# Ahead of the Count: An Algorithm for Probabilistic Prediction of Instant Runoff Elections

Nicholas Kapoor and P. Christopher Staecker *

January 5, 2025

**Abstract**

How can we probabilistically predict the winner in a ranked-choice election without all ballots being counted? In this study, we introduce a novel algorithm designed to predict outcomes in Instant Runoff Voting (IRV) elections.

The algorithm takes as input a set of discrete probability distributions describing vote totals for each candidate ranking and calculates the probability that each candidate will win the election. In fact, we calculate all possible sequences of eliminations that might occur in the IRV rounds and assign a probability to each.

The discrete probability distributions can be arbitrary and, in applications, could be measured empirically from pre-election polling data or from partial vote tallies of an in-progress election.

The algorithm is effective for elections with a small number of candidates (five or fewer), with fast execution on typical consumer computers. The run-time is short enough for our method to be used for real-time election night modeling where new predictions are made continuously as more and more vote information becomes available. We demonstrate the algorithm in abstract examples, and also using real data from the 2022 Alaska state elections to simulate election-night predictions and also predictions of election recounts.

Keywords: Ranked choice voting, instant runoff voting, election predictions, probabilistic models, electoral outcomes, voting algorithms, political forecasting

## 1 Introduction

?⟨introsection⟩? In democracies worldwide, election seasons build up to the electrifying and high-stakes moment of election night, fueled by an insatiable need to know the

*Department of Mathematics, Fairfield University, Fairfield CT, USA, nicholas.kapoor@fairfield.edu (corresponding author), cstaecker@fairfield.edu.

winner. The analysis of the electorate's decisions – before they are known – begins immediately with exit polling and the first trickles of vote tallies. Due to different laws around counting votes, tallies are produced at varying rates, in some cases taking several days.

Regardless of the speed of counting the votes, different news agencies begin predicting the winner. In a traditional plurality voting system (sometimes called first-past-the-post, or FPTP), predicting the winner based on polling data and other models is fairly straightforward from a mathematical point of view. In this paper, we give a method for predicting the winner in elections using Hare Instant Runoff Voting (IRV), commonly called "ranked choice voting" (RCV) in the United States.

IRV elections have been used by various countries and localities worldwide for quite some time. Although this paper exclusively uses American examples, the mathematics and algorithm presented can apply to any IRV election.

In the United States, initiatives to replace FPTP elections with IRV elections have had several wins on municipal ballots [16], although, in 2024, the forward momentum was paused by various statewide ballot measure defeats [2]. Nevertheless, the IRV movement in the United States is still alive. It is a significant voting reform with many supporters who will indeed attempt to get IRV ballot measures on statewide ballots in the very near future. Therefore, even with the 2024 setbacks, IRV has growth potential in America. As IRV seeks to become a larger part of American politics and in the countries and cities across the globe where it is more established, different tools are required for analysts who want to predict election outcomes probabilistically. For FPTP elections, for example, *The New York Times* uses their infamous "needle" that tracks in real-time how likely one candidate is to defeat the other as the votes are being counted. Our paper will present a basic mathematical framework for the design of a similar "needle" for IRV elections. More generally, our work can be used to make probabilistic predictions of IRV elections long before Election Day based on polling or other data, or it can even be applied after elections to make predictions of recount results.

In the United States, the 2019 New York City Mayoral and City Council Democratic primaries were among the first significant uses of IRV in the United States in multiple races on the same Election Day. In [10], Jelvani and Marian provide an algorithm to determine the possible elimination outcomes in the 2019 New York City Democratic Mayoral primary based on incomplete tallying of the ballots. Our paper can be seen as an extension of this work to a probabilistic setting.

Another related work is [3], which gives methods using a sample of votes to predict the winner of an election probabilistically. That paper examines many different voting methods, not just IRV, and focuses on how large the sample must be to make robust predictions. Similarly, our work can also make predictions of elections where only a sample of votes are known: this is the main point of view in the examples of Section 2.1.

When an election result is "close," recounts are the typical method of ensuring the declared winner is the actual winner. In IRV elections, an election which

seems "close" in some particular round may not actually be a close election at all. Section 2.2 utilizes our algorithm to inform the literature of the differences between FPTP and IRV recounts. This work has real-world impacts for the task of determining if a recount is even necessary.

