# The Inverse Shapley Value Problem

Anindya De[1⋆], Ilias Diakonikolas[1⋆⋆], and Rocco Servedio[2⋆⋆⋆]

[1] UC Berkeley {anindya,ilias}@cs.berkeley.edu
[2] Columbia University rocco@cs.columbia.edu

**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 $\mathrm{poly}(n)$ time (the degree of the polynomial is independent of $\epsilon$) 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 $\epsilon$. If there is a "reasonable" voting scheme in which all voting weights are integers at most $\mathrm{poly}(n)$ that approximately achieves the desired Shapley values, then our algorithm runs in time $\mathrm{poly}(n)$ 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? [3] 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 be the case that we would like to assign different levels of

⋆ Research supported by NSF award CCF-1118083.
⋆⋆ Research supported by a Simons Postdoctoral Fellowship.
⋆⋆⋆ Research supported in part by NSF awards CCF-0915929 and CCF-1115703.

[3] 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.

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 weighted 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 [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, e.g. [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. 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 5 for Theorem 3 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 $\tilde{a}(1), \ldots, \tilde{a}(n)$. 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 $\tilde{a}(1), \ldots, \tilde{a}(n)$, then with very high probability the algorithm outputs $v \in \mathbb{R}^n$, $\theta \in \mathbb{R}$ such that the linear threshold function $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has Shapley values $\epsilon$-close to those of $f$.*

Our second main theorem gives an even stronger guarantee if there is a weighted voting scheme with small weights (at most $\mathrm{poly}(n)$) whose Shapley values are close to the desired values. For this problem we give an algorithm which achieves $1/\mathrm{poly}(n)$ accuracy in $\mathrm{poly}(n)$ time. An informal statement of this result is (see Section 5 for Theorem 4 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 $\tilde{a}(1), \ldots, \tilde{a}(n)$. The algorithm has the following performance guarantee: if there is any monotone increasing reasonable LTF $f(x) = \mathrm{sign}(w \cdot x - \theta)$ whose Shapley values are very close to the given values $\tilde{a}(1), \ldots, \tilde{a}(n)$ 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 $h(x) = \mathrm{sign}(v \cdot x - \theta)$ 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 $\mu$ 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 $\mu$ as required for our results. This is done by showing that any linear threshold function can be expressed with "nice" weights, 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 3 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 $(\mathbf{E}[f(x)x_i])_{i=1,\ldots,n}$ (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 $\mu$ 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 $\mu$* and relating these coefficients to the Shapley values; see Section 2. [4] Armed with this notion, we prove a key result (Lemma 6) saying that if the LTF $f$ is anti-concentrated under distribution $\mu$, then any bounded function $g$ which closely matches the degree-1 Fourier coefficients of $f$ must be close to $f$ in $\ell_1$-measure with respect to $\mu$. (This is why anti-concentration with respect to $\mu$ 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 $f$.

[4] 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 6. We define both notions and establish the necessary relations between them in Section 2.

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 "$\mu$-distribution" Fourier coefficients, because we need to ultimately relate the Shapley values to anti-concentration with respect to $\mu$.

**Organization.** Because of space constraints most proofs are deferred to the full version. In Section 2 we define the distribution $\mu$ 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 6, which says that anti-concentration of affine forms and closeness in Fourier coefficients together suffice to establish closeness in $\ell_1$ distance. Section 3 proves that "nice" affine forms have the required anti-concentration, and Section 4 describes the algorithmic tool from [TTV08] that lets us establish closeness of coordinate correlation coefficients. Section 5 puts the pieces together to prove our main theorems.

## 2 Reformulation of Shapley-Shubik Indices

Given $f : \{-1,1\}^n \to \{-1,1\}$, we will denote by $\tilde{f}(i)$ the *$i$-th* Shapley value of $f$. The original definition of Shapley values 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 $\mu$ that plays a central role in our reformulations. We start by defining this distribution $\mu$ and introducing some relevant notation, and then give our results. Because of space constraints all proofs are deferred to the full version.

**The distribution $\mu$.** Let us define $\Lambda(n) := \sum_{0<k<n} \frac{1}{k} + \frac{1}{n-k}$; clearly we have $\Lambda(n) = \Theta(\log n)$, and more precisely we have $\Lambda(n) \leq 2\log n$. We also define $Q(n,k)$ as $Q(n,k) := \frac{1}{k} + \frac{1}{n-k}$ for $0 < k < n$, so we have $\Lambda(n) = Q(n,1) + \cdots + Q(n,n-1)$.

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

The distribution $\mu$ is supported on $B_n$ and is defined as follows: to make a draw from $\mu$, sample $k \in \{1,\ldots,n-1\}$ 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 $\{-1,1\}^n_{=k} := \{x \in \{-1,1\}^n : \mathrm{wt}(x) = k\}$.

