Is submodularity testable?

C. Seshadhri*
Sandia National Labs, Livermore
scomand@sandia.gov

Jan Vondrák
IBM Almaden Research Center
jvondrak@us.ibm.com

Abstract

We initiate the study of property testing of submodularity on the boolean hypercube. Submodular functions come up in a variety of applications in combinatorial optimization. For a vast range of algorithms, the existence of an oracle to a submodular function is assumed. But how does one check if this oracle indeed represents a submodular function?

Consider a function $f : \{0,1\}^n \to \mathbb{R}$. The *distance to submodularity* is the minimum fraction of values of f that need to be modified to make f submodular. If this distance is more than $\epsilon > 0$, then we say that f is ϵ-far from being submodular. The aim is to have an efficient procedure that, given input f that is ϵ-far from being submodular, certifies that f is not submodular. We analyze a natural tester for this problem, and prove that it runs in subexponential time. This gives the first non-trivial tester for submodularity. On the other hand, we prove an interesting lower bound (that is, unfortunately, quite far from the upper bound) suggesting that this tester cannot be efficient in terms of ϵ. This involves non-trivial examples of functions which are far from submodular and yet do not exhibit too many local violations.

We also provide some constructions indicating the difficulty in designing a tester for submodularity. We construct a partial function defined on exponentially many points that cannot be extended to a submodular function, but any strict subset of these values can be extended to a submodular function.

*This work was funded by the applied mathematics program at the United States Department of Energy and performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

1 Introduction

Submodular functions have been studied in great depth in combinatorial optimization [Edm70, NWF78, FNW78, Lov83, Fra97, Sch00, IFF01]. A set function $2^U \to \mathbb{R}$ is submodular if $\forall S, T \subseteq U$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. An alternative and equivalent view of submodularity is the monotonicity of *marginal values*. For all $S \subset T$ and elements $i \notin T$, a submodular function satisfies $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)$. We will think of f as a function in $\{0,1\}^n \to \mathbb{R}$.

These functions are often used in many algorithmic applications and naturally show up when modeling utilities. It is quite common to assume that algorithms have oracle access to some submodular function: given a set S, we have access to $f(S)$. We call this a *value query*, and an oracle answering such queries a *value oracle*. Observe that, in general, the description of the submodular function f has size that is exponential in n, whereas most algorithms that use f run in polynomial time. This means that these algorithms look at a tiny fraction of f, yet their behavior depends on a global property of f. This leads to the natural question: what if the function f provided to the algorithm was *not* submodular? Could the algorithm detect this, or would it get fooled? If f is constructed by taking a submodular function and making a few changes to the values, then perhaps the performance of an algorithm would not be affected. On the other hand, if f is "significantly different" from a submodular function, the behavior of an algorithm could be different.

Let us formally explain the notion of being different from a submodular function. Since polynomial time algorithms are *sublinear* with respect to the size of f, it is natural to use property testing terminology. A function f is *ϵ-far from submodular* if at least an ϵ-fraction of the values of f needs to be changed to make it submodular. In polynomial time, can we detect that such a function is not submodular? If this is not possible, then this raises questions about algorithms for submodular functions. If the plethora of known algorithms cannot tell whether their input f is submodular or not, then this would suggest that the algorithms perhaps exploit a property more general than submodularity. (Of course, this is by no means a formal argument. One can test approximate the entropy of a *monotone* distribution in time much less than is required to test the monotonicity of a distribution [BDKR05, BFRV11].)

The main question here is whether submodularity is testable, i.e, is there a procedure that distinguishes submodular functions from those that are ϵ-far, in time polynomial in n? (This question was first posed as an open problem in [PRR03], in the context of submodularity testing over grids. Their results focused on testing over large low-dimensional grids rather than the high-dimensional hypercube $\{0,1\}^n$.) More concretely, what are the kind of structural properties of submodularity that would be useful for testing? Known property testing algorithms, especially those for functions on the hypercube, usually check for some *local property*. These algorithms check if the desired property holds in a small local neighborhood, for some randomly chosen neighborhoods. If no deviation is detected, then property testers conclude that the input function is close to the property. Do similar statements hold for submodularity? We show non-trivial bounds on the relationship between local violations of submodularity and distance from submodularity.

Analysis of property testing algorithms often shows that a function is close to a property by explicitly modifying the function to make it have the property. Usually, there is some procedural method to perform this conversion. This raises an interesting question about *partial* submodular functions: suppose one is given a *partial* function over the hypercube. This means that some set of values is defined, but the remaining are left undefined. Under what circumstances can this be extended to a submodular function? If this cannot be completed, can we provide a small certificate of this? For most natural testable properties (over functions on the hypercube, e.g. monotonicity) such small certificates do exist. Unfortunately, this is no longer true for submodularity. We present an example showing that a minimal certificate of non-extendability can be exponentially large in n,

the dimension of the hypercube.

1.1 Our results

Before we state our main theorems, let us introduce some notation.

Definition 1.1 *Denote by $\mathbf{e}_i \in \{0,1\}^n$ the canonical basis vector which has 1 in the i-th coordinate and 0 everywhere else. For vectors $x, y \in \{0,1\}^n$, we define a partial ordering where $x \leq y$ iff $x_i \leq y_i$ for all i.*

For a function $f : \{0,1\}^n \to \mathbb{R}$, $i \in [n]$ and $x \in \{0,1\}^n$ such that $x_i = 0$, we define the marginal value of i (or discrete derivative) at x as $\partial_i f(x) = f(x + \mathbf{e}_i) - f(x)$.

A function f is submodular, if for any $i \in [n]$ and $x, y \in \{0,1\}^n$ such that $x_i = y_i = 0$ and $x \leq y$, $\partial_i f(x) \geq \partial_i f(y)$.

The distance $d(f, g)$ between two functions f and g is the fraction of points x where $f(x) \neq g(x)$. Let $\mathcal{S}$ be the set of all submodular functions. The distance of f to submodularity *is* $\min_{g \in \mathcal{S}} d(f, g)$. *We say f* is ϵ-far from submodular *if the distance of f to submodularity is at least ϵ.*

Definition 1.2 *A* property tester for submodularity *is an algorithm with the following properties.*[1]

- *If f is submodular, then the algorithm answers YES with probability* 1.
- *If f is ϵ-far from submodular, then the algorithm answers NO with probability at least* 2/3.
- *The number of queries made to f is* sublinear *in the domain size, which is 2^n. (Ideally, the number of queries is polynomial in n and $1/\epsilon$.)*

Submodularity vs. monotonicity. Our first observation is that testing submodularity is at least as hard as testing monotonicity. More formally, the problem of testing monotonicity for a function $f : \{0,1\}^n \to \mathbb{R}$ can be reduced to the problem of testing submodularity for a function $f' : \{0,1\}^{n+1} \to \mathbb{R}$. We present this reduction in Section 5.

A consequence of this is that known lower bounds for monotonicity testing apply also to submodularity testing. For example, it is known that any monotonicity tester requires at least $\Omega(\sqrt{n})$ queries [FLN$^+$02].

Submodularity can be naturally viewed as "second-degree monotonicity", i.e., monotonicity of the discrete partial derivatives $\partial_i f$. So a natural test for submodularity is to simply run a monotonicity tester on the functions $\partial_i f$, and return YES if the monotonicity test passes for all i. In one direction, it is clear that for a submodular function, such a tester would always accept. However, it is not clear whether this tester would recognize functions that are far from submodular and label them as such.

Monotonicity testers search randomly for pairs $x, x + \mathbf{e}_i$ such that $f(x) > f(x + \mathbf{e}_i)$. Such a pair of points can be naturally called a "violated pair". It is known that if f is ϵ-far from monotone, then the fraction of violated pairs is $\Omega(\epsilon/n)$ [CS12]. If we want to test submodularity by reducing to a monotonicity tester in each direction, this means that we are looking for violations of the following type: $x \in \{0,1\}^n$ such that $x_i = x_j = 0$ and $f(x + \mathbf{e}_i) - f(x) < f(x + \mathbf{e}_i + \mathbf{e}_j) - f(x + \mathbf{e}_j)$. We call such violations *violated squares*.

Definition 1.3 *We call $\{x, x + \mathbf{e}_i, x + \mathbf{e}_j, x + \mathbf{e}_i + \mathbf{e}_j\}$ a* square. *A violated square is such that $f(x) + f(x + \mathbf{e}_i + \mathbf{e}_j) > f(x + \mathbf{e}_i) + f(x + \mathbf{e}_j)$. The* density *of violated squares is the number of violated squares divided by $\binom{n}{2} 2^{n-2}$.*

[1] We deal with one-sided error testers here. If we allow a probability of error in both cases, that would be a two-sided error tester.

Our main combinatorial result consists of two bounds on the relationship of the distance from submodularity and the density of violated squares.

Theorem 1.4 *Let n be a sufficiently large integer.*

- *Let $\epsilon \in (0, e^{-5})$. For any function $f : \{0,1\}^n \to \mathbb{R}$ that is ϵ-far from submodular, the density of violated squares is at least $\epsilon^{O(\sqrt{n}\log n)}$.*
- *For any $k \leq n$, there is a function $f : \{0,1\}^n \to \mathbb{R}$ which is ϵ-far from submodular, with $\epsilon = \Omega(2^{-0.104k})$, and its density of violated squares is $2^{-k/2} = O(\epsilon^{4.8})$.*

The first part of the theorem is proven through relatively basic observations. The second part is quite technical and requires a deeper understanding of submodularity.

Theorem 1.4 provides some evidence that testing submodularity is substantially different from testing monotonicity. An intuition one might get from monotonicity testing is that if a natural extension to submodularity exists, its dependence on ϵ should be relatively mild, perhaps linear or quadratic. We show that this is not the case, in particular if the dependence is a polynomial in $1/\epsilon$, the degree of the polynomial would have to be at least 5. This holds even in the range of exponentially small $\epsilon = 2^{-\Theta(n)}$, which means that $poly(n)/\epsilon^{4.8}$ queries for any polynomial in n are not enough. This might be interpreted as counterintuitive to the notion that the dependence is polynomial at all. However, we cannot rule out that the dependence is in fact polynomial.

The first part of Theorem 1.4 implies immediately that a submodularity tester that checks $q = 1/\epsilon^{O(\sqrt{n}\log n)}$ random squares succeeds with high probability[2]. Note that this is a non-adaptive tester, because the queries do not depend on the function values. This is the first testing result asymptotically better than the trivial tester checking $2^{\Theta(n)}$ squares.

Corollary 1.5 *There is a subexponential time non-adaptive tester for submodularity. This procedure samples $1/\epsilon^{O(\sqrt{n}\log n)}$ sqaures at random and rejects if any are violated. If the input f is ϵ-far from submodular, this procedure rejects with high probability.*

Extending partial functions. A *partial function* f is one that is defined on only some subset of the hypercube. Such a function is *extendable*, if the remaining values can be filled in to get a submodular function. Although the question of extending partial functions is interesting in itself, it also has some relevance to question of testing submodularity.

