Journal of Optimization Theory and Applications (2019) 181:184–196
https://doi.org/10.1007/s10957-018-1442-y

Power Diagram Detection with Applications to Information Elicitation

Steffen Borgwardt[1] · **Rafael M. Frongillo**[2]

Received: 11 December 2017 / Accepted: 11 November 2018 / Published online: 22 November 2018

Abstract

Power diagrams, a type of weighted Voronoi diagram, have many applications throughout operations research. We study the problem of power diagram detection: determining whether a given finite partition of $\mathbb{R}^d$ takes the form of a power diagram. This detection problem is particularly prevalent in the field of information elicitation, where one wishes to design contracts to incentivize self-minded agents to provide honest information. We devise a simple linear program to decide whether a polyhedral cell complex can be described as a power diagram. For positive instances, a representation of the cell complex as a power diagram is returned. Further, we discuss applications to property elicitation, peer prediction, and mechanism design, where this question arises. Our model can efficiently decide the question for complexes of $\mathbb{R}^d$ or of a convex subset thereof. The approach is based on the use of an alternative representation of power diagrams and invariance of a power diagram under uniform scaling of the parameters in this representation.

Keywords Power diagram · Information elicitation · Linear programming

Mathematics Subject Classification 90C05 · 90C90 · 91B06 · 62C05

Communicated by Anita Schöbel.

✉ Steffen Borgwardt
steffen.borgwardt@ucdenver.edu

Rafael M. Frongillo
raf@colorado.edu

1 University of Colorado Denver, Denver, CO, USA

2 University of Colorado Boulder, Boulder, CO, USA

1 Introduction

Power diagrams play an important role in many disciplines within operations research, ranging from balanced least-squares clustering [1–3] to multiclass classification [4–9]. They have recently surfaced in the domain of information elicitation, where one wishes to design contracts or mechanisms to incentivize a self-minded agent to reveal their private information truthfully [10–14]. In the latter, power diagrams are shown in various settings to characterize the possible sets of information that can be treated identically under the mechanism; this characterization is an important step in understanding which mechanisms are or are not *truthful*, i.e., which elicit this information effectively. Motivated by this application in particular, where one wishes to know whether a given mechanism or contract could be truthful, we study the problem of power diagram detection: deciding whether a given cell complex is in fact a power diagram.

Informally, a *cell complex* is a partition of $\mathbb{R}^d$ into a finite set of j-faces of dimension $0 \leq j \leq d$. A *polyhedral cell complex* is a cell complex where all faces are polyhedra. For our purposes, it suffices to consider the full-dimensional d-faces, called *cells*. The cells form a partition of $\mathbb{R}^d$ in the sense that their union is $\mathbb{R}^d$ and only their boundaries may intersect.

A *power diagram* is a cell complex defined by a set of sites, each with its own weight. Each of the sites defines one of the cells. The cell corresponding to a site is the set of all points which are closest to the site in squared Euclidean distance, discounting by its weight squared; see also Definition 2.1. Geometrically, separating hyperplanes between the cells are defined by the intersection of balls of varying radii around the sites. The different weights correspond to the different radii. An example is depicted in Fig. 1.

Any power diagram is a polyhedral cell complex. In particular, the shared boundary of any pair of cells is either empty or is contained in a hyperplane and thus can be described via linear constraints. The problem we consider is the following: given

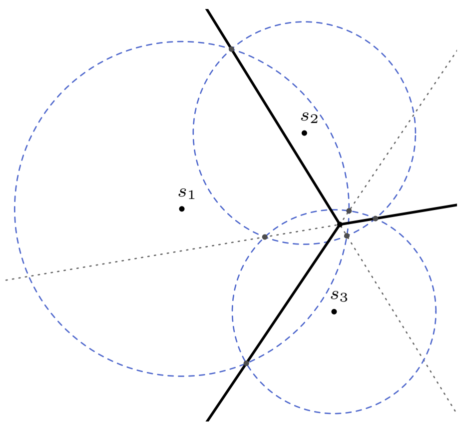

Fig. 1 A power diagram with three cells

some polyhedral cell complex P, where the cells are described by linear inequalities, determine whether or not P is a power diagram. In other words, determine whether there is a representation of the cell complex P as a power diagram.

It is well known that all simple, polyhedral cell complexes in dimension 3 or higher can be represented as a power diagram [15,16]. We want to stress that we do *not* assume simplicity and we do not assume dimension 3 or higher. We do not know whether a given cell complex is simple or not. Non-simple cell complexes may have a representation as a power diagram, but they do not have to. In dimension 2, even simple cell complexes may not be power diagrams. Moreover, all of these cases are relevant for applications.

In this paper, we devise a novel linear program to solve this problem of *power diagram detection*, detailed in Sect. 2, with a running time complexity that is (at most) weakly polynomial. For positive instances, a representation of the cell complex as a power diagram is returned. We treat two variants of the problem, where the complex decomposes $\mathbb{R}^d$, and where it decomposes a restricted convex subset of $\mathbb{R}^d$. We then present and discuss several applications to information elicitation in Sect. 3. We conclude in Sect. 4 with a brief review of our contribution and additional challenges in practice.

