SIAM J. DISCRETE MATH.
Vol. 29, No. 3, pp. 1585–1599

RECOGNIZING COVERAGE FUNCTIONS*

DEEPARNAB CHAKRABARTY† AND ZHIYI HUANG‡

Abstract. A *coverage function* f over a ground set $[m]$ is associated with a universe U of weighted elements and m sets $A_1, \ldots, A_m \subseteq U$, and for any $T \subseteq [m]$, $f(T)$ is defined as the total weight of the elements in the union $\cup_{j \in T} A_j$. Coverage functions are an important special case of submodular functions, and arise in many applications, for instance, as a class of utility functions of agents in combinatorial auctions. Naïve representations of coverage functions have size exponential in m, and in algorithmic applications, an access to a value oracle is assumed. In this paper, we ask whether one can recognize if a given oracle is that of a coverage function or not. We demonstrate an algorithm which makes $O(m|U|)$ queries to an oracle of a coverage function and completely reconstructs it. This is polynomial time whenever $|U|$ is polynomially bounded implying the function has a succinct description. To complement the above result, we show a negative result. We prove that "noncoverageness" needs large certificates—there exists a function which is not coverage and yet any algorithm making fewer than 2^{m-1} queries cannot distinguish this function from some coverage function. Our positive result shows that the property of coverageness has $O(m|U|)$-query proximity oblivious testers, while our negative result shows an exponential lower bound. We believe our lower bound also goes through for general property testers, and provide some evidence of the same.

Key words. property testing, coverage functions, learning, Farkas lemma

AMS subject classifications. 68Q99, 68W20

DOI. 10.1137/140964072

1. Introduction. A function $f : 2^{[m]} \mapsto \mathbb{R}_{\geq 0}$ defined over subsets of $[m] \doteq \{1, 2, \ldots, m\}$ is a *coverage* function iff there exists a universe U, nonnegative reals w_i for each element $i \in U$, and subsets $A_1, A_2, \ldots, A_m$ of U such that

$$\forall T \subseteq [m] : f(T) = w\left(\bigcup_{i \in T} A_i\right), \tag{1}$$

where we use the notation $w(S) := \sum_{i \in S} w_i$. The set system $(U; A_1, \ldots, A_m)$ is said to *induce* the coverage function f. The size of the universe U is unspecified; the function f is called *succinct* if $|U|$ is upper bounded by a fixed polynomial of m. Observe that succinct coverage functions have succinct representations.

Coverage functions form an important class of *submodular* functions: a function f is submodular if $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for any pair of subsets A and B. Coverage functions arise in many applications (plant location [7], machine learning [16]), an important one being that in auction theory [5, 17] where utilities of agents are often modeled as coverage functions. Many auction mechanisms take advantage of the specific property of these utility functions; a notable one is the recent work of Dughmi, Roughgarden, and Yan [9] and Dughmi [10] who give $O(1)$-approximate and truthful-in-expectation mechanisms when utilities of agents are coverage. (Such a result is impossible for general submodular functions [8].)

*Received by the editors April 8, 2014; accepted for publication (in revised form) June 4, 2015; published electronically September 1, 2015. A preliminary version [6] of this work titled "Testing Coverage Functions" was presented at the 39th ICALP [*Testing coverage functions* in Automata, Languages, and Programming, Springer, Heidelberg, 2012, pp. 170–181].
http://www.siam.org/journals/sidma/29-3/96407.html
†Microsoft Research, 9 Lavelle Road, Bangalore, 560001, India (dechakr@microsoft.com).
‡The University of Hong Kong, Pokfulam Road, Hong Kong (zhiyi@cs.hku.hk).

In this paper we are interested in the problem of *recognizing* coverage functions given access only to a *value oracle* for the function. More precisely, we are interested in algorithms which can query the value of f at any subset $T \subseteq [m]$, and we want to know how many queries are needed to decide whether or not the function is coverage.

Our first result is a positive one. We give a deterministic *reconstruction* algorithm which makes $O(m|U|)$ queries to a value oracle of a *true* coverage function and reconstructs the coverage function, that is, deduces the underlying set system $(U; A_1, \ldots, A_m)$ and weights of the elements in U. Thus, we give a deterministic, polynomial time, *exact* learning algorithm for succinct coverage functions. Our algorithm can also be interpreted as an algorithm to reconstruct a sparse signal from a (highly) stylized form of linear measurements. We elucidate on this in section 3 where we also describe our algorithm.

How about general coverage functions? It is not clear whether these functions can be succinctly represented and thus the reconstruction problem may be hard. But it is possible that for any "noncoverage" function, there is a succinct certificate which proves the function is not coverage, and if one is able to find this, one could *distinguish* between coverage and noncoverage functions with a polynomial sized proof. For instance, for submodular functions, there is a certificate of nonsubmodularity of size 4—these are the function value at sets $A, B, A \cup B$, and $A \cap B$ which violate submodularity. Our second result is a negative one. We show that there exists a function $\mathbf{f}^*$ which is not coverage, but any certificate of noncoverageness is of size 2^{m-1}.

Connection to property testing. In property testing [14, 15], the objective is to distinguish functions which have a certain property, say coverage, from functions which are "far" from having the property. The usual notion of distance is Hamming distance w.r.t. the uniform distribution, that is, a function f is said to be ε-far from coverage if it needs to be altered in at least $\varepsilon \cdot 2^m$ places to make it coverage. A coverage tester takes as input a parameter ε and accepts if f is coverage and rejects if f is ε-far; it is allowed to err with probability $< 1/3$ in either case. A *proximity oblivious* tester [13] doesn't take ε as input, rather, it rejects a function f with probability at least $\rho(\varepsilon)$, where $\rho : (0, 1] \mapsto (0, 1]$ is a monotonically increasing function with $\rho(0) = 0$.

Our positive result immediately implies an $O(m|U|)$-proximity oblivious tester. We run the reconstruction algorithm. If there is any inconsistency found (that is, some $w_i < 0$), then we reject. If no inconsistencies are found, then we pick a set T at random and compare $f(T)$ and $f'(T)$, where f' is our reconstruction, rejecting if they do not agree. If f is coverage, then we always accept. If f is ε-far, then the final comparison will disagree with probability at least ε since f' is coverage. In contrast, our negative result rules out a proximity oblivious tester making fewer than 2^{m-1} queries for general coverage functions. This is because the proximity oblivious tester must query a certificate when the input is $\mathbf{f}^*$.

We describe a noncoverage function g^* related to $\mathbf{f}^*$ (they arise out of the same construction with different parameters) which requires a $2^{\Omega(m)}$ certificate, and we *believe* that the distance of this function from coverage is close to 1. In fact, g^* has what we call "W-distance" $1 - \exp(-\Theta(m))$, and if a certain upper bound regarding the number of roots of a certain multilinear polynomial holds, then indeed g^* has large Hamming distance from coverage as well. We don't know how to upper bound the roots to the degree we require, but we provide some evidence that such a bound may hold. This is elucidated in section 4.