Any analysis of a property tester must show that if the tester accepts a function f (with high probability), then f must be ϵ-close to submodular. This is usually done by arguing that if f has a sufficiently low density of local violations, one can modify an ϵ-fraction of values and remove all "obstructions" to submodularity. Since an f that passes the tester must have a low density of local violations, f is ϵ-close. An understanding of these obstructions to submodularity is often helpful for designing testers. An obstruction is just a subset of values that cannot exist in any submodular function.

Given a partial function f that is not extendable, we would ideally like to find a small certificate for this property. Unfortunately, we will show that such certificates can be exponentially large. (As opposed to a fully defined function, which always has a small certificate — a violated square — if it is not submodular.) We give a partial function with a surprising property: The partial function f is defined on an exponentially large set and is not extendable. If any *single* value is removed, then this new function is extendable.

[2]We use "high probability" to refer to probability $> 2/3$.

Definition 1.6 *For a partial function f, let def(f) be the set of domain points where f is defined. Let $\mathcal{A} \subseteq \{0,1\}^n$. The* restriction of f to $\mathcal{A}$, $f|_{\mathcal{A}}$, *is the partial function that agrees with f on $\mathcal{A}$ and is undefined everywhere else. The partial function f is* minimally non-extendable *if $f|_{\mathcal{A}}$ is extendable for all $\mathcal{A} \subsetneq$ def(f).*

Theorem 1.7 *There exists a minimally non-extendable function f such that $|\text{def}(f)| = 2^{\Omega(n)}$.*

Summary: the difficulty in testing submodularity. To summarize, the values of f can interact in non-trivial ways to create obstructions to submodularity. Contrast this to monotonicity. A partial function f (on the hypercube) cannot be extended to a non-decreasing monotone function iff there is a pair of sets $S \subset T$ such that $f(S) > f(T)$. There is always a certificate of size 2 that a partial function cannot be extended. So this completely characterizes the obstructions to monotonicity, and is indeed one of the reasons why monotonicity testers work. Our work implies that such a simple characterization does not exist for submodularity. Indeed, as Theorem 1.7 claims, obstructions to submodularity can have an extremely complicated structure.

Functions that are far from submodular can "hide" their bad behavior. In Theorem 3.3, we show the existence of a function f with exactly *one* violated square, but making f submodular requires changing $2^{n/2}$ values. Somehow, even though the function is (in a weak sense) "far" from submodular, the only local violation that manifests itself is a single square. The functions described by the second part of Theorem 1.4 are constructed through generalizations of this example.

1.2 Previous work

Property testing, which was defined in [RS96, GGR98], is a well-studied field of theoretical computer science. Efficient testers have been given for a wide variety of combinatorial, algebraic, and geometric problems (see surveys [Fis01, Gol98, Ron01]). The problem of property testing for monotonicity over the hypercube has been studied in [GGL$^+$00, DGL$^+$99, FLN$^+$02, Fis04, FR, BCGSM12]. In particular, monotonicity of a function over $\{0,1\}^n$ can be tested using $O(n/\epsilon)$ non-adaptive queries [CS12] and $\Omega(n)$ queries are necessary [BBM11].

As mentioned earlier, the problem of testing submodularity was first raised first by [PRR03]. They considered submodularity over general grid structures (of which the hypercube is a special case). Their focus was on testing submodularity over 2-dimensional grids. Specifically, [PRR03] gave strong results for testing *Monge matrices*. Monge matrices are essentially submodular functions over the $n \times m$ integer grid. Here, the dimension is 2, but the domain in each component is large. In contrast, we are studying submodular functions over high-dimensional domains, where each component is binary. Hence, our problem is different from testing Mongeness, and we need a different set of techniques.

Another related set of results is recent work on learning and approximating submodular functions [GHIM09, BH11, CKKL12]. Here, we want to examine a function given by an oracle through polynomially many queries (which is similar to our setting) and learn sufficient information so that we are able to answer queries about the function. The difference is that in this model, the input function is guaranteed to be submodular, rather than possibly being corrupted. For example, [GHIM09] shows that we can "learn" a monotone submodular function using polynomially many queries so that afterwards we can answer value queries within a multiplicative $\widetilde{O}(\sqrt{n})$ factor, and this is optimal up to logarithmic factors. In contrast, the input function in our model might be masquerading as a submodular function but in truth be far from submodular.

Subsequence to the conference publication of this work, there has been work on efficient testers for special classes of submodular functions, called *succinct coverage functions* [CH12].

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we present our basic submodularity tester and prove the first part of Theorem 1.4. In Section 3, we present our construction of submodular functions from lattices and prove the second part of Theorem 1.4. In Section 4, we discuss extendability of submodular functions and prove Theorem 1.7. In Section 5, we present the reduction from monotonicity testing to submodularity testing. In Section 6, we discuss future directions.

2 A subexponential submodularity tester

The violated-square tester: given a function $f : \{0,1\}^n \to \mathbb{R}$ by a value oracle,

- For a parameter $q \in \mathbb{Z}$, repeat the following q times.
- Sample uniformly at random $i \neq j$ from $[n]$, and x from $\{x \in \{0,1\}^n : x_i = x_j = 0\}$. If

$$f(x) + f(x + \mathbf{e}_i + \mathbf{e}_j) > f(x + \mathbf{e}_i) + f(x + \mathbf{e}_j),$$

  i.e., if $\{x, x + \mathbf{e}_i, x + \mathbf{e}_j, x + \mathbf{e}_i + \mathbf{e}_j\}$ is a violated square, then return NO.
- If none of the tested squares is violated, then return YES.

Clearly, if the input function is submodular, the tester answers YES. We would like to understand how well this tester performs in case the input function is ϵ-far from submodular. The following observation is standard and reduces this question to a combinatorial problem about violated squares.

Observation 2.1 *The following two statements are equivalent:*

- *The violated-square tester using $q(n, \epsilon)$ queries detects every function that is ϵ-far from submodular with constant probability.*
- *For every function which is ϵ-far from submodular, the density of violated squares is $\Omega(1/q(n, \epsilon))$.*

Therefore, to understand this tester we need to understand the relationship between the distance from submodularity and the density of violated squares. In the rest of this section, our main goal is to prove the first part of Theorem 1.4, i.e., the claim that if a function is ϵ-far from submodular, then the density of violated squares must be at least $\epsilon^{O(\sqrt{n} \log n)}$. Using Lemma 2.1, this implies Corollary 1.5. First, we prove the following lemma.

Lemma 2.2 *Assume $\{x, x + \mathbf{e}_i, x + \mathbf{e}_j, x + \mathbf{e}_i + \mathbf{e}_j\}$ is a violated square. Then it is possible to decrease all the values either in $\{y : y \leq x\}$ or in $\{y : y \geq x + \mathbf{e}_i + \mathbf{e}_j\}$ by a constant so that the square $\{x, x + \mathbf{e}_i, x + \mathbf{e}_j, x + \mathbf{e}_i + \mathbf{e}_j\}$ is no longer violated and no new violated square is created.*

Proof: Denote by $d = f(x) + f(x + \mathbf{e}_i + \mathbf{e}_j) - f(x + \mathbf{e}_i) - f(x + \mathbf{e}_j)$ the "deficit" of the violated square. One way to fix this square is to decrease the value of $f(x)$ by d; however, this might create new violated squares. Instead, we decrease the value of $f(y)$ for every $y \leq x$; i.e., we define a new function $\tilde{f}(y) = f(y) - d$ for $y \leq x$, and $\tilde{f}(y) = f(y)$ otherwise. (Alternatively, we can define $\tilde{f}(y) = f(y) - d$ for $y \geq x + \mathbf{e}_i + \mathbf{e}_j$, and $\tilde{f}(y) = f(y)$ otherwise; the analysis is symmetric and we omit this case.)

Consider any other square that was previously not violated, i.e., $f(x') + f(x' + \mathbf{e}_{i'} + \mathbf{e}_{j'}) \leq f(x' + \mathbf{e}_{i'}) + f(x' + \mathbf{e}_{j'})$ and $x'_{i'} = x'_{j'} = 0$. We consider two cases:

- If $x' + \mathbf{e}_{i'} \leq x$ and $x' + \mathbf{e}_{j'} \leq x$, then we also have $x' \leq x' + \mathbf{e}_{i'} + \mathbf{e}_{j'} \leq x$ (obviously on coordinates $\ell \neq i', j'$, and also on i', j' since $x_{i'} = x_{j'} = 1$ in this case). Then, we decrease $f(x' + \mathbf{e}_{i'}) + f(x' + \mathbf{e}_{j'})$ by $2d$, and we also decrease $f(x) + f(x' + \mathbf{e}_{i'} + \mathbf{e}_{j'})$ by $2d$.

- If exactly one of $x' + \mathbf{e}_{i'} \leq x$ and $x' + \mathbf{e}_{j'} \leq x$ holds, then we still have $x' \leq x$. Thus we decrease $f(x' + \mathbf{e}_{i'}) + f(x' + \mathbf{e}_{j'})$ by d, and we also decrease $f(x) + f(x' + \mathbf{e}_{i'} + \mathbf{e}_{j'})$ by d.
- If neither $x' + \mathbf{e}_{i'} \leq x$ nor $x' + \mathbf{e}_{j'} \leq x$ holds, we do not decrease $f(x' + \mathbf{e}_{i'}) + f(x' + \mathbf{e}_{j'})$ and the square cannot become violated.

This means we can fix violated squares one by one, and the number of violated squares decreases by one every time. The cost we pay for each fix is the number of points in the cube above or below the respective square. Recall that we count the number of modified values overall, and hence what counts is the union of all the cubes modified in the process. Intuitively, it is more frugal to choose the cubes above for violated squares that are above the middle layer of the hypercube, and the cubes below for squares that are below the middle. A counting argument gives the following.

