Representation with Incomplete Votes

Daniel Halpern, Gregory Kehne, Ariel D. Procaccia, Jamie Tucker-Foltz and **Manuel Wüthrich**

Harvard University

Abstract

Platforms for online civic participation rely heavily on methods for condensing thousands of comments into a relevant handful, based on whether participants agree or disagree with them. These methods should guarantee fair representation of the participants, as their outcomes may affect the health of the conversation and inform impactful downstream decisions. To that end, we draw on the literature on approval-based committee elections. Our setting is novel in that the approval votes are incomplete since participants will typically not vote on all comments. We prove that this complication renders non-adaptive algorithms impractical in terms of the amount of information they must gather. Therefore, we develop an adaptive algorithm that uses information more efficiently by presenting incoming participants with statements that appear promising based on votes by previous participants. We prove that this method satisfies commonly used notions of fair representation, even when participants only vote on a small fraction of comments. Finally, an empirical evaluation using real data shows that the proposed algorithm provides representative outcomes in practice.

1 Introduction

A recent surge of interest in empowering citizens through online civic participation has spurred the development of a number of platforms (Salganik and Levy 2015; Ito et al. 2020; Shibata et al. 2019; Fishkin et al. 2019; Aragón et al. 2017; Iandoli, Klein, and Zollo 2009). A particularly successful example is *Polis* (Small et al. 2021),[1] an open-source "system for gathering, analyzing, and understanding what large groups of people think in their own words." It has been widely used by local and national government agencies around the world. Most notably, it is the basis of vTaiwan, a system commissioned by the government of Taiwan, whose participatory process — involving thousands of ordinary citizens — has led to new regulation of ride-sharing services and financial technology. A similar (albeit commercial) system called *Remesh*[2] allows users to "save resources by democratizing insights in live, flexible conversations with up to 1,000 people at the same time."[3]

The key idea underlying both systems is simple and broadly applicable: Participants can submit free-text comments about the discussion topic at hand and choose to agree or disagree with others' comments presented to them by the platform. An essential part of the process is the aggregation of these opinions toward an "understanding of what large groups of people think." Polis, for instance, displays a list of comments that received the most support among participants to whom they were shown. But this aggregation method may fail to represent minority groups, even those that are very large: if 51% of participants agree with one set of comments, while 49% of participants agree with another set of comments, only comments from the first set will appear on this list. Polis has recognized this problem and sought to mitigate it by employing a second, more elaborate procedure (Small et al. 2021).[4] While this procedure has produced interesting results in practice, it does not guarantee summarizations that are representative of the discussion in a rigorous sense.

In this paper, we reexamine opinion aggregation in systems such as Polis and Remesh through the lens of computational social choice (Brandt et al. 2016). We observe that *selecting a subset of comments based on agreements and disagreements is equivalent to electing a committee based on approval votes*. From this viewpoint, the primary aggregation method used by Polis corresponds to classical approval voting (AV). There is substantial work — starting with the paper of Aziz et al. (2017) — on *approval-based committee elections* that seeks to avoid the shortcomings of AV by guaranteeing that the selected committee satisfies fairness notions. To define one such notion (which is not satisfied by AV), note that if the size of the committee is k and the number of voters is n, a subset of n/k voters is large enough to demand a seat on the committee if they agree on at least one candidate. This intuition is captured by a property called *justified representation (JR)*, which guarantees that every such subset of voters has an approved candidate on

[1] https://pol.is

[2] https://www.remesh.ai

[3] See also consider.it, citizens.is, make.org, kialo.com.

[4] sThe idea is to find clusters of participants with similar opinions and then ensure that each cluster is represented by comments that distinguish it from the others.

the committee.

There is a major gap, however, between the literature on approval-based committee elections and the reality of systems like Polis and Remesh: these systems only have access to partial votes. For example, in the discussion facilitated by Polis around ridesharing regulation in Taiwan, 197 comments were submitted, but each participant only cast a vote on 10.57 comments on average — roughly 5% of all comments. Our main conceptual insight is that we can overcome the partial-information gap via statistical estimation and adaptive querying (i.e., by deciding which comments to show to incoming users based on previous votes).[5]

Our approach and results. In our model, each voter (user) can be asked to express their opinion (approval/disapproval) about at most t candidates (comments). More formally, a *query* asks a randomly-chosen voter for their approval votes on a subset of candidates S of size $|S| \leq t$. Note that this query model is consistent with how Polis works, where participants express their agreement or disagreement with the comments shown to them by the system. We can view the response to such a query, i.e., the approval votes of a single voter, as noisy information about the profile of the entire population of voters (restricted to these t candidates). Therefore, we refer to these real-world queries as *noisy queries*.

Before we discuss this setting, we consider a simplified setting in Section 3, where queries yield the profile of the entire population of voters on the t candidates in the query. While such *exact queries* are not realistic, they provide an abstraction that is easier to study and allows us to derive lower bounds on the number of queries required to achieve JR (which apply also to the noisy-query setting, since it is strictly harder). We start by studying the required number of queries of *non-adaptive* algorithms, which decide on their queries before any votes are cast. While non-adaptive algorithms may be preferable in some cases (e.g., because no voter can influence what alternatives are shown to other voters or because computation can be performed offline), we show that they are impractical because they must ask at least $\Omega(m^{11})$ queries (and hence voters) to achieve JR, where m is the number of candidates.

