On the Complexity of the Inverse Semivalue Problem for Weighted Voting Games

Ilias Diakonikolas
University of Southern California
USA
ilias.diakonikolas@gmail.com

Chrystalla Pavlou
University of Edinburgh
UK
C.Pavlou@sms.ed.ac.uk

Abstract

Weighted voting games are a family of cooperative games, typically used to model voting situations where a number of agents (players) vote against or for a proposal. In such games, a proposal is accepted if an appropriately weighted sum of the votes exceeds a prespecified threshold. As the influence of a player over the voting outcome is not in general proportional to her assigned weight, various power indices have been proposed to measure each player's influence. The inverse power index problem is the problem of designing a weighted voting game that achieves a set of target influences according to a predefined power index. In this work, we study the computational complexity of the inverse problem when the power index belongs to the class of semivalues. We prove that the inverse problem is computationally intractable for a broad family of semivalues, including all regular semivalues. As a special case of our general result, we establish computational hardness of the inverse problem for the Banzhaf indices and the Shapley values, arguably the most popular power indices.

Introduction

Background and Motivation Weighted voting games are a classical family of cooperative games that have been extensively studied in the game theory and social choice literature. Such games model a common voting scenario where each agent (player), associated with a weight, casts a "YES" (for) or "NO" (against) vote: if the weighted sum of the "YES" votes exceeds a threshold, then the voting outcome is "YES", otherwise the outcome is "NO". Examples of such practical scenarios are the voting system of the European Union, stockholder companies and resource allocation in multiagent systems (Elkind et al. 2008; De Keijzer, Klos, and Zhang 2014).

Although having a larger weight might help an agent affect the voting outcome, her influence on the result of the game is not always proportional to her weight. Thus, instead of using agents' weights, the power of an agent over the outcome is usually measured in a systematic way by a power index. Over the years, many power indices have been proposed and studied, such as the Shapley value (Shapley 1953) (known also as Shapley-Shubik index for weighted voting games (Shapley and Shubik 1954)), the Banzhaf index (Banzhaf III 1964), the Deegan-Packel index (Deegan and Packel 1978), and the Holler index (Holler 1982). The problem of computing the agents' power indices in a given game has received ample attention and its computational complexity is well-understood for many game representations and power index functions (Prasad and Kelly 1990; Deng and Papadimitriou 1994; Aziz 2008).

Our Contributions In this work, we focus on the inverse power index problem — that is, the problem of *designing* a weighted voting game with a given set of power indices. As we will explain in detail below, the inverse problem has been extensively studied in various fields, including game theory, social choice theory, and learning theory. Various works have provided heuristic methods, exponential time algorithms, or polynomial time approximation algorithms with provable performance guarantees for this problem.

Despite this wealth of prior work on the algorithmic version of the inverse problem, its computational complexity is poorly understood, even for the most popular power indices (Shapley values, Banzhaf indices). *In this paper, we study and essentially resolve the computational complexity of the inverse power index for weighted voting games with respect to a broad and extensively studied family of power indices. Specifically, we prove that the inverse problem is computationally intractable, under standard complexity assumptions.* More specifically, we prove that for a large class of power indices — that includes the popular Banzhaf and Shapley indices, the class of semivalues (Weber 1979) — the inverse problem cannot be in the polynomial hierarchy (PH), unless the polynomial hierarchy collapses. Prior to this work, it was conceivable that there exists an (exact) polynomial time algorithm for this problem. It follows from our hardness result that the existence of such an algorithm is unlikely.

Related work Several heuristic algorithms for the inverse Banzhaf index problem have been proposed in the social choice and game theory literature. Aziz, Paterson, and Leech (2007) give an approximation algorithm that given target Banzhaf indices and a desired ℓ_2-distance outputs a weighted voting game with integer weights that has Banzhaf indices within the desired distance. Unfortunately, no theoretical guarantees are given about the convergence rate or the approximation error of this method, and it is not known whether it converges to the optimal solution. Two related heuristic algorithms that return weighted voting games with Banzhaf indices within a given distance from the target indices are given by Laruelle and Widgrén (1998) and Leech (2003). Similarly to Aziz, Paterson, and Leech (2007), Fatima, Wooldridge, and Jennings (2008) give an iterative approximation algorithm for the inverse Shapley value problem that given target Shapley values, a quota and a desired average percentage difference, outputs a weighted voting game with the given quota that has Shapley values within the desired distance. It is shown that each iteration runs in quadratic time and that the algorithm eventually converges, but no theoretical guarantees are given regarding the approximation error and the convergence rate.

An exact algorithm for both the inverse Banzhaf index and inverse Shapley value problems is given by Kurz (2012). The proposed method relies on integer linear programming and returns a weighted voting game that minimizes the ℓ_1-distance from the target power indices. Unfortunately, this method has running time exponential in the number of players. Another exact but exponential-time algorithm for the inverse power index problem is given by De Keijzer, Klos, and Zhang (2014). The proposed algorithm outputs a weighted voting game where the power indices of players are as close to the target vector as possible. The presented algorithm is based on an enumeration of all weighted voting games: for each weighted voting game, the algorithm computes the power indices, their distance from the target ones and it then outputs the game with the smallest distance. Since there exist $2^{\Omega(n^2)}$ weighted voting games with n players, this algorithm also runs in exponential time.

In addition to the several heuristics and exponential time algorithms that have been proposed, recent works (De et al. 2014; De, Diakonikolas, and Servedio 2017) have obtained polynomial time approximation schemes with provable guarantees for the inverse problem with respect to both the Banzhaf indices and the Shapley values. These algorithms output a weighted voting game whose power indices have small ℓ_2-distance from the target indices.

The inverse power index problem is also important in various other fields, such as circuit complexity and computational learning theory, see (O'Donnell and Servedio 2011) and references therein. In these fields, linear threshold functions, which are equivalent to weighted voting games if we allow negative weights (De Keijzer, Klos, and Zhang 2014), have been of great significance. An important result of the 60s (Chow 1961), shows that linear threshold functions are characterized by the degree-0 and degree-1 "Fourier coefficients", known as Chow parameters. Given this structural result, a natural question, known as "the Chow parameters problem" arises: given the Chow parameters of a linear threshold function, reconstruct a weights-based representation of the function. Interestingly, the Chow parameters are exactly the non-normalized Banzhaf indices (Dubey and Shapley 1979) and therefore, the inverse Banzhaf index problem is closely related to the Chow parameters problem.

