Electronic Colloquium on Computational Complexity, Report No. 181 (2012)

The Inverse Shapley Value Problem*

Anindya De[†]
University of California, Berkeley

Ilias Diakonikolas[‡]
University of Edinburgh

Rocco A. Servedio[§]
Columbia University

December 20, 2012

Abstract

For f a weighted voting scheme used by n voters to choose between two candidates, the n *Shapley-Shubik Indices* (or *Shapley values*) of f provide a measure of how much control each voter can exert over the overall outcome of the vote. Shapley-Shubik indices were introduced by Lloyd Shapley and Martin Shubik in 1954 [SS54] and are widely studied in social choice theory as a measure of the "influence" of voters. The *Inverse Shapley Value Problem* is the problem of designing a weighted voting scheme which (approximately) achieves a desired input vector of values for the Shapley-Shubik indices. Despite much interest in this problem no provably correct and efficient algorithm was known prior to our work.

We give the first efficient algorithm with provable performance guarantees for the Inverse Shapley Value Problem. For any constant $\epsilon > 0$ our algorithm runs in fixed poly(n) time (the degree of the polynomial is independent of ϵ) and has the following performance guarantee: given as input a vector of desired Shapley values, if any "reasonable" weighted voting scheme (roughly, one in which the threshold is not too skewed) approximately matches the desired vector of values to within some small error, then our algorithm explicitly outputs a weighted voting scheme that achieves this vector of Shapley values to within error ϵ. If there is a "reasonable" voting scheme in which all voting weights are integers at most poly(n) that approximately achieves the desired Shapley values, then our algorithm runs in time poly(n) and outputs a weighted voting scheme that achieves the target vector of Shapley values to within error $\epsilon = n^{-1/8}$.

1 Introduction

In this paper we consider the common scenario in which each of n voters must cast a binary vote for or against some proposal. What is the best way to design such a voting scheme? Throughout the paper we consider only *weighted voting schemes,* in which the proposal passes if a weighted sum of yes-votes exceeds a predetermined threshold. Weighted voting schemes are predominant in voting theory and have been extensively studied for many years, see [EGGW07, ZFBE08] and references therein. In computer science language, we are dealing with *linear threshold functions* (henceforth abbreviated as *LTFs*) over n Boolean variables.

If it is desired that each of the n voters should have the same "amount of power" over the outcome, then a simple majority vote is the obvious solution. However, in many scenarios it may

*An extended abstract of this work appeared in the *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP 2012).*

[†] anindya@cs.berkeley.edu. Research supported by NSF award CCF-0915929, CCF-1017403 and CCF-1118083.

[‡] ilias.d@ed.ac.uk. This work was done while the author was at UC Berkeley supported by a Simons Postdoctoral Fellowship.

[§] rocco@cs.columbia.edu. Supported by NSF grants CNS-0716245, CCF-0915929, and CCF-1115703.

ISSN 1433-8092

be the case that we would like to assign different levels of voting power to the n voters – perhaps they are shareholders who own different amounts of stock in a corporation, or representatives of differently sized populations. In such a setting it is much less obvious how to design the right voting scheme; indeed, it is far from obvious how to correctly quantify the notion of the "amount of power" that a voter has under a given fixed voting scheme. As a simple example, consider an election with three voters who have voting weights 49, 49 and 2, in which a total of 51 votes are required for the proposition to pass. While the disparity between voting weights may at first suggest that the two voters with 49 votes each have most of the "power," any coalition of two voters is sufficient to pass the proposition and any single voter is insufficient, so the voting power of all three voters is in fact equal.

Many different *power indices* (methods of measuring the voting power of individuals under a given voting scheme) have been proposed over the course of decades. These include the Banzhaf index [Ban65], the Deegan-Packel index [DP78], the Holler index [Hol82], and others (see the extensive survey of de Keijzer [dK08]). Perhaps the best known, and certainly the oldest, of these indices is the *Shapley-Shubik index* [SS54], which is also known as the index of *Shapley values* (we shall henceforth refer to it as such). Informally, the Shapley value of a voter i among the n voters is the fraction of all $n!$ orderings of the voters in which she "casts the pivotal vote" (see Definition 1 in Section 2 for a precise definition, and [Rot88] for much more on Shapley values). We shall work with the Shapley values throughout this paper.

Given a particular weighted voting scheme (i.e., an n-variable linear threshold function), standard sampling-based approaches can be used to efficiently obtain highly accurate estimates of the n Shapley values (see also the works of [Lee03, BMR$^+$10]). However, the *inverse* problem is much more challenging: given a vector of n desired values for the Shapley values, how can one design a weighted voting scheme that (approximately) achieves these Shapley values? This problem, which we refer to as the *Inverse Shapley Value Problem*, is quite natural and has received considerable attention; various heuristics and exponential-time algorithms have been proposed [APL07, FWJ08, dKKZ10, Kur11], but prior to our work no provably correct and efficient algorithms were known.

Our Results. We give the first efficient algorithm with provable performance guarantees for the Inverse Shapley Value Problem. Our results apply to "reasonable" voting schemes; roughly, we say that a weighted voting scheme is "reasonable" if fixing a tiny fraction of the voting weight does not already determine the outcome, i.e., if the threshold of the linear threshold function is not too extreme. (See Definition 2 in Section 2 for a precise definition.) This seems to be a plausible property for natural voting schemes. Roughly speaking, we show that if there is any reasonable weighted voting scheme that approximately achieves the desired input vector of Shapley values, then our algorithm finds such a weighted voting scheme. Our algorithm runs in fixed polynomial time in n, the number of voters, for any constant error parameter $\epsilon > 0$. In a bit more detail, our first main theorem, stated informally, is as follows (see Section 6 for Theorem 26 which gives a precise theorem statement):

Main Theorem (arbitrary weights, informal statement). *There is a poly(n)-time algorithm with the following properties: The algorithm is given any constant accuracy parameter $\epsilon > 0$ and any vector of n real values $a(1), \ldots, a(n)$. The algorithm has the following performance guarantee: if there is any monotone increasing reasonable LTF $f(x)$ whose Shapley values are very close to the given values $a(1), \ldots, a(n)$, then with very high probability the algorithm outputs $v \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ such that the linear threshold function $h(x) = \text{sign}(v \cdot x - \theta)$ has Shapley values ϵ-close to those of f.*

We emphasize that the exponent of the poly(n) running time is a fixed constant that is independent of ϵ.

Our second main theorem gives an even stronger guarantee if there is a weighted voting scheme with small weights (at most poly(n)) whose Shapley values are close to the desired values. For this problem we give an algorithm which achieves $1/\text{poly}(n)$ accuracy in poly(n) time. An informal statement of this result is (see Section 6 for Theorem 27 which gives a precise theorem statement):

Main Theorem (bounded weights, informal statement). *There is a poly(n, W)-time algorithm with the following properties: The algorithm is given a weight bound W and any vector of n real values $a(1), \dots, a(n)$. The algorithm has the following performance guarantee: if there is any monotone increasing reasonable LTF $f(x) = \text{sign}(w \cdot x - \theta)$ whose Shapley values are very close to the given values $a(1), \dots, a(n)$ and where each w_i is an integer of magnitude at most W, then with very high probability the algorithm outputs $v \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ such that the linear threshold function $h(x) = \text{sign}(v \cdot x - \theta)$ has Shapley values $n^{-1/8}$-close to those of f.*

Discussion and Our Approach. At a high level, the Inverse Shapley Value Problem that we consider is similar to the "Chow Parameters Problem" that has been the subject of several recent papers [Gol06, OS08, DDFS12]. The Chow parameters are another name for the n Banzhaf indices; the Chow Parameters Problem is to output a linear threshold function which approximately matches a given input vector of Chow parameters. (To align with the terminology of the current paper, the "Chow Parameters Problem" might perhaps better be described as the "Inverse Banzhaf Problem.")

Let us briefly describe the approaches in [OS08] and [DDFS12] at a high level for the purpose of establishing a clear comparison with this paper. Each of the papers [OS08, DDFS12] combines structural results on linear threshold functions with an algorithmic component. The structural results in [OS08] deal with anti-concentration of affine forms $w \cdot x - \theta$ where $x \in \{-1, 1\}^n$ is uniformly distributed over the Boolean hypercube, while the algorithmic ingredient of [OS08] is a rather straightforward brute-force search. In contrast, the key structural results of [DDFS12] are geometric statements about how n-dimensional hyperplanes interact with the Boolean hypercube, which are combined with linear-algebraic (rather than anti-concentration) arguments. The algorithmic ingredient of [DDFS12] is more sophisticated, employing a boosting-based approach inspired by the work of [TTV08, Imp95].

Our approach combines aspects of both the [OS08] and [DDFS12] approaches. Very roughly speaking, we establish new structural results which show that linear threshold functions have good anti-concentration (similar to [OS08]), and use a boosting-based approach derived from [TTV08] as the algorithmic component (similar to [DDFS12]). However, this high-level description glosses over many "Shapley-specific" issues and complications that do not arise in these earlier works; below we describe two of the main challenges that arise, and sketch how we meet them in this paper.

First challenge: establishing anti-concentration with respect to non-standard distributions. The Chow parameters (i.e., Banzhaf indices) have a natural definition in terms of the uniform distribution over the Boolean hypercube $\{-1, 1\}^n$. Being able to use the uniform distribution with its many nice properties (such as complete independence among all coordinates) is very useful in proving the required anti-concentration results that are at the heart of [OS08]. In contrast, it is not *a priori* clear what is (or even whether there exists) the "right" distribution over $\{-1, 1\}^n$ corresponding to the Shapley values. In this paper we derive such a distribution μ over $\{-1, 1\}^n$, but it is much less well-behaved than the uniform distribution (it is supported on a proper subset of $\{-1, 1\}^n$, and it is not even pairwise independent). Nevertheless, we are able to establish anti-concentration results for affine forms $w \cdot x - \theta$ corresponding to linear threshold functions under the distribution μ as required for our results. This is done by showing that any reasonable linear threshold function can be expressed with "nice" weights (see Theorem 3 of Section 2), and establishing anti-concentration for any "nice" weight vector by carefully combining anti-concentration bounds for p-biased distributions across a continuous family of different choices of p (see Section 4 for details).

Second challenge: using anti-concentration to solve the Inverse Shapley problem. The main algorithmic ingredient that we use is a procedure from [TTV08]. Given a vector of values $(\mathbf{E}[f(x)x_i])_{i=1,\dots,n}$ (correlations between the unknown linear threshold function f and the individual input variables), it efficiently constructs a bounded function $g : \{-1,1\}^n \to [-1,1]$ which closely matches these correlations, i.e., $\mathbf{E}[f(x)x_i] \approx \mathbf{E}[g(x)x_i]$ for all i. Such a procedure is very useful for the Chow parameters problem, because the Chow parameters correspond precisely to the values $\mathbf{E}[f(x)x_i]$ – i.e., the degree-1 Fourier coefficients of f – with respect to the uniform distribution. (This correspondence is at the heart of Chow's original proof [Cho61] showing that the exact values of the Chow parameters suffice to information-theoretically specify any linear threshold function; anti-concentration is used in [OS08] to extend Chow's original arguments about degree-1 Fourier coefficients to the setting of approximate reconstruction.)

For the inverse Shapley problem, there is no obvious correspondence between the correlations of individual input variables and the Shapley values. Moreover, without a notion of "degree-1 Fourier coefficients" for the Shapley setting, it is not clear why anti-concentration statements with respect to μ should be useful for approximate reconstruction. We deal with both these issues by developing a notion of the *degree-*1 *Fourier coefficients of* f *with respect to distribution* μ and relating these coefficients to the Shapley values [1]. (We actually require two related notions: one is the "coordinate correlation coefficient" $\mathbf{E}_{x\sim\mu}[f(x)x_i]$, which is necessary for the algorithmic [TTV08] ingredient, and one is the "Fourier coefficient" $\hat{f}(i) = \mathbf{E}_{x\sim\mu}[f(x)L_i]$, which is necessary for Lemma 15, see below.) We define both notions and establish the necessary relations between them in Section 3.

Armed with the notion of the degree-1 Fourier coefficients under distribution μ, we prove a key result (Lemma 15) saying that if the LTF f is anti-concentrated under distribution μ, then any bounded function g which closely matches the degree-1 Fourier coefficients of f must be close to f in ℓ_1 distance with respect to μ. (This is why anti-concentration with respect to μ is useful for us.) From this point, exploiting properties of the [TTV08] algorithm, we can pass from g to an LTF whose Shapley values closely match those of f.

Organization. Useful preliminaries are given in Section 2, including the crucial fact (Theorem 3) that all "reasonable" linear threshold functions have weight representations with "nice" weights. In Section 3 we define the distribution μ and the notions of Fourier coefficients and "coordinate correlation coefficients," and the relations between them, that we will need. At the end of that section we prove a crucial lemma, Lemma 15, which says that anti-concentration of affine forms and closeness in Fourier coefficients together suffice to establish closeness in ℓ_1 distance. Section 4 proves that "nice" affine forms have the required anti-concentration, and Section 5 describes the algorithmic tool from [TTV08] that lets us establish closeness of coordinate correlation coefficients. Section 6 puts the pieces together to prove our main theorems. Finally, in Section 7 we conclude the paper and present a few open problems.

[1] We note that Owen [Owe72] has given a characterization of the Shapley values as a weighted average of p-biased influences (see also [KS06]). However, this is not as useful for us as our characterization in terms of "μ-distribution" Fourier coefficients, because we need to ultimately relate the Shapley values to anti-concentration with respect to μ.

2 Preliminaries

Notation and terminology. For $n \in \mathbb{Z}_+$, we denote by $[n] \overset{\text{def}}{=} \{1, 2, \ldots, n\}$. For $i, j \in \mathbb{Z}_+$, $i \leq j$, we denote $[i, j] \overset{\text{def}}{=} \{i, i+1, \ldots, j\}$.

Given a vector $w = (w_1, \ldots, w_n) \in \mathbb{R}^n$ we write $\|w\|_1$ to denote $\sum_{i=1}^n |w_i|$. A *linear threshold function*, or LTF, is a function $f : \{-1,1\}^n \to \{-1,1\}$ which is such that $f(x) = \text{sign}(w \cdot x - \theta)$ for some $w \in \mathbb{R}^n, \theta \in \mathbb{R}$.