Therefore, we focus on *adaptive* algorithms in the rest of the paper. In Section 3.2 we adapt a local-search algorithm of Aziz et al. (2018) to the case of exact queries and show that it can achieve JR (and even stronger properties) with $\mathcal{O}(mk^2 \log k)$ queries.

In Section 4, we move on to the realistic, noisy-query model, where a query corresponds to a single voter. Since we need to estimate the answer to each exact query using multiple noisy queries to control uncertainty, the query complexity of the adaptive algorithm for the same guarantees increases to $\mathcal{O}\left(mk^6 \log k \log m\right)$. By applying martingale theory, we develop an extension of this algorithm that allows the reuse of votes in a statistically sound way.

In Section 5 we show empirically (on real datasets from Polis and Reddit) that this extension allows us to find committees satisfying (approximate) JR (and stronger properties) despite access to little information (i.e., few voters, each voting on only a small fraction of the comments).

[5]There is a body of work in computational social choice related to incomplete votes. For instance, some papers aim to find winning committees, given incomplete approval votes, or to fill in the missing votes, given knowledge about the domain of approval profiles (Imber et al. 2022; Terzopoulou, Karpov, and Obraztsova 2021; Zhou, Yang, and Guo 2019). However, these papers are primarily concerned with the computational complexity of these problems, while we focus on information-theoretic questions. There is also related work that studies the problem of determining the winner given only partial rankings (Xia and Conitzer 2011; Filmus and Oren 2014), but this setting is mathematically different from ours. Furthermore, prior work does not consider the adaptive setting, where we query voters sequentially and decide on the next question based on previous votes.

2 Preliminaries

We begin by introducing the standard approval-based committee selection setting (Aziz et al. 2017). For $s \in \mathbb{N}$, we use the notation $[s] = \{1, \ldots, s\}$. We have a set $N = [n]$ of n voters and a set C of m candidates. Each voter $i \in N$ approves a set of candidates $A_i \subseteq C$. We refer to the vector $\mathbf{A} = (A_1, \ldots, A_n)$ as an *approval profile*. The goal is to choose a *committee* $W \subseteq C$ of size $k \leq m$. The value k is called the *target committee size*. We refer to an algorithm that takes as input the profile and candidates and outputs a committee of size k as a *k-committee-selection algorithm*.

Notions of representation. We say that a group of voters $V \subseteq N$ is *ℓ-large* if $|V| \geq \ell \cdot \frac{n}{k}$; V is *ℓ-cohesive* if $|\bigcap_{i \in V} A_i| \geq \ell$. Aziz et al. (2017) introduced the following two fairness notions:

Definition 2.1 (Justified Representation (JR))**.** *A committee W provides JR if for every 1-large, 1-cohesive group of voters V, there exists a voter $i \in V$ who approves a member of W, i.e., $|A_i \cap W| \geq 1$.*

Definition 2.2 (Extended Justified Representation (EJR))**.** *A committee W provides EJR if for every $\ell \in [k]$ and every ℓ-large, ℓ-cohesive group of voters V, there exists a voter $i \in V$ who approves at least ℓ members of W, i.e., $|A_i \cap W| \geq \ell$.*

We also study the following approximate version of EJR:

Definition 2.3 (α-Extended Justified Representation (α-EJR))**.** *A committee W provides α-EJR if for every $\ell \in [k]$ and every $\frac{\ell}{\alpha}$-large, ℓ-cohesive group of voters V, there exists a voter $i \in V$ who approves at least ℓ members of W, i.e., $|A_i \cap W| \geq \ell$.*

Fernández et al. (2017) proposed another notion of representation called the *average satisfaction* of a group of voters V for a committee W, defined as $\text{avs}_W(V) = \frac{1}{|V|} \sum_{i \in V} |A_i \cap W|$. Related to this quantity, we define the following property:

Definition 2.4 (α-Optimal Average Satisfaction (α-OAS))**.** *A committee W provides α-OAS if for every $\lambda \in [0, k]$ and every $\frac{\lambda+1}{\alpha}$-large, λ-cohesive group of voters V, we have $avs_W(V) \geq \lambda$.*

This property measures how close a committee is to the maximum average satisfaction that can be guaranteed to hold for all elections. To see this, note that the condition above is equivalent to the following condition: for every $\ell \in [\frac{1}{\alpha}, \frac{k+1}{\alpha}]$ and every ℓ-large, $(\alpha\ell - 1)$-cohesive group of voters V, we have $\text{avs}_W(V) \geq \alpha\ell - 1$. This implies a *proportionality guarantee* (Skowron 2021) of $g(\ell, k) = \alpha\ell - 1$. Since there is no selection rule that satisfies a proportionality guarantee with $g(\ell, k) > \ell - 1$ for all elections (Aziz et al. 2018; Skowron 2021), $\alpha = 1$ is the best we can hope for, so we refer to 1-OAS simply as OAS.

Proportional approval voting. *Proportional Approval Voting (PAV)* is a widely-studied committee selection algorithm: given an approval profile $\mathbf{A}$ and a committee size k, it returns a committee W of size k maximizing the *PAV score*, defined as

$$\text{PAV-SC}(W) := \frac{1}{n} \sum_{i \in N} \sum_{j=1}^{|A_i \cap W|} \frac{1}{j}.$$

PAV satisfies EJR and OAS (Fernández et al. 2017; Aziz et al. 2018), but is NP-hard to compute (Aziz et al. 2015). Consequently, Aziz et al. (2018) propose a local search approximation of PAV (LS-PAV), which continues to satisfy EJR and OAS, but, unlike PAV, runs in polynomial time. As we shall see, LS-PAV is a useful basis for algorithms in our query model.

