost no loss of efficiency, any general sampling algorithm for a symmetric function can be simulated by one that makes queries that are independent and uniformly distributed.

We then observe that a sampling algorithm of query complexity k that approximates $f : A^n \to B$ and makes only independent uniform queries can be used to form a *statistical test* that uses k samples to distinguish between the distributions U_x and U_y for any pair of ϵ-disjoint inputs x and y. Here, U_x and U_y are the distributions on A obtained by picking $i \in [n]$ uniformly at random and returning x_i and y_i respectively. Clearly, the closer U_x and U_y are, the harder it is to distinguish between them. Thus, by giving a lower bound on the number of samples required to distinguish between any two such input distributions, we obtain a lower bound on the query complexity of the function.

We state the lower bounds in terms of new properties of functions. These properties use the Hellinger distance and the variational distance between distributions:

Definition 4.2 (Statistical distance measures) *Let P and Q be two distributions on the same probability space Ω. The* Hellinger distance *h and the* variational distance *d between P and Q are defined as follows:*

$$h(P,Q) \stackrel{\text{def}}{=} (1 - \textstyle\sum_{\omega\in\Omega} \sqrt{P(\omega)Q(\omega)})^{\frac{1}{2}} = (\frac{1}{2}\textstyle\sum_{\omega\in\Omega}(\sqrt{P(\omega)} - \sqrt{Q(\omega)})^2)^{\frac{1}{2}}$$

$$d(P,Q) \stackrel{\text{def}}{=} \tfrac{1}{2}\textstyle\sum_{\omega\in\Omega} |P(\omega) - Q(\omega)| = \max_{D\subseteq\Omega} |P(D) - Q(D)|$$

We use these distance measures to define the following properties:

Definition 4.3 (Minimum Hellinger/Variational distance) *For a function $f : A^n \to B$ and a distance measure L between distributions on A, the* ϵ-minimum L-distance of f *is*

$$L_\epsilon(f) \stackrel{\text{def}}{=} \min\{L(U_x, U_y) \mid x, y \in A^n \textit{ are } \epsilon\textit{-disjoint}\}$$

Our main theorem gives two lower bounds on the worst-case query complexity of symmetric function in terms of their minimum Hellinger and variational distances:

Theorem 4.4 (Main theorem) *For every $\epsilon \geq 0$ and for every ϵ-symmetric function $f : A^n \to B$,*
(1) if $\frac{1}{2n} < \delta < \frac{1}{2}$, $R^w_{\epsilon,\delta}(f) \leq \sqrt{2\delta n}$, and $h_\epsilon(f) \leq \frac{1}{2}$, then

$$R^w_{\epsilon,\delta}(f) \geq \frac{1}{4h^2_\epsilon(f)} \ln \frac{1}{8\delta}$$

(2) if $\frac{2}{n} < \delta < \frac{1}{2}$, $R^w_{\epsilon,\delta}(f) \leq n/2 - \sqrt{n/2\delta}$, and $d_\epsilon(f) \leq \frac{1}{2}$, then

$$R^w_{\epsilon,\delta}(f) \geq \frac{1}{8d_\epsilon(f)} \ln \frac{1}{8\delta}$$

Our strategy for the proof of Theorem 4.4 is the following: We first consider (Section 4.1) a variant of the decision tree model, in which all queries are made according to a fixed query distribution. We then prove (Theorem 4.11) that, almost without any loss in efficiency, any symmetric function can be computed by such decision trees that probe the input at uniformly chosen positions. The consequence of this theorem is that if a decision tree makes k queries to an input x, the distribution it "sees" is U_x^k, the k-fold product of U_x.

Lemma 4.12 shows that a decision tree that (ϵ, δ)-approximates f using k (worst-case) queries can be used to distinguish between the distributions U_x^k and U_y^k for any ϵ-disjoint x, y, with advantage 2δ. This immediately implies that $d(U_x^k, U_y^k) \geq 1 - 2\delta$. Lemma 4.12 essentially reduces the problem of (ϵ, δ)-approximating $f(x)$ to the problem of statistical hypothesis testing. In hypothesis testing, one is given samples from a distribution D which is known to be either P or Q, and is required to decide whether $D = P$ or $D = Q$. In our case the two distributions are U_x and U_y, and the hypothesis tester is the decision tree: if the tree outputs a value in $C_{f,\epsilon}(x)$, we decide "U_x", and otherwise we decide "U_y". Since x and y are ϵ-disjoint, and since the tree outputs an ϵ-approximation with probability at least $1 - \delta$, this gives a statistical test that succeeds with probability at least $1 - 2\delta$.

We then derive a lower bound (Lemma 4.14) on the number of samples needed for an hypothesis test to distinguish between two distributions (with a certain advantage) in terms of their Hellinger distance. The tightness of this bound follows from a useful multiplicativity property of the Hellinger distance.

Some technical complications make the above argument work only for functions with query complexity which is at most $O(\sqrt{n})$. For functions with larger query complexity we can prove only a weaker lower bound in terms of the minimum variational distance.

In Section 4.6 we show a general reduction from worst-case query complexity to expected query complexity, which implies that for any function f, $R^w_{\epsilon,\frac{\delta}{3}}(f) \leq R^e_{\epsilon,\delta}(f) \cdot O(\log \frac{1}{\delta})$. This reduction implies that Theorem 4.4 holds also for the expected query complexity except for the dependence on the confidence δ:

Theorem 4.5 *For every $\epsilon \geq 0$ and for every ϵ-symmetric function $f : A^n \to B$,*
(1) if $\frac{1}{6n} < \delta < \frac{1}{576}$, $R^w_{\epsilon,3\delta}(f) \leq \sqrt{6\delta n}$, and $h_\epsilon(f) \leq \frac{1}{2}$, then

$$R^e_{\epsilon,\delta}(f) \geq \frac{1}{64 h^2_\epsilon(f)}$$

(2) if $\frac{2}{3n} < \delta < \frac{1}{576}$, $R^w_{\epsilon,3\delta}(f) \leq n/2 - \sqrt{n/6\delta}$, and $d_\epsilon(f) \leq \frac{1}{2}$, then

$$R^e_{\epsilon,\delta}(f) \geq \frac{1}{128 d_\epsilon(f)}$$

We end the section by presenting (Section 4.7) a simple and general lower bound on the deterministic query complexity of symmetric functions in terms of their *minimum Hamming distance*:

Definition 4.6 (Minimum Hamming distance) *The* ϵ-minimum Hamming distance *of a function $f : A^n \to B$ is:*

$$\mathrm{Ham}_\epsilon(f) \stackrel{\mathrm{def}}{=} \min\{\mathrm{Ham}(x,y) \mid x, y \in A^n \ \textit{are } \epsilon\textit{-disjoint}\}$$

where $\mathrm{Ham}(x,y) = \frac{1}{n}|\{i \in [n] \mid x_i \neq y_i\}|$.

A deterministic algorithm can distinguish between ϵ-disjoint inputs only if it finds a variable on which they disagree. Since for symmetric functions the order of the input elements does not matter, an adversary can re-order these elements in a way such that the algorithm will have query all the variables on which the disjoint inputs agree before it gets to find a location on which they disagree. This is the intuition behind the following lower bound:

Theorem 4.7 *For every $\epsilon \geq 0$ and for every ϵ-symmetric function $f : A^n \to B$,*

$$D_\epsilon(f) > n \cdot (1 - \mathrm{Ham}_\epsilon(f))$$

This lower bound implies that in most practical situations a deterministic algorithm will need a linear number of queries in order to approximate a symmetric function. This shows that deterministic sampling algorithms are usually non-practical.

For simplicity of notation, we will assume throughout this section that the domain A is finite. Everything we prove in the sequel can be generalized to arbitrary domains by replacing summation by integration.

4.1 Oblivious Decision Trees

In this section we introduce a restricted class of decision trees, called *oblivious decision trees*, in which the queries are chosen in advance according to a fixed query distribution, independently of the input and of the random choices of the tree.

In order to formally define oblivious decision trees, we first introduce the following transformation on decision trees, called *variable relabeling.* Let T be a decision tree of worst-case query complexity k, and let $I = (i_1, \ldots, i_k)$ be a sequence of k input variables. The *I-relabeling* of T, denoted T^I, is a decision tree obtained from T by relabeling all its query nodes of depth j (that is, $S(v) = j$), $j = 1, \ldots, k$, by i_j.

An oblivious decision tree is a pair (D_k, T) where D_k is a *query distribution* on $[n]^k$ and T is a decision tree. k is the maximum number of queries T may need (i.e., the worst-case query complexity of T). An oblivious decision tree chooses k query variables $I = (i_1, \ldots, i_k)$ *in advance* according to the distribution D_k, and then runs T^I on the input x.

Note that, as opposed to standard decision trees, a query node v of an oblivious decision tree is not associated with a single query variable i_v. Several different query variables may be "plugged-in" into v, depending on the choices made by D_k. In particular, the identity of this query variable is not apparent from the query node. We therefore consider two variants of oblivious decision trees: *variable-aware* and *variable-oblivious.* In variable-aware decision trees, the tree is provided with the identities of the variables queried as well as with the outcomes of the queries. This is implemented by augmenting the child selection function of every query node v in T and the output function of every leaf u in T with an additional input argument specifying the variables queried along the paths leading to v and u. That is, $N_v : ([n] \times A)^{S(v)} \to [\deg(v)]$ and $O_u : ([n] \times A)^{S(u)} \to B$. Variable-oblivious decision trees, on the other hand, are not provided with any explicit information about the variables queried; they get only the outcomes of the queries. In such trees, therefore, the child selection functions and the output functions are as usual: $N_v : A^{S(v)} \to [\deg(v)]$ and $O_u : A^{S(u)} \to B$.

Oblivious decision trees are indeed a sub-class of standard decision trees, as long as the query distribution D_k consists only of rational probabilities, because the choice of the query variables according to D_k can be simulated by random coin nodes.

We call an oblivious decision tree with a query distribution D_k a *D_k-decision tree.* Two examples of query distributions we will use are the following:

(1) U_k – *Uniform queries with replacement*: picks k variables from $[n]$ independently and uniformly at random.

(2) W_k – *Uniform queries without replacement*: picks a uniform subset of $[n]$ of size k.

A variable-aware D_k-decision tree T for a function $f : A^n \to B$ can be viewed as a mapping $T(I, x|_I, R)$ that maps the chosen query sequence $I = (i_1, \ldots, i_k)$, the query outcomes $x|_I$, and the random choices of the tree R into an output value in B. A variable-oblivious D_k-decision tree is a mapping $T(x|_I, R)$ that maps only the query outcomes and the random choices into a value in B. The inputs of these mappings distribute as follows: I distributes according to the query distribution D_k, $x|_I$ distributes according to the *query outcome distribution* $D_{k,x}$ (the distribution on A^k induced by D_k and x), and R distributes uniformly at random. We call the joint distribution of I and $x|_I$ the *full query distribution* and denote it by $\tilde{D}_{k,x}$. Thus, variable-aware decision trees have access to the full query distribution, while variable-oblivious decision trees have access only to the query outcome distribution.