Our arguments will also use a variant of linear threshold functions which we call *linear bounded functions* (LBFs). The projection function $P_1 : \mathbb{R} \to [-1, 1]$ is defined by $P_1(t) = t$ for $|t| \leq 1$ and $P_1(t) = \text{sign}(t)$ otherwise. An LBF $g : \{-1,1\}^n \to [-1,1]$ is a function $g(x) = P_1(w \cdot x - \theta)$.

Shapley values. Here and throughout the paper we write $\mathbb{S}_n$ to denote the symmetric group of all $n!$ permutations over $[n]$. Given a permutation $\pi \in \mathbb{S}_n$ and an index $i \in [n]$, we write $x(\pi, i)$ to denote the string in $\{-1,1\}^n$ that has a 1 in coordinate j if and only if $\pi(j) < \pi(i)$, and we write $x^+(\pi, i)$ to denote the string obtained from $x(\pi, i)$ by flipping coordinate i from -1 to 1. With this notation in place we can define the generalized Shapley indices of a Boolean function as follows:

Definition 1. (Generalized Shapley values) *Given $f : \{-1,1\}^n \to \{-1,1\}$, the* i-th generalized Shapley value of f *is the value*

$$\tilde{f}(i) \overset{def}{=} \mathbf{E}_{\pi \sim_R \mathbb{S}_n}[f(x^+(\pi, i)) - f(x(\pi, i))] \tag{1}$$

(where "$\pi \sim_R \mathbb{S}_n$" means that π is selected uniformly at random from $\mathbb{S}_n$).

A function $f : \{-1,1\}^n \to \{-1,1\}$ is said to be *monotone increasing* if for all $i \in [n]$, whenever two input strings $x, y \in \{-1,1\}^n$ differ precisely in coordinate i and have $x_i = -1$, $y_i = 1$, it is the case that $f(x) \leq f(y)$. It is easy to check that for monotone functions our definition of generalized Shapley values agrees with the usual notion of Shapley values (which are typically defined only for monotone functions) up to a multiplicative factor of 2; in the rest of the paper we omit "generalized" and refer to these values simply as the Shapley values of f.

We will use the following notion of the "distance" between the vectors of Shapley values for two functions $f, g : \{-1,1\}^n \to [-1,1]$:

$$d_{\text{Shapley}}(f, g) \overset{\text{def}}{=} \sqrt{\sum_{i=1}^n (\tilde{f}(i) - \tilde{g}(i))^2},$$

i.e., the Shapley distance $d_{\text{Shapley}}(f, g)$ is simply the Euclidean distance between the two n-dimensional vectors of Shapley values. Given a vector $a = (a(1), \ldots, a(n)) \in \mathbb{R}^n$ we will also use $d_{\text{Shapley}}(a, f)$ to denote $\sqrt{\sum_{i=1}^n (\tilde{f}(i) - a(i))^2}$.

The linear threshold functions that we consider. Our algorithmic results hold for linear threshold functions which are not too "extreme" (in the sense of having a very skewed threshold). We will use the following definition:

Definition 2. (η-reasonable LTF) *Let $f : \{-1,1\}^n \to \{-1,1\}$, $f(x) = \text{sign}(w \cdot x - \theta)$ be an LTF. For $0 < \eta < 1$ we say that f is* η-reasonable *if $\theta \in [-(1-\eta)\|w\|_1, (1-\eta)\|w\|_1]$.*

All our results will deal with η-reasonable LTFs; throughout the paper η should be thought of as a small fixed absolute constant (such as 1/1000). LTFs that are not η-reasonable do not seem to correspond to very interesting voting schemes since typically they will be very close to constant functions. (For example, even at $\eta = 0.99$, if the LTF $f(x) = \text{sign}(x_1 + \cdots + x_n - \theta)$ has a threshold $\theta > 0$ which makes it not an η-reasonable LTF, then f agrees with the constant function -1 on all but a $2^{-\Omega(n)}$ fraction of inputs in $\{-1,1\}^n$.)

Turning from the threshold to the weights, some of the proofs in our paper will require us to work with LTFs that have "nice" weights in a certain technical sense. Prior work [Ser07, OS11] has shown that for any LTF, there is a weight vector realizing that LTF that has essentially the properties we need; however, since the exact technical condition that we require is not guaranteed by any of the previous works, we give a full proof that any LTF has a representation of the desired form. The following theorem is proved in Appendix A:

Theorem 3. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be an η-reasonable LTF and $k \in [2,n]$. There exists a representation of f as $f(x) = \text{sign}(v_0 + \sum_{i=1}^n v_i x_i)$ such that (after reordering coordinates so that condition (i) below holds) we have: (i) $|v_i| \geq |v_{i+1}|$, $i \in [n-1]$; (ii) $|v_0| \leq (1-\eta)\sum_{i=1}^n |v_i|$; and (iii) for all $i \in [0, k-1]$ we have $|v_i| \leq (2/\eta) \cdot \sqrt{n} \cdot k^{\frac{k}{2}} \cdot \sigma_k$, where $\sigma_k \overset{def}{=} \sqrt{\sum_{j \geq k} v_j^2}$.*

Tools from probability. We will use the following standard tail bound:

Theorem 4. (Chernoff Bounds) *Let X be a random variable taking values in $[-a,a]$ and let $X_1, \ldots, X_t$ be i.i.d. samples drawn from X. Let $\overline{X} = \sum_{i=1}^t X_i/t$. Then for any $\gamma > 0$, we have*

$$\mathbf{Pr}\left[\left|\overline{X} - \mathbf{E}[X]\right| \geq \gamma\right] \leq 2\exp(-\gamma^2 t/(2a^2)).$$

We will also use the Littlewood-Offord inequality for p-biased distributions over $\{-1,1\}^n$. One way to prove this is by using the LYM inequality (which can be found e.g. as Theorem 8.6 of [Juk01]); for an explicit reference and proof of the following statement see e.g. [AGKW09].

Theorem 5. *Fix $\delta \in (0,1)$ and let D_δ denote the δ-*biased distribution over *$\{-1,1\}^n$ (under which each coordinate is set to 1 independently with probability δ.) Fix $w \in \mathbb{R}^n$ and define $S = \{i : |w_i| \geq \epsilon\}$. If $|S| \geq K$, then for all $\theta \in \mathbb{R}$ we have $\mathbf{Pr}_{x \sim D_\delta}[|w \cdot x - \theta| < \epsilon] \leq \frac{1}{\sqrt{K\delta(1-\delta)}}$.*

Basic Facts about function spaces. We will use the following basic facts:

Fact 6. *The $n+1$ functions $1, x_1, \ldots, x_n$ are linearly independent and form a basis for the subspace $V = \{f : \{-1,1\}^n \to \mathbb{R}$ and f is linear $\}$.*

Fact 7. *Fix any $\Omega \subseteq \{-1,1\}^n$ and let μ be a probability distribution over Ω such that $\mu(x) > 0$ for all $x \in \Omega$. We define $\langle f, g\rangle_\mu \overset{def}{=} \mathbf{E}_{\omega \sim \mu}[f(\omega)g(\omega)]$ for $f, g : \Omega \to \mathbb{R}$. Suppose that $f_1, \ldots, f_m : \Omega \to \mathbb{R}$ is an orthonormal set of functions, i.e., $\langle f_i, f_j\rangle_\mu = \delta_{ij}$ for all $i, j \in [m]$. Then we have $\langle f, f\rangle_\mu^2 \geq \sum_{i=1}^m \langle f, f_i\rangle_\mu^2$. As a corollary, if $f, h : \Omega \to \{-1,1\}$ then we have $\sqrt{\sum_{i=1}^m \langle f - h, f_i\rangle_\mu^2} \leq 2\sqrt{\mathbf{Pr}_{x\sim\mu}[f(x) \neq h(x)]}$.*

3 Analytic Reformulation of Shapley values

The definition of Shapley values given in Definition 1 is somewhat cumbersome to work with. In this section we derive alternate characterizations of Shapley values in terms of "Fourier coefficients" and "coordinate correlation coefficients" and establish various technical results relating Shapley values and these coefficients; these technical results will be crucially used in the proof of our main theorems.

There is a particular distribution μ that plays a central role in our reformulations. We start by defining this distribution μ and introducing some relevant notation, and then give our results.

The distribution μ. Let us define $\Lambda(n) \stackrel{\text{def}}{=} \sum_{0<k<n} \frac{1}{k} + \frac{1}{n-k}$; clearly we have $\Lambda(n) = \Theta(\log n)$, and more precisely we have $\Lambda(n) \le 2\log n$. We also define $Q(n,k)$ as $Q(n,k) \stackrel{\text{def}}{=} \frac{1}{k} + \frac{1}{n-k}$ for $0 < k < n$, so we have $\Lambda(n) = \sum_{k=1}^{n-1} Q(n,k)$.

For $x \in \{-1,1\}^n$ we write $\mathrm{wt}(x)$ to denote the number of 1's in x. We define the set B_n to be $B_n \stackrel{\text{def}}{=} \{x \in \{-1,1\}^n : 0 < \mathrm{wt}(x) < n\}$, i.e., $B_n = \{-1,1\}^n \setminus \{\mathbf{1}, \mathbf{-1}\}$.

The distribution μ is supported on B_n and is defined as follows: to make a draw from μ, sample $k \in \{1,\dots,n-1\}$ with probability $Q(n,k)/\Lambda(n)$. Choose $x \in \{-1,1\}^n$ uniformly at random from the k-th "weight level" of $\{-1,1\}^n$, i.e., from $\{-1,1\}^n_{=k} \stackrel{\text{def}}{=} \{x \in \{-1,1\}^n : \mathrm{wt}(x) = k\}$.

Useful notation. For $i = 0,\dots,n$ we define the "coordinate correlation coefficients" of a function $f : \{-1,1\}^n \to \mathbb{R}$ (with respect to μ) as:

$$f^*(i) \stackrel{\text{def}}{=} \mathbf{E}_{x\sim\mu}[f(x) \cdot x_i] \tag{2}$$

(here and throughout the paper x_0 denotes the constant 1).

Later in this section we will define an orthonormal set of linear functions $L_0, L_1, \dots, L_n : \{-1,1\}^n \to \mathbb{R}$. We define the "Fourier coefficients" of f (with respect to μ) as:

$$\hat{f}(i) \stackrel{\text{def}}{=} \mathbf{E}_{x\sim\mu}[f(x) \cdot L_i(x)]. \tag{3}$$

An alternative expression for the Shapley values. We start by expressing the Shapley values in terms of the coordinate correlation coefficients:

Lemma 8. *Given $f : \{-1,1\}^n \to [-1,1]$, for each $i = 1,\dots,n$ we have*

$$\tilde{f}(i) = \frac{f(\mathbf{1}) - f(\mathbf{-1})}{n} + \frac{\Lambda(n)}{2} \cdot \left(f^*(i) - \frac{1}{n}\sum_{j=1}^{n} f^*(j) \right),$$

or equivalently,

$$f^*(i) = \frac{2}{\Lambda(n)} \cdot \left(\tilde{f}(i) - \frac{f(\mathbf{1}) - f(\mathbf{-1})}{n} \right) + \frac{1}{n}\sum_{j=1}^{n} f^*(j).$$

Proof. Recall that $\tilde{f}(i)$ can be expressed as follows:

$$\tilde{f}(i) = \mathbf{E}_{\pi\sim_R \mathbb{S}_n}[f(x^+(\pi,i)) - f(x(\pi,i))]. \tag{4}$$

Since the i-th coordinate of $x^+(\pi,i)$ is 1 and the i-th coordinate of $x(\pi,i)$ is -1, we see that $\tilde{f}(i)$ is a weighted sum of $\{f(x)x_i\}_{x\in\{-1,1\}^n}$. We now compute the weights associated with any such $x \in \{-1,1\}^n$.

- Let x be a string that has $\mathrm{wt}(x)$ coordinates that are 1 and has $x_i = 1$. Then the total number of permutations $\pi \in \mathbb{S}_n$ such that $x^+(\pi,i) = x$ is $(\mathrm{wt}(x)-1)!(n-\mathrm{wt}(x))!$. Consequently the weight associated with $f(x)x_i$ for such an x is $(\mathrm{wt}(x)-1)! \cdot (n-\mathrm{wt}(x))!/n!$.

- Now let x be a string that has $\mathrm{wt}(x)$ coordinates that are 1 and has $x_i = -1$. Then the total number of permutations $\pi \in \mathbb{S}_n$ such that $x(\pi,i) = x$ is $\mathrm{wt}(x)!(n-\mathrm{wt}(x)-1)!$. Consequently the weight associated with $f(x)x_i$ for such an x is $\mathrm{wt}(x)! \cdot (n-\mathrm{wt}(x)-1)!/n!$.