Related work. The work most relevant to, and indeed which inspired this paper, is that of Seshadhri and Vondrák [21], which addresses the question of testing general submodular set functions. The authors focus on a particular simple testing algorithm, the "square tester," which samples a random set R, $i, j \notin R$ and checks whether or not $f(R, i, j) + f(R) \leq f(R, i) + f(R, j)$. Seshadhri and Vondrák [21] show that $\varepsilon^{-\tilde{O}(\sqrt{m})}$ random samples are sufficient to distinguish submodular functions from those ε-far from submodularity and, furthermore, at least $\varepsilon^{-4.8}$ samples are necessary. Apart from the obvious problem of closing this rather large gap, Seshadhri and Vondrák [21] suggest tackling special, well-motivated cases of submodularity. In fact, the question of testing coverage functions was specifically raised by Seshadhri [20] (attributed to N. Nisan). It is instructive to compare our results with that of Seshadhri and Vondrák [21]. First, although coverage functions are a special case of submodular functions, the subexponential time tester of Seshadhri and Vondrák [21] *does not* imply a tester for coverage functions. This is because a function might be submodular but far from coverage; in fact, our lower bound examples are submodular. Given our result that there are no small certificates of noncoverageness, we believe testing coverageness may be more difficult than testing submodularity. Indeed, if the conjecture referred to in the previous paragraph is true, then this is the case.

Another relevant paper is that of Badanidiyuru et al. [2]. Among other results, Badanidiyuru et al. [2] show that any coverage function f can be arbitrarily well approximated by a succinct coverage function. More precisely, if f is defined via $(U; A_1, \ldots, A_m)$ and weights w, then for any $\varepsilon > 0$, there exists another coverage function f' defined via $(U'; A'_1, \ldots, A'_m)$ and weights w', with $|U'| = \text{poly}(m, 1/\varepsilon)$, such that $f'(T)$ is within $(1 \pm \varepsilon)f(T)$ for all subsets T. This, in some sense, shows that succinct coverage functions capture the essence of all coverage functions. Unfortunately, this "sketch" is found using random sampling on the universe U and it is unknown whether this can be obtained via polynomially many (in m) queries to an oracle for f. If this were possible, then one could possibly couple this sketching algorithm along with our succinct tester to determine if a given function is close to coverage pointwise.

Recently, there has been some work in learning submodular functions and in particular learning coverage functions. Feldman and Kothari [11] describe an algorithm that given values of a coverage function with values in $[0, 1]$ at $O(\log n/\varepsilon^4)$ *uniformly at random* points, constructs a coverage function h such that $||f - h||_1 \leq \varepsilon$. Note that the learning framework differs from our framework in two ways: first, error is ℓ_1 error, and h may differ from f in all the points of the domain, but not by much and, second, the algorithm has access only to random points and not membership queries. Feldman and Vondrák [12] extend this result to general submodular functions, except the dependence on ε is exponential. Another model of learning is due to Balcan and Harvey [3], which they call the PMAC model. Given parameters $\varepsilon, \delta, \alpha$, and samples from a distribution, the goal is to output an hypothesis with probability at least $(1 - \varepsilon)$, which dominates the function and is within α times the function values *pointwise* for $(1 - \delta)$ mass of the points with respect to the same distribution. Balcan and Harvey [3] describe an algorithm with $\alpha = O(\log(1/\varepsilon))$ taking $\text{poly}(n, \log(1/\delta), 1/\varepsilon)$ samples, which works for monotone submodular functions and any product distribution. Feldman and Vondrák [12] describe an algorithm which removes the monotonicity assumption, works for α arbitrarily close to 1, but works only for uniform distribution and the dependence on $\varepsilon, \delta, (\alpha - 1)$ is exponential.

2. Characterizing coverage functions.

Given a set function $f : 2^{[m]} \mapsto \mathbb{R}_{\geq 0}$, we define the W-transform $w : 2^{[m]} \setminus \emptyset \mapsto \mathbb{R}$ as

$$\forall S \in 2^{[m]} \setminus \emptyset, \qquad w(S) = \sum_{T : S \cup T = [m]} (-1)^{|S \cap T| + 1} f(T). \tag{2}$$

We call the resulting sets $\{w(S) : S \subseteq [m]\}$ the W-coefficients of f. The W-coefficients uniquely define a set function; this follows since the $(2^m - 1) \times (2^m - 1)$ matrix M defined as $M(S,T) = (-1)^{|S\cap T|+1}$ if $S \cup T = [m]$ and 0 otherwise, is full rank.[1] Inverting we get the *unique* evaluation of f in terms of its W-coefficients:

$$\forall T \subseteq [m], \quad f(T) = \sum_{S\subseteq[m]:S\cap T\neq\emptyset} w(S). \tag{3}$$

In fact, $w(S)$ is total weight covered by each set in S and not covered by sets outside of S.

THEOREM 2.1. *A set function $f : 2^{[m]} \mapsto \mathbb{R}_{\geq 0}$ is coverage iff all its W-coefficients are nonnegative.*

Proof. Suppose that f is a function with all W-coefficients nonnegative. Consider a universe U consisting of $\{S : S \subseteq [m]\}$ with weight of element S being $w(S)$, the Sth W-coefficient of f. Given U, for $i = 1 \ldots m$, define $A_i := \{S \subseteq [m] : i \in S\}$. For any $T \subseteq [m]$, $\bigcup_{i\in T} A_i = \{S \subseteq [m] : S \cap T \neq \emptyset\}$. From (3) we get $f(T) = w(\bigcup_{i\in T} A_i)$ proving that f is a coverage function.

Suppose f is a coverage function. By definition, there exists $(U; A_1, \ldots, A_m)$ with nonnegative weights on elements in U such that $f(T) = w\left(\bigcup_{i\in T} A_i\right)$. Each $S \in U$ corresponds to a subset of $[m]$ defined as $\{i : S \in A_i\}$. We may further assume each element of U corresponds to a unique subset; if more than one element has the same incidence structure, we may merge them into one element with weight equaling the sum of both the weights. This transformation doesn't change the function value (and thus the W-coefficients) and keeps the weights nonnegative. Furthermore, we may also assume every subset on $[m]$ is an element of U by giving weights equal to 0; this doesn't change the function value either. In particular, $|U|$ may be assumed to be 2^m. As before, one can check that for any $T \subseteq [m]$, $f(T) = \sum_{S:S\cap T\neq\emptyset} w(S)$. From (3) we get that these are the W-coefficients of f, and are hence nonnegative. ∎

From the second part of the proof above, note that the positive W-coefficients of a coverage function f correspond to the elements in the universe U. Let $\{S : w(S) > 0\}$ be the *support* of a coverage function f. Note that succinct coverage functions are precisely those with support size bounded by a polynomial in m.

One can use Theorem 2.1 to certify noncoverageness of a function f: one of its W-coefficients $w(S)$ must be negative, and the function values in the summand of (2) certify it. Observe, however, that this certificate can be exponentially large. In section 4 we will show this is inherent in *any* certificate of coverageness. The W-transformation also motivates the following notion of distance to coverage functions.

DEFINITION 1. *The W-distance of a function f (from coverage functions) is the fraction of its negative W-coefficients.*

Comparison with Fourier transformation. Readers who are familiar with harmonic analysis of Boolean functions might find (2) similar to the Fourier transformation. Indeed, if we sum over all T in the summation of (2) instead of only over the T s.t. $S \cup T = [m]$, then it becomes the Fourier transformation. However, it is worth pointing out that due to this subtle change, the W-transformation behaves quite differently to the representation by Fourier basis. In particular, unlike the Fourier basis, the basis of the W-transform is not orthonormal with respect to the usual notion of inner product.