There is also some literature on the related subject of a "risk-limiting audit" (RLA) for IRV elections. The goal of an RLA is to confirm the results of an already-tabulated election to some predetermined level of certainty. A correct RLA procedure is one which provably will never confirm a false result, although it may, in some cases, fail to confirm a true result. The paper [8] presents a Bayesian method for IRV risk-limiting audits.

Although an RLA procedure's particular setting and goals differ from our work, both seek to determine information about the true IRV result using only partial information about the ballots. Typical RLA procedures use ballot sampling, while our method is more generally applied in any situation with some probabilistic ballot information.

In this paper, we present an algorithm for calculating the set of possible outcomes of an IRV election and assigning probabilities to each outcome.

The input to the algorithm is a set of discrete probability density functions describing anticipated vote totals for each possible candidate ranking. For example, in an election with three candidates, A, B, and C, these probability distributions might look like [1]:

| Votes | $f_A$ | $f_B$ | $f_C$ | $f_{AB}$ | $f_{AC}$ | $f_{BA}$ | $f_{BC}$ | $f_{CA}$ | $f_{CB}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.50 | 0.10 | 0.02 | 0.01 | 0.50 | . | . | 0.09 | 0.17 |
| 100 | 0.50 | 0.30 | 0.33 | 0.17 | 0.40 | . | 0.10 | 0.17 | 0.75 |
| 200 | . | 0.30 | 0.21 | 0.34 | 0.07 | 0.45 | 0.30 | 0.37 | 0.08 |
| 300 | . | 0.20 | 0.20 | 0.25 | 0.03 | 0.31 | 0.27 | 0.21 | . |
| 400 | . | 0.10 | 0.15 | 0.13 | . | 0.20 | 0.19 | 0.10 | . |
| 500 | . | . | 0.09 | 0.10 | . | 0.04 | 0.14 | 0.06 | . |

We note that our method makes no assumptions on the shape of the individual probability distributions. There seems to be some debate in the literature on which distribution functions will most accurately model voting results. The obvious candidate is the normal distribution, but some have suggested that this is unrealistic [9, 7, 5]. We take no position on the matter, and our method can be applied to any imagined probability distribution function, either standard mathematical distributions (e.g., the normal distribution) or arbitrary distributions constructed from empirical measurements. We acknowledge that creating these distributions can be difficult and rife with assumptions that can lead our algorithm to produce different results for various underlying distributions. Nevertheless, as mentioned previously, the algorithm we present in this paper can handle the most arbitrary distributions fed into it.

From the table above, our algorithm calculates probabilities for victory by each of A, B, and C. In this example, we compute winning probabilities as

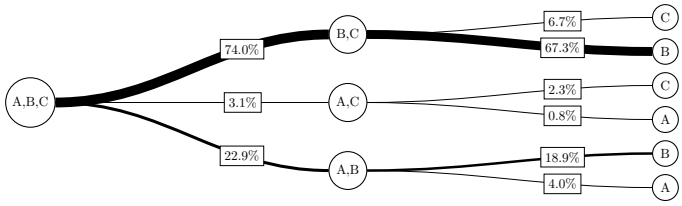
[1] For example, the third column above indicates that the ranking $BA$ will receive 200 votes with probability 45%, 300 votes with probability 31%, etc. Dots represent the probability of 0.

follows:

$$A : 4.8\% \qquad B : 86.1\% \qquad C : 8.9\%$$

Full details of the calculation are presented later in Example 9.

In fact, our algorithm gives the probabilities associated with every possible round in the IRV calculation. In this example, we can present the probabilities of the various rounds in a "weighted elimination tree" as follows:




Above, each node represents a round involving the labeled candidates, and the weighted edges represent the probability of each round following from the previous. For example, the three percentages on the left are the probabilities of each of A, B, and C being eliminated in the first round. The 67.3% at the upper right is the probability that B wins the election after A was eliminated first. This number, plus the 18.9% below, adds up to the total probability of a win by B. For readability, the thickness of each edge is drawn in proportion to its probability.

This tree provides a probabilistic analog of similar diagrams from [10], for example, their Figure 1 which calculates which eliminations are mathematically possible but does not indicate their probabilities.