The definitions of (ϵ, δ)-approximation and expected/worst-case query complexity are modified to take into account both the probability over the random choices of the tree and the probability over the random query variables.

4.2 Uniform Decision Trees for Symmetric Functions

The aim of this section is to show (Theorem 4.11) that for symmetric functions the restricted class of U_k-decision tree (i.e., ones that query only uniformly at random with replacement) is almost as good as general decision trees. We initially prove (Lemma 4.8) that any decision tree for a symmetric function can be simulated by a variable-oblivious W_k-decision tree with the same query complexity; we then present (Lemma 4.10) two simulations of variable-oblivious W_k-decision trees by U_k-decision trees: a variable-oblivious simulation that works only for $k \leq O(\sqrt{n})$, and a variable-aware simulation that works for $k \leq n/2 - o(n)$.

Lemma 4.8 *For every $\epsilon \geq 0$ and every ϵ-symmetric function $f : A^n \to B$ there exist a variable-oblivious W_k-decision tree of expected query complexity $R^e_{\epsilon,\delta}(f)$ that (ϵ, δ)-approximate f. Similarly, there exists a variable-oblivious W_k-decision tree of worst-case query complexity $R^w_{\epsilon,\delta}(f)$ that (ϵ, δ)-approximate f.*

Proof. Let T be any decision tree that (ϵ, δ)-approximates f. Without loss of generality, we assume that T never queries the same variable more than once. We will show a construction of a variable-oblivious W-decision tree T' that simulates T with the same expected and worst-case query complexities.

Let k_e be expected query complexity of T and let k_w be its worst-case query complexity. In the following we denote $k = k_w$.

T' will be a W_k-decision tree that uses the tree T. That is, T' choose $I = (i_1, \ldots, i_k)$ according to W_k and runs T^I on x. Since T is a standard decision tree, its child selection functions and its output functions depend only on query outcomes and not on the query variables. Therefore, T' is variable-oblivious. Clearly, T' has worst-case query complexity k. We are left to prove that T' (ϵ, δ)-approximates f, and that its expected query complexity is k_e.

We will use the view of T as a distribution μ over deterministic decision tress $T_1, \ldots, T_m$. Under this view, T' first picks I, then chooses T_j according to μ, and finally runs T^I_j on x. We next prove some basic properties of the I-relabeling transformation that will be useful for our analysis:

Proposition 4.9 *Let T be a deterministic decision tree of complexity k that never queries the same variable twice. Then, for any sequence of k distinct variables $I = (i_1, \ldots, i_k)$ and for every input x, there is a set of permutations $S^I_x \subseteq S_n$ such that for every $\pi \in S^I_x$, the path of x in T^I is identical* *to the path of $\pi(x)$ in T. Moreover, every permutation $\pi \in S_n$ belongs to S_x^I for some choice of I, and for every $I \neq I'$, either $S_x^I = S_x^{I'}$ or $S_x^I \cap S_x^{I'} = \emptyset$.*

Proof. We iteratively construct a partial permutation $\sigma \in S_n$, such that x has the same path in T^I as the path of $\overline{\sigma}(x)$ in T, where $\overline{\sigma}$ is any permutation consistent with σ. S_x^I will be the set of all such permutations consistent with σ.

In the ℓ-th step of the construction σ is defined on $i_1, \ldots, i_{\ell-1}$ and satisfies that the first ℓ nodes on the path of x in T^I are the same as the first ℓ nodes on the path of $\overline{\sigma}(x)$ in T.

The induction base, $\ell = 1$, is trivial, since we choose σ to be empty (and therefore every permutation is consistent with it) and since the first node on the path of every input is the root.

Assume now we defined a partial permutation σ on $i_1, \ldots, i_{\ell-1}$, such that the first ℓ nodes on the path of x in T^I are identical to the first ℓ nodes on the path of $\overline{\sigma}(x)$ in T. Let us call these nodes $v_1, \ldots, v_\ell$. If v_ℓ is a leaf, we stop, and define S_x^I to be all the permutations consistent with σ. Otherwise, v_ℓ is a query node; let j_ℓ be the variable labeling it in T. We define $\sigma(i_\ell) = j_\ell$. Recall that since v_ℓ is of depth ℓ, it is labeled by i_ℓ in T^I. Therefore, the item queried in v_ℓ when running T^I on x is x_{i_ℓ}. On the other hand, the item queried in v_ℓ when running T on $\overline{\sigma}(x)$ is $\overline{\sigma}(x)_{j_\ell} = x_{i_\ell}$. This implies that the paths of x and $\overline{\sigma}(x)$ will share also the $\ell+1$-st node.

Consider now any permutation $\pi \in S_n$. Let $v_1, \ldots, v_\ell$ ($\ell \leq k$) be the query nodes on the path of $\pi(x)$ in T. Let $j_1, \ldots, j_\ell$ be the input variables labeling $v_1, \ldots, v_\ell$. We choose $I = i_1, \ldots, i_k$ as follows: $i_1 = \pi^{-1}(j_1), \ldots, i_\ell = \pi^{-1}(j_\ell)$ and $i_{\ell+1}, \ldots, i_k$ are arbitrary distinct variables different from $i_1, \ldots, i_\ell$. The path of $\pi(x)$ in T is identical to the path of x in T^I, and therefore $\pi \in S_x^I$.

Finally, consider two sequences of input variables I and I'. Let σ_I and $\sigma_{I'}$ be the partial permutations defining S_x^I and $S_x^{I'}$. Let $i_1, \ldots, i_\ell$ be the prefix of I on which σ_I is defined and let $i'_1, \ldots, i'_{\ell'}$ be the prefix of I' on which $\sigma_{I'}$ is defined. If $(\imath_1, \ldots, \imath_l) = (i'_1, \ldots, i'_{\ell'})$, then $\sigma_I = \sigma_{I'}$ and therefore $S_x^I = S_x^{I'}$. Otherwise, notice that $(i_1, \ldots, i_\ell)$ cannot be a prefix of $(i'_1, \ldots, i'_{\ell'})$ (and vice versa); therefore, let t be the first index such that $i_t \neq i'_t$. Since $(i_1, \ldots, i_{t-1}) = (i'_1, \ldots, i'_{t-1})$, the first t nodes on the path of $\pi(x)$ in T are identical to the first t nodes on the path of $\sigma(x)$ in T for every $\pi \in S_x^I$ and every $\sigma \in S_x^{I'}$. Let j_t be the variable labeling the t-th node in these paths. By the definition of S_x^I and $S_x^{I'}$, $\pi(j_t) = i_t$ and $\sigma(j_t) = i'_t$ for every $\pi \in S_x^I$ and every $\sigma \in S_x^{I'}$. Therefore, $S_x^I \cap S_x^{I'} = \emptyset$. $\square$

By Proposition 4.9, for any tree T_j, any query sequence I, and any input x, there is a set of permutations $S_x^{j,I}$ such that for all $\pi \in S_x^{j,I}$, $T_j^I(x) = T_j(\pi(x))$. Denote by $U_{j,I,x}$ the uniform distribution on $S_x^{j,I}$. We obtain:

$$\Pr(T'(x) \in C_{f,\epsilon}(x)) = \Pr_{T_j,I}(T_j^I(x) \in C_{f,\epsilon}(x)) = \Pr_{T_j,I,\pi\in U_{j,I,x}}(T_j(\pi(x)) \in C_{f,\epsilon}(x))$$

Since f is ϵ-symmetric, $C_{f,\epsilon}(x) = C_{f,\epsilon}(\pi(x))$. Therefore,

$$\Pr(T'(x) \in C_{f,\epsilon}(x)) = \Pr_{T_j,I,\pi\in U_{j,I,x}}(T_j(\pi(x)) \in C_{f,\epsilon}(\pi(x))) = E_{T_j}(\Pr_{I,\pi\in U_{j,I,x}}(T_j(\pi(x)) \in C_{f,\epsilon}(\pi(x))))$$

Proposition 4.9 implies that for every fixed input x and decision tree T_j, the distribution of π when choosing I according to W_k and then π uniformly at random from $S_x^{j,I}$ is uniform in S_n. Denote by $U(S_n)$ the uniform distribution on S_n. We thus obtain:

$$\begin{aligned}\Pr(T'(x) \in C_{f,\epsilon}(x)) &= E_{T_j}(\Pr_{\pi \in U(S_n)}(T_j(\pi(x)) \in C_{f,\epsilon}(\pi(x)))) \\ &= E_{\pi \in U(S_n)}(\Pr_{T_j}(T_j(\pi(x)) \in C_{f,\epsilon}(\pi(x)))) \ > \ 1-\delta\end{aligned}$$

where the last inequality follows from the fact T (ϵ, δ)-approximates f. This proves that T' (ϵ, δ)-approximates f.

By Proposition 4.9 for any choice of T_j, any choice of I, and any input x, for all permutations $\pi \in S_x^{j,I}$, $S(T_j^I, x) = S(T_j, \pi(x))$. Therefore,

$$\begin{aligned}S^e(T') &= \max_x E_{I,T_j}(S(T_j^I, x)) \ = \ \max_x E_{I,T_j,\pi \in U_{j,I,x}}(S(T_j, \pi(x)) \\ &= \max_x E_{T_j, \pi \in U(S_n)}(S(T_j, \pi(x))) \ \leq \ \max_x \max_{\pi \in S_n} E_{T_j}(S(T_j, \pi(x))) = k_e\end{aligned}$$

□

The following lemma presents two simulations of variable-oblivious W_k-decision trees by U_k-decision trees with almost the same query complexity. The first one works only decision trees of small enough query complexity ($k \leq O(\sqrt{n})$), but can be carried out variable-obliviously; the second one works for almost any query complexity ($k \leq n/2 - o(n)$), but is variable-aware.

Lemma 4.10 *Let T be a variable-oblivious W_k-decision tree of worst-case query complexity k_w and expected query complexity k_e that (ϵ, δ)-approximates a function $f : A^n \to B$. Then,*

(1) if $\frac{1}{2n} < \delta < \frac{1}{2}$ and $k_w \leq \sqrt{2\delta n}$, there exists a variable-oblivious U_{k_w}-decision tree T' of worst-case query complexity k_w and expected query complexity $(1-\delta)k_e + \delta k_w$ that $(\epsilon, 2\delta)$-approximates f.

(2) if $\frac{2}{n} < \delta < \frac{1}{2}$ and $k_w \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$, there exists a variable-aware U_{2k_w}-decision tree T'' of worst-case query complexity $2k_w$ and expected query complexity $2k_e$ that $(\epsilon, 2\delta)$-approximates f.

Proof.

We view T as a distribution μ over deterministic decision trees $T_1, \ldots, T_m$. Given input x, T chooses I uniformly at random without replacement and T_j according to μ and runs T_j^I on x.

In the following we denote $k = k_w$.