Characterization via multilinear extension. We can use the above characterization in Theorem 2.1 to obtain another characterization via derivatives of the multilinear

[1] One can check $M^{-1}(S,T) = 1$ if $S \cap T \neq \emptyset$ and 0 otherwise.

extension. Such a result was known [1], but to our knowledge hasn't appeared in print. The proof of this straightforward theorem is provided in Appendix A. Given a set function $f : 2^{[n]} \mapsto \mathbb{R}$, one defines the multilinear $F : [0,1]^n$ as follows:

$$F(x) := \sum_{S \subseteq [n]} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i).$$

Given a subset $T \subseteq [m]$ with $|T| = k$, we use the shorthand $\frac{\partial^k f}{\partial T}$ to denote $\frac{\partial^k f}{\partial x_{i_1} \cdots \partial x_{i_k}}$, where $T = \{i_1, \ldots, i_k\}$. We call a partial derivative $\frac{\partial^k f}{\partial T}$ *odd* if $|T|$ is odd, and even otherwise.

THEOREM 2.2. *A function f is coverage iff the odd derivatives of F are nonnegative, and the evens are nonpositive for all $x \in [0,1]^n$.*

3. Reconstructing succinct coverage functions. Given a coverage function f, suppose $\{S_1, \ldots, S_n\}$ is the support of f (w.r.t. W-coefficients). (Note that $|U| \geq n$.) That is, these are the sets in the W-transform of f with $w(S_i) > 0$, and all the other sets have weight 0. We now give an algorithm to find these sets and weights using $O(mn)$ queries. As a corollary, we will obtain a polynomial time algorithm for testing succinct coverage functions, where $n = \text{poly}(m)$.

The procedure is iterative and is similar to the Goldreich–Levin algorithm to compute "large" Fourier coefficients. The algorithm maintains a partition of $2^{[m]}$ at all times, and for each part in the partition, stores the total weight of all the sets contained in the part. We start with the trivial partition containing all sets whose weight is given by $f([m])$. In each iteration, these partitions are refined; for instance, in the first iteration we divide the partition into sets containing a given element i and those that don't contain the element i. The total weights of the first collection can be found by querying $f(\{i\})$. Any time the sum of a part evaluates to 0, we discard it and subdivide it no more. After m iterations, the remaining n parts give the support sets and their weights. To describe formally, we introduce some notation.

Given a vector $\mathbf{x} \in \{0,1\}^k$ we let $\text{supp}(\mathbf{x})$ be the subset of $[k]$ containing the elements i with $\mathbf{x}(i) = 1$. Let $\mathcal{F}(\mathbf{x}) := \{S \subseteq [m] : S \cap [k] = \text{supp}(\mathbf{x})\}$, that is, subsets of $[m]$ which "match" with the vector $\mathbf{x}$ on the elements in $[k]$. Note that $|\mathcal{F}(\mathbf{x})| = 2^{m-k}$, and $\{\mathcal{F}(\mathbf{x}) : \mathbf{x} \in \{0,1\}^k\}$ is a partition of $2^{[m]}$; if $k = 0$, then $\mathcal{F}(\mathbf{x})$ is the trivial partition consisting of all subsets of $[m]$. Given $\mathbf{x} \in \{0,1\}^k$, we let $\mathbf{x} \oplus 0$ be the $(k+1)$ dimensional vector with $\mathbf{x}$ appended with a 0. Similarly, define $\mathbf{x} \oplus 1$. At the kth iteration, the algorithm maintains the partition $\{\mathcal{F}(\mathbf{x}) : \mathbf{x} \in \{0,1\}^k\}$ and the total weight of subsets in each $\mathcal{F}(\mathbf{x})$. Subsequent iteration refines each partition $\mathcal{F}(\mathbf{x})$ into $\mathcal{F}(\mathbf{x} \oplus 0)$ and $\mathcal{F}(\mathbf{x} \oplus 1)$, evaluating the total weight in each part. Observe, if the total weight of $\mathcal{F}(\mathbf{x})$ equals 0, then since the function is coverage, so must be the total weight of both $\mathcal{F}(\mathbf{x} \oplus 0)$ and $\mathcal{F}(\mathbf{x} \oplus 1)$, and thus the algorithm need not explicitly refine such an $\mathbf{x}$. When the total weight of $\mathcal{F}(\mathbf{x})$ is not 0, the algorithm makes $O(1)$ queries to the oracle to refine. In the end, only n sets have positive weight, and so the algorithm terminates in m iterations making $O(mn)$ queries.

We now describe the refinement procedure. In what follows, we say a vector $\mathbf{y} \leq \mathbf{x}$ if they are of the same dimension and $\mathbf{y}(i) \leq \mathbf{x}(i)$ for all i. We say $\mathbf{y} < \mathbf{x}$ if $\mathbf{y} \leq \mathbf{x}$ and $\mathbf{y} \neq \mathbf{x}$. The procedure goes over the vectors $\mathbf{x}$ in *increasing* number of ones, and for each calculates the weight of $\mathcal{F}(\mathbf{x} \oplus 1)$. This immediately gives the weight of $\mathcal{F}(\mathbf{x} \oplus 0)$ since $w(\mathcal{F}(\mathbf{x} \oplus 0)) = w(\mathcal{F}(\mathbf{x})) - w(\mathcal{F}(\mathbf{x} \oplus 1))$. It is essential we perform this operation in this order, as weights calculated earlier are required later (see step 6 below).

CLAIM 3.1. *The procedure* Refine *returns the correct weights of the refinement.*

Procedure Refine.

Input: $0 \le k \le m$, and weights from the previous iteration $\{w(\mathcal{F}(\mathbf{x})) > 0 : \mathbf{x} \in \{0,1\}^k\}$.
Output: $\{w(\mathcal{F}(\mathbf{x} \oplus 0)), w(\mathcal{F}(\mathbf{x} \oplus 1)) : \mathbf{x} \in \{0,1\}^k\}$.
Order $\{\mathbf{x} : w(\mathcal{F}(\mathbf{x})) > 0\}$ by increasing number of 1's breaking ties arbitrarily.
Call the order $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$;
for $i = 1 \to N$ **do**
  Query $f([k] \setminus \mathrm{supp}(\mathbf{x}_i)) =: F_i^0$ and $f(([k] \setminus \mathrm{supp}(\mathbf{x}_i)) \cup \{k+1\}) =: F_i^1$.
  Define $\Delta_{\mathbf{x}_i} := F_i^1 - F_i^0 - \sum_{\mathbf{x}_j < \mathbf{x}_i} \Delta_{\mathbf{x}_j}$.
  $w(\mathcal{F}(\mathbf{x}_i \oplus 1)) = \Delta_{\mathbf{x}_i}$; $w(\mathcal{F}(\mathbf{x} \oplus 0)) = w(\mathcal{F}(\mathbf{x}_i)) - \Delta_{\mathbf{x}_i}$.
end for

Proof. Recall that $\Delta_{\mathbf{x}_i} := F_i^1 - F_i^0 - \sum_{\mathbf{x}_j < \mathbf{x}_i} \Delta_{\mathbf{x}_j}$ as defined in procedure Refine. It suffices to show that $\Delta_{\mathbf{x}_i} = w(\mathcal{F}(\mathbf{x}_i \oplus 1)) = \sum_{S : S \cap [k] = \mathbf{x}_i, k+1 \in S} w(S)$. The right-hand side (RHS) equals

$$\sum_{S : S \cap [k] \subseteq \mathbf{x}_i,\ k+1 \in S} w(S) \ - \ \sum_{\mathbf{y} < \mathbf{x}_i} \ \sum_{S \cap [k] = \mathbf{y}, k+1 \in S} w(S). \tag{4}$$

The first term above equates to

$$\sum_{\substack{S : S \cap [k] \setminus \mathbf{x}_i = \emptyset, \\ k+1 \in S}} w(S) = \sum_{S : S \cap ([k] \setminus \mathbf{x}_i \cup k+1) \neq \emptyset} w(S) \ - \sum_{S : S \cap ([k] \setminus \mathbf{x}_i) \neq \emptyset} w(S) \ = \ F_i^1 - F_i^0.$$

The second term in (4) is precisely $\sum_{\mathbf{y} < \mathbf{x}_i} w(\mathcal{F}(\mathbf{y} \oplus 1))$. If $i = 1$, then this is 0; for other i this equates to $\sum_{\mathbf{x}_j < \mathbf{x}_i} \Delta_{\mathbf{x}_j}$ by induction. ∎

Procedure Recover Coverage.