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

$$f^*(i) := \mathbf{E}_{x\sim\mu}[f(x) \cdot x_i] \tag{1}$$

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

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

$$\hat{f}(i) := \mathbf{E}_{x\sim\mu}[f(x) \cdot L_i(x)]. \tag{2}$$

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

**Lemma 1.** *Given $f : \{-1,1\}^n \to [-1,1]$, for each $i = 1,\ldots,n$ we have $\tilde{f}(i) = \frac{f(\mathbf{1})-f(-\mathbf{1})}{n} + \frac{\Lambda(n)}{2} \cdot \left(f^*(i) - \frac{1}{n}\sum_{j=1}^n f^*(j)\right)$.*

**Construction of a Fourier basis for distribution $\mu$.** For all $x \in B_n$ we have that $\mu(x) > 0$, and consequently 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 $L_0, \ldots, L_n$ for this subspace, i.e. one that satisfies $\langle L_i, L_i \rangle_\mu = 1$ 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 $\alpha$ and $\beta$. The following Lemma gives the values of $\alpha$ and $\beta$:

**Lemma 2.** *For the choices* $\alpha = \frac{1}{n} \cdot \left( \sqrt{\frac{\Lambda(n)}{n\Lambda(n) - 4(n-1)}} - \frac{\sqrt{\Lambda(n)}}{2} \right)$, $\beta = \frac{\sqrt{\Lambda(n)}}{2}$, *the set* $\{L_i\}_{i=0}^n$ *is an orthonormal set of linear functions under the distribution* $\mu$.

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

**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 3.** *Let* $f : \{-1,1\}^n \to [-1,1]$ *be any function. Then for each* $i = 1, \ldots, n$ *we have* $\hat{f}(i) = \frac{2\beta}{\Lambda(n)} \cdot \left( \tilde{f}(i) - \frac{f(\mathbf{1}) - f(-\mathbf{1})}{n} \right) + \frac{1}{n} \cdot \sum_{j=1}^n \hat{f}(j)$.

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

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

**Lemma 4.** *Let* $f, g : \{-1,1\}^n \to [-1,1]$. *Then,* $d_{\mathrm{Shapley}}(f,g) \leq \frac{4}{\sqrt{n}} + \frac{\Lambda(n)}{2\beta} \cdot d_{\mathrm{Fourier}}(f,g)$.

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

**Lemma 5.** *Let* $f, g : \{-1,1\}^n \to \mathbb{R}$. *Then we have* $d_{\mathrm{Fourier}}(f,g) \leq O(\sqrt{\log n}) \cdot \sqrt{\sum_{i=0}^n (f^*(i) - g^*(i))^2}$.

**From Fourier closeness to $\ell_1$-closeness.** An important technical ingredient in our work is the notion of an affine form $\ell(x)$ having "good anti-concentration" under distribution $\mu$; we now give a precise definition to capture this.

**Definition 1 (Anti-concentration).** *Fix* $w \in \mathbb{R}^n$ *and* $\theta \in \mathbb{R}$, *and let the affine form* $\ell(x)$ *be* $\ell(x) := w \cdot x - \theta$. *We say that* $\ell(x)$ *is* $(\delta, \kappa)$-anti-concentrated under $\mu$ *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 $f = \mathrm{sign}(w \cdot x - \theta)$, 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 $d_{\mathrm{Fourier}}(f,g)$ small must in fact be close to $f$ in $\ell_1$ distance under $\mu$.

**Lemma 6.** *Let $f : \{-1,1\}^n \to \{-1,1\}$, $f = \mathrm{sign}(w \cdot x - \theta)$ be such that $w \cdot x - \theta$ is $(\delta,\kappa)$-anti-concentrated under $\mu$ (for some $\kappa \leq 1/2$), where $|\theta| \leq \|w\|_1$. Let $g : \{-1,1\}^n \to [-1,1]$ be such that $d_{\mathrm{Fourier}}(f,g) \leq \rho$. Then we have*

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

## 3 A Useful Anti-concentration Result

In this section we prove an anti-concentration result for monotone increasing $\eta$-reasonable affine forms under the distribution $\mu$. 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 5.

**Theorem 1.** *Let $L(x) = w_0 + \sum_{i=1}^n w_i x_i$ be a monotone increasing $\eta$-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 $\mu$, 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 $\mu$ is not even a pairwise independent distribution, so standard approaches for proving anti-concentration cannot be directly applied. Instead, we exploit the fact that $\mu$ 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 proof is given in the full version.

## 4 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 2.** *[TTV08] Let $X$ be a finite domain, $\mu$ 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 $w_\ell$'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 $v_\ell$'s are integers whose absolute values sum to $O(1/\xi^2)$.*

We note that Theorem 2 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 $a_\ell$ 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 full version we state the algorithm in full and sketch a proof of correctness of this algorithm using results that are explicitly proved in [TTV08].

## 5 Our Main Results

In this section we combine ingredients from the previous subsections and prove our main results, Theorems 3 and 4.