3 Exact Queries

In the exact-query setting, the response R to a query Q consists of a proportion p_S for every subset $S \subseteq Q$, where p_S is the proportion of voters who only approve the candidates in S among the queried candidates Q, i.e.,

$$p_S := \frac{1}{n} \sum_{i \in N} \mathbb{I}[A_i \cap Q = S],$$

where $\mathbb{I}$ is the indicator function. We refer to an algorithm that makes queries of size t, receives this type of response, and outputs a committee of size k as a *(k,t)-committee selection algorithm with exact queries*. We say an algorithm is *adaptive* if the queries it chooses depend on responses from previous queries. Note that we allow all of our algorithms to be randomized. In the following, we ask how many queries are needed to guarantee the notions of representation introduced in Section 2.

3.1 Nonadaptive Algorithms

In this section, we think of m as large (many comments will be submitted to the system), while we think of k and t as small constants (since we wish to select only a few comments and voters have limited time). Since we are primarily interested in *lower* bounds on the query complexity of non-adaptive algorithms, we consider only JR, the weakest fairness criterion.

An initial observation is that, if $t \geq k$, JR can always be guaranteed with $O(m^k)$ queries, as simply querying every set of k candidates provides all the information necessary to run PAV. For $k = 1$, this bound is tight, as voters could all approve only a single candidate, which will take a linear number of queries to find. Our first result is a tight quadratic lower bound for $k = 2$.

Theorem 3.1. *For any constants k and t such that $k \geq 2$, and any $\varepsilon > 0$, any non-adaptive (k,t)-committee selection algorithm that makes fewer than $\Omega(m^2)$ queries satisfies JR with probability at most ε.*

This result provides a separation between the non-adaptive and the adaptive settings: In Section 3.2, we discuss an adaptive (k,t)-committee selection algorithm guaranteeing JR with only $O(m)$ queries for any k and t such that $k < t$.

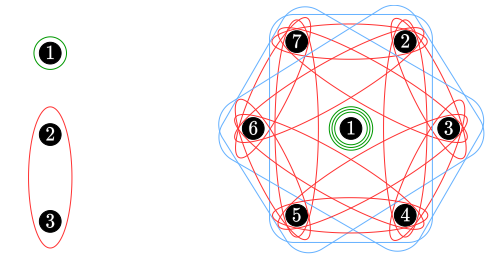
Theorem 3.1 follows from a more general result that we present formally in Appendix B. Here, we illustrate the argument by considering the special case where $t = k = 2$ and $\varepsilon = \frac{5}{6}$: Consider an adversary that picks a random set of 3 candidates, call them 1, 2, and 3, and answers queries according to the approval matrix visualized in Figure 1(a): half of the voters approve only candidate 1, and the other half of the voters approve only candidates 2 and 3. To satisfy JR, the algorithm needs to include candidate 1 in the committee. However, if the algorithm never queries $\{1,2\}$, $\{1,3\}$, or $\{2,3\}$, it receives no information that can distinguish candidates 1, 2, and 3 from each other, so it can do no better than selecting a random pair from these three candidates, which will succeed with probability $\frac{2}{3}$. This problematic case will occur frequently if the number of queries is not very large, say $\frac{1}{18} \cdot \binom{m}{2}$: Since there are $\binom{m}{2}$ pairs of candidates, the probability that the algorithm queries any randomly selected pair of candidates is at most $\frac{1}{18}$. By the union bound, the probability that the algorithm queries any of $\{1,2\}$, $\{1,3\}$, or $\{2,3\}$ is at most $3 \cdot \frac{1}{18} = \frac{1}{6}$. To summarize, for the algorithm to succeed, it either needs to get lucky during the querying phase, which happens with probability at most $\frac{1}{6}$, or during the selection phase, which happens with probability at most $\frac{2}{3}$. By the union bound, the algorithm succeeds with probability at most $\frac{1}{6} + \frac{2}{3} = \frac{5}{6}$.

A natural follow-up question is whether the $O(m^k)$ upper bound is tight for larger k. Interestingly, this is *not* the case for $k \geq 3$, as we prove in Appendix A:

Theorem 3.2. *For any $t \geq \frac{2}{3}k$, there exists a (k,t)-committee selection algorithm guaranteeing JR with $O(m^t)$ exact queries.*

However, the exponent does have a dependence on k. In particular, we find that guaranteeing JR requires $\Omega(m^3)$ queries starting at $k = 6$. The adversary employs an analogous strategy, now picking 7 random candidates and imposing the approval matrix depicted in Figure 1(b). Satisfying JR requires that the algorithm include candidate 1, which is indistinguishable from the other six candidates unless the algorithm makes $\Omega(m^3)$ queries, since every candidate is approved by $\frac{6}{18}$ of the voters and every pair of candidates is approved by $\frac{2}{18}$ of the voters.

In Appendix B, we describe a computational search we conducted to find similar instances for larger values of k. The best lower bound obtained is as follows.



(a) $k = 2$ (b) $k = 6$

Figure 1: Adversarial approval matrices. Each region represents a disjoint, equally-sized set of voters who approve only the candidates within the region. In (a), queries of size $t \geq 2$ are needed to distinguish the candidates; in (b), we need $t \geq 3$.