Thus we may rewrite Equation (4) as

$$\begin{aligned}\tilde{f}(i) \quad = \quad & \sum_{x:\{-1,1\}^n:x_i=1} \frac{(\mathrm{wt}(x)-1)!(n-\mathrm{wt}(x))!}{n!} f(x)\cdot x_i + \\ & \sum_{x:\{-1,1\}^n:x_i=-1} \frac{\mathrm{wt}(x)!(n-\mathrm{wt}(x)-1)!}{n!} f(x)\cdot x_i .\end{aligned}$$

Let us now define $\nu(f) \stackrel{\mathrm{def}}{=} (f(\mathbf{1}) - f(-\mathbf{1}))/n$. Using the fact that $x_i^2 = 1$, it is easy to see that one gets

$$\begin{aligned} 2\tilde{f}(i) &= 2\nu(f) + \\ & \quad 2\left(\sum_{x\in B_n} f(x) \cdot \frac{(\mathrm{wt}(x)-1)!(n-\mathrm{wt}(x)-1)!}{n!} \cdot ((n/2 - \mathrm{wt}(x)) + (nx_i)/2) \right) \\ &= 2\nu(f) + \sum_{x\in B_n} \left(f(x) \cdot \frac{(\mathrm{wt}(x)-1)!(n-\mathrm{wt}(x)-1)!}{(n-1)!} \cdot x_i + \right. \\ & \qquad \left. f(x) \cdot \frac{(\mathrm{wt}(x)-1)!(n-\mathrm{wt}(x)-1)!}{n!} \cdot (n - 2\mathrm{wt}(x)) \right) \\ &= 2\nu(f) + \sum_{x\in B_n} \left(f(x) \cdot \frac{n}{\mathrm{wt}(x)(n-\mathrm{wt}(x))\binom{n}{\mathrm{wt}(x)}} \cdot x_i + \right. \\ & \qquad \left. f(x) \cdot \frac{1}{\mathrm{wt}(x)(n-\mathrm{wt}(x))\binom{n}{\mathrm{wt}(x)}} \cdot (n - 2\mathrm{wt}(x)) \right) . \end{aligned} \tag{5}$$

We next observe that $n - 2\mathrm{wt}(x) = -(\sum_{j\in[n]} x_j)$. Next, let us define $P(n,k)$ (for $k \in [1, n-1]$) as follows :

$$P(n,k) \stackrel{\mathrm{def}}{=} \frac{Q(n,k)}{\binom{n}{k}} = \frac{\frac{1}{k} + \frac{1}{n-k}}{\binom{n}{k}}.$$

So we may rewrite Equation (5) in terms of $P(n, \mathrm{wt}(x))$ as

$$2\tilde{f}(i) = 2\nu(f) + \sum_{x\in B_n} [f(x)\cdot x_i \cdot P(n,\mathrm{wt}(x))] - \sum_{x\in B_n} \left[f(x) \cdot P(n,\mathrm{wt}(x)) \cdot (\sum_{i=1}^n x_i)/n \right].$$

We have

$$\sum_{x\in B_n} P(n,\mathrm{wt}(x)) = \sum_{k=1}^{n-1} \sum_{x\in\{-1,1\}^n_{=k}} P(n,\mathrm{wt}(x)) = \sum_{k=1}^{n-1} \binom{n}{k} \cdot P(n,k) = \sum_{k=1}^{n-1} Q(n,k) = \Lambda(n),$$

and consequently we get

$$2\tilde{f}(i) = 2\nu(f) + \Lambda(n) \cdot \left(\mathop{\mathbf{E}}_{x\sim\mu} [f(x)\cdot x_i] - \mathop{\mathbf{E}}_{x\sim\mu} \left[f(x) \cdot (\sum_{i=1}^n x_i)/n \right] \right),$$

finishing the proof. □

Construction of a Fourier basis for distribution μ. For all $x \in B_n$ we have that $\mu(x) > 0$, and consequently by Fact 6 we know that the functions $1, x_1, \ldots, x_{n+1}$ form a basis for the subspace of linear functions from $B_n \to \mathbb{R}$. By Gram-Schmidt orthogonalization, we can obtain an orthonormal basis $L_0, \dots, L_n$ for this subspace, i.e., a set of linear functions such that $\langle L_i, L_i \rangle_\mu = 1$ for all i and $\langle L_i, L_j \rangle_\mu = 0$ for all $i \neq j$.

We now give explicit expressions for these basis functions. We start by defining $L_0 : B_n \to \mathbb{R}$ as $L_0 : x \mapsto 1$. Next, by symmetry, we can express each L_i as

$$L_i(x) = \alpha(x_1 + \ldots + x_n) + \beta x_i.$$

Using the orthonormality properties it is straightforward to solve for α and β. The following Lemma gives the values of α and β:

Lemma 9. *For the choices*

$$\alpha \overset{def}{=} \frac{1}{n} \cdot \left(\sqrt{\frac{\Lambda(n)}{n\Lambda(n) - 4(n-1)}} - \frac{\sqrt{\Lambda(n)}}{2} \right), \quad \beta \overset{def}{=} \frac{\sqrt{\Lambda(n)}}{2},$$

the set $\{L_i\}_{i=0}^n$ is an orthonormal set of linear functions under the distribution μ.

We note for later reference that $\alpha = -\Theta\left(\frac{\sqrt{\log n}}{n}\right)$ and $\beta = \Theta(\sqrt{\log n})$.

We start with the following proposition which gives an explicit expression for $\mathbf{E}_{x\sim\mu}[x_i x_j]$ when $i \neq j$; we will use it in the proof of Lemma 9.

Proposition 10. *For all $1 \le i < j \le n$ we have $\mathbf{E}_{x\sim\mu}[x_i x_j] = 1 - \frac{4}{\Lambda(n)}$.*

Proof. For brevity let us write $A_k = \{-1,1\}^n_{=k}$, i.e., $A_k = \{x \in \{-1,1\}^n : \mathrm{wt}(x) = k\}$, the k-th "slice" of the hypercube. Since μ is supported on $B_n = \cup_{k=1}^{n-1} A_k$, we have

$$\mathbf{E}_{x\sim\mu}[x_i x_j] = \sum_{0<k<n} \mathop{\mathbf{E}}_{x\sim\mu}[x_i x_j \mid x \in A_k] \cdot \mathbf{Pr}_{x\sim\mu}[x \in A_k].$$

If $k = 1$ or $n-1$, it is clear that

$$\mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k] = 1 - \frac{2}{n} - \frac{2}{n} = 1 - \frac{4}{n},$$

and when $2 \le k \le n-2$, we have

$$\mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k] = \frac{1}{\binom{n}{k}} \cdot \left(2\binom{n-2}{k-2} + 2\binom{n-2}{k} - \binom{n}{k} \right).$$

Recall that $\Lambda(n) = \sum_{0<k<n} \frac{1}{k} + \frac{1}{n-k}$ and $Q(n,k) = \frac{1}{k} + \frac{1}{n-k}$ for $0 < k < n$. This means that we have

$$\mathbf{Pr}_{x\sim\mu}[x \in A_k] = Q(n,k)/\Lambda(n).$$

Thus we may write $\mathbf{E}_{x\sim\mu}[x_i x_j]$ as

$$\begin{aligned} \mathbf{E}_{x\sim\mu}[x_i x_j] \quad = \quad & \sum_{2\le k\le n-2} \frac{Q(n,k)}{\Lambda(n)} \cdot \mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k] + \\ & \sum_{k\in\{1,n-1\}} \frac{Q(n,k)}{\Lambda(n)} \cdot \mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k]. \end{aligned}$$

For the latter sum, we have

$$\sum_{k\in\{1,n-1\}} \frac{Q(n,k)}{\Lambda(n)} \cdot \mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k] = \frac{1}{\Lambda(n)}\left(1-\frac{4}{n}\right)\cdot\frac{2n}{n-1}.$$

For the former, we can write

$$\begin{aligned}
&\sum_{k=2}^{n-2} \frac{Q(n,k)}{\Lambda(n)} \cdot \mathbf{E}_{x\sim\mu}[x_i x_j \mid x \in A_k] \\
= &\sum_{k=2}^{n-2} \frac{1}{\Lambda(n)} \frac{(k-1)!(n-k-1)!}{(n-1)!} \cdot \left(2\binom{n-2}{k-2} + 2\binom{n-2}{k} - \binom{n}{k}\right) \\
= &\sum_{k=2}^{n-2} \frac{1}{\Lambda(n)} \cdot \left(\frac{2(k-1)}{(n-1)(n-k)} + \frac{2(n-k-1)}{(n-1)k} - \frac{n}{k(n-k)}\right) \\
= &\sum_{k=2}^{n-2} \frac{1}{\Lambda(n)} \cdot \left(\frac{2}{n-k} - \frac{2}{n-1} + \frac{2}{k} - \frac{2}{n-1} - \frac{1}{k} - \frac{1}{n-k}\right) \\
= &\sum_{k=2}^{n-2} \frac{1}{\Lambda(n)} \cdot \left(\frac{1}{n-k} + \frac{1}{k} - \frac{4}{n-1}\right).
\end{aligned}$$

Thus, we get that overall $\mathbf{E}_{x\sim\mu}[x_i x_j]$ equals

$$\begin{aligned}
&\frac{1}{\Lambda(n)}\left(1-\frac{4}{n}\right)\cdot\frac{2n}{n-1} + \sum_{k=2}^{n-2} \frac{1}{\Lambda(n)} \cdot \left(\frac{1}{n-k} + \frac{1}{k} - \frac{4}{n-1}\right) \\
= &\frac{1}{\Lambda(n)}\left(2 + \frac{2}{n-1} - \frac{8}{n-1}\right) + \frac{1}{\Lambda(n)}\left(\sum_{k=2}^{n-2} \frac{1}{k} + \frac{1}{n-k}\right) - \frac{4}{\Lambda(n)} + \frac{8}{\Lambda(n)(n-1)} \\
= &\frac{1}{\Lambda(n)}\left(\sum_{k=1}^{n-1} Q(n,k)\right) - \frac{4}{\Lambda(n)} = 1 - \frac{4}{\Lambda(n)},
\end{aligned}$$

as was to be shown. □

Proof of Lemma 9. We begin by observing that

$$\mathbf{E}_{x\sim\mu}[L_i(x)L_0(x)] = \mathbf{E}_{x\sim\mu}[L_i(x)] = \mathbf{E}_{x\sim\mu}[\alpha(x_1 + \ldots + x_n) + \beta x_i] = 0$$

since $\mathbf{E}_{x\sim\mu}[x_i] = 0$. Next, we solve for α and β using the orthonormality conditions on the set $\{L_i\}_{i=1}^n$. As $\mathbf{E}_{x\sim\mu}[L_i(x)L_j(x)] = 0$ and $\mathbf{E}_{x\sim\mu}[L_i(x)L_i(x)] = 1$, we get that $\mathbf{E}_{x\sim\mu}[L_i(x)(L_i(x) - L_j(x))] = 1$. This gives

$$\begin{aligned}
\mathbf{E}_{x\sim\mu}[L_i(x)\cdot(L_i(x) - L_j(x))] &= \mathbf{E}_{x\sim\mu}[L_i(x)\cdot\beta(x_i - x_j)] \\
&= \mathbf{E}_{x\sim\mu}[\beta((\alpha+\beta)x_i + \alpha x_j)\cdot(x_i - x_j)] \\
&= \alpha\beta + \beta^2 - \alpha\beta - \beta^2\mathbf{E}_{x\sim\mu}[x_j x_i] \\
&= \beta^2(1 - \mathbf{E}_{x\sim\mu}[x_i x_j]) = 4\beta^2/\Lambda(n) = 1,
\end{aligned}$$

where the penultimate equation above uses Proposition 10. Thus, we have shown that $\beta = \frac{\sqrt{\Lambda(n)}}{2}$. To solve for α, we note that

$$\sum_{i=1}^{n} L_i(x) = (\alpha n + \beta)(x_1 + \ldots + x_n).$$

However, since the set $\{L_i\}_{i=1}^{n}$ is orthonormal with respect to the distribution μ, we get that

$$\mathbf{E}_{x\sim\mu}[(L_1(x) + \ldots + L_n(x))(L_1(x) + \ldots + L_n(x))] = n$$

and consequently

$$(\alpha n + \beta)^2 \, \mathbf{E}_{x\sim\mu}[(x_1 + \ldots + x_n)(x_1 + \ldots + x_n)] = n$$

Now, using Proposition 10, we get

$$\begin{aligned} \mathbf{E}_{x\sim\mu}[(x_1 + \ldots + x_n)(x_1 + \ldots + x_n)] &= \sum_{i=1}^{n} \mathbf{E}_{x\sim\mu}[x_i^2] + \sum_{i\neq j} \mathbf{E}_{x\sim\mu}[x_i x_j] \\ &= n + n(n-1)\cdot\left(1 - \frac{4}{\Lambda(n)}\right) \end{aligned}$$

Thus, we get that

$$(\alpha n + \beta)^2 \cdot \left(n + n(n-1)\cdot\left(1 - \frac{4}{\Lambda(n)}\right)\right) = n.$$

Simplifying further,

$$(\alpha n + \beta) = \sqrt{\frac{\Lambda(n)}{n\Lambda(n) - 4(n-1)}}$$

and thus

$$\alpha = \frac{1}{n}\cdot\left(\sqrt{\frac{\Lambda(n)}{n\Lambda(n) - 4(n-1)}} - \frac{\sqrt{\Lambda(n)}}{2}\right)$$

as was to be shown. □

Relating the Shapley values to the Fourier coefficients. The next lemma gives a useful expression for $\hat{f}(i)$ in terms of $\tilde{f}(i)$:

Lemma 11. *Let $f : \{-1,1\}^n \to [-1,1]$ be any bounded function. Then for each $i = 1, \ldots, n$ we have*

$$\hat{f}(i) = \frac{2\beta}{\Lambda(n)}\cdot\left(\tilde{f}(i) - \frac{f(\mathbf{1}) - f(-\mathbf{1})}{n}\right) + \frac{1}{n}\cdot\sum_{j=1}^{n}\hat{f}(j).$$

Proof. Lemma 9 gives us that $L_i(x) = \alpha(x_1 + \ldots + x_n) + \beta x_i$, and thus we have

$$\begin{aligned} \hat{f}(i) \equiv \mathbf{E}_{x\sim\mu}[f(x)\cdot L_i(x)] &= \alpha\left(\sum_{j=1}^{n}\mathbf{E}_{x\sim\mu}[f(x)\cdot x_j]\right) + \beta\mathbf{E}_{x\sim\mu}[f(x)\cdot x_i] \\ &= \alpha\sum_{j=1}^{n} f^*(j) + \beta f^*(i). \qquad (6) \end{aligned}$$

Summing this for $i = 1$ to n, we get that

$$\sum_{j=1}^{n} \hat{f}(j) = (\alpha n + \beta) \sum_{j=1}^{n} f^*(j). \tag{7}$$

Plugging this into (6), we get that

$$f^*(i) = \frac{1}{\beta} \cdot \left(\hat{f}(i) - \frac{\alpha}{\alpha n + \beta} \cdot \sum_{j=1}^{n} \hat{f}(j) \right) \tag{8}$$

Now recall that from Lemma 8, we have

$$\begin{aligned} \tilde{f}(i) &= \nu(f) + \frac{\Lambda(n)}{2} \cdot \left(\mathop{\mathbf{E}}_{x \sim \mu} [f(x) \cdot x_i] - \mathop{\mathbf{E}}_{x \sim \mu} \left[f(x) \cdot (\sum_{i=1}^{n} x_i)/n \right] \right) \\ &= \nu(f) + \frac{\Lambda(n)}{2} \cdot \left(f^*(i) - \frac{\sum_{j=1}^{n} f^*(j)}{n} \right) \end{aligned}$$

where $\nu(f) = (f(\mathbf{1}) - f(-\mathbf{1}))/n$. Hence, combining the above with (7) and (8), we get

$$\frac{1}{\beta} \cdot \left(\hat{f}(i) - \frac{\alpha}{\alpha n + \beta} \cdot \sum_{j=1}^{n} \hat{f}(j) \right) = \frac{2}{\Lambda(n)} \cdot (\tilde{f}(i) - \nu(f)) + \frac{1}{n(\alpha n + \beta)} \cdot \sum_{j=1}^{n} \hat{f}(j).$$

From this, it follows that

$$\frac{1}{\beta} \cdot \hat{f}(i) = \frac{2}{\Lambda(n)} \cdot (\tilde{f}(i) - \nu(f)) + \frac{1}{\alpha n + \beta} \cdot \left(\frac{1}{n} + \frac{\alpha}{\beta} \right) \cdot \sum_{j=1}^{n} \hat{f}(j),$$

and hence

$$\hat{f}(i) = \frac{2\beta}{\Lambda(n)} \cdot (\tilde{f}(i) - \nu(f)) + \frac{1}{n} \cdot \sum_{j=1}^{n} \hat{f}(j)$$

as desired. □

Bounding Shapley distance in terms of Fourier distance. Recall that the Shapley distance $d_{\text{Shapley}}(f, g)$ between $f, g : \{-1, 1\}^n \to [-1, 1]$ is defined as $d_{\text{Shapley}}(f, g) \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{n} (\tilde{f}(i) - \tilde{g}(i))^2}$. We define the *Fourier distance* between f and g as $d_{\text{Fourier}}(f, g) \stackrel{\text{def}}{=} \sqrt{\sum_{i=0}^{n} (\hat{f}(i) - \hat{g}(i))^2}$.