Proof of part (1):

Given input x, T' picks $I = i_1, \ldots, i_k$ uniformly at random *with* replacement and a decision tree T_j according to μ, and runs T_j^I on x. Clearly, T' is a U_k-decision tree whose worst-case query complexity is k. Since T is variable-oblivious then so is T'. We are left to prove that if $k \leq \sqrt{2\delta n}$, then T' $(\epsilon, 2\delta)$-approximates f and that its expected query complexity is $(1-\delta)k_e + \delta k$.

Let E denote the event that $i_1, \ldots, i_k$ are distinct, and let $\overline{E}$ denote its complement. Note that conditioning on E, the distribution of $i_1, \ldots, i_k$ is uniform without replacement. Thus, for any input x,

$$
\begin{aligned}
&\Pr_{I\in U_k,T_j}(T_j^I(x)\notin C_{f,\epsilon}(x)) = \\
&= \Pr_{I\in U_k,T_j}(T_j^I(x)\notin C_{f,\epsilon}(x) \mid E)\cdot \Pr_{I\in U_k}(E) + \Pr_{I\in U_k,T_j}(T_j^I(x)\notin C_{f,\epsilon}(x) \mid \overline{E})\cdot \Pr_{I\in U_k}(\overline{E}) \\
&\le \Pr_{I\in W_k,T_j}(T_j^I(x)\notin C_{f,\epsilon}(x)) + \Pr_{I\in U_k}(\overline{E}) \\
&< \delta + \Pr_{I\in U_k}(\overline{E})
\end{aligned}
$$

We next show that $\Pr(\overline{E})\le\delta$, implying that T' $(\epsilon,2\delta)$-approximates f. Define $\binom{k}{2}$ indicator random variables X_{jl}, such that $X_{jl}=1$ iff $i_j=i_l$. $i_1,\ldots,i_k$ are not all distinct if and only if $\sum_{j,l}X_{jl}\ge 1$. The expectation of $\sum_{j,l}X_{jl}$ is $\binom{k}{2}\frac{1}{n}\le\frac{k^2}{2n}$. Therefore, using Markov's inequality, $\Pr(\overline{E})\le\frac{k^2}{2n}\le\delta$.

We finish the proof of part (1) by proving that $S^e(T')\le(1-\delta)k_e+\delta k$:

$$
\begin{aligned}
S^e(T') &= \max_x E_{I\in U_k,T_j}(S(T_j^I,x)) \\
&= \max_x (E_{I\in U_k,T_j}(S(T_j^I,x)\mid E)\Pr_{I\in U_k}(E) + E_{I\in U_k,T_j}(S(T_j^I,x)\mid\overline{E})\Pr_{I\in U_k}(\overline{E})) \\
&\le \max_x (E_{I\in W_k,T_j}(S(T_j^I,x)))\cdot(1-\Pr(\overline{E})) + k\cdot\Pr(\overline{E}) \;\le\; k_e(1-\delta)+k\delta
\end{aligned}
$$

Proof of part (2):

T'' is a U_{2k}-decision tree that uses a transformation $\hat{T}$ of T. Each query node of T is replaced by a sub-tree that continuously performs queries until either obtaining a variable that was not queried before or until reaching a total of $2k$ queries. If the latter happens, T'' outputs an arbitrary value and stops. Note that this sub-tree is deterministic (i.e., does not consist of random coin nodes), and therefore when viewing $\hat{T}$ as a distribution $\hat{\mu}$ over deterministic decision trees $\hat{T}_1,\ldots,\hat{T}_m$, $\hat{\mu}=\mu$, and each $\hat{T}_j$ is obtained from T_j by the same transformation described above.

Given input x, T'' picks $2k$ variables $I=i_1,\ldots,i_{2k}$ uniformly at random with replacement and a decision tree $\hat{T}_j$ according to μ, and runs $\hat{T}_j^I$ on x. T'' is clearly a U_{2k}-decision tree of worst-case query complexity $2k$. T'' is variable-aware, since it uses the query variables to determine whether each new query variable occurred already before or not. We next show that if $k\le n/2-\sqrt{n/2\delta}$, then T'' $(\epsilon,2\delta)$-approximates f and that $S^e(T'')\le 2k_e$.

Again, we define E to be the event that at least k of the $2k$ variables chosen are distinct and $\overline{E}$ to be its complement. For a variable sequence $I\in[n]^{2k}$ that has at least k distinct variable, denote by $\hat{I}$ the sub-sequence of I consisting of the first k distinct variables. Note that conditioning on E, the distribution of $\hat{I}$ is uniform without replacement. Note also that for I with at least k distinct variables and for every T_j, ${\hat{T}_j}^I(x)=T_j^{\hat{I}}(x)$. Therefore,

$$
\begin{aligned}
&\Pr_{I\in U_{2k},\hat{T}_j}(\hat{T}_j^I(x)\notin C_{f,\epsilon}(x)) = \\
&= \Pr_{I\in U_{2k},\hat{T}_j}(\hat{T}_j^I(x)\notin C_{f,\epsilon}(x) \mid E)\cdot\Pr_{I\in U_{2k}}(E) + \Pr_{I\in U_{2k},T_j}(\hat{T}_j^I(x)\notin C_{f,\epsilon}(x)\mid\overline{E})\cdot\Pr_{I\in U_{2k}}(\overline{E}) \\
&\le \Pr_{\hat{I}\in W_k,T_j}(T_j^{\hat{I}}(x)\notin C_{f,\epsilon}(x)) + \Pr_{I\in U_{2k}}(\overline{E}) \;<\; \delta + \Pr_{I\in U_{2k}}(\overline{E})
\end{aligned}
$$

It thus suffices to prove that $\Pr_{I\in U_{2k}}(\overline{E}) \leq \delta$. In order to do that, we define as before $\binom{2k}{2}$ indicator random variables X_{jl}, such that $X_{jl} = 1$ iff $i_j = i_l$. If the list $i_1, \ldots, i_{2k}$ contains at most k distinct values, then $\sum_{j,l} X_{jl} \geq k$ (because the way to create the least number of collisions is to have two occurrences for each of the k values). On the other hand, the expectation of $\sum_{j,l} X_{jl}$ is $\binom{2k}{2}\frac{1}{n} \leq \frac{2k^2}{n}$. Using the pairwise independence of X_{jl} and Chebyshev's inequality we obtain:

$$
\begin{aligned}
\Pr(\overline{E}) &\leq \Pr(\sum_{j,l} X_{jl} \geq k) \leq \Pr(\left|\sum_{j,l} X_{jl} - E(\sum_{j,l} X_{jl})\right| \geq k - \frac{2k^2}{n}) \leq \\
&\leq \frac{\mathrm{Var}(\sum_{j,l} X_{jl})}{k^2(1-\frac{2k}{n})^2} = \frac{\binom{2k}{2}\frac{1}{n}\frac{n-1}{n}}{k^2(1-\frac{2k}{n})^2} \leq \frac{\frac{2k^2}{n}}{k^2(1-\frac{2k}{n})^2} = \frac{2}{n(1-\frac{2k}{n})^2}
\end{aligned}
$$

The last expression is at most δ for $k \leq n/2 - \sqrt{n/2\delta}$.

We now show that $S^e(T'') \leq 2k_e$. The main observation here is that the expected number of U-queries needed to get each W-query (i.e., a variable that did not occur before) is at most 2, because there are at least $2k$ possible variables, and T uses at most k of them. Therefore,

$$
\begin{aligned}
S^e(T'') &= \max_x E_{I\in U_{2k},\hat{T}_j}(S(\hat{T}_j^I, x)) = \max_x E_{\hat{T}_j}(E_{I\in U_{2k}}(S(\hat{T}_j^I, x))) \\
&\leq \max_x E_{\hat{T}_j}(2 \cdot E_{\hat{I}\in W_k}(S(T_j^{\hat{I}}, x))) = 2 \cdot \max_x E_{\hat{I}\in W_k, T_j}(S(T_j^{\hat{I}}, x)) = 2k_e
\end{aligned}
$$

In the inequality we used Wald's identity (cf. [Sie85]) stating that if $z_1, z_2, \ldots$ is a sequence of i.i.d. random variables with expectation e and S is a "stopping rule" random variable on $\{1, 2, \ldots\}$ such that the event $\{S = t\}$ is independent of $z_{t+1}, z_{t+2}, \ldots$, then $E(\sum_{i=1}^{S} z_i) = e \cdot E(S)$. In our case z_i is the number of U-queries needed in the i-th query node of T_j to obtain a variable that did not occur before. $S = t$, if the first t queries lead to a leaf. $\square$

Combining Lemma 4.8 and Lemma 4.10, we obtain the main result of this section:

Theorem 4.11 *Let $f : A^n \to B$ be an ϵ-symmetric function with $R^w_{\epsilon,\delta}(f) = k$. Then:*

(1) if $\frac{1}{2n} < \delta < \frac{1}{2}$ and $k \leq \sqrt{2\delta n}$, there exists a variable-oblivious U_k-decision tree of worst-case query complexity k that $(\epsilon, 2\delta)$-approximates f.

(2) if $\frac{2}{n} < \delta < \frac{1}{2}$ and $k \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$, there exists a variable-aware U_{2k}-decision tree of worst-case query complexity $2k$ that $(\epsilon, 2\delta)$-approximates f.

4.3 Approximation Implies Distinguishability

The main lemma of this section proves that any D_k-decision tree that approximates a function f can be used to distinguish between the full query distributions $\tilde{D}_{k,x}$ and $\tilde{D}_{k,y}$ of any two ϵ-disjoint inputs $x, y \in A^n$. Moreover, if the tree happens to be variable-oblivious, it can be used also to distinguish between the query outcome distributions $D_{k,x}$ and $D_{k,y}$.

Lemma 4.12 *Let T be a D_k-decision tree that (ϵ, δ)-approximates a function $f : A^n \to B$. Then for any two ϵ-disjoint inputs $x, y \in A^n$, $d(\tilde{D}_{k,x}, \tilde{D}_{k,y}) \geq 1 - 2\delta$. Furthermore, if T is variable-oblivious, then also $d(D_{k,x}, D_{k,y}) \geq 1 - 2\delta$.*

Proof. The basic idea underlying the proof will be to show that the distribution of values read off the input by the random queries along the tree's paths is very different when x is the input from when y is the input.

Without loss of generality, we assume that T is *regular*, that is, it always makes exactly k queries.

Recall that when T is variable-aware it maps the query sequence I, the query outcomes $x|_I$, and the random choices of the tree R into a value in the range B. When T is variable-oblivious it maps only $x|_I$ and R into B. We define: $J_x = (I, x|_I)$ and $O_x = x|_I$. In the following we show that when T is variable-aware the distributions of J_x and J_y are far from each other for ϵ-disjoint x and y, and similarly for O_x and O_y when T is variable-oblivious. Since the argument for both cases is the same, we generically view T as mapping a pair (S_x, R) to B, and we will argue that the distance between the distributions of S_x and S_y is large. We denote the distribution of S_x by P_x, and its domain by S (in the first case $P_x = \tilde{D}_{k,x}$ and $S = ([n] \times A)^k$, and in the second case $P_x = D_{k,x}$ and $S = A^k$).

