Abductive and Contrastive Explanations for Scoring Rules in Voting

Clément Contet, Umberto Grandi, Jérôme Mengin

ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240911

Abductive and Contrastive Explanations for Scoring Rules in Voting

Clément Contet, Umberto Grandi and Jérôme Mengin

IRIT, Université de Toulouse
{clement.contet, umberto.grandi, jerome.mengin}@irit.fr

Abstract. We view voting rules as classifiers that assign a winner (a class) to a profile of voters' preferences (an instance). We propose to apply techniques from formal explainability, most notably abductive and contrastive explanations, to identify minimal subsets of a preference profile that either imply the current winner or explain why a different candidate was not elected. Formal explanations turn out to have strong connections with classical problems studied in computational social choice such as bribery, possible and necessary winner identification, and preference learning. We design algorithms for computing abductive and contrastive explanations for scoring rules. For the Borda rule, we find a lower bound on the size of the smallest abductive explanations, and we conduct simulations to identify correlations between properties of preference profiles and the size of their smallest abductive explanations.

1 Introduction

Explaining the outcomes of artificial intelligence algorithms is one of the main challenges of present research in this field, with a plethora of competing approaches and venues to discuss them (see, e.g., the recent special issue edited by Miller et al. [30]). This problem is traditionally less of an issue for voting rules, where social choice theorists have used the axiomatic approach to justify the use of a specific voting rule. More recently, a stream of papers have used such axiomatic properties to obtain justifications for the outcome of a voting rule on a given preference profile [8, 33, 31]. Moreover, voting rules are no black boxes. They are usually given in the form of an explicit function or procedure that computes the winner of a given election. As already observed by Ebadian et al. [16], classical voting rules typically admit straightforward procedural explanations.

However, the realm of digital democracy can be led to implement arguably complex rules that require lengthy explanations to justify or explain how the outcome is computed from voters' preferences. To give two examples, the deliberation platform *LiquidFeedback*[1] uses the Schultze voting rule [2], and many participatory budgeting bodies are experimenting with the recently proposed method of equal shares[2] [34]. Moreover, recent work is starting to consider social choice on large number of alternatives, be them a list of government proposals or sentences produced by LLMs in response to prompts (see, e.g., [32, 11, 19]). In this setting, even voting rules whose functioning is easy to explain—such as scoring rules—may profit from the development of magnifying lenses, in the form of well-studied and scientifically grounded algorithms for explanations, that are able to point at which subsets of the preference profile determines the top positions in the collective ranking, arguably augmenting the voters' trust in the outcome of the vote while at the same time providing the analyst with important information on the structure of the voters' preferences.

This paper argues that such tools can be obtained by suitably adapting notions from the growing body of research on formal explanations in machine learning, most notably *abductive explanations* also called *prime implicant explanations* [35, 14, 29] and their dual *contrastive explanations* [22]. The underlying idea behind this endeavour is to explain the decision of supervised machine learning algorithms known as classifiers by identifying subsets of the feature space for which all possible completions do not change the classifier's decision (abductive explanations) or such that there exists a completion changing the decision (contrastive explanations, a special case of counterfactual explanations [37, 27]).

We will focus on a specific class of voting rules known as scoring rules, which includes the plurality rule, k-approval rules, and the Borda rule. Our choice is motivated by their computational properties—most notably the fact that computing a necessary winner can be done in polynomial time—as well as due to its wide application in recent digital democracy applications (see, e.g., [25, 32]).

Our contribution. We first adapt the definitions of abductive and contrastive explanations to the realm of voting rules in Section 2. We design an appropriate feature space on which explanations can be computed. Given our focus on scoring rules, we chose to represent profiles of preferences as *rank matrices*, in which $n \times m$ non-binary features represent (in a non-anonymous way) which among m candidates is placed in k-th entry by voter i in its voting ballot. We also show a useful connection between formal explanations and the classical concepts of possible and necessary winner which will form the basis of our algorithms for computing explanations.

Computing formal explanations for binary classifiers is a search problem in an exponential space, but efficient algorithms have been proposed in the literature, often using SAT solvers. Computing formal explanations in voting requires tackling the additional problem of interconnected features, implied by the assumptions that preferences are complete, transitive, and irreflexive binary relations. We define and analyse our algorithms for the computation of formal explanations in voting in Section 3.

Typically, the winner of a scoring rule on a preference profile can be explained by a large number of formal explanations. We are therefore interested in providing bounds on the size of the smallest ones.

[1] https://liquidfeedback.com/en/
[2] https://equalshares.net/

Such size can be considered as an intrinsic measure of the richness or the complexity of a preference profile, and it has surprising links with the robustness of the resulting winner. In Section 4 we first prove a lower bound on the size of abductive explanations for the Borda rule, showing that it is linear in the number of voters. To complement our worst-case analysis, we conduct computer simulations using the recent tool of map of elections [36, 3], showing an interesting negative correlation between the size of the smallest explanation and both the margin of victory of the winner and the intrinsic agreement of the preference profile.[3]

Related work The closest work and the source of inspiration for our research is the recent stream of papers that justifies or explains the result of a voting rule with a logical calculus on sequences of axiomatic properties. This approach was started by Cailloux and Endriss [8] for the Borda rule, then complemented by Boixel et al. [4] with a full calculus. Algorithms for computing justifications were presented by Nardi et al. [31] and implemented in a demo [5]. Peters et al. [33] gives minimal bounds on the length of the explanations.

Our encoding of preference profiles in a space of non-independent features is an instance of classifiers under constraints (our constraints being the transitivity, completeness and asymmetry of users' preferences). Formal explanations under constraints are the subject of a recent paper by Cooper and Amgoud [13], from which we borrow the concept of irredundant explanations.

Black-box machine learning techniques have shown to be effective in learning the behaviour of voting rules [1, 7], but their potential for obtaining explanations is limited. A recent paper by Kang et al. [23] proposes the use of decision trees, producing human-readable diagrams of voting rules, albeit on small number of candidates. Explainability of voting rules is at the center of a related paper by Ebadian et al. [16], which explores the tradeoff between adding randomisation to voting rules while preserving their procedural explanatory power.

To the best of our knowledge this is the first paper to consider counterfactual reasoning to explain voting rules. However, abductive and constrastive explanation have direct connections with known and well-studied concepts in voting theory, such as necessary and possible winners [24], bribery and control [17], defense against such attacks [28], and communication complexity of voting rules [10]. We explicit these connections in our concluding section.

2 Preliminaries

In this section we introduce the notions of abductive and contrastive explanations and apply them to the realm of voting theory.

2.1 Formal explanations

Formal explanations are tools recently developed for explainable artificial intelligence [35, 14, 22, 29, 27]. Given a classification problem, the aim is to find inclusion-minimal subsets of features that are able to explain the classifier decision on a given instance. For *abductive explanations* we look for minimal subsets of features such that any extension does not change the classifier decision, and for *contrastive explanations* we look for minimal subsets such that there exists an extension that reverts the classifier's decision. Both types of explanations are based on the computation of minimal conjunctions of literals representing whether the value of a feature can vary.