Our next lemma shows that if the Fourier distance between f and g is small then so is the Shapley distance.

Lemma 12. *Let $f, g : \{-1, 1\}^n \to [-1, 1]$. Then,*

$$d_{\text{Shapley}}(f, g) \leq \frac{4}{\sqrt{n}} + \frac{\Lambda(n)}{2\beta} \cdot d_{\text{Fourier}}(f, g).$$

Proof. Let $\nu(f) = (f(\mathbf{1}) - f(-\mathbf{1}))/n$ and $\nu(g) = (g(\mathbf{1}) - g(-\mathbf{1}))/n$. From Lemma 11, we have that for all $1 \leq i \leq n$,

$$\frac{\Lambda(n)}{2\beta} \cdot \left(\hat{f}(i) - \frac{\sum_{j=1}^{n} \hat{f}(j)}{n} \right) + \nu(f) = \tilde{f}(i).$$

Using a similar relation for g, we get that for every $1 \le i \le n$,

$$\frac{\Lambda(n)}{2\beta} \cdot \left(\hat{f}(i) - \frac{\sum_{j=1}^n \hat{f}(j)}{n} - \hat{g}(i) + \frac{\sum_{j=1}^n \hat{g}(j)}{n} \right) + \nu(f) - \nu(g) = \tilde{f}(i) - \tilde{g}(i).$$

We next define the following vectors: let $v \in \mathbb{R}^n$ be defined by $v_i = \tilde{f}(i) - \tilde{g}(i)$, $i \in [n]$ (so our goal is to bound $\|v\|_2$). Let $u \in \mathbb{R}^n$ be defined by $u_i = \nu(f) - \nu(g)$, $i \in [n]$. Finally, let $w \in \mathbb{R}^n$ be defined by

$$w_i = \left(\hat{f}(i) - \frac{\sum_{j=1}^n \hat{f}(j)}{n} - \hat{g}(i) + \frac{\sum_{j=1}^n \hat{g}(j)}{n} \right), \quad i \in [n].$$

With these definitions the vectors u, v and w satisfy $\frac{\Lambda(n)}{2\beta} \cdot w + u = v$, and hence we have

$$\|v\|_2 \le \|u\|_2 + \frac{\Lambda(n)}{2\beta} \cdot \|w\|_2.$$

Since the range of f and g is $[-1, 1]$, we immediately have that

$$\|u\|_2 = \left(\frac{f(\mathbf{1}) - g(\mathbf{1}) - f(-\mathbf{1}) + g(-\mathbf{1})}{n} \right) \cdot \sqrt{n} \le \frac{4}{\sqrt{n}},$$

so all that remains is to bound $\|w\|_2$ from above. To do this, let us define another vector $w' \in \mathbb{R}^n$ by $w'_i = \hat{f}(i) - \hat{g}(i)$. Let $\mathbf{e} \in \mathbb{R}^n$ denote the unit vector $\mathbf{e} = (1/\sqrt{n}, \ldots, 1/\sqrt{n})$. Letting $w'_{\mathbf{e}}$ denote the projection of w along $\mathbf{e}$, it is easy to see that

$$w'_{\mathbf{e}} = \left(\frac{\sum_{j=1}^n (\hat{f}(j) - \hat{g}(j))}{n}, \ldots, \frac{\sum_{j=1}^n (\hat{f}(j) - \hat{g}(j))}{n} \right).$$

This means that $w = w' - w'_{\mathbf{e}}$ and that w is the projection of w' in the space orthogonal to $\mathbf{e}$. Consequently we have $\|w\|_2 \le \|w'\|_2$, and hence

$$\|v\|_2 \le \frac{4}{\sqrt{n}} + \frac{\Lambda(n)}{2\beta} \|w'\|_2$$

as was to be shown. □

Bounding Fourier distance by "correlation distance." The following lemma will be useful for us since it lets us bound from above Fourier distance in terms of the distance between vectors of correlations with individual variables:

Lemma 13. *Let $f, g : \{-1, 1\}^n \to \mathbb{R}$. Then we have*

$$d_{\text{Fourier}}(f, g) \le O(\sqrt{\log n}) \cdot \sqrt{\sum_{i=0}^n (f^*(i) - g^*(i))^2}.$$

Proof. We first observe that $\hat{f}(0) = f^*(0)$ and $\hat{g}(0) = g^*(0)$, so $(\hat{f}(0) - \hat{g}(0))^2 = (f^*(0) - g^*(0))^2$. Consequently it suffices to prove that

$$\sqrt{\sum_{i=1}^n (\hat{f}(i) - \hat{g}(i))^2} \le O(\sqrt{\log n}) \cdot \sqrt{\sum_{i=1}^n (f^*(i) - g^*(i))^2},$$

which is what we show below.

From (6), we get

$$\hat{f}(i) = \alpha \sum_{j=1}^{n} f^*(j) + \beta f^*(i) \quad \text{and} \quad \hat{g}(i) = \alpha \sum_{j=1}^{n} g^*(j) + \beta g^*(i).$$

and thus we have

$$(\hat{f}(i) - \hat{g}(i)) = \alpha \left(\sum_{j=1}^{n} f^*(j) - \sum_{j=1}^{n} g^*(j) \right) + \beta(f^*(i) - g^*(i)).$$

Now consider vectors $u, v, w \in \mathbb{R}^n$ where for $i \in [n]$,

$$u_i = (\hat{f}(i) - \hat{g}(i)), \quad v_i = \left(\sum_{j=1}^{n} f^*(j) - \sum_{j=1}^{n} g^*(j) \right), \quad \text{and} \quad w_i = (f^*(i) - g^*(i))$$

By combining the triangle inequality and Cauchy-Schwarz, we have

$$\|u\|_2^2 \leq 2(\alpha^2 \|v\|_2^2 + \beta^2 \|w\|_2^2),$$

and moreover

$$\|v\|_2^2 = n \left(\sum_{j=1}^{n} f^*(j) - \sum_{j=1}^{n} g^*(j) \right)^2 \leq n^2 \left(\sum_{j=1}^{n} (f^*(j) - g^*(j))^2 \right) = n^2 \|w\|_2^2.$$

Hence, we obtain

$$\|u\|_2^2 \leq 2(\alpha^2 n^2 + \beta^2) \|w\|_2^2$$

Recalling that $\alpha^2 n^2 = \Theta(\log n)$ and $\beta^2 = \Theta(\log n)$, we conclude that

$$d_{\text{Fourier}}(f, g) = \sqrt{\sum_{i=1}^{n} (\hat{f}(i) - \hat{g}(i))^2} \leq O(\sqrt{\log n}) \cdot \sqrt{\sum_{i=1}^{n} (f^*(i) - g^*(i))^2}$$

which completes the proof. □

From Fourier closeness to ℓ_1-closeness. An important technical ingredient in our work is the notion of an affine form $\ell(x)$ having "good anti-concentration" under distribution μ; we now give a precise definition to capture this.

Definition 14 (Anti-concentration). *Fix $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$, and let the affine form $\ell(x)$ be $\ell(x) \overset{def}{=} w \cdot x - \theta$. We say that $\ell(x)$ is (δ, κ)*-anti-concentrated under μ *if $\mathbf{Pr}_{x \sim \mu}[|\ell(x)| \leq \delta] \leq \kappa$.*

The next lemma plays a crucial role in our results. It essentially shows that for $f = \text{sign}(w \cdot x - \theta)$, if the affine form $\ell(x) = w \cdot x - \theta$ is anti-concentrated, then *any* bounded function $g : \{-1,1\}^n \to [-1,1]$ that has $d_{\text{Fourier}}(f, g)$ small must in fact be close to f in ℓ_1 distance under μ.

Lemma 15. *Let $f : \{-1,1\}^n \to \{-1,1\}$, $f = \text{sign}(w \cdot x - \theta)$ be such that $w \cdot x - \theta$ is (δ, κ)-anti-concentrated under μ (for some $\kappa \leq 1/2$), where $|\theta| \leq \|w\|_1$. Let $g : \{-1,1\}^n \to [-1,1]$ be such that $d_{\text{Fourier}}(f, g) \leq \rho$. Then we have*

$$\mathbf{E}_{x \sim \mu}[|f(x) - g(x)|] \leq (4\|w\|_1 \sqrt{\rho})/\delta + 2\kappa.$$

Proof. Let us rewrite $\ell(x) \overset{\text{def}}{=} w \cdot x - \theta$ as a linear combination of the orthonormal basis elements $L_0, L_1, \ldots, L_n$ (w.r.t. μ), i.e.,

$$\ell(x) = \hat{\ell}(\emptyset) L_0 + \sum_{i=1}^{n} \hat{\ell}(i) L_i.$$

Recalling the definitions of L_i for $i = 1, \ldots, n$ and the fact that $L_0 = 1$, we get $\hat{\ell}(\emptyset) = -\theta$.

We first establish an upper bound on $\theta^2 + \sum_{j=1}^{n} \hat{\ell}(j)^2$ as follows :

$$\begin{aligned} \theta^2 + \sum_{j=1}^{n} \hat{\ell}(j)^2 = \mathbf{E}_{x \sim \mu}[(w \cdot x - \theta)^2] &\leq 2\mathbf{E}_{x \sim \mu}[(w \cdot x)^2] + 2\theta^2 \\ &\leq 2\|w\|_1^2 + 2\|w\|_1^2 = 4\|w\|_1^2. \end{aligned}$$

The first equality above uses the fact that the L_i's are orthonormal under μ, while the first inequality uses $(a+b)^2 \leq 2(a^2+b^2)$ for $a, b \in \mathbb{R}$. The second inequality uses the assumed bound on $|\theta|$ and the fact that $|w \cdot x|$ is always at most $\|w\|_1$.

Next, linearity of expectation gives us that

$$\begin{aligned} \mathbf{E}_{x \sim \mu}[(f(x) - g(x)) \cdot (w \cdot x - \theta)] &= \theta(\hat{g}(0) - \hat{f}(0)) + \sum_{j=1}^{n} \hat{\ell}(i)(\hat{f}(i) - \hat{g}(i)) \\ &\leq \sqrt{\sum_{j=0}^{n} (\hat{f}(j) - \hat{g}(j))^2} \cdot \sqrt{\theta^2 + \sum_{j=1}^{n} \hat{\ell}(i)^2} \\ &\leq 2\|w\|_1 \sqrt{\rho} \qquad (9) \end{aligned}$$

where the first inequality is Cauchy-Schwarz and the second follows by the conditions of the lemma.

Now note that since $f = \text{sign}(w \cdot x - \theta)$, for all $x \in \{-1,1\}^n$ we have

$$(f(x) - g(x)) \cdot (w \cdot x - \theta) = |f(x) - g(x)| \cdot |w \cdot x - \theta|$$

Let E denote the event that $|w \cdot x - \theta| > \delta$. Using the fact that the affine form $w \cdot x - \theta$ is (δ, κ)-anti-concentrated, we get that $\mathbf{Pr}[E] \geq 1 - \kappa$, and hence

$$\begin{aligned} \mathbf{E}_{x \sim \mu}[(f(x) - g(x)) \cdot (w \cdot x - \theta)] &\geq \mathbf{E}_{x \sim \mu}[(f(x) - g(x)) \cdot (w \cdot x - \theta) \mid E] \, \mathbf{Pr}[E] \\ &\geq \delta(1-\kappa) \mathbf{E}_{x \sim \mu}[|f(x) - g(x)| \mid E]. \end{aligned}$$

Recalling that $\kappa \leq 1/2$, this together with (9) implies that

$$\mathbf{E}_{x \sim \mu}[|f(x) - g(x)| \mid E] \leq \frac{4\|w\|_1 \sqrt{\rho}}{\delta},$$

which in turn implies (since $|f(x) - g(x)| \leq 2$ for all $x \in \{-1,1\}^n$) that

$$\mathbf{E}_{x \sim \mu}[|f(x) - g(x)|] \leq \frac{4\|w\|_1 \sqrt{\rho}}{\delta} + 2\kappa$$

as was to be shown. □

4 A Useful Anti-concentration Result

In this section we prove an anti-concentration result for monotone increasing η-reasonable affine forms (see Definition 2) under the distribution μ. Note that even if k is a constant the result gives an anti-concentration probability of $O(1/\log n)$; this will be crucial in the proof of our first main result in Section 6.

Theorem 16. *Let $L(x) = w_0 + \sum_{i=1}^n w_i x_i$ be a monotone increasing η-reasonable affine form, so $w_i \geq 0$ for $i \in [n]$ and $|w_0| \leq (1-\eta)\sum_{i=1}^n |w_i|$. Let $k \in [n], 0 < \zeta < 1/2$, $k \geq 2/\eta$ and $r \in \mathbb{R}_+$ be such that $|S| \geq k$, where $S := \{i \in [n] : |w_i| \geq r\}$. Then*

$$\mathbf{Pr}_{x\sim\mu}\left[|L(x)| < r\right] = O\left(\frac{1}{\log n} \cdot \frac{1}{k^{1/3-\zeta}} \cdot \left(\frac{1}{\zeta} + \frac{1}{\eta}\right)\right).$$

This theorem essentially says that under the distribution μ, the random variable $L(x)$ falls in the interval $[-r, r]$ with only a very small probability. Such theorems are known in the literature as "anti-concentration" results, but almost all such results are for the uniform distribution or for other product distributions, and indeed the proofs of such results typically crucially use the fact that the distributions are product distributions.