Let $q_z \stackrel{\text{def}}{=} \Pr_{S_z,R}(T(S_z, R) \notin C_{f,\epsilon}(z))$ for an input z. By definition, $q_z \leq \delta$ for every input z. We write q_z as $q_z = \sum_{s \in S} P_z(s) Q_z(s)$, where $Q_z(s) \stackrel{\text{def}}{=} \Pr_R(T(s, R) \notin C_{f,\epsilon}(z))$.

Note that since $C_{f,\epsilon}(x) \cap C_{f,\epsilon}(y) = \emptyset$, we have $Q_y(s) \geq \Pr_R(T(s, R) \in C_{f,\epsilon}(x)) = 1 - Q_x(s)$.

We can rewrite the sum $q_x + q_y$ as follows:

$$\begin{aligned}
q_x + q_y &= \sum_{s \in S} P_x(s) Q_x(s) + P_y(s) Q_y(s) \\
&\geq \sum_{s \in S} P_x(s) Q_x(s) + P_y(s)(1 - Q_x(s)) \\
&\geq \sum_{s \in S} \min(P_x(s), P_y(s)) \\
&= \frac{1}{2} \sum_{s \in S} P_x(s) + P_y(s) - |P_x(s) - P_y(s)| \\
&= 1 - d(P_x, P_y)
\end{aligned}$$

Since $q_x + q_y \leq 2\delta$, then $d(P_x, P_y) \geq 1 - 2\delta$. □

4.4 Sample Complexity for Distinguishing Distributions

The main aim of this section is to obtain a lower bound on the number of samples required in order to distinguish between two distributions. We will use two measures of distance between distributions: the variational distance and the Hellinger distance.

Variational distance is a measure of distinguishability between distributions. In order to get a lower bound on the number of samples required to distinguish two distributions P and Q with some advantage α, we need to find a lower bound on the minimal integer k for which $d(P^k, Q^k) \geq \alpha$. Here, P^k and Q^k are distributions on Ω^k obtained by picking k independent random samples from P and Q respectively. However, variational distance does not behave well under product distributions, and in particular there is no simple formula that describes $d(P^k, Q^k)$ in terms of $d(P, Q)$. We therefore use the Hellinger distance, which has a nice multiplicativity property and bounds the variational distance from both sides:

Proposition 4.13

(1) $1 - h^2(P^k, Q^k) = (1 - h^2(P, Q))^k$

(2) $h^2(P, Q) \;\leq\; d(P, Q) \;\leq\; h(P, Q)\sqrt{2 - h^2(P, Q)}$.

The proof appears, e.g., in [LY90]. We repeat it here for completeness:

Proof. Proof of (1):

$$\begin{aligned}
1 - h^2(P^k, Q^k) &= \sum_{\omega_1,\ldots,\omega_k \in \Omega} \sqrt{P^k(\omega_1, \ldots, \omega_k) Q^k(\omega_1, \ldots, \omega_k)} \\
&= \sum_{\omega_1,\ldots,\omega_k \in \Omega} \sqrt{P(\omega_1) \cdots P(\omega_k) \cdot Q(\omega_1) \cdots Q(\omega_k)} \\
&= (\sum_{\omega \in \Omega} \sqrt{P(\omega) Q(\omega)})^k \\
&= (1 - h^2(P, Q))^k
\end{aligned}$$

Proof of (2):

$$\begin{aligned}
h^2(P, Q) &= \frac{1}{2} \sum_{\omega \in \Omega} (\sqrt{P(\omega)} - \sqrt{Q(\omega)})^2 \;\leq\; \frac{1}{2} \sum_{\omega \in \Omega} \left| \sqrt{P(\omega)} - \sqrt{Q(\omega)} \right| (\sqrt{P(\omega)} + \sqrt{Q(\omega)}) \\
&= \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)| \;=\; d(P, Q)
\end{aligned}$$

Using the Cauchy-Schwartz inequality:

$$\begin{aligned}
d(P, Q) &= \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)| \;=\; \frac{1}{2} \sum_{\omega \in \Omega} \left| \sqrt{P(\omega)} - \sqrt{Q(\omega)} \right| (\sqrt{P(\omega)} + \sqrt{Q(\omega)}) \\
&\leq \sqrt{\frac{1}{2} \sum_{\omega \in \Omega} (\sqrt{P(\omega)} - \sqrt{Q(\omega)})^2 \cdot \frac{1}{2} \sum_{\omega \in \Omega} (\sqrt{P(\omega)} + \sqrt{Q(\omega)})^2} \\
&= \sqrt{h^2(P, Q)} \cdot \sqrt{\frac{1}{2} (\sum_{\omega \in \Omega} P(\omega) + \sum_{\omega \in \Omega} Q(\omega) + 2 \sum_{\omega \in \Omega} \sqrt{P(\omega) Q(\omega)})} \\
&= h(P, Q) \cdot \sqrt{2 - h^2(P, Q)}
\end{aligned}$$

$\square$

The following lemma provides the desired lower bound in terms of the distributions' Hellinger distance. It was pointed out to us by David Zuckerman [personal communication, September 2000].

Lemma 4.14 *Let P and Q be two distributions on Ω with $h^2(P,Q) \leq 1/2$, let $0 < \alpha < 1$, and let k be an integer such that $d(P^k, Q^k) \geq \alpha$. Then:*

$$k \geq \frac{1}{4h^2(P,Q)} \ln \frac{1}{1-\alpha^2}$$

Proof. By Proposition 4.13 (part (2)), we have

$$h(P^k, Q^k)\sqrt{2 - h^2(P^k, Q^k)} \;\geq\; d(P^k, Q^k) \;\geq\; \alpha$$

which implies that $h^4(P^k, Q^k) - 2h^2(P^k, Q^k) + \alpha^2 \leq 0$. The solution to this quadratic inequality gives $h^2(P^k, Q^k) \geq 1 - \sqrt{1-\alpha^2}$.

By Proposition 4.13 (part (1)), we have $1 - (1 - h^2(P,Q))^k \geq 1 - \sqrt{1-\alpha^2}$. Using the inequality $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$, we have:

$$k \geq \frac{\ln \frac{1}{\sqrt{1-\alpha^2}}}{\ln \frac{1}{1-h^2(P,Q)}} \geq \frac{1}{4h^2(P,Q)} \ln \frac{1}{1-\alpha^2}$$

□

4.5 The Lower Bound

We are now ready to prove our main theorem:

Theorem 4.4 (restated) *For every $\epsilon \geq 0$ and for every ϵ-symmetric function $f : A^n \to B$,*

(1) if $\frac{1}{2n} < \delta < \frac{1}{2}$, $R^w_{\epsilon,\delta}(f) \leq \sqrt{2\delta n}$, and $h_\epsilon(f) \leq \frac{1}{2}$, then

$$R^w_{\epsilon,\delta}(f) \geq \frac{1}{4h^2_\epsilon(f)} \ln \frac{1}{8\delta}$$

(2) if $\frac{2}{n} < \delta < \frac{1}{2}$, $R^w_{\epsilon,\delta}(f) \leq n/2 - \sqrt{n/2\delta}$, and $d_\epsilon(f) \leq \frac{1}{2}$, then

$$R^w_{\epsilon,\delta}(f) \geq \frac{1}{8d_\epsilon(f)} \ln \frac{1}{8\delta}$$

Proof. Let $k = R^w_{\epsilon,\delta}(f)$. We prove each of the two parts separately:

Proof of part (1):

By part (1) of Theorem 4.11, if $k \le \sqrt{2\delta n}$, there exists a variable-oblivious U_k-decision tree T that $(\epsilon, 2\delta)$-approximates f. Let x and y be the ϵ-disjoint inputs that achieve the function's minimum Hellinger distance: $h(U_x, U_y) = h_\epsilon(f)$. Lemma 4.12 implies that $d(U_{k,x}, U_{k,y}) \ge 1 - 4\delta$. Since $U_{k,x}$ and $U_{k,y}$ are the k-fold products of U_x and U_y, and since $h^2(U_x, U_y) \le 1/2$, we can apply the distinguishability lower bound (Lemma 4.14) to complete the proof of this part.

Proof of part (2):

By part (2) of Theorem 4.11, if $k \le \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$, then there exists a variable-aware U_{2k}-decision tree T that $(\epsilon, 2\delta)$-approximates f. Let x and y be the ϵ-disjoint inputs that achieve the function's minimum variational distance: $d(U_x, U_y) = d_\epsilon(f)$. Lemma 4.12 implies that $d(\tilde{U}_{2k,x}, \tilde{U}_{2k,y}) \ge 1 - 4\delta$.

We denote by $\tilde{U}_x$ and $\tilde{U}_y$ the distributions on $[n] \times A$ obtained by picking $i \in [n]$ uniformly at random and returning the pairs (i, x_i) and (i, y_i) respectively. Clearly, $\tilde{U}_{2k,x}$ and $\tilde{U}_{2k,y}$ are the $2k$-fold product distributions of $\tilde{U}_x$ and $\tilde{U}_y$. We next prove two simple facts about the distance between $\tilde{U}_x$ and $\tilde{U}_y$.

Claim 4.15 *For all inputs $x, y \in A^n$,*

$$h^2(\tilde{U}_x, \tilde{U}_y) = d(\tilde{U}_x, \tilde{U}_y) = \mathrm{Ham}(x, y)$$

Proof. By definition (see Definition 4.2), we have:

$$h^2(\tilde{U}_x, \tilde{U}_y) = \frac{1}{2} \sum_{i \in [n], a \in A} \left(\sqrt{\tilde{U}_x(i,a)} - \sqrt{\tilde{U}_y(i,a)}\right)^2$$

$$d(\tilde{U}_x, \tilde{U}_y) = \frac{1}{2} \sum_{i \in [n], a \in A} |\tilde{U}_x(i,a) - \tilde{U}_y(i,a)|$$

Every term in each of the above sums corresponds to a pair $(i, a) \in [n] \times A$. In both sums, this term is $1/n$ if and only if exactly one of x_i, y_i equals a, and is 0 otherwise. Thus, each variable $i \in [n]$ for which $x_i \ne y_i$ contributes $2/n$ to each of the two sums ($1/n$ in the term (i, x_i) and $1/n$ in the term (i, y_i)), while each variable i for which $x_i = y_i$ contributes 0. It follows that:

$$h^2(\tilde{U}_x, \tilde{U}_y) = d(\tilde{U}_x, \tilde{U}_y) = \frac{1}{2} \cdot \frac{2}{n} \cdot |\{i \in [n] \mid x_i \ne y_i\}| = \mathrm{Ham}(x, y)$$

□

Claim 4.16 *For all inputs $x, y \in A^n$,*

$$d(U_x, U_y) = \min\{d(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}) \mid \pi, \sigma \in S_n\}$$

Proof.