Theorem 3.3. *For any $\varepsilon > 0$, there exists a target committee size k with $k = \Theta(\log 1/\varepsilon)$ such that for all t, any non-adaptive (k,t)-committee selection algorithm with exact queries that makes fewer than $\Omega(m^{11})$ queries satisfies JR with probability at most ε.*

This theorem closes the book on the (im)practicality of non-adaptive committee selection algorithms. We therefore turn our attention to adaptive algorithms.

3.2 An Efficient Adaptive Algorithm

In this section, we propose an adaptive algorithm based on LS-PAV (Aziz et al. 2018), and we show that it achieves EJR and OAS with a practically-feasible number of queries.

For convenience, we introduce the following notation: For a committee W and candidates $c \in W$ and $c' \notin W$, let

$$\Delta(W, c', c) := \text{PAV-SC}(W \cup \{c'\} \setminus \{c\}) - \text{PAV-SC}(W)$$

denote the difference in PAV score obtained by replacing c with c' in W. Additionally, let

$$\Delta(W, c) := \text{PAV-SC}(W \cup \{c\}) - \text{PAV-SC}(W)$$

denote the marginal gain in PAV score by adding c to W.

LS-PAV starts with an arbitrary committee W and repeatedly replaces a committee member $c \in W$ with a candidate $c' \notin W$, provided the improvement to the PAV score satisfies $\Delta(W, c', c) \geq \frac{1}{k^2}$. Aziz et al. (2018) show that after at most $\mathcal{O}(k^2 \log k)$ swaps, no such swap pairs c, c' remain, at which point W satisfies OAS and EJR.

We first observe that LS-PAV can be implemented using exact queries: For any set of candidates S, PAV-SC(S) can be computed using any query $Q \supseteq S$, as it is sufficient to know the proportion of voters that approve each subset of S. Hence, for any W, $c \in W$, and $c' \notin W$, $\Delta(W, c', c)$ can be computed using a query Q that contains both W and c'. Using $\left\lceil \frac{m-k}{t-k} \right\rceil$ queries of size t, we can cover all $m - k$ candidates that are not in W, which leads to an overall query complexity of $\mathcal{O}\left(mk^2 \log k\right)$.

We next present a version of LS-PAV, which we call α-PAV (Algorithm 1), that has the same query complexity as LS-PAV for finding a committee that satisfies EJR and

Algorithm 1: (k,t)-α-PAV

Choose $W \in \binom{C}{k}$, $c \in W$, and $c' \notin W$ arbitrarily
$\gamma \leftarrow \infty$
while $\gamma \geq \frac{1}{\alpha k}$ **do**
  $W \leftarrow W \cup \{c'\} \setminus \{c\}$
  Choose $\mathcal{Q} = \{Q_i\}_i$, with $|Q_i| = t$, s.t. $W \subseteq \bigcap \mathcal{Q}$ and $C \subseteq \bigcup \mathcal{Q}$
  $c' \leftarrow \arg\max_{x \notin W} \Delta(W, x)$ ▷ (using $\mathcal{Q}$)
  $c \leftarrow \arg\max_{x \in W} \Delta(W, c', x)$ ▷ (using $\mathcal{Q}$)
  $\gamma \leftarrow \Delta(W, c')$
return W

OAS, but lower query complexity for approximate ($\alpha < 1$) α-EJR and α-OAS. Besides introducing the approximation parameter α, we make two other modifications to LS-PAV: First, Algorithm 1 terminates when there is no candidate c' such that $\Delta(W, c') \geq \frac{1}{k}$ (for $\alpha = 1$), while LS-PAV terminates when there is no pair c, c' such that $\Delta(W, c', c) \geq \frac{1}{k^2}$. As we shall see in Lemma 3.6, the termination condition of Algorithm 1 is weaker than that of LS-PAV, implying that it may terminate earlier. Second, instead of considering all possible swaps c, c', we only consider adding the candidate c' with the largest $\Delta(W, c')$. This modification makes the algorithm slightly simpler and more computationally efficient (by a factor of k).

Theorem 3.4. *For any $m \geq t > k$, Algorithm 1 yields a committee satisfying α-OAS and α-EJR while making at most*

$$\left\lceil \frac{m-k}{t-k} \right\rceil \frac{\alpha k^2}{(1-\alpha)k + 1} H_k$$

queries, where H_k is the k^{th} harmonic number. For $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^2 \log k\right)$ while for any fixed $\alpha < 1$, this leads to a query complexity of $\mathcal{O}\left(mk \log k\right)$.

The proof of Theorem 3.4 essentially follows from the following two lemmas, the first of which uses the notation

$$\Delta^*(W) := \max_{c \in C} \Delta(W, c).$$

Lemma 3.5. *If a committee W satisfies $\Delta^*(W) < \frac{1}{\alpha k}$, then W satisfies α-EJR and α-OAS.*

Lemma 3.6. *For any committee W and $c \notin W$, we have that $\max_{x \in W} \Delta(W, c, x) \geq \frac{(k+1)\Delta(W,c)-1}{k}$. In particular, if $\Delta(W, c) \geq \frac{1}{\alpha k}$, then $\max_{x \in W} \Delta(W, c, x) \geq \frac{(1-\alpha)k+1}{\alpha k^2}$.*

Lemma 3.5 guarantees that when Algorithm 1 terminates the desired fairness properties are satisfied. Lemma 3.6 establishes that the PAV score increases over the algorithm's run. This bounds the number of swaps it performs since PAV-SC(W) is at most H_k.

