Crowdsourcing for Participatory Democracies: Efficient Elicitation of Social Choice Functions

David T. Lee
Stanford University
davidtlee@stanford.edu

Ashish Goel
Stanford University
ashishg@stanford.edu

Tanja Aitamurto
Stanford University
tanjaa@stanford.edu

Hélène Landemore
Yale University
helene.landemore@yale.edu

Abstract

We present theoretical and empirical results demonstrating the usefulness of social choice functions in crowdsourcing for participatory democracies. First, we demonstrate the scalability of social choice functions by defining a natural notion of ϵ-approximation, and giving algorithms which efficiently elicit such approximations for two prominent social choice functions: the Borda rule and the Condorcet winner. This result circumvents previous prohibitive lower bounds and is surprisingly strong: even if the number of ideas is as large as the number of participants, each participant will only have to make a logarithmic number of comparisons, an exponential improvement over the linear number of comparisons previously needed. Second, we apply these ideas to Finland's recent off-road traffic law reform, an experiment on participatory democracy in real life. This allows us to verify the scaling predicted in our theory and show that the constant involved is also not large. In addition, by collecting data on the time that users take to complete rankings of varying sizes, we observe that eliciting partial rankings can further decrease elicitation time as compared to the common method of eliciting pairwise comparisons. Finally, we give a few variations of our initial algorithms that improve practical use: we improve elicitation time by taking advantage of the specific preference distribution and we show how one can handle streams of ideas arriving over time in a way which does not significantly increase the total comparisons elicited.

Introduction

Recent years have seen an increase in democratic innovations (Smith 2009) aimed at increasing the participation of the public in policy-making. This observation, coupled with the increasing prevalence of internet-based communication, points to a very real possibility of implementing participatory democracies on a mass-scale in which every individual is invited to contribute their ideas and opinions (Salganik and Levy 2012).

One important question in implementing crowdsourcing experiments of this type is the aggregation problem: given a large number of ideas, how can one identify the top ideas without requiring any individual, whether an appointed government expert or a participant, to spend too much time or effort in the evaluation process? A natural approach that one might want to use in the democratic setting is to use voting rules, also known as social choice functions (Brams and Fishburn 2002), to find the top ideas or overall ranking. Unfortunately, in the standard setting of rank aggregation, each participant is required to submit a full ranking of all the ideas, a task which is both cognitively burdensome and time-consuming.

In this paper, we present theoretical and empirical results indicating the usefulness of social choice functions for participatory democracies. Our main contributions are two-fold. First, we demonstrate the scalability of social choice functions by defining a natural notion of an approximate winner or ranking. This allows us to design algorithms which are able to approximate two prominent social choice functions, the Borda rule and the Condorcet winner, while only using a small number of pairwise comparisons. The algorithms are extremely simple, but are able to circumvent previous prohibitive lower bounds (Conitzer and Sandholm 2005; Service and Adams 2012) and are surprisingly strong in two ways. First, the total number of pairwise comparisons is independent of the number of participants. Second, even if the number of ideas is as large as the number of participants, each participant will only have to make a logarithmic number of comparisons, an exponential improvement over the linear number of comparisons previously needed.

	Borda	Condorcet
CS '05	$\Omega(nm \log m)$	$\Omega(nm)$
SA '12	$\Omega(nm \log \frac{1}{\epsilon})$	N/A
Our Results	$O(\frac{m}{\epsilon^2} \log \frac{m}{\delta})$	$O(\frac{m}{\epsilon^4} \log^2 \frac{m}{\delta\epsilon^2})$

Figure 1: Comparison of our results with prior lower bounds. We circumvent these lower bounds by finding Monte Carlo randomized algorithms which achieve an ϵ-approximate result with high probability (at least $1 - \delta$). The lower bounds of CS '05 and SA '12 only hold for Los Vegas randomized algorithms in which one can distinguish between successful and unsuccessful instances.

Essentially, we show that these social choice functions, which scale inefficiently in general, can be easy to implement when the winner wins by a margin or an approximation suffices.

Second, we demonstrate the practicality of these ideas by applying them to Finland's recent off-road traffic law reform, an experiment on participatory democracy in real life (Aitamurto and Landemore 2013). The Finnish experiment, as we will refer to it from this point on, engaged the Finnish people in 1) identifying problems, 2) proposing solutions, and 3) evaluating ideas for a revision of their off-road traffic law. In the evaluation stage, 308 participants took part in ranking and rating ideas in 41 different topics, each of which had a number of ideas ranging from 2 to 15. For an approximation of $\epsilon = 0.05$ and 0.1, we are able to observe scaling which matches our theory and show that the constant involved is also not large. As an example, when the derived trend is extrapolated to the case of aggregating 100 ideas with 1000 participants, an error of 0.05 only requires each participant to make 19 comparisons. If we only need an error of 0.1, then 8 comparisons suffice.

In addition, by collecting data on the time that users take to complete rankings of varying sizes, we are able to show that eliciting partial rankings can further decrease elicitation time as compared to the common method of eliciting pairwise comparisons. Though users naturally take longer to rank a larger number of proposals, the data demonstrates that the time per bit elicited decreases by 33% when ranking 3 proposals, 48% when ranking 4 proposals, and 63% when ranking 6 proposals. Finally, we give further improvements that can be used in practical situations. We first show that one can further decrease elicitation time by taking advantage of the preference distribution. Then we show that one can also handle the case when ideas are streaming while still keeping elicitation time small.

Outline

The remainder of the paper will be structured as follows. We first describe related work and briefly highlight our contributions. This is followed by a section on our theoretical work on the efficient elicitation of social choice functions, in which we state and motivate our definitions for ϵ-approximate rankings and winners, detail our elicitation algorithms, and prove that these algorithms are able to efficiently find ϵ-approximations. We then describe the Finnish experiment and use it to show the practicality of the previously stated algorithms. This section concludes with data showing that one can further decrease elicitation time by eliciting partial rankings. Finally, we give two further algorithmic improvements that further improve the practical applicability of our results by further reducing elicitation time and by showing that one can handle ideas arriving in a streaming fashion.

Background and Related Work

Participatory democracies and crowdsourcing

The notion of participatory democracy emphasizes the role of citizen involvement in political decision-making as a supplement to, and even a substitute for, representative democratic institutions. Participatory democracy dates back at least to the 1960s and 1970s. Its core argument states that democracy is made more meaningful when citizens are directly involved in the direction of political affairs (Pateman 1979; Macpherson 1977; Fung and Wright 2003). Many participatory democrats also emphasize the necessity of empowering citizens in other areas of their lives as well, such as the workplace and the family.

Participatory democracy has been applied over the decades in various settings, for instance in participatory budgeting projects, which started in Brazil in the 1960s and have since spread globally (Cabannes 2004). In recent years, as a parallel development to improved digital technologies, there has been a rise of online applications that support civic engagement in participatory democracy. One of these applications is crowdsourcing, which can be used as knowledge search mechanism in policy-making. In crowdsourced policy-making, citizens are asked to contribute their knowledge and ideas online (Aitamurto 2012; Brahbam 2013). Crowdsourcing has even been recently applied in the context of constitutional reform in Iceland (Landemore 2014). One important challenge in large-scale ideation and deliberation systems is the assessment of the crowd-generated ideas, particularly in the instances where the volume of participation is high: How can the ideas be assessed by the crowd in an effective and reliable way?

Aggregation and preference elicitation

One method of aggregation which has been long associated to the democratic setting is rank aggregation, originating from the theory of social choice (Brams and Fishburn 2002). In rank aggregation, n voters have preferences over m candidates, and one would like to design a social choice function which takes this set of rankings as an input, and outputs an aggregate ranking or winner. Two of the early and prominent social choice functions proposed were the Borda rule and Condorcet criterion[1]. In the Borda rule, Borda proposed that scores should be assigned to each candidate based on their position in the set of rankings. A rank of one gives a candidate $m - 1$ points, a second-place gives a candidate $m - 2$ points, all the way down to the lowest rank, which gives a candidate 0 points. Condorcet took an axiomatic approach, stating that if a candidate beats all other candidates in a pairwise election, then this should naturally be the winner.

One difference in applying social choice to crowdsourcing, as compared to the standard setting, is that there may be a large number of candidates, making it impractical for a voter to submit a complete ranking. In such a setting, one wants to understand whether it is possible to use a small number of simple queries (such as pairwise comparisons) to find the desired output.

(Conitzer and Sandholm 2002) showed that finding out how to elicit optimally is NP-complete and (Conitzer and Sandholm 2005) showed that, for many common voting

[1] Young has a fascinating historical discussion on these two rules (Young 1988).

rules, the number of bits necessary to elicit the output ranking or winner cannot be reduced by more than a constant factor. This was extended by (Service and Adams 2012), who showed that many of these lower bounds also held for finding approximate winners. Attempts to circumvent these lower bounds typically either restrict the set of rankings allowed (the preference domain), assume distributions on voter preferences, or focus on experimental analysis of elicitation.