$\le$:

Since U_x and U_y can obtained from $\tilde{U}_x$ and $\tilde{U}_y$ by projecting on their second coordinate, then $d(U_x, U_y) \leq d(\tilde{U}_x, \tilde{U}_y)$. Let $\pi^*, \sigma^* \in S_n$ be the permutations in which the minimum of $d(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)})$ is obtained. Note that $d(U_x, U_y)$ is invariant under permutations: for all $\pi, \sigma \in S_n$, $d(U_x, U_y) = d(U_{\pi(x)}, U_{\sigma(y)})$. Thus,

$$d(U_x, U_y) = d(U_{\pi^*(x)}, U_{\sigma^*(y)}) \leq d(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}) = \min\{d(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}) \mid \pi, \sigma \in S_n\}$$

$\geq$:

We first describe a procedure to permute x and y. We then claim that the obtained permutations π^* and σ^* satisfy $d(U_x, U_y) \geq d(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)})$.

The main idea is to move all the shared elements of x and y to their front, such that they will have a maximal common prefix. Formally, for each $a \in A$, denote by $f_x(a)$ and $f_y(a)$ the frequencies of a in x and y respectively (i.e., $f_x(a) = |\{i \in [n] \mid x_i = a\}|$ and similarly for $f_y(a)$). Let $m(a) = \min(f_x(a), f_y(a))$ and $M(a) = \max(f_x(a), f_y(a))$. Fix some arbitrary order on A, and permute x and y as follows: enumerate the elements in $a \in A$ that have at least one occurrence in either x or y from smallest to largest according to the fixed order; for each such a, move its first $m(a)$ occurrences in x and in y to the front of x and y respectively. Note that after this procedure ends, x and y have a common prefix of length $m \stackrel{\text{def}}{=} \sum_{a \in A} m(a)$; their two suffixes (of length $n-m$) are different in every position.

Denote by π^* and σ^* the permutations this procedure induces on x and y respectively. By Claim 4.15, $d(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)})$ equals the relative Hamming distance between $\pi^*(x)$ and $\sigma^*(y)$. Therefore,

$$d(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}) = \frac{n-m}{n}$$

On the other hand, if we define $M \stackrel{\text{def}}{=} \sum_{a \in A} M(a)$, then we have:

$$d(U_{\pi^*(x)}, U_{\sigma^*(y)}) = d(U_x, U_y) = \frac{1}{2} \sum_{a \in A} |U_x(a) - U_y(a)| = \frac{1}{2} \sum_{a \in A} \frac{M(a) - m(a)}{n} = \frac{M - m}{2n}$$

Note that $M + m = 2n$ (since $M + m$ is the sum of frequencies of all $a \in A$ in x and y), implying that $(M - m)/2n = (n - m)/n$. Therefore,

$$d(U_x, U_y) = d(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}) \geq \min\{d(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}) \mid \pi, \sigma \in S_n\}$$

□

Going back to the proof of the theorem, we pick permutations $\pi, \sigma \in S_n$, such that $d(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}) = d(U_x, U_y)$. Note that $d(U_{\pi(x)}, U_{\sigma(y)}) = d(U_x, U_y) = d_\epsilon(f)$. Thus, without loss of generality, we can assume that x and y themselves were chosen such that $d(U_x, U_y) = d(\tilde{U}_x, \tilde{U}_y)$.

Since $\tilde{U}_{2k,x}$ and $\tilde{U}_{2k,y}$ are the $2k$-fold product distributions of $\tilde{U}_x$ and $\tilde{U}_y$, and since $h^2(\tilde{U}_x, \tilde{U}_y) = d(\tilde{U}_x, \tilde{U}_y) = d(U_x, U_y) \leq 1/2$, then we can apply the distinguishability lower bound, and obtain:

$$k \geq \frac{1}{8h^2(\tilde{U}_x, \tilde{U}_y)} \ln \frac{1}{8\delta} = \frac{1}{8d(U_x, U_y)} \ln \frac{1}{8\delta}$$

□

4.6 Expected Query Complexity vs. Worst-Case Query Complexity

In this section we show that the worst-case query complexity of any function is at most a factor of $O(\log\frac{1}{\delta})$ away from its expected query complexity:

Proposition 4.17 *For every function $f : A^n \to B$, every $\epsilon \geq 0$, and every $0 < \delta < 1/3$,*

$$R^w_{\epsilon,3\delta}(f) \;\leq\; 2R^e_{\epsilon,\delta}(f) \cdot \lceil \log \frac{1}{\delta} \rceil$$

Proof. Let T be a decision tree of expected query complexity k that (ϵ, δ)-approximates f. We show that there exists a decision tree T' that $(\epsilon, 3\delta)$-approximates f, and has worst-case query complexity of $2k\lceil \log \frac{1}{\delta} \rceil$.

Given input x, T' runs at most $d \stackrel{\text{def}}{=} \lceil \log \frac{1}{\delta} \rceil$ independent simulations of T one after the other. Each simulation is carried out for at most $2k$ queries. If at any simulation T outputs a value before the $2k + 1$-st query, T' outputs the same value and stops. Otherwise, T' continues to the next simulation. If none of the d simulations outputs a value by the $2k$-th query, T' outputs an arbitrary value and stops.

It is clear that T' has a worst-case query complexity of at most $2kd = 2k\lceil \log \frac{1}{\delta} \rceil$. We are left to show that it $(\epsilon, 3\delta)$-approximates f. Fix some input x, and let $T_1, \ldots, T_d$ be the d simulations of T. We call T_i "bad" if $S(T_i, x) > 2k$ (that is, T_i needs more than $2k$ queries). We denote by B_i the event that T_i is bad. Since $E(S(T_i, x)) = S^e(T_i) = k$, then by Markov's inequality, $\Pr(B_i) = Pr(S(T_i, x) > 2k) < 1/2$. We denote by C_i the event that $S(T_i, x) \leq 2k$ and $T_i(x) \notin C_{f,\epsilon}(x)$. Since T_i (ϵ, δ)-approximates f, $\Pr(C_i) \leq \Pr(T_i(x) \notin C_{f,\epsilon}(x)) < \delta$.

T' fails if and only if one of the following disjoint $i + 1$ events occurs: $C_1, B_1 \wedge C_2, B_1 \wedge B_2 \wedge C_3, \ldots, \bigwedge_{i=1}^{d-1} B_i \wedge C_d, \bigwedge_{i=1}^{d} B_i$. Using the mutual independence of the simulations we have:

$$\Pr(T'(x) \notin C_{f,\epsilon}(x)) = \sum_{i=1}^{d} \left(\prod_{j=1}^{i-1} \Pr(B_j) \cdot \Pr(C_{i+1}) \right) + \prod_{j=1}^{d} \Pr(B_j) < \sum_{i=1}^{d} \frac{1}{2^{i-1}} \cdot \delta + \frac{1}{2^d} \leq 3\delta$$

$\square$

Theorem 4.4 and Proposition 4.17 immediately imply now Theorem 4.5.

4.7 Lower Bound on the Deterministic Query Complexity

Theorem 4.7 (restated) *For every $\epsilon \geq 0$ and for every ϵ-symmetric function $f : A^n \to B$,*

$$D_\epsilon(f) > n \cdot (1 - \mathrm{Ham}_\epsilon(f))$$

Proof. Let x and y be the two ϵ-disjoint inputs that achieve the minimum Hamming distance: $\mathrm{Ham}(x, y) = \mathrm{Ham}_\epsilon(f)$. Let $t \stackrel{\text{def}}{=} n(1 - \mathrm{Ham}(x, y))$, and let $I = (i_1, \ldots, i_t)$ be the sequence of variables on which x and y agree.

Let T be the optimal deterministic decision tree that ϵ-approximates f. Consider the path of x in T^I, the I-relabeling of T. By Proposition 4.9, there is a permutation $\pi \in S_n$, such that $\pi(x)$ has the same path in T as the path of x in T^I. Similarly, there is a permutation $\sigma \in S_n$ such that $\sigma(y)$ has the same path in T as the path of y in T^I.

Note that since the first t variables queried in T^I are always $i_1, \ldots, i_t$, and since x and y agree on these variables, the first $t+1$ nodes on the path of x in T^I are identical to the first $t+1$ nodes on the path of y in T^I. It follows that also the paths of $\pi(x)$ and $\sigma(y)$ in T share the same first $t+1$ nodes. Now, if the query complexity of T is at most t, then necessarily the paths of $\pi(x)$ and $\sigma(y)$ in T share the same leaf (output node). Therefore, $T(\pi(x)) = T(\sigma(y))$. However, since f is ϵ-symmetric and since x and y are ϵ-disjoint, then also $\pi(x)$ and $\sigma(y)$ are ϵ-disjoint. This implies that T outputs a wrong value either on $\pi(x)$ or on $\sigma(y)$. Therefore, $S(T) > t = n(1 - \mathrm{Ham}_\epsilon(f))$. □

5 Applications

In this section we apply the lower bound technique obtained in the previous section to derive lower bounds on the query complexity of specific functions. For brevity, we state only the lower bounds on the worst-case query complexity. Identical lower bounds, modulo the dependence on the confidence δ, hold also for the expected query complexity.

5.1 Statistical Moments

The k-th statistical moment, $\mu_k : [0,1]^n \to \mathbf{R}$, is defined as $\mu_k(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^n x_i^k$. The first moment, for example, is simply the mean. An additive approximation of μ_k can be easily performed by taking a random sample of size $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$, and computing the sample's k-th moment. Chernoff-Hoeffding bound [Che52, Hoe63] implies that this yields an ϵ-additive approximation with probability $1-\delta$. Canetti *et al.* [CEG95] prove that the mean (and, implicitly, all the other moments too) require $\Omega(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$ samples. Using Theorem 4.4 we obtain a much simpler proof for this result:

Theorem 5.1 *For all* $\frac{1}{2n} < \delta < \frac{1}{2}$ *and* $\Omega(\sqrt{\frac{\ln(1/\delta)}{\sqrt{\delta n}}}) < \epsilon \leq \frac{1}{\sqrt{8}}$,

$$R^w_{\epsilon,\delta}(\mu_k) \geq \frac{1}{64\epsilon^2}\ln\frac{1}{8\delta}$$

Proof. Consider the two inputs (or equivalently, distributions) x and y. x is defined to be 0 with probability $1/2-\epsilon$ and 1 otherwise; y is defined to be 0 with probability $1/2+\epsilon$ and 1 otherwise. Since $\mu_k(x) = 1/2+\epsilon$ and $\mu_k(y) = 1/2-\epsilon$, x and y are ϵ-disjoint. We bound the Hellinger distance of x and y as follows: $h^2(U_x, U_y) = 1 - 2\sqrt{(1/2+\epsilon)(1/2-\epsilon)} = 1 - \sqrt{1-4\epsilon^2} \leq 1 - (1-4\epsilon^2) = 4\epsilon^2$. The lower bound follows from part (1) of Theorem 4.4. □

5.2 Frequency Moments

Given a sequence $x = x_1, \ldots, x_n$ where $x_i \in [m]$, the k-th frequency moment of x is defined to be $F_k(x) \stackrel{\text{def}}{=} \sum_{i=1}^{m} f_i^k$, where $f_i = |\{j \mid x_j = i\}|$, the number of occurrences of i in the sequence. We prove upper and lower bounds for a relative approximation of F_k that are almost matching (in terms of n and m).