Input: Value oracle to coverage function f.
Output: $\{S_1, \ldots, S_n\}$ with $w(S_i) > 0$.
Initialize $k = 0$, $\mathbf{x}$ to be the empty vector, and list L to contain $\mathbf{x}$.
Let $w(\mathcal{F}(\mathbf{x})) = f([m])$.
for $k = 1 \to m$ **do**
  Run Refine on each $\mathbf{x}$ in list L and remove it.
  Add $\mathbf{x} \oplus 0$ and $\mathbf{x} \oplus 1$ to L only if the weights evaluate to positive.
end for
For each $\mathbf{x} \in \{0,1\}^m$ in L, return corresponding set and weight calculated by the Refine procedure.

Theorem 3.2. *Given value oracle access to a coverage function f with positive weight sets $\{S_1, \ldots, S_n\}$, the procedure* Recover Coverage *returns the correct weights with $O(mn)$ queries to the oracle.*

Proof. Let L_k be the list L in the algorithm at iteration k. The number of queries made by Refine when input a list L is $O(|L|)$. Therefore, the total number of queries made by the algorithm is $O\left(\sum_{k=1}^m |L_i|\right) = O(mn)$ since $|L_i| \le |L_m| = n$. ∎

As mentioned in the introduction, the above reconstruction algorithm implies a tester. More precisely, given any n, there exists an $O(mn + \epsilon^{-1})$ time tester which will return Yes for coverage functions having W-support size at most n, and return No with $\Omega(1)$ probability for functions that are ϵ-far from the set of coverage functions with W-support at most n. Run the reconstruction algorithm described above. If we get a set with negative weight, return No. If we succeed, then if f is truly a coverage function, we have derived the unique weights. We sample $O(\varepsilon^{-1})$ random sets and compare the value of our computed function with that of the oracle; if the function is ε-far from coverage, then we will catch it with probability $O(1)$.

We end this section with a lower bound.

THEOREM 3.3. *Reconstructing coverage functions on m elements with W-support size n requires at least $\Omega(mn/\log n)$ probes.*

Proof. Consider the bipartite graphs. On one side (A side) let there be m vertices that correspond to the sets. On the other side (U side) let there be n vertices that correspond to the elements in U. Let the weight be 1 on all vertices in U. Each nonisomorphic bipartite graph (on permutation of the U vertices) maps to a different coverage function over the A side: the neighborhood of a vertex $A_i \in A$ is precisely the elements it contains. Note each such a graph corresponds to a way of allocating n identical balls (U-side vertices) into 2^m different bins (different choice of set of adjacent A-side vertices), where each bin can contain multiple balls. This number is at least $\binom{2^m+n-1}{n-1} \geq (\frac{2^m}{n})^{n-1}$. Hence, we need at least $\Omega(mn)$ bits of information. Notice that each probe of function value only provides $O(\log n)$ bits of information; since the function value is always an integer between 0 and n, we get the lower bound in Theorem 3.3. $\square$

4. Noncoverageness needs exponential sized certificates. In this section we demonstrate noncoverage functions which need exponentially many queries to distinguish them from coverage functions. In fact, we describe a family of functions. For any integer $k \in [m]$, consider the function $\mathbf{f}_k$ which has W-coefficients $\mathbf{w}_k(S) = -1$ if $|S| > k$, and $\mathbf{w}_k(S) = N$ if $|S| \leq k$, where N is a large positive integer which will be precisely determined later. Clearly $\mathbf{f}_k$ is not a coverage function for $k < m$. Henceforth, we will use boldface when we talk about this specific noncoverage function and light face for a generic function.

First, observe that from (2) it follows that $w(S)$ for any function f can be precisely determined by querying the $2^{|S|}$ sets in $\{T : T \cup S = [m]\} = \{\overline{S} \cup X : X \subseteq S\}$. It follows that $\mathbf{f}_k$ can be distinguished from coverage using 2^{k+1} queries. We show an almost tight lower bound: any tester which makes less than 2^k queries cannot distinguish $\mathbf{f}_k$ from a coverage function, and thus any certificate of noncoverageness needs to access the 2^k values. More precisely, we show that given the value of $\mathbf{f}_k$ on a collection of sets $\mathcal{J}$ with $|\mathcal{J}| < 2^k$, there exists a *coverage* function f which has the same values on the sets in $\mathcal{J}$. This bound is information theoretic and holds even if the tester has infinite computational power.

THEOREM 4.1. *There exists a coverage function consistent with the queries of $\mathbf{f}_k$ on $\mathcal{J}$ if $|\mathcal{J}| < 2^k$.*

COROLLARY 4.2. *Any certificate of noncoverageness of $\mathbf{f}_k$ must be of size at least 2^k.*

Setting $k(m) = m/4$, we get $\mathbf{f}^* = \mathbf{f}_{k(m)}$ has W-distance at least $(1 - e^{-\Theta(m)})$, giving us the following corollary.

COROLLARY 4.3. *Any tester distinguishing between coverage functions and functions of W-distance as large as $(1 - e^{-\Theta(m)})$ needs at least $2^{\Theta(m)}$ queries.*

We give a sketch of the proof before diving into the details. Suppose a tester queries the collection $\mathcal{J}$. We first observe that the existence of a coverage function consistent with the queries in $\mathcal{J}$ can be expressed as a set of linear inequalities.

Using Farkas' lemma, we get a certificate of the *nonexistence* of such a completion. This certificate, at a high level, corresponds to an assignment of values on the m-dimensional hypercube satisfying certain linear constraints. We show that if the parameter N is properly chosen, most of these assignments can be assumed to be 0. In the next step we use this property to show that unless the size of $|\mathcal{J}| \geq 2^k$, all the assignments need to be 0 which contradicts the Farkas linear constraints, thereby proving the existence of the coverage function consistent with $\mathbf{f}_k$ values given on $\mathcal{J}$.