In our setting, the distribution μ is not even a pairwise independent distribution, so standard approaches for proving anti-concentration cannot be directly applied. Instead, we exploit the fact that μ is a *symmetric* distribution; a distribution is symmetric if the probability mass it assigns to an n-bit string $x \in \{-1,1\}^n$ depends only on the number of 1's of x (and not on their location within the string). This enables us to perform a somewhat delicate reduction to known anti-concentration results for biased product distributions. Our proof adopts a point of view which is inspired by the combinatorial proof of the basic Littlewood-Offord theorem (under the uniform distribution on the hypercube) due to Benjamini et. al. [BKS99]. The detailed proof is given in the following subsection.

4.1 Proof of Theorem 16.

Recall that $\{-1,1\}^n_{=i}$ denotes the i-th "weight level" of the hypercube, i.e., $\{x \in \{-1,1\}^n : \mathrm{wt}(x) = i\}$. We view a random draw $x \sim \mu$ as being done according to a two-stage process:

1. Draw $i \in [n-1]$ with probability $q(n,i) \overset{\text{def}}{=} Q(n,i)/\Lambda(n)$. (Note that this is the probability μ assigns to $\{-1,1\}^n_{=i}$.)
2. Independently pick a uniformly random permutation $\pi : [n] \to [n]$, i.e., $\pi \sim_R \mathbb{S}_n$. The string x is defined to have $x_{\pi(1)} = \ldots = x_{\pi(i)} = 1$ and $x_{\pi(i+1)} = \ldots = x_{\pi(n)} = -1$.

It is easy to see that the above description of μ is equivalent to its original definition. Another crucial observation is that any symmetric distribution can be sampled in the same way, with $q(n,k)$ being the only quantity dependent on the particular distribution. We next define a (r,i)-balanced permutation.

Definition 17 ((r,i)-balanced permutation)**.** *A permutation $\pi : [n] \to [n]$ is called (r,i)-balanced if $|w_0 + \sum_{j=1}^i w_{\pi(j)} - \sum_{j=i+1}^n w_{\pi(j)}| \leq r$.*

For $i \in [n-1]$, let us denote by $p(r,i)$ the fraction of all $n!$ permutations that are (r,i) balanced. That is,

$$p(r,i) = \mathbf{Pr}_{\pi\sim_R\mathbb{S}_n}\left[|w_0 + \sum_{j=1}^{i} w_{\pi(j)} - \sum_{j=i+1}^{n} w_{\pi(j)}| \leq r\right].$$

At this point, as done in [BKS99], we use the above two-stage process defining μ to express the desired "small ball" probability in a more convenient way. Conditioning on the event that the i-th layer is selected in the first stage, the probability that $|L(x)| < r$ is $p(r,i)$. By the law of total probability we can write:

$$\mathbf{Pr}_{x\sim\mu}\left[|L(x)| < r\right] = \sum_{i=1}^{n-1} p(r,i)q(n,i).$$

We again observe that $p(r,i)$ is only dependent on the affine form $L(x)$ and does not depend on the particular symmetric distribution; $q(n,i)$ is the only part dependent on the distribution. The high-level idea of bounding the quantity $\sum_{i=1}^{n-1} p(r,i)q(n,i)$ is as follows: For i which are "close to 1 or $n-1$", we use Markov's inequality to argue that the corresponding $p(r,i)$'s are suitably small; this allows us to bound the contribution of these indices to the sum, using the fact that each $q(n,i)$ is small. For the remaining i's, we use the fact that the p_i's are identical for all symmetric distributions. This allows us to perform a subtle "reduction" to known anti-concentration results for biased product distributions.

We start with the following simple claim, a consequence of Markov's inequality, that shows that if one of i or $n-i$ is reasonably small, the probability $p(r,i)$ is quite small.

Claim 18. *For all $i \in [n-1]$ we have*

$$p(r,i) \leq (4/\eta)\cdot \min\{i, n-i\}/n.$$

Proof. For $i \in [n-1]$, let $\mathcal{E}_i = \{\pi \in \mathbb{S}_n : |w_0 + \sum_{j=1}^{i} w_{\pi(j)} - \sum_{j=i+1}^{n} w_{\pi(j)}| \leq r\}$. By definition we have that $p(r,i) = \mathbf{Pr}_{\pi\sim_R \mathbb{S}_n}\left[\mathcal{E}_i\right]$.

Let $i \leq n/2$. If the event $\mathcal{E}_i$ occurs, we certainly have that $w_0 + \sum_{j=1}^{i} w_{\pi(j)} - \sum_{j=i+1}^{n} w_{\pi(j)} \geq -r$ which yields that

$$\sum_{j=1}^{i} w_{\pi(j)} \geq (1/2)(\sum_{i=1}^{n} w_i - r - w_0).$$

That is,

$$p(r,i) \leq \mathbf{Pr}_{\pi\sim_R \mathbb{S}_n}\left[\sum_{j=1}^{i} w_{\pi(j)} \geq (1/2)(\sum_{i=1}^{n} w_i - r - w_0)\right].$$

Consider the random variable $X = \sum_{j=1}^{i} w_{\pi(j)}$ and denote $\alpha \overset{\text{def}}{=} (1/2)(\sum_{i=1}^{n} w_i - r - w_0)$. We will bound from above the probability

$$\mathbf{Pr}_{\pi\sim_R \mathbb{S}_n}\left[X \geq \alpha\right].$$

Since π is chosen uniformly from $\mathbb{S}_n$, we have that $\mathbf{E}_{\pi\sim_R \mathbb{S}_n}[w_{\pi(j)}] = (1/n)\cdot\sum_{i=1}^{n} w_i$, hence

$$\mathbf{E}_{\pi\sim_R \mathbb{S}_n}[X] = (i/n)\cdot\sum_{i=1}^{n} w_i.$$

Recalling that $|w_0| \leq (1-\eta)\cdot\sum_{i=1}^{n} w_i$ and noting that $\sum_{i=1}^{n} w_i \geq \sum_{i\in S} w_i \geq kr \geq (2/\eta)\cdot r$, we get

$$\alpha \geq (\eta/4)\cdot\sum_{i=1}^{n} w_i.$$

Therefore, noting that $X > 0$, by Markov's inequality, we obtain that

$$\mathbf{Pr}_{\pi\sim_R \mathbb{S}_n}\left[X \geq \alpha\right] \leq \frac{\mathbf{E}_{\pi\sim_R \mathbb{S}_n}[X]}{\alpha} \leq (4/\eta)\cdot(i/n)$$

as was to be proven.

If $i \geq n/2$, we proceed analogously. If $\mathcal{E}_i$ occurs, we have $w_0 + \sum_{j=1}^{i} w_{\pi(j)} - \sum_{j=i+1}^{n} w_{\pi(j)} \leq r$ which yields that

$$\sum_{j=i+1}^{n} w_{\pi(j)} \geq (1/2)(\sum_{i=1}^{n} w_i + w_0 - r).$$

We then repeat the exact same Markov type argument for the random variable $\sum_{j=i+1}^{n} w_{\pi(j)}$. This completes the proof of the claim. □

Of course, the above lemma is only useful when either i or $n-i$ is relatively small. Fix $i_0 < n/2$ (to be chosen later). Note that, for all $i \leq n/2$, it holds $q(n,i) \leq \frac{2}{i \cdot \Lambda(n)}$. By Claim 18 we thus get that

$$\sum_{i=1}^{i_0} p(r,i)q(n,i) \leq \sum_{i=1}^{i_0} \frac{2}{i \cdot \Lambda(n)} \cdot \frac{4}{\eta} \cdot \frac{i}{n} \leq \frac{8 i_0}{\eta \cdot n \cdot \Lambda(n)}. \tag{10}$$

By symmetry, we get

$$\sum_{i=n-i_0}^{n-1} p(r,i)q(n,i) \leq \frac{8 i_0}{\eta \cdot n \cdot \Lambda(n)}. \tag{11}$$

We proceed to bound from above the term $\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i)$. To this end, we exploit the fact, mentioned earlier, that the $p(r,i)$'s depend only on the affine form and not on the particular symmetric distribution over weight levels. We use a subtle argument to essentially reduce anti-concentration statements about μ to known anti-concentration results.

For $\delta \in (0,1)$ let D_δ be the δ-biased distribution over $\{-1,1\}^n$; that is the product distribution in which each coordinate is 1 with probability δ and -1 with probability $1-\delta$. Denote by $g(\delta,i)$ the probability that D_δ assigns to $\{-1,1\}^n_{=i}$, i.e., $g(\delta,i) = \binom{n}{i}\delta^i(1-\delta)^{n-i}$. Theorem 5 now yields

$$\mathbf{Pr}_{x\sim D_\delta}\left[|L(x)| < r\right] \leq \frac{1}{\sqrt{k\delta(1-\delta)}}.$$

Using symmetry, we view a random draw $x \sim D_\delta$ as a two-stage procedure, exactly as in μ, the only difference being that in the first stage we pick the i-th weight level of the hypercube, $i \in [0,n]$, with probability $g(\delta,i)$. We can therefore write

$$\mathbf{Pr}_{x\sim D_\delta}\left[|L(x)| < r\right] = \sum_{i=0}^{n} g(\delta,i)p(r,i)$$

and thus conclude that

$$\sum_{i=i_0+1}^{n-i_0-1} g(\delta,i)p(r,i) \leq \sum_{i=0}^{n} g(\delta,i)p(r,i) \leq \frac{1}{\sqrt{k\delta(1-\delta)}}. \tag{12}$$

We now state and prove the following crucial lemma. The idea of the lemma is to bound from above the sum $\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i)$ by suitably averaging over anti-concentration bounds obtained from the δ-biased product distributions:

Lemma 19. *Let $F : [0,1] \to \mathbb{R}_+$ be such that $q(n,i) \leq \int_{\delta=0}^{1} F(\delta)g(\delta,i)d\delta$ for all $i \in [i_0+1, n-i_0-1]$. Then,*

$$\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i) \leq \frac{1}{\sqrt{k}} \cdot \int_{\delta=0}^{1} \frac{F(\delta)}{\sqrt{\delta(1-\delta)}} d\delta.$$

Proof. We have the following sequence of inequalities

$$\begin{aligned}\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i) &\leq \sum_{i=i_0+1}^{n-1-i_0} \left(\int_{\delta=0}^{1} F(\delta)g(\delta,i)d\delta\right) \cdot p(r,i) \\ &= \int_{\delta=0}^{1} F(\delta) \left(\sum_{i=i_0+1}^{n-i_0-1} g(\delta,i)p(r,i)\right) d\delta \\ &\leq \frac{1}{\sqrt{k}} \cdot \int_{\delta=0}^{1} \frac{F(\delta)}{\sqrt{\delta(1-\delta)}} d\delta\end{aligned}$$

where the first line follows from the assumption of the lemma, the second uses linearity and the third uses (12). □

We thus need to choose appropriately a function F satisfying the lemma statement which can give a non-trivial bound on the desired sum. Fix $\zeta > 0$, and define $F(\delta)$ as

$$F(\delta) \stackrel{\text{def}}{=} \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \left(\frac{1}{\delta^{1/2-\zeta}} + \frac{1}{(1-\delta)^{1/2-\zeta}}\right).$$

The following claim (proved in Section 4.2) says that this choice of $F(\delta)$ satisfies the conditions of Lemma 19:

Claim 20. *For the above choice of $F(\delta)$ and $i_0 \leq i \leq n - i_0$, $q(n,i) \leq \int_{\delta=0}^{1} F(\delta)g(\delta,i)d\delta$.*

Now, applying Lemma 19, for this choice of $F(\delta)$, we get that

$$\begin{aligned}&\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i) \\ \leq\ & \frac{1}{\sqrt{k}} \cdot \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \int_{\delta=0}^{1} \left(\frac{1}{\delta^{1/2-\zeta}} + \frac{1}{(1-\delta)^{1/2-\zeta}}\right) \frac{1}{\sqrt{\delta(1-\delta)}} d\delta. \\ =\ & O\left(\frac{1}{\zeta} \cdot \frac{1}{\sqrt{k}} \cdot \frac{1}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}}\right).\end{aligned}$$

We choose (with foresight) $i_0 = \lceil \frac{n}{k^{1/3}} \rceil$. Then the above expression simplifies to

$$\sum_{i=i_0+1}^{n-i_0-1} p(r,i)q(n,i) = O\left(\frac{1}{\zeta} \cdot \frac{1}{\Lambda(n)} \cdot \frac{1}{k^{1/3-\zeta}}\right)$$

Now plugging $i_0 = \lceil \frac{n}{k^{1/3}} \rceil$ in (10) and (11), we get

$$\sum_{i \leq i_0 \vee i \geq n-i_0} p(r,i)q(n,i) = O\left(\frac{1}{\eta\Lambda(n)} \cdot \frac{1}{k^{1/3}}\right)$$

Combining these equations, we get the final result, and Theorem 16 is proved. □

4.2 Proof of Claim 20

We will need the following basic facts :

Fact 21. *For $x, y \in \mathbb{R}_+$ let $\Gamma : \mathbb{R}_+ \to \mathbb{R}$ be the usual "Gamma" function, so that*

$$\int_{\delta=0}^{1} \delta^x (1-\delta)^y d\delta = \frac{\Gamma(x+1)\cdot\Gamma(y+1)}{\Gamma(x+y+2)}$$

Recall that for $z \in \mathbb{Z}_+$, $\Gamma(z) = (z-1)!$.