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

**Theorem 3.** *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 $\tilde{a}(1), \ldots, \tilde{a}(n)$; 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 $\eta$-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 $(\tilde{a}(1), \ldots, \tilde{a}(n))$, 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 $\delta$.*

*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 $\epsilon, \delta$ and $\tilde{a}(1), \ldots, \tilde{a}(n)$ that satisfy $d_{\mathrm{Shapley}}(a, f) \leq 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$ for some monotone increasing $\eta$-reasonable LTF $f$. The idealized version of the problem is the following: we assume that the algorithm is also given the two real values $f^*(0)$, $(f^*(1) + \ldots + f^*(n))/n$. It is also helpful to note that since $f$ is monotone and $\eta$-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 1, 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; $\mu$ is the distribution; and the parameter $\xi$ 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; details in the full version.) 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 2, 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 5 implies that $d_{\mathrm{Fourier}}(f, h) \leq \rho := 1/\mathrm{poly}(n, 2^{\mathrm{poly}(1/\epsilon)})$.

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 6 and thereby assert that the $\ell_1$ distance between $f$ and $h$ (with respect to $\mu$) is small. To see that we can do this, we first claim (see full version for details) that since $f$ is a monotone increasing $\eta$-reasonable LTF, it has a representation as $f(x) = \mathrm{sign}(w \cdot x + w_0)$ whose weights satisfy the following property: for any choice of $\zeta > 0$, after rescaling all the weights, the largest-magnitude weight has magnitude 1, and the $k := \Theta_{\zeta,\eta}(1/\epsilon^{6+2\zeta})$ largest-magnitude weights each have magnitude at least $r := 1/(n \cdot k^{O(k)})$. (Note that since $\epsilon \geq n^{-c}$ we indeed have $k \leq n$ as required.) Given this, Theorem 1 implies that the affine form $L(x) = w \cdot x + w_0$ satisfies

$$\mathbf{Pr}_{x\sim\mu}[|L(x)| < r] \leq \kappa := \epsilon^2/(1024 \log(n)), \tag{3}$$

i.e. it is $(r, \kappa)$-anticoncentrated with $\kappa = \epsilon^2/(1024 \log(n))$. Thus we may indeed apply Lemma 6, and it gives us that

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

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 $\mu$, 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 $\mathbf{Pr}_{x\sim\mu}[f(x) \neq h'(x)] \leq \epsilon^2/(64 \log n)$. By a standard argument, we obtain that $d_{\mathrm{Fourier}}(f, h') \leq \epsilon/(4\sqrt{\log n})$. Finally, Lemma 4 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 3, where we are not given the values of $f^*(0)$ and $(f^*(1) + \ldots + f^*(n))/n$. To get around this, we note that $f^*(0)$, $(f^*(1) + \ldots + f^*(n))/n \in [-1, 1]$. 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 $(f^*(1) + \ldots + f^*(n))/n$. At the end of each such run we have a "candidate" LTF $h'$; we use a simple procedure Shapley-Estimate 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 $(f^*(1) + \ldots + f^*(n))/n$ 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 3 is proved. □

For any monotone $\eta$-reasonable target LTF $f$, Theorem 3 constructs an output LTF whose Shapley distance from $f$ is at most $\epsilon$, but the running time is exponential in $\mathrm{poly}(1/\epsilon)$. We now show that if the target monotone $\eta$-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 3 for such small $\epsilon$. (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 4.** *There is an algorithm ISBW (for Inverse-Shapley with Bounded Weights) with the following properties. ISBW is given as input a weight bound $W \in \mathbf{N}$, a confidence parameter $\delta > 0$, and $n$ real values $\tilde{a}(1), \ldots, \tilde{a}(n)$; 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 $\eta$-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 $(\tilde{a}(1), \ldots, \tilde{a}(n))$, the probability that IS outputs $v, \theta$ such that the LTF $h(x) = \mathrm{sign}(v \cdot x - \theta)$ has $d_{\mathrm{Shapley}}(f, h) > n^{-1/8}$ is at most $\delta$.*

*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 $\theta$ 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 1 immediately yields the following corollary:

**Corollary 1.** *Let $L(x) = \sum_{i=1}^{n} u_i x_i - \theta$ be a monotone increasing $\eta$-reasonable affine form. Suppose that $u_i \geq r$ for all $i = 1, \ldots, n$. 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 4 closely follows the proof of Theorem 3. The algorithm runs Boosting-TTV with $\mathcal{L}$, $a^*(i)$ and $\mu$ as before but now with $\xi$ set to $1/\mathrm{poly}(n, W)$. The LBF $h$ that Boosting-TTV outputs satisfies $d_{\mathrm{Fourier}}(f, h) \leq \rho := 1/\mathrm{poly}(n, W)$. We apply Corollary 1 to the affine form $L(x) := \frac{u}{\|u\|_1} \cdot x - \frac{\theta}{\|u\|_1}$ and get that for $r = 1/\mathrm{poly}(n, W)$, we have

$$\mathbf{Pr}_{x \sim \mu}[|L(x)| < r] \leq \kappa := \epsilon^2/(1024 \log n) \tag{5}$$

where now $\epsilon := n^{-1/8}$, in place of Equation (3). Applying Lemma 6 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 (4). 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. □

**Acknowledgement.** We thank Christos Papadimitriou for helpful conversations.

## References

[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. To appear in *STOC*, 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.

[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.

[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.

[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.

[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.

[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*, pages 215–220, 2008.