Theorem 5.2 *Let $r = \min(n, m)$. For all $k \geq 2$, $\frac{2}{n} < \delta < \frac{1}{2}$, and $\Omega(\sqrt{\frac{2^k \cdot k! \cdot \log(1/\delta)}{n^{1/k}}}) < \epsilon < \frac{1}{3}$,*

$$
\begin{aligned}
R^w_{\epsilon,\delta}(F_k) &\geq \Omega\left(\frac{r^{1-1/k}}{\epsilon^{1/k}} \log \frac{1}{\delta}\right) \\
R^w_{\epsilon,\delta}(F_k) &\leq O\left(\frac{2^k \cdot k! \cdot r^{1-1/k}}{\epsilon^2} \log \frac{1}{\delta}\right)
\end{aligned}
$$

Proof. For the lower bound, we will use part (2) of Theorem 4.4, since we do not have an upper bound of $O(\sqrt{n})$ on $R^w_{\epsilon,\delta}(F_k)$. For simplicity of exposition, we assume $m \geq n$. The case $m < n$ is almost identical. Consider the following two inputs: the first input is $x = 1, \ldots, n$. The second input is constructed from x as follows. Divide the input of length n into $n^{1-1/k}/(3\epsilon)^{1/k}$ blocks of size $(3\epsilon n)^{1/k}$. y is x except in the first block where it is all 1's. Clearly, $F_k(x) = n$ while $F_k(y) = 3\epsilon n + (n - (3\epsilon n)^{1/k}) = n(1 + 3\epsilon) - (3\epsilon n)^{1/k}$, implying x and y are ϵ-disjoint. Since $d(U_x, U_y) \leq O(\epsilon^{1/k}/n^{1-1/k})$, the lower bound follows from Theorem 4.4.

Now, we show that this lower bound is almost tight. Note that for any input x, $U_x(i) = q_i/n$. Thus, computing $F_k(x)$ is equivalent to estimating the k-wise collision probability of U_x, since $F_k(x) = \sum_{i=1}^{m} q_i^k = n^k \sum_{i=1}^{n} U_x(i)^k$. So, the sampling algorithm picks ℓ samples $x_1, \ldots, x_\ell$ using U_x and counts the k-wise collisions C_k, our basic estimator. It is clear that $E(C_k) = \binom{\ell}{k} \|U_x\|_k^k$. The following lemma, whose proof generalizes that of [GR00] for F_2, helps us to bound the variance of this estimator.

Lemma 5.3 $\mathrm{Var}(C_k) \leq 2^k \cdot (k!)^{1-1/k} \cdot E(C_k)^{2-1/k}$.

Proof. For every subset $I \subseteq [\ell]$ of size k, we define X_I to be a 0-1 random variable that gets the value 1 iff all the values $\{x_i\}_{i \in I}$ are the same (i.e., I forms a collision). Clearly, $C_k = \sum_{I \subseteq [\ell], |I|=k} X_I$ and $E(X_I) = \|U_x\|_k^k$.

Define $\overline{X}_I \stackrel{\text{def}}{=} X_I - \|U_x\|_k^k$. Note that for $I \neq J$, $I \cap J = \emptyset$, X_I and X_J are independent, and therefore $E(\overline{X}_I \cdot \overline{X}_J) = E(\overline{X}_I) \cdot E(\overline{X}_J) = 0$. If $I \cap J \neq \emptyset$, then X_I and X_J do depend on each other, but $X_I \cdot X_J$ is a 0-1 random variable that gets the value 1 iff all the values in $I \cup J$ are the same. Thus, if $|I \cup J| = k + t$ then $E(\overline{X}_I \cdot \overline{X}_J) = E(X_I \cdot X_j) - \|U_x\|_k^{2k} = \|U_x\|_{k+t}^{k+t} - \|U_x\|_k^{2k}$. We have:

$$
\begin{aligned}
\mathrm{Var}(C_k) &= E\left((\sum_I \overline{X}_I)^2\right) = \sum_{I,J} E(\overline{X}_I \cdot \overline{X}_J) = \sum_{t=0}^{k-1} \sum_{I,J,|I \cup J|=k+t} \|U_x\|_{k+t}^{k+t} - \|U_x\|_k^{2k} \\
&\leq \sum_{t=0}^{k-1} \binom{\ell}{k} \cdot \binom{\ell-k}{t} \cdot \binom{k}{t} \cdot \|U_x\|_{k+t}^{k+t}
\end{aligned}
$$

Note that $\|U_x\|_{k+t}^{k+t} \leq \|U_x\|_\infty^t \cdot \|U_x\|_k^k \leq \|U_x\|_k^t \cdot \|U_x\|_k^k = \|U_x\|_k^{k+t}$. Therefore,

$$\mathrm{Var}(C_k) \;\leq\; \binom{\ell}{k}\|U_x\|_k^k \cdot \sum_{t=0}^{k-1}\binom{k}{t} \cdot \binom{\ell-k}{t} \cdot \|U_x\|_k^t \;\leq\; \binom{\ell}{k}\|U_x\|_k^k \cdot \sum_{t=0}^{k-1}\binom{k}{t} \cdot \frac{(\ell-k)^t}{t!} \cdot \|U_x\|_k^t$$

Since for all $k \geq 2$ and $0 \leq t \leq k-1$, $k \leq 2t!(k-t)$, then $\binom{k}{t}/t! \leq 2\binom{k-1}{t}$. Therefore,

$$\mathrm{Var}(C_k) \;\leq\; 2\binom{\ell}{k}\|U_x\|_k^k \cdot \sum_{t=0}^{k-1}\binom{k-1}{t} \cdot (\ell-k)^t \cdot \|U_x\|_k^t \;=\; 2\binom{\ell}{k}\|U_x\|_k^k \cdot ((\ell-k)\cdot\|U_x\|_k + 1)^{k-1}$$

Note that $\|U_x\|_k^k \geq 1/r^{k-1}$ (because $F_k(x) \geq n$ and $(1/m)F_k(x) \geq (n/m)^k$ for all x). We will choose ℓ to be large enough such that $\ell - k \geq r^{1-1/k}$. Therefore, $(\ell-k)\cdot\|U_x\|_k \geq 1$. We thus obtain:

$$\begin{aligned}
\mathrm{Var}(C_k) \;&\leq\; 2\binom{\ell}{k}\|U_x\|_k^k \cdot (2(\ell-k)\cdot\|U_x\|_k)^{k-1} \;=\; 2^k\binom{\ell}{k}\|U_x\|_k^k \cdot (\ell-k)^{k-1}\cdot\|U_x\|_k^{k-1} \\
&\leq\; 2^k\binom{\ell}{k}\|U_x\|_k^k \cdot (k!\cdot\binom{\ell}{k}\cdot\|U_x\|_k^k)^{(k-1)/k} \;=\; 2^k\cdot(k!)^{1-1/k}\cdot E(C_k)^{2-1/k}
\end{aligned}$$

□

Since $\|U_x\|_k^k \geq 1/r^{k-1}$, then by Chebyshev's inequality,

$$\Pr(|C_k - E(C_k)| \geq \epsilon E(C_k)) \;\leq\; \frac{2^k\cdot(k!)^{1-1/k}}{\epsilon^2 E(C_k)^{1/k}} \;\leq\; \frac{2^k\cdot(k!)^{1-1/k}\cdot r^{(k-1)/k}\cdot(k!)^{1/k}}{(\ell-k)\cdot\epsilon^2}$$

For this to be a constant, say, 3/4, we get $\ell \geq \Omega(2^k \cdot k! \cdot r^{1-1/k}/\epsilon^2)$. Now, this experiment can be repeated $O(\log 1/\delta)$ times and the median of these various trials is correct with probability at least $1-\delta$. □

We next give a simple new proof for (a slightly weaker version of) the sampling lower bound of Charikar *et al.* [CCMN00] for a ratio approximation of F_0:

Theorem 5.4 *For all $\frac{2}{n} < \delta < \frac{1}{2}$ and $0 \leq \epsilon \leq 1 - \sqrt{\frac{2}{n}}$,*

$$R^w_{\epsilon,\delta}(F_0) \geq \min(\frac{n}{2} - \sqrt{n/2\delta}\ ,\ \Omega(n\cdot(1-\epsilon)^2\cdot\ln\frac{1}{\delta}))$$

Proof. If $R^w_{\epsilon,\delta}(F_0) \geq n/2 - \sqrt{n/2\delta}$ we are done. Otherwise, we use part (2) of Theorem 4.4.

Let x be the input that is all 1's and let y be the input that has $n - \lfloor 1/(1-\epsilon)^2\rfloor$ 1's, followed by $2, 3, \ldots, \lfloor 1/(1-\epsilon)^2\rfloor + 1$. Note that $F_0(z)$ counts the number of distinct elements in z, and thus $F_0(x) = 1, F_0(y) \geq 1/(1-\epsilon)^2$, implying x and y are ϵ-disjoint. Now, $d(U_x, U_y) = \lfloor 1/(1-\epsilon)^2\rfloor/n \leq 1/(n(1-\epsilon)^2)$. The lower bound follows from part (2) of Theorem 4.4. □

5.3 Selection Functions

Chernoff bound implies that the median of a random sample of size $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$ yields an ϵ-approximate median. The same result holds for Select_q for any constant $0 < q < 1$. However, for $q = 0$ (minimum) and for $q = 1$ (maximum), a sample of size $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ is sufficient. We next show that these bounds are tight:

Theorem 5.5 *For all* $\frac{1}{2n} < \delta < \frac{1}{2}$ *and* $\Omega(\sqrt{\frac{\ln(1/\delta)}{\sqrt{\delta n}}}) < \epsilon < \frac{1}{\sqrt{8}}$,

(1) For $\epsilon \le q \le 1-\epsilon$, $R^w_{\epsilon,\delta}(\text{Select}_q) \ge \Omega(\frac{q(1-q)}{\epsilon^2}\log\frac{1}{\delta})$

(2) For $q < \epsilon, q > 1-\epsilon$, $R^w_{\epsilon,\delta}(\text{Select}_q) \ge \Omega(\frac{1}{\epsilon}\log\frac{1}{\delta})$

Proof. Assume, initially, that $\epsilon \le q \le 1-\epsilon$. We pick $x, y \in \{0,1\}^n$ as follows: x has $(q+\epsilon)n$ 0's and the rest are 1's, and y has $(q-\epsilon)n$ 0's and the rest are 1's. Clearly, $C_{\text{Select}_{q,\epsilon}}(x) = \{0\}$ and $C_{\text{Select}_{q,\epsilon}}(y) = \{1\}$, implying x and y are ϵ-disjoint. Since $h^2(U_x, U_y) \le \epsilon^2/(q(1-q))$, the bound follows from part (1) of Theorem 4.4.