2 Detection by Linear Programming

We begin with a formal definition of a power diagram [17].

Definition 2.1 (*Power Diagram, Original*) Given sites $s_1, \ldots, s_k \in \mathbb{R}^d$ and weights $w_1, \ldots, w_k \in \mathbb{R}$, we define the cell corresponding to site s_i as

$$\text{cell}(s_i) = \left\{ x \in \mathbb{R}^d : i \in \text{argmin}_j \|x - s_j\|^2 - w_j^2 \right\}. \tag{1}$$

A cell complex $P = (P_1, \ldots, P_k)$ of $\mathbb{R}^d$ is a *power diagram*, if there exist $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and $\{w_i\}_{i=1}^k \subseteq \mathbb{R}$ so that for all $i \in \{1, \ldots, k\}$ we have $P_i = \text{cell}(s_i)$, as in Eq. (1).

Here $\|\cdot\| = \|\cdot\|_2$ refers to the Euclidean norm. Several basic observations follow from this definition. First, note that $\text{cell}(s_i)$ depends on *all* sites $s_1, \ldots, s_k$ and weights $w_1, \ldots, w_k$. Second, by setting $w_1 = \cdots = w_k$ we recover the definition of a Voronoi diagram, where points are assigned to a partition according to which site is closest in Euclidean distance. Power diagrams exhibit many useful properties beyond these observations above; see, e.g., [5,16–19].

We devise a linear program to check whether a given polyhedral cell complex can be represented as a power diagram. For this, recall that polyhedra can be described as the intersection of finitely many halfspaces. Let us begin with a formal problem statement.

Problem 2.1 (*Power Diagram Detection*) Let $P = (P_1, \ldots, P_k)$ be a polyhedral cell complex defined by a set of separating hyperplanes between the cells. Let the index sets $J_i = \{j : \dim(P_i \cap P_j) = d - 1\}$ give the adjacency structure in the complex.

Finally, let $a_{ij} \in \mathbb{R}^d$ and $\gamma_{ij} \in \mathbb{R}$ with

$$P_i = \left\{ x \in \mathbb{R}^d : a_{ij}^T x \leq \gamma_{ij} \ \forall j \in J_i \right\}. \quad (2)$$

Decide: Is there a set of sites $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and weights $\{w_i\}_{i=1}^k \subseteq \mathbb{R}$ such that $P_i = \text{cell}(s_i)$ for all $i \in \{1, \ldots, k\}$?

Related questions have been studied for stronger input or the easier special case of Voronoi diagrams in the literature: Aurenhammer proved that, based on the information in an incidence lattice of a cell complex, it is possible to determine whether the cell complex is the Voronoi diagram of a set of sites in time linear in the number of facets of the cell complex [20]. A similar result, with time linear in the number of vertices of the cell complex, holds in the dual setting of Dirichlet tesselations, as shown by Ash and Bolker [21]. Hartvigsen [22] devised two algorithms, one of which runs in strongly polynomial time, for the recognition whether a given cell complex is a Voronoi diagram; see also [23]. The algorithms use input similar to the input for Problem 2.1 and are based on linear programming, but they are quite different from the methods we will present below. The main reason is the easier setting for Voronoi diagrams, where the sites of two neighboring cells have to be of equal distance to the separating hyperplane. This is a defining property for Voronoi diagrams, but not true for power diagrams. It is the key idea to both algorithms. Finally, Rybnikov presented an algorithm to decide whether a cell complex given through an incidence graph with information on facets and $(d-2)$–faces is a power diagram [24]. The algorithm is based on finding a feasible point in a set of equalities and strict inequalities. Its running time is weakly polynomial, provided one can address its numerical instability and the challenging implementation of a tailored interior point method.

In this paper, we devise a simple solution to Problem 2.1 that will give a straightforward resolution for applications such as those outlined in Sect. 3. The advantages lie in a simple implementation, solvability by any linear programming algorithm, return of a constructive solution, numerical stability, and a practical performance that matches its (favorable) theoretical bound; see Theorems 2.1 and 2.2.

Our first tool is the use of an alternative definition of power diagrams, which has been devised in different ways, ranging from pairwise linear separation in a multiclass setting [4,5,18] to duality theory [19].

Definition 2.2 (*Power Diagram, Alternative*) For a given set of sites $s_1, \ldots, s_k \in \mathbb{R}^d$ and parameters $\gamma_1, \ldots, \gamma_k \in \mathbb{R}$, we define the cell corresponding to site s_i as

$$\text{cell}(s_i) = \left\{ x \in \mathbb{R}^d : (s_j - s_i)^T x \leq \gamma_j - \gamma_i \ \forall j \neq i \right\}. \quad (3)$$

A cell complex $P = (P_1, \ldots, P_k)$ of $\mathbb{R}^d$ is a *power diagram*, if there exist $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and $\gamma_1, \ldots, \gamma_k \in \mathbb{R}$ so that for all $i \in \{1, \ldots, k\}$ we have $P_i = \text{cell}(s_i)$, as in Eq. (3).