Fact 22. *(Stirling's approximation) For $z \in \mathbb{R}_+$, we have $\Gamma(z) = \sqrt{\frac{2\pi}{z}} \cdot \left(\frac{z}{e}\right)^z \cdot \left(1 + O\left(\frac{1}{z}\right)\right)$. In particular, there is an absolute constant $c_0 > 0$ such that for $z \geq c_0$*

$$\frac{1}{2} \cdot \sqrt{\frac{2\pi}{z}} \cdot \left(\frac{z}{e}\right)^z \leq \Gamma(z) \leq 2 \cdot \sqrt{\frac{2\pi}{z}} \cdot \left(\frac{z}{e}\right)^z .$$

Fact 23. *For $x \in \mathbb{R}$ and $x \geq 2$, we have $\left(1 - \frac{1}{x}\right)^x \geq \frac{1}{4}$.*

We can now proceed with the proof of Claim 20. We consider the case when $i_0 \leq i \leq n/2$. (The proof of the complementary case ($n - i_0 - 1 \geq i > n/2$) is essentially identical.) We have the following chain of inequalities:

$$\begin{aligned}
&\int_{\delta=0}^{1} F(\delta) g(\delta, i) d\delta \\
= \quad & \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \cdot \binom{n}{i} \cdot \int_{\delta=0}^{1} \delta^i (1-\delta)^{n-i} \cdot \left(\frac{1}{\delta^{1/2-\zeta}} + \frac{1}{(1-\delta)^{1/2-\zeta}} \right) \, d\delta \\
\geq \quad & \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \cdot \binom{n}{i} \cdot \int_{\delta=0}^{1} \delta^{i-1/2+\zeta} (1-\delta)^{n-i} \, d\delta \\
= \quad & \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \cdot \binom{n}{i} \cdot \frac{\Gamma(n-i+1)\cdot\Gamma(i+1/2+\zeta)}{\Gamma(n+3/2+\zeta)} \quad \text{(using Fact 21)} \\
= \quad & \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \cdot \frac{\Gamma(n+1)}{\Gamma(i+1)\cdot\Gamma(n-i+1)} \cdot \frac{\Gamma(n-i+1)\cdot\Gamma(i+1/2+\zeta)}{\Gamma(n+3/2+\zeta)} \\
= \quad & \frac{1024}{\Lambda(n)} \cdot \frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}} \cdot \frac{\Gamma(n+1)\cdot\Gamma(i+1/2+\zeta)}{\Gamma(i+1)\cdot\Gamma(n+3/2+\zeta)}
\end{aligned}$$

We now proceed to bound from below the right hand side of the last inequality. Towards that, using Fact 22 and assuming n and i are large enough, we have

$$\begin{aligned}
&\frac{\Gamma(n+1)\cdot\Gamma(i+1/2+\zeta)}{\Gamma(i+1)\cdot\Gamma(n+3/2+\zeta)} \\
\geq \quad & \frac{1}{16} \cdot \frac{(n+1)^{n+1/2}}{(i+1)^{i+1/2}} \cdot \frac{(i+1/2+\zeta)^{i+\zeta}}{(n+3/2+\zeta)^{n+\zeta+1}} \\
\geq \quad & \frac{1}{16} \cdot \frac{1}{n+2} \cdot \frac{(n+1)^{n+1/2}}{(i+1)^{i+1/2}} \cdot \frac{(i+1/2+\zeta)^{i+\zeta}}{(n+3/2+\zeta)^{n+\zeta}} \\
\geq \quad & \frac{1}{16} \cdot \frac{1}{n+2} \cdot \frac{(n+1)^{n+\zeta}}{(n+3/2+\zeta)^{n+\zeta}} \cdot \frac{(i+1/2+\zeta)^{i+\zeta}}{(i+1)^{i+\zeta}} \cdot \frac{(n+1)^{1/2-\zeta}}{(i+1)^{1/2-\zeta}} \\
\geq \quad & \frac{1}{256} \cdot \frac{1}{n+2} \cdot \frac{(n+1)^{1/2-\zeta}}{(i+1)^{1/2-\zeta}} \geq \frac{1}{512} \cdot \frac{1}{(n+1)^{1/2+\zeta}} \cdot \frac{1}{(i+1)^{1/2-\zeta}}
\end{aligned}$$

Plugging this back, we get

$$\begin{aligned}\int_{\delta=0}^{1} F(\delta)g(\delta,i)d\delta &\geq \frac{1024}{\Lambda(n)}\cdot\frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}}\cdot\frac{\Gamma(n+1)\cdot\Gamma(i+1/2+\zeta)}{\Gamma(i+1)\cdot\Gamma(n+3/2+\zeta)}\\ &\geq \frac{1024}{\Lambda(n)}\cdot\frac{(n+1)^{1/2+\zeta}}{i_0^{1/2+\zeta}}\cdot\frac{1}{512}\cdot\frac{1}{(n+1)^{1/2+\zeta}}\cdot\frac{1}{(i+1)^{1/2-\zeta}}\\ &= \frac{2}{\Lambda(n)}\cdot\frac{1}{i_0^{1/2+\zeta}}\cdot\frac{1}{(i+1)^{1/2-\zeta}}\geq\frac{2}{\Lambda(n)}\cdot\frac{1}{i}\geq q(n,i)\end{aligned}$$

which concludes the proof of the claim. □

5 A Useful Algorithmic Tool

In this section we describe a useful algorithmic tool arising from recent work in computational complexity theory. The main result we will need is the following theorem of [TTV08] (the ideas go back to [Imp95] and were used in a different form in [DDFS12]):

Theorem 24. *([TTV08]) Let X be a finite domain, μ be a samplable probability distribution over X, $f : X \to [-1,1]$ be a bounded function, and $\mathcal{L}$ be a finite family of Boolean functions $\ell : X \to \{-1,1\}$. There is an algorithm Boosting-TTV with the following properties: Suppose Boosting-TTV is given as input a list $(a_\ell)_{\ell\in\mathcal{L}}$ of real values and a parameter $\xi > 0$ such that $|\mathbf{E}_{x\sim\mu}[f(x)\ell(x)] - a_\ell| \leq \xi/16$ for every $\ell \in \mathcal{L}$. Then Boosting-TTV outputs a function $h : X \to [-1,1]$ with the following properties:*

- *(i) $|\mathbf{E}_{x\sim\mu}[\ell(x)h(x) - \ell(x)f(x)]| \leq \xi$ for every $\ell \in \mathcal{L}$;*
- *(ii) $h(x)$ is of the form $h(x) = P_1(\frac{\xi}{2}\cdot\sum_{\ell\in\mathcal{L}} w_\ell\ell(x))$ where the w_ℓ's are integers whose absolute values sum to $O(1/\xi^2)$.*

The algorithm runs for $O(1/\xi^2)$ iterations, where in each iteration it estimates $\mathbf{E}_{x\sim\mu}[h'(x)\ell(x)]$ to within additive accuracy $\pm\xi/16$. Here each h' is a function of the form $h'(x) = P_1(\frac{\xi}{2}\cdot\sum_{\ell\in\mathcal{L}} v_\ell\ell(x))$, where the v_ℓ's are integers whose absolute values sum to $O(1/\xi^2)$.

We note that Theorem 24 is not explicitly stated in the above form in [TTV08]; in particular, neither the time complexity of the algorithm nor the fact that it suffices for the algorithm to be given "noisy" estimates a_ℓ of the values $\mathbf{E}_{x\sim\mu}[f(x)\ell(x)]$ is explicitly stated in [TTV08]. So for the sake of completeness, in the following we state the algorithm in full (see Figure 5) and sketch a proof of correctness of this algorithm using results that are explicitly proved in [TTV08].

Proof of Theorem 24. It is clear from the description of the algorithm that (if and) when the algorithm Boosting-TTV terminates, the output h satisfies property (i) and has the form $h(x) = P_1(\frac{\xi}{2}\cdot\sum_{\ell\in\mathcal{L}} w_\ell\ell(x))$ where each w_ℓ is an integer. It remains to bound the number of iterations (which gives a bound on the sum of magnitudes of w_ℓ's) and indeed to show that the algorithm terminates at all.

Towards this, we recall Claim 3.4 in [TTV08] states the following:

Claim 25. *For all $x \in supp(\mu)$ and all $t \geq 1$, we have $\sum_{j=1}^{t} f_j(x)\cdot(f(x)-h_{j-1}(x)) \leq (4/\gamma)+(\gamma t)/2$.*

Boosting-TTV

Parameters:

ξ := positive real number
μ := samplable distribution over finite domain X
$\mathcal{L}$:= finite list of functions such that all $\ell \in \mathcal{L}$ maps X to $\{-1,1\}$.
$(a_\ell)_{\ell\in\mathcal{L}}$:= list of real numbers with the promise that some $f : X \to [-1,1]$ has $|\mathbf{E}_{x\sim\mu}[f(x)\ell(x)] - a_\ell| \le \xi/16$ for all $\ell \in \mathcal{L}$.

Output:
An LBF $h(x) \equiv P_1(\sum_{\ell\in\mathcal{L}} w_\ell \ell(x))$, where $w_\ell \in \mathbb{Z}$, such that $\mathbf{E}_{x\sim\mu}[h(x)\ell(x)] - f(x)\ell(x)| \le \xi$ for all $\ell \in \mathcal{L}$.

Algorithm:

```
1. Let $\mathcal{L}^0 \overset{\text{def}}{=} \{\ell : \ell \in \mathcal{L} \text{ or } -\ell \in \mathcal{L}\}$. Fix $\gamma \overset{\text{def}}{=} \xi/2$.
2. Let $h_0 \overset{\text{def}}{=} 0$. Set $t = 0$.
3. For each $\ell \in \mathcal{L}$, find $a_{\ell,t} \in \mathbb{R}$ such that $|\mathbf{E}_{x\sim\mu}[h_t(x)\ell(x)] - a_{\ell,t}| \le \xi/16$.
4. If $|a_\ell - a_{\ell,t}| \le \gamma$ for all $\ell \in \mathcal{L}$, then stop and output $h_t$. Otherwise, fix $\ell$ to be any element of
   $\mathcal{L}$ such that $|a_\ell - a_{\ell,t}| > \gamma$.
      • If $a_\ell - a_{\ell,t} > \gamma$ then set $f_{t+1} \overset{\text{def}}{=} \ell$ else set $f_{t+1} \overset{\text{def}}{=} -\gamma$. Note that $f_{t+1} \in \mathcal{L}^0$.
      • Define $h_{t+1}$ as $h_{t+1}(x) \overset{\text{def}}{=} P_1(\gamma(\sum_{j=1}^{t+1} f_j(x)))$.
5. Set $t = t + 1$ and go to Step 3.
```

Figure 1: Boosting based algorithm from [TTV08]

We now show how this immediately gives Theorem 24. Fix any $j \ge 0$, and suppose without loss of generality that $a_\ell - a_{\ell,j} > \gamma$. We have that

$$|\mathbf{E}_{x\sim\mu}[f_{j+1}(x)h_j(x)] - a_{\ell,j}| \le \xi/16 \quad \text{and hence} \quad \mathbf{E}_{x\sim\mu}[f_{j+1}(x)h_j(x)] \le a_{\ell,j} + \xi/16,$$

and similarly

$$|\mathbf{E}_{x\sim\mu}[f_{j+1}(x)f(x)] - a_\ell| \le \xi/16 \quad \text{and hence} \quad \mathbf{E}_{x\sim\mu}[f_{j+1}(x)f(x)] \ge a_\ell - \xi/16.$$

Combining these inequalities with $a_\ell - a_{\ell,j} > \gamma = \xi/2$, we conclude that

$$\mathbf{E}_{x\sim\mu}[f_{j+1}(x)(f(x) - h_j(x))] \ge 3\xi/8.$$

Putting this together with Claim 25, we get that

$$\frac{3\xi t}{8} \le \sum_{j=1}^{t} \mathbf{E}_{x\sim\mu}[f_j(x)(f(x) - h_{j-1}(x))] \le \frac{4}{\gamma} + \frac{\gamma t}{2}.$$

Since $\gamma = \xi/2$, this means that if the algorithm runs for t time steps, then $8/\xi \ge (\xi t)/8$, which implies that $t \le 64/\xi^2$. This concludes the proof. □

6 Our Main Results

In this section we combine ingredients from the previous subsections and prove our main results, Theorems 26 and 27.

Our first main result gives an algorithm that works if *any* monotone increasing η-reasonable LTF has approximately the right Shapley values:

Theorem 26. *There is an algorithm* IS *(for* Inverse-Shapley*) with the following properties.* IS *is given as input an accuracy parameter $\epsilon > 0$, a confidence parameter $\delta > 0$, and n real values $a(1), \ldots, a(n)$; its output is a pair $v \in \mathbb{R}^n, \theta \in \mathbb{R}$. Its running time is* $\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)}, \log(1/\delta))$*. The performance guarantees of* IS *are the following:*

1. *Suppose there is a monotone increasing η-reasonable LTF $f(x)$ such that $d_{\mathrm{Shapley}}(a, f) \leq 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$. Then with probability $1 - \delta$ algorithm* IS *outputs $v \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ which are such that the LTF $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has $d_{\mathrm{Shapley}}(f, h) \leq \epsilon$.*

2. *For any input vector $(a(1), \ldots, a(n))$, the probability that* IS *outputs $v \in \mathbb{R}^n, \theta \in \mathbb{R}$ such that the LTF $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has $d_{\mathrm{Shapley}}(f, h) > \epsilon$ is at most δ.*

Proof. We first note that we may assume $\epsilon > n^{-c}$ for a constant $c > 0$ of our choosing, for if $\epsilon \leq n^{-c}$ then the claimed running time is $2^{\Omega(n^2 \log n)}$. In this much time we can easily enumerate all LTFs over n variables (by trying all weight vectors with integer weights at most n^n; this suffices by [MTT61]) and compute their Shapley values exactly, and thus solve the problem. So for the rest of the proof we assume that $\epsilon > n^{-c}$.

It will be obvious from the description of IS that property (2) above is satisfied, so the main job is to establish (1). Before giving the formal proof we first describe an algorithm and analysis achieving (1) for an idealized version of the problem. We then describe the actual algorithm and its analysis (which build on the idealized version).

Recall that the algorithm is given as input ϵ, δ and $a(1), \ldots, a(n)$ that satisfy $d_{\mathrm{Shapley}}(a, f) \leq 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$ for some monotone increasing η-reasonable LTF f. The idealized version of the problem is the following: we assume that the algorithm is also given the two real values $f^*(0)$, $\sum_{i=1}^n f^*(i)/n$. It is also helpful to note that since f is monotone and η-reasonable (and hence is not a constant function), it must be the case that $f(\mathbf{1}) = 1$ and $f(-\mathbf{1}) = -1$.

The algorithm for this idealized version is as follows: first, using Lemma 8, the values $\tilde{f}(i)$, $i = 1, \ldots, n$ are converted into values $a^*(i)$ which are approximations for the values $f^*(i)$. Each $a^*(i)$ satisfies $|a^*(i) - f^*(i)| \leq 1/\mathrm{poly}(n, 2^{O(\mathrm{poly}(1/\epsilon))})$. The algorithm sets $a^*(0)$ to $f^*(0)$. Next, the algorithm runs Boosting-TTV with the following input: the family $\mathcal{L}$ of Boolean functions is $\{1, x_1, \ldots, x_n\}$; the values $a^*(0), \ldots, a^*(n)$ comprise the list of real values; μ is the distribution; and the parameter ξ is set to $1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$. (We note that each execution of Step 3 of Boosting-TTV, namely finding values that closely estimate $\mathbf{E}_{x \sim \mu}[h_t(x) x_i]$ as required, is easily achieved using a standard sampling scheme; for completeness in Appendix B we describe a procedure Estimate-Correlation that can be used to do all the required estimations with overall failure probability at most δ.) Boosting-TTV outputs an LBF $h(x) = P_1(v \cdot x - \theta)$; the output of our overall algorithm is the LTF $h'(x) = \mathrm{sign}(v \cdot x - \theta)$.