For instance, (Conitzer 2009) considers how to elicit single-peaked preferences using pairwise comparisons and (Goel and Lee 2012) considers the same setting but assume that candidates and participants coincide, and that the single-peaked axis is unknown.

Several papers have discussed top-k elicitation in which one asks participants to reveal their top ranked ideas. The lower bounds of (Conitzer and Sandholm 2005) and (Service and Adams 2012) still apply to these settings since they are lower bounds on the number of bits elicited. (Kalech et al. 2011) shows that an iterative algorithm does well experimentally on datasets and under certain distributional assumptions. (Oren, Filmus, and Boutilier 2013) derives lower bounds on k under distributional assumptions. They also show that if preferences are drawn from a Mallow's model, then there are conditions for which knowing the model parameters themselves are enough to determine the Borda rule with high probability.

(Lu and Boutilier 2011a) considers a notion of approximate winners based on maximum regret. For a scoring function associated to a given voting rule, the regret is the worst-case score of that candidate over all possible rankings consistent with the revealed preferences. They propose a greedy heuristic for vote elicitation in which they minimize the maximum regret at each step and analyze it experimentally on various datasets. For the Borda rule, their notion of maximum regret is similar to the notion in (Service and Adams 2012) after a suitable normalization. Since their algorithm is deterministic, the lower bounds of (Service and Adams 2012) apply. (Lu and Boutilier 2011b) also uses the maximum regret framework and considers how to find the number k such that eliciting the top-k candidates of all participants results in finding a winner with bounded maximum regret. They present an algorithm that finds an estimate for k given a distributional assumption on the preference profile. As above, for the Borda rule, the lower bounds of (Service and Adams 2012) still apply.

(Caragiannis, Procaccia, and Shah 2013) is slightly different in that it focuses on sampling participants, each of whom must give their full ranking. Assuming that all participants are drawn from a Mallows model, they then show that the number of sampled participants needed to find the correct ranking with high probability is logarithmic in the number of candidates for a large class of voting rules. (Chevaleyre et al. 2011) considers elicitation given dynamic sets of candidates. Since they are looking for an exact winner, the lower bounds of (Conitzer and Sandholm 2005) apply. (Procaccia 2008) discusses elicitation of the Condorcet winner and finds the exact complexity required, when queries are complete pairwise elections between pairs of candidates.

The application of social choice to crowdsourcing has also been considered in (Mao, Procaccia, and Chen 2013) through the maximum-likelihood perspective to social choice functions. The general problem of aggregation has also been studied outside of social choice. These include learning parameters of statistical models (Salganik and Levy 2012), or the use of ratings instead of rankings (Balinski and Laraki 2007).

Our Contributions

We define ϵ-approximations of social choice functions and show that, despite previous lower bounds of $\Omega(mn)$, we are able to elicit these approximations in time $O(m \log m)$ (and factors depending on the accuracy required), which makes elicitation schemes incredibly practical for crowdsourcing applications. Prior results have all required restrictions of the preferences or have been experimental.

Our definitions of approximation are novel in that they relate to the underlying preference profile as opposed to only the output produced: an ϵ-approximate ranking is a ranking that could have been the output ranking by changing at most ϵ fraction of the comparison values involving each candidate. Such a definition is intuitive for the crowdsourcing problem in that the notion of distance can be directly translated to the question of "how close was this ranking from being the winner"?

We note that the concept of distances dependent on the underlying preference profile is not new and is rooted in the literature on distance rationalizability of voting rules (Meskanen and Nurmi 2008; Elkind, Faliszewski, and Slinko 2010); however, the application of this towards approximation and efficient elicitation has been unexplored thus far.

We verify these results through a real life participatory democracy, in which citizens of Finland voted on proposals which they created for 41 separate topics. This allows us to experimentally analyze the scaling of the elicitation scheme as a function of the number of proposals as well as determine the constant involved in the algorithm for this instance. We are not aware of other experimental work which analyzes the scaling of algorithms. Importantly, our experiments also identify a behavioral phenomena which may be a further means to decrease elicitation time from a human interaction perspective. Specifically, we show that the time per bit of information elicited decreases when one uses partial rankings as opposed to the standard method of pairwise comparisons.

Finally, we show that one can take advantage of the distribution to further improve elicitation time. We also give a novel algorithm which handles preference elicitation when ideas arrive in a streaming fashion.

Efficient elicitation of social choice functions

In the following section, we will present our theoretical work on the efficient elicitation of social choice functions. We first define and motivate our notion of approximate winners and rankings. We then show that a simple sampling algorithm is able to approximate the Borda rule in only $O(\frac{m}{\epsilon^2} \log \frac{m}{\delta})$ comparisons and highlight an interesting insight about the role honesty plays in achieving such a result. Finally, we show a variant of this algorithm which is able to approximate the Condorcet winner.

Notation

Let C denote the set of ideas and V the participants. Let m and n denote the number of ideas and participants respectively. If participant i prefers idea x to y, we denote this by $x \succ_i y$.

The *Borda score* of an idea x is defined as $s(x) = \sum_{i \in V}(m - r_i(x))$, where $r_i(x)$ denotes the rank that participant i gives to idea x. That is, an idea receives $m-1$ points if it was ranked first in a ranking, $m-2$ if it was ranked second, and so on. Equivalently, the Borda score can also be defined as $s(x) = \sum_{i \in V} \sum_{y \in C \setminus \{x\}} \mathbb{1}_{\{x \succ_i y\}}$, the total number of comparisons in which x won. The *Borda winner* x^* is any idea with the highest score, i.e. $s(x^*) = \max_x s(x)$. The Condorcet winner is defined as an idea x which beats all other ideas in a pairwise election.

Approximate rankings and winners

Define the *normalized Borda score* of an idea x to be $n(x) = s(x)/\sum_{x' \in C} s(x')$, so that the sum of all the Borda scores is 1. Define an ϵ-Borda winner to be any idea x such that $s(x) \geq (1-\epsilon)s(x^*)$. Define an ϵ-Borda ranking to be any ranking resulting from a normalized score vector $\hat{n}$ such that for any idea x, $|\hat{n}(x) - n(x)| \leq 2\epsilon/m$.

Define an ϵ-Condorcet winner to be an idea x which receives at least $(1-\epsilon)\frac{n}{2}$ votes against at least $(1-\epsilon)(m-1)$ other candidates.

Example 1. *Suppose that* 100 *participants supply rankings of* 3 *candidates, and that these result in the following comparisons (the value in the table represents the number of participants preferring the left candidate to the top candidate):*

	A	B	C
A	-	*52*	*45*
B	*48*	-	*64*
C	*55*	*36*	-

The Borda scores for candidates A, B, and C are 97, 112, and 91 respectively, so that the Borda ranking is $B \succ A \succ C$ and the Borda winner is B. There is no Condorcet winner since A beats B, B beats C, and C beats A.

Simple calculations show us that A is a 0.14-Borda winner since $1 - 97/112 \approx 0.134$ and that C is a 0.19-Borda winner since $1 - 91/112 \approx 0.188$. We can also calculate the normalized Borda scores, which are approximately $0.323, 0.373, 0.303$ respectively. By noting that the ranking $B \succ C \succ A$ could have resulted from the normalized Borda score $0.313, 0.373, 0.313$, we get that $|\hat{n}(x) - n(x)| \leq 2 \cdot 0.015/3$ for all x, which means that this ranking is a 0.015-Borda ranking. We also see that B is a 0.04-Condorcet winner since $1 - 48/50 = 0.04$ and A is a 0.1-Condorcet winner since $1 - 45/50 = 0.1$.

We can see that the definitions capture the intuition that neither A or C are close to being Borda winners, but that the ranking $B \succ C \succ A$ is close to being a Borda ranking. We also see agreement in the intuition that both A and B are approximate Condorcet winners, B being closer than A. □

These two approximation definitions fall under the umbrella of broader definitions which have a practical intuition. Consider any social choice function which only depends on the number of comparisons won for every pairwise election (as in the table of Example 1). Then a broader definition of an ϵ approximate winner is a candidate that could be made the winner by changing at most ϵ fraction of the comparisons involving that candidate. Similarly, a broader definition of an ϵ approximate ranking is a ranking that could be made the output ranking by changing at most ϵ fraction of the comparisons values involving each candidate[2].

Theorem 1. *If x is an ϵ-Borda (or ϵ-Condorcet) winner, then it is possible to make it the Borda (or Condorcet) winner by changing at most ϵ fraction of the $(m-1)n$ comparisons involving x.*

Proof. Suppose x is an ϵ-Borda winner so that $s(x) \geq (1-\epsilon)s(x^*)$. Since the Borda score is just equal to the number of comparisons won by a given candidate, this means that x can be made the winner by changing $\epsilon \cdot s(x^*)$ of his losing comparisons into winning ones. Since $\epsilon \cdot s(x^*) \leq \epsilon(m-1)n$, we are done.