Lemma 2.3 *Let $\epsilon \in (0, e^{-5})$, $n \geq 4$ and let f have at most $\epsilon^{\sqrt{n}\log n} 2^n$ violated squares. Then these violated squares can be fixed by modifying at most $\epsilon 2^n$ values.*

Proof: Denote by B the set of bottom points for the violated squares which are below the middle layer; i.e., we have $||x||_1 \leq n/2$ for each $x \in B$. (The squares above the middle layer can be handled symmetrically.) We choose to modify the down-closed cube, $C_x = \{y \in \{0,1\}^n : y \leq x\}$, for each $x \in B$. We can fix the violated square one by one, by modifying the values in the cubes C_x. The total number of modified values is $|\bigcup_{x\in B} C_x|$. We bound the cardinality of this union by combining two simple bounds across levels of the hypercube. Denote $L_j = \{x \in \{0,1\}^n : ||x||_1 = j\}$. We have

$$\Big|\bigcup_{x\in B} C_x\Big| = \sum_{j=0}^{n/2} \Big|\bigcup_{x\in B} (C_x \cap L_j)\Big|.$$

First, by the union bound, we have

$$\Big|\bigcup_{x\in B} (C_x \cap L_j)\Big| \leq \sum_{x\in B} |C_x \cap L_j| = \sum_{x\in B} \binom{||x||_1}{j} \leq |B| \binom{n/2}{j}.$$

Secondly, we have (trivially)

$$\Big|\bigcup_{x\in B} (C_x \cap L_j)\Big| \leq |L_j| = \binom{n}{j}.$$

We choose the better of the two bounds depending on j. In particular, for $j \leq n/2 - a\sqrt{n}$, we get $\sum_{j=0}^{n/2-a\sqrt{n}} \binom{n}{j} = 2^n \Pr[X \leq n/2 - a\sqrt{n}] \leq 2^n e^{-a^2}$ where X is a binomial $Bi(n, 1/2)$ random variable and the last inequality is a standard Chernoff bound. For $j > n/2 - a\sqrt{n}$, we use $\sum_{j=n/2-a\sqrt{n}}^{n/2} |B| \binom{n/2}{j} = \sum_{j=0}^{a\sqrt{n}} |B| \binom{n/2}{j} \leq |B| n^{a\sqrt{n}}$, since the number of subsets of size at most $a\sqrt{n}$ is at most $n^{a\sqrt{n}}$. We conclude that

$$\Big|\bigcup_{x\in B} C_x\Big| = \sum_{j=0}^{n/2} \Big|\bigcup_{x\in B} (C_x \cap L_j)\Big| \leq 2^n e^{-a^2} + |B| n^{a\sqrt{n}}.$$

Let $a = \frac{1}{2}\ln(1/\epsilon)$; by assumption of the lemma we also have $|B| \leq 2^n \epsilon^{\sqrt{n}\ln n}$. For $\epsilon \in (0, e^{-5})$, this implies

$$\Big|\bigcup_{x\in B} C_x\Big| \leq 2^n e^{-(\frac{1}{2}\ln(1/\epsilon))^2} + 2^n \epsilon^{\sqrt{n}\ln n} n^{\frac{1}{2}\sqrt{n}\ln(1/\epsilon)} = (\epsilon^{\frac{1}{4}\ln(1/\epsilon)} + \epsilon^{\frac{1}{2}\sqrt{n}\ln n}) 2^n \leq \frac{1}{2}\epsilon 2^n$$

for $n \geq 4$, since $\epsilon^{\frac{1}{4}\ln(1/\epsilon)} + \epsilon^{\frac{1}{2}\sqrt{n}\ln n} \leq \epsilon^{5/4} + \epsilon^{\ln 4} \leq \frac{1}{2}\epsilon$ for $\epsilon \leq e^{-5}$. □

This lemma immediately implies the first part of Theorem 1.4. Assuming that f is ϵ-far from submodular, we get that the number of violated squares is at least $\epsilon^{\sqrt{n}\log n}2^n$ for $\epsilon \in (0, e^{-5})$, i.e., the density of violated squares is at least $\epsilon^{\sqrt{n}\log n}$.

3 Functions violating few squares, yet far from submodular

We now give a construction of functions that have large distance from submodularity but a relatively small fraction of violated squares. As we mentioned earlier, these bounds are nowhere near our positive results. Nonetheless, we are able to show a certain difference between the behavior of monotonicity and submodularity.

Our first tool to construct these functions is an interesting family of submodular functions. It is known that that the set of minimizers of a submodular function (points x minimizing $f(x)$ over $x \in \{0,1\}^n$) always forms a lattice[3] [Edm70]. We prove that conversely, for any lattice $\mathcal{L} \subset \{0,1\}^n$ there is a submodular function whose set of minimizers is exactly $\mathcal{L}$. We will then piece together these submodular functions to construct a non-submodular function with the desired properties.

3.1 Submodular functions from lattices

Lemma 3.1 *Let $\mathcal{L} \subset \{0,1\}^n$ be a lattice, i.e a set of points closed under coordinate-wise minimum and maximum. Then the following Hamming distance function is submodular:*

$$d_{\mathcal{L}}(x) = \min_{y \in \mathcal{L}} ||x - y||_1.$$

Proof: In this proof, we use the set-function notation and identify $\{0,1\}^n$ with subsets of $[n]$. In this formalism, a lattice $\mathcal{L} \subset \{0,1\}^n$ is a family of sets closed under taking unions and intersections. The distance function $d_{\mathcal{L}}$ can be written as

$$d_{\mathcal{L}}(S) = \min_{L \in \mathcal{L}} |S \Delta L|$$

where $S\Delta L$ denotes the symmetric difference between S and L. Assume that $d_{\mathcal{L}}(S) = |S\Delta U|$ and $d_{\mathcal{L}}(T) = |T\Delta V|$ for some $U, V \in \mathcal{L}$. We want to prove $d_{\mathcal{L}}(S \cup T) + d_{\mathcal{L}}(S \cap T) \leq d_{\mathcal{L}}(S) + d_{\mathcal{L}}(T)$. We prove in fact that

$$|(S \cup T)\Delta(U \cup V)| + |(S \cap T)\Delta(U \cap V)| \leq |S\Delta U| + |T\Delta V|$$

which is sufficient since $U \cup V, U \cap V \in \mathcal{L}$ by the lattice property, and therefore $d_{\mathcal{L}}(S \cup T) \leq |(S \cup T)\Delta(U \cup V)|, d_{\mathcal{L}}(S \cap T) \leq |(S \cap T)\Delta(U \cap V)|$. These two symmetric differences can be bounded as follows:

$$\begin{aligned} |(S \cup T)\Delta(U \cup V)| &= |(S \cup T) \setminus (U \cup V)| + |(U \cup V) \setminus (S \cup T)| \\ &= |S \cap \bar{U} \cap \bar{V}| + |\bar{S} \cap T \cap \bar{U} \cap \bar{V}| + |U \cap \bar{S} \cap \bar{T}| + |\bar{U} \cap V \cap \bar{S} \cap \bar{T}| \\ &\leq |S \cap \bar{U} \cap \bar{V}| + |\bar{S} \cap T \cap \bar{V}| + |U \cap \bar{S} \cap \bar{T}| + |\bar{U} \cap V \cap \bar{T}|, \end{aligned}$$

[3] A lattice is any partial order with the operations of "meet" and "join". In our setting, this means a subset of $\{0,1\}^n$ closed under taking coordinate-wise minimum and maximum. Or equivalently, a family of sets closed under taking intersections and unions.

$$\begin{aligned}|(S\cap T)\Delta(U\cap V)| &= |(S\cap T)\setminus(U\cap V)| + |(U\cap V)\setminus(S\cap T)| \\ &= |S\cap T\cap \bar{V}| + |S\cap T\cap \bar{U}\cap V| + |U\cap V\cap \bar{T}| + |U\cap V\cap \bar{S}\cap T| \\ &\leq |S\cap T\cap \bar{V}| + |S\cap \bar{U}\cap V| + |U\cap V\cap \bar{T}| + |U\cap \bar{S}\cap T|.\end{aligned}$$

Adding up the two bounds and merging terms such as $|S\cap\bar{U}\cap\bar{V}|+|S\cap\bar{U}\cap V| = |S\cap\bar{U}|$, we obtain

$$|(S\cup T)\Delta(U\cup V)| + |(S\cap T)\Delta(U\cap V)| \leq |S\cap\bar{U}| + |T\cap\bar{V}| + |U\cap\bar{S}| + |V\cap\bar{T}| = |S\Delta U| + |T\Delta V|.$$

□

Considering the known fact that the minimizers of any submodular function form a lattice, we get the following characterization.

Corollary 3.2 *Let $\mathcal{S}\subseteq\{0,1\}^N$. Then the following statements are equivalent:*

1. *$\mathcal{S}$ is a lattice.*
2. *The Hamming distance function $d_{\mathcal{S}}(x) = \min_{y\in\mathcal{S}} ||x-y||_1$ is submodular.*
3. *$\mathcal{S}$ is the set of minimizers of some submodular function.*

Proof: If $\mathcal{S}$ is a lattice, then the distance function $d_{\mathcal{S}}(x)$ is submodular by Lemma 3.1. Hence $\mathcal{S}$ is the set of minimizers of some submodular function ($d_{\mathcal{S}}$). To complete the circle, it is known that the set of minimizers of any submodular function forms a lattice. □

3.2 Functions with one violated square

We start with the following counter-intuitive result.

Theorem 3.3 *For every even n, there is a function $f:\{0,1\}^n\to\mathbb{R}$ which has exactly one violated square but $2^{n/2}$ values must be modified to make it submodular.*

We remark that this statement is tight in the sense that for any function with exactly one violated square, it is sufficient to modify $2^{n/2}$ values (which follows from Lemma 2.2). To prove Theorem 3.3, we use Lemma 3.1 which says that any lattice in $\{0,1\}^n$ yields a natural submodular function. This function does not have any violated squares. However, we will add two additional dimensions and extend the function in such a way that each point of the lattice will produce exactly one violated square. Moreover, due to the nature of the distance function, the function we construct will be a linear function (by which we mean $f(x)=\sum a_i x_i$) in a large neighborhood of each violated square. This will imply that we cannot simply change one value in each violated square if we want to make the function submodular — such changes would propagate and force many other values to be changed as well. We make this argument precise later. The construction is as follows.

Construction. *Given:* Lattice $\mathcal{L}\subset\{0,1\}^n$. *Output:* Function $f_{\mathcal{L}}:\{0,1\}^{n+2}\to\mathbb{R}$.

- We denote the arguments of $f_{\mathcal{L}}$ by (a,b,x) where $x\in\{0,1\}^n$ and $a,b\in\{0,1\}$. We call a,b "special bits".
- Let $f_{\mathcal{L}}(0,0,x) = ||x||_1 = \sum_{i=1}^n x_i$.
- Let $f_{\mathcal{L}}(1,1,x) = 1 - ||x||_1 = 1 - \sum_{i=1}^n x_i$.
- Let $f_{\mathcal{L}}(0,1,x) = f_{\mathcal{L}}(1,0,x) = d_{\mathcal{L}}(x)$, the Hamming distance function from $\mathcal{L}$.

Lemma 3.4 *The function $f_{\mathcal{L}}(a,b,x)$ constructed above has exactly $|\mathcal{L}|$ violated squares, namely $\{(0,0,x),\ (0,1,x),\ (1,0,x),\ (1,1,x)\}$ for each $x\in\mathcal{L}$.*

Proof: Observe that for any fixed $a, b \in \{0,1\}$, $f_{\mathcal{L}}(a,b,x)$ is a submodular function of x. Therefore, there is no violated square $\{z, z+\mathbf{e}_i, z+\mathbf{e}_j, z+\mathbf{e}_i+\mathbf{e}_j\}$ unless at least one of i,j is a special bit.