[3] The code and data of our experiment are available at https://gitlab.irit.fr/ccontet1/axp-and-cxp-for-scoring-rules

As an explanatory example, assume that the decision of whether to allow a bank client to open a mortgage is done by a classifier on the basis of age and revenue. A negative decision by the classifier on an instance can be explained by an abductive explanation pointing out that the revenue is too low, so that the answer is negative irrespective of the age of the client. A contrastive explanation, on the other hand, would point out that by keeping the same age and changing the revenue it would be possible to obtain the mortgage.

Formally, following the notations used by Marques-Silva [29], we are given a set of N features $\mathcal{F}$, with each i-th feature having values in finite domain $\mathcal{D}_i$, and a set of classes $\mathcal{K}$. A feature space is defined by $\mathbb{F} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N$ and a classifier is any non-constant function κ that maps the feature space $\mathbb{F}$ into the set of classes $\mathcal{K}$, i.e. $\kappa : \mathbb{F} \to \mathcal{K}$.

Definition 1. *Given* $\mathrm{v} \in \mathbb{F}$*, an abductive explanation (AXp) for the classification of* v *by* κ *is any minimal subset* $\mathcal{X} \subseteq \mathcal{F}$ *such that*

$$\forall(\mathrm{x} \in \mathbb{F}).\ \left[\left(\bigwedge_{i \in \mathcal{X}} (x_i = v_i)\right) \implies (\kappa(\mathrm{x}) = \kappa(\mathrm{v}))\right] \tag{1}$$

Given $\mathrm{v} \in \mathbb{F}$*, a contrastive explanation (CXp) for the classification of* v *by* κ *is any minimal subset* $\mathcal{Y} \subseteq \mathcal{F}$ *such that*

$$\exists(\mathrm{x} \in \mathbb{F}).\ \left[\left(\bigwedge_{i \in \mathcal{F} \setminus \mathcal{Y}} (x_i = v_i)\right) \wedge (\kappa(\mathrm{x}) \neq \kappa(\mathrm{v}))\right] \tag{2}$$

Intuitively, given a specific instance, an AXp is a minimal subset of issue values to keep in order to ensure that the outcome of κ remains unchanged, while a CXp is a minimal subset of issue values to erase in order to be able to change the outcome of κ.

2.2 Elections and scoring rules

An *election* consists of a set $\mathcal{A}$ of n agents, a set $\mathcal{C}$ of m candidates, and a *preference profile* $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_n)$, where each $\mathcal{P}_i$ is a linear order over $\mathcal{C}$, called a *preference relation* or *ballot*. We denote $\mathbb{P}$ the set of all possible profiles. A *non-resolute voting rule* F is a function that maps $\mathbb{P}$ into a subset of candidates, i.e., $F : \mathbb{P} \to P(\mathcal{C})$ with $P(\mathcal{C})$ the power set of the set of candidates.

There exists a particular class of voting rules known as *scoring rules*. Given a vector of weights $(w_1, w_2, \ldots, w_m)$, for each ballot the i^{th} candidate scores w_i points. The winners are then the candidates with the highest total score over all the ballots. Formally, given a preference profile $\mathcal{P}$, let $pos(c, \mathcal{P}_i)$ be the position of candidate c in the linear order $\mathcal{P}_i$ submitted by voter i, with the top-ranked candidate being in position 1. A scoring rule assigns to each candidate c a score equals to $\sum_{i \in \mathcal{A}} w_{pos(c, \mathcal{P}_i)}$, and the candidates with the highest score are declared the winners. Notable examples include the Borda rule defined by scoring vector $(w_1, w_2, \ldots, w_m) = (m-1, m-2, \ldots, 0)$, and k-approval rules defined by vectors $(1, \ldots, 1, 0 \ldots, 0)$, with k being the number of 1s in the scoring vector (with 1-approval being the plurality rule). For an introduction on voting rules we refer to Brams and Fishburn [6] and Zwicker [39].

Example 1. *Consider four candidates* A, B, C, D *and four voters. We represent the preferences of the voters with the rows of the following matrix, with the most preferred candidate on the left and the least preferred on the right:*

$$\mathcal{R} = \begin{array}{c} \begin{bmatrix} A & B & C & D \\ B & C & D & A \\ A & D & C & B \\ D & C & A & B \end{bmatrix} \end{array} \begin{array}{l} \textit{voter 1} \\ \textit{voter 2} \\ \textit{voter 3} \\ \textit{voter 4} \end{array}$$

The Borda scores of candidate A is 7, since it is ranked first by two voters and third by one voters. For the remaining candidates B, C, and D, the Borda scores are 5, 6, and 6, respectively. Hence, A is the winner for the Borda rule on this preference profile.

2.3 Formal explanations for elections

Voting rules can be seen as special cases of classifiers that take profiles of linear orders as input and output a set of winning candidates. However, this representation depends on how input profiles are encoded into a feature space. This choice is crucial to formal explanations, as the encoding into a feature space determines the space of possible explanations (see Definition 1), with important consequences on their expressiveness and computational complexity.

In the literature on (computational) social choice, incomplete preferences are usually represented as partial orders (see, e.g., [24, 38]). However, for the specific case of scoring rules we propose the use of *partial rank matrices* in view of their more compact representation.

Definition 2. *A* ***rank matrix*** $\mathcal{R} = (\mathcal{R}_1, \dots, \mathcal{R}_n)$ *is an* $n \times m$ *matrix, where each row* $\mathcal{R}_i$ *is a permutation of the set of candidates* $\mathcal{C}$.

Profiles can naturally be represented with rank matrices (see Example 1). In this context, each row represents a voter's ballot as a linear order over candidates. Given a candidate $c \in \mathcal{C}$ and $k \leq m$, $\mathcal{R}_{i,k} = c$ means that c is in the k^{th} position in the ballot of voter i. Throughout this paper, we will use interchangeably the terms vote profile and rank matrix.

Definition 3. *A* ***partial rank matrix*** $\mathcal{R} = (\mathcal{R}_1, \dots, \mathcal{R}_n)$ *is an* $n \times m$ *matrix with values in* $\mathcal{C} \cup \{null\}$*, such that on every row* $\mathcal{R}_i$ *each element of* $\mathcal{C}$ *appears at most once.*

If $\mathcal{R}_{i,k} = null$, we say that k is a *free entry* in $\mathcal{R}_i$. Conversely, k is a *locked entry* in $\mathcal{R}_i$ if $\mathcal{R}_{i,k} \neq null$. When displayed in our examples, free entries will be represented by middots ($\cdot$). We define the size of a partial rank matrix $\mathcal{R}$ (respectively ballot $\mathcal{R}_i$), noted $|\mathcal{R}|$ (respectively $|\mathcal{R}_i|$), as the number of its non-null entries.