Suppose x is an ϵ-Condorcet winner so that x receives at least $(1-\epsilon)\frac{n}{2}$ votes against at least $(1-\epsilon)(m-1)$ other candidates. For every other candidate y, change enough comparisons between x and y so that x wins at least $\frac{n}{2}$ of these comparisons, making it the Condorcet winner. This number will be at most $(1-\epsilon)(m-1)\epsilon\frac{n}{2} + \epsilon(m-1)\frac{n}{2} \leq \epsilon(m-1)n$. □

Theorem 2. *If $\succ$ is an ϵ-Borda ranking, then it is possible to make it the Borda ranking by changing some number of comparisons such that for any candidate x, at most ϵ fraction of the $(m-1)n$ comparisons involving x were changed.*

Proof. Suppose that $\succ$ is an ϵ-Borda ranking. By definition, it must result from a normalized score vector $\hat{n}$ such that for any idea x, $|\hat{n}(x) - n(x)| \leq 2\epsilon/m$. Letting $\hat{s}$ denote the corresponding score vector, we have that $|\hat{s}(x) - s(x)| \leq \frac{2\epsilon}{m} \cdot \frac{m(m-1)n}{2} = \epsilon(m-1)n$. This means that we can reach $\hat{s}$ by changing some number of comparisons of which at most $\epsilon(m-1)n$ correspond to any given candidate. □

We emphasize the practical nature of this broader definition: if a candidate is an approximate winner then this literally means that it could have been a winner given a small perturbation in voter inputs. The smaller the approximation distance, the smaller the perturbation is needed to make it a winner. In many practical settings, the entire set of voters in the preference profile is only a subset of the "true" voting population, many of which did not submit a ranking at all. This means that the actual winner produced under the given preference profile may also be only an approximation of the "true winner" had everyone participated, implying that it is reasonable to allow slight perturbations in the input. The same holds for the broader definition of an approximate ranking.

[2]Note: we allow changing a fractional number of comparisons to prevent artificially large errors resulting from having an odd or even number of voters.

ALGORITHM 1: Approximating the Borda rule

Input: m ideas, n participants, a number of samples N
Output: An output ranking
count$[\cdot] = 0$;
for $i \leftarrow 1$ **to** N **do**
    Sample ideas c_1 and c_2 and a participant v uniformly at random;
    if $c_1 \succ_v c_2$ **then**
        count$[c_1]$ = count$[c_1] + 1$;
    else
        count$[c_2]$ = count$[c_2] + 1$;
return any ranking such that for any x ranked higher than y, count$[x] \geq$ count$[y]$;

Simple does it: eliciting the Borda rule with naive sampling

Consider the following sampling algorithm (Algorithm 1). At each step, sample a participant uniformly at random and ask him to compare two ideas sampled uniformly at random. Increment a counter for the idea chosen by the participant and repeat N times. Now form an output ranking by ordering the ideas from those with the highest to lowest counter values, with ties broken arbitrarily.

Theorem 3. *For any $\epsilon, \delta \in (0, 1)$, Algorithm 1 with $N = O(\frac{m}{\epsilon^2} \ln \frac{m}{\delta})$ returns an ϵ-Borda ranking with probability at least $1 - \delta$. Also, the top idea in the returned ranking is an ϵ-Borda winner with probability at least $1 - \delta$.*

Proof. To prove that the output ranking is an ϵ-Borda ranking, we just need to identify a normalized score vector $\hat{n}$ for the resulting output ranking such that $|\hat{n}(x) - n(x)| \leq 2\epsilon/m$. Let $\hat{n}(x) = \text{count}[x]/\sum_{x' \in C} \text{count}[x']$ and, for convenience, let $Z = \binom{m}{2} n$. Then,

$$\mathbb{E}[\hat{n}(x)] = \frac{1}{Z} \sum_{i \in V} \sum_{y \in C \setminus \{x\}} \mathbb{1}_{\{x \succ_i y\}}$$
$$= \frac{1}{Z} \sum_{i \in V} (m - r_i(x)) = \frac{1}{Z} s(x) = n(x)$$

We will now use the following special form of Chernoff bounds (see (Motwani and Raghavan 1995), Theorems 4.2, 4.3): *Let S be a sum of independent Bernoulli random variables, and μ be an arbitrary number such that $\mu \geq \mathbb{E}[S]$. Then for any $\epsilon \in [0, 1]$,*

$$\textbf{Pr}\left[|S - \mathbb{E}[S]| > \epsilon\mu\right] \leq 2e^{-\epsilon^2 \mu/4}.$$

We are going to apply Chernoff bounds to the random variable $S(x) = \text{count}[x]$. Observe that $S(x)$ is a sum of independent Bernoulli random variables since our comparisons are sampled with replacement (it is possible to sample the exact same comparison twice), that $S(x) = N \cdot \hat{n}(x)$, and $\mathbb{E}[S(x)] = N \cdot n(x)$. Further, let $\mu = 2N/m$ denote the expected number of comparisons in which x participates. Since x must participate in a trial to win, we must have $\mathbb{E}[S(x)] \leq \mu$ and hence, we can apply the Chernoff bound stated above, i.e., for any $\epsilon \in [0, 1]$, we have:

$$\textbf{Pr}\left[|S(x) - \mathbb{E}[S(x)]| > \epsilon\mu\right] \leq 2e^{-\epsilon^2 \mu/4}.$$

Equivalently, we have

$$\textbf{Pr}\left[|\hat{n}(x) - n(x)| > 2\epsilon/m\right] \leq 2e^{-\epsilon^2 N/(2m)}.$$

Applying union bound, and setting $N = \frac{2m}{\epsilon^2} \ln \frac{2m}{\delta}$, the probability that the ranking is not an ϵ-Borda ranking is at most $2me^{-\epsilon^2 N/(2m)} = \delta$. It immediately follows that the top idea is an ϵ-Borda winner with probability at least $1 - \delta$. □

When the number of candidates is comparable to the number of voters, i.e. $m = O(n)$, then Theorem 3 states that each voter only needs to make $O(\log m)$ comparisons, which is exponentially better than the $O(m)$ comparisons required by the lower bound. Note for concreteness that for $m \leq 1{,}000{,}000$, $\log m \leq 20$. We note the surprising simplicity of Algorithm 1. The simplicity is so extreme that one does not know whether to be delighted or disappointed.

For practical considerations, we also note that one can decrease the number of samples required by a factor of 2 by decrementing the count of the losing idea in Algorithm 3.

A brief note on the importance of honesty

We take a brief detour to discuss Theorem 3 in light of the previously known lower bound. Let $\gamma = 1 - \frac{1}{\sqrt{2}}$. In (Service and Adams 2012), it was stated that any deterministic algorithm producing the ϵ-Borda winner, for $\epsilon \in [0, \gamma]$, must elicit $\Omega(\frac{(\gamma-\epsilon)^3}{\log(1/(\gamma-\epsilon))} mn - \log\log \frac{1}{\gamma - \epsilon})$ bits of information[3].

As noted in (Conitzer and Sandholm 2005), the method that is used (of fooling sets) means that this lower bound applies not only to deterministic algorithms, but also to any nondeterministic algorithm which is able to find the approximate Borda winner along some computation path. In other words, this lower bound applies to all algorithms which are able to produce a "certificate" of their computation which, if shown to some neutral party, can be used to verify that the produced candidate is indeed an ϵ-Borda winner. The reason that Algorithm 1 is able to circumvent this lower bound is that it is not able to produce such a certificate. That is, it is possible that after eliciting more comparisons, we find that our given candidate was not an ϵ-Borda after all. The algorithm relies on the uniform sampling used to obtain probabilistic guarantees regarding how the other comparisons are distributed.

This observation results in an interesting and thought-provoking insight on the importance of honesty (or trust, security, etc…) in participatory democracies. Specifically, if the governing body in charge of the elicitation process is required to produce evidence that the comparisons elicited prove that some candidate is an ϵ-Borda winner, then the lower bound given in (Service and Adams 2012) states that it is impossible to elicit efficiently (do better than $O(m)$ comparisons per participant). However, if the governing body is honest in choosing its comparisons uniformly at random, if the people trust that this is true, and if the algorithm is secure to outside manipulations of its randomness, then we

[3]We note that their result is stated in terms of $\delta = \gamma - \epsilon$. We translated it here to be consistent with the notation we have been using.