If exactly one of i,j is a special bit, we can assume that it is the first special bit. First assume the other special bit is 0, therefore we are looking at a square with values $f_{\mathcal{L}}(0,0,x)$, $f_{\mathcal{L}}(1,0,x)$, $f_{\mathcal{L}}(0,0,x+\mathbf{e}_i$, $f_{\mathcal{L}}(1,0,x+\mathbf{e}_i)$. By construction, we know that $f_{\mathcal{L}}(0,0,x+\mathbf{e}_i)-f_{\mathcal{L}}(0,0,x)=1$ and $f_{\mathcal{L}}(1,0,x+\mathbf{e}_i)-f_{\mathcal{L}}(1,0,x)=d_{\mathcal{L}}(x+\mathbf{e}_i)-d_{\mathcal{L}}(x)\leq 1$, therefore the square cannot be violated. Similarly, if the other special bit is 1, we are looking at a square with values $f_{\mathcal{L}}(0,1,x)$, $f_{\mathcal{L}}(1,1,x)$, $f_{\mathcal{L}}(0,1,x+\mathbf{e}_i$, $f_{\mathcal{L}}(1,1,x+\mathbf{e}_i)$. Here, we always have $f_{\mathcal{L}}(1,1,x+\mathbf{e}_i)-f_{\mathcal{L}}(1,1,x)=-1$, and $f_{\mathcal{L}}(0,1,x+\mathbf{e}_i)-f_{\mathcal{L}}(0,1,x)=d_{\mathcal{L}}(x+\mathbf{e}_i)-d_{\mathcal{L}}(x)\geq -1$. So again, the square cannot be violated.

Finally, consider a square where i,j are exactly the special bits. The square has values $f_{\mathcal{L}}(0,0,x)$, $f_{\mathcal{L}}(0,1,x)$, $f_{\mathcal{L}}(1,0,x)$, $f_{\mathcal{L}}(1,1,x)$. Observe that $f_{\mathcal{L}}(0,0,x)+f_{\mathcal{L}}(1,1,x)=1$, and $f_{\mathcal{L}}(0,1,x)+f_{\mathcal{L}}(1,0,x)=2d_{\mathcal{L}}(x)$. The square is violated if and only if $2d_{\mathcal{L}}(x)<1$, i.e., when $x\in\mathcal{L}$. This means that we have a one-to-one correspondence between violated squares and the points of the lattice. □

Thus we can generate functions with a prescribed number of violated squares, depending on our initial lattice $\mathcal{L}$. The simplest example is generated by $\mathcal{L}=\{x_0\}$ being a 1-point lattice. In this case, it is easy to verify directly that the function $d_{\mathcal{L}}(x)=||x-x_0||_1$ is submodular, and hence our construction produces exactly one violated square.

The second part of our argument, however, should be that such a function is not close to submodular. In particular, consider $\mathcal{L}=\{x_0\}$ where $||x_0||_1=n/2$. Suppose that we want to modify some values so that the function $f_{\mathcal{L}}$ becomes submodular. We certainly have to modify at least one value in the violated square $\{(a,b,x_0): a,b\in\{0,1\}\}$. However, for each fixed choice of $a,b\in\{0,1\}$, the function $f_{\mathcal{L}}(a,b,x)$ is linear. The last point in our argument is that it is impossible to modify a small number of values "in the middle" of a linear function (with many values both above and below), so that the resulting function is submodular. First, we prove the following lemma.

Lemma 3.5 *Suppose $f:\{0,1\}^n\to\mathbb{R}$ is a submodular function and $f(\mathbf{0})>0$. Then there are at least 2^{n-1} points $x\in\{0,1\}^n$ such that $f(x)\neq 0$.*

Note that this is tight, for example by taking $f(x)=1-x_1$.

Proof: We prove the statement by induction on n. Obviously it is true for $n=1$. For $n>1$, we partition the cube $\{0,1\}^n$ as follows: for each $i\in[n]$, let

$$Q_i=\{x\in\{0,1\}^n : x_1=\ldots=x_{i-1}=0, x_i=1\}.$$

In other words, Q_i is the set of points such that the first nonzero coordinate is x_i. Observe that Q_i is a subcube of $\{0,1\}^n$ and the minimum point in Q_i is $\mathbf{e}_i$. We have $\{0,1\}^n=\{\mathbf{0}\}\cup\bigcup_{i=1}^n Q_i$. Now consider a submodular function $f:\{0,1\}^n\to\mathbb{R}$ such that $f(\mathbf{0})>0$. We consider two cases.

If there is coordinate i such that $f(\mathbf{e}_i)\leq 0$, then the discrete derivative $\partial_i f(\mathbf{0})$ is negative. By submodularity, $\partial_i f$ must be negative everywhere. Hence, for any point x such that $x_i=0$, at least one of $f(x), f(x+\mathbf{e}_i)$ is nonzero.

The other case is that $f(\mathbf{e}_i)>0$ for all $i\in[n]$. Then we apply the inductive hypothesis to each Q_i (using the fact that its minimum point is $\mathbf{e}_i$), which implies that at least $\frac{1}{2}|Q_i|$ values in Q_i are nonzero. By adding up the contributions from $Q_1,\ldots,Q_n$, we conclude that at least half of all the values in $\{0,1\}^n$ are nonzero. □

To rephrase the lemma, we can start with a zero function on $\{0,1\}^n$, increase the value of $f(\mathbf{0})$ to a positive value, and ask how many other values we have to modify to make the function submodular.

The lemma says that at least $2^{n-1}-1$ additional values must be modified. In fact, adding a linear function to a submodular function preserves submodularity, so we can apply the same argument to any linear function: It is impossible to increase the value of a linear function at the lowest point of a cube, without changing a lot of other values in the cube.

Note that it is possible to *decrease* the value of a linear function at the lowest point of a cube and this does not create any violation of submodularity. What is impossible is to decrease the value "in the middle" of a linear function, without changing a lot of other values. This is the content of the next lemma.

Lemma 3.6 *Suppose n is even, $f : \{0,1\}^n \to \mathbb{R}$ is a submodular function and $f(x) < 0$ for some $x, ||x||_1 = n/2$. Then there are at least $2^{n/2}$ points $y \in \{0,1\}^n$ such that $f(y) \neq 0$.*

This lemma is also tight, by taking $f(y) = -1$ whenever $y \leq x$ and $f(y) = 0$ otherwise.

Proof: Consider $Q = \{y \in \{0,1\}^n : y \leq x\}$; this is a subcube of dimension $n/2$ embedded in $\{0,1\}^n$, hence $|Q| = 2^{n/2}$. If $f(y) \neq 0$ for all $y \in Q$, we are done. Therefore, assume that there is some point $y \in Q$ such that $f(y) = 0$. Then consider a monotone path from y to x; there must be an edge $(y', y' + \mathbf{e}_i)$ of negative marginal value. By submodularity, all edges $(z', z' + \mathbf{e}_i)$ for $z' \geq y'$ must have negative marginal value. There are at least $2^{n/2}$ such edges, since all the $n/2$ zero bits in x are also zero in y' and can be increased arbitrarily to obtain a point $z' \geq y'$. Each of these (disjoint) edges $(z', z' + \mathbf{e}_i)$ contains a point of nonzero value, and hence there are at least $2^{n/2}$ such points. □

Now we can complete the proof of Theorem 3.3.

Proof: [Theorem 3.3] Consider the function $f_{\mathcal{L}} : \{0,1\}^{n+2} \to \mathbb{R}$ defined as above using a 1-point lattice $\mathcal{L} = \{x_0\}$ where $||x_0||_1 = n/2$. By Lemma 3.4, $f_{\mathcal{L}}$ has exactly one violated square. Note that by construction, for each fixed $a, b \in \{0,1\}$, the function $f_{\mathcal{L}}(a, b, x)$ is linear as a function of x.

Suppose $f' : \{0,1\}^{n+2} \to \mathbb{R}$ is a submodular function closest to $f_{\mathcal{L}}$. Since f has a violated square $\{(0,0,x_0), (0,1,x_0), (1,0,x_0), (1,1,x_0)\}$, f' must differ from $f_{\mathcal{L}}$ on at least one of these values. Fix $a, b \in \{0,1\}$ such that $f'(a,b,x_0) \neq f_{\mathcal{L}}(a,b,x_0)$ and consider the function $f'(a,b,x) - f_{\mathcal{L}}(a,b,x)$ as a function of x. Since $f_{\mathcal{L}}$ is linear, $f' - f_{\mathcal{L}}$ is submodular as a function of x (for fixed a, b). We have $(f' - f_{\mathcal{L}})(x_0) \neq 0$. If $(f' - f_{\mathcal{L}})(x_0) > 0$, we apply Lemma 3.5 to the cube $\{y : y \geq x_0\}$; if $(f' - f_{\mathcal{L}})(x_0) < 0$, we apply Lemma 3.6. In both cases, we conclude that there are at least $2^{n/2}$ values $x \in \{0,1\}^n$ such that $f'(x) \neq f_{\mathcal{L}}(x)$. Therefore, $f_{\mathcal{L}}$ is $2^{-n/2}$-far from submodular. □

3.3 Boosting the example to increase distance

Observe that in Theorem 3.3, the relationship between relative distance to submodularity and density of violated squares is quadratic: we have relative distance $\epsilon = 2^{-n/2}$ and density of violated squares $\simeq \epsilon^2 = 2^{-n}$. In order to prove the second part of Theorem 1.4, we need to consider a denser lattice. Since the regions where the function is linear will be more complicated here, we need a more general statement to argue about the number of values that must be fixed to make a function submodular.

Lemma 3.7 *Let $f : \{0,1\}^n \to \mathbb{R}$ be submodular on a down-closed set[4] $\mathcal{D} \subset \{0,1\}^n$. If $f(0) > 0$ then there are at least $\frac{1}{n+1}|\mathcal{D}|$ points $y \in \mathcal{D}$ such that $f(y) \neq 0$.*

[4] A set $\mathcal{D} \subset \{0,1\}^n$ is down-closed if $y \in \mathcal{D}$, $x \leq y$ implies $x \in \mathcal{D}$. We call a function on $\mathcal{D}$ submodular, if $f(x + \mathbf{e}_i) + f(x + \mathbf{e}_j) \geq f(x) + f(x + \mathbf{e}_i + \mathbf{e}_j)$ whenever $x + \mathbf{e}_i + \mathbf{e}_j \in \mathcal{D}$.

This is also tight - consider, for example, $\mathcal{D} = \{0, \mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $f(x) = 1 - ||x||_1$.

Proof: Suppose $f(y) = 0$ for some $y \in \mathcal{D}$. Then let $x \leq y$ be minimal such that $f(x) \leq 0$. Since x is minimal (and cannot be 0 because $f(0) > 0$), for every $i \in [n]$ such that $x_i = 1$ we have $f(x - \mathbf{e}_i) > 0$. (Note that $x, x - \mathbf{e}_i \in \mathcal{D}$.) Hence $f(x) - f(x - \mathbf{e}_i) < 0$ and by submodularity $f(y) - f(y - \mathbf{e}_i) < 0$. Since $f(y) = 0$, this implies that $f(y - \mathbf{e}_i) > 0$. In this case, we call $y - \mathbf{e}_i$ a witness for y.