Lemma 3.5 is a generalization of the lower bound from Lemma 1 of Skowron (2021). This generalization is useful because it states that to establish EJR and OAS of any given committee W (no matter how it is derived), it is sufficient to prove that $\Delta^*(W)$ is small; hence it can be used as

a certificate of satisfaction. In Appendix E, we show that standard PAV and LS-PAV satisfy $\Delta^*(W) < \frac{n}{k}$, which is noteworthy in that it provides a simple proof of the known result that they satisfy EJR and OAS.

We observe that, for exact queries, an α-approximation with $\alpha < 1$ improves the query complexity by a factor of k. In the next section, we will see that such an approximation yields an even larger improvement in query complexity for noisy queries, as it also reduces the accuracy with which we need to estimate $\Delta(W, c', c)$.

4 Noisy Queries

We now turn to a query model that includes the noise we abstracted away in Section 3. In order to represent voters arriving to the platform one-by-one, we assume that each time the algorithm performs a query $Q \subseteq C$ a voter $i \in N$ is selected independently and uniformly at random[6]and then the algorithm observes their votes on the queried candidates $Q \cap A_i$. We refer to an algorithm that performs queries of size t, receives as a response the votes of a single voter, and outputs a committee of size k as a (k, t)-*committee selection algorithm with noisy queries*.

To see the connection between this query model and the previous one, note that an algorithm with noisy queries can approximate an exact query Q by estimating the values of p_S by taking the empirical proportion of repeated samples. By standard sample complexity bounds, using $\Theta\left(\log(2^t/\delta)/\varepsilon^2\right)$ queries, a noisy-query algorithm could guarantee $\pm\varepsilon$ estimates of p_S for all $S \subseteq Q$ with probability $1 - \delta$. Hence if an exact-query algorithm requires no more than $\text{poly}(m)$ queries with additive ε error, then it can be implemented using a factor of $\Theta(\log m)$ more noisy queries and yield a correct result with probability $1-\delta$. What's more, this $\log$ factor is in some cases necessary when moving from the exact-query to the noisy-query setting. In Appendix C, we demonstrate instances for which a non-adaptive exact-query algorithm needs only $\Theta(m)$ queries, while in order to be correct with any fixed probability δ, a non-adaptive noisy-query algorithm requires $\Omega(m \log m)$ queries.

Conversely, notice that one can use exact queries to simulate noisy queries. Indeed, p_S is exactly the probability that an incoming voter will vote yes on candidates S and no on $Q \setminus S$ in response to a query Q. An algorithm with access to exact query values can simply sample a voter response and feed it to a noisy-query algorithm. Therefore, the lower bounds on the query complexity of exact-query, non-adaptive algorithms, in particular Theorem 3.3, apply to noisy-query, non-adaptive algorithms as well. As the number of candidates becomes large, adaptivity is therefore necessary to attain theoretical guarantees — mirroring the approach of online platforms in practice.

[6]Note that a voter-profile A_i may be queried more than once during the run of the algorithm because we sample *with replacement*. This model simplifies the statistical analysis and has a natural interpretation: Rather than thinking of a finite population of voters, we draw samples from an underlying population distribution where each profile $A_1, .., A_n$ has the same frequency (probability). Furthermore, our model approaches sampling without replacement if the size of the underlying population n is large compared to the number of queried voters, hence both models are qualitatively interchangeable.

Algorithm 2: (k, t)-noisy-α-PAV

$\ell \leftarrow \left\lceil 288 \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)^2 \log\left(\frac{8mk^4}{\delta}\right)\right\rceil$
Choose $W \in \binom{C}{k}$, $c \in W$, and $c' \notin W$ arbitrarily
$\gamma \leftarrow \infty$
while $\gamma \geq 1/(\alpha k) - ((1-\alpha)k+1)/(12\alpha k^2)$ **do**
  $W \leftarrow W \cup \{c'\} \setminus \{c\}$
  Choose $\mathcal{Q} = \{Q_i\}_i$, with $|Q_i| = t$, such that $W \subseteq \bigcap \mathcal{Q}$ and $C \subseteq \bigcup \mathcal{Q}$
  Ask each query $Q \in \mathcal{Q}$ to ℓ new voters
  $\hat{\Delta}(W, x) \leftarrow$ estimate of $\Delta(W, x)$ using ℓ voters from query Q containing $W \cup \{x\}$ ▷ $\forall x \notin W$
  $\hat{\Delta}(W, x, y) \leftarrow$ estimate of $\Delta(W, x, y)$ using ℓ voters from Q containing $W \cup \{x\}$ ▷ $\forall x \notin W, \forall y \in W$
  $c' \leftarrow \arg\max_{x \notin W} \hat{\Delta}(W, x)$
  $c \leftarrow \arg\max_{x \in W} \hat{\Delta}(W, c', x)$
  $\gamma \leftarrow \hat{\Delta}(W, c')$
return W

A natural starting point is the exact-query adaptive algorithm, namely Algorithm 1. Indeed, it can be adapted to the noisy setting by replacing exact queries with a sufficient number of noisy queries, ℓ, to obtain high-probability bounds on Δ, yielding Algorithm 2.

The key is to choose ℓ large enough that if the termination condition is not met, i.e., we have $\hat{\Delta}(W, c') < \frac{1}{\alpha k} - \frac{(1-\alpha)k+1}{12\alpha k^2}$, the resulting swap is guaranteed to yield a positive improvement in the PAV-score, such that the number of steps of the algorithm is bounded. With the choice of ℓ in Algorithm 2, we obtain the following theorem, whose proof can be found in Appendix F.