It is easy to match this definition with the original notion of a power diagram through a small transformation [5]: The cells P_i and P_j are separated by the hyperplane

$$\begin{aligned} H_{ij} &:= \left\{ x \in \mathbb{R}^d : \|s_i - x\|^2 - w_i = \|s_j - x\|^2 - w_j \right\} \\ &= \left\{ x \in \mathbb{R}^d : 2(s_j - s_i)^T x = \left(s_j^T s_j - w_j \right) - \left(s_i^T s_i - w_i \right) \right\}. \end{aligned}$$

Choosing $\gamma_i = \frac{1}{2}\left(s_i^T s_i - w_i\right)$ for $i \leq k$ yields this new representation, which fits with the definition of piecewise-linear separability [4], with the small exception that we use weak instead of strict inequalities. The right-hand sides in the above take the form $\gamma_j - \gamma_i$. This means that the ability to devise a set of sites $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and $\gamma_1, \dots, \gamma_k$ would be a positive resolution to Problem 2.1.

Our second tool is the invariance of power diagrams under scaling [18,25]. More precisely, the above representation of power diagrams is invariant under scaling of all s_i and γ_i, for all $i \leq k$, with the same parameter. For our purposes, it is important to note that in the representation of H_{ij}, all parameters can be scaled with a $\lambda_{ij} > 0$ without changing H_{ij}:

$$\begin{aligned} & 2(\lambda_{ij} s_j - \lambda_{ij} s_i)^T x = \lambda_{ij}\gamma_j - \lambda_{ij}\gamma_i \\ \iff\ & \lambda_{ij} \cdot 2(s_j - s_i)^T x = \lambda_{ij} \cdot (\gamma_j - \gamma_i) \\ \iff\ & 2(s_j - s_i)^T x = \gamma_j - \gamma_i. \end{aligned}$$

Of course, the hyperplane specified by $a_{ij}^T x \leq \gamma_{ij}$ in Eq. (2) is similarly invariant under joint scaling of the left-hand and right-hand sides.

These tools allow us to construct a linear program to check whether the cells specified by the given input can be represented as the cells of a power diagram. Recall Eq. (2), which represents the cells of the input in the form

$$P_i = \left\{ x \in \mathbb{R}^d : a_{ij}^T x \leq \gamma_{ij}\ \forall j \in J_i \right\}.$$

We wish to match this form to the description of the cells from Eq. (3),

$$\mathrm{cell}(s_i) = \left\{ x \in \mathbb{R}^d : (s_j - s_i)^T x \leq \gamma_j - \gamma_i\ \forall j \neq i \right\}.$$

By the above, we may use any joint scaling of a_{ij} and γ_{ij} by a factor λ_{ij} in this match.

Consider the following linear program for power diagram detection

$$\begin{aligned} \lambda_{ij} \cdot a_{ij} &= s_i - s_j & \forall i \leq k, \quad \forall j \in J_i \\ \lambda_{ij} \cdot \gamma_{ij} &= \gamma_j - \gamma_i & \forall i \leq k, \quad \forall j \in J_i \\ \lambda_{ij} &\geq 1 & \forall i \leq k, \quad \forall j \in J_i \end{aligned} \tag{PDD}$$

First, note that the a_{ij} and γ_{ij} are given constants. The $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$, the $\gamma_1, \dots, \gamma_k \in \mathbb{R}$, and the $\lambda_{ij}\ \forall i \leq k, \forall j \in J_i$ are variables. Further, note that we do not specify an objective function. We are only interested in finding a feasible solution, so a dummy objective function or a Phase-0 formulation of this program will be sufficient.

For all hyperplanes in the form $a_{ij}^T x \leq \gamma_{ij}$ specified by the original input, this linear program checks whether a_{ij} and γ_{ij} can be jointly scaled to match Definition 2.2 of a power diagram. These are the first two lines of the program.

Due to the invariance of power diagrams under joint scaling of all their parameters, the constraints $\lambda_{ij} \geq 1 \quad \forall i \leq k, \forall j \in J_i$ may be imposed without losing the ability to construct a power diagram (in the cases where this is possible). These additional constraints guarantee that no trivial scaling $\lambda_{ij} = 0$ and $s_j = s_i$ is feasible, as well as that no negative scaling $\lambda_{ij} < 0$ is feasible, so that the direction of the separation of cells P_i and P_j cannot change.

The program (PDD) suggests the following algorithm, which we use to show the weakly polynomial solvability of Problem 2.1.