To summarize, for every $y \in \mathcal{D}$ we have either $f(y) \neq 0$ or $f(y - \mathbf{e}_i) \neq 0$ for some witness $y - \mathbf{e}_i$ of y. Since every point can serve as a witness for at most n other points, the number of nonzero values must be at least $|\mathcal{D}|/(n+1)$. □

Now we are ready to prove the second part of Theorem 1.4.

Proof: We define $\mathcal{L} \subset \{0,1\}^n$ as follows:

- Consider an even n and partition $[n]$ into pairs $\{2i-1, 2i\}, 1 \leq i \leq n/2$.
- Let $\mathcal{L} = \{x \in \{0,1\}^n : \forall i \in [n/2]; x_{2i-1} = x_{2i}\}$.

Obviously, this is a lattice. In fact, it is isomorphic to a cube of dimension $n/2$. The function $f_{\mathcal{L}} : \{0,1\}^{n+2} \to \mathbb{R}$ based on this lattice has exactly $2^{n/2}$ violated squares, due to Lemma 3.4. It remains to bound the distance of $f_{\mathcal{L}}$ from submodularity.

To that end, focus on the "middle layer" of the lattice, $\mathcal{M} = \{x \in \mathcal{L} : ||x||_1 = n/2\}$. Points in $\mathcal{M}$ have exactly half of the pairs of coordinates equal to $(0,0)$ and half equal to $(1,1)$. For each such point x, consider points $y \geq x$ such that y still has the same pairs equal to $(1,1)$ as x. Formally, let

$$Q_x = \{y \geq x : \forall i \in [n/2]; y_{2i-1} = y_{2i} = 1 \Rightarrow x_{2i-1} = x_{2i} = 1\}.$$

The reason for this definition is that for any point $y \in Q_x$, it is possible to trace it back to x (by zeroing out all the pairs which are not equal to $(1,1)$, we obtain x). Hence the sets Q_x are disjoint. The path from y to x is also the shortest path from y to a point of the lattice, because it is necessary to modify all pairs which are equal to $(1,0)$ or $(0,1)$. In other words, $d_{\mathcal{L}}(y) = ||x - y||_1$ for all $y \in Q_x$. This implies that the function $f_{\mathcal{L}}(a, b, y)$ for any fixed a, b is linear as a function of $y \in Q_x$.

Similarly, we define Q_x^- as the set of points $y \leq x$ such that the set of $(0,0)$ pairs is the same in y and x. Again, each $y \in Q_x^-$ can be traced back to x and so the sets Q_x^- are disjoint.

Let us compute the size of Q_x (or Q_x^-) for a point $x \in \mathcal{M}$. We have $n/4$ pairs of value $(0,0)$ which can be modified, and we have 3 choices for each (we avoid $(1,1)$ for such pairs). Therefore, $|Q_x| = 3^{n/4}$. Similarly, $|Q_x^-| = 3^{n/4}$ as well.

Our final argument is that in order to make $f_{\mathcal{L}}$ submodular, we have to fix many values for each $x \in \mathcal{M}$. Let us assume that f' is a submodular function closest to $f_{\mathcal{L}}$. Since $f_{\mathcal{L}}$ has a violated square $\{(0,0,x), (0,1,x), (1,0,x), (1,1,x)\}$ for each $x \in \mathcal{M}$, f' must be different from $f_{\mathcal{L}}$ on at least one point in each such square. More specifically, f' must be larger than $f_{\mathcal{L}}$ for one of the points $(0,1,x), (1,0,x)$ or f' must be smaller than $f_{\mathcal{L}}$ for one of the points $(0,0,x), (1,1,x)$.

Fix $x \in \mathcal{M}$ and a, b so that $f'(a,b,x)$ differs from $f_{\mathcal{L}}(a,b,x)$ as above. Since $f_{\mathcal{L}}$ is linear on Q_x, $f' - f_{\mathcal{L}}$ is submodular on Q_x and $(f' - f_{\mathcal{L}})(a,b,x) \neq 0$. If $a \neq b$, we must have $(f' - f_{\mathcal{L}})(a,b,x) > 0$. Then applying Lemma 3.7 to the set $Q_x - x$, we conclude that $f' - f_{\mathcal{L}}$ must be nonzero on at least $\frac{1}{n+1}|Q_x|$ points in Q_x.

In the other case, $a = b$, we have $(f' - f_{\mathcal{L}})(a,b,x) < 0$. Note that in this case $f_{\mathcal{L}}$ is linear on all of $\{0,1\}^n$ and $f' - f_{\mathcal{L}}$ is submodular everywhere. Then we use arguments similar to Lemma 3.6. From the proof of Lemma 3.6, we obtain that either $(f' - f_{\mathcal{L}})(a,b,y) < 0$ for all $y \in Q_x^-$, or else there is $i \in [n]$ such that $(f' - f_{\mathcal{L}})(a,b,x) < (f' - f_{\mathcal{L}})(a,b,x - \mathbf{e}_i)$. This implies that all edges of direction $\mathbf{e}_i$ above this edge have negative marginal value. I.e., at least half of the points in $Q_x \cup (Q_x - \mathbf{e}_i)$ must have nonzero value in terms of $f' - f_{\mathcal{L}}$.

This holds for every lattice point in the middle layer $\mathcal{M}$. Therefore, each lattice point $x \in \mathcal{M}$ contributes at least $3^{n/4}/(n+1)$ points y in Q_x, or Q_x^-, or $Q_x \cup (Q_x - \mathbf{e}_i)$ for some $i \in [n]$, such that $(f' - f_{\mathcal{L}})(y) \neq 0$. There are $\binom{n/2}{n/4} = \Omega(2^{n/2}/n)$ points in $\mathcal{M}$. While the sets Q_x for $x \in \mathcal{M}$ are disjoint, the sets $Q_x - \mathbf{e}_i$ are not; a nonzero point for $f' - f_{\mathcal{L}}$ can be possibly contained in up to n sets $Q_x - \mathbf{e}_i$ for different $x \in \mathcal{M}$ and $i \in [n]$. Taking this overlap factor of n into account, we get by a standard counting argument that $f' - f_{\mathcal{L}}$ has at least $\Omega(2^{n/2}3^{n/4}/n^3)$ nonzero points. This means that the distance ϵ of $f_{\mathcal{L}}$ from being submodular is $\epsilon = \Omega(2^{-n/2}3^{n/4}/n^3)$. A calculation reveals that $\epsilon = \Omega(2^{-0.104n})$, while the density of violated squares is $2^{-n/2} = O(\epsilon^{4.8})$.

To obtain the statement of the theorem, we consider $n = k + \ell$ and we construct the above example on k coordinates, so the distance from submodularity is $\Omega(2^{-0.104k})$ and the density of violated squares is $2^{-k/2}$. We extend $f_{\mathcal{L}} : \{0,1\}^k \to \mathbb{R}$ to a function $\tilde{f} : \{0,1\}^{k+\ell} \to \mathbb{R}$ in such a way that $\tilde{f}$ does not depend on the new ℓ coordinates: if y denotes the new coordinates, $\tilde{f}(x, y) = f_{\mathcal{L}}(x)$. There are no violated squares involving the new coordinates and hence the density of violated squares remains unchanged. To make $\tilde{f}$ submodular, we must make $\tilde{f}(x, y) = f_{\mathcal{L}}(x)$ submodular for each fixed value of y. Hence, the relative distance of $\tilde{f}$ from being submodular is also unchanged. □

4 Path certificates for submodular extension

Given a partial function f, we get a precise characterization of when f is submodular-extendable using LP duality. In this subsection, f will be some fixed partial function. We remind the reader that $\text{def}(f)$ is the set of domain points where f is defined. We denote $\mathcal{B} = \{0,1\}^n$, $\mathcal{D} = \text{def}(f)$, and $\mathcal{U} = \mathcal{B} \setminus \mathcal{D}$. We use *set* to refer to a domain point in $\mathcal{B}$. We say *edge* to refer to an adjacent pair in $\mathcal{B}$, and abusing notation, use $\mathcal{B}$ to refer to the undirected graph on the boolean hypercube. We use the following notation for a domain point $S \in \{0,1\}^n$ and edge $e = (S, S \cup \{i\})$:

- $A^+(S) = \{(S, S \cup \{i\}) |\ i \notin S\}$
- $A^-(S) = \{(S \backslash \{i\}, S) |\ i \in S\}$
- $\Gamma^+(e) = \{(S \cup \{j\}, S \cup \{i,j\}) |\ j \notin S\}$
- $\Gamma^-(e) = \{(S \backslash \{j\}, S \cup \{i\} \backslash \{j\}) |\ j \in S \backslash \{i\}\}$

We will write an LP that is feasible iff f is submodular-extendable. We associate a variable x_e for every e of the hypercube, and a variable w_S for every set $S \in \mathcal{U}$. Again, we set $e = (S, S \cup \{i\})$ for convenience.

$$\begin{aligned}
\forall e \text{ and } e' \in \Gamma^+(e), &\quad x_e - x_{e'} \geq 0 \\
\forall S \in \mathcal{D} \text{ and } S \cup \{i\} \in \mathcal{D}, &\quad x_e = f(S \cup \{i\}) - f(S) \\
\forall S \in \mathcal{D} \text{ and } S \cup \{i\} \in \mathcal{U}, &\quad x_e - w_{S \cup \{i\}} = -f(S) \\
\forall S \in \mathcal{U} \text{ and } S \cup \{i\} \in \mathcal{D}, &\quad x_e + w_S = f(S \cup \{i\}) \\
\forall S \in \mathcal{U} \text{ and } S \cup \{i\} \in \mathcal{U}, &\quad x_e - w_{S \cup \{i\}} + w_S = 0
\end{aligned}$$

This LP exactly captures the notion of submodular extension. Suppose this LP had a feasible point. If we set $f(S) = w_S$ for each $S \in \mathcal{U}$, the resulting function is submodular. Furthermore, any valid submodular extension of f must satisfy this LP.

Using Farkas' lemma, if this is infeasible, then we can derive a contradiction from these equations. So, we have dual variables $y_{e,e'}$ for each pair of edges and z_e for each edge. We will refer to this as

the "dual LP".

$$\begin{aligned} \forall e, \quad & \textstyle\sum_{e' \in \Gamma^+(e)} y_{e,e'} = \sum_{e' \in \Gamma^-(e)} y_{e',e} + z_e \\ \forall S \in \mathcal{U}, \quad & \textstyle\sum_{e \in A^+(S)} z_e = \sum_{e \in A^-(S)} z_e \\ \forall e, e' \in \Gamma^+(e), \quad & y_{e,e'} \geq 0 \\ & \textstyle\sum_{S \in \mathcal{D}} [\sum_{e \in A^-(S)} z_e - \sum_{e \in A^+(S)} z_e] f(S) < 0 \end{aligned}$$

Hence, if f is *not* submodular extendable, then this LP is feasible.