Theorem 4.1. *For any $m \geq t > k$, with probability at least $1 - \delta$, Algorithm 2 returns a committee that satisfies α-EJR and α-OAS after querying no more than*

$$578 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)^3 \log\left(\frac{4mk^4}{\delta}\right)$$

voters. For any fixed $\delta > 0$, if $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^6 \log k \log m\right)$, and if $\alpha < 1$, this leads to a query complextiy of $\mathcal{O}\left(mk^3 \log k \log m\right)$.

While Algorithm 2 achieves good worst-case query complexity, it may be suboptimal on certain instances because of two reasons: (i) after each swap, Algorithm 2 discards all previous information so each candidate is reassessed from scratch, and (ii) it presents each candidate $c \notin W$ to the same number of voters, even though it may quickly become apparent that some candidates are more promising than others.

To address issue (i), we can use all past votes to compute bounds on Δ. A difficulty with this approach is that past voters may not have voted on all candidates in W (which is necessary to directly estimate $\Delta(W, c)$), since they may

have been queried on a different committee W'. But we can nonetheless use these past votes to obtain upper and lower bounds on estimated values. To address issue (ii), we can present promising candidates to voters more often. Further, it is possible to perform swaps as soon as we are confident they yield an increase of the PAV-score of at least some value ε, rather than first querying a predetermined number of voters as in Algorithm 2.

These ideas are incorporated into Algorithm 4, called ucb-α-PAV; see Appendix G for a formal description of the algorithm and an analysis of its query complexity.

5 Experiments

Since the analysis in the theoretical sections considers worst-case approval profiles, it is possible that, in practice, we may be able to find good committees with fewer queries than required by Theorem 4.1. We investigate this question empirically on real data from online discussions with only a few hundred voters, each voting on only a fraction of all comments.

Datasets. Polis provides open-use data from real deliberations hosted on their platform.[7] These include, for instance, a discussion organized by the government of Taiwan, which led to the successful regulation of Uber. Since participants typically only vote on a fraction of comments, most votes are missing. To be able to simulate the proposed adaptive algorithms, we first infer these missing votes using a matrix factorization library, LensKit.[8] Importantly, we infer votes only for the purpose of the experiments; if our algorithms were executed during the discussion, they would adaptively query users about the relevant comments without relying on any inference method.

In most datasets, we observe several comments that are nearly universally approved. Since these comments make achieving EJR and OAS trivial, we remove comments approved by more than 60% of participants. This step may also be appropriate in practice to gain insights into participants' opinions beyond uncontroversial issues.

The number of queried voters L ranges from 87 to 1000 across the 13 datasets (see Appendix H for details). For all datasets, we assume that each voter votes on $t = 20$ comments. Since the total number of comments m ranges from 31 to 1719 across datasets, the percentage of comments each voter votes on, t/m, ranges from 1% to 65%. For each dataset, we run the algorithms with target committee sizes $k = 5, 7, 10$. Hence, there are a total of $13 \cdot 3 = 39$ experiments (times 10 random seeds).

The second dataset we consider consists of Reddit discussions.[9] To obtain an interesting dataset, we combined voting data from two subreddits, r/politics and r/Conservative, which are arguably situated at opposite ends of the American political spectrum. More details about this dataset can also be found in Appendix H.

[7] https://github.com/compdemocracy/openData

[8] https://lenskit.org

[9] https://www.kaggle.com/datasets/josephleake/huge-collection-of-reddit-votes

Algorithms. We evaluate noisy-α-PAV (Algorithm 2) and ucb-α-PAV (Algorithm 4). Both query L voters in random order, each of whom votes on $t = 20$ comments. To enable these algorithms to swap candidates after querying only a small number of voters, we make the following modifications: For both Algorithm 2 and Algorithm 4 we treat ℓ, the number of times we ask voters about each candidate, as a parameter. In addition, for Algorithm 4, we replace the numerator in the confidence intervals err_s with a parameter θ. Both ℓ and θ were chosen based on validation on a separate dataset, see Appendix H for details. We run both algorithms on all the L voters, rather than terminating as soon as we can guarantee $\Delta^*(W) < \frac{1}{\alpha k}$ (and hence EJR and OAS).

To obtain an upper bound on the attainable performance, we execute α-PAV (Algorithm 1) with access to exact queries. To obtain the best possible α, we let Algorithm 1 run as long as the swap increases the PAV score, i.e., $\Delta(W, c', c) > 0$, instead of terminating as soon as $\Delta(W, c') < 1/k$ (which would be sufficient to guarantee EJR and OAS).

To verify that the proposed algorithms do indeed take the complementarity of different candidates into account, we also compare against standard approval voting (AV) with access to all votes, which simply selects the k candidates with the most approval votes.