Definition 4. *Given two partial rank matrices* $\mathcal{R}$ *and* $\mathcal{R}'$*, we say that* $\mathcal{R}'$ *is an* ***extension*** *of* $\mathcal{R}$*, denoted* $\mathcal{R} \subseteq \mathcal{R}'$*, if for all* (i, k) *such that* $\mathcal{R}_{i,k} \neq null$ *we have that* $\mathcal{R}_{i,k} = \mathcal{R}'_{i,k}$*. We denote with* $Ext(\mathcal{R})$ *the set of all* ***complete extensions*** *of a partial rank matrix* $\mathcal{R}$*, i.e., extensions of* $\mathcal{R}$ *with no* $null$ *value.*

Given two rank matrices $\mathcal{R}'$ and $\mathcal{R}''$ such that $\mathcal{R}' \subseteq \mathcal{R}''$, we define $\mathcal{R} = \mathcal{R}'' \setminus \mathcal{R}'$ to be the partial rank matrix such that $\mathcal{R}_{i,k} = null$ for all (i, k) such that $\mathcal{R}'_{i,k} \neq null$, and $\mathcal{R}_{i,k} = \mathcal{R}''_{i,k}$ for all other entries. We can now adapt Definition 1 for scoring rules.

Definition 5. *Given a complete rank matrix* $\mathcal{R}$*, a voting rule* F *and a winning candidate* $w \in F(\mathcal{R})$*, an AXp of* $w \in F(\mathcal{R})$ *is a* $\subseteq$*-minimal partial rank matrix* $\mathcal{X}$ *such that* $\mathcal{X} \subseteq \mathcal{R}$ *and*

$$\forall (\mathcal{R}' \in \mathbb{P}), \mathcal{X} \subseteq \mathcal{R}' \implies w \in F(\mathcal{R}'). \tag{3}$$

A CXp of $w \in F(\mathcal{R})$ *is a* $\subseteq$*-minimal partial rank matrix* $\mathcal{Y}$ *such that* $\mathcal{Y} \subseteq \mathcal{R}$ *and*

$$\exists (\mathcal{R}' \in \mathbb{P}), (\mathcal{R} \setminus \mathcal{Y}) \subseteq \mathcal{R}' \text{ and } w \notin F(\mathcal{R}'). \tag{4}$$

Example 2. *Consider the following partial rank matrix:*

$$\mathcal{X}^1 = \begin{bmatrix} A & B & \cdot & \cdot \\ \cdot & C & D & \cdot \\ A & D & \cdot & \cdot \\ \cdot & \cdot & \cdot & B \end{bmatrix}$$

In any complete extension of $\mathcal{X}^1$*, candidate* A *scores 3 points under the Borda rule with the ballot of voter 1, 3 points with voter 3, and at least 1 point with voter 4 (since for that voter,* B *is already in the last position); so the overall score of* A *cannot be less than 7. Conversely,* B *may score* 3 *with voter 2, but cannot score more than 1 with voter 3 (in the second-to-last position), so the overall Borda score for* B *cannot be more than 6. Thus* A *beats* B *with the Borda rule in any complete extension of* $\mathcal{X}^1$*, and it is not difficult to check that neither* C *nor* D *can beat* A*; although there can be ties,* A *is assured to be one of the Borda winners. Moreover, if any of the non-null entries in* $\mathcal{X}^1$ *is freed (i.e., replaced with* $\cdot$*), then it can be shown that there is a complete extension in which* A *is not a Borda winner anymore. Since* $\mathcal{X}^1 \subseteq \mathcal{R}$*, this shows that* $\mathcal{X}^1$ *is an AXp for* $A \in \mathrm{Borda}(\mathcal{R})$*.*

Example 3. *Consider* $\mathcal{Y}^1$ *below and its complement wrt.* $\mathcal{R}$*:*

$$\mathcal{Y}^1 = \begin{bmatrix} A & \cdot & C & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \qquad \mathcal{R} \setminus \mathcal{Y}^1 = \begin{bmatrix} \cdot & B & \cdot & D \\ B & C & D & A \\ A & D & C & B \\ D & C & A & B \end{bmatrix}$$

If $\mathcal{R}^2$ *is the completion of* $\mathcal{R} \setminus \mathcal{Y}^1$ *with* C *in the first position for voter 1, and* A *in the third position, then the Borda score of* A *in* $\mathcal{R}^2$ *is* $1 + 0 + 3 + 1 = 5$*, the score of* C *is* $3 + 2 + 1 + 2 = 8$*, thus* $A \notin \mathrm{Borda}(\mathcal{R}^2)$*. This shows that* $\mathcal{Y}^1$ *is a CXp for* $A \in \mathrm{Borda}(\mathcal{R})$*.*

As observed previously, the feature space we just defined is not composed of independent features (the cells of a rank matrix). The following example calls for a refinement of the notion of AXp.

Example 4. *Consider the two partial rank matrices* $\mathcal{X}^2$ *and* $\mathcal{X}^3$ *below, of which the rank matrix* $\mathcal{R}$ *of Example 1 is a complete extension. Observe that* $\mathcal{X}^1$ *and* $\mathcal{X}^2$ *only differ in the first row:*

$$\mathcal{X}^2 = \begin{bmatrix} \cdot & B & C & D \\ B & \cdot & \cdot & \cdot \\ A & D & \cdot & \cdot \\ \cdot & \cdot & \cdot & B \end{bmatrix} \qquad \mathcal{X}^3 = \begin{bmatrix} A & \cdot & C & D \\ B & \cdot & \cdot & \cdot \\ A & D & \cdot & \cdot \\ \cdot & \cdot & \cdot & B \end{bmatrix}$$

Both $\mathcal{X}^2$ *and* $\mathcal{X}^3$ *are AXps for* $A \in \mathrm{Borda}(\mathcal{R})$*. However, it can be argued that they represent the same explanation, because the first row of* $\mathcal{X}^2$ *and* $\mathcal{X}^3$ *have a single possible extension, namely* $[A\ B\ C\ D]$*. In other words:* $\mathcal{X}^2$ *and* $\mathcal{X}^3$ *represent the same explanation, which is better represented with the full ballot* $[A\ B\ C\ D]$ *on the first row.*

As the example above shows, we can have multiple AXps (up to m) which differ only by one null entry in a row, and this is due to the constraints defining a linear order of candidates. To avoid the redundancy caused by multiple equivalent AXps we propose the following:

Definition 6. *Given a complete rank matrix* $\mathcal{R}$*, a voting rule* F *and a winning candidate* $w \in F(\mathcal{R})$*, an* ***irredundant abductive explanation****, or iAXp, of* $w \in F(\mathcal{R})$ *is a* $\subseteq$*-minimal partial rank matrix* $\mathcal{X}$ *such that* $\mathcal{X} \subseteq \mathcal{R}$*, it verifies equation (3), and such that every row has either none or at least 2* $null$ *entries.*

Formal explanations under constraints have been studied by Cooper and Amgoud [13]. Our *iAXps* correspond to their notion of *subset-minimal, coverage-based prime-implicant explanation (mCPI-Xp)*. Note that an AXp typically is also an *iAXp*, except when it has one or more rows with one null entry only. Given their closeness, in the remainder of the paper we will talk about AXps in general, except when discussing algorithms for their computation.