We introduce a series of definitions that will help in formalizing certificates of non-extendability. We consider the standard partial order by set containment on the domain $\{0,1\}^n$. For two edges e, e' of $\{0,1\}^n$, we denote $e \prec e'$ if e and e' are along the same dimension and the smaller end of e is less than the smaller end of e' (smaller according to the partial order on the domain). Note that this definition is the same for directed and undirected edges.

A *directed path* in the hypercube is a path in $\mathcal{B}$ where edges are oriented along the path, and the terminals are in $\mathcal{D}$. Formally, any set of ordered pairs $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{r-1}, v_r)\}$ where $v_1, v_r \in \mathcal{D}$ forms a directed path. Similarly, we can define directed cycles. Let $\mathbf{P}$ be a collection of directed cycles or directed paths in the hypercube with endpoints in $\mathcal{D}$. These paths are not necessarily simple and could contains many copies of a single edge.

- An edge is *upward* if it is directed from the lower end to the upper (according to the partial order on $\mathcal{B}$), and *downward* otherwise.
- Let $\mathbf{U}$ be the multiset of upward edges of $\mathbf{P}$ and $\mathbf{D}$ be the multiset of downward edges (so we keep as many copies of edge e as occurrences in $\mathbf{P}$).
- Let G be the bipartite graph (with links, instead of edges) between $\mathbf{U}$ and $\mathbf{D}$, where an edge $e \in \mathbf{U}$ is linked to $e' \in \mathbf{D}$ if $e \prec e'$. The set of paths $\mathbf{P}$ is *matched* if there is a perfect matching in G.
- The *value* of a directed path $\mathcal{P}$, $val(\mathcal{P})$, that starts at $S \in \mathcal{D}$ and ends at $T \in \mathcal{D}$ is $f(T) - f(S)$. Cycles have value 0. The value of $\mathbf{P}$ is the sum of values of the paths in $\mathbf{P}$.
- If $\mathbf{P}$ is matched and has negative value, then $\mathbf{P}$ is referred to as a *path certificate*.

We use $\mathbf{C}$ to denote the directed graph where vertices are *edges* of the hypercube, and there is a directed link from e to every member of $\Gamma^+(e)$. This gives n disconnected graphs, each of which is a directed hypercube in $n-1$ dimensions.

The significance of these definitions should hopefully become clear with the next claim.

Claim 4.1 *Suppose there exists a feasible integral solution $\mathbf{y}, \mathbf{z}$ for the dual LP. Then there exists a path certificate $\mathbf{P}$ for f.*

Proof: Let us first understand a feasible solution for the dual LP. We think of the z_e's as forming a flow. The second set of equalities is a flow conservation constraint for all vertices in $\mathcal{U}$. Hence, we can think of the z_e's is a flow where the terminals are $\mathcal{D}$. Precisely, z_e is the flow in e from the lower end to the higher end (in $\mathcal{B}$).

The first constraint is a little stranger[5]. Think of $y_{e,e'}$ as a flow in $\mathbf{C}$. Note that this is always positive. We do not have a flow conservation condition, because of the extra z_e. Add a extra terminal for every e that is attached to the vertex e in $\mathbf{C}$. This is called the terminal e in $\mathbf{C}$. Think of z_e amount of flow being injected (if $z_e \geq 0$) or removed (if $z_e < 0$) from terminal e. Then, we have a legitimate flow in $\mathbf{C}$ represented by the $y_{e,e'}$'s and z_e's.

[5]By that we mean, somewhat different, and not an unknown dwarf.

Starting from an integral $\mathbf{y}, \mathbf{z}$, we obtain $\mathbf{P}$ by a flow decomposition of $\mathbf{z}$. Since $\mathbf{z}$ is integral, we can decompose $\mathbf{z}$ into a collection of directed paths (with terminals in $\mathcal{D}$) and cycles. Furthermore, if z_e is positive (resp. negative), then the number of upward (resp. downward) occurrences of e is $|z_e|$ and the number of downward (resp. upward) occurrences is 0. Thus, any edge is always either upward or downward in $\mathbf{P}$. Observe that the contribution of a path $\mathcal{P} \in \mathbf{P}$ from S to T to the objective of the dual LP is exactly $f(T) - f(S)$. Hence, $val(\mathbf{P})$ is negative.

We now need to construct the matching between $\mathbf{U}$ and $\mathbf{D}$. For this, we perform a flow decomposition of $\mathbf{y}$. Since the flow is positive, this gives a collection of paths $\mathbf{Q}$ in $\mathbf{C}$, where each path goes from an edge e to e', $e \prec e'$. Furthermore, if z_e is positive (resp. negative), the number of paths that start (resp. end) at e is $|z_e|$ and those that end (resp. start) at e is 0. Hence, we can get a perfect matching between $\mathbf{U}$ and $\mathbf{D}$ by matching (a copy of) e to (a copy of) e' for each path in $\mathbf{Q}$ going from e to e'. This completes the path certificate. □

Lemma 4.2 *The partial function f is not submodular-extendable iff f contains a path certificate.*

Proof: Suppose $\mathbf{P}$ is a path certificate, but f can be extended to a submodular function f'. Let $\mathbf{U}$ be the multiset of upward edges in $\mathbf{P}$ and $\mathbf{D}$ the multiset of downward edges. We have a perfect matching between $\mathbf{U}$ and $\mathbf{D}$. Consider a matched pair (e, e'). We have $e \prec e'$. For directed edge $e = (S, T)$, define $g(e)$ to be $f'(T) - f'(S)$. By submodularity of f', $g(e) + g(e') \geq 0$. Summing over all matched pairs, $\sum_{e \in \mathbf{P}} g(e) \geq 0$.

Take a directed path $\mathcal{P} \in \mathbf{P}$. Considering $\mathcal{P}$ as a multiset of directed edges, observe that $val'(\mathcal{P}) = \sum_{e \in \mathcal{P}} g(e)$, where val' is computed over f'. Because $\mathcal{P}$ is a directed path, this is a telescoping sum leading to $val(\mathcal{P}) = f'(T) - f'(S)$, where $\mathcal{P}$ starts at S and ends at T. Note that $val(\mathcal{P})$ is the same in f and f', since f' extends f. Hence, $val'(\mathcal{P}) = val(\mathcal{P})$ and $\sum_{\mathcal{P} \in \mathbf{P}} val(\mathcal{P}) \geq 0$. This contradicts the fact that $\mathbf{P}$ is a path certificate.

Suppose f cannot be extended to a submodular function. We will produce a path certificate in f. By Farkas' lemma, the dual LP is feasible and there exists a rational feasible point. By scaling, there exists an integral feasible point. We now invoke Claim 4.1. □

A path in $\mathbf{P}$ is called a *singleton* if it consists of only a single edge. We will prove "clean-up" claims for path certificates.

Claim 4.3 *Let f be a partial function such that for any square of $\mathcal{B}$, at most 2 points are present in def(f). Let f contain a path certificate $\mathbf{P}$. There exists a path certificate $\mathbf{Q}$ with no singleton edge.*

Proof: We will show how to remove any singleton in $\mathbf{P}$ and give an "equivalent" certificate $\mathbf{Q}$. The value will remain the same. Suppose there is a singleton path consisting of upward edge e. Some downward edge e', $e' \succ e$ must occur in path $\mathcal{P} \in \mathbf{P}$. Let $e = (S, S \cup \{i\})$ and $e' = (T \cup \{i\}, T)$, for some $S \subset T$. We will split $\mathcal{P}$ into two paths. Let $\mathcal{P}_1$ be the portion of $\mathcal{P}$ before e' and $\mathcal{P}_2$ be the portion after e'. Note that $\mathcal{P}_1$ ends at $T \cup \{i\}$ and $\mathcal{P}_2$ starts at T. Consider a downward path $\mathcal{Q}_1$ from $T \cup \{i\}$ to $S \cup \{i\}$ and a parallel upward path $\mathcal{Q}_2$ from S to T. Observe that there is a perfect matching between the edges of $\mathcal{Q}_1$ to those of $\mathcal{Q}_2$.

Consider the path $\mathcal{Q}'_1$ formed by joining $\mathcal{P}_1$ to $\mathcal{Q}_1$. Similarly, $\mathcal{Q}'_2$ is formed by joining $\mathcal{Q}_2$ to $\mathcal{P}_2$. Note that $\mathcal{Q}'_1$ ends at $S \cup \{i\}$ and $\mathcal{Q}'_2$ starts at S. To get $\mathbf{Q}$, we replace the singleton $\{e\}$ and $\mathcal{P}$ in $\mathbf{P}$ by $\mathcal{Q}'_1$ and $\mathcal{Q}'_2$. The collection $\mathbf{Q}$ is completely matched. The edges in $\mathcal{Q}_1$ and $\mathcal{Q}_2$ (matched to each other) are disjoint. The singleton edge e starts at S and ends at $S \cup \{i\}$. So $val(\mathcal{Q}'_1) + val(\mathcal{Q}'_2) = val(e) + val(\mathcal{P})$. and $val(\mathbf{Q}) = val(\mathbf{P})$.

Suppose $\mathcal{Q}'_1$ is a singleton. Then e and $\mathcal{Q}'_1$ are distinct edges incident to $S \cup \{i\}$ with both ends defined. This contradicts the assumption that no square has more than 2 points in $def(f)$. Arguing similarly for $\mathcal{Q}'_2$, $\mathcal{Q}'_1$ and $\mathcal{Q}'_2$ are at least of length 2. The total number of singletons has decreased by 1. We can repeatedly apply this procedure, and remove all singletons. □

Claim 4.4 *Let f be a partial function containing a path certificate $\mathbf{P}$. There exists a path certificate $\mathbf{P}'$ where no edge is matched to a copy of itself.*

Proof: Let $e = (S, S \cup \{i\})$. Suppose path $\mathcal{P}_u$ contains edge e upwards, $\mathcal{P}_d$ contains it downwards, and these copies of e were matched to each other. We can split $\mathcal{P}_u$ into portions $\mathcal{P}_{1,u}$ and $\mathcal{P}_{2,u}$ such that the former is the part before e and the latter is after e. Similarly, we can get $\mathcal{P}_{1,d}$ and $\mathcal{P}_{2,d}$. Note that $\mathcal{P}_{1,u}$ ends at S and $\mathcal{P}_{2,d}$ starts at S. Similarly, $\mathcal{P}_{2,u}$ ends at $S \cup \{i\}$ and $\mathcal{P}_{1,d}$ starts at $S \cup \{i\}$. We can combine $\mathcal{P}_{1,u}$ and $\mathcal{P}_{2,d}$ to get a path $\mathcal{P}'_1$. Similarly, we get $\mathcal{P}'_2$. We replace $\mathcal{P}_u$ and $\mathcal{P}_d$ by $\mathcal{P}'_1$ and $\mathcal{P}'_2$. Note that the sum of values does not change. Also, the only edges removed are the upward and downward copies of e and the matching on the remaining edges stays the same. We can repeatedly perform this operation to remove all such matched pairs between copies of edges. □

4.1 Large minimal certificates

We will start by giving a construction of a long cycle in $\mathcal{B}$ with some special properties. This cycle will be a sort of "frame" on which we can define f. Roughly speaking, the values of f are going to be on this cycle. On this cycle, we will find a large set of paths of length 2, and the endpoints of these will be defined.