Gopalan, Nisan, and Roughgarden (2015) study the convex polytope consisting of the Chow parameters of all Boolean functions. They show that the linear optimization problem over the Chow parameters polytope is #P-hard; a result that indicates, but does not logically imply, that the Chow parameters problem is intractable. Aziz (2008), Elkind et al. (2009) and Elkind et al. (2008) study the computational complexity of problems related to weighted voting games. Aziz (2008) studies the complexity of computing various indices such as the Shapley values, the Banzhaf and the Deegan-Packel indices for a given simple game when the game is given in different forms. Finally, Faliszewski and Hemaspaandra (2008) study the complexity of the power index *comparison problem*: given two weighted voting games and a player, decide on which game the given player has higher influence as it is computed by a specific power index. They show that this problem is intractable, namely PP-complete, for both the Shapley values and Banzhaf index. Along this, they extend the #P-metric-completeness of computing the Shapley values, proved by Deng and Papadimitriou (1994). They prove that, whereas computing the Banzhaf indices of a weighted voting game is #P-parsimonious-complete (Prasad and Kelly 1990), computing the Shapley values is #P-many-one complete and it cannot be strengthened to #P-parsimonious-complete.

Preliminaries

Notation We write $wt(x)$ to denote the *weight* of a Boolean vector $x \in \{-1,1\}^n$, i.e., the number of 1's in x. We denote by **1** (resp. **-1**) the vector in $\{-1,1\}^n$ with all coordinates equal to 1 (resp. -1). We will use sign : $\mathbb{R} \to \mathbb{R}$ for the function that takes value 1 if $z \geq 0$ and value -1 if $z < 0$.

Our basic object of study is the family of linear threshold functions over $\{-1,1\}^n$:

Definition 1 (Linear Threshold Function)**.** *A* linear threshold function (LTF) *is any function* $f_{w,\theta} : \{-1,1\}^n \to \{-1,1\}$ *such that* $f_{w,\theta}(x) = \text{sign}(w \cdot x - \theta)$ *for some weight vector* $w \in \mathbb{R}^n$ *and threshold* $\theta \in \mathbb{R}$.

Note that weighted voting games are equivalent to LTFs with *non-negative weights*. We leverage this equivalence throughout this paper.

Semivalues We focus on power indices that belong to *the class of semivalues*. Semivalues are a fundamental family of power indices, introduced by Weber (1979) as generalizations of the Shapley value that do not satisfy the efficiency axiom (Dubey, Neyman, and Weber 1981). Since their introduction, semivalues have received considerable attention, see, e.g., (Einy 1987; Carreras, Freixas, and Puente 2003; Carreras and Freixas 2008).

We give the definition of semivalues (Shapley and Roth 1988) in terms of weighting coefficients, as they were characterized by Dubey, Neyman, and Weber (1981):

Definition 2 (Semivalues)**.** *For a positive integer* n*, let* $p^n = (p_0^n, \ldots, p_{n-1}^n)$ *be a vector such that* $\sum_{t=0}^{n-1} \binom{n-1}{t} p_t^n = 1$ *and* $p_t^n \geq 0$*, for* $t \in \{0, \ldots, n-1\}$*. Then, the i-th semivalue* *corresponding to the probability vector p^n of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is the value*

$$\widetilde{f}^{p^n}(i) = \sum_{x\in\{-1,1\}^n : x_i=-1} p^n_{wt(x)} f(x) x_i + \sum_{x\in\{-1,1\}^n : x_i=1} p^n_{wt(x)-1} f(x) x_i \ ,$$

for $i \in \{1, \dots, n\}$.

Intuitively, if we interpret p^n as the vector of probabilities that a given player will join a coalition of size t (Carreras and Freixas 2008), $0 \le t \le n-1$, the i-th semivalue computes the probability of the event that player i is a *pivot*, i.e., the probability that the output of the game would change from 1 to -1 if the i-th player (the i-th variable) were to change her vote from 1 to -1.

We note that the Shapley values and Banzhaf indices are the semivalues defined by $p^n_t = \frac{(n-t-1)!t!}{n!}$ (Shapley and Shubik 1954) and $p^n_t = \frac{1}{2^{n-1}}$ (Dubey and Shapley 1979), respectively. Both indices are regular semivalues, i.e., semivalues that are defined by strictly positive probability vectors (Carreras and Freixas 2008).

Reformulation of Semivalues The following equivalent way to express semivalues will be useful throughout this paper. Setting $p^n_{-1} = p^n_n = 0$, we observe that we can rewrite the semivalues vector as follows: for $i \in \{1, \dots, n\}$,

$$\widetilde{f}^{p^n}(i) = \frac{1}{2} \sum_{x\in\{-1,1\}^n} f(x) x_i (p^n_{wt(x)} + p^n_{wt(x)-1}) + \frac{1}{2} \sum_{x\in\{-1,1\}^n} f(x) (p^n_{wt(x)-1} - p^n_{wt(x)}) \ . \quad (1)$$

From this representation, a probability distribution μ_{p^n} over $\{-1,1\}^n$ emerges: for $x \in \{-1,1\}^n$,

$$\mu_{p^n}(x) := \frac{\mu'_{p^n}(x)}{\Lambda(p^n)} \ ,$$

where $\mu'_{p^n}(x) := p^n_{wt(x)} + p^n_{wt(x)-1}$ and $\Lambda(p^n) = \sum_{x\in\{-1,1\}^n} \mu'_{p^n}(x)$ is the normalizing factor.