Algorithm 1 Unrestricted Power Diagram Detection

Input: $J_i \subseteq \{1, \ldots, k\}, a_{ij} \in \mathbb{R}^d, \gamma_{ij} \in \mathbb{R}$ for all $i, j \in \{1, \ldots, k\}$
Set up an instance I of linear program (PDD) using the constants a_{ij}, γ_{ij}, and J_i
if I is not feasible, **return** NO
Let $\{s_i\}_{i=1}^k, \{\gamma\}_{i=1}^k, \{\lambda_{ij} : i \leq k,\ j \in J_i\}$ be a feasible solution
Let $w_i = s_i^T s_i - 2\gamma_i$ for all $i \in \{1, \ldots, k\}$
return $\{s_i\}_{i=1}^k, \{w_i\}_{i=1}^k$

Theorem 2.1 *For all rational input, Algorithm 1 solves Problem 2.1 in weakly polynomial time, returning a representation of the power diagram for yes-instances.*

Proof Recall that linear programs can be solved in weakly polynomial time through some variants of interior point methods, or the Ellipsoid method. As all the parameters in (PDD) are from the original input, it suffices to show that the number of variables and constraints is strongly polynomial in the size of the input. Let $p = \sum_{i=1}^k |J_i|$. There are exactly $3p$ constraints and k variable vectors of type $s_i \in \mathbb{R}^d$, k variables of type γ_i, and p variables of type λ_{ij}, giving a total of $kd + k + p$ variables.

The variables $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and $\gamma_1, \ldots, \gamma_k \in \mathbb{R}$ of a feasible solution represent a power diagram for a yes-instance, as in Definition 2.2. Further, corresponding values for weights w_i (as in Defintion 2.1) can be derived from the γ_i through the equality $\gamma_i = \frac{1}{2}(s_i^T s_i - w_i)$ for $i \leq k$. □

In the next section, we exhibit a collection of applications in which power diagram detection arises. The practical problems essentially fit with the statement of Problem 2.1, with the exception that often the domain is restricted, in the sense that the cells P_i given are actually $P_i \cap D$ for some (full-dimensional) domain D. We therefore must address the problem of power diagram detection in restricted domains, where one asks whether the given cells can be expressed as a power diagram *restricted to* D. We formally capture this problem in the following.

Problem 2.2 (*Power Diagram Detection for Restricted Domains*) Let $D \subseteq \mathbb{R}^d$ be a convex domain, and let $P = (P_1, \ldots, P_k)$, $P_i \subseteq D$, be a polyhedral cell complex restricted to D defined by a set of separating hyperplanes between the cells. Let the index sets $J_i^D = \{j : \dim(P_i \cap P_j) = d - 1\}$ give the adjacency structure in the complex. Finally, let $a_{ij} \in \mathbb{R}^d$ and $\gamma_{ij} \in \mathbb{R}$ with

$$P_i = \left\{x \in D : a_{ij}^T x \leq \gamma_{ij} \ \forall j \in J_i^D \right\}. \tag{4}$$

Decide: Is there a set of sites $\{s_i\}_{i=1}^k \subseteq \mathbb{R}^d$ and weights $\{w_i\}_{i=1}^k \subseteq \mathbb{R}$ such that $P_i = \text{cell}(s_i) \cap D$ for all $i \in \{1, \ldots, k\}$?

This problem variant can be resolved through a small tweak to (PDD): the index sets J_i are replaced with the index sets J_i^D. We obtain the following linear program for restricted power diagram detection (r-PDD) for the detection of a power diagram in a restricted domain.

$$\begin{aligned} \lambda_{ij} \cdot a_{ij} &= s_i - s_j & \forall i \leq k, \quad \forall j \in J_i^D \\ \lambda_{ij} \cdot \gamma_{ij} &= \gamma_j - \gamma_i & \forall i \leq k, \quad \forall j \in J_i^D \\ \lambda_{ij} &\geq 1 & \forall i \leq k, \quad \forall j \in J_i^D \end{aligned} \tag{r-PDD}$$

We now show that Problem 2.2 is also weakly polynomially solvable. On the one hand, we do not find this result surprising, as the structure of the associated linear program remains essentially the same as for Problem 2.1. One must take care, however, to understand the impact of the restricted domain and why the program does not give false-positive answers.

Theorem 2.2 *For all rational input, Algorithm 1, when run on the input* $(\{J_i^D\}_i, \{a_{ij}\}_{ij}, \{\gamma_{ij}\}_{ij})$*, solves Problem 2.2 in weakly polynomial time, returning a representation of the power diagram for yes-instances.*

Proof The running time claim, and the return of a representation of a power diagram for yes-instances, follows analogously to the proof of Theorem 2.1. The single, yet important, difference lies in the restricted domain D. We have to consider how the index set J_i^D in Problem 2.2 relates to the cells P_i in the restricted problem. More precisely, we have to make sure that a yes-answer to the problem corresponds precisely to those inputs, where the polyhedral cells in the domain D come from the intersection of a power diagram in $\mathbb{R}^d$ with the domain D.

To this end, let P' be a cell complex in $\mathbb{R}^d$ as defined in Problem 2.1, and let J_i be the corresponding index sets. Further, let J_i^D be the index sets as defined in Problem 2.2. Note $J_i^D \subseteq J_i$. Thus any feasible solution (s, γ) to (PDD) (for index sets J_i) is also a feasible solution to (r-PDD) (for index sets J_i^D). Recall that, by Definition 2.2, a feasible solution (s, γ) for (PDD) provides the parameters to represent a power diagram P' in $\mathbb{R}^d$. This implies that if there exists a power diagram P' in $\mathbb{R}^d$ with $P = P' \cap D$, the corresponding (s, γ) is a feasible solution to both (PDD) and (r-PDD) and we correctly identify a yes-instance for Problem 2.2.