Let us analyze this algorithm for the idealized scenario. By Theorem 24, the output function h that is produced by Boosting-TTV is an LBF $h(x) = P_1(v \cdot x - \theta)$ that satisfies $\sqrt{\sum_{j=0}^n (h^*(j) - f^*(j))^2} = 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$. Given this, Lemma 13 implies that $d_{\mathrm{Fourier}}(f, h) \leq \rho \overset{\mathrm{def}}{=} 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$.

At this point, we have established that h is a bounded function that has $d_{\mathrm{Fourier}}(f, h) \leq 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$. We would like to apply Lemma 15 and thereby assert that the ℓ_1 distance between f and h (with respect to μ) is small. To see that we can do this, we first note that since f is a monotone increasing η-reasonable LTF, by Theorem 3 it has a representation as $f(x) = \mathrm{sign}(w \cdot x + w_0)$ whose weights satisfy the properties claimed in that theorem; in particular, for any choice of $\zeta > 0$, after rescaling all the weights, the largest-magnitude weight has magnitude 1, and the $k \overset{\mathrm{def}}{=} \Theta_{\zeta,\eta}(1/\epsilon^{6+2\zeta})$ largest-magnitude weights each have magnitude at least $r \overset{\mathrm{def}}{=} 1/(n \cdot k^{O(k)})$. (Note that since $\epsilon \geq n^{-c}$ we indeed have $k \leq n$ as required.) Given this, Theorem 16 implies that the affine form $L(x) = w \cdot x + w_0$ satisfies

$$\mathbf{Pr}_{x\sim\mu}[|L(x)| < r] \leq \kappa \overset{\mathrm{def}}{=} \epsilon^2/(512\log(n)), \tag{13}$$

i.e., it is (r,κ)-anticoncentrated with $\kappa = \epsilon^2/(512\log(n))$. Thus we may indeed apply Lemma 15, and it gives us that

$$\mathbf{E}_{x\sim\mu}[|f(x) - h(x)|] \leq \frac{4\|w\|_1\sqrt{\rho}}{r} + 2\kappa \leq \epsilon^2/(128\log n). \tag{14}$$

Now let $h' : \{-1,1\}^n \to \{-1,1\}$ be the LTF defined as $h'(x) = \mathrm{sign}(v \cdot x - \theta)$ (recall that h is the LBF $P_1(v \cdot x - \theta)$). Since f is a $\{-1,1\}$-valued function, it is clear that for every input x in the support of μ, the contribution of x to $\mathbf{Pr}_{x\sim\mu}[f(x) \neq h'(x)]$ is at most twice its contribution to $\mathbf{E}_{x\sim\mu}[|f(x) - h(x)|]$. Thus we have that $\mathbf{Pr}_{x\sim\mu}[f(x) \neq h'(x)] \leq \epsilon^2/(64\log n)$. We may now apply Fact 7 to obtain that $d_{\mathrm{Fourier}}(f,h') \leq \epsilon/(4\sqrt{\log n})$. Finally, Lemma 12 gives that

$$d_{\mathrm{Shapley}}(f,h') \leq 4/\sqrt{n} + \sqrt{\Lambda(n)} \cdot \epsilon/(4\sqrt{\log n}) < \epsilon/2.$$

So indeed the LTF $h'(x) = \mathrm{sign}(v \cdot x - \theta)$ satisfies $d_{\mathrm{Shapley}}(f,h') \leq \epsilon/2$ as desired.

Now we turn from the idealized scenario to actually prove Theorem 26, where we are not given the values of $f^*(0)$ and $\sum_{i=1}^n f^*(i)/n$. To get around this, we note that $f^*(0)$, $\sum_{i=1}^n f^*(i)/n \in [-1,1]$. So the idea is that we will run the idealized algorithm repeatedly, trying "all" possibilities (up to some prescribed granularity) for $f^*(0)$ and for $\sum_{i=1}^n f^*(i)/n$. At the end of each such run we have a "candidate" LTF h'; we use a simple procedure Shapley-Estimate (see Appendix B) to estimate $d_{\mathrm{Shapley}}(f,h')$ to within additive accuracy $\pm\epsilon/10$, and we output any h' whose estimated value of $d_{\mathrm{Shapley}}(f,h')$ is at most $8\epsilon/10$.

We may run the idealized algorithm $\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$ times without changing its overall running time (up to polynomial factors). Thus we can try a net of possible guesses for $f^*(0)$ and $\sum_{i=1}^n f^*(i)/n$ which is such that one guess will be within $\pm 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$ of the the correct values for both parameters. It is straightforward to verify that the analysis of the idealized scenario given above is sufficiently robust that when these "good" guesses are encountered, the algorithm will with high probability generate an LTF h' that has $d_{\mathrm{Shapley}}(f,h') \leq 6\epsilon/10$. A straightforward analysis of running time and failure probability shows that properties (1) and (2) are achieved as desired, and Theorem 26 is proved. □

For any monotone η-reasonable target LTF f, Theorem 26 constructs an output LTF whose Shapley distance from f is at most ϵ, but the running time is exponential in $\mathrm{poly}(1/\epsilon)$. We now show that if the target monotone η-reasonable LTF f has integer weights that are at most W, then we can construct an output LTF h with $d_{\mathrm{Shapley}}(f,h) \leq n^{-1/8}$ running in time $\mathrm{poly}(n,W)$; this is a far faster running time than provided by Theorem 26 for such small ϵ. (The "1/8" is chosen for convenience; it will be clear from the proof that any constant strictly less than 1/6 would suffice.)

Theorem 27. *There is an algorithm* ISBW *(for* Inverse-Shapley *with* Bounded Weights*) with the following properties.* ISBW *is given as input a weight bound $W \in \mathbb{Z}_+$, a confidence parameter $\delta > 0$, and n real values $a(1), \dots, a(n)$; its output is a pair $v \in \mathbb{R}^n, \theta \in \mathbb{R}$. Its running time is* $\mathrm{poly}(n, W, \log(1/\delta))$*. The performance guarantees of* ISBW *are the following:*

1. *Suppose there is a monotone increasing η-reasonable LTF $f(x) = \mathrm{sign}(u \cdot x - \theta)$, where each u_i is an integer with $|u_i| \leq W$, such that $d_{\mathrm{Shapley}}(a, f) \leq 1/\mathrm{poly}(n, W)$. Then with probability $1-\delta$ algorithm* ISBW *outputs $v \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ which are such that the LTF $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has $d_{\mathrm{Shapley}}(f, h) \leq n^{-1/8}$.*

2. *For any input vector $(a(1), \dots, a(n))$, the probability that* IS *outputs v, θ such that the LTF $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has $d_{\mathrm{Shapley}}(f, h) > n^{-1/8}$ is at most δ.*

Proof. Let $f(x) = \mathrm{sign}(u \cdot x - \theta)$ be as described in the theorem statement. We may assume that each $|u_i| \geq 1$ (by scaling all the u_i's and θ by $2n$ and then replacing any zero-weight u_i with 1). Next we observe that for such an affine form $u \cdot x - \theta$, Theorem 16 immediately yields the following corollary:

Corollary 28. *Let $L(x) = \sum_{i=1}^n u_i x_i - \theta$ be a monotone increasing η-reasonable affine form. Suppose that $u_i \geq r$ for all $i = 1, \dots, n$. Then for any $\zeta > 0$, we have*

$$\mathbf{Pr}_{x \sim \mu}\left[|L(x)| < r\right] = O\left(\frac{1}{\log n} \cdot \frac{1}{n^{1/3-\zeta}} \cdot \left(\frac{1}{\zeta} + \frac{1}{\eta}\right)\right).$$

With this anti-concentration statement in hand, the proof of Theorem 27 closely follows the proof of Theorem 26. The algorithm runs Boosting-TTV with $\mathcal{L}$, $a^*(i)$ and μ as before but now with ξ set to $1/\mathrm{poly}(n, W)$. The LBF h that Boosting-TTV outputs satisfies $d_{\mathrm{Fourier}}(f, h) \leq \rho \overset{\mathrm{def}}{=} 1/\mathrm{poly}(n, W)$. We apply Corollary 28 to the affine form $L(x) \overset{\mathrm{def}}{=} \frac{u}{\|u\|_1} \cdot x - \frac{\theta}{\|u\|_1}$ and get that for $r = 1/\mathrm{poly}(n, W)$, we have

$$\mathbf{Pr}_{x \sim \mu}[|L(x)| < r] \leq \kappa \overset{\mathrm{def}}{=} \epsilon^2/(1024 \log n) \tag{15}$$

where now $\epsilon \overset{\mathrm{def}}{=} n^{-1/8}$, in place of Equation (13). Applying Lemma 15 we get that

$$\mathbf{E}_{x \sim \mu}[|f(x) - h(x)|] \leq \frac{4\|w\|_1 \sqrt{\rho}}{r} + 4\kappa \leq \epsilon^2/(128 \log n)$$

analogous to (14). The rest of the analysis goes through exactly as before, and we get that the LTF $h'(x) = \mathrm{sign}(v \cdot x - \theta)$ satisfies $d_{\mathrm{Shapley}}(f, h') \leq \epsilon/2$ as desired. The rest of the argument is unchanged so we do not repeat it. □

7 Conclusions and Future Work

The problem of designing a weighted voting game that (exactly or approximately) achieves a desired set of Shapley values has received considerable attention in the social choice literature, where several heuristics and exponential time algorithms have been proposed. This work provides the first provably correct efficient approximation algorithm for this problem.

An obvious open problem is to improve the dependence on the error parameter ϵ in the running time. Since the running time of our algorithm is of the form $\alpha(\epsilon) \cdot n^c$ for a fixed universal constant c, the algorithm is an Efficient Polynomial Time Approximation Scheme (EPTAS). Is there a

Fully Polynomial Time Approximation Scheme (FPTAS), i.e., an algorithm with running time $\mathrm{poly}(n, 1/\epsilon)$?

It would also be interesting to characterize the complexity of the *exact* problem (i.e., that of designing a weighted voting game that *exactly* achieves a given set of Shapley values, or deciding that no such game exists). We conjecture that the exact problem is intractable, namely $\sharp P$-hard.

Acknowledgement. We would like to thank Edith Elkind for asking the question about Shapley values and for useful pointers to the literature. We thank Christos Papadimitriou for insightful conversations.

References

[AGKW09] M. Aizenman, F. Germinet, A. Klein, and S. Warzel. On Bernoulli decompositions for random variables, concentration bounds, and spectral localization. *Probability Theory and Related Fields*, 143(1-2):219–238, 2009.

[APL07] H. Aziz, M. Paterson, and D. Leech. Efficient algorithm for designing weighted voting games. In *IEEE Intl. Multitopic Conf.*, pages 1–6, 2007.

[Ban65] J. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[BKS99] I. Benjamini, G. Kalai, and O. Schramm. Noise sensitivity of Boolean functions and applications to percolation. *Inst. Hautes Études Sci. Publ. Math.*, 90:5–43, 1999.

[BMR$^+$10] Y. Bachrach, E. Markakis, E. Resnick, A. Procaccia, J. Rosenschein, and A. Saberi. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20(2):105–122, 2010.

[Cho61] C.K. Chow. On the characterization of threshold functions. In *Proc. 2nd FOCS*, pages 34–38, 1961.

[DDFS12] A. De, I. Diakonikolas, V. Feldman, and R. Servedio. Near-optimal solutions for the Chow Parameters Problem and low-weight approximation of halfspaces. In *STOC*, pages 709–728, 2012.

[dK08] Bart de Keijzer. A survey on the computation of power indices. Available at http://www.st.ewi.tudelft.nl/∼tomas/theses/DeKeijzerSurvey.pdf, 2008.

[dKKZ10] Bart de Keijzer, Tomas Klos, and Yingqian Zhang. Enumeration and exact design of weighted voting games. In *AAMAS*, pages 391–398, 2010.

[DP78] J. Deegan and E. Packel. A new index of power for simple n-person games. *International Journal of Game Theory*, 7:113–123, 1978.

[DS09] I. Diakonikolas and R. Servedio. Improved approximation of linear threshold functions. In *Proc. 24th CCC*, pages 161–172, 2009.

[EGGW07] E. Elkind, L.A. Goldberg, P.W. Goldberg, and M. Wooldridge. Computational complexity of weighted voting games. In *AAAI*, pages 718–723, 2007.

[FWJ08] S. Fatima, M. Wooldridge, and N. Jennings. An Anytime Approximation Method for the Inverse Shapley Value Problem. In *AAMAS'08*, pages 935–942, 2008.

[Gol06] P. Goldberg. A Bound on the Precision Required to Estimate a Boolean Perceptron from its Average Satisfying Assignment. *SIDMA*, 20:328–343, 2006.

[Hås94] J. Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.

[Hol82] M.J. Holler. Forming coalitions and measuring voting power. *Political studies*, 30:262–271, 1982.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proc. 36th FOCS*, pages 538–545, 1995.

[Juk01] S. Jukna. *Extremal combinatorics with applications in computer science*. Springer, 2001.

[KS06] G. Kalai and S. Safra. Threshold phenomena and influence. In *Computational Complexity and Statistical Physics*, pages 25–60. Oxford University Press, 2006.

[Kur11] S. Kurz. On the inverse power index problem. *Optimization*, 2011. DOI:10.1080/02331934.2011.587008.

[Lee03] D. Leech. Computing power indices for large voting games. *Management Science*, 49(6), 2003.

[MTT61] S. Muroga, I. Toda, and S. Takasu. Theory of majority switching elements. *J. Franklin Institute*, 271:376–418, 1961.

[OS08] R. O'Donnell and R. Servedio. The Chow Parameters Problem. In *Proc. 40th STOC*, pages 517–526, 2008.

[OS11] R. O'Donnell and R. Servedio. The Chow Parameters Problem. *SIAM J. on Comput.*, 40(1):165–199, 2011.

[Owe72] G. Owen. Multilinear extensions of games. *Management Science*, 18(5):64–79, 1972. Part 2, Game theory and Gaming.