For a function $f : \{-1,1\}^n \to \{-1,1\}$, we write

$$\widehat{f}^{p^n}(i) = \sum_{x\in\{-1,1\}^n} \mu'_{p^n}(x) f(x) x_i$$

for the first term of (1) and

$$C^{p^n}_f = \sum_{x\in\{-1,1\}^n} f(x) (p^n_{wt(x)-1} - p^n_{wt(x)})$$

for the second term of (1). One can view the first term, $\widehat{f}^{p^n}(i)$, as the expectation $\mathbb{E}_{x\sim\mu_{p^n}}[f(x)x_i]$ (up to the normalizing factor $\Lambda(p^n)$).

We now define the notion of a *reasonable* probability vector to describe the family of semivalues for which our computational hardness results apply:

Definition 3 (Reasonable Probability Vector)**.** *A probability vector $p^n \in \mathbb{R}^n$ is called* reasonable *if there exists a t with $t = \Omega(n)$ and $n - t = \Omega(n)$ such that $p^n_t > 0$.*

The intuition behind the above definition is that the distribution μ_{p^n} has support $2^{\Omega(n)}$. Note that this happens when there exists a t with $\binom{n}{t} = 2^{\Omega(n)}$ such that $p^n_t > 0$.

We note that all regular semivalues (including the Banzhaf indices and Shapley values) satisfy the above property.

Remark 1. *For computational purposes, throughout this paper, we will assume that for any $t \in \{0, \dots, n-1\}$, the probability p^n_t can be described in $\mathrm{poly}(n)$ bits.*

Inverse Semivalues Problem We are ready to define the inverse semivalues problem. Given a vector of target semivalues, we want to either find a weighted voting game with these pre-specified semivalues or decide that there does not exist any weighted voting game with the target semivalues.

Name: SV_{p^n}-Inverse Problem
Input: A vector $(c_1, \dots, c_n) \in \mathbb{Q}^n$ and $\theta \in \mathbb{Q}$.
Question: Output $w \in \mathbb{Q}^n_+$ with $\sum_{i=1}^n w_i = 1$ such that $\widetilde{f}^{p^n}_{w,\theta}(i) = c_i$, for $i \in \{1, \dots, n\}$, or "NO" if no such w exists.

Main Result: Computational Intractability of Inverse Power-Index Problem

The main result of this paper is the following:

Theorem 1 (Main)**.** *For semivalues defined by the probability vector p^n, if p^n is a* reasonable *probability vector, then the SV_{p^n}-Inverse problem is not in the polynomial hierarchy, unless the polynomial hierarchy collapses.*

As an immediate corollary of Theorem 1, we obtain that the inverse power index problem is intractable for the class of regular semivalues, which includes the Shapley values and the Banzhaf indices.

Proof Overview To prove hardness of the inverse problem, we examine the convex polytope $\mathcal{C}_{p^{n+2}}$ consisting of the convex combinations of the semivalues of linear threshold functions with 0-threshold and weight vectors of a specific form described below (Definition 4). We prove (Theorem 2) that if the probability distribution defining the semivalues is a *reasonable* distribution, then the linear optimization over $\mathcal{C}_{p^{n+2}}$ is #P-hard (under Turing reductions). Then, we show (Theorem 4) that the optimization problem can be solved using an oracle for the semivalues *verification problem*, i.e., the problem of verifying that the given semivalues are the true semivalues of a given linear threshold function, or an oracle for the *inverse problem* for weight vectors of the aforementioned specific form. We thus conclude that the verification and the inverse semivalues problem for linear threshold functions with specific weight structure cannot be in the polynomial hierarchy. Finally, using a lemma that shows that semivalues characterize the space of linear threshold functions with the same threshold (Lemma 3), we show hardness of the inverse and verification problems for linear threshold functions with positive weights, i.e., for weighted voting games.

Our proof strategy bears some similarities to the approach used by Gopalan, Nisan, and Roughgarden (2015) (also exploited by Dughmi and Xu (2016)). Gopalan, Nisan, and Roughgarden (2015) show that the linear optimization problem over the polytope consisting of the Chow parameters of all Boolean functions is #P-complete and therefore, there cannot exist an efficient membership oracle for this polytope. We note that our results include the results of (Gopalan, Nisan, and Roughgarden 2015) regarding Chow parameters as a very special case: as previously mentioned, the non-normalized Banzhaf indices of a linear threshold function are equal to its Chow parameters and they are semivalues.

Despite this similarity, our proof involves a number of new ideas that seem necessary in order to handle a broad range of probability distributions that could define a semivalue. One of the difficulties comes from the fact that we want to prove hardness for the class of weighted voting games, i.e., LTFs with *positive* weights. While this requirement is easy to handle for the Banzhaf indices (Chow parameters), it poses non-trivial difficulties for more general semivalues. In particular, we propose a generalization of the definition of the Khintchine constant from the uniform distribution to any probability distribution and establish that it is hard to compute under a restricted set of weights that is crucial for our proof (Theorem 3). Another crucial ingredient of our proof is a new structural result (Lemma 3) establishing that the set of semivalues uniquely determines a weighted voting game.

Proof of Main Result

In this subsection, we proceed with the detailed proof of Theorem 1.

Semivalues Polytope Our analysis makes essential use of the convex polytope defined as the convex hull of the set of semivalues for all linear threshold functions whose weights-based representation is of a specific form: Namely, their threshold $\theta = 0$ and their weight vectors consist of n positive coordinates and two coordinates each of whose weights is equal to minus a half times the sum of the first n coordinates. Formally, we introduce the following definition:

Definition 4. *For a positive integer n, define $\mathcal{C}_{p^{n+2}} := Conv(\mathcal{A}_{p^{n+2}})$, where $\mathcal{A}_{p^{n+2}} = \{(c_1, \ldots, c_{n+2}) \in \mathbb{Q}^{n+2} : \exists w \in \mathbb{Q}^{n+2},\ w_i > 0,\ 1 \le i \le n,\ w_{n+1} = w_{n+2} = -\sum_{i=1}^{n} w_i/2,\ c_i = \widetilde{f}_{w,0}^{p^{n+2}}(i),\ 1 \le i \le n+2\}$.*

Linear Optimization over $\mathcal{C}_{p^{n+2}}$ We firstly prove that if the semivalues' probability distribution defined by p^{n+2} has a sufficiently large support, then the linear optimization problem over the polytope $\mathcal{C}_{p^{n+2}}$ is #P-hard. The linear optimization problem for semivalues defined by any probability vector p^{n+2} is captured by the following family of problems:

Name: $\text{SV}_{p^{n+2}}$-Optimization Problem
Input: A vector $a = (a_1, \ldots, a_{n+2}) \in \mathbb{Q}^{n+2}$.
Question: Compute $\max_{c \in \mathcal{C}_{p^{n+2}}} a \cdot c$.