2.4 Necessary winner and explanations

The well-studied concept of necessary and possible winner of voting rules [24] can be used to obtain a useful equivalent formulation for Definition 5. Recall that given a candidate $c \in \mathcal{C}$, a voting rule F, and a partial rank matrix $\mathcal{R}$, c is a necessary winner if for every extension $\mathcal{R}'$ of $\mathcal{R}$ we have that $c \in F(\mathcal{R}')$, and we write $c \in \text{NW}_F(\mathcal{R})$. Similarly, c is a possible winner for $\mathcal{R}$ and F if there exists an extension $\mathcal{R}'$ of $\mathcal{R}$ such that $c \in F(\mathcal{R}')$, and we write $c \in \text{PW}_F(\mathcal{R})$. It is now straightforward to show the following:

Proposition 1. *Given a complete rank matrix $\mathcal{R}$, a voting rule F and a winning candidate $w \in F(\mathcal{R})$, $\mathcal{X}$ is an AXp of $\mathcal{R}$ iff $\mathcal{X}$ is a $\subseteq$-minimal partial rank matrix s.t. $\mathcal{X} \subseteq \mathcal{R}$ and $w \in NW_F(\mathcal{X})$. Moreover, $\mathcal{Y}$ is a CXp of $\mathcal{R}$ iff $\mathcal{Y}$ is a $\subseteq$-minimal partial rank matrix s.t. $\mathcal{Y} \subseteq \mathcal{R}$ and $w \notin NW_F(\mathcal{R} \setminus \mathcal{Y})$.*

For scoring rules, the problem of deciding whether or not a candidate is a necessary winner can efficiently be solved thanks to a characterization based on candidates' minimal and maximal achievable scores [24]. Given its importance for the rest of the paper, we reframe the original result here with our notation.

Given a (possibly partial) rank matrix $\mathcal{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_n)$, let us denote the score obtained by c in $\mathcal{R}$ as $\sigma_{\mathcal{R}}(c)$ and the score in a single ballot $\mathcal{R}_i$ as $\sigma_{\mathcal{R}_i}(c)$. Clearly, $\sigma_{\mathcal{R}}(c) = \sum_{\mathcal{R}_i \in \mathcal{R}} \sigma_{\mathcal{R}_i}(c)$. For any partial rank matrix $\mathcal{R}$ and any candidate c, we introduce $\sigma^{min}_{\mathcal{R}}(c) = min_{\mathcal{R}' \in Ext(\mathcal{R})} \sigma_{\mathcal{R}'}(c)$ (resp. $\sigma^{max}_{\mathcal{R}}(c) = max_{\mathcal{R}' \in Ext(\mathcal{R})} \sigma_{\mathcal{R}'}(c)$) to be the minimal (resp. maximal) score that candidate c can obtain in any complete extension of $\mathcal{R}$. Konczak and Lang [24] proved the following characterization of necessary winners:

Proposition 2. *Let F be a scoring rule, $\mathcal{R}$ a (possibly partial) rank matrix, and c a candidate. $c \in NW_F(\mathcal{R})$ if and only if for all $c' \in \mathcal{C} \setminus \{c\}$,*

$$\sigma^{min}_{\mathcal{R}}(c) \geq \sigma^{max}_{\mathcal{R}}(c') \tag{5}$$

A more detailed analysis and proof of this proposition can be found in the extended version of this paper [12].

Example 5. *Consider the partial rank matrix $\mathcal{X}^1$ from Example 2. The table below gives the minimal and maximal Borda scores $\sigma^{min}_{\mathcal{X}^1}(A)$ and $\sigma^{max}_{\mathcal{X}^1}(B)$ as well as $\sigma^{max}_{\mathcal{X}^1}(C)$ and $\sigma^{max}_{\mathcal{X}^1}(D)$:*

$$\mathcal{X}^1 = \begin{bmatrix} A & B & \cdot & \cdot \\ \cdot & C & D & \cdot \\ A & D & \cdot & \cdot \\ \cdot & \cdot & \cdot & B \end{bmatrix}$$

	A	B	C	D
$\sigma^{min\|max}_{\mathcal{X}^1}$	min 7	max 6	max 7	max 7

Using Proposition 2 we can infer that $A \in NW_{\text{Borda}}(R^1)$. This, combined with the fact that freeing any other non-null entry in $\mathcal{X}^1$ would decrease $\sigma^{min}_{\mathcal{X}^1}(A)$, or increase $\sigma^{max}_{\mathcal{X}^1}$ for some of the other candidates, shows that $\mathcal{X}^1$ is an AXp of $A \in \text{Borda}(\mathcal{R})$ by Proposition 1.

Similarly, consider the partial rank matrix $\mathcal{Y}^1$ of Example 3. The table below gives minimal and maximal Borda scores for A, B, C and D in any complete extension of $\mathcal{R} \setminus \mathcal{Y}^1$:

$$\mathcal{R} \setminus \mathcal{Y}^1 = \begin{bmatrix} \cdot & B & \cdot & D \\ B & C & D & A \\ A & D & C & B \\ D & C & A & B \end{bmatrix}$$

	A	B	C	D
$\sigma^{min\|max}_{\mathcal{R} \setminus \mathcal{Y}^1}$	min 5	max 5	max 8	max 6

Observe that $\sigma^{min}_{\mathcal{R} \setminus \mathcal{Y}^1}(A) < \sigma^{max}_{\mathcal{R} \setminus \mathcal{Y}^1}(C)$, thus $A \notin NW_{\text{Borda}}(\mathcal{R} \setminus \mathcal{Y}^1)$. Moreover $\mathcal{Y}^1$ is a minimal partial rank matrix with this property, making $\mathcal{Y}^1$ a CXp for $A \in \text{Borda}(\mathcal{R})$ by Proposition 1.

3 Computing explanations for voting rules

Existing algorithms to compute formal explanations range from finding one AXp or CXp, obtaining one (cardinal-wise) smallest AXp or CXp, and enumerating all AXps and CXps (we refer to [29] for a recent survey). All these algorithms are specializations of algorithms used to compute Minimal Unsatisfiable Sets (MUS) and Minimal Correction Sets (MCS) [22]. In this section we adapt these algorithms to the computation of formal explanations for scoring rules.

As previously observed, in our settings the features are not independent, since they are entries of a rank matrix. Hence, our algorithms have to include additional guardrails to ignore redundant explanations generated by the non-independence of our features.

3.1 Computing one explanation

Algorithm 1 finds one CXp for a given rank matrix and one of the winning candidates. Because the non-independence of features does not play a role in the computation of CXps, it is a straightforward adaptation of the classical algorithm for finding contrastive explanations for arbitrary classifiers. Indeed, when computing CXps we look for the minimal amount of matrix entries to remove to be able to change the winner, and removing a single entry in a row of a rank matrix does not change the set of possible extensions. Algorithm 1 goes across each entry of the rank matrix and locks as many as possible while keeping the constraint $w \in \text{NW}_F(\mathcal{R} \setminus \mathcal{Y})$. If this condition is broken, the algorithm rolls back by one step and frees the entry that has just been locked. Since it is easier to verify that $w \in \text{NW}_F(\bar{\mathcal{Y}})$ where $\bar{\mathcal{Y}} = \mathcal{R} \setminus \mathcal{Y}$, the algorithm works on the complement of the CXp, namely $\bar{\mathcal{Y}}$. Due to the monotonicity of explanations (i.e., if we have enough information to declare the winner, having even more information will also be enough to declare the winner), the entry removed and then added back will not have to be tested again later.