ALGORITHM 2: Approximating the Condorcet winner

Input: m ideas, n participants, number of candidate pairs N_1, participant sample size N_2
Output: An output ranking
count$[\cdot] = 0$;
for $i \leftarrow 1$ **to** N_1 **do**
  Sample ideas c_1 and c_2 and N_2 participants uniformly at random;
  if c_1 *receives at least* $\frac{1}{2}(1-2\epsilon)$ *fraction of the votes* **then**
    count$[c_1]$ = count$[c_1] + 1$;
  if c_2 *receives at least* $\frac{1}{2}(1-2\epsilon)$ *fraction of the votes* **then**
    count$[c_2]$ = count$[c_2] + 1$;
return any ranking such that for any x ranked higher than y, count$[x] \geq$ count$[y]$;

are suddenly able to reduce the elicitation time to $O(\log m)$ comparisons per participant through the use of Algorithm 1.

Eliciting the Condorcet winner with another simple sampling algorithm

The algorithm for eliciting the Condorcet winner (Algorithm 2) has a very similar flavor to that of the Borda rule. At each step, sample a set of N_1 participants uniformly at random and ask them to compare two ideas sampled uniformly at random. Increment a counter for the idea chosen by a larger number of the sampled participants and repeat N_2 times. Now form an output ranking by ordering the ideas from those with the highest to lowest counter values, with ties broken arbitrarily.

Theorem 4. *For any $\epsilon, \delta \in (0,1)$, consider Algorithm 2 with $N_1 = O(\frac{m}{\epsilon^2}\ln\frac{m}{\delta})$ and $N_2 = O(\frac{1}{\epsilon^2}\ln\frac{N_1}{\delta})$. Then if an ϵ-Condorcet winner exists, then the top idea in the returned ranking is a 3ϵ-Condorcet winner with probability at least $1-\delta$.*

Proof. Suppose that $O(\frac{1}{\epsilon^2}\ln\frac{\alpha}{\delta})$ voters are chosen to vote in each round. By Chernoff bounds, we can conclude that the following "bad" events can happen with probability at most $O(\frac{\delta}{\alpha})$ for a given round.

1. A candidate who is preferred by at least $\frac{n}{2}(1-\epsilon)$ voters receives less than $\frac{1}{2}(1-2\epsilon)$ fraction of the sampled votes.
2. A candidate who is preferred by at most $\frac{n}{2}(1-3\epsilon)$ voters receives more than $\frac{1}{2}(1-2\epsilon)$ fraction of the sampled votes.

Choosing $\alpha = N_1$ then guarantees that the probability of any of these happening for any of the rounds is at most $O(\delta)$. Now suppose that $O(\frac{m}{\epsilon^2}\ln\frac{\beta}{\delta})$ pairs of candidates are chosen. Then by Chernoff bounds, assuming that none of the previously mentioned bad events occur, we can conclude that the following "bad" events can happen with probability at most $O(\frac{\delta}{\beta})$ for a given candidate.

1. A candidate who receives at least $\frac{n}{2}(1-\epsilon)$ votes against at least $(1-\epsilon)(m-1)$ candidates receives counter increments for less than $\frac{2}{m}(1-2\epsilon)$ fraction of the sampled pairs.
2. A candidate who receives at most $\frac{n}{2}(1-3\epsilon)$ votes against at least $3\epsilon(m-1)$ candidates receives counter increments for more than $\frac{2}{m}(1-2\epsilon)$ fraction of the sampled pairs.

Choosing $\beta = m$ then guarantees that the probability of any of these happening for any of the candidates is at most $O(m)$ so we are done since this implies that any candidate who is not a 3ϵ-Condorcet winner must have a lower count than the ϵ-Condorcet winner. □

Although this result is only interesting when an ϵ-Condorcet winner exists, it can be generalized to finding an ϵ-Copeland winner using the same argument. Unfortunately, although it is intuitive that the ranking returned by Algorithm 2 might give an ϵ-Copeland winner, it is not straightforward to prove that this is true.

Using the Kendall-tau distance to define approximations

In the previous sections, our broader definition of an ϵ-approximate winner or ranking was a candidate or ranking that could be made the output by changing at most ϵ fraction of the comparison values involving each candidate. There are two potentially undesirable properties of this definition. First, the resulting set of comparisons may not actually be achievable with a set of rankings since we allow swapping arbitrary comparisons. Second, this formulation treats all comparisons equally. This may not be natural since swapping a first and last ranked candidate is not a "small perturbation" in the participant preferences.

To deal with this, one can generalize the previous definitions: given any distance metric d between two preference profiles and the true profile $\mathcal{P}$, a ranking or candidate is an ϵ-approximate ranking or winner with respect to d if it is the winning ranking or candidate under a profile $\mathcal{P}'$ such that $d(\mathcal{P}, \mathcal{P}') \leq \epsilon$. This notion of using distances between preference profiles is the same setup as the literature on distance rationalizability in which voting rules are interpreted as finding the closest "consensus profile" to the given profile (Meskanen and Nurmi 2008; Elkind, Faliszewski, and Slinko 2010).

With this framework, a better distance between two preference profiles might be the number of *adjacent* swaps needed to go from one profile to the other, divided by $(m-1)n$ which is the total number of comparisons that a given candidate is a part of. It is not hard to show that Algorithm 1 also returns an ϵ-Borda winner under this definition using essentially the same number of comparisons.

Theorem 5. *For any $\epsilon, \delta \in (0,1)$, the top idea in the returned ranking of Algorithm 1 with $N = O(\frac{m}{\epsilon^2}\ln\frac{m}{\delta})$ is an ϵ-Borda winner with respect to the Kendall-tau distance defined above with probability at least $1-\delta$.*

Proof. The proof is similar to that of Theorem 3. Let x be the returned winner and x^* be the true Borda winner. By the proof of Theorem 3, and using 4 times the number of comparisons, the Borda score of x must be at least the

Borda score of x^* minus $\epsilon n(m-1)$. Then consider making $\epsilon n(m-1)$ adjacent swaps in the original profile, each of which moves x to a higher position. In this new profile, the Borda score of x increased by $\epsilon n(m-1)$ and the Borda score of all other candidates either stayed the same or went down. Therefore, x is the Borda winner in this new profile and we are done. □

Unfortunately, Algorithm 2 is unable to find an ϵ-Condorcet winner under this definition.

Empirical Insights on Elicitation from the Finnish experiment

In this section, we introduce some experimental work done in collaboration with the Finland Ministry of the Environment on crowdsourcing the Finland off-road traffic law. First, we analyze Algorithm 1 on the comparison data collected for 41 different topics, showing that the total number of comparisons required to achieve $\epsilon = 0.05, 0.1$ scales linearly (slightly smaller than the $O(m \log m)$ guarantees). The practicality of the given algorithm is further supported by showing that the constant involved in the scaling is not large.

Following this, we present an HCI (human computer interaction) centric approach to reducing elicitation time. Specifically, we analyze the time it takes for participants to rank sets of proposals of varying sizes, with the hypothesis that the time vs. information tradeoff may be favorable for small sets. When plotting the time per bit elicited against the size of the rankings, we find that eliciting partial rankings of up to size six can reduce elicitation time by around a factor of three.

The Finnish experiment: background and experimental setup

The Finnish experiment was an experiment in policy crowdsourcing aiming to reform an existing law on off-road traffic regulation, (which applied essentially to the regulation of snowmobiles and ATVs in nature). The crowdsourced law reform was organized in collaboration with the Finland Ministry of the Environment over a period of several months (January 2013 to October 2013) and sought to engage the Finnish people in three successive tasks: 1) identifying problems with the existing law, 2) proposing solutions or elements of solution to these problems, and 3) evaluating the generated ideas so as to offer the civil servants and members of Parliament a ranking of the top ideas according to the crowd. The crowdsourcing phase of the experiment was largely successful and is now over. It remains to be seen whether the classical representative institutions and actors involved will now use the crowd's input in any productive and significant way (for more details, see (Aitamurto and Landemore 2013)).

This paper will focus on the evaluation portion of the experiment, as related to the use of social choice functions for participatory democracies. In-depth information and insights learned on the overall participatory process will be published in a forthcoming paper.

The evaluation was carried out in the following way: participants logged into a website and were presented with a series of "actions", each of which could take the form of a rating, a comparison, or a ranking. A rating action presented the participant with a single idea and asked for a rating from one to five stars; a comparison action presented the participant with two ideas and asked which one they thought was better; finally, a ranking action presented the participant with three or more ideas and asked them to order them from most liked to least liked.

There were 308 participants, 41 different topics, and 240 total ideas among the different topics. The topics ranged in size from two to fifteen ideas. The number of topics of a certain size was $5, 10, 5, 4, 4, 3, 1, 2, 0, 3, 1, 0, 1, 2$ for sizes two to fifteen respectively.

For most of the topics[4], the following process was used to assign actions to each participant (all randomness is independent unless otherwise stated). Let m denote the number of proposals in the topic.

1. For each proposal, a rating action is assigned with probability 0.18.
2. For each pair of proposals, a comparison action is assigned with probability p, where $p = 0.3, 0.15, 0.03, 0.02$ depending on if $m = 2$, $m = 3, 4$, $m = 5$, or $m > 5$ respectively.
3. If $m \geq 5$, then a single ranking action is assigned of length $\lceil m/2 \rceil$ with probability 0.5. If the preceding action was not assigned, and $m \geq 7$, then a single ranking action is assigned of length 3 with half the remaining probability (0.25).