The main result of this subsection is the following:

Theorem 2. *If p^{n+2} is a* reasonable *probability vector, the $\text{SV}_{p^{n+2}}$-Optimization Problem is #P-hard.*

We prove Theorem 2 by reducing from an intermediate problem — that of computing the Khintchine constant of a vector *with respect to the probability distribution μ_{p^n}*:

Name: Khintchine-μ_{p^n}
Input: A vector $a = (a_1, \ldots, a_n) \in \mathbb{Q}^n$.
Question: Compute $K_{\mu_{p^n}}(a) = \mathbb{E}_{x \sim \mu_{p^n}}[|a \cdot x|]$.

Theorem 3. *If p^{n+2} is a* reasonable *probability vector, the Khintchine-$\mu_{p^{n+2}}$ problem is #P-hard, even restricted to inputs $(a_1, \ldots, a_n, -A/2, -A/2)$, where $A = \sum_{i=1}^{n} a_i$ and $a_i > 0$ for $i \in \{1, \ldots, n\}$.*

Proof. We start by showing that the #Partition problem for the distribution μ_{p^n} is hard and then reduce the latter problem to the former.

Name: #Partition-μ_{p^n}
Input: A vector $w = (w_1, \ldots, w_n) \in \mathbb{Z}^n$.
Question: Compute $\Pr_{x \sim \mu_{p^n}}[w \cdot x = 0]$.

We start from the following proposition, whose proof is omitted due to space limitations and is available in the full version of this paper.

Proposition 1. *If p^{n+2} is a* reasonable *probability vector, #Partition-$\mu_{p^{n+2}}$ is #P-hard, even restricted to inputs $(w_1, \ldots, w_n, -W/2, -W/2)$, where $W = \sum_{i=1}^{n} w_i$ and $w_i > 0$ for $1 \le i \le n$.*

Given an instance of #Partition-$\mu_{p^{n+2}}$, i.e., $a = (a_1, \ldots, a_n, -A/2, -A/2)$, where $A = \sum_{i=1}^{n} a_i$ and $a_i > 0$ for $1 \le i \le n$, we construct the following three Khintchine-$\mu_{p^{n+2}}$ instances:

$$c = 2(a_1, a_2, \ldots, a_n, -A/2, -A/2),$$
$$d = (a_1 - y, a_2, \ldots, a_n, -A/2 + y/2, -A/2 + y/2),$$
$$e = (a_1 + y, a_2, \ldots, a_n, -A/2 - y/2, -A/2 - y/2),$$

where $0 < y < 1/2$.
For any $x \in \{-1, 1\}^{n+2}$:

$$|d \cdot x| = \begin{cases} |\sum_{i=1}^{n} a_i x_i + A - 2y| & x_1 = 1, \\ & x_{n+1} = x_{n+2} = -1 \\ |\sum_{i=1}^{n} a_i x_i + A| & x_1 = -1, \\ & x_{n+1} = x_{n+2} = -1 \\ |\sum_{i=1}^{n} a_i x_i - A| & x_1 = 1, \\ & x_{n+1} = x_{n+2} = 1 \\ |\sum_{i=1}^{n} a_i x_i - A + 2y| & x_1 = -1 \\ & x_{n+1} = x_{n+2} = 1 \\ |\sum_{i=1}^{n} a_i x_i - y| & x_1 = 1, \\ & x_{n+1} \neq x_{n+2} \\ |\sum_{i=1}^{n} a_i x_i + y| & x_1 = -1, \\ & x_{n+1} \neq x_{n+2} \end{cases}$$

$$|e \cdot x| = \begin{cases} |\sum_{i=1}^n a_i x_i + A + 2y| & x_1 = 1, \\ & x_{n+1} = x_{n+2} = -1 \\ |\sum_{i=1}^n a_i x_i + A| & x_1 = -1, \\ & x_{n+1} = x_{n+2} = -1 \\ |\sum_{i=1}^n a_i x_i - A| & x_1 = 1, \\ & x_{n+1} = x_{n+2} = 1 \\ |\sum_{i=1}^n a_i x_i - A - 2y| & x_1 = -1, \\ & x_{n+1} = x_{n+2} = 1 \\ |\sum_{i=1}^n a_i x_i + y| & x_1 = 1, \\ & x_{n+1} \neq x_{n+2} \\ |\sum_{i=1}^n a_i x_i - y| & x_1 = -1, \\ & x_{n+1} \neq x_{n+2} \end{cases}$$

We observe that the following hold:
For $x \in \{-1,1\}^{n+2}$, $wt(x) \in \{0, n+2\}$,

$$|c \cdot x| = |e \cdot x| = |d \cdot x| = 0$$

For $x \in \{-1,1\}^{n+2}$, $x_{n+1} = x_{n+2} = x_1$

$$|d \cdot x| + |e \cdot x| = |c \cdot x|$$

For $x \in \{-1,1\}^{n+2}$, $x_{n+1} \neq x_{n+2}$, if $c \cdot x \neq 0$,

$$|d \cdot x| + |e \cdot x| = |c \cdot x|$$

as it holds that

$$|\sum_{i=1}^n a_i x_i - y| + |\sum_{i=1}^n a_i x_i + y| = 2\max(|\sum_{i=1}^n a_i x_i|, |y|)$$

and $|\sum_{i=1}^n a_i x_i| \geq 1$. When $c \cdot x = 0$, $|d \cdot x| + |e \cdot x| = 2y$.