## 2 Sample applications of our algorithm

?⟨alaskasection⟩? After adopting Ballot Measure 2 in 2020, Alaska was the first state to implement a top-four primary followed by an IRV general election. All qualifying candidates are on a single primary ballot where voters vote for their top choice only. The top four vote-getters in the primary then proceed to an IRV general election. Ballot Measure 2 passed with 50.55% vote in 2020 [6] and has survived a court challenge in the Alaska Supreme Court [1]. A 2024 ballot measure to repeal IRV elections failed, albeit with an extremely close margin. After a recount on 2024's Ballot Measure 2, 160,973 Alaskans voted to keep IRV, and 160,230 voted to repeal IRV [14].

We will present examples modeled on real data from the 2022 General Election in Alaska, which used IRV for all contests. The complete voting data for these elections is publicly available online from the Alaska Division of Elections.[2]

[2] https://bit.ly/Alaska2022GE

## 2.1 Winner prediction based on partial vote counts

?⟨unboundsec⟩?
?⟨house18exl⟩?

**Example 1. 2022 Alaska House District 18.**

As a fairly straightforward example, we first consider the 2022 election for Alaska House District 18. The ballot listed Democrat and eventual winner Cliff Groh, Democrat Lyn Franks, and Republican David Nelson. We denote these candidates by G, F, and N, respectively.

In the voting data published by the state of Alaska, each vote comes tagged with a geographical identifier called the "precinct portion," which tracks where the vote originated. The 2022 election data lists votes from about 500 different precinct portions.

We will apply our algorithm to a hypothetical election-night scenario in which only a portion of the ballots have been counted. Taking the complete set of around 2,100 votes cast in the House District 18 election, we choose 50% of these votes and tally them precisely.

In this particular instance, after counting 50% of the votes, our tally is as follows:

| ranking: | F | N | G | FN | FG | NF | NG | GF | GN |
|---|---|---|---|---|---|---|---|---|---|
| votes: | 55 | 301 | 96 | 20 | 147 | 38 | 52 | 245 | 42 |

(1) ?FGNvotes?

For the remaining 50% of the votes, we must make a probabilistic prediction of their contents. In a real-life application of our algorithm, this prediction could be modeled using pre-election polling data, ideally coupled with geographic and other information, which allows more precise predictions for the particular group of votes that are still uncounted.

As a simple proof-of-concept, without access to real polling data, we simply predict that the uncounted ballots will be tallied roughly in proportion to those that have already been counted. Specifically, for some ranking $\ell$, we set the probability distribution function $f_\ell$ equal to a discretized normal distribution with mean and standard deviation in proportion to the mean and standard deviation of the already-counted ballots.

In this case, our probability distributions are as follows:

| Votes | $f_F$ | $f_N$ | $f_G$ | $f_{FN}$ | $f_{FG}$ | $f_{NF}$ | $f_{NG}$ | $f_{GF}$ | $f_{GN}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | . | . | . | 1.00 | . | . | . | . | . |
| 75 | 0.39 | . | 0.21 | . | . | 0.49 | 0.40 | . | 0.45 |
| 150 | 0.61 | . | 0.33 | . | 0.14 | 0.51 | 0.60 | . | 0.55 |
| 225 | . | . | 0.29 | . | 0.20 | . | . | 0.08 | . |
| 300 | . | 0.07 | 0.14 | . | 0.23 | . | . | 0.11 | . |
| 375 | . | 0.09 | 0.04 | . | 0.20 | . | . | 0.13 | . |
| 450 | . | 0.10 | . | . | 0.13 | . | . | 0.14 | . |
| 525 | . | 0.11 | . | . | 0.07 | . | . | 0.14 | . |
| 600 | . | 0.11 | . | . | 0.03 | . | . | 0.12 | . |
| 675 | . | 0.11 | . | . | . | . | . | 0.10 | . |
| 750 | . | 0.10 | . | . | . | . | . | 0.07 | . |
| 825 | . | 0.09 | . | . | . | . | . | 0.05 | . |
| 900 | . | 0.07 | . | . | . | . | . | 0.03 | . |
| 975 | . | 0.05 | . | . | . | . | . | 0.02 | . |
| 1050 | . | 0.04 | . | . | . | . | . | 0.01 | . |
| 1125 | . | 0.03 | . | . | . | . | . | . | . |
| 1200 | . | 0.02 | . | . | . | . | . | . | . |
| 1275 | . | 0.01 | . | . | . | . | . | . | . |
| 1350 | . | 0.01 | . | . | . | . | . | . | . |