In expectation, each participant received around 84 actions, of which 25 were comparisons, 44 were ratings, and 15 were rankings of varying lengths. In total, $72{,}003$ effective comparisons and $13{,}300$ ratings were collected.

As an example of types of ideas we saw, one of the topics was: "Establishing new routes: What would be the best way to regulate the establishment of a new route on landowner property even if the landowner is resisting the route?" The five proposals for this topic were:

1. It should not be possible to establish a route on private property without the landowner's consent. Landowners should have an ultimate right to control their own property.
2. It should be possible to establish a route on private property without the landowner's consent but only when this does not cause harm to the landowner.
3. It should be possible to establish a route on private property, but the need for a route should be justified by more important reasons than improving the public traffic network or common recreational use.

[4] Two of the topics of length 11 were special in that they listed topic categories, and asked participants to rank them either in order of importance or in predicted order of importance. All participants were asked to rank the full list of topic categories based on these two metrics, and no ratings were collected. For one of the topics of length 5, we asked all participants to rate and rank all the ideas.

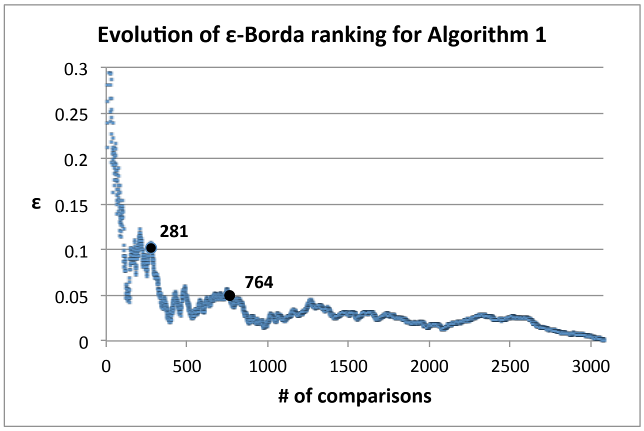


Figure 2: This shows a single simulation for the evolution of the ϵ-Borda ranking of Algorithm 1. There were 308 participants, all of which were asked to rank all 5 proposals of this topic, resulting in a total of 3080 effective comparisons. One can see that ϵ stays below 0.05 and 0.1 after 764 and 281 comparisons respectively. Since there were 308 participants, the number of comparisons per participant is only around two or one respectively.

4. There should be more strict and clear criteria in the existing law for justifying bypassing landowners' consent when establishing a new route.
5. It should be possible to establish a route on private property without the landowner's consent whenever there's a need to set up a route.

Scaling of Algorithm 1 in the Finnish experiment

One challenge that often arises in applying algorithmic theorems to practice is that even though the asymptotic behavior is good, the constants involved may still render the algorithm impractical. For instance, the factor of $\frac{1}{\epsilon^2}$ in Theorem 3 is a factor of 400 for $\epsilon = 0.05$. We analyze the scaling of Algorithm 1 in the Finnish experimental data. Recall that Theorem 3 stated that for fixed ϵ, the total number of comparisons required to find an approximate Borda winner or ranking is $O(m \log m)$.

For $\epsilon = 0.05$ and 0.1, we show that the total number of comparisons required scales linearly in m,[5] and that the constant involved is not large. As an example, when the derived trend is extrapolated to the case of aggregating 100 ideas with 1000 participants, an error of 0.05 only requires each participant to make 19 comparisons. If we only need an error of 0.1, then 8 comparisons suffice.

Our method is as follows. For each topic, we compute $n(x)$ as the total normalized number of collected comparisons for which x won. We note that this is not the true value of $n(x)$ since we did not collect the full set of comparisons. However, it is a good approximation because of

[5]This is a log factor smaller than the upper bound. This could be because the range of m is not large enough to detect the log factor or because the distribution of rankings in practice is slightly better than the worst case.

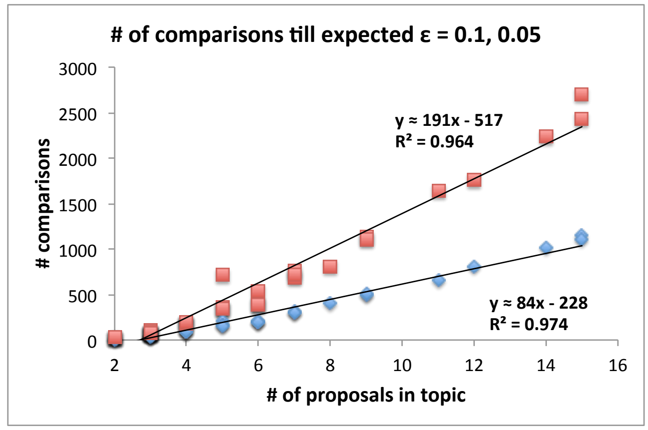


Figure 3: The red and blue series show the number of comparisons needed to reach an expected error of 0.05 and 0.1 respectively. Each point represents one of the 41 topics.

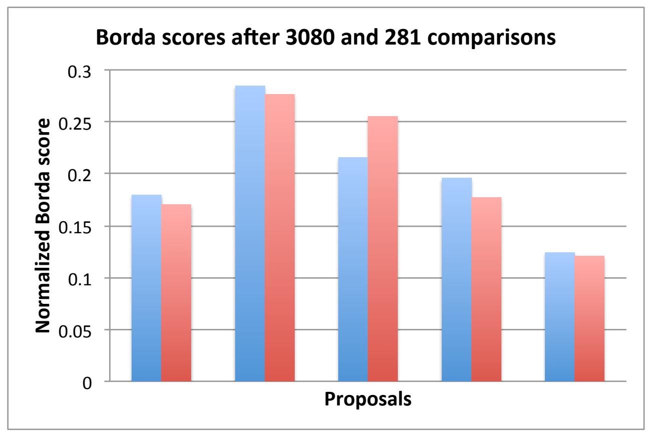


Figure 4: The blue series plot the true normalized Borda scores, while the red series plot the sampled Borda after 281 comparisons, which corresponds to $\epsilon = 0.1$.

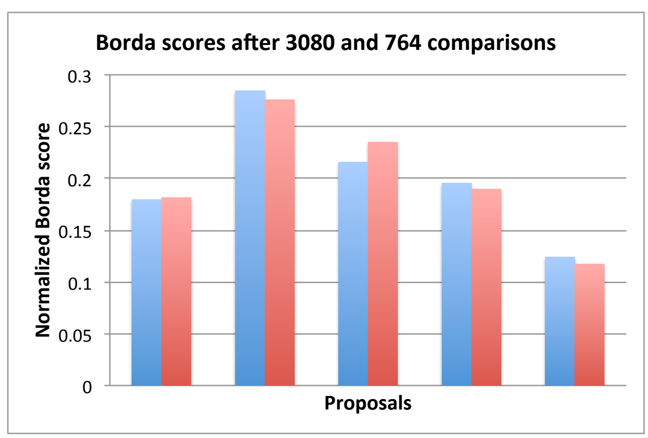


Figure 5: The blue series plot the true normalized Borda scores, while the red series plot the sampled Borda after 764 comparisons, which corresponds to $\epsilon = 0.05$.

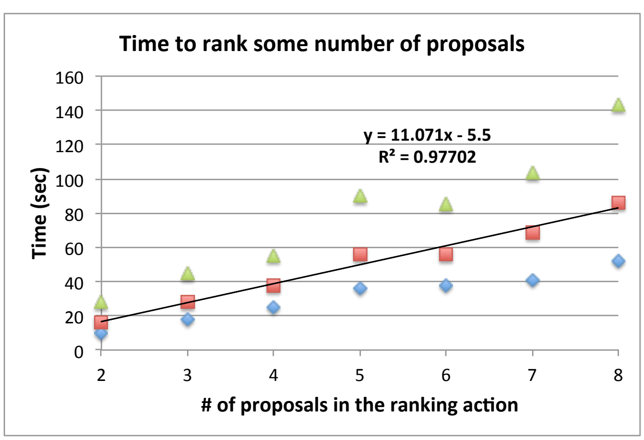
the large number of users and comparisons we sample (over one-eighth of all possible comparisons) at random. Algorithm 1 is then simulated by shuffling the collected comparisons randomly so that the ordered comparisons correspond to a sequence of samples of Algorithm 1. We repeat this 100 times and calculate the average value of ϵ achieved at each point in time (see Figure 2 for an example of how ϵ evolves in a single simulation - the average values are of course much smoother). We find the time at which ϵ equals 0.05 and 0.1 and plot this against the number of ideas in that topic.