For $x \in \{-1,1\}^{n+2}$, $x_{n+1} = x_{n+2}$, $x_1 \neq x_{n+1}$

$$|d \cdot x| + |e \cdot x| = |c \cdot x|$$

as similarly with the above case, we have that

$$|\sum_{i=1}^{n+2} a_i x_i - 2y| + |\sum_{i=1}^{n+2} a_i x_i + 2y| = 2\max(|\sum_{i=1}^{n+2} a_i x_i|, |2y|)$$

and $|\sum_{i=1}^{n+2} a_i x_i| \geq 1$.

Hence, we have:

$$\begin{aligned} & K_{\mu_{p^{n+2}}}(d) + K_{\mu_{p^{n+2}}}(e) - K_{\mu_{p^{n+2}}}(c) \\ &= \mathbb{E}_{x\sim\mu_{p^{n+2}}}[|d \cdot x|] + \mathbb{E}_{x\sim\mu_{p^{n+2}}}[|e \cdot x|] - \mathbb{E}_{x\sim\mu_{p^{n+2}}}[|c \cdot x|] \\ &= \sum_{x\in\{-1,1\}^{n+2}} 2y\mu_{p^{n+2}}(x)\mathbb{1}_{c\cdot x=0, x\notin\{\mathbf{-1},\mathbf{1}\}} . \end{aligned}$$

So, we get

$$\begin{aligned} & \Pr_{x\sim\mu_{p^{n+2}}}[a \cdot x = 0] \\ &= \frac{K_{\mu_{p^{n+2}}}(d) + K_{\mu_{p^{n+2}}}(e) - K_{\mu_{p^{n+2}}}(c)}{2y} \\ &+ \Pr_{x\sim\mu_{p^{n+2}}}[x = \mathbf{-1}] + \Pr_{x\sim\mu_{p^{n+2}}}[x = \mathbf{1}] \\ &= \frac{K_{\mu_{p^{n+2}}}(d) + K_{\mu_{p^{n+2}}}(e) - K_{\mu_{p^{n+2}}}(c)}{2y} \\ &+ \frac{p_0 + p_{n+1}}{\Lambda(p^{n+2})} . \end{aligned}$$

□

We are now ready to prove Theorem 2.

Proof of Theorem 2. We reduce from the Khintchine-$\mu_{p^{n+2}}$ problem: given input $a = (a_1, \ldots, a_n, a_{n+1} = -A/2, a_{n+2} = -A/2) \in \mathbb{Q}^{n+2}$, where $A = \sum_{i=1}^n a_i$ and $a_i > 0$ for $0 \leq i \leq n$, we compute $\max_{c\in\mathcal{C}_{p^{n+2}}} a \cdot c$. For any c in $C_{p^{n+2}}$ we have:

$$\begin{aligned} a \cdot c &= \frac{1}{2}\sum_{i=1}^{n+2} a_i \widehat{f}^{p^{n+2}}(i) + \frac{1}{2}\sum_{i=1}^{n+2} a_i C_f^{p^{n+2}} \\ &= \frac{\Lambda(p^{n+2})}{2} \sum_{x\in\{-1,1\}^{n+2}} f(x)\mu_{p^{n+2}}(x) \sum_{i=1}^{n+2} a_i x_i \\ &\leq \frac{\Lambda(p^{n+2})}{2} \sum_{x\in\{-1,1\}^{n+2}} \mu_{p^{n+2}}(x) |\sum_{i=1}^{n+2} a_i x_i| , \end{aligned}$$

where f is a linear threshold function.
Reducing from a vector that has sum 0 was essential for this step: the term $C_f^{p^{n+2}}$ vanishes and we end up with the term $\sum_{x\in\{-1,1\}^{n+2}} f(x)\mu_{p^{n+2}}(x) \sum_{i=1}^{n+2} a_i x_i$ that is upper bounded by $\sum_{x\in\{-1,1\}^{n+2}} \mu_{p^{n+2}}(x) |\sum_{i=1}^{n+2} a_i x_i|$.

This upper bound is tight, as we show below, which is crucial for our argument. If one were to reduce from a vector with sum different than 0 or include in the polytope only linear threshold functions with positive weights, it would not have been possible to obtain a tight upper bound.

Observe that

$$a \cdot c = \frac{\Lambda(p^{n+2})}{2} \sum_{x\in\{-1,1\}^{n+2}} \mu_{p^{n+2}}(x) |\sum_{i=1}^{n+2} a_i x_i| ,$$

for $c = (\widetilde{f}^{p^{n+2}}(1), \ldots, \widetilde{f}^{p^{n+2}}(n+2))$, where $f(x) = \mathrm{sign}(\sum_{i=1}^{n+2} a_i x_i)$. So,

$$\mathbb{E}_{x\sim\mu_{p^{n+2}}}[|\sum_{i=1}^{n+2} a_i x_i|] = \frac{2\max_{c\in\mathcal{C}_{p^{n+2}}} a \cdot c}{\Lambda(p^{n+2})} .$$

□

Linear Optimization Using a Verification Oracle We now prove the intractability of the "restricted" verification problem, where the input linear threshold functions are defined by weight vectors of the specific form described in Definition 4.

Name: SVR$_{p^{n+2}}$-Verification Problem
Input: A vector $(c_1, \ldots, c_{n+2}) \in \mathbb{Q}^{n+2}$ and a vector $w = (w_1, \ldots, w_{n+2}) \in \mathbb{Q}^{n+2}$ such that $w_{n+1} = w_{n+2} = -\sum_{i=1}^n w_i/2$ and $w_i > 0, i \in \{1, \ldots, n\}$.
Question: Does it hold that $\widetilde{f}_{w,0}^{p^{n+2}}(i) = c_i$ for $i \in \{1, \ldots, n+2\}$?

Theorem 4. *If* p^{n+2} *is a* reasonable *probability vector, the SVR*$_{p^{n+2}}$*-Verification problem is not in the k-th level of the polynomial hierarchy, unless #P is contained in the k+2-level.*

The main idea behind the proof is that one can solve the linear optimization problem using a membership oracle of the polytope which can be obtained if we have an efficient algorithm for the "restricted" verification problem. Since the vertices of the polytope correspond to semivalues of linear threshold functions, if we have an efficient algorithm for the verification problem, then we can efficiently verify that a vector is a vertex of the polytope, and using Caratheodory's theorem, we can get a membership oracle. In this way, we obtain a contradiction, unless the polynomial hierarchy collapses: if the verification problem is in the polynomial hierarchy, then a #P-hard problem lies in the polynomial hierarchy.