Algorithm 1 and, later, Algorithm 2, take a seed in input parameters. This seed will be used for computations in Algorithms 4 and 5 to have some control on the explanation returned. The seed plays the role of a shortcut: one can view it as if all the entries it contains were the first iterations of the for loop on line 2. Since we have $w \notin \text{NW}_F(\mathcal{R} \setminus \mathcal{S})$, we know that no rollback would have occurred. Algorithms 1 and 2 can be run with the whole rank matrix as the seed to output a minimal explanation.

Algorithm 1: FINDCXP – Finding one CXp

Data: Rank matrix $\mathcal{R}$, scoring rule F, winning candidate $w \in F(\mathcal{R})$, partial rank matrix $\mathcal{S}$ as a seed s.t. $\mathcal{S} \subseteq \mathcal{R}$ and $w \notin \text{NW}_F(\mathcal{R} \setminus \mathcal{S})$

Result: a CXp

1 $\bar{\mathcal{Y}} \leftarrow \mathcal{R} \setminus \mathcal{S}$ // $\mathcal{Y} \leftarrow \mathcal{S}$

2 **for** $(i, j) \in [|1, n|] \times [|1, m|]$ *and* $\mathcal{S}_{i,j} \neq null$ **do**

3 $\quad \bar{\mathcal{Y}}_{i,j} \leftarrow \mathcal{S}_{i,j}$ // $\mathcal{Y}_{i,j} \leftarrow null$

4 $\quad$ **if** $w \in NW_F(\bar{\mathcal{Y}})$ **then** // if $w \in \text{NW}_F(\mathcal{R} \setminus \mathcal{Y})$

5 $\quad\quad \bar{\mathcal{Y}}_{i,j} \leftarrow null$ // $\mathcal{Y}_{i,j} \leftarrow \mathcal{S}_{i,j}$

6 **return** $\mathcal{R} \setminus \bar{\mathcal{Y}}$ // return $\mathcal{Y}$

Algorithm 2 finds one irredundant AXp for a given rank matrix and one of the winning candidates. Because of the duality between the definitions of AXp and CXp (see, e.g., [22]), its structure is very similar to Algorithm 1 with additional steps to take into account the non-independence of features. This takes the form of both an additional constraint on the seed (namely that $\forall i$, $|\mathcal{S}_i| \neq m - 1$) and a

Algorithm 2: FINDIAXP – Finding one iAXp

Data: Rank matrix $\mathcal{R}$, scoring rule F, winning candidate $w \in F(\mathcal{R})$, partial rank matrix $\mathcal{S}$ as a seed s.t. $\mathcal{S} \subseteq \mathcal{R}$, $w \in \mathrm{NW}_F(\mathcal{S})$ and $\forall i$, $|\mathcal{S}_i| \neq m-1$
Result: an iAXp

1 $\mathcal{X} \leftarrow \mathcal{S}$
2 **for** $(i,j) \in [|1,n|] \times [|1,m|]$ *and* $\mathcal{S}_{i,j} \neq null$ **do**
3     $\mathcal{X}_{i,j} \leftarrow null$
4   **if** $|\mathcal{X}_i| = m-1$ **then**
5   ENSUREIRR$(\mathcal{R}, F, w, \mathcal{S}, \mathcal{X}, i, j)$
6   **else**
7   **if** $w \notin NW_F(\mathcal{X})$ **then**
8             $\mathcal{X}_{i,j} \leftarrow \mathcal{S}_{i,j}$
9 **return** $\mathcal{X}$

Algorithm 3: ENSUREIRR – Ensuring irredundancy of the iAXp currently computed

Data: Rank matrix $\mathcal{R}$, scoring rule F, winning candidate $w \in F(\mathcal{R})$, partial rank matrix $\mathcal{S}$, partial rank matrix $\mathcal{X}$ s.t. $\mathcal{X} \subseteq \mathcal{S}$, $w \in \mathrm{NW}_F(\mathcal{X})$, index i s.t. $|\mathcal{X}_i| = m-1$, index j s.t. $\mathcal{X}_{i,j} = null$

1 **for** $j_2 \in [|1,m|]$ *and* $j_2 \neq j$ **do**
2     $\mathcal{X}_{i,j_2} \leftarrow null$
3   **if** $w \notin NW_F(\mathcal{X})$ **then**
4         $\mathcal{X}_{i,j_2} \leftarrow \mathcal{S}_{i,j_2}$
5   **else**
6   **break**
7 $\mathcal{X}_{i,j} \leftarrow \mathcal{S}_{i,j}$

sub-function ENSUREIRR described in Algorithm 3, that makes sure that no row in the resulting explanation has one only free entry. This mechanism triggers when, for the first time, an entry is freed in a row (line 4 of Algorithm 2). In that case, Algorithm 3 looks for another entry to free in the same ballot. If one is found the execution goes on normally (break instruction on line 6) and if none, the entry initially freed belongs to the AXp (line 7 after the main for loop).

Proposition 3. *Finding one CXp (Algorithm 1) can be done in $\Theta(nm^2)$ and finding one iAXp (Algorithm 2) in $\Theta(nm^3)$.*

Proof sketch. Algorithm 1 goes across all the nm entries of the rank matrix, tries to remove it and checks if it changes the outcome of the necessary winner test. Algorithm 2 works similarly but for the addition of Algorithm 3, which goes through a whole row adding a complexity factor of m in complexity. Hence, the complexity of finding explanations greatly depends on our ability to decide whether or not a candidate is a necessary winner of a partial rank matrix. To do this, we use the characterisation introduced in Proposition 2. Since rows of a rank matrix are independent from each other, the minimal (resp. maximal) score can be decomposed as the sum of minimal (resp. maximal) scores on each row: $\sigma_{\mathcal{R}}^{min}(c) = \sum_{\mathcal{R}_i \in \mathcal{R}} \sigma_{\mathcal{R}_i}^{min}(c)$ (resp. $\sigma_{\mathcal{R}}^{max}(c) = \sum_{\mathcal{R}_i \in \mathcal{R}} \sigma_{\mathcal{R}_i}^{max}(c)$). It is possible to compute minimal and maximal possible scores of all candidates by simply scanning through the whole row of length m. By repeating it for the n rows, we have a straightforward algorithm to check if a candidate is a necessary winner which runs in $\Theta(nm)$. However, in our setting, it is possible to improve this bound. Observe that the necessary winner check in our algorithms is done repeatedly on rank matrices differing by only one entry. We can therefore keep in memory the minimal and maximal possible scores for all rows, and at each step of the for loop simply update the score of the row that has been modified, which as discussed above can be done in $\Theta(m)$. □

3.2 Enumeration and smallest explanations

Algorithm 4 enumerates all iAXps and CXps by exploring the whole search space. It iteratively generates a seed which will lead to a new CXp or iAXp with a call to an NP oracle. In this case, the problem of finding new instances to explore is encoded in a SAT formula. Every entry of the rank matrix is associated to a literal where a *True* assignment means that the entry is fixed. A SAT solver is then used to efficiently find new instances thanks to the monotonicity of explanations. An upper bound on the total number of AXps and CXps is $2\binom{nm}{\lfloor \frac{nm}{2} \rfloor}$, which can be derived via Sperner's Theorem (for a deeper analysis of this mechanism see [26]). Finally, to prevent issues with the non-independence of our features in Algorithm 4 and later Algorithm 5, instances containing a row with only one free entry are removed (lines 2 to 3). The added clauses can be read as: for each entry i, j_0, if x_{i,j_0} is false (the entry is free), then $\bigvee_{j \neq j_0} \neg x_{i,j}$ is true (at least another entry in the row is free).