These paths will collectively form the path certificate, showing that f is non-extendable. Since they form a cycle, every single path is essential for maintaining the matching property of path certificates. Hence, even if a single value in f is removed, the path certificate breaks down, and the resulting partial function is extendable.

The simple cycle will be obtained by performing a series of moves in $\mathcal{B}$. An *upward* (resp. *downward*) step is one where some coordinates is incremented (resp. *decremented*). We will assume that $n = 2m + 4$. The cycle will only involve points in the $m+1, m+2, m+3, m+4$ levels of B. We will call these levels the $1, 2, 3, 4$ levels. Any point is represented as $(b_1, b_2, b_3, b_4, S, T)$, where b_i's are bits, and S and T are sets on m elements. We will denote the starting (and hence, ending) point of the cycle to be $(0, 0, 1, 0, \emptyset, [m])$, where $[m]$ represents the complete set on m elements. The cycle $\mathcal{C}$ has the following properties:

- The cycle is simple, i.e., does not intersect itself.
- The cycle can be divided into a sequence of contiguous *chunks* of three steps. Every odd (resp. even) chunk has three upward (resp. downward) steps. There are an even number of chunks.
- The cycle has $M \geq 2^m$ chunks.
- Let the ith chunk be denoted by K_i. The second edge e of K_i is parallel to the first edge e' of $K_{i+1 (\mathrm{mod}\ M)}$. Suppose i is odd. Then K_i has upward steps, and hence $e' \succ e$. Similarly, if i is even, $e' \prec e$.

A crucial combinatorial property of the hypercube that we use is the existence of Hamiltonian circuits (Gray codes are one such well-known construction [Gil58]). We set $\mathcal{H}$ to be a (directed) Hamiltonian circuit on the m-dimensional hypercube. For any set $R \in \mathcal{H}$, $s(R)$ denotes the successor

of R in $\mathcal{H}$. The *complement path* $\overline{\mathcal{H}}$ is the Hamiltonian circuit obtained by taking the set-complement of every point in $\mathcal{H}$.

Lemma 4.5 *There exists a cycle $\mathcal{C}$ with the properties above.*

Proof: Starting from a point $(0,0,1,0,R,\overline{R})$, we will give a sequence of 4 chunks that will end at $(0,0,1,0,s(s(R)),\overline{s(s(R))})$. Since $\mathcal{H}$ is a Hamiltonian circuit, we get a cycle. The reason we keep R and $\overline{R}$ is that from $(\cdots, R, \overline{R})$, we can perform a single upward and then downward step to reach $(\cdots, s(R), \overline{s(R)})$. This is because moving from R to $s(R)$ is upward iff moving from $\overline{R}$ to $\overline{s(R)}$ is downward. We can assume that the moves to both $s(R)$ and $s(s(R))$ are upward. Whenever either of these is not the case, we can just reverse the roles of R (or $s(R)$) and $\overline{R}$ (or $\overline{s(R)}$).

We describe the sequence of chunks. In the arrows below, the labels above them represents the coordinate being changed. The numbers $1,2,3,4$ represent the first four coordinates. If the label has a set, then that set is being changed by moving along (appropriately) either $\mathcal{H}$ or $\overline{\mathcal{H}}$. These labels help verify the matching property. The first and third chunks only have upward steps, and the remaining have only downward steps. For convenience, $S = s(R)$ and $T = s(S)$.

1. $(0,0,1,0,R,\overline{R}) \stackrel{1}{\to} (1,0,1,0,R,\overline{R}) \stackrel{2}{\to} (1,1,1,0,R,\overline{R}) \stackrel{R}{\to} (1,1,1,0,S,\overline{R})$.
2. $(1,1,1,0,S,\overline{R}) \stackrel{2}{\to} (1,0,1,0,S,\overline{R}) \stackrel{3}{\to} (1,0,0,0,S,\overline{R}) \stackrel{\overline{R}}{\to} (1,0,0,0,S,\overline{S})$.
3. $(1,0,0,0,S,\overline{S}) \stackrel{3}{\to} (1,0,1,0,S,\overline{S}) \stackrel{4}{\to} (1,0,1,1,S,\overline{S}) \stackrel{S}{\to} (1,0,1,1,T,\overline{S})$.
4. $(1,0,1,1,T,\overline{S}) \stackrel{4}{\to} (1,0,1,0,T,\overline{S}) \stackrel{1}{\to} (0,0,1,0,T,\overline{S}) \stackrel{\overline{S}}{\to} (0,0,1,0,T,\overline{T})$.

It is easy to see that no point can occur in two different chunks, because the sets on $\mathcal{H}$ or $\overline{\mathcal{H}}$ are different. So, the cycle is simple. The number of chunks is at least the number of points in the m-dimensional hypercube. The matching property should be clear. □

We now define the function f. Let the directed path consisting of the first two edges of chunk K_i be $\mathcal{P}_i$. Note that $\mathcal{P}_{2i}$ is downward and $\mathcal{P}_{2i+1}$ is upward. We describe the function f and state many properties of $\mathrm{def}(f)$. It will be convenient to have define the following sequences of 4 bits. We set $B_1 = (0,0,1,0)$, $B_2 = (1,0,0,0)$, $C_1 = (1,1,1,0)$, and $C_2 = (1,0,1,1)$. We use A to denote any one of these.

- The function f will be defined on all the endpoints of the $\mathcal{P}_i$'s.
- For $\mathcal{P}_1$, the small endpoint has value v (the exact choice for this is immaterial), and the larger endpoint has value $v+1$. For $\mathcal{P}_{2i+1}$ $(i > 0)$, the small end has value v and the large end has value $v+2$. For $\mathcal{P}_{2i}$ $(\forall i)$, the large end has value $v+2$ and the small end has value v.
- Fix any R. One and only one point of the form $(B_j, R, \overline{R})$ is present in $\mathrm{def}(f)$. Similarly, one and only one of $(C_j, R, \overline{R})$ is present in $\mathrm{def}(f)$. We also have $(B_j, R, \overline{R}) \in \mathrm{def}(f)$ iff $(C_j, R, \overline{R}) \in \mathrm{def}(f)$. No other point is present in levels 1 and 3.
- Fix any R. Suppose $s(R) \supset R$. One and only one of $(B_j, s(R), \overline{R})$ is present in R. Similarly, one and only one of $(C_j, s(R), \overline{R})$ is present in R. We also have $(B_j, s(R), \overline{R}) \in \mathrm{def}(f)$ iff $(C_j, s(R), \overline{R}) \in \mathrm{def}(f)$. No other point is present in levels 2 and 4.

  Suppose $s(R) \subset R$. Then these points are of the form $(A, R, \overline{s(R)})$.

- Pairs of neighbors in def(f) are either level 1-level 2 pairs, or level 3-level 4 pairs. They are always of the following form: $(A, R, \overline{R}) \to (A, s(R), \overline{R})$ (if $R \subset s(R)$) or $(A, R, \overline{R}) \to (A, R, \overline{s(R)})$ (if $R \supset s(R)$).
- For any point of def(f), there is at most one neighbor present in def(f). Hence, any square of $\mathcal{B}$ contains at most 2 points of def(f).
- Consider some point $(B_j, R, \overline{R})$ in level 1. The only point in level 3 at a Hamming distance 2 from this point is $(C_j, R, \overline{R})$. A similar statement holds for points in level 2.

Claim 4.6 *The function f is not submodular-extendable.*

Proof: By Lemma 4.2, it suffices to show a path certificate. As the astute reader might have guessed, all the $\mathcal{P}_i$'s form such a set. A matching exists because of the fourth property of the cycle $\mathcal{C}$. The value of P_1 is 1. The value of any other P_{2i+1} is 2. Every P_{2i} has value -2. Since the total number of chunks is even, the value of this set of paths is -1. □

We will now show that $f|_S$ for any $S \subset \text{def}(f)$ is extendable, by proving that any path certificate for f must essentially be the P_i's.

Claim 4.7 *Suppose f contains a set of matched paths $\mathbf{P}$ with no singletons. This $\mathbf{P}$ must be the set of all $\mathcal{P}_i$'s.*

Proof: Consider a point X in $\mathbf{P}$ that lies in the lowest level (the number of 1s in the representation of the point is minimized). We argue that this point only has upward edges incident to it. If there is a downward edge e incident to it, then $\mathbf{P}$ must contain an upward edge e' that is matched to e. Therefore, $e' \prec e$ and the lower end of e' must lie in a lower level than S. This contradicts the choice of S. Hence, X only has upward edges incident to it. This means that it can never be in the interior of a path, and must be a terminal. Therefore, $X \in \text{def}(f)$. Similarly, points in $\mathbf{P}$ that lie in the highest level only have downward edges incident to them, and are also in def(f).

The points of def(f) lie in levels $m+1, m+2, m+3, m+4$, called the $1, 2, 3, 4$ levels. Edges between the 1 and 2 levels are called *low edges*, those between the 2 and 3 levels are *middle edges*, and those between the 3 and 4 levels are *high edges*. All edges of $\mathbf{P}$ fall into one of these three sets. Low edges are always upward and high edges are always downward. Middle edges are matched to either low or high edges. Therefore, the number of middle edges is exactly the same as the total number of low and high edges. Since $\mathbf{P}$ contains no singletons, every path must contain at least one middle edge. The total number of low and high edges in a path is at most 1. This implies that *every* path in $\mathbf{P}$ has exactly two edges and has one of the two forms: an upward low and middle edge, or a downward top and bottom edge. The former paths go from level 1 to level 3 and the latter from level 4 to level 2. We must have at least one path of each type to get both upward and downward edges. Therefore there is some level 1 point of def(f) in $\mathbf{P}$.