Performance Metric. As a performance metric, we use $\hat{\alpha} := \frac{1}{k\Delta^*(W)}$, where W is the committee selected by the respective algorithm. According to Lemma 3.5, $\alpha > \hat{\alpha}$, so this implies $\hat{\alpha}$-EJR and $\hat{\alpha}$-OAS. As discussed in Section 2, $\alpha = 1$ is the best that can be guaranteed across all possible approval profiles. Note that α may be larger than $\hat{\alpha}$, hence obtaining $\hat{\alpha} = 1$ is a sufficient, but not a necessary condition for OAS and EJR. Nevertheless, we will use $\hat{\alpha}$ as a metric for two reasons: first, verifying whether $\alpha \geq 1$ (i.e., whether a committee satisfies EJR and OAS) is computationally hard (Aziz et al. 2017), which makes it impractical for evaluation; and the stronger condition $\hat{\alpha} \geq 1$ provides the additional benefit that EJR and OAS can easily be verified through Lemma 3.5. Second, one could argue that $\hat{\alpha}$ is a meaningful quantity in its own right since it (or rather its inverse $1/\hat{\alpha}$) measures how much voter satisfaction could be improved by adding another candidate (giving lower weight to voters who already have many approved candidates).

Polis Results. In Figure 2, we show the $\hat{\alpha}$ achieved on all the Polis datasets for each of the four algorithms. Recall that higher $\hat{\alpha}$ is better and that $\hat{\alpha} \geq 1$ implies OAS and EJR. As expected, α-PAV performs best since it has access to exact queries. Note that it often achieves an α substantially larger than 1, which means that the corresponding instance allows for better representation than can be guaranteed in the worst case. AV performs surprisingly well in most experiments, but in 38% of the cases, it yields $\hat{\alpha}$ smaller than 1 (and sometimes much smaller). We conclude that for some datasets, it is important to take the complementarity of candidates into account rather than selecting them individually. The challenge for the proposed algorithms is to do so while being sample-efficient. We see that noisy-α-PAV

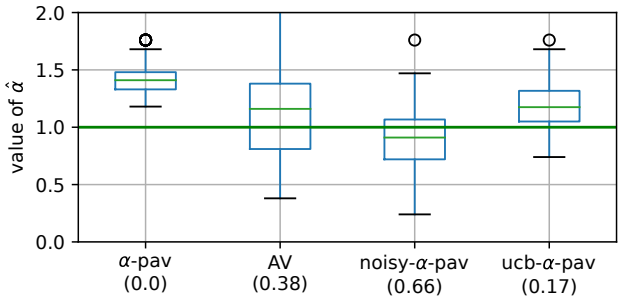


Figure 2: Boxplots where datapoints correspond to the 39 Polis problems ($\times 10$ random seeds). The top / bottom whiskers indicate the maximal / minimal points (except outliers, which are marked by circles), the line in the middle is the median, and the bottom and top of the boxes are the 1st and 3rd quartiles, respectively. The numbers in parenthesis are the fractions of problems where the respective algorithm yields a $\hat{\alpha} \leq 1$.

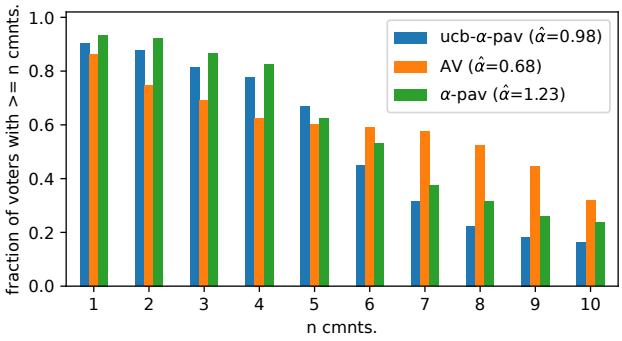


Figure 3: Results on Reddit dataset (with $L = 608, m = 2135, k = 10$): the fraction of voters (y-axis) that approve of at least $1, 2, .., 10$ candidates (x-axis) among the selected committee of size $k = 10$.

often fails to achieve an $\hat{\alpha} \geq 1$. We know from Theorem 4.1 that given enough queries, noisy-α-PAV achieves $\hat{\alpha} \geq 1$, so this failure is due to the low number of queried voters. By contrast, ucb-α-PAV yields $\hat{\alpha} \geq 1$ in 83% of the cases, and $\hat{\alpha} \geq 0.75$ in all cases, which indicates that the proposed extensions (i.e., querying promising candidates more often, swapping as soon as possible, and reusing voters) indeed lead to more efficient use of data.

Reddit Results. To illustrate why approval voting can perform poorly despite having access to the full votes, we execute the algorithms on the Reddit dataset described above. In this experiment, AV achieves only $\hat{\alpha} = 0.68$. To understand why this happens, we show in Figure 3 the fraction of voters who have at least $1, \ldots, 10$ approved comments in the committee. We see that AV yields a committee where a high fraction of voters approve many candidates, e.g., about 60% of voters approve 7 or more candidates, whereas for α-PAV, this is the case for only about 40%. This comes at the cost of a high fraction of voters who are poorly represented by AV, e.g., about 25% of voters get at most one approved candidate, whereas for α-PAV, this percentage is less than 10%. This is to be expected as approval voting does not take the complementarity of candidates into account and can therefore lead to less equitable results. Finally, we observe that ucb-α-PAV achieves an $\hat{\alpha}$ close to 1, and its approval fractions look similar to α-PAV, i.e., more equitable than AV. It is interesting that ucb-α-PAV performs well on this example, since it only has access to $t = 20$ votes for each of the $L = 608$ queried candidates, while it has to select from a large number of comments, $m = 2135$.

6 Discussion

This work bridges the gap between online civic-participation systems, such as Polis, and committee-election methods by enabling them to handle incomplete votes. To deploy the proposed algorithms on such platforms, two practical issues must be considered.