Our results are summarized in Figure 3. The main observation to note is that the data series has a good linear fit, and that the constants are reasonable. Given a number of ideas, one can use the linear trend to calculate the total number of comparisons needed to achieve a desired approximation. Since the comparisons are randomly assigned, the expected number of comparisons per participant can be calculated by dividing the resulting number by n. For instance, when $x = 100$ and $n = 1000$, the linear trends indicate that $(191 * 100 - 517)/1000 \approx 18.6$ comparisons per participant are needed to achieve $\epsilon = 0.05$ and $(84 * 100 - 228)/1000 \approx 8.2$ are needed for $\epsilon = 0.1$. If one only needs to find winning ideas, one can do even better.

In the example simulation shown in Figure 2, the error reached and stayed below 0.1 at 281 comparisons, and did so for an error of 0.05 after 764 comparisons. Figures 4 and 5 give a sense of what $\epsilon = 0.1$ and $\epsilon = 0.05$ mean by plotting the normalized borda score vectors $\hat{n}(x)$ as calculated by the counts of Algorithm 1 along with the real borda scores $n(x)$. It is important to point out that the rankings returned at each of these times is already exactly identical to the true ranking, which means that ϵ is actually equal to 0 by the broader definition we give. Our calculation of ϵ in these simulations is actually only an upper bound on the true error since we calculate it based on one specific choice of the normalized borda score vector, and not the one which finds the tightest bound on ϵ.

Decreasing elicitation time with partial rankings

As noted previously, the comparison data we collected was elicited through rankings of different sizes. We find that eliciting partial rankings may be a useful approach to decreasing evaluation time.



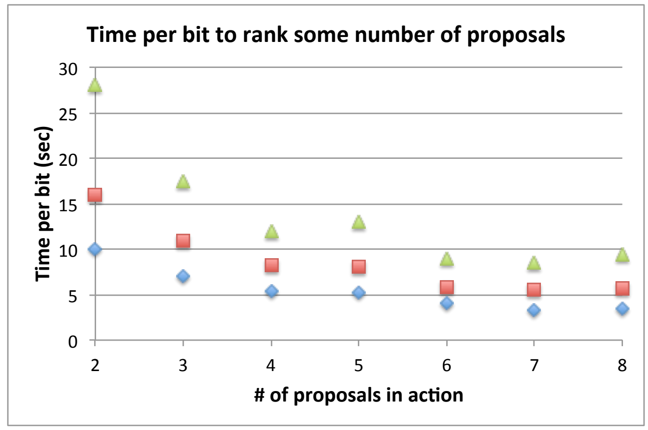


Figure 6: (a) The three series are the three quartiles (Q1, Q2, Q3) of the time it takes to rank some number of proposals. We can see that this time scales linearly in this range. (b) When we normalize by the number of bits in a ranking ($\log_2 k!$ for a ranking of size k), we see that the time per bit of information drops off so that by six proposals, the time per bit has been cut to one-third of the initial time.

Figure 6a plots the time it takes for participants to complete rankings of differing sizes. Since there were many outliers (possibly occuring when participants left their browser window open), we plot these times in terms of the quartiles. The second quartile is the median value, and the first and third quartiles mark off the middle 50% of the values. One can see that the time approximately follows a linear trend in this regime.

Figure 6b plots the time per bit of information that participants take to complete a ranking of size k. Since there are $k!$ different rankings of length k, the number of bits in a ranking is $\log_2 k!$. By using partial rankings of length three, one can already reduce the elicitation time by around one-third. With partial rankings of length four or six, the elicitation time is reduced by around half or two-thirds respectively.

Further improvements for use in practice

Dynamic sampling improves elicitation time

Algorithms 1 and 2 use simple random sampling to significantly improve elicitation time from previous lower bounds. In this section, we show that if one only needs to find a single winning idea (and not a full ranking), one can further reduce elicitation time by dynamically sampling comparisons through the use of multi-armed bandit algorithms. The intuition is that one should be able to quickly get rid of ideas which are ranked very poorly and for which we can quickly determine that they are not ϵ-winners. While we will spend most of our time considering how to find an ϵ-Borda winner, the techniques used here can be applied in the exact same manner to decreasing elicitation time for finding an ϵ-Condorcet winner.

Borda The main intuition to Theorem 3 was to notice that the Borda score of an idea x is proportional to the probability that x wins a randomly sampled comparison. This allows us to view Algorithm 1 from the perspective of multi-armed bandits (Bubeck and Cesa-Bianchi 2012). In this setting, each idea $x \in C$ can be viewed as an arm (think of a slot machine in a casino) which, when pulled, returns a reward of either 1 with probability $p(x) = s(x)/(n(m-1))$ or 0 otherwise[6]. Each time we elicit a random comparison containing x, this is analogous to pulling the arm since the probability of winning the comparison is exactly $p(x)$. The goal is then to identify the best arm (or approximately best arm) with a small number of arm pulls.

This problem is known as the 'best arm identification' problem in the multi-armed bandit literature and has been studied by (Even-Dar, Mannor, and Mansour 2006). They propose an algorithm called Successive Elimination (Algorithm 3), which we adapt to our setting.

Theorem 6. *For any $\epsilon, \delta \in (0, 1)$, Algorithm 3 with $N = O(\frac{1}{\epsilon^2} \ln \frac{m}{\delta})$ returns an ϵ-Borda winner with probability at least $1 - \delta$. Moreover, the total number of comparisons elicited is*[7]

$$O\left(\sum_{x \in C} \min\left[\frac{1}{\epsilon^2}, \frac{1}{(p(x^*) - p(x))^2}\right] \ln \frac{m}{\delta\epsilon}\right)$$

Proof. We sketch the proof here for the reader's convenience. Details can be found in (Even-Dar, Mannor, and Mansour 2006). Let $N(x)$ be the total number of rounds for which idea x is in set S, and let T = the total number of comparisons elicited. Clearly, $T = \sum_x N(x)$.

The basic idea is to first show that for any idea x, UB$[x]$ after $t = O(\frac{1}{(p(x^*)-p(x))^2} \ln \frac{m}{\delta\epsilon})$ rounds must be less than LB$[x^*]$ at t. Once this happens, x is no longer in the set S which means that $N(x) = O(\frac{1}{(p(x^*)-p(x))^2} \ln \frac{m}{\delta\epsilon})$. But since we also know that the maximum number of rounds is $N = O(\frac{1}{\epsilon^2} \ln \frac{m}{\delta})$, this implies that $N(x) = O(\min\left[\frac{1}{\epsilon^2}, \frac{1}{(p(x^*)-p(x))^2}\right] \ln \frac{m}{\delta\epsilon})$. Summing up over all ideas gives us our bound for T.

What remains is to show that the idea returned is an ϵ-Borda winner. To do this, one can show that with probability at least $1 - \delta$,

- any non ϵ-Borda winner will have been removed from S after all N rounds, and
- x^* is never removed from S.

[6]In Algorithm 1, we sampled random pairs of ideas at a time. This is equivalent to pulling arms corresponding to each of the sampled ideas (even though they are dependent, all pulls of a given arm are still independent). However, it is easier to see the analogy to multi-armed bandits if one considers a slight variation in which each idea x is compared against $\frac{1}{\epsilon^2} \log \frac{m}{\delta}$ random idea (with random voters) with count$[x]$ equaling the number of wins received.

[7]Slightly tighter bounds (within the log factors) can be given for both N and the total number of comparisons elicited. However, we chose to present the bound in this way to best communicate the intuition.

ALGORITHM 3: Dynamically approximating the Borda rule via Even-Dar et al.'s Successive Elimination

Input: m ideas, n participants, a number of rounds N
Output: An output winner
$S = C$;
count$[\cdot] = 0$;
for $t \leftarrow 1$ **to** N **do**
  for $x \in S$ **do**
    Sample an idea y and a participant v uniformly at random;
    if $x \succ_v y$ **then**
      count$[x]$ = count$[x] + 1$;
    UB$[x]$ = count$[x] + \sqrt{t \ln \frac{4mt^2}{\delta}}$;
    LB$[x]$ = count$[x] - \sqrt{t \log \frac{4mt^2}{\delta}}$;
  $S = S \setminus \{x \in S : \text{UB}[x] \leq \text{LB}[y] \text{ for some } y\}$;
return the idea maximizing count$[\cdot]$;

This means that the idea maximizing count$[\cdot]$ must be an ϵ-Borda winner. □

We can now apply this to a simple voting scenario in which people are mostly divided into two camps which vote in opposite manners. As shown in Corollary 1, if one camp clearly dominates the other, then Algorithm 3 is essentially able to shave off an extra factor of $\frac{1}{\epsilon}$ to a total of only $O(\frac{m}{\epsilon} \ln \frac{m}{\delta\epsilon})$ comparisons.