Consider some point $X = (0, 0, 1, 0, R, \overline{R})$ at level 1 that is a terminal in $\mathbf{P}$. Let path $\mathcal{Q} \in \mathbf{P}$ start from here. Note that this is the endpoint for some $\mathcal{P}_i$, which is $(0, 0, 1, 0, R, \overline{R}) \to (1, 0, 1, 0, R, \overline{R}) \to (1, 1, 1, 0, R, \overline{R})$. The certificate $\mathbf{P}$ has an upward path of length 2 from X. The properties of def(f) tells us that the other end of $\mathcal{Q}$ can only be $(1, 1, 1, 0, R, \overline{R})$. It does not immediately follow that $\mathcal{Q}$ is $\mathcal{P}_i$, since there are two different paths between these points (the endpoints differ in coordinates 1 and 2). But observe that the second edge of $\mathcal{Q}$ must be matched by an downward edge between levels 4 and 3. This edge has an endpoint in level 4 that must be a neighbor of $(1, 1, 1, 0, R, \overline{R})$. By the properties of def(f), this point must be $(1, 1, 1, 0, s(R), \overline{R})$ (assuming $s(R) \supset R$). All downward paths of length 2 from this point end at $(1, 0, 0, 0, s(R), \overline{R})$. The path changes in coordinates 2 and

3. Since the second edge of $\mathcal{Q}$ is matched to the first edge of this path, both of these edges must be along coordinate 2. Hence, $\mathcal{Q}$ is $\mathcal{P}_i$, and $\mathcal{P}_{i+1(\mathrm{mod}M)}$ also lies in $\mathbf{P}$. Repeating the argument, we get that all $\mathcal{P}_i$'s lie in $\mathbf{P}$. This completes the proof. □

Proof: (Theorem 1.7) By Claim 4.6, the function f is not submodular-extendable. For some subset $\mathcal{A} \subset \mathrm{def}(f)$, suppose $f|_{\mathcal{A}}$ is not submodular-extendable. Since $\mathrm{def}(f)$ contains no squares, by Claim 4.3, there is a path certificate $\mathbf{P}$ in $\mathrm{def}(f|_{\mathcal{A}})$ that contains no singletons. Note that $\mathbf{P}$ is also a path certificate for f. By Claim 4.7, $\mathbf{P}$ contains all $\mathcal{P}_i$s. But that means that $\mathbf{P}$ contains all points in $\mathrm{def}(f)$. Contradiction. □

5 From monotonicity to submodularity

In this section, we show a simple reduction from testing monotonicity to testing submodularity.

Lemma 5.1 *Given $f : \{0,1\}^n \to \mathbb{R}^+\backslash\{0\}$, there exists a function $g : \{0,1\}^{n+1} \to \mathbb{R}$ with the following properties:*

- *If f is monotonically non-increasing, then g is submodular.*
- *If f is ϵ-far from monotonically non-increasing, then g is $\epsilon/2$-far from submodular.*
- *The value $g(x)$ can be computed by looking at 2 values of f.*

Proof: We will use small letters x, y to denote points in $\{0,1\}^n$. Points in $\{0,1\}^{n+1}$ will be denoted by $(0,x)$ or $(1,x)$, where the first bit denotes the absence or presence of the new element. We use $\mathbf{e}_*$ to denote the unit vector corresponding to the new element, and $\mathbf{e}_i$ ($i \in [n]$) to denote the other unit vectors. For convenience, monotone will mean monotonically non-increasing. Define $h(x) = f(\emptyset)\|x\|_1(n - \|x\|_1)$. We define g by the following: $g(0,x) = h(x)$, and $g(1,x) = f(x) + h(x)$. So any value of g can be computed by looking at 2 values of f.

We first show that h is submodular. Consider x and i such that $x_i = 0$. Let $\|x\|_1 = r$.

$$h(x + \mathbf{e}_i) - h(x) = M[(r+1)(n-r-1) - r(n-r)] = M[n - 2r]$$

This is a decreasing function of r, and hence h is submodular. Furthermore, note that for i, j such that $x_i, x_j = 0$, $h(x + \mathbf{e}_i) + h(x + \mathbf{e}_j) - h(x + \mathbf{e}_i + \mathbf{e}_j) - h(x) = 2M$.

Assume that f is monotone. Then, for any x, $f(x) \leq f(\emptyset) = M$. Since $f(x + \mathbf{e}_i) + f(x + \mathbf{e}_j) - f(x + \mathbf{e}_i + \mathbf{e}_j) - f(x) \geq -2M$, the function $f + h$ is also submodular.

Suppose g is not submodular. Then there exists a violated square in g. Assume all points in this square have the same value for the new coordinate. This square is contained in a copy of $\{0,1\}^n$ where the function is equal to h or $f + h$. But this would imply that either h or $f + h$ is non-submodular. So, this square must involve points differing in the new coordinate. Then we have the following:

$$0 < g(0,x) + g(1, x + \mathbf{e}_i) - g(0, x + \mathbf{e}_i) - g(1,x) = f(x + \mathbf{e}_i) - f(x).$$

This violates the non-increasing property of f. Hence, we conclude that g is submodular.

Now, suppose that f is ϵ-far from monotone. Furthermore, suppose we can modify $\epsilon 2^n$ values of g to get a submodular function g'. Consider the function $f'(x) = g'(1,x) - g'(x)$. Since g' is submodular, f' must be monotone. Since g' differs from g in at most $\epsilon 2^n$ values, the monotone function f' differs from f in at most $\epsilon 2^n$ values. This is a contradiction. So, g must be $\epsilon/2$-far from submodular. □

A recent result [BBM11] gives a linear lower bound for testing monotonicity, which leads to a lower bound for testing submodularity (the following is also stated as Corollary IV.I in their paper). Their monotonicity lower bound even extends to the regime when the range is $[\sqrt{n}]$, so we can apply Lemma 5.1 for functions on the positive real numbers.

Corollary 5.2 *Any tester (even adaptive two-sided) for submodularity requires $\Omega(n)$ queries.*

6 Future work

All of this work is centered on the following question: what makes a function submodular? Of course, it is "just" monotonicity of marginal values, but this does not capture the full structure of submodular functions. We want to understand how different sets of values in a submodular function interact and influence each other. The problem of property testing submodularity appears to be an appealing way of studying this question. Our constructions show that functions far from submodular could have marginal values that are much closer to being monotone.

The problem of completing partial functions comes up when we try to understand how to convert a non-submodular function into a submodular one (this arises in some methods for analyzing a property tester). Again, our constructions yield insight into how seemingly unconnected parts of a submodular function must be related.

The authors believe there is a lot of scope for further research directions. There are many interesting questions to be answered, and we have barely seen the tip of the iceberg. We state some questions here.

1. *Relation between violated squares and distance to submodularity:* For a function f ϵ-far from submodular, what is the minimum (as a function of ϵ and n) density of violated squares it can have? Can we prove that this minimum density is at least poly(ϵ/n)?

2. *Efficient testers for submodularity:* Does there exist a tester for submodularity with running time poly(n/ϵ) or maybe poly(n) for constant ϵ? Perhaps we can find an efficient *adaptive* tester, or a tester that searches for obstructions other than violated squares.

3. *Testing rank functions:* A matroid gives rise to a *rank function*, which is always submodular. A function is a rank function iff it is a submodular function with marginal values 0 or 1. Can we test whether an input function f is a rank function? Note that even though these are a special case of submodular functions, it is not clear whether this is easier (or harder). This is because the *distance to a rank function* might be significantly different from the distance to submodularity.

4. *Testing matroid independence oracles:* Any matroid can be represented as a collection of independent sets. So the matroid can be thought of as a function $f : \{0,1\}^n \longrightarrow \{0,1\}$, where $f(S)$ is 1 iff S is an independent set. Can we efficiently test whether an input boolean function represents a matroid? This seems like a rather fundamental question about matroids.

Acknowledgement. We thank Deeparnab Chakrabarty for useful discussions. Indeed, the main question whether submodularity is testable came up during discussions with him.

References

[BBM11] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. In *Proceedings of the 26th Conference on Computational Complexity (CCC)*, pages 210–220, 2011.

[BCGSM12] J. Briët, S. Chakraborty, D. García-Soriano, and A. Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, (32):35–53, 2012. Conference version in RANDOM 2010.

[BDKR05] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1):132–150, 2005.

[BFRV11] A. Bhattacharyya, E. Fischer, R. Rubinfeld, and P. Valiant. Testing monotonicity of distributions over general partial orders. In *Proceedings of Innovations in Computer Science (ICS)*, pages 239–252, 2011.

[BH11] M.-F. Balcan and N. Harvey. Learning submodular functions. In *Proceedings of the 43rd Annual Symposium on Theory of Computing (STOC)*, pages 793–802, 2011.

[CH12] D. Chakrabarty and Z. Huang. Testing coverage functions. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 170–181, 2012.

[CKKL12] M. Cheraghchi, A. Klivans, P. Kothari, and H.K. Lee. Submodular functions are noise stable. In *Proceedings of the 25th Annual Symposium on Discrete Algorithms (SODA)*, 2012.

[CS12] D. Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over the hypercube. Technical Report TR12-030, ECCC, 2012.

[DGL+99] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonicity. *Proceedings of the 3rd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 97–108, 1999.

[Edm70] J. Edmonds. Matroids, submodular functions and certain polyhedra. *Combinatorial Structures and Their Applications*, pages 69–87, 1970.

[Fis01] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of EATCS*, 75:97–126, 2001.

[Fis04] E. Fischer. On the strength of comparisons in property testing. *Information and Computation*, 189(1):107–116, 2004.

[FLN+02] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the 34th Annual Symposium on Theory of Computing (STOC)*, pages 474–483, 2002.

[FNW78] M.L. Fisher, G.L. Nemhauser, and L.A. Wolsey. An analysis of approximations for maximizing submodular set functions ii. *Mathematical Programming Study*, 8:73–87, 1978.

[FR] S. Fattal and D. Ron. Approximating the distance to monotonicity in high dimensions. In *http://www.eng.tau.ac.il/ danar/Public-pdf/app-mon-long.pdf*.

[Fra97] A. Frank. Matroids and submodular functions. *Annotated Biblographies in Combinatorial Optimization*, pages 65–80, 1997.

[GGL+00] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samordinsky. Testing monotonicity. *Combinatorica*, 20:301–337, 2000. Conference Version in FOCS 1998.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. Conference version in FOCS 1996.

[GHIM09] M. Goemans, N. Harvey, S. Iwata, and V. Mirrokni. Approximating submodular functions everywhere. In *Proceedings of 22th Annual Symposium on Discrete Algorithms (SODA)*, pages 535–544, 2009.

[Gil58] E. N. Gilbert. Gray codes and paths on the n-cube. *Bell System Technical Journal*, 37:815–826, 1958.

[Gol98] O. Goldreich. Combinatorial property testing - a survey. *Randomization Methods in Algorithm Design*, pages 45–60, 1998.

[IFF01] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. *Journal of the ACM*, 48:4:761–777, 2001. Conference version in STOC 2000.

[Lov83] L. Lovász. Submodular functions and convexity. *Mathematical Programmming: The State of the Art*, pages 235–257, 1983.

[NWF78] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions i. *Mathematical Programming*, 14:265–294, 1978.

[PRR03] M. Parnas, D. Ron, and R. Rubinfeld. On testing convexity and submodularity. *SIAM Journal on Computing*, 32(5):1158–1184, 2003. Conference version in RANDOM 2002.

[Ron01] D. Ron. Property testing. *Handbook on Randomization*, II:597–649, 2001.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25:647–668, 1996.

[Sch00] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80:346–355, 2000.