The "restricted" membership problem for any probability vector p^{n+2} is defined below:

Name: $\mathrm{SVR}_{p^{n+2}}$-Membership Problem
Input: A vector $c = (c_1, \ldots, c_{n+2}) \in \mathbb{Q}^{n+2}$.
Question: Is c in $\mathcal{C}_{p^{n+2}}$?

Proof of Theorem 4. We firstly prove the following lemma that shows how an efficient oracle for the "restricted" verification problem can be used to obtain a membership oracle:

Lemma 1. *If the $\mathrm{SVR}_{p^{n+2}}$-Verification problem is in the k-th level of PH, then the $\mathrm{SVR}_{p^{n+2}}$-Membership problem is in the $(k+1)$-level.*

Proof. Assume that the $\mathrm{SVR}_{p^{n+2}}$-Verification problem is in the k-th level of PH. By Caratheodory's theorem a point c is in $\mathcal{C}_{p^{n+2}}$ iff it is a convex combination of at most $n+3$ vertices of $\mathcal{C}_{p^{n+2}}$, i.e., $c = \sum_{i=1}^{m} \lambda_i x^i$, where x^i is a vertex, $\lambda_i \geq 0$ for $1 \leq i \leq m \leq n+3$ and $\sum_{i=1}^{m} \lambda_i = 1$. So, it can be certified that a given point c is in $\mathcal{C}_{p^{n+2}}$ by finding the $m \leq n+3$ vertices x^i and computing the m scaling factors λ_i. Given the x^i, one can verify that x^i is a vertex of $\mathcal{C}_{p^{n+2}}$ by finding w^i (of the form described in Definition 4) such that x^i is the $\widetilde{f}^{p^{n+2}}_{w^i,0}$ vector, as the vertices of $\mathcal{C}_{p^{n+2}}$ correspond to linear threshold functions with weights of this specific form. So, if we are given the x^i and the corresponding w^i, we can verify in polynomial time with a k-th level oracle that x^i is the $\widetilde{f}^{p^{n+2}}_{w^i,0}$ vector as we assumed that the SVR_p^{n+2}-Verification problem is in the k-th level of PH. Thus, there is a polynomial-size certificate that can be checked in polynomial time with a k-th level oracle when c is in $\mathcal{C}_{p^{n+2}}$: the $m \leq n+3$ vertices x^i, where each x^i can be represented by poly(n) bits by Remark 1, and the m corresponding w^i vectors, where each w^i can be represented by poly(n) bits as every linear threshold function can be represented with weight $w = (w_1, \ldots, w_{n+2})$ such that each w_i is an integer that satisfies $|w_i| \leq 2^{(O(n \log n))}$ (Muroga, Toda, and Takasu 1961). Given the x^i and the w^i, it can be verified in polynomial time with a k-th level oracle that the x^i are vertices and then we can compute in polynomial time the λ_i coefficients by solving the linear system $c = \sum_{i=1}^{m} \lambda_i x^i$. Thus, if the SVR-Verification problem is in the k-th level of PH, the $\mathrm{SVR}_{p^{n+2}}$-Membership problem is in the $(k+1)$-level. □

As $\mathcal{C}_{p^{n+2}}$ has non-empty interior, if the $\mathrm{SVR}_{p^{n+2}}$-Membership problem is in the $(k+1)$-level of PH, then using the ellipsoid algorithm, we could solve the optimization problem using a polynomial number of membership-oracle calls (page 189, (Schrijver 1998)). Hence, we would have that the linear optimization problem, which is #P-hard, is in the $(k+2)$-level of PH. □

Hardness of the Verification Problem for Weighted Voting Games One issue is that the computational problems we have studied so far involve linear threshold functions some of whose weights can be negative. This seemed necessary to some extent, as it is crucially exploited in the proof of Theorem 2.

We now show how to switch to weighted voting games, which was our initial goal. Using a bijection between the semivalues of a linear threshold function with weight vector w of the form described in Definition 4 and a linear threshold function with weight vector the absolute values of w, we show the equivalence between the "restricted" verification problem and the verification problem for linear threshold functions defined by positive weights.

Name: SV_{p^n}-Verification Problem
Input: A vector $(c_1, \ldots, c_n) \in \mathbb{Q}^n$, a vector $w = (w_1, \ldots, w_n) \in \mathbb{Q}^n_+$ and $\theta \in \mathbb{Q}$.
Question: Does it hold that $\widetilde{f}^{p^n}_{w,\theta}(i) = c_i$ for $i \in \{1, \ldots, n\}$?

Theorem 5. *If the $\mathrm{SV}_{p^{n+2}}$-Verification problem is in the k-th level of PH, then the $\mathrm{SVR}_{p^{n+2}}$-Verification problem is in the k-th level of PH.*

Proof. We use the following lemma (due to space limitations, the proof is omitted and is available in the full version of this paper) that shows how one can compute the semivalues of a linear threshold function with weight vector w of the form described in Definition 4 given the semivalues of the linear threshold function with weight vector the absolute values of w, and vice-versa.