Corollary 1. *Let $C = \{0, 1, \ldots, m-1\}$ and consider a set V of voters, where $|V| = n$. Suppose that $k = \frac{n}{2}(1+\gamma)$ voters have the ranking $0 \succ 1 \succ \ldots \succ m-1$ and that the other $\frac{n}{2}(1-\gamma)$ have the ranking $m-1 \succ m-2 \succ \ldots \succ 0$, where $\gamma \in [0, 1]$. Then Algorithm 3 will require*

$$O\left(\frac{m}{\epsilon \max(\epsilon, \gamma)} \log \frac{m}{\delta\epsilon}\right)$$

comparisons.

Proof. It is not hard to verify that the Borda score of idea i is $s(i) = \frac{n}{2}(1+\gamma)i + \frac{n}{2}(1-\gamma)(m-1-i) = \frac{n}{2}(1-\gamma)(m-1) + n\gamma i$ and that

$$p(i) = \frac{1}{2}(1-\gamma) + \frac{\gamma i}{m-1}$$

The Borda winner s^* is idea $m-1$ with $p(m-1) = \frac{1}{2}(1+\gamma)$ so that all ideas with $p(i) \geq \frac{1}{2}(1+\gamma)(1-\epsilon)$ are ϵ-Borda winners. Plugging in and solving shows us that all ideas $i \geq \max(0, 1 - \frac{1+\gamma}{2\gamma}\epsilon)(m-1)$ are ϵ-Borda winners.

If $\frac{1+\gamma}{2\gamma}\epsilon \geq 1$, then all ideas are ϵ-Borda winners, so that we still require $O(\frac{m}{\epsilon^2} \log \frac{m}{\delta})$ comparisons. However, if $\frac{1+\gamma}{2\gamma}\epsilon < 1$, then we have that the total number of comparisons elicited are $O(A + B)$, where

$$A = \frac{1+\gamma}{2\gamma}\epsilon(m-1)\frac{1}{\epsilon^2} \log \frac{m}{\delta} \leq \frac{m}{\gamma\epsilon} \log \frac{m}{\delta}$$

and

$$B = \left[\sum_{i=0}^{(m-1)(1-\frac{1+\gamma}{2\gamma}\epsilon)} \left(\frac{m-1}{\gamma(m-1-i)}\right)^2\right] \log \frac{m}{\delta\epsilon}$$

$$= \frac{(m-1)^2}{\gamma^2} \log \frac{m}{\delta\epsilon} \sum_{i=0}^{(m-1)(1-\frac{1+\gamma}{2\gamma}\epsilon)} \frac{1}{(m-1-i)^2}$$

$$\leq \frac{(m-1)^2}{\gamma^2} \log \frac{m}{\delta\epsilon} \frac{1}{m(\frac{1+\gamma}{2\gamma})\epsilon}$$

$$\leq \frac{2m}{\gamma\epsilon} \log \frac{m}{\delta\epsilon}$$

Noting that $\gamma \geq \epsilon$ in this case concludes our proof. □

Condorcet The method for dynamically sampling an ϵ-Condorcet winner is similar, so we will only briefly describe it. In Algorithm 2, the outer loop is essentially doing the same process as Algorithm 1, except that each pair of ideas gets compared by multiple voters. Therefore, we can apply essentially the same modification of Algorithm 1 to Algorithm 3 to the outer loop of Algorithm 2. The only difference is that ϵ is relative to a $p(x)$ of 1 since a Condorcet winner is defined as an idea which beats *all* other ideas. Because of this, we do not need to keep track of lower bounds. We simply remove ideas as their upper bound falls below $1-\epsilon$ since this means they cannot be an ϵ-Condorcet winner.

This gives us the bound of

$$N_1 = O\left(\sum_{x\in C} \min\left[\frac{1}{\epsilon^2}, \frac{1}{(1-p(x))^2}\right] \ln \frac{m}{\delta\epsilon}\right)$$

where $p(x)$ is the fraction of ideas against which x wins at least $\frac{1}{2}(1-2\epsilon)$ fraction of votes. However, we can further improve this to get that

$$N_1 = O(\frac{m}{\epsilon} \log \frac{m}{\delta\epsilon})$$

independent of the voter rankings. This is because only $O(k\epsilon)$ fraction of candidates can have $p(x) \geq 1 - k\epsilon$, which allows us to upper bound the general bound.

Handling streaming ideas

Another challenge in implementing the aforementioned algorithms in practical crowdsourcing scenarios is that ideas may be arriving in a streaming manner as they are generated by crowdsourcing participants. In such scenarios, one needs to be able to return the ϵ-approximate winner at any given time. This may cause one to sample more comparisons than needed for the early ideas than one needs once all the ideas have been generated.

We will only discuss the case of finding an ϵ-Borda winner. However, the exact same modification can be made to finding an ϵ-Condorcet winner. Concretely, we consider times $t = 1, 2, \ldots, m$. Idea c_t is generated at time t. Any number of comparisons can then be sampled, but one needs to be able to return an ϵ-Borda winner at that time.

Unlike the prior section, we are unable to take advantage of results in the multi-armed bandit literature. In the standard multi-armed bandit setting, dynamically arriving arms can be easily dealt with since the arms are independent. In the case of our voting scenario, this is not true since a new idea *also changes the Borda score (and thus the probabilities) of existing ideas*. This in turn invalidates the past comparisons sampled since they were sampled for a different $p(x)$.

A naive approach would be to simply toss away the old comparisons and start over. This would be disastrous since the number of comparisons sampled for each arm would then increase to $\sum_{k=1}^{m} O(\log k) = O(m \log m)$. In this section, we show that one can handle dynamically arriving ideas without increasing the number of comparisons elicited significantly. The key is to notice that we can reuse most of our past comparisons.

Our algorithm for sampling comparisons at each time t is codified in Algorithm 4. At the beginning of time $t = k+1$, there are $k+1$ total ideas, $C_{k+1} = \{c_1, c_2, \ldots, c_{k+1}\}$. The first k arms $c_1, c_2, \ldots, c_k$ have already been pulled $\frac{c}{\epsilon^2} \log \frac{k}{\delta}$ times, but with incorrect sampling since c_{k+1} did not exist at that time. Nevertheless, an important fact is that for any given arm $x \in C_k$, each pull is the result of a randomly sampled voter v and the a randomly sampled idea from $C_k \setminus x$.

Thus, the following procedure is a valid sample from the newly updated arm x: with probability $\frac{k-1}{k}$, reuse a comparison from the previous round; otherwise, sample a random voter v and have him compare x with c_{k+1}[8]. We can repeat this until we have $\frac{c}{\epsilon^2} \log \frac{k+1}{\delta}$ valid samples.

Theorem 7. *Algorithm 4 elicits comparisons that allows one to find the ϵ-Borda winner at each time. Moreover, at time $t = m$, the total number of comparisons elicited for one arm throughout the process is*

$$O\left(\frac{1}{\epsilon^2} \log \frac{m}{\delta} \ln m\right)$$

with probability at least $1-\delta$.

Proof. At any time k, one can find an ϵ-Borda winner by simply calculating the number of wins in $Z'(x)$ for each idea x and then returning the idea with the largest number of wins. This follows from the fact that Algorithm 4 ensures that $Z'(x)$ consists of $\frac{c}{\epsilon^2} \log \frac{k}{\delta}$ randomly sampled comparisons, which allows us to apply the same proof as Theorem 1.

Therefore, our main challenge is to show that for any m, the total number of comparisons elicited in times $t = 2, \ldots, m$ is not too large (at time $t = 1$, there are no comparisons elicited since there is only one idea).

Note that any newly sampled comparison corresponds to some u_{ki}, $k \in \{2, 3, \ldots, m\}$, $i \in \{1, 2, \ldots, \frac{c}{\epsilon^2} \log \frac{k}{\delta}\}$ and only in the condition that either

- $u_{ki} \leq \frac{1}{k-1}$ or,

[8]Note that we are only reusing comparisons from the previous round for simplicity. One can generalize the given algorithm appropriately to reuse comparisons from all prior rounds.

ALGORITHM 4: Streaming Borda subroutine after idea c_{k+1} arrives

Input: The newly submitted idea c_{k+1}, the n participants, the k prior ideas C_k, and the set of $\frac{c}{\epsilon^2}\log\frac{k}{\delta}$ prior comparisons $Z(x)$ for each $x \in C_k$ (each of which is represented as a tuple (c_1, c_2, v) containing the two ideas compared and the voter)

Output: A set $Z'(x)$ of $\frac{c}{\epsilon^2}\log\frac{k+1}{\delta}$ corresponding comparisons per idea

for $x \in C_k$ **do**
- $Z'(x) = \emptyset$;
- **for** $i \leftarrow 1$ **to** $\frac{c}{\epsilon^2}\log\frac{k+1}{\delta}$ **do**
  - $u_{(k+1)i} \sim$ Unif[0, 1];
  - **if** $u_{(k+1)i} \le (k-1)/k$ *and* $|Z(x)| > 0$ **then**
    - Let z be an arbitrary comparison in $Z(x)$;
    - $Z(x) = Z(x) \setminus z$;
    - $Z'(x) = Z'(x) \cup z$;
  - **else**
    - Sample a participant v uniformly at random;
    - $Z'(x) = Z'(x) \cup (x, c_{k+1}, v)$;

$Z'(c_{k+1}) = \emptyset$;

for $i \leftarrow 1$ **to** $\frac{c}{\epsilon^2}\log\frac{k+1}{\delta}$ **do**
- Sample an idea y and a participant v uniformly at random;
- $Z'(c_{k+1}) = Z'(c_{k+1}) \cup (c_{k+1}, y, v)$;

return $Z'(x)$ for all x;

- there are no more comparisons from the previous round to reuse.