4.1. Consistent coverage functions and Farkas' lemma. We start by choosing $N \geq (2^m)!$. First, note that this implies $\mathbf{f}_k(T) \geq 0$ for all $T \subseteq [m]$. This follows from (3); in fact, this only requires $N \geq 2^m$. Now recall, from Theorem 2.1, a function f defined on subsets of $[m]$ is coverage iff it satisfies

$$\forall S \subseteq [m] : \qquad \sum_{T: S\cup T=[m]} (-1)^{|S\cap T|+1} f(T) \geq 0,$$

$$\forall T \subseteq [m] : \qquad f(T) \geq 0.$$

Let $\mathcal{J}$ be the collection of sets on which the function $\mathbf{f}_k$ has been queried. Define

$$\mathbf{b}_k(S) := \sum_{T\in\mathcal{J}: S\cup T=[m]} (-1)^{|S\cap T|} \mathbf{f}_k(T).$$

Therefore, if we can find assignments $f : 2^{[m]} \setminus \mathcal{J} \mapsto \mathbb{R}_{\geq 0}$ satisfying

$$\forall S \subseteq [m] : \qquad \sum_{T\notin\mathcal{J}: S\cup T=[m]} (-1)^{|S\cap T|+1} f(T) \geq \mathbf{b}_k(S), \tag{5}$$

$$\forall T \notin \mathcal{J} : \qquad f(T) \geq 0, \tag{6}$$

we have exhibited a coverage function consistent with the values of $\mathbf{f}_k$ on $\mathcal{J}$. Applying Farkas' lemma (see, for instance, [4]), we see that there is *no* feasible solution to (5), (6) iff there is a feasible solution $\alpha : 2^{[m]} \mapsto \mathbb{R}_{\geq 0}$ satisfying

$$\sum_{S\subseteq[m]} \alpha(S)\mathbf{b}_k(S) > 0, \tag{7}$$

$$\forall T \notin \mathcal{J} : \qquad \sum_{S: S\cup T=[m]} (-1)^{|S\cap T|+1} \alpha(S) \leq 0, \tag{8}$$

$$\forall S \subseteq [m] : \qquad \alpha(S) \geq 0. \tag{9}$$

Henceforth, we assume there exists a feasible solution to the system above. We make the following observation.

CLAIM 4.4. *If there exists a solution α satisfying* (7), (8), *and* (9), *then* $\sum_{S\subseteq[m]} \alpha(S)\mathbf{w}_k(S) < 0$, *where $\mathbf{w}_k$ are the W-coefficients w.r.t. $\mathbf{f}_k$.*

Proof. By definition, $\forall S \subseteq [m]$:

$$\mathbf{w}_k(S) = \sum_{T: S\cup T=[m]} (-1)^{|S\cap T|+1} \mathbf{f}_k(T) = \sum_{T\notin\mathcal{J}: S\cup T=[m]} (-1)^{|S\cap T|+1} \mathbf{f}_k(T) - \mathbf{b}_k(S).$$

Therefore, (7) along with the above equality implies

$$\sum_{T\notin\mathcal{J}} \sum_{S: S\cup T=[m]} \alpha(S)(-1)^{|S\cap T|+1} \mathbf{f}_k(T) - \sum_{S\subseteq[m]} \alpha(S)\mathbf{w}_k(S) = \sum_{S\subseteq[m]} \alpha(S)\mathbf{b}_k(S) > 0.$$

But by (8), $\sum_{S\subseteq[m]}\alpha(S)(-1)^{|S\cap T|+1}\le 0$ for all $T\notin\mathcal{J}$, and $\mathbf{f}_k(T)\ge 0$ for all $T\subseteq[m]$. So we have that $\sum_{S\subseteq[m]}\alpha(S)\mathbf{w}_k(S)<0$. ∎

In the next lemma we show that one can assume there is a feasible solution to (7), (8), and (9) with many of the $\alpha(S)$'s set to 0.

LEMMA 4.5. *If there exists α satisfying* (7)*,* (8)*, and* (9)*, then we may assume $\alpha_S=0$ for all S such that $|S|\le k$.*

Intuitively, what this lemma says is that the constraint (5) for sets of size $\le k$ should not help in catching the function not being coverage. This is because the true function values satisfies the constraints with huge "redundancy":

$$\sum_{T:S\cup T=[m]}(-1)^{|S\cap T|+1}\mathbf{f}_k(T)=N\gg 0.$$

Proof. Suppose there is an α satisfying (7), (8), and (9). Then, by scaling we may assume that

$$\sum_{S\subseteq[m]}\alpha(S)=1. \tag{10}$$

Choose α to be a basic feasible solution satisfying (8), (9), (10). Such a solution makes 2^m of the inequalities tight, and by Cramer's rule, all nonzero $\alpha(S)\ge\frac{1}{(2^m)!}$ since all coefficients are $\{\pm1,0\}$.

From Claim 4.4, we get $\sum_{S\subseteq[m]}\alpha(S)\mathbf{w}_k(S)<0$. Assume for contradiction that there exists S_0, $|S_0|\le k$, such that $\alpha_{S_0}>0$. Since $\alpha_{S_0}\ge\frac{1}{(2^m)!}$, we have $\sum_{S\subseteq[m]}\alpha(S)\mathbf{w}_k(S)\ge\frac{1}{(2^m)!}N-\sum_{S\subseteq[m]:|S|>k}\alpha(S)>1-1=0$, a contradiction. The latter inequality follows from (10) and our assumption that $N>(2^m)!$. ∎

4.2. Nullity of Farkas certificate. In the following discussion, due to Lemma 4.5 we assume $\alpha(S)=0$ for all S, $|S|\le k$. Consider the following linear function of the α's. For a set T, define

$$g(T):=\sum_{S:S\cup T=[m]}(-1)^{|S\cap T|+1}\alpha(S).$$

From (8), we get $g(T)\le 0$ for all $T\notin\mathcal{J}$. The following lemma shows that given Lemma 4.5, and the fact that some α must be strictly positive (otherwise it violates (7)), one can lower bound the size of $\mathcal{J}$.

LEMMA 4.6. *If $\alpha(S)=0$ $\forall|S|\le k$, and $\alpha(S)>0$ for some S, then $g(T)>0$ for $\ge 2^k$ subsets $T\subseteq[m]$.*

Proof. It will be useful to note that α's can be written in terms of g's as follows:

$$\alpha(S)=\sum_{T:T\cap S\neq\emptyset}g(T)\;=\;G-\sum_{T\subseteq\overline{S}}g(T),\qquad\text{where } G:=\sum_{T\subseteq[m]}g(T). \tag{11}$$

Let S be a minimal set with $\alpha(S)>0$. Note that $|S|\ge k+1$. Consider any $i\in S$. By minimality, we have $\alpha(S\setminus i)=0$, giving us

$$0=G-\sum_{T\subseteq\overline{S\setminus i}}g(T)=G-\sum_{T\subseteq\overline{S}}g(T)-\sum_{T\subseteq\overline{S}}g(T\cup i). \tag{12}$$

Observe that the first two terms evaluate to $\alpha(S)$; this is established in (11). Therefore, we have established that for all $i\in S$, $\sum_{T\subseteq\overline{S}}g(T\cup i)=\alpha(S)>0$. By induction, we can extend the above calculation to any nonempty subset $X \subseteq S$. We remind the reader here that this is for a minimal positive set S and not any set S.

(13) For any minimal set S with $\alpha(S) > 0$ and any *proper, nonempty*

$$X \subseteq S, \sum_{T \subseteq \overline{S}} g(T \cup X) = (-1)^{|X|+1}\alpha(S).$$

Note that the summands in (13) are disjoint for different sets X and, furthermore, whenever $|X|$ is odd, the sum is > 0 implying at least one of the summands must be positive for each odd subset $X \subseteq S$. This proves the lemma since $|S| \geq k+1$.