It remains to show that all no-instances are identified correctly, as well. Assume there is a feasible solution (s, γ) to (r-PDD), and let P' be the corresponding power diagram in $\mathbb{R}^d$. Let index sets J_i and J_i^D represent the adjacency of cells of P' in

$\mathbb{R}^d$, respectively, in D, as defined in Problems 2.1 and 2.2. We have to show that $P = P' \cap D$.

By construction, $P_i = \{x \in D : (s_j - s_i)^T x \leq \gamma_j - \gamma_i \ \forall j \in J_i^D\}$ and $P_i' \cap D = \{x \in D : (s_j - s_i)^T x \leq \gamma_j - \gamma_i \ \forall j \in J_i\}$ for all $i = 1, \ldots, k$. This gives $(P_i' \cap D) \subseteq P_i$ for all $i = 1, \ldots, k$.

Finally, recall that the cells $P_1, \ldots, P_k$ decompose the convex domain D, i.e., $\bigcup_{i=1}^k P_i = D$ with $\text{int}(P_i) \cap \text{int}(P_j) = \emptyset$ for $i \neq j$. But as P' is a power diagram of $\mathbb{R}^d$, the $(P_i' \cap D)$ also decompose D, i.e., we have $\bigcup_{i=1}^k (P_i' \cap D) = D$ and $\text{int}(P_i') \cap \text{int}(P_j') = \emptyset$ for $i \neq j$. Together with $(P_i' \cap D) \subseteq P_i$, we obtain $P_i = (P_i' \cap D)$ for all $i = 1, \ldots, k$. This proves the claim. □

3 Applications to Information Elicitation

In the domain of information elicitation, one wishes to design contracts or mechanisms to incentivize a self-minded agent to reveal their private information truthfully. Often this private information comes in one of two varieties: a *belief* about some future event, or a *utility* or valuation of a particular commodity or outcome. Power diagrams play a central role in the design of such mechanisms, as characterization theorems show that contracts or mechanisms are truthful (have the correct incentives) if and only if they partition the information space (or *type space*) into cells that form a power diagram. As the information space is typically not all of $\mathbb{R}^d$, we will work with the restricted version, Problem 2.2. In what follows, we show how to apply Theorem 2.2 to three such information elicitation scenarios where power diagrams arise.

3.1 Property Elicitation

One of the most basic information elicitation tasks is to design a contract to elicit some function, or *property*, of an agent's belief. The literature dates back to Savage [26] and Osband [27], with its modern incarnation beginning with Lambert et al. [10,28]. Concretely, consider a finite set of outcomes $\mathcal{O}$, the set of probability distributions $\Delta(\mathcal{O})$, a finite set of possible *reports* $\mathcal{R}$, and a property $\Gamma : \Delta(\mathcal{O}) \to 2^{\mathcal{R}}$ which designates a set of reports considered correct or desired for an agent with a particular belief. For example, the map $\Gamma(p) = \text{argmax}_{o \in \mathcal{O}} p(o)$ is the mode functional, which one notes is set-valued whenever there are multiple outcomes with the highest probability.

To elicit the property Γ from an agent, we wish to design a contract $S : \mathcal{R} \times \mathcal{O} \to \mathbb{R}$ which determines the payment to the agent once the outcome materializes. The protocol is thus: the score S is announced, the agent reports some $r \in \mathcal{R}$, the outcome $o \in \mathcal{O}$ is revealed, and the agent is paid $S(r, o)$. We say that the score S *elicits* the property Γ if for all beliefs $p \in \Delta(\mathcal{O})$, we have

$$\Gamma(p) = \underset{r \in \mathcal{R}}{\text{argmax}} \, \mathbb{E}_{o \sim p}[S(r, o)], \tag{5}$$

that is, the agent maximizes their expected score according to their belief p by reporting $r \in \Gamma(p)$. As a simple example, if one would like to know which of sun, rain, or

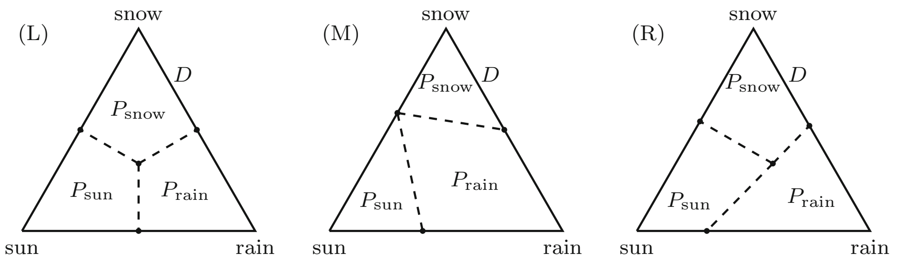

Fig. 2 **(L)** The cell complex corresponding to the mode functional, which is a power diagram (indeed, a Voronoi diagram) intersected with D, the probability simplex over outcomes {sun, rain, snow} projected onto $\mathbb{R}^2$. (Take e.g., sites at the corners of the simplex, with equal weights.) **(M)** Another cell complex which is a power diagram intersected with D, and therefore corresponds to an elicitable property. Note that $J_{\text{sun}} = \{\text{rain, snow}\}$ would be the index set in Problem 2.1, whereas $J^D_{\text{sun}} = \{\text{rain}\}$ in Problem 2.2. **(R)** A cell complex, which is not a power diagram intersected with D, and therefore not an elicitable property

snow, is most likely for the weather tomorrow, one would set $\mathcal{R} = \mathcal{O} = \{\text{sun, rain, snow}\}$ and offer to pay the agent $S(r, o) = \mathbb{1}\{r = o\}$, as this score elicits the mode: $\text{argmax}_r \mathbb{E}_{o\sim p}[\mathbb{1}\{r = o\}] = \text{argmax}_r\, p(r)$.