Algorithm 4: ENUMXP – Enumerating all iAXps and CXps

Data: Rank matrix $\mathcal{R}$, scoring rule F, winning candidate $w \in F(\mathcal{R})$
Result: List of all iAXps and CXps

1 *Map* $\leftarrow$ True
2 **for** $(i, j_0) \in [|1,n|] \times [|1,m|]$ **do**
3   *Map* $\leftarrow$ *Map* $\wedge \left(x_{i,j_0} \vee \bigvee_{j \neq j_0} \neg x_{i,j}\right)$
4 **while** *Map is satisfiable* **do**
5     $u \leftarrow$ SAT(*Map*)
6     $\mathcal{S} \leftarrow \{\mathcal{R}_{i,j}$ if $u_{i,j}$ else $null$; $(i,j) \in [|1,n|] \times [|1,m|]\}$
7   **if** $w \in NW_F(\mathcal{S})$ **then**
8   *iAXp* $\leftarrow$ FINDIAXP$(\mathcal{R}, F, w, \mathcal{S})$
9   *iAXps* $\leftarrow$ *iAXps* $\cup$ *iAXp*
10   *Map* $\leftarrow$ *Map* $\wedge \left(\bigvee_{(i,j) \in iAXp} \neg x_{i,j}\right)$
11   **else**
12   *CXp* $\leftarrow$ FINDCXP$(\mathcal{R}, F, w, \mathcal{S})$
13   *CXps* $\leftarrow$ *CXps* $\cup$ *CXp*
14   *Map* $\leftarrow$ *Map* $\wedge \left(\bigvee_{(i,j) \in CXp} x_{i,j}\right)$
15 **return** *(iAXps ,CXps)*

When selecting one of the many possible formal explanations we will be interested in selecting the smallest one. Recall that the size of a partial rank matrix is the number of its non-null entries. We can therefore define the set of smallest iAXps, denoted as SiAXp, as the $\mathrm{argmin}_{\{\mathcal{X} | \mathcal{X} \text{ is iAXp of } w \in F(\mathcal{R})\}} |X|$. Finding a smallest CXp is relatively easy and can be done by computing a smallest cost solution to a SAT problem. However, finding a SiAXp is generally harder. Algorithm 5 is based on a previous solution which uses the hitting set duality of MUS and MCS and iteratively computes minimum (cardinal-wise) solutions to a hitting set problem [21].

4 Smallest abductive explanations for Borda

One of the main challenges of dealing with formal explanations is the existence of a large number of possible explanations. Indeed, the

Algorithm 5: FINDSIAXP – Finding one smallest iAXp

Data: Rank matrix $\mathcal{R}$, scoring rule F, winning candidate $w \in F(\mathcal{R})$
Result: a smallest iAXp

Map $\leftarrow$ True
for $(i, j_0) \in [|1, n|] \times [|1, m|]$ **do**
  Map $\leftarrow$ *Map* $\wedge \left(x_{i,j_0} \vee \bigvee_{j \neq j_0} \neg x_{i,j}\right)$
while *true* **do**
  $u \leftarrow$ MINIMUMHITTINGSET(*Map*)
  $\mathcal{S} \leftarrow \{\mathcal{R}_{i,j}$ if $u_{i,j}$ else $null$; $(i,j) \in [|1,n|] \times [|1,m|]\}$
  if $w \in NW_F(\mathcal{S})$ **then**
    return $\mathcal{S}$
  else
    CXp $\leftarrow$ FINDCXP$(\mathcal{R}, F, w, \mathcal{S})$
    Map $\leftarrow$ *Map* $\wedge \left(\bigvee_{(i,j) \in CXp} x_{i,j}\right)$

simple rank matrix introduced in Example 1 already has 14 different iAXps and 17 CXps. We also run preliminary experiments on a simplified version of the preference profile used by Peters et al. [33] as their main example, with 8 voters and 4 candidates, and we obtained 3244 iAXps and 321 CXps. As iAXps and CXps can be used to improve trust and understanding of a voting rule, the size of the *smallest* iAXps is of primary importance. In this section we provide tight lower bounds on sAXps complemented by experimental results using a suitably defined map of elections. Results in this section are restricted to the study of smallest AXps for the Borda rule. Naturally, they also apply to smallest iAXps.

4.1 The size of abductive explanations for Borda

We first give a lower bound on the size of abductive explanations for the Borda rule. The proof is non-trivial, and uses a suitably defined notion of normal form for ballots of rank matrices, hinging on the characterisation of necessary winners proved in Proposition 2. We give a sketch of the proof here, a detailed proof can be found in the extended version of this paper [12].

Theorem 4. *Let $\mathcal{R}$ be a rank matrix with n voters and m candidates s.t. $w \in \mathcal{C}$ is a Borda winner of $\mathcal{R}$. For all AXp $\mathcal{X}$ of $\mathcal{R}$, we have:*

$$|\mathcal{X}| \geq n - \left\lfloor \frac{n}{m} \right\rfloor \tag{6}$$

Proof sketch. First, we observe that to bound the size of AXps of an arbitrary rank matrix, we can construct a different rank matrix which admits an AXp of size smaller or equal to the original matrix. This observation is at the basis of our normal form construction below. Second, let a *weak AXp* be any partial rank matrix verifying constraint (1) but that is not necessarily minimal. Because of subset-minimality, smallest weak AXps are AXps, and since AXps are also weak AXps, then proving (6) restricted to weak AXps is equivalent to proving the bound in Theorem 4.

Our proof then uses the characterisation proved in Proposition 2: if $\mathcal{X}$ is an AXp for $w \in \mathrm{Borda}(\mathcal{R})$, w is a necessary winner of $\mathcal{X}$ if and only if $\sigma^{min}_{\mathcal{X}}(w) \geq \sigma^{max}_{\mathcal{X}}(c')$ for every other candidate c'. We now define $\Delta^w_{\mathcal{X}}(c') = \sigma^{min}_{\mathcal{X}}(w) - \sigma^{max}_{\mathcal{X}}(c')$, and we call $\Delta^w_{\mathcal{X}} = \sum_{c' \neq c} \Delta^w_{\mathcal{X}}(c')$ the total margin of victory. Since w is a necessary winner for $\mathcal{X}$, then $\Delta^w_{\mathcal{X}} \geq 0$. We also show that the computation of the total margin of victory can be decomposed as the sum of *total score margins*, computed ballot by ballot. Hence, in the start of the proof we will work at the level of a single ballot.