Proof of (13). Let's denote the sum $\sum_{T \subseteq \overline{S}} g(T \cup X)$ as $h(X)$. We need to prove for every proper, nonempty subset $X \subseteq S$, $h(X) = (-1)^{|X|+1}\alpha(S)$. Let us establish this first for singletons. For $X = \{i\}$, we need to show $h(\{i\}) = \alpha(S)$; this is established by (12) and (11). We now prove by induction on $|X|$ with the previous statement establishing the base case. By induction, $h(Y) = (-1)^{|Y|+1}\alpha(S)$ for every nonempty proper subset $Y \subset X$. By minimality of S we get $\alpha(S \setminus X) = 0$. This gives us (from (11))

$$0 = \alpha(S \setminus X) = G - \sum_{T \subseteq \overline{S \setminus X}} g(T) = G - \sum_{Y \subseteq X} h(Y).$$

Rearranging, $h(X) = G - \sum_{Y \subsetneq X} h(Y) = \alpha(S) - \sum_{i=1}^{|X|-1} \binom{|X|}{i}(-1)^{i+1}\alpha(S) = (-1)^{|X|+1}\alpha(S)$. ∎

Proof of Theorem 4.1. Suppose there is no consistent completion, implying α's satisfying (7), (8), and (9). By Lemmas 4.5 and 4.6, we get that if (8) holds, then $|\mathcal{J}| \geq 2^k$. ∎

4.3. Lower bounds for property testing. As noted in the introduction, Theorem 4.1 implies lower bounds for proximity oblivious testers—for any $k < m$, a proximity oblivious tester must reject $\mathbf{f}_k$ with some nonzero probability, and if it does reject $\mathbf{f}_k$ making $< 2^k$ queries, then it must reject the consistent coverage function, which isn't allowed by the definition.

However, $\mathbf{f}_k$ could be very close to a coverage function in the usual notion of Hamming distance w.r.t. the uniform distribution. From Corollary 4.3, we know that for $k = k(m)$, $\mathbf{f}^*$ has a large W-distance from coverage. It is natural to ask how do these two distances relate. We first note that the two notions are unrelated; in particular, we show two functions each "far" in one notion, but "near" in the other. The proofs of the following two lemmas are provided in Appendix B.

LEMMA 4.7. *There is a function with W-distance $1 - e^{-\Theta(m)}$ whose distance to coverage is $e^{-\Theta(m)}$.*

LEMMA 4.8. *There is a function with W-distance $O(m^2/2^m)$ whose distance to coverage is $\Omega(1)$.*

Despite the fact that the two notions are not comparable, we *believe* that $\mathbf{f}^*$ may be far in the usual notion of distance as well. In fact, if the following conjecture regarding roots of certain multilinear polynomials is true, then the above is indeed the case. Unfortunately, we are unable to prove/disprove this conjecture and leave it as an open question.

CONJECTURE 4.9. *For any m-variate multilinear polynomials*

$$p(\boldsymbol{x}) = \sum_{S \subseteq [m]} \lambda_S \prod_{i \in S} x_S$$

with $\lambda_S < 0$ for all $|S| \geq k$, has at most $O(k2^m/\sqrt{m})$ zeroes on the hypercube $\{0,1\}^m$.

In fact, we conjecture that the maximum number of zeros is achieved when the $k+1$ layers of function values in the "middle of the hypercube" are zero, that is, $p(\boldsymbol{x}) = 0$ iff $(m-k)/2 \leq \|\boldsymbol{x}\|_1 \leq (m+k)/2$. At the end of this section, we present some evidence for this conjecture by proving it for symmetric polynomials, that is, when $p(x_1, \ldots, x_m) = p(x_{\sigma(1)}, \ldots, x_{\sigma(m)})$ for any permutation σ of $[m]$. We now show that the conjecture implies $\mathbf{f}^*$ is far from coverage in the usual notion of distance.

LEMMA 4.10. *Assuming Conjecture* 4.9, *with* $k(m) = o(\sqrt{m})$, $\mathbf{f}^*$ *is* $1 - o(1)$ *far from coverage.*

Remark 1. Theorem 3.3 implies there is such an $\mathbf{f}^*$ requiring superpolynomial queries to test as long as we have $k(m) = \omega(\log m)$.

Proof. Consider the coverage function f that is closest to $\mathbf{f}^*$ in the usual notion of distance. Let $w, \mathbf{w}^*$ be the W-coefficients of f, f^*. Define the function $f' := f - \mathbf{f}^*$ and let $w' := w - \mathbf{w}^*$. By linearity of W-transformation, we get that w' are the W-coefficients for f'. Therefore,

$$f'(T) = \sum_{S: T \cap S \neq \emptyset} w'(S) = \sum_{S \subseteq [m]} w'(S)(1 - \mathbf{1}_{T \cap S = \emptyset}).$$

Consider the following binary vector representation of $T \subseteq [m]$: $\mathbf{x} \in \{0,1\}^m$ such that $\mathbf{x}_i = 0$ iff $i \in T$. Using this, the function f' can be interpreted as $f'(\mathbf{x}) = W' - \sum_{S \subseteq [m]} w'(S) \prod_{i \in S} x_i$, where $W' = \sum_{S \subseteq [m]} w'(S)$ is a constant. We are using here the fact that $T \cap S = \emptyset$ is equivalent to $S \subseteq \overline{T}$. By our choice of $\mathbf{w}^*$ and the assumption that $w(S) \geq 0$ for all S, we have $w'(S) \geq 1$ for all $|S| > k$. From Conjecture 4.9, we get that at most $O(k/\sqrt{m})$-fraction of the function values of f' are zeroes implying f is at least $1 - O(k/\sqrt{m})$ far from $\mathbf{f}^*$. The lemma follows since $k = o(\sqrt{m})$. ∎

Proof of Conjecture 4.9 *for symmetric functions.* Since p is symmetric, each λ_S is equal for sets of the same cardinality. Let λ_j denote the value of λ_S when $|S| = j$. Then p is equivalent to the function $g : \{1, \ldots, m\} \mapsto \mathbb{R}$,

$$g^{(0)}(i) := g(i) = p(\boldsymbol{x} : \|\boldsymbol{x}\|_1 = i) = \sum_{j=0}^{m} \sum_{S:|S|=j} \lambda_j \prod_{i \in S} x_i = \sum_{j=0}^{m} \lambda_j \binom{i}{j}. \tag{14}$$

We now define functions $g^{(t)}(i) := g^{(t-1)}(i+1) - g^{(t-1)}(i)$ for $t > 0$. Note that $g^{(t)}$ is a function from $\{1, \ldots, m-t\}$ to the reals. We will let $\lambda_j^{(t)}$ denote the coefficient of $\binom{i}{j}$ in the decomposition of $g^{(t)}$ as in (14). By our assumption, $\lambda_j^{(0)} < 0$ for all $j \geq k$.