Let I_{ki} be an indicator variable for the event in which a newly sampled comparison is made after sampling u_{ki}. The total number of comparisons elicited is then

$$T = \sum_{k=2}^{m} \sum_{i=1}^{\frac{c}{\epsilon^2}\log\frac{k}{\delta}} I_{ki}$$

We now split all u_{ki} into two sets: $U_1 = \{u_{ki} : 1 \le i \le \frac{c}{\epsilon^2}\log\frac{k-1}{\delta}\}$ and $U_2 = \{u_{ki} : \frac{c}{\epsilon^2}\log\frac{k-1}{\delta} < i \le \frac{c}{\epsilon^2}\log\frac{k}{\delta}\}$.

For $u_{ki} \in U_1$, it cannot be the case that there are no comparisons from the previous round to reuse since the previous round has $\frac{c}{\epsilon^2}\log\frac{k-1}{\delta}$ comparisons. Therefore, a newly sampled comparison only occurs iff $u_{ki} \le \frac{1}{k-1}$. From this observation, and by defining $T_1 = \sum_{k,i:u_{ki}\in U_1} I_{ki}$, we have

$$\begin{aligned}
\mathbb{E}[T_1] &= \sum_{k=2}^{m} \sum_{i=1}^{\frac{c}{\epsilon^2}\log\frac{k-1}{\delta}} \frac{1}{k-1} = \sum_{k=1}^{m-1} \sum_{i=1}^{\frac{c}{\epsilon^2}\log\frac{k}{\delta}} \frac{1}{k} \\
&= \sum_{k=1}^{m-1} \frac{1}{k}\frac{c}{\epsilon^2}\log\frac{k}{\delta} \le \int_1^m \frac{1}{k}\frac{c}{\epsilon^2}\log\frac{k}{\delta}\,dk \\
&= \int_{\ln\frac{1}{\delta}}^{\ln\frac{m}{\delta}} \frac{c}{\epsilon^2} u\,du = \frac{c}{2\epsilon^2}\left[\ln^2\frac{m}{\delta} - \ln^2\frac{1}{\delta}\right] \\
&\le \frac{c}{\epsilon^2}\ln\frac{m}{\delta}\ln m
\end{aligned}$$

But then a Chernoff bound gives

$$\mathbf{Pr}[T_1 \ge \frac{2c}{\epsilon^2}\ln\frac{m}{\delta}\ln m] \le \exp(-\frac{c}{4\epsilon^2}\ln\frac{m}{\delta}\ln m) \le \frac{\delta}{m}$$

We now define $T_2 = \sum_{k,i:u_{ki}\in U_2} I_{ki}$. But since $I_{ki} \le 1$, we have the trivial bound of $T_2 \le |U_2| = \frac{c}{\epsilon^2}\ln\frac{m}{\delta}$. Therefore, since $T = T_1 + T_2$, we have that with high probability, $T = O(\frac{1}{\epsilon^2}\ln\frac{m}{\delta}\ln m)$. $\square$

Conclusion and Future Work

To conclude, our algorithms and experiments show that social choice functions that were previously thought to place high cognitive burdens on participants can indeed be implemented at scale, a promising sign for the use of crowdsourcing in democratic policy-making.

There are many directions to pursue for future work. The theoretical results we found relied heavily on nice interpretations of definitions for the Borda rule and the Condorcet winner. Continuing this work on other voting rules may require more involved algorithms. Also, it would be useful to find algorithms that elicit ϵ-Condorcet winners with respect to the swap distance as described in the case of the Borda rule.

Acknowledgments

This work was supported in part by the McMillan Center at Yale University, The Committee for the Future in the Parliament of Finland, and NSF Grant No. 0904325. David T. Lee is supported by an NSF graduate research fellowship and a Brown Institute for Media Innovation Magic Grant.

References

Aitamurto, T., and Landemore, H. 2013. Democratic participation and deliberation in crowdsourced legislative processes: the case of the law on off-road traffic in finland. In *The 6th Conference on Communities and Technologies (C&T). Workshop: Large-Scale Idea Management and Deliberation Systems*.

Aitamurto, T. 2012. Crowdsourcing for democracy: New era in policy-making. Committee for the Future, Parliament of Finland. Available online at http://www.scribd.com/doc/118136573/Crowdsourcing-for-Democracy-New-Era-in-Policy-Making.

Balinski, M., and Laraki, R. 2007. A theory of measuring, electing, and ranking. *Proceedings of the National Academy of Sciences (PNAS)* 104(21):8720–8725.

Brahbam, D. 2013. *Crowdsourcing*. Cambridge: MIT Press.

Brams, S. J., and Fishburn, P. C. 2002. Voting procedures. In Arrow, K. J.; Sen, A.; and Suzumura, K., eds., *Handbook of Social Choice and Welfare*, volume 1. Elsevier. 173–236.

Bubeck, S., and Cesa-Bianchi, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* 5(1):1–122.

Cabannes, Y. 2004. Participatory budgeting: A significant contribution to participatory democracy. *Environment and Urbanization* 16:27–43.

Caragiannis, I.; Procaccia, A. D.; and Shah, N. 2013. When do noisy votes reveal the truth? In *Proceedings of the 14th ACM Conference on Electronic Commerce (ACM-EC)*.

Chevaleyre, Y.; Lang, J.; Maudet, N.; and Monnot, J. 2011. Compilation and communication protocols for voting rules with a dynamic set of candidates. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*.

Conitzer, V., and Sandholm, T. 2002. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the 17th AAAI Conference*.

Conitzer, V., and Sandholm, T. 2005. Communication complexity of common voting rules. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM-EC)*, 78–87.

Conitzer, V. 2009. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research* 35:161–191.

Elkind, E.; Faliszewski, P.; and Slinko, A. M. 2010. On the role of distances in defining voting rules. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 375–382.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research* 7:1079–1105.

Fung, A., and Wright, E. O. 2003. *Deepening Democracy: Institutional Innovations in Empowered Participatory Governance*. Verso Press.

Goel, A., and Lee, D. T. 2012. Triadic consensus: A randomized algorithm for voting in a crowd. In *Proceedings of the 8th Workshop on Internet and Network Economics (WINE)*.

Kalech, M.; Kraus, S.; Kaminka, G. A.; and Goldman, C. V. 2011. Practical voting rules with partial information. *Journal of Autonomous Agents and Multi-Agent Systems* 22(1):151–182.

Landemore, H. 2014. Inclusive constitution-making: The icelandic experiment. *Journal of Political Philosophy*.

Lu, T., and Boutilier, C. 2011a. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*.

Lu, T., and Boutilier, C. 2011b. Vote elicitation with probabilistic preference models: empirical estimation and cost tradeoffs. In *Proceedings of the 2nd International Conference on Algorithmic Decision Theory (ADT)*.

Macpherson, C. B. 1977. *Life and Times of Liberal Democracy*. Oxford University Press.

Mao, A.; Procaccia, A. D.; and Chen, Y. 2013. Better human computation through principled voting. In *Proceedings of the 27th Conference on Artificial Intelligence (AAAI)*.

Meskanen, T., and Nurmi, H. 2008. Closeness counts in social choice. In Braham, M., and Steffen, F., eds., *Power, Freedom, and Voting*. Springer-Verlag.

Motwani, R., and Raghavan, P. 1995. *Randomized Algorithms*. Cambridge University Press.

Oren, J.; Filmus, Y.; and Boutilier, C. 2013. Efficient vote elicitation under candidate uncertainty. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.

Pateman, C. 1979. *The Problem of Political Obligation: A Critique of Liberal Theory*. New York: John Wiley & Sons.

Procaccia, A. 2008. A note on the query complexity of the condorcet winner problem. *Information Processing Letters* 108(6):390–393.

Salganik, M. J., and Levy, K. E. C. 2012. Wiki surveys: Open and quantifiable social data collection.

Service, T. C., and Adams, J. A. 2012. Communication complexity of approximating voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 593–602.

Smith, G. 2009. *Democratic Innovations: Designing Institutions for Citizen Participation*. Cambridge University Press.

Young, H. P. 1988. Condorcet's theory of voting. *The American Political Science Review* 82(4):1231–1244.