Lemma 2. *Let $f(x) = \mathrm{sign}(a_1x_1 + \cdots + a_nx_n - A/2x_{n+1} - A/2x_{n+2})$, $g(x) = \mathrm{sign}(a_1x_1 + \cdots + a_nx_n + A/2x_{n+1} + A/2x_{n+2})$, $a_i > 0, 1 \leq i \leq n$, $A = \sum_{i=1}^{n} a_i$. Then, for $1 \leq i \leq n$, $\widetilde{g}^{p^{n+2}}(i) = \widetilde{f}^{p^{n+2}}(i) - 2(p^{n+2}_{n+1} - p^{n+2}_{n-1})$ and for $n+1 \leq i \leq n+2$, $\widetilde{g}^{p^{n+2}}(i) = \widetilde{f}^{p^{n+2}}(i) + 2\sum_{t=0}^{n-1}\binom{n}{t}(p^{n+2}_t + p^{n+2}_{t+1})$.*

Given an instance $(a_1, \ldots, a_n, -A/2, -A/2)$, $(c_1, \ldots, c_{n+2})$ of the $\mathrm{SVR}_{p^{n+2}}$-Verification problem, we construct the following instance of $\mathrm{SV}_{p^{n+2}}$-Verification problem: $(a_1, \ldots, a_n, +A/2, +A/2)$, $\theta = 0$,

$$(c_1 - 2(p^{n+2}_{n+1} - p^{n+2}_{n-1}), \ldots, c_n - 2(p^{n+2}_{n+1} - p^{n+2}_{n-1}),$$
$$c_{n+1} + 2\sum_{t=0}^{n-1}\binom{n}{t}(p^{n+2}_t + p^{n+2}_{t+1}),$$
$$c_{n+2} + 2\sum_{t=0}^{n-1}\binom{n}{t}(p^{n+2}_t + p^{n+2}_{t+1}))\ .$$

According to Lemma 2, we have that the $\mathrm{SVR}_{p^{n+2}}$-Verification instance is a YES-instance iff the $\mathrm{SV}_{p^{n+2}}$-Verification instance is a YES-instance. □

Verification Using Inverse Oracle The final step of our proof is to show that the inverse problem for semivalues is at least as hard as the verification problem. While this is intuitively obvious, the proof requires the following non-trivial structural result: The semivalues of a weighted voting game characterize the game within the space of weighted voting games.

Theorem 6. *If the SV_{p^n}-Inverse problem is in the k-th level of PH, then the SV_{p^n}-Verification problem is in the k+1-level.*

Proof. The proof makes essential use of the following lemma that shows that if two LTFs with normalized weights and the same threshold have the same semivalues, then they define the same function, for all Boolean vectors that are given positive probability by μ'_{p^n}. This lemma is qualitatively similar to (and inspired by) Chow's Theorem (Chow 1961), that shows that a linear threshold function is uniquely determined by its Chow parameters (Banzhaf indices).

Lemma 3. *Let $f(x) = \mathrm{sign}(w \cdot x - \theta)$ and $g(x) = \mathrm{sign}(v \cdot x - \theta)$ where $\sum_{i=1}^n w_i = \sum_{i=1}^n v_i$. If $\widetilde{f}^{p^n}(i) = \widetilde{g}^{p^n}(i)$ for $i \in \{1, \dots, n\}$, then $f(x) = g(x)$ for all $x \in \{-1,1\}^n$ such that $p^n_{wt(x)} + p^n_{wt(x)-1} \neq 0$ and $|\sum_{i=1}^n w_i x_i - \theta| + |\sum_{i=1}^n v_i x_i - \theta| \neq 0$.*

Proof. We can write:

$$
\begin{aligned}
&2\sum_{i=1}^n w_i(\widetilde{f}^{p^n}(i) - \widetilde{g}^{p^n}(i)) + 2\sum_{i=1}^n v_i(\widetilde{g}^{p^n}(i) - \widetilde{f}^{p^n}(i)) \\
&- \theta \sum_{x\in\{-1,1\}^n} (f(x) - g(x))\mu'_{p^n}(x) \\
&- \theta \sum_{x\in\{-1,1\}^n} (g(x) - f(x))\mu'_{p^n}(x) = 0 \\
&\Leftrightarrow \sum_{i=1}^n w_i(\widehat{f}^{p^n}(i) - \widehat{g}^{p^n}(i) + C_f^{p^n} - C_g^{p^n}) \\
&+ \sum_{i=1}^n v_i(\widehat{g}^{p^n}(i) - \widehat{f}^{p^n}(i) + C_g^{p^n} - C_f^{p^n}) \\
&- \theta \sum_{x\in\{-1,1\}^n} (f(x) - g(x))\mu'_{p^n}(x) \\
&- \theta \sum_{x\in\{-1,1\}^n} (g(x) - f(x))\mu'_{p^n}(x) = 0 \\
&\Leftrightarrow \sum_{x\in\{-1,1\}^n} (f(x) - g(x))\mu'_{p^n}(x) \sum_{i=1}^n w_i x_i \\
&+ \sum_{x\in\{-1,1\}^n} (g(x) - f(x))\mu'_{p^n}(x) \sum_{i=1}^n v_i x_i \\
&- \theta \sum_{x\in\{-1,1\}^n} (f(x) - g(x))\mu'_{p^n}(x) \\
&- \theta \sum_{x\in\{-1,1\}^n} (g(x) - f(x))\mu'_{p^n}(x) = 0 \\
&\Leftrightarrow \sum_{x\in\{-1,1\}^n} (f(x) - g(x))\mu'_{p^n}(x)(\sum_{i=1}^n w_i x_i - \theta) \\
&+ \sum_{x\in\{-1,1\}^n} (g(x) - f(x))\mu'_{p^n}(x)(\sum_{i=1}^n v_i x_i - \theta) = 0 \\
&\Leftrightarrow \sum_{x\in\{-1,1\}^n} \mu'_{p^n}(x)|f(x) - g(x)||\sum_{i=1}^n w_i x_i - \theta| \\
&+ \sum_{x\in\{-1,1\}^n} \mu'_{p^n}(x)|g(x) - f(x)||\sum_{i=1}^n v_i x_i - \theta| = 0 \\
&\Leftrightarrow \sum_{x\in\{-1,1\}^n} \mu'_{p^n}(x)|f(x) - g(x)|(|\sum_{i=1}^n w_i x_i - \theta| \\
&+ |\sum_{i=1}^n v_i x_i - \theta|) = 0\ .
\end{aligned}
$$

Hence, for any $x \in \{-1,1\}^n$ such that $|\sum_{i=1}^n w_i x_i - \theta| + |\sum_{i=1}^n v_i x_i - \theta| \neq 0$ and $\mu'_{p^n}(x) \neq 0$, we have that $f(x) = g(x)$. □