Note that $\lambda_j^{(t)} = \lambda_{j+1}^{(t-1)}$. This is because

$$\begin{aligned} g^{(t)}(i) = g^{(t-1)}(i+1) - g^{(t-1)}(i) &= \sum_{j=0}^{m} \lambda_j^{(t-1)} \binom{i+1}{j} - \sum_{j=0}^{m} \lambda_j^{(t-1)} \binom{i}{j} \\ &= \sum_{j=0}^{m} \lambda_j^{(t-1)} \binom{i}{j-1}. \end{aligned}$$

Therefore, we get $\lambda_j^{(t)} < 0$ for all $j > k - t$. In particular, $g^{(k)}$ is negative in

$$\{1, \ldots, m-k\}.$$

Given a univariate function f, let's say that f *crosses zero* at i if $f(i) \leq 0$ and $f(i+1) \geq 0$, or, $f(i) \geq 0$ and $f(i+1) \leq 0$. If f crosses zero at most ℓ times, it's clear it can have at most ℓ zeroes. Thus it suffices to show that $g^{(0)}$ crosses zero at most k times. Note that $g^{(k)}$ never crosses zero. The proof follows from the claim that if $g^{(t)}$ crosses zero r times, then $g^{(t-1)}$ can cross zero at most $r+1$ times. This is because at the places at which $g^{(t-1)}$ crosses zero, $g^{(t)}$ changes sign (or remains 0) and thus crosses zero as well. This is true for all except the last place since $g^{(t)}$ is defined on a domain which is smaller by 1. This ends the proof.

Appendix A. Proof of Theorem 2.2. A calculation gives $\frac{\partial^k F(x)}{\partial T} = \sum_{S\subset[n]:S\cap T=\emptyset} h_S(T)\prod_{i\in S} x_i \prod_{i\notin S\cup T}(1-x_i)$, where

$$h_S(T) := (-1)^{|T|}\sum_{Z\subseteq T}(-1)^{|Z|} f(S\cup Z).$$

To prove the theorem we need to show that f is coverage iff for all T and all sets $S\cap T=\emptyset$, $h_S(T)\geq 0$ if $|T|$ is odd and $h_S(T)<0$ if $|T|$ is even. We claim the following lemma.

Lemma A.1. *For any set S such that*

$$S\cap T=\emptyset, h_S(T) = (-1)^{|T|+1}\sum_{T\subseteq U\subseteq[m]\setminus S} w(U).$$

If f is coverage, we get that $sign(h_S(T)) = sign((-1)^{|T|+1})$, which is what we want. If f is not coverage, and thus $w(U)<0$ for some nonempty subset U, then $h_{[m]\setminus U}(U) = (-1)^{|U|+1}w(U)$ which has different parity as $|U|$. Thus the above lemma completes the proof of the theorem.

Proof. We can use (3) to get

$$\begin{aligned} h_S(T) &= (-1)^{|T|}\sum_{Z\subseteq T}(-1)^{|Z|}\sum_{U:U\cap(S\cup Z)\neq\emptyset} w(U) \\ &= (-1)^{|T|}\sum_{U\subseteq[m]} w(U)\sum_{Z\subseteq T:U\cap(S\cup Z)\neq\emptyset}(-1)^{|Z|}. \end{aligned}$$

Note that if $U\cap S\neq\emptyset$, the second summation boils to $\sum_{Z\subseteq T}(-1)^{|Z|}=0$. Otherwise, it is

$$\sum_{Z\subseteq T:U\cap Z\neq\emptyset}(-1)^{|Z|} = \sum_{Y_1\subseteq T\setminus U,\quad \emptyset\neq Y_2\subseteq T\cap U}(-1)^{|Y_1|+|Y_2|}.$$

Now, unless $T\setminus U=\emptyset$, we have $\sum_{Y_1\subseteq T\setminus U}(-1)^{|Y_1|}=0$. Thus, we get

$$\sum_{Z\subseteq T:U\cap Z\neq\emptyset}(-1)^{|Z|} = \begin{cases}\sum_{\emptyset\neq Z\subseteq T}(-1)^{|Z|}=-1 & \text{if } T\subseteq U,\\ 0 & \text{otherwise.}\end{cases}$$

Putting it all together we get

$$h_S(T) = -(-1)^{|T|}\sum_{U:U\cap S=\emptyset, T\subseteq U} w(U). \qquad \square$$

Appendix B. Proof of Lemmas 4.7 and 4.8.

Proof of Lemma 4.7. Let us consider a function f which is similar to the lower bound example in section 4. More concisely, f's W-representation satisfies that

$w(S) = -1$ if $m > |S| > k$, and $w(S) = N$ if $|S| \leq k$ or $S = [m]$. Here we will let $k = m/4$ and $N > 0$ is a sufficiently large number; $N = 5m!$ suffices. For simplicity, assume that m is a multiple of 8.

First, it follows immediately from the definition that the fraction of weights that are negative is at least $(1 - e^{-\Theta(k)}) = (1 - e^{-\Theta(m)})$. Next, let us prove that there exists a coverage function f' such that the function values of f and f' differ in at most $e^{-\Theta(m)}$ fraction of the entries. Let w' denote the W-representation of f'. Let $\Delta f = f' - f$ and $\Delta w = w' - w$. Note that Δw is the W-representation of Δf. In the remainder, we will find a function Δf over the subsets of $[m]$ satisfying the following properties: (a) Δf is nonzero on at most a $e^{-\Theta(m)}$ fraction of the subsets, and (b) the W-representation of Δf, that is Δw, has the property that $\Delta w(S) \geq 1$ if $m > |S| > m/4$ and $\Delta w(S) \geq -N$ if $|S| \leq k$ or $S = [m]$.

In particular, we will consider Δf that is symmetric, that is, for any T and T' with $|T| = |T'| = i$, we have $\Delta f(T) = \Delta f(T') = \widehat{f}(i)$ for some $\widehat{f} : [m] \mapsto \mathbb{R}$. Note that for symmetric functions, the W-representation is also symmetric, that is, given by $\Delta w(S) = \Delta w(S') = \widehat{w}(j)$ whenever $|S| = |S'| = j$. One can easily get a relation between $\widehat{w}$ and $\widehat{f}$ as follows:

$$\begin{aligned}
\widehat{w}(j) = \Delta w(S) &= \sum_{T: S\cup T = [m]} (-1)^{|S\cap T|+1} \Delta f(T) \\
&= \sum_{i=0}^{m} \sum_{T: S\cup T=[m], |T|=i} (-1)^{i+j+m+1} \widehat{f}(i) \\
&= \sum_{i=0}^{m} \binom{|S|}{i - (m - |S|)} (-1)^{i+j+m+1} \widehat{f}(i) \\
&= \sum_{i=0}^{m} \binom{j}{m-i} (-1)^{j+(m-i)+1} \widehat{f}(i) \\
&= \sum_{i=0}^{m} \binom{j}{i} (-1)^{i+j+1} \widehat{f}(m-i). \qquad (15)
\end{aligned}$$

In the first equality S is an arbitrary subset of size j. We now show that there exists a choice of $\widehat{f} : [m] \mapsto \mathbb{R}$ such that (a′) $\widehat{f}(i) = 0$ for $3m/8 \leq i \leq 5m/8$. Note that this will imply Δf is zero on at least $(1 - e^{-\Theta(m)})$ subsets implying condition (a). Furthermore, the choice of $\widehat{f}$ will imply that (b′) $\widehat{w}(j) \geq 1$ whenever $m > j > m/4$, and $\widehat{w}(j) \geq -N$ otherwise. This implies condition (b).

For this, let $\alpha_i := (-1)^{i+1}\widehat{f}(m-i)$. From (15), we get $(-1)^j \widehat{w}(j) = \sum_{i=0}^{m} \alpha_i \binom{j}{i}$. We consider the RHS as a polynomial over j and, in fact, the degree i polynomials $\binom{j}{i} := \frac{j(j-1)\ldots(j-i+1)}{i!}$ form what is known as the Mahler bases of rational polynomials (see, for example, [18, 19]).