The main steps of the proof continues as follows, starting from any AXp $\mathcal{X}$ of an arbitrary rank matrix:

1. We prove that there exists a ballot $\mathcal{X}'_i$ (possibly part of a different weak AXp) which has a particular form, which we call "normal form", such that $|\mathcal{X}'_i| = |\mathcal{X}_i|$ and $\Delta^w_{\mathcal{X}'_i} \geq \Delta^w_{\mathcal{X}_i}$. This normal form is obtained by repeated permutations of candidates, to bring (or insert) w at the top of the ballot, and "descending" some other candidates as much as possible while not decreasing Δ^w.
2. Thanks to our definition of normal form, we are able to prove that $\Delta^w_{\mathcal{X}'_i} \leq (m-1)(m|\mathcal{X}'_i| - n(m-1))$.
3. Summing on voters we have $\Delta^w_{\mathcal{X}'} \leq (m-1)(m|\mathcal{X}'| - n(m-1))$, which implies that $\Delta^w_{\mathcal{X}} \leq (m-1)(m|\mathcal{X}| - n(m-1))$.
4. Recall that $0 \leq \Delta^w_{\mathcal{X}}$ since w is a necessary winner, thus we have $0 \leq (m-1)(m|\mathcal{X}| - n(m-1))$ which implies the result. □

We are then able to show that the lower bound of Theorem 4 is tight: let $\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$ be the partial rank matrix where every entry is $null$ except that $\mathcal{X}_{i,1} = w$ for $1 \leq i \leq n - \left\lfloor \frac{n}{m} \right\rfloor$. It is not difficult to check that the minimum possible score for w is always greater than, or equal to, the maximum possible score for all other candidates. By Theorem 4, $\mathcal{X}$ is minimal and therefore it is an AXp for any complete profile $\mathcal{R}$ that extends it.

Theorem 5. *Let $\mathcal{R}$ be a rank matrix with n voters and m candidates and $w \in \mathcal{C}$ a Borda winner of $\mathcal{R}$ s.t. $\mathcal{X} \subseteq \mathcal{R}$ where $\mathcal{X}$ is the partial rank matrix where w is ranked first for $n - \left\lfloor \frac{n}{m} \right\rfloor$ ballots and every other entry is null. $\mathcal{X}$ is an AXp of $\mathcal{R}$ and $|\mathcal{X}| = n - \left\lfloor \frac{n}{m} \right\rfloor$.*

What can be concluded from Theorems 4 and 5 is that even the smallest AXps of a preference profile are rather long to be considered compact explanations, at least for large voter populations—the bound is linear in the number of voters—opening an interesting challenge of identifying feature spaces that result in more compact explanations. In Section 5 we discuss this point in more details.

4.2 A map of elections for SiAXps for Borda

To get a clearer picture of the sizes of SiAXps of the Borda rule we set up a map of elections using the tools by Szufa et al. [36]. A map of elections starts from a set of profiles, generated using known preference distributions, and generates a 2D embedding of the space obtained by calculating the isomorphic swap-distance between the profiles. Following their approach, to improve the interpretability of the map we introduce specific extreme profiles to act as a compass [3, 18]. Our data set is composed of 146 profiles with 4 candidates and 12 voters generated from 17 different cultures. The details of our dataset can be found in the extended version of this paper [12]. Preliminary experiments with higher number of voters and candidates led to excessive execution time for Algorithm 5.

The first map in Figure 1 plots the map of elections for our dataset on the left, which is coherent with those obtained by Szufa et al. [36] in past research. The compass profiles in red are the identical culture (ID) where all ballots are identical, the antagonism culture (AN) where half of the ballots are identical and the other half is their opposite, and the uniform culture (UN) where ballots are drawn uniformly at random among all the possible ones (there are four of them to account for randomness). Ten instances were generated for each of the remaining fourteen cultures. For a detailed description of

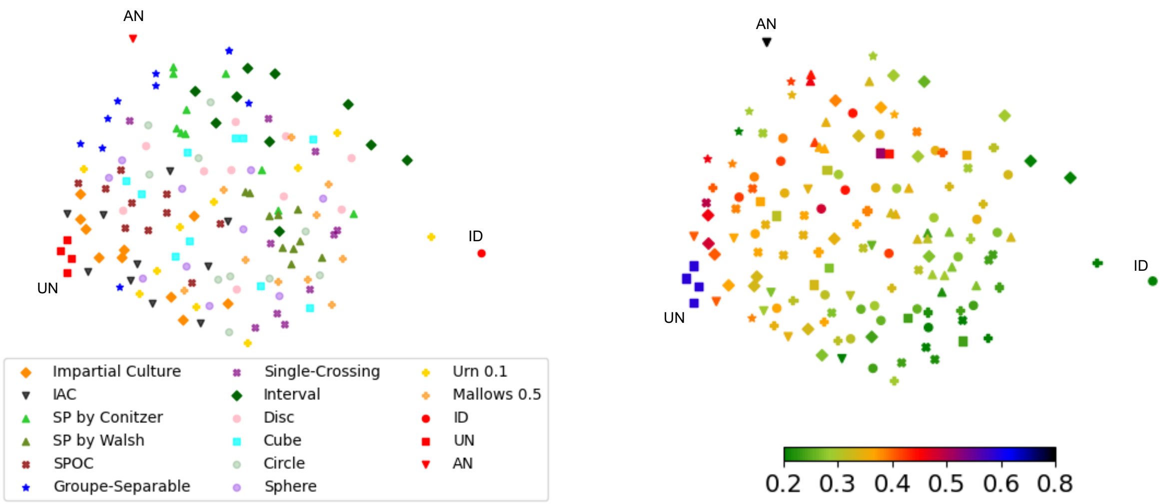

Figure 1. Map of elections for 146 preference profiles generated by fourteen different cultures plus six compass profiles (AN, ID, and four for UN). The map on the right shows for each preference profile the size of its SiAXp, normalised as $\frac{|SiAXp|}{nm}$.

statistical cultures for preference profile generation used by maps of elections we refer to the presentation by Szufa et al. [36].

The second map in Figure 1 shows the size of the SiAXp for each of the profile in the dataset. We can observe that the antagonism and uniform profiles have the largest SiAXp in our data set, and the identical profile has the smallest one (in coherence with our Theorem 5). When considering prefence distributions, we can observe that impartial culture (IC) generates profiles with long SiAXps, given their closeness to uniform profiles. A similar observation can be drawn for the impartial anonymous culture (IAC) and single-peaked on a circle (SPOC). A surprising observation is that the generation of single-peaked profiles have a strong impact on the size of SiAXps, as can be seen by the Conitzer-SP profiles which have large SiAXps while the Walsh-SP profiles have short SiAXps. The well-known Urn and Mallow models (plotted here for one specific parameter only), have relatively small SiAXps with some exceptions.

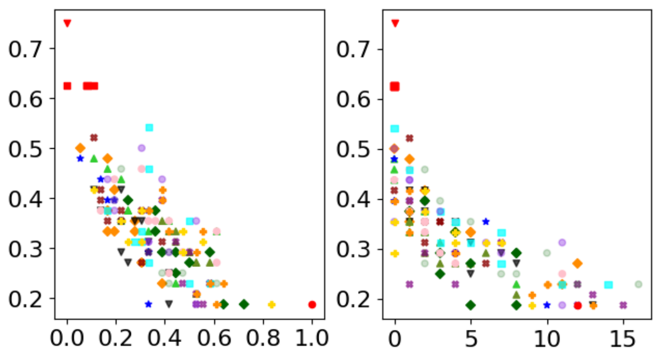

Figure 2. Normalised SiAXp size on the x-axis compared with agreement index (left) and margin of victory (right) for the 146 profiles in our dataset.