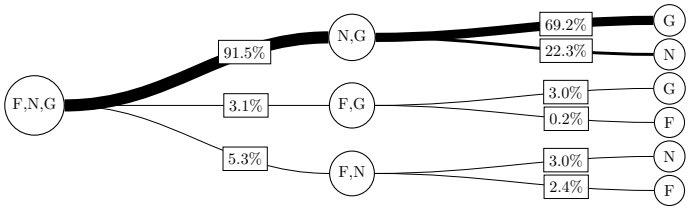
(2) `?house18pt?`

Above, for the sake of easily printing out the probabilities, we have grouped the vote counts into 19 rows, which we call the "vote buckets". As described in Appendix B, informal testing suggests a general rule-of-thumb of about 500 buckets for achieving a good balance between efficiency and accuracy.

The algorithm we will describe in Section 5 computes the probabilities of each candidate winning the election as follows:

$$F : 2.6\% \qquad N : 25.3\% \qquad G : 72.1\%$$

with the following weighted elimination tree:




Above, based on counting half of the votes, we see a very high probability that Franks will be eliminated first, followed by a likely win by Groh. This is indeed what happened in this election.

To get a feel for the algorithm's effectiveness, we can repeat the calculation above based on counting $x\%$ of the vote for various $x \in [0, 100]$. For a more realistic imitation of an election night count in progress, we will simulate a

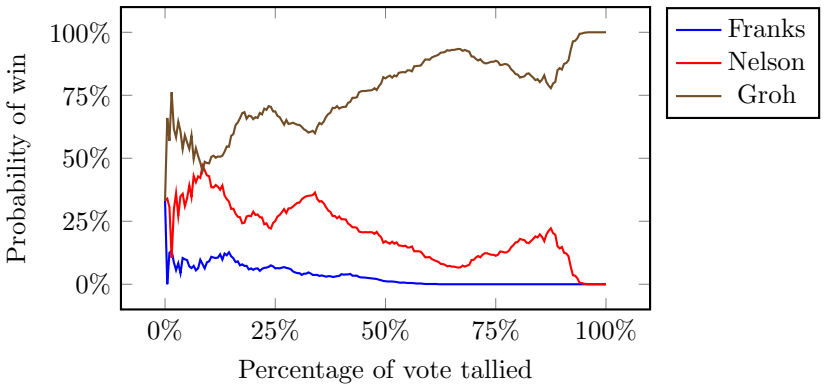



Figure 1: Win probabilities for Alaska House 18, as votes are tallied

partial count of votes by choosing a random ordering of the precinct portions and then counting votes in order according to their precinct portions. Thus, as in a real election, our votes will be counted in some order determined by geography.

The winning probabilities at each value of $x$ are shown in Figure 1.

A visualization like Figure 1 would be a convenient way to display live predictions of an in-progress election on election night. Early during the counting process (on the left side of the diagram), the predictions are less specific, and as more and more votes are counted, it becomes clearer who the eventual winner will be. Since there are precisely 3 candidates, we can also visualize the same data in the ternary plot of Figure 2.

**Example 2. 2022 US Senator from Alaska**

?⟨ussenateexl⟩?

The 2022 ballot for US Senator from Alaska featured Republican incumbent and eventual winner Lisa Murkowski, along with fellow Republican Kelly Tshibaka, as the main contenders. Democrat Pat Chesbro also appeared on the ballot, as did Buzz Kelley, a Republican who had suspended his campaign and officially endorsed Tshibaka two months before the election. Performing the same analysis as above gives Figure 3.

Early in the counting in Figure 3, we see some significant possibility of a win by Tshibaka. Still, as votes are counted, it becomes increasingly clear that Murkowski will win. The probabilities of wins by Kelly and Chesbro never rise above 0.1%.

The shape of the curves in Figures 1 and 3 heavily depends on the order in which the precinct portions are tabulated. Charts resulting from three other orderings of the precinct portions in the US Senate race are given in Figure 4.