A natural question is thus the following: which properties Γ are *elicitable* in the sense that one can devise a score which elicits it? Perhaps surprisingly, the answer is simple: the elicitable properties Γ are precisely those for which the partition $P = (P_r)_{r\in\mathcal{R}}$ given by $P_r = \{p : r \in \Gamma(p)\}$ forms a power diagram [10,12]. More precisely, after projecting to $\mathbb{R}^{|\mathcal{O}|-1}$ (e.g., by dropping the last coordinate) the partition forms a power diagram intersected with the projected probability simplex. See Fig. 2(L) for an illustration of such a projection for the mode. The rough intuition for this elicitability result is as follows: Eq. (1) can be expressed via an argmin over k affine functions of x by dropping the irrelevant $\|x\|^2$ term, and then as the negation of an argmax of affine functions. As the expected score is linear (and therefore affine) in the agent's belief p, the distributions p which share the same optimal report (i.e., the same argmax) must form a cell of a power diagram.

As a practical matter, the question remains of how to test whether a given property is elicitable. Assuming the property is given as a polyhedral cell complex, which divides the probability simplex into regions with the same report, in light of the above characterization, this test can be done via Theorem 2.2, where the restricted domain D is the probability simplex, $D = \{x \in \mathbb{R}^d : \sum_{i=1}^d x_i = 1, x_i \geq 0 \quad \forall i \leq d\}$. (Technically, we would first project everything onto $\mathbb{R}^{d-1}$, as described above.) Note that the restricted domain can indeed change the index sets from (PDD) to (r-PDD), and even when $P_i \cap P_j \neq \emptyset$, one could still have $j \notin J_i^D$, as we illustrate in Fig. 2(M).

Essentially, the same formalism as described above arises in machine learning as well, in the context of designing loss functions for classification or ranking tasks [13, 29,30]. Here, one may wish to assign certain distributions over the labels (or classes) to different reports, and one may ask, for which such assignments does there exist a loss function which, when minimized, would yield these reports. This is again the same question as whether the property corresponding to this assignment is elicitable, where we simply negate the score to obtain a loss.

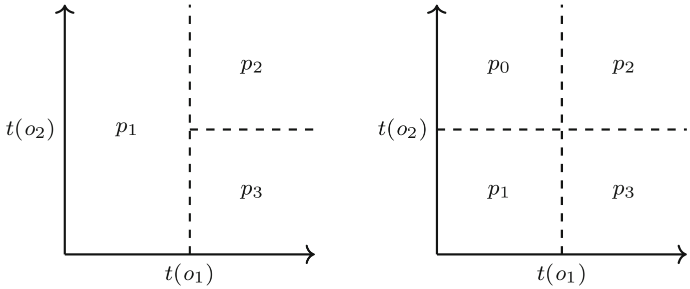

Fig. 3 An allocation rule "skeleton" which cannot be completed to an implementable allocation rule for any distinct choices of p_i (left), and one which could be so completed for appropriate choices of p_i (right). Here t denotes the type of an agent, and t being in the cell corresponding to p_i implies the allocation $f(t) = p_i$

3.2 Peer Prediction

A problem closely related to property elicitation is peer prediction, where one has access to several agents rather than one, but no direct access to objective information. Instead of designing a contract which scores an agent's report based on some observed outcome, one will never see the outcome and therefore must score the agents based on each other's reports [31,32].

To make headway, one typically assumes some structure about the underlying Bayesian (or pseudo-Bayesian) process through which agents form their beliefs. In particular, one assumes that agents receive some *signal* $S \in \mathcal{O}$ about the true outcome (like the quality of a hotel, or the correct label for an image classification task), and from this signal form a posterior belief $p(\cdot|S)$ about the true outcome, or about what another agent's signal was. Through assumptions about the possible values of this posterior, one can design mechanisms with a truthful equilibrium, in which each agent simply reports their signal S.

Constraints on the possible posteriors in the literature often take the form $p(\cdot|S = o) \in P_o$ for some sets $\{P_o : o \in \mathcal{O}\}$, dubbed a *belief model constraint* in Frongillo and Witkowski [14]. Their work shows that there exists a mechanism with a truthful equilibrium if and only if the sets $\{P_o\}$ form a power diagram [14, Corollary 3.5]. (More precisely, there must exist a power diagram with sites s_o such that $P_o \subseteq \text{cell}(s_o)$ for all $o \in \mathcal{O}$. We restrict attention to *maximal* constraints, where every distribution could be the posterior following some signal, i.e., $\bigcup_{o\in\mathcal{O}} P_o = \Delta(\mathcal{O})$.) The problem of detecting whether a belief model constraint supports a mechanism, and constructing said mechanism if so, thus reduces to detecting and constructing a power diagram in a restricted domain (again the probability simplex), to which Theorem 2.2 applies.