To investigate closely which properties of a preference profile are indicators of a large SiAXp, Figure 2 compares the size of SiAXps with two measures: the *margin of victory*, which is the difference in Borda score between the winner and the second-best candidate in a profile, and the (normalized) *agreement index*, formally defined as $\sum_{a,b\in\mathcal{C},\, a\neq b} |N_{a>b} - N_{b>a}| / \left(n\binom{|\mathcal{C}|}{2}\right)$ where $N_{a>b}$ is the number of voters preferring candidate a to candidate b. The agreement index was studied recently by Faliszewski et al. [18], based on previous work by Hashemi and Endriss [20] and Can et al. [9].

We observe that the size of the SiAXp is negatively correlated in our dataset with both the agreement index and the margin of victory. Spearman test gives respectively a correlation coefficient of -0.761 (p-value smaller than 0.001) and -0.828 (again, $p < 0.001$). Both results have an intuitive interpretation. Elections with a small margin of victory require more information to identify which of two candidates is the winner. Similarly, the less the voters agree the more the candidates' scores are similar, and the more information is needed to determine the winner of the election.

5 Conclusions and future work

A primary direction for future research is to test formal explanations on voting rules defined on the majority graph, devising a feature encoding that result in compact formal explanations. While the axiomatic calculus devised by Nardi et al. [31] leads to human-readable explanations supporting the choice of the winner in an election, our proposal of adapting formal explanations for scoring rules does not seem fit for this application, due to the large number of possible explanations and the size of smallest explanations, which is high as soon as the preference profile shows some complexity.

A surprising connection can be shown between formal explanations and bribery in voting, with CXps identifying the positions in the rankings that allow for a minimal destructive bribery attack, and AXps defining optimal protection against such attacks [28]. Formal explanations in voting can also be used to study optimal preference elicitation strategies to compute the winner of a given election. This problem has been well-studied in the literature as the communication complexity [10] and the sample complexity [15] of voting rules, with the size of smallest abductive explanations having a natural correspondence with the former concept.

References

[1] C. Anil and X. Bao. Learning to elect. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[2] J. Behrens, A. Kistner, A. Nitsche, and B. Swierczek. *The principles of LiquidFeedback*. Interacktive Demokratie, 2014.

[3] N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, and S. Szufa. Putting a compass on the map of elections. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[4] A. Boixel, U. Endriss, and R. de Haan. A calculus for computing structured justifications for election outcomes. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[5] A. Boixel, U. Endriss, and O. Nardi. Displaying justifications for collective decisions. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. Demo Paper.

[6] S. J. Brams and P. C. Fishburn. Chapter 4: Voting procedures. In *Handbook of Social Choice and Welfare*, volume 1 of *Handbook of Social Choice and Welfare*, pages 173–236. Elsevier, 2002.

[7] D. Burka, C. Puppe, L. Szepesváry, and A. Tasnádi. Voting: A machine learning approach. *European Journal of Operational Research*, 299(3): 1003–1017, 2022.

[8] O. Cailloux and U. Endriss. Arguing about voting rules. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2016.

[9] B. Can, A. I. Ozkes, and T. Storcken. Measuring polarization in preferences. *Mathematical Social Sciences*, 78:76–79, 2015.

[10] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In J. Riedl, M. J. Kearns, and M. K. Reiter, editors, *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*, 2005.

[11] V. Conitzer, R. Freedman, J. Heitzig, W. H. Holliday, B. M. Jacobs, N. Lambert, M. Mossé, E. Pacuit, S. Russell, H. Schoelkopf, E. Tewolde, and W. S. Zwicker. Social choice should guide ai alignment in dealing with diverse human feedback. arXiv:2404.10271, 2024.

[12] C. Contet, U. Grandi, and J. Mengin. Abductive and contrastive explanations for scoring rules in voting. arXiv:2408.12927, 2024.

[13] M. C. Cooper and L. Amgoud. Abductive explanations of classifiers under constraints: Complexity and properties. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*, 2023.

[14] A. Darwiche and A. Hirth. On the reasons behind decisions. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*. 2020.

[15] P. Dey and A. Bhattacharyya. Sample complexity for winner prediction in elections. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015.

[16] S. Ebadian, A. Filos-Ratsikas, M. Latifian, and N. Shah. Explainable and efficient randomized voting rules. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[17] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016.

[18] P. Faliszewski, A. Kaczmarczyk, K. Sornat, S. Szufa, and T. Wąs. Diversity, agreement, and polarization in elections. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[19] J. Gudiño-Rosero, U. Grandi, and C. A. Hidalgo. Large language models (llms) as agents for augmented democracy. arXiv:2405.03452, 2024.

[20] V. Hashemi and U. Endriss. Measuring diversity of preferences in a group. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 2014.

[21] A. Ignatiev, A. Previti, M. Liffiton, and J. Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*, 2015.

[22] A. Ignatiev, N. Narodytska, N. Asher, and J. Marques-Silva. On relating 'why?' and 'why not?' explanations. arXiv:2012.11067, 2020.

[23] I. Kang, Q. Han, and L. Xia. Learning to explain voting rules. In N. Agmon, B. An, A. Ricci, and W. Yeoh, editors, *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

[24] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI Workshop on Advances in Preference Handling*, 2005.

[25] M. K. Lee, D. Kusbit, A. Kahng, J. T. Kim, X. Yuan, A. Chan, D. See, R. Noothigattu, S. Lee, A. Psomas, et al. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on human-computer interaction*, 2019.

[26] M. H. Liffiton, A. Previti, A. Malik, and J. Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2), 2016.

[27] X. Liu and E. Lorini. A unified logical framework for explanations in classifier systems. *Journal of Logic and Computation*, 33(2):485–515, 2023.

[28] Y. Lu, W. Shi, and N. Shah. Computational complexity characterization of protecting elections from bribery. In *Proceedings of the 26th International Conference on Computing and Combinatorics (COCOON)*, 2020.

[29] J. Marques-Silva. Logic-based explainability in machine learning. In *Tutorial Lectures of the 18th International Summer School on Reasoning Web*, 2022.

[30] T. Miller, R. R. Hoffman, O. Amir, and A. Holzinger. Special issue on explainable artificial intelligence (XAI). *Artificial Intelligence*, 307: 103705, 2022.

[31] O. Nardi, A. Boixel, and U. Endriss. A graph-based algorithm for the automated justification of collective decisions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022.

[32] C. Navarrete, M. Macedo, R. Colley, J. Zhang, N. Ferrada, M. E. Mello, R. Lira, C. Bastos-Filho, U. Grandi, J. Lang, et al. Understanding political divisiveness using online participation data from the 2022 French and Brazilian presidential elections. *Nature Human Behaviour*, 8(1): 137–148, 2024.

[33] D. Peters, A. D. Procaccia, A. Psomas, and Z. Zhou. Explainable voting. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[34] D. Peters, G. Pierczyński, and P. Skowron. Proportional participatory budgeting with additive utilities. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[35] A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[36] S. Szufa, P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Drawing a map of elections in the space of statistical cultures. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

[37] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law and Technology*, 31:841, 2017.

[38] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research (JAIR)*, 41:25–67, 2011.

[39] W. S. Zwicker. Introduction to the theory of voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 23–56. Cambridge University Press, 2016.