The first chart in Figure 4 is similar to the one in Figure 3. The next two are instances in which the votes counted early indicated a strong showing by Tshibaka. This early lead by Tshibaka quickly evaporated as more votes were

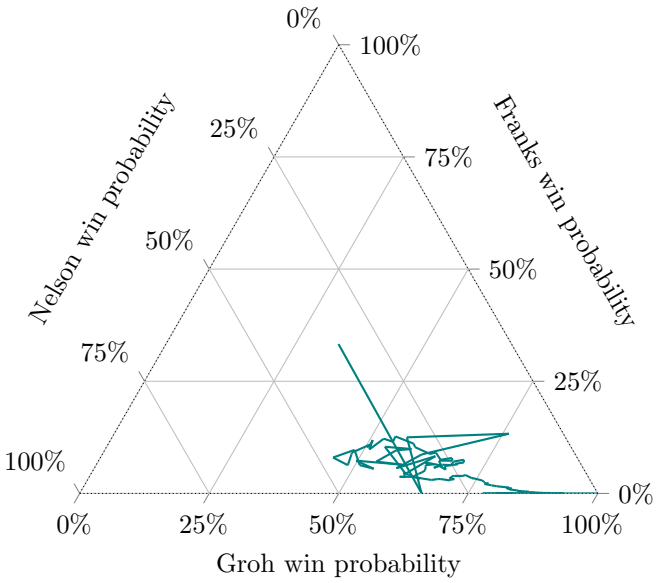



Figure 2: Ternary plot of data from Figure 1

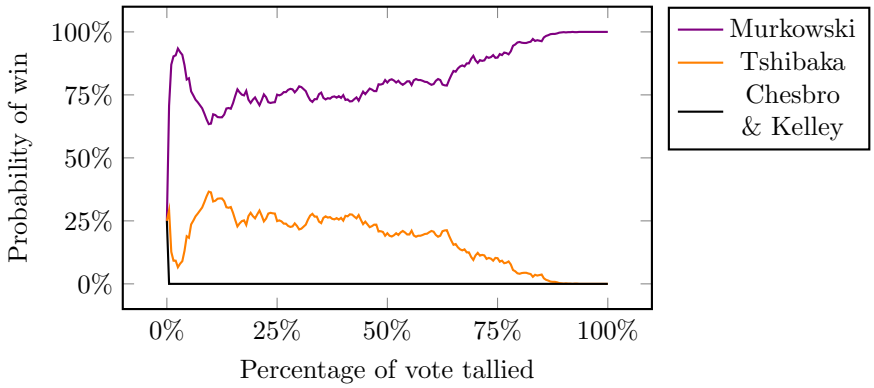



Figure 3: Win probabilities for Alaska US Senate race, as votes are tallied

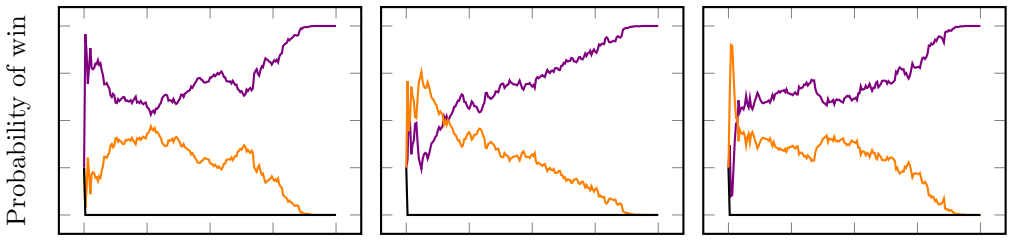



Figure 4: Alternative runs of Figure 3, using different randomized orderings of precinct portions

?⟨geographicfig⟩?

counted. Kelly and Chesbro never achieved any significant probability of winning in any of our tests.

## 2.2 Recount predictions

?⟨recountsection⟩? Our algorithm can also be applied after the election results are known fully when an election result seems close and a recount is considered. The very notion of a "close" election can be counterintuitive in the IRV setting. Typically, the *margin of victory* in any election is defined as the smallest number of votes that must be changed for the election outcome to become different [12, 4]. In an FPTP election, this margin of victory is simply the difference in vote totals between the first and second-place finishers.