[Rot88] A.E. Roth, editor. *The Shapley value*. University of Cambridge Press, 1988.

[Ser07] R. Servedio. Every linear threshold function has a low-weight approximator. *Comput. Complexity*, 16(2):180–209, 2007.

[SS54] L. Shapley and M. Shubik. A Method for Evaluating the Distribution of Power in a Committee System. *American Political Science Review*, 48:787–792, 1954.

[TTV08] L. Trevisan, M. Tulsiani, and S. Vadhan. Regularity, Boosting and Efficiently Simulating every High Entropy Distribution . Technical Report 103, ECCC, 2008. Conference version in Proc. CCC 2009.

[ZFBE08] M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind. Manipulating the quota in weighted voting games. In *AAAI*, 2008.

Appendix

A LTF representations with "nice" weights

In this section, we prove Theorem 3. This theorem essentially says that given any η-reasonable LTF, there is an equivalent representation of the LTF which is also η-reasonable and is such that the weights of the linear form (when arranged in decreasing order of magnitude) decrease somewhat "smoothly." For convenience we recall the exact statement of the theorem:

Theorem 3. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be an η-reasonable LTF and $k \in [2,n]$. There exists a representation of f as $f(x) = \mathrm{sign}(v_0 + \sum_{i=1}^n v_i x_i)$ such that (after reordering coordinates so that condition (i) below holds) we have: (i) $|v_i| \geq |v_{i+1}|$, $i \in [n-1]$; (ii) $|v_0| \leq (1-\eta)\sum_{i=1}^n |v_i|$; and (iii) for all $i \in [0, k-1]$ we have $|v_i| \leq (2/\eta)\cdot\sqrt{n}\cdot k^{\frac{k}{2}}\cdot\sigma_k$, where $\sigma_k \overset{def}{=} \sqrt{\sum_{j\geq k} v_j^2}$.*

Proof of Theorem 3. The proof proceeds along similar lines as the proof of Lemma 5.1 from [OS11] (itself an adaptation of the argument of Muroga et. al. from [MTT61]) with some crucial modifications.

Since f is η-reasonable, there exists a representation as $f(x) = \mathrm{sign}(w_0 + \sum_{i=1}^n w_i x_i)$ (where we assume w.l.o.g. that $|w_i| \geq |w_{i+1}|$ for all $i \in [n-1]$) such that $|w_0| \leq (1-\eta)\sum_{i=1}^n |w_i|$. Of course, this representation may not satisfy condition (iii) of the theorem statement. We proceed to construct the desired alternate representation as follows: First, we set $v_i = w_i$ for all $i \geq k$. We then set up a feasible linear program $\mathcal{LP}$ with variables $u_0, \ldots, u_{k-1}$ and argue that there exists a feasible solution to $\mathcal{LP}$ with the desired properties.

Let $h : \{\pm 1\}^{k-1} \to \mathbb{R}$ denote the affine form $h(x) = w_0 + \sum_{j=1}^{k-1} w_j x_j$. We consider the following linear system $\mathcal{S}$ of 2^{k-1} equations in k unknowns $u_0, \ldots, u_{k-1}$: For each $x \in \{\pm 1\}^{k-1}$ we include the equation

$$u_0 + \sum_{i=1}^{k-1} u_i x_i = h(x).$$

It is clear that the system $\mathcal{S}$ is satisfiable, since $(u_0, \ldots, u_{k-1}) = (w_0, \ldots, w_{k-1})$ is a solution.

We now relax the above linear system into the linear program $\mathcal{LP}$ (over the same variables) as follows: Let $C \overset{\text{def}}{=} \sqrt{n}\sigma_k$. Our linear program has the following constraints:

- For each $x \in \{\pm 1\}^{k-1}$ we include the (in)equality:

$$u_0 + \sum_{i=1}^{k-1} u_i x_i \begin{cases} \geq C & \text{if } h(x) \geq C, \\ = h(x) & \text{if } |h(x)| < C, \\ \leq -C & \text{if } h(x) \leq -C. \end{cases} \tag{16}$$

- For each $i \in [0, k-1]$, we add the constraints $\mathrm{sign}(u_i) = \mathrm{sign}(w_i)$. Since the w_i's are known, these are linear constraints, i.e., constraints like $u_1 \leq 0$, $u_2 \geq 0$, etc.

- We also add the constraints of the form $|u_i| \geq |u_{i+1}|$ for $1 \leq i \leq k-2$ and also $|u_{k-1}| \geq |w_k|$. Note that these constraints are equivalent to the linear constraints: $u_i \cdot \mathrm{sign}(w_i) \geq u_{i+1} \cdot \mathrm{sign}(w_{i+1})$ and $\mathrm{sign}(w_{k-1}) \cdot u_{k-1} \geq |w_k|$.

- We let $q = \lceil 1/\eta \rceil$ and $\eta' = 1/q$. Clearly, $\eta' \leq \eta$. We now add the constraint $|u_0| \leq (1-\eta') \cdot \left(\sum_{j=1}^{k-1} |u_j| + \sum_{j=k}^{n} |w_j| \right)$. Note that this is also a linear constraint over the variables

$u_0, u_1, \dots, u_{k-1}$. Indeed, it can be equivalently written as:

$$\text{sign}(w_0) \cdot u_0 - (1-\eta') \sum_{j=1}^{k-1} \text{sign}(w_j) \cdot u_j \le (1-\eta') \sum_{j=k}^{n} |w_j|.$$

Note that the RHS is strictly bounded from above by C, since

$$\sum_{j=k}^{n} |w_j| \le \sqrt{n-k+1} \cdot \sigma_k < \sqrt{n}\sigma_k,$$

where the first inequality is Cauchy-Schwarz and the second uses the fact that $k \ge 2$.

We observe that the above linear program is feasible. Indeed, it is straightforward to verify that all the constraints are satisfied by the vector $(w_0, \dots, w_{k-1})$. In particular, the last constraint is satisfied because $|w_0| \le (1-\eta) \cdot \left(\sum_{j=1}^{k-1} |w_j| + \sum_{j=k}^{n} |w_j|\right)$ and hence *a fortiori*, $|w_0| \le (1-\eta') \cdot \left(\sum_{j=1}^{k-1} |w_j| + \sum_{j=k}^{n} |w_j|\right)$.

Claim 29. *Let $(v_0, \dots, v_{k-1})$ be* any *feasible solution to $\mathcal{LP}$ and consider the LTF*

$$f'(x) = \text{sign}(v_0 + \sum_{j=1}^{k-1} v_j x_j + \sum_{j=k}^{n} w_j x_j).$$

Then $f'(x) = f(x)$ for all $x \in \{-1,1\}^n$.

Proof. Given $x \in \{-1,1\}^n$, we have

$$h(x) = h(x_1, \dots, x_{k-1}) = w_0 + \sum_{j=1}^{k-1} w_j x_j;$$

Let us also define

$$h'(x) = h'(x_1, \dots, x_{k-1}) = v_0 + \sum_{j=1}^{k-1} v_j x_j$$

$$t(x) = \sum_{j \ge k} w_j x_j$$

Then, we have $f(x) = \text{sign}\left(h(x) + t(x)\right)$ and $f'(x) = \text{sign}\left(h'(x) + t(x)\right)$. Now, if $x \in \{-1,1\}^n$ is an input such that $|h(x)| < C$, then we have $h'(x) = h(x)$ by construction, and hence $f(x) = f'(x)$. If $x \in \{-1,1\}^n$ is such that $|h(x)| \ge C$, then by construction we also have that $|h'(x)| \ge C$. Also, note that $h(x)$ and $h'(x)$ always have the same sign. Hence, in order for f and f' to disagree on x, it must be the case that $|t(x)| \ge C$. But this is not possible, since $|t(x)| \le \sum_{j=k}^{n} |w_j| \le \sqrt{n-1} \cdot \sigma_k < C$. This completes the proof of the claim. □

We are almost done, except that we need to choose a solution $(v_0, \dots, v_{k-1})$ to $\mathcal{LP}$ satisfying property (iii) in the statement of the theorem. The next claim ensures that this can always be achieved.

Claim 30. *There is a feasible solution $v = (v_0, \dots, v_{k-1})$ to the $\mathcal{LP}$ which satisfies property (iii) in the statement of the theorem.*

Proof. We select a feasible solution $v = (v_0, \dots, v_{k-1})$ to the $\mathcal{LP}$ that maximizes the number of *tight* inequalities (i.e., satisfied with equality). If more than one feasible solutions satisfy this property, we choose one arbitrarily. We require the following fact from [MTT61] (a proof can be found in [Hås94, DS09]).

Fact 31. *There exists a linear system $A \cdot v = b$ that uniquely specifies the vector v. The rows of (A, b) correspond to rows of the constraint matrix of $\mathcal{LP}$ and the corresponding RHS respectively.*

At this point, we use Cramer's rule to complete the argument. In particular, note that $v_i = \det(A_i)/\det(A)$ where A_i is the matrix obtained by replacing the i-th column of A by b. In particular, we want to give an upper bound on the magnitude of v_i; we do this by showing a lower bound on $|\det(A)|$ and an upper bound on $|\det(A_i)|$.

We start by showing that $|\det(A)| \geq \eta'$. First, since A is invertible, $\det(A) \neq 0$. Now, note that all rows of A have entries in $\{-1, 0, 1\}$ except potentially one "special" row which has entries from the set $\{\pm 1, \pm(1-\eta')\}$. If the special row does not appear, it is clear that $|\det(A)| \geq 1$, since it is not zero and the entries of A are all integers. If, on the other hand, the special row appears, simply expanding $\det(A)$ along that row gives that $\det(A) = a \cdot (1-\eta') + b$ where $a, b \in \mathbb{Z}$. As $\eta' = 1/q$ for some $q \in Z$ and $\det(A) \neq 0$, we deduce that $|\det(A)| \geq \eta'$, as desired.

We bound $|\det(A_i)|$ from above by recalling the following fact.

Fact 32. *(Hadamard's inequality) If $A \in \mathbb{R}^{n \times n}$ and $v_1, \ldots, v_n \in \mathbb{R}^n$ are the columns of A, then $|\det(A)| \leq \prod_{j=1}^{n} \|v_j\|_2$.*

Now, observe that for all i, the i-th column of A_i (i.e., vector b) has all its entries bounded by C, hence $\|v_i\|_2 \leq C\sqrt{k}$. All other columns have entries bounded from above by 1 and thus for $j \neq i$, $\|v_j\|_2 \leq \sqrt{k}$. Therefore, $\det(A_i) \leq C \cdot k^{k/2}$. Thus, we conclude that $|v_i| \leq (C \cdot k^{k/2})/\eta'$. Further, as $(1/\eta') = \lceil (1/\eta) \rceil \leq (2/\eta)$, we get $|v_i| \leq 2C \cdot k^{k/2}/\eta$, completing the proof of the claim. □

The proof of Theorem 3 is now complete. □

B Estimating correlations and Shapley values

Our algorithms need to estimate expectations of the form $f^*(i) = \mathbf{E}_{x \sim \mu}[f(x)x_i]$ and to estimate Shapley values $\tilde{f}(i)$, where $f : \{-1,1\}^n \to [-1,1]$ is an explicitly given function (an LBF). This is quite straightforward using standard techniques (see e.g. [BMR+10]) but for completeness we briefly state and prove the estimation guarantees that we will need.

Estimating correlations with variables. We will use the following:

Proposition 33. *There is a procedure* Estimate-Correlation *with the following properties: The procedure is given oracle access to a function $f : \{-1,1\}^n \to [-1,1]$, a desired accuracy parameter γ, and a desired failure probability δ. The procedure makes $O(n \log(n/\delta)/\gamma^2)$ oracle calls to f and runs in time $O(n^2 \log(n/\delta)/\gamma^2)$ (counting each oracle call to f as taking one time step). With probability $1-\delta$ it outputs a list of numbers $a^*(0), a^*(1), \ldots, a^*(n)$ such that $|a^*(j) - f^*(j)| \leq \gamma/\sqrt{n+1}$ for all $j = 0, \ldots, n$. (Recall that $f^*(j)$ equals $\mathbf{E}_{x \sim \mu}[f(x)x_j]$, where $x_0 \equiv 1$).*

Proof. The procedure works simply by empirically estimating all the values $f^*(j) = \mathbf{E}_{x \sim \mu}[f(x)x_j]$, $j = 0, \ldots, n$, using a single sample of m independent draws from μ. Since the random variable $(f(x)x_j)_{x \sim \mu}$ is bounded by 1 in absolute value, a straightforward Chernoff bound gives that for $m = O(n \log(n/\delta)/\gamma^2)$, each estimate $a^*(j)$ of $f^*(j)$ is accurate to within an additive $\pm\gamma/\sqrt{n+1}$ with failure probability at most $\delta/(n+1)$. A union bound over $j = 0, \ldots, n$ finishes the argument. □

Estimating Shapley values. This is equally straightforward:

Proposition 34. *There is a procedure* Estimate-Shapley *with the following properties: The procedure is given oracle access to a function $f : \{-1,1\}^n \to [-1,1]$, a desired accuracy parameter γ, and a desired failure probability δ. The procedure makes $O(n\log(n/\delta)/\gamma^2)$ oracle calls to f and runs in time $O(n^2\log(n/\delta)/\gamma^2)$ (counting each oracle call to f as taking one time step). With probability $1-\delta$ it outputs a list of numbers $\tilde{a}(1),\ldots,\tilde{a}(n)$ such that $d_{\mathrm{Shapley}}(a,f) \le \gamma$.*

Proof. The procedure empirically estimates each $\tilde{f}(j)$, $j = 1,\ldots,n$, to additive accuracy $\gamma/\sqrt{n}$ using Equation (1). This is done by generating a uniform random $\pi \sim \mathbb{S}_n$ and then, for each $i = 1,\ldots,n$, constructing the two inputs $x^+(\pi,i)$ and $x(\pi,i)$ and calling the oracle for f twice to compute $f(x^+(\pi,i)) - f(x(\pi,i))$. Since $|f(x^+(\pi,i)) - f(x(\pi,i))| \le 2$ always, a sample of $m = O(n\log(n/\delta)/\gamma^2)$ permutations suffices to estimate all the $\tilde{f}(i)$ values to additive accuracy $\pm\gamma/\sqrt{n}$ with total failure probability at most δ. If each estimate $\tilde{a}(i)$ is additively accurate to within $\pm\gamma/\sqrt{n}$, then $d_{\mathrm{Shapley}}(a,f) \le \gamma$ as desired. □