3.3 Mechanism Design

In mechanism design, one wishes to design an algorithm to choose an outcome based on the reports of the participants, but in a way which is robust to strategic misreports. From the well-known *revelation principle*, we may assume without loss of generality that the reports or "bids" submitted to a mechanism take the form of utility functions: each participant reports their utility for each possible outcome. From the reported utilities (called *types*) of the agents, the mechanism then chooses a distribution over outcomes (called the *allocation*) as well as a payment owed the mechanism by each participant. For simplicity, we will consider the case of a single agent.

Given the agent's type $t \in \mathbb{R}^{\mathcal{O}}$ which encodes their utility $t(o)$ following each outcome o, the mechanism wishes to choose a random allocation from some set $\mathcal{O}$ of outcomes, as well as the amount the agent should pay. Formally, given a finite outcome space $\mathcal{O}$ and a convex *type space* $\mathcal{T} \subseteq \mathbb{R}^{\mathcal{O}}$, a (direct, randomized) *mechanism* is a pair (f, p) where $f : \mathcal{T} \to \Delta_{\mathcal{O}}$ is the *allocation rule* and $p : \mathcal{T} \to \mathbb{R}$ is the *payment* function. We typically assume that the agent's utility is *quasi-linear* in the sense that their net utility upon allocation o and payment c is the difference $t(o) - c$. Note that the expected utility can be written $U(t', t) = f(t') \cdot t - p(t')$, where the inner product is between elements of $\mathbb{R}^{\mathcal{O}}$, where we consider $f(t') \in \Delta_{\mathcal{O}} \subset \mathbb{R}^{\mathcal{O}}$ to be the vector form of the allocation distribution. We say the mechanism (f, p) is *truthful* if $U(t', t) \leq U(t, t)$ for all types $t, t' \in \mathcal{T}$.

A fundamental question in mechanism design is *implementability*: given a desired allocation rule f, can one find a payment function p such that the pair (f, p) is a truthful mechanism? In other words, can one design payments to implement the desired allocation rule while being robust to the incentives of the agents? If given the proposed allocations themselves, one can use the fact that weak monotonicity (WMON) is sufficient and perform checks in $O(n^2)$ time [33]. We instead consider the following variant of this classic question: given a "skeleton" of an allocation rule, that decides upon "cells" of types to assign the same allocation but not what that allocation should be, could there be any assignment of allocations to cells such that the resulting allocation rule is implementable? See Fig. 3 for an illustration. More formally, given the skeleton $g : \mathcal{T} \to \{1, \ldots, m\}$, could there be any implementable allocation rule of the form $f = h \circ g$ for some $h : \{1, \ldots, m\} \to \Delta(\mathcal{O})$? From [12,33], such skeletons must form a power diagram, and thus, given g in the form (4) where $D = \mathcal{T}$ is the type space, this problem can also be solved as stated in Theorem 2.2.

4 Conclusions

We have presented a simple linear program for power diagram detection, the problem of deciding whether a polyhedral cell complex given by linear equalities can be represented as a power diagram. The problem of power diagram detection arises in information elicitation applications, and we have detailed three such applications: property elicitation, peer prediction, and mechanism design. In each of these problems, the domain is typically restricted to a subset D of $\mathbb{R}^d$, and thus, we have also addressed the restricted power diagram detection problem (Problem 2.2 and Theorem 2.2), to determine whether a given complex can be represented as a power diagram restricted to D. Combined, our results give a constructive, (weakly) polynomial time algorithm to determine whether a given property is elicitable, whether a given belief model constraint corresponds to a truthful peer prediction mechanism, or whether a given allocation "skeleton" could extend to an implementable allocation rule.

As a practical concern, we note that information elicitation problems do not necessarily exhibit polyhedral representations of their cells. More precisely, it is possible that the cells of a proposed property, belief model constraint, or allocation rule are not convex polyhedra and thus could not possibly be represented as those of a power diagram. Even when they allow a polyhedral representation, the input may be given in a form such that this is not obvious. For example, given a representation of a polyhedron as the convex hull of its vertices, it typically is not efficient to devise a representation as the intersection of halfspaces. In many applications, expert knowledge will allow a viable resolution of these issues, before the methods presented in this paper will be applied. In general however, they pose significant challenges before obtaining a broader detection algorithm in practice.

Acknowledgements We thank Ian Kash for helpful discussions on the applications of our results. Borgwardt gratefully acknowledges support through the Collaboration Grant for Mathematicians, *Polyhedral Theory in Data Analytics*, of the Simons Foundation, and Frongillo through the Computer and Information Science and Engineering Research Initiation Initiative Grant, *Characterization and Complexity of Information Elicitation*, of the National Science Foundation.