When $q < \epsilon$, we pick x as before and $y = 1^n$, and when $q > 1-\epsilon$, we pick y as before and $x = 0^n$. The fact that $h^2(U_x, U_y) \le 2\epsilon$ implies the lower bound. □

5.4 Extractor and Disperser Lower Bounds

Using the lower bounds for the mean and the maximum, we can immediately derive the lower bounds on the seed length of extractors and dispersers originally proved by Radhakrishnan and Ta-Shma [RT00]. Our bounds hold, however, only to extractors and dispersers with high min-entropy. We first recall the definitions of dispersers and extractors. In the definitions the *min-entropy* of a distribution P on a set Ω is defined as $H_\infty(P) = \min_{\omega\in\Omega}\log(1/P(\omega))$. U_r denotes the uniform distribution on $\{0,1\}^r$.

Definition 5.6 (Disperser [Sip88]) *A function* $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)-disperser*, if for all distributions* X *on* $\{0,1\}^n$ *with min-entropy at least* k *and for every subset* $S \subseteq \{0,1\}^m$ *of size at least* $\epsilon 2^m$, $\Pr(E(X, U_d) \in S) > 0$.

Definition 5.7 (Extractor [NZ96]) *A function* $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)-extractor*, if for all distributions* X *on* $\{0,1\}^n$ *with min-entropy at least* k, $d(E(X,U_d), U_m) < \epsilon$.

Zuckerman [Zuc97] showed that any extractor implies an (oblivious) sampling algorithm for the mean:

Lemma 5.8 (Zuckerman) *If there exists a* (k,ϵ)*-extractor* $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, *then there exists a sampling algorithm* T *of worst-case query complexity* 2^d *that* $(\epsilon, 1/2^{n-k-1})$*-approximates the mean* $\mu_1 : [0,1]^{2^m} \to [0,1]$.

A similar argument can show that any disperser implies a sampling algorithm for the maximum:

Lemma 5.9 *If there exists a (k, ϵ)-disperser $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, then there exists a sampling algorithm T of worst-case query complexity 2^d that $(\epsilon, 1/2^{n-k})$-approximates the maximum* $\max : \mathbf{R}^{2^m} \to \mathbf{R}$.

Proof. Given the disperser E, our sampling algorithm T for approximating max will work as follows: T picks a random $z \in \{0,1\}^n$, queries the input $x \in \mathbf{R}^{2^m}$ in the locations designated by $\{E(z,r)\}_{r \in \{0,1\}^d}$, and outputs the maximum of the values obtained. Clearly, T has worst-case query complexity 2^d. We next show it indeed $(\epsilon, 1/2^{n-k})$-approximates max.

Suppose not. Then there exists an input $x \in \mathbf{R}^{2^m}$ for which with probability at least $1/2^{n-k}$ over the choice of $z \in U_n$, $\max_{r \in \{0,1\}^d} x_{E(z,r)}$ is not one of the $\epsilon 2^m$ biggest elements of x. Let us denote by $S \subseteq \{0,1\}^m$ the set of indices of these $\epsilon 2^m$ biggest elements. We thus have:

$$\Pr_{z \in U_n}(\Pr_{r \in U_d}(E(z,r) \in S) = 0) \geq \frac{1}{2^{n-k}}$$

Therefore, if we denote by $R \subseteq \{0,1\}^n$ the set of z's for which $\Pr_{r \in U_d}(E(z,r) \in S) = 0$, we have $|R| \geq 2^n/2^{n-k} = 2^k$. Therefore, the uniform distribution on R, which we denote by X, has min-entropy k and satisfies: $\Pr(E(X, U_d) \in S) = 0$. This is a contradiction to E being a (k, ϵ)-disperser.
□

Using these connections and our lower bounds for the mean and the maximum, we can obtain a simple proof for extractor and disperser seed length lower bounds:

Theorem 5.10 *For all $n, m \leq n$, $n - m - 1 \leq k \leq n - 1$, and $\sqrt{\frac{n-k}{2^{(n+m-k)/2}}} < \epsilon < 1/\sqrt{8}$,*

(1) For every (k, ϵ)-disperser $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, $d \geq \log(n-k) + \log \frac{1}{\epsilon} - O(1)$.

(2) For every (k, ϵ)-extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, $d \geq \log(n-k) + 2\log \frac{1}{\epsilon} - O(1)$.

Note that the restriction on k together with the fact $m \leq k + t - 2\log(1/\epsilon)$ [RT00] implies that this lower bound holds only for $k \geq \Omega(n)$.

Proof. Using Lemma 5.9, the existence of a (k, ϵ)-disperser E implies the existence of a sampling algorithm T of worst-case query complexity 2^d that $(\epsilon, 1/2^{n-k})$-approximates $\max : \mathbf{R}^{2^m} \to \mathbf{R}$. Our sampling lower bound for the maximum (Theorem 5.5) implies that:

$$2^d \geq \Omega(\frac{1}{\epsilon}\log(2^{n-k})) = \Omega(\frac{1}{\epsilon}(n-k))$$

which derives part (1) of the theorem.

Similarly, Lemma 5.8 implies that the existence of a (k, ϵ)-extractor E yields a sampling algorithm T that $(\epsilon, 1/2^{n-k-1})$-approximates $\mu_1 : [0,1]^{2^m} \to [0,1]$ with 2^d queries. Our lower bound for the mean (Theorem 5.1) implies that:

$$2^d \geq \Omega(\frac{1}{\epsilon^2}\log(2^{n-k-1})) = \Omega(\frac{1}{\epsilon^2}(n-k-1))$$

which derives part (2) of the theorem. □

6 Tightness of the Bounds

In this section, we discuss the tightness of our bounds. First, we show that for symmetric functions the lower bound achieved in Section 4 is always at least as strong as the lower bound via block sensitivity.

Theorem 6.1 *For all $\epsilon \geq 0$ and every ϵ-symmetric function $f : A^n \to B$,*

$$\frac{1}{h_\epsilon^2(f)} \geq \frac{1}{d_\epsilon(f)} \geq \mathrm{bs}_\epsilon(f)$$

Proof. The first inequality follows from Proposition 4.13. For the second inequality let x be the input that achieves the block sensitivity of f, and let $I \subseteq [n]$ be the largest block to which f is ϵ-sensitive on x. Note that $\mathrm{bs}_\epsilon(f) = \mathrm{bs}_\epsilon(f,x) \leq n/|I|$.

Let $Q \in A^{|I|}$ be the assignment to I such that x and $y = x^{(I \leftarrow Q)}$ are ϵ-disjoint. Since x and y differ in $|I|$ positions, then $d(U_x, U_y) \leq |I|/n$. Therefore, by definition, $d_\epsilon(f) \leq |I|/n$. The theorem follows. $\square$

We now show that the lower bound achieved in Section 4 is reasonably tight for functions over small finite domains (e.g., functions over a Boolean domain). The proof, due to Luca Trevisan [personal communication, March 2001], improves a slightly weaker bound we achieved before.

Theorem 6.2 *If $f : A^n \to B$ is an ϵ-symmetric function with approximation $C_{f,\epsilon}$ that satisfies the sunflower property, then*

$$R^w_{\epsilon,\delta}(f) \leq O(|A| \frac{1}{d_\epsilon^2(f)} \ln \frac{1}{\delta})$$

Proof. Given input x, the algorithm for approximating f will try to construct a distribution Q_x which approximates U_x in variational distance to within an additive factor of $\beta = d_\epsilon(f)/18$.

The algorithm constructs Q_x by picking k independent samples $x_1, \ldots, x_k$ uniformly at random from x, and computing for all $a \in A$: $Q_x(a) = (1/k)|\{i \mid x_i = a\}|$. We choose $k = |A|/\beta^2$.

Define $e_a = |Q_x(a) - U_x(a)|$, the error made on a; we bound $E(e_a)$ as follows: $E(e_a)^2 \leq E(e_a^2) = E(|Q_x(a) - U_x(a)|^2) = \mathrm{Var}(Q_x(a))$. Define for all $i \in [k]$, $Z_i = 1$, if $x_i = a$ and $Z_i = 0$, otherwise. Then, $Q_x(a) = \frac{1}{k}\sum_{i\in[k]} Z_i$. We obtain $E(e_a)^2 \leq \mathrm{Var}(\frac{1}{k}\sum_{i\in[k]} Z_i) = \frac{1}{k^2}\sum_{i\in[k]} \mathrm{Var}(Z_i) = \frac{1}{k^2} \cdot k \cdot U_x(a)(1 - U_x(a)) \leq U_x(a)/k$. Now, if $U_x(a) \leq 1/|A|$, then $E(e_a) \leq \sqrt{(1/|A|)(\beta^2/|A|)} = \beta/|A|$. If $U_x(a) > 1/|A|$, then $E(e_a) \leq \sqrt{U_x(a)(\beta^2/|A|)} \leq \sqrt{U_x(a)^2\beta^2} = \beta U_x(a)$. Thus, for all $a \in A$, $E(e_a) \leq \max\{\beta/|A|, \beta U_x(a)\}$.

Note that $d(Q_x, U_x) = \frac{1}{2}\sum_{a\in A} e_a$. Thus,

$$E(d(Q_x, U_x)) \quad = \quad \frac{1}{2}\sum_{a\in A} E(e_a)$$

$$
\begin{aligned}
&\leq \frac{1}{2}\left(\sum_{a:U_x(a)\leq\frac{1}{|A|}}\frac{\beta}{|A|}+\sum_{a:U_x(a)>\frac{1}{|A|}}\beta U_x(a)\right)\\
&\leq \frac{1}{2}(|A|\cdot(\beta/|A|)+\beta\cdot 1)=\beta.
\end{aligned}
$$

Therefore, by Markov's inequality, $\Pr(d(Q_x, U_x) \geq 3\beta) \leq 1/3$.

The algorithm will generate ℓ independent approximations $Q_x^1, \ldots, Q_x^\ell$ for U_x, where $\ell = O(\ln(1/\delta))$. We call an approximation Q_x^i "good", if it is at distance of less than 3β from U_x. Chernoff bound implies that with probability at least $1-\delta$, more than half of the approximations are good. Note that by the triangle inequality every two good approximations are at distance of less than 6β from each other. The algorithm thus picks an approximation Q_x^j that is of distance less than 6β from more than half of the other approximations, if one exists. If indeed more than half of the approximations are good, then (1) Q_x^j exists (each good approximation can serve as Q_x^j) and (2) any Q_x^j we pick has at least one good approximation at distance of less than 6β; therefore, Q_x^j is at distance of less than 9β from U_x. We conclude that with probability $1-\delta$ our algorithm finds an approximation Q_x that satisfies $d(Q_x, U_x) < 9\beta = d_\epsilon(f)/2$.