As a result, one can choose α_i for $0 \leq i < 3m/8$ such that $\sum_{i=0}^{3m/8} \alpha_i \binom{j}{i}$ is *any* desired rational polynomial of degree $(3m/8 - 1)$. In particular, we choose α_i's so that

$$\sum_{i=0}^{3m/8-1} \alpha_i \binom{j}{i} = h_1(j) = 4(-1)^{5m/8} \prod_{k=m/4+1}^{5m/8-1} (j - k - 1/2). \qquad (16)$$

Similarly, $\sum_{m>i>5m/8} \alpha_i \binom{j}{i}$ can be chosen to be $j(j-1)\cdots(j-5m/8)g(j)$ for any degree $(3m/8-2)$ polynomial $g(j)$. We choose α_i's for $5m/8 < i < m$ so that

$$\sum_{5m/8<i<m} \alpha_i \binom{j}{i} = h_2(j)$$
$$= (20(m!)+4)(-1)^{m-1} j(j-1)\dots(j-5m/8) \prod_{k=5m/8+1}^{m-2} (j-k-1/2). \tag{17}$$

Finally, as promised, we let $\alpha_i = 0$ for $3m/8 \le i \le 5m/8$. We now argue that condition (b') holds. Note that $\widehat{w}(j) = (-1)^j(h_1(j)+h_2(j))$.

If $m/4 < j \le 5m/8$: From (17), $h_2(j) = 0$. Also, from (16), we get that the sign of $h_1(j)$ for $m/4 < j < 5m/8$ is precisely $(-1)^{5m/8}(-1)^{5m/8-j} = (-1)^j$. So, $(-1)^j h_1(j)$ is positive. Furthermore, the absolute value of $h_1(j)$ is at least 1, implying $\widehat{w}(j) \ge 1$.

If $5m/8 < j < m$: We use that $\widehat{w}(j) \ge (-1)^j h_2(j) - |h_1(j)|$. The former term is at least $5m!$ via a similar reasoning as above. $|h_1(j)|$, as follows from (16), is at most $4m!$. This is because the product is at most $m!$ in absolute value. This gives $\widehat{w}(j) \ge m! \ge 1$ in this range.

If $0 \le j \le m/4$ or $j = m$: Once again, we get that $\widehat{w}(j) = (-1)^j h_1(j)$ which changes its sign as j changes. However, the absolute value is at most $4m!$, so choosing $N = 5m!$, we get $\widehat{w}(j) \ge -N$.

Thus, condition (b′) is also satisfied, in turn implying that (b) is satisfied. ∎

Proof of Lemma 4.8. Consider the function f whose W-representation satisfies that $w(S) = m$ if $|S| = 1$, $w(S) = -1$ if $|S| = 2$, and $w(S) = 0$ if $|S| \ge 3$.

We first note that for any subset T and any $i, j \notin T$, we have

$$f(T+i+j) - f(T+i) - f(T+j) + f(T) = -\sum_{S:i,j\in S} w(S) = -w(i,j) = 1.$$

Therefore, the function is supermodular. So for any subset R and any $i \notin R$, we have

$$f(R+i) - f(R) \ge f(i) - f(\emptyset) = \sum_{S:i\in S} w(S) = m - (m-1) = 1.$$

Hence, the function is monotonically increasing. Note that $f(\emptyset) = 0$. We get that f is nonnegative.

Next, we will show that f is at least $1/4$ far from coverage functions. Let us partition all the 2^m subsets into groups of size 4 such that for any subset S of $[m]-i-j$, we let S, $S+i$, $S+j$, and $S+i+j$ be in the same group. Note that the function is strictly supermodular yet any coverage function must be submodular. So at least one of the four function values in each group need to be changed in each group in order to make it a coverage function. ∎

Acknowledgments. The authors wish to thank C. Seshadhri, Jan Vondrák, Sampath Kannan, Jim Geelen, Arnab Bhattacharyya, Prateek Jain, Ashwin Badanidiyuru, and Mike Saks for very fruitful conversations. DC especially thanks Sesh for illuminating conversations over the past few years, and Mike for asking insightful questions.

REFERENCES

[1] A. Badanidiyuru, *private communication*, 2013.
[2] A. Badanidiyuru, S. Dobzinski, H. Fu, R. Kleinberg, N. Nisan, and T. Roughgarden, *Sketching valuation functions*, in Symposium on ACM-SIAM Discrete Algorithms (SODA), SIAM, Philadelphia, 2012, pp. 1025–1035.

[3] M. Balcan and N. Harvey, *Learning submodular functions*, in Symposium on Theory of Computing (STOC), ACM, New York, 2011, pp. 793–802.

[4] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.

[5] L. Blumrosen and N. Nisan, *Combinatorial auctions*, in Algorithmic Game Theory, Cambridge University Press, Cambridge, 2007, pp. 267–300.

[6] D. Chakrabarty and Z. Huang, *Testing coverage functions*, in Automata, Languages, and Programming, Springer, Heidelberg, 2012, pp. 170–181.

[7] G. Cornuejols, M. Fisher, and G. Nemhauser, *Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms*, Management Sci., 23 (1977), pp. 789–810.

[8] S. Dughmi and J. Vondrák, *Limitations of randomized mechanisms for combinatorial auctions*, in Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2011, pp. 502–511.

[9] S. Dughmi, T. Roughgarden, and Q. Yan, *From convex optimization to randomized mechanisms: Toward optimal combinatorial auctions*, in Symposium on Theory of Computing (STOC), ACM, New York, 2011, pp. 149–158.

[10] S. Dughmi, *A truthful randomized mechanism for combinatorial public projects via convex optimization*, in Proceedings of the 12th ACM Conference on Electronic Commerce, ACM, New York, 2011, pp. 263–272.

[11] V. Feldman and P. Kothari, *Learning Coverage Functions and Private Release of Marginals*, Conference on Learning Theory (COLT), 2014, pp. 679–702.

[12] V. Feldman and J. Vondrák, *Optimal bounds on approximation of submodular and xos functions by juntas*, in Proceedings, FOCS, 2013.

[13] O. Goldreich and D. Ron, *On proximity oblivious testing*, in Proceedings of the 41st Annual Symposium on Theory of Computing (STOC), ACM, New York, 2009, pp. 141–150.

[14] O. Goldreich, *Combinatorial property testing (a survey)*, Randomization Methods in Algorithm Des., 43 (1999), pp. 45–59.

[15] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750.

[16] A. Krause, H. McMahan, C. Guestrin, and A. Gupta, *Robust submodular observation selection*, J. Mach. Learn. Res., 9 (2008), pp. 2761–2801.

[17] B. Lehmann, D. Lehmann, and N. Nisan, *Combinatorial auctions with decreasing marginal utilities*, Games Econom. Behav., 55 (2006), pp. 270–296.

[18] K. Mahler, *Introduction to p-Adic numbers and their Functions*, Cambridge University Press, Cambridge, 1973.

[19] A. Robert, *A Course in p-Adic Analysis*, Grad. Texts Math. 198, Springer Verlag, New York, 2000.

[20] C. Seshadhri, *Open Problems* 2: *Open Problems in Data Streams, Property Testing, and Related Topics*, http://www.cs.umass.edu/∼mcgregor/papers/11-openproblems.pdf (2011).

[21] C. Seshadhri and J. Vondrák, *Is submodularity testable?*, Algorithmica, 69 (2014), pp. 1–25.