References

1. Borgwardt, S., Brieden, A., Gritzmann, P.: Geometric clustering for the consolidation of farmland and woodland. Math. Intell. **36**(2), 37–44 (2014)
2. Borgwardt, S., Brieden, A., Gritzmann, P.: An LP-based k-means algorithm for balancing weighted point sets. Eur. J. Oper. Res. **263**(2), 349–355 (2017)
3. Bradley, P.S., Bennett, K.P., Demiriz, A.: Constrained k-Means Clustering. Technical Report, p. MSR-TR-2000-65, Microsoft Research (2000)
4. Bennett, K.P., Mangasarian, O.L.: Multicategory discrimination via linear programming. Optim. Methods Softw. **3**, 27–39 (1992)
5. Borgwardt, S.: On soft power diagrams. J. Math. Model. Algorithms Oper. Res. **14**(2), 173–196 (2015)
6. Bredensteiner, E.J., Bennett, K.P.: Multicategory classification by support vector machines. Comput. Optim. Appl. **12**, 53–79 (1999)
7. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2**, 265–292 (2001)
8. Vapnik, V.: Statistical Learning Theory. Wiley, Hoboken (1998)
9. Weston, J., Watkins, C.: Multi-class Support Vector Machines. Technical Reports, University of London (1998)
10. Lambert, N.S., Shoham, Y.: Eliciting truthful answers to multiple-choice questions. In: Proceedings of the 10th ACM Conference on Electronic Commerce, pp. 109–118 (2009)
11. Lambert, N.S.: Elicitation and evaluation of statistical forecasts. Preprint (2018)
12. Frongillo, R., Kash, I.: General truthfulness characterizations via convex analysis. In: Web and Internet Economics, pp. 354–370 (2014)
13. Agarwal, A., Agarwal, S.: On consistent surrogate risk minimization and property elicitation. In: JMLR Workshop and Conference Proceedings, vol. **40**, pp. 1–19 (2015)
14. Frongillo, R., Witkowski, J.: A geometric perspective on minimal peer prediction. ACM Trans. Econ. Comput. **5(3), 17:1–17**, 27 (2017)
15. Davis, C.: The set of non-linearity of a piecewise-linear function. Scr. Math. **24**, 219–228 (1959)
16. Aurenhammer, F.: A criterion for the affine equivalence of cell complexes in $\mathbb{R}^d$ and convex polyhedra in $\mathbb{R}^{d+1}$. Discrete Comput. Geom. **2**(1), 49–64 (1987)
17. Aurenhammer, F.: Power diagrams: properties, algorithms and applications. SIAM J. Comput. **16**(1), 78–96 (1987)
18. Borgwardt, S.: A Combinatorial Optimization Approach to Constrained Clustering. Ph.D. Thesis (2010)

19. Brieden, A., Gritzmann, P.: On optimal weighted balanced clusterings: gravity bodies and power diagrams. SIAM J. Discrete Math. **26**, 415–434 (2012)
20. Aurenhammer, F.: Recognising polytopical cell complexes and constructing projection polyhedra. J. Symb. Comput. **3**(3), 249–255 (1987)
21. Ash, P.F., Bolker, E.D.: Recognizing Dirichlet tesselations. Geometriae Dedicata **19**, 175–206 (1985)
22. Hartvigsen, D.: Recognizing Voronoi diagrams with linear programming. INFORMS J. Comput. **4**(4), 357–452 (1992)
23. Aurenhammer, F., Klein, R., Lee, D.T.: Voronoi Diagrams and Delaunay Triangulations. World Scientific Publishing, Singapore (2013)
24. Rybnikov, K.: Stresses and liftings of cell-complexes. Discrete Comput. Geom. **21**(4), 481–517 (1999)
25. Aurenhammer, F., Hoffmann, F., Aronov, B.: Minkowski-type theorems and least-squares clustering. Algorithmica **20**, 61–76 (1998)
26. Savage, L.J.: Elicitation of personal probabilities and expectations. J. Am. Stat. Assoc. **66**(336), 783–801 (1971)
27. Osband, K.H.: Providing Incentives for Better Cost Forecasting. University of California, Berkeley (1985)
28. Lambert, N.S., Pennock, D.M., Shoham, Y.: Eliciting properties of probability distributions. In: Proceedings of the 9th ACM Conference on Electronic Commerce, pp. 129–138 (2008)
29. Bartlett, P.L., Jordan, M.I., McAuliffe, J.D.: Convexity, classification, and risk bounds. J. Am. Stat. Assoc. **101**(473), 138–156 (2006)
30. Ramaswamy, H.G., Agarwal, S., Tewari, A.: Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In: Advances in Neural Information Processing Systems, vol. **26**, pp. 1475–1483. Curran Associates Inc. (2013)
31. Miller, N., Resnick, P., Zeckhauser, R.: Eliciting informative feedback: the peer-prediction method. Manag. Sci. **51**(9), 1359–1373 (2005)
32. Witkowski, J., Parkes, D.C.: A robust bayesian truth serum for small populations. In: AAAI Conference on Artificial Intelligence (AAAI) (2012)
33. Saks, M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 286–293 (2005)