The algorithm now looks for the set D of all inputs y for which $d(Q_x, U_y) < d_\epsilon(f)/2$. What we proved above shows that with probability $1-\delta$, $x \in D$. Moreover, by the triangle inequality for every two inputs $y, y' \in D$, $d(U_y, P_{y'}) < d_\epsilon(f)$. It follows from the definition of $d_\epsilon(f)$ that $C_{f,\epsilon}(y) \cap C_{f,\epsilon}(y') \neq \emptyset$. Using now the sunflower property of $C_{f,\epsilon}$, we obtain that there exists a value $b \in \bigcap_{y\in D} C_{f,\epsilon}(y) \subseteq C_{f,\epsilon}(x)$. The algorithm outputs b. □

The analysis of the technique applied in this theorem is tight in terms of its dependence on $|A|$, since there is a lower bound of $\Omega(|A|)$ on the number of samples required to approximate an unknown distribution in L_1 distance [BFR+00]. Therefore, in order to obtain a better upper bound one should apply a different technique.

For small domains this upper bound can be tight up to a constant factor. The tightness depends on the difference between $d_\epsilon(f)$ and $h_\epsilon(f)$. In the Boolean median, for example, $d_\epsilon(\text{median}) = h_\epsilon(\text{median}) = \epsilon$, which means that the bound is tight. In any case, since $d_\epsilon(f) \geq h_\epsilon^2(f)$, the difference between the upper and lower bounds is at most quadratic for small domains.

Finally, we show that block sensitivity is not a good lower bound in some cases.

Proposition 6.3 *There is a function $f : A^n \to B$ such that $R^w_{\epsilon,\delta}(f) = \Omega(\sqrt{n})$, while $\mathrm{bs}_\epsilon(f) = O(1/\epsilon)$.*

Proof. Let f be the Boolean function on bounded-degree d graphs with n vertices such that $f(x) = 1$ iff x is a bipartite graph. x is represented by the graph's edge list (a list of dn integers in $[n]$). We consider the property testing approximation. If x is bipartite, to make it ϵ-far from being bipartite, we need to make at least ϵdn modifications to its edge list, implying $\Omega(\epsilon dn)$-sized blocks, which implies only $O(1/\epsilon)$ block sensitivity. On the other hand, Goldreich and Ron [GR97] prove that $R^w_{\epsilon,\delta}(f) = \Omega(\sqrt{n})$. □

7 Related Work

Most of the computer science literature on the query complexity of sampling algorithms concentrated on the variant of the model in which the functions are *Boolean* (i.e., $A = B = \{0, 1\}$) and the computation is required to be *exact*. This variant is captured by the famous Boolean decision tree model (see [Bd99] for a recent survey about this area).

The query complexity of deterministic, randomized, and non-deterministic computations of Boolean functions is well understood, and is known to be related to various function properties, like sensitivity and block sensitivity, certificate complexity, and degree of representing and approximating polynomials. The situation for the query complexity of approximation of general functions seems to be very different. For example, it is known that the query complexity of (even randomized) computation of any non-constant Boolean symmetric function is $\Omega(n)$ [Pat92, vR97]. On the other hand, the query complexity of approximating some non-Boolean symmetric functions (like the mean and the median) is only constant.

We are aware of only a few lower bounds in the theoretical computer science literature on the query complexity of non-Boolean function approximations; all of these are tailored to specific functions. Canetti *et al.* [CEG95] show a lower bound for additive approximation of the mean; Dagum *et al.* [DKLR95] (and also, implicitly, Schulman and Vazirani [SV99]) give lower bounds for relative approximation of the mean on any input x. Charikar *et al.* [CCMN00] prove a lower bound for ratio approximation of the frequency moment of order 0. Nayak and Wu [NW99] give a lower bound on the *quantum* query complexity of the median and some other statistics.

Sampling algorithms can be viewed as a special case of the general framework studied in statistical decision theory [Ber85]. Classical decision theory takes a point of view, which is somewhat orthogonal to our goals, by seeking ways to get the best approximation from a given set of sample points. In particular, in their setting the algorithm has no control over the samples, their number, or their distribution.

Several sampling lower bounds from sub-areas of statistical decision theory are related to our work. The Cramér-Rao inequality (see, e.g., [Van68], Chapter 2.4) is a classical result in statistical estimation theory, which provides a lower bound on the variance of any unbiased estimator for an unknown parameter (that is, an estimator whose expectation is the true value of the parameter) in terms of the *Fisher information* of the sample distribution. Statistical learning theory [Val84, KV94, Vap98] includes lower bounds on the number of samples needed to learn an unknown function in terms of the VC dimension of the hypothesis space. The optimality of the sequential probability ratio test is a fundamental result in statistical sequential analysis (see, e.g., [Sie85], Chapter 1), which gives a lower bound on the expected number of samples needed to distinguish between two distributions.

All of the above bounds hold for algorithms whose queries are independent and identically distributed. Thus, the bounds do not apply, at least directly, to our setting, where the sampling algorithm may apply different sample distributions for each query, and may have queries that depend on each other. In fact, as we mentioned already before, our work on symmetric functions can be viewed as an equivalence between the computer science setting and the statistical i.i.d. setting. We can then apply known tools from statistics, like the Hellinger distance, to obtain the lower bounds.

8 Discussion and Open Problems

This paper concentrates on a worst-case analysis of the (expected and worst-case) query complexity. Specifically, we defined the query complexity of a decision tree as its query complexity on the "worst" input. However, sometimes the query complexity varies significantly for different inputs and a worst-case analysis gives too pessimistic a view. For example, the query complexity of a relative approximation of the mean is $\Omega(n)$, since in such an approximation the inputs 0^n and $0^{n-1}1$ are ϵ-disjoint, while having Hellinger distance of $O(1/\sqrt{n})$. However, the mean of an input $x \in \{0,1\}^n$ with a constant number of 1's can be approximated with only $O(\epsilon^{-2} \log \delta^{-1})$ queries. This motivates the definition of the query complexity of a function $f : A^n \to B$ on a *specific* input $x \in A^n$, as $R^e_{\epsilon,\delta}(f,x) = \min\{R^e_{\epsilon,\delta}(T,x) \mid T \ (\epsilon,\delta)\text{-approximates } f\}$ (and similarly for $R^w_{\epsilon,\delta}(f,x)$). Our two lower bounds (Theorem 3.2 and Theorem 4.4) can be stated in terms of the query complexity of f on x: $R^e_{\epsilon,\delta}(f,x) \geq (1-2\delta)\mathrm{bs}_\epsilon(f,x)$ and $R^w_{\epsilon,\delta}(f,x) \geq \Omega(\frac{1}{h^2_\epsilon(f,x)} \log \frac{1}{\delta})$, where $h^2_\epsilon(f,x) = \min\{h^2(U_x,U_y) \mid y \text{ is } \epsilon\text{-disjoint from } x\}$. Using this formulation, one can prove, for example, that a relative approximation of the mean on input $x \in \{0,1\}^n$ requires $\Omega(\frac{n}{\epsilon^2|x|} \log \frac{1}{\delta})$ queries, where $|x|$ denotes the Hamming weight of x.

As mentioned above, Boolean query complexity is known to be related to several function properties, one of which is block sensitivity. In this paper we explored only the generalization of block sensitivity to non-Boolean functions. Similar generalizations may be possible also for some of the other properties. However, it is not clear yet whether such generalizations would yield lower or upper bounds for the query complexity of general function approximation.

Our lower bound for the expected query complexity of symmetric functions (Theorem 4.5) has no dependence on the confidence δ. An interesting open problem is to improve this bound to include the missing $O(\log(1/\delta))$ factor.

The lower bound for symmetric function (Theorem 4.4) is based on the fact that an algorithm that approximates the function has to be able to distinguish between any pair of ϵ-disjoint inputs. However, in some cases there are many inputs that are all mutually disjoint from each other, and the algorithm manages to distinguish between all of them simultaneously. It seems that the *number* of mutually disjoint inputs should be a parameter in the lower bound (i.e., the greater this number is the higher the lower bound should be). One case where this comes into play is the estimation of the *empirical distribution* U_x of a given string x: $f : A^n \to [0,1]^{|A|}$, $f(x) = U_x$, $C_{f,\epsilon}(x) = \{D \mid d(D,U_x) < \epsilon\}$. For this function, Batu *et al.* [BFR+00] show a $\Omega(|A|)$ lower bound (with no dependence on ϵ or δ). Our results imply only an $\Omega(\epsilon^{-1}\log(1/\delta))$ lower bound (with no dependence on $|A|$). It seems that our bounded misses the $|A|$ factor, because it ignores the fact that this function has exponentially many mutually disjoint inputs.

Finally, an important open problem is to find tight lower bounds for the query complexity of non-symmetric functions. Currently, we have only the block sensitivity bound, which seems to be too weak.

Acknowledgments

We thank Ron Fagin, Michael Jordan, Christos Papadimitriou, Moni Naor, Andrew Ng, Luca Trevisan, and David Zuckerman for useful discussions.

References

[AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[Bd99] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey, 1999. Available at `http://www.cwi.nl/~rdewolf`.

[Ber85] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

[BFR$^+$00] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–269, 2000.

[CCMN00] M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *Proceedings of the 19th Annual ACM Symposium on Principles of Database Systems (PODS)*, pages 268–279, 2000.

[CEG95] R. Canetti, G. Even, and O. Goldreich. Lower bounds for sampling algorithms for estimating the average. *Information Processing Letters*, 53:17–25, 1995.

[Che52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *American Mathematical Society*, 23:493–507, 1952.

[DKLR95] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. In *Proceedings of the 36th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 142–149, 1995.

[FKSV99] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L^1-difference algorithm for massive data streams. In *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 501–511, 1999.

[FKSV00] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. Testing and spot-checking of data streams. In *Proceedings of the 11th IEEE Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 165–174, 2000.

[GR97] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 406–415, 1997.

[GR00] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 2000. TR00-020.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[HRR99] M. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. In *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, volume 50, pages 107–118, 1999.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory.* The MIT Press, 1994.

[LY90] L. Le Cam and G. L. Yang. *Asymptotics in Statistics - Some Basic Concepts*, pages 24–30. Springer-Verlag, 1990.

[Nis91] N. Nisan. CREW PRAMs and Decision Trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991.

[NW99] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC)*, pages 384–393, 1999.

[NZ96] N. Nisan and D. Zuckerman. Randomness is linear in space. *J. of Computer and System Sciences (JCSS)*, 52(1):43–52, 1996.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 468–474, 1992.

[RT00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. on Discrete Math*, 13(1):2–24, 2000.

[Sie85] D. Siegmund. *Sequential Analysis - Tests and Confidence Intervals.* Springer-Verlag, 1985.

[Sip88] M. Sipser. Expanders, randomness, or time versus space. *J. of Computer and System Sciences (JCSS)*, 36(3):379–383, 1988.

[SV99] L. Schulman and V. V. Vazirani. Majorizing estimators and the approximation of #P-complete problems. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC)*, pages 288–294, 1999.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Van68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory.* Jon Wiley & Sons, Inc., 1968.

[Vap98] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc., 1998.

[vR97] J. von zur Gathen and J. R. Roche. Polynomials with two values. *Combinatorica*, 17(3):345–362, 1997.

[Zuc97] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11(4):345–367, 1997.