Given an SV_{p^n}-Verification instance $a = (a_1, \dots, a_n)$, θ, $(c_1, \dots, c_n)$, we create the following instance of the SV_{p^n}-Inverse problem: $\theta / \sum_{i=1}^n a_i$, $(c_1, \dots, c_n)$. Then, if the SV_{p^n}-Inverse instance is a NO-instance, we have a NO-instance of the SV_{p^n}-Verification problem. If the SV_{p^n}-Inverse output is a weight vector $w = (w_1, \dots, w_n)$, we can check with a co-NP oracle if the functions $f_{w,\theta/\sum_{i=1}^n a_i}$ and $f_{a,\theta}$ have the same semivalues: according to Lemma 3 they have the same semivalues iff there is no $x \in \{-1,1\}^n$ such that $\mu'_{p^n}(x) > 0$ and $f_{w,\theta/\sum_{i=1}^n a_i}(x) \neq f_{a,\theta}(x) = f_{\frac{a}{\sum_{i=1}^n a_i}, \frac{\theta}{\sum_{i=1}^n a_i}}(x)$. □

Discussion

The inverse power index problem has received considerable attention in game theory and social choice, and the inverse Banzhaf index problem has been relevant in other fields as well, such as circuit theory and computational learning. In this paper, we proved that the inverse semivalue problem, for *reasonable* probability distributions, is computationally intractable. As special cases, we deduce that the inverse Banzhaf index and inverse Shapley value problems are also intractable. A number of interesting open questions remain: Can we design efficient approximation algorithms for the inverse problem in the case of more general semivalues? Can we characterize the computational complexity of the inverse power index problem for power indices that do not belong in the semivalues class?

References

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

Aziz, H. 2008. Complexity of comparison of influence of players in simple games. In *Proceedings of the Second International Workshop on Computational Social Choice*, 61–72.

Banzhaf III, J. F. 1964. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.* 19:317.

Carreras, F., and Freixas, J. 2008. On ordinal equivalence of power measures given by regular semivalues. *Mathematical Social Sciences* 55(2):221–234.

Carreras, F.; Freixas, J.; and Puente, M. A. 2003. Semivalues as power indices. *European Journal of Operational Research* 149(3):676–687.

Chow, C.-K. 1961. On the characterization of threshold functions. In *Switching Circuit Theory and Logical Design, 1961. SWCT 1961. Proceedings of the Second Annual Symposium on*, 34–38. IEEE.

De, A.; Diakonikolas, I.; Feldman, V.; and Servedio, R. A. 2014. Nearly optimal solutions for the chow parameters problem and low-weight approximation of halfspaces. *Journal of the ACM (JACM)* 61(2):11.

De, A.; Diakonikolas, I.; and Servedio, R. A. 2017. The inverse shapley value problem. *Games and Economic Behavior* 105:122–147.

De Keijzer, B.; Klos, T. B.; and Zhang, Y. 2014. Finding optimal solutions for voting game design problems. *Journal of Artificial Intelligence Research* 50:105–140.

Deegan, J., and Packel, E. W. 1978. A new index of power for simplen-person games. *International Journal of Game Theory* 7(2):113–123.

Deng, X., and Papadimitriou, C. H. 1994. On the complexity of cooperative solution concepts. *Mathematics of Operations Research* 19(2):257–266.

Dubey, P., and Shapley, L. S. 1979. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research* 4(2):99–131.

Dubey, P.; Neyman, A.; and Weber, R. J. 1981. Value theory without efficiency. *Mathematics of Operations Research* 6(1):122–128.

Dughmi, S., and Xu, H. 2016. Algorithmic bayesian persuasion. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 412–425. ACM.

Einy, E. 1987. Semivalues of simple games. *Mathematics of Operations Research* 12(2):185–192.

Elkind, E.; Goldberg, L. A.; Goldberg, P. W.; and Wooldridge, M. 2008. On the dimensionality of voting games. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 69–74.

Elkind, E.; Goldberg, L. A.; Goldberg, P. W.; and Wooldridge, M. 2009. On the computational complexity of weighted voting games. *Annals of Mathematics and Artificial Intelligence* 56(2):109–131.

Faliszewski, P., and Hemaspaandra, L. A. 2008. The complexity of power-index comparison. In *International Conference on Algorithmic Applications in Management*, 177–187. Springer.

Fatima, S.; Wooldridge, M.; and Jennings, N. R. 2008. An anytime approximation method for the inverse shapley value problem. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 935–942.

Gopalan, P.; Nisan, N.; and Roughgarden, T. 2015. Public projects, boolean functions and the borders of border's theorem. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, 395–395.

Holler, M. J. 1982. Forming coalitions and measuring voting power. *Political studies* 30(2):262–271.

Kurz, S. 2012. On the inverse power index problem. *Optimization* 61(8):989–1011.

Laruelle, A., and Widgrén, M. 1998. Is the allocation of voting power among eu states fair? *Public Choice* 94(3-4):317–339.

Leech, D. 2003. Power indices as an aid to institutional design: the generalised apportionment problem. *Yearbook on New Political Economy*.

Muroga, S.; Toda, I.; and Takasu, S. 1961. Theory of majority decision elements. *Journal of the Franklin Institute* 271(5):376–418.

O'Donnell, R., and Servedio, R. A. 2011. The chow parameters problem. *SIAM Journal on Computing* 40(1):165–199.

Prasad, K., and Kelly, J. S. 1990. Np-completeness of some problems concerning voting games. *International Journal of Game Theory* 19(1):1–9.

Schrijver, A. 1998. *Theory of linear and integer programming*. John Wiley & Sons.

Shapley, L. S., and Roth, A. E., eds. 1988. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.

Shapley, L. S., and Shubik, M. 1954. A method for evaluating the distribution of power in a committee system. *American political science review* 48(3):787–792.

Shapley, L. S. 1953. A value for n-person games. *Contributions to the Theory of Games* 2(28):307–317.

Weber, R. J. 1979. Subjectivity in the valuation of games. Technical report, Yale Univ New Haven Conn Cowles Foundation For Research In Economics.