First, our adaptive approach requires control over what the Polis creators call *comment routing* (Small et al. 2021): the algorithm that decides which comments are shown to which participants. If on a given platform a comment-routing algorithm is already in place, shared control is possible: each algorithm could determine part of the slate of comments shown to a participant, or the participants themselves can be divided between the algorithms.

Second, in our analysis, we assumed that all comments have been submitted — or all candidates are known — at the time we run our algorithms. Nevertheless, our algorithms can be extended straightforwardly to a growing set of comments, but we would inevitably lose the representation guarantees for comments that were submitted late if not enough participants could vote on them. In practice, this could be resolved by setting a comment submission deadline, which has been done previously by Polis.

An alternative to our approach would be to complete partial approval votes using collaborative filtering (Resnick and Varian 1997). The completed approval votes can then be aggregated through any approval-based committee election rule, such as PAV. The disadvantage of this approach is that it is unlikely to lead to worst-case guarantees of the type we establish in this paper.

Finally, we emphasize that our approach may be applicable to social media more generally. For instance, as mentioned in Section 5, Reddit users also approve or disapprove comments through upvotes and downvotes. However, Reddit uses these inputs to produce a *ranking* of the comments, in contrast to our goal of selecting a subset. There is work on obtaining justified-representation-type guarantees for rankings (Skowron et al. 2017), which could possibly be extended to the setting of incomplete votes using the techniques developed in this paper. More broadly, this article provides insights into how to fairly represent opinions of groups given incomplete information, which may be relevant for the design of more constructive online ecosystems.

References

Aragón, P.; Kaltenbrunner, A.; Calleja-López, A.; Pereira, A.; Monterde, A.; Barandiaran, X. E.; and Gómez, V. 2017. Deliberative platform design: The case study of the online discussions in decidim Barcelona. In *Proceedings of the 9th International Conference on Social Informatics (SocInf)*, 277–287.

Ash, R. B. 1990. *Information Theory*. Dover Publications.

Aziz, H.; Brill, M.; Elkind, E.; Freeman, R.; and Walsh, T. 2017. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare*, 42(2): 461–485.

Aziz, H.; Elkind, E.; Huang, S.; Lackner, M.; Sánchez-Fernández, L.; and Skowron, P. 2018. On the Complexity of Extended and Proportional Justified Representation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 902–909.

Aziz, H.; Gaspers, S.; Gudmundsson, J.; Mackenzie, S.; Mattei, N.; and Walsh, T. 2015. Computational Aspects of Multi-Winner Approval Voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 107–115.

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

Fernández, L. S.; Elkind, E.; Lackner, M.; García, N. F.; Arias-Fisteus, J.; Basanta-Val, P.; and Skowron, P. 2017. Proportional Justified Representation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 670–676.

Filmus, Y.; and Oren, J. 2014. Efficient Voting via the Top-k Elicitation Scheme: A Probabilistic Approach. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*, 295–312.

Fishkin, J.; Garg, N.; Gelauff, L.; Goel, A.; Munagala, K.; Sakshuwong, S.; Siu, A.; and Yandamuri, S. 2019. Deliberative democracy with the Online Deliberation platform. In *Proceedings of the 7th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 1–2.

Iandoli, L.; Klein, M.; and Zollo, G. 2009. Enabling On-Line Deliberation and Collective Decision-Making through Large-Scale Argumentation: A New Approach to the Design of an Internet-Based Mass Collaboration Platform. *International Journal of Decision Support System Technology (IJDSST)*, 1(1): 69–92.

Imber, A.; Israel, J.; Brill, M.; and Kimelfeld, B. 2022. Approval-Based Committee Voting under Incomplete Information. *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 36(5): 5076–5083.

Ito, T.; Suzuki, S.; Yamaguchi, N.; Nishida, T.; Hiraishi, K.; and Yoshino, K. 2020. D-Agree: Crowd Discussion Support System Based on Automated Facilitation Agent. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 13614–13615.

Resnick, P.; and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM*, 40(3): 56–58.

Salganik, M. J.; and Levy, K. E. C. 2015. Wiki surveys: open and quantifiable social data collection. *PloS one*, 10(5): e0123483.

Shibata, D.; Moustafa, A.; Ito, T.; and Suzuki, S. 2019. On Facilitating Large-Scale Online Discussions. In *PRICAI 2019: Trends in Artificial Intelligence*, 608–620. Springer International Publishing.

Skowron, P. 2021. Proportionality Degree of Multiwinner Rules. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, 820–840.

Skowron, P.; Lackner, M.; Brill, M.; Peters, D.; and Elkind, E. 2017. Proportional Rankings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 409–415.

Small, C.; Bjorkegren, M.; Erkkilä, T.; Shaw, L.; and Megill, C. 2021. Polis: Scaling Deliberation by Mapping High Dimensional Opinion Spaces. *Revista De Pensament I Anàlisi*, 26(2).

Terzopoulou, Z.; Karpov, A.; and Obraztsova, S. 2021. Restricted Domains of Dichotomous Preferences with Possibly Incomplete Information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6): 5726–5733.

Xia, L.; and Conitzer, V. 2011. Determining Possible and Necessary Winners Given Partial Orders. *Journal of Artificial Intelligence Research*, 41: 25–67.

Zhou, A.; Yang, Y.; and Guo, J. 2019. Parameterized Complexity of Committee Elections with Dichotomous and Trichotomous Votes. In *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, AAMAS '19, 503–510. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.