It has been known for some time that determination of the margin of victory in an IRV election is not an easy problem. The natural algorithm presented in [12] is shown to have factorial complexity in the number of candidates. Some authors have produced algorithms that run faster but give only approximate results [17, 3]. For 3 or 4 candidates, however, the naïve algorithm performs well enough.

There has been some empirical study of typical recount results: we refer to the data gathered by FairVote, which collected results of all recounts in all US State elections from 2000 to 2023 [15]. This comprised 37 elections that were recounted, almost exclusively using first-past-the-post style procedures. The data in this case shows that each recount resulted in an average change in vote totals by 0.077%, with a standard deviation of 0.146%. Plotting the various individual results in a histogram shows a distribution that appears roughly normal. See Figure 5.

Thus, we will take as an operating assumption that a recount will generally change the vote totals (in percentage) according to a normal distribution with mean 0.077% and standard deviation 0.146%. This will allow us to estimate the probability of each candidate winning the election after the recount.

?⟨simplerecountex⟩?

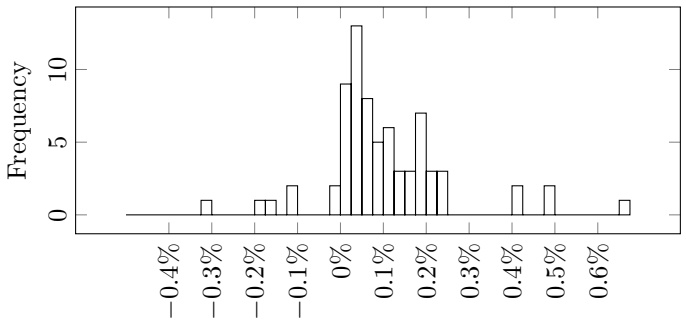



Figure 5: Histogram of historical vote shifts after recounts in US state elections, 2000–2023

?⟨recounthistogramfig⟩?

**Example 3.** Consider an election between three candidates, A, B, and C, in which we record a final vote tally as follows:

| ranking: | AB | AC | BA | BC | CA | CB | A | B | C |
|---|---|---|---|---|---|---|---|---|---|
| votes: | 501 | 300 | 400 | 400 | 200 | 600 | 500 | 400 | 500 |

Calculating the IRV result, we see first-round totals of:

$$A : 1301, \qquad B : 1200, \qquad C : 1300$$

so, B is eliminated, and the second-round totals are as follows:

$$A : 1701, \qquad C : 1700$$

and A is the winner. In this case, C will request a recount, and we would like to calculate the probability of each candidate winning after the recount. Intuitively, we expect A to remain the winner, with C having some nonzero chance of winning and B having no chance of winning after the recount.[3]

As described above, we assume that the vote results after the recount will be normally distributed. In this case, the probability distributions for votes after

[3]In this case, we have 2401 total voters, so we expect that a recount will result in a change in about $2401 \cdot 0.00077 = 1.85$ votes.

the recount are [4]:

| Votes | $f_A$ | $f_B$ | $f_C$ | $f_{AB}$ | $f_{AC}$ | $f_{BA}$ | $f_{BC}$ | $f_{CA}$ | $f_{CB}$ |
|---|---|---|---|---|---|---|---|---|---|
| 200 | . | . | . | . | . | . | . | 0.98 | . |
| 201 | . | . | . | . | . | . | . | 0.02 | . |
| 299 | . | . | . | . | 0.02 | . | . | . | . |
| 300 | . | . | . | . | 0.79 | . | . | . | . |
| 301 | . | . | . | . | 0.19 | . | . | . | . |
| 399 | . | 0.06 | . | . | . | 0.06 | 0.06 | . | . |
| 400 | . | 0.59 | . | . | . | 0.59 | 0.59 | . | . |
| 401 | . | 0.34 | . | . | . | 0.34 | 0.34 | . | . |
| 402 | . | 0.01 | . | . | . | 0.01 | 0.01 | . | . |
| 499 | 0.09 | . | 0.09 | . | . | . | . | . | . |
| 500 | 0.48 | . | 0.48 | 0.09 | . | . | . | . | . |
| 501 | 0.38 | . | 0.38 | 0.47 | . | . | . | . | . |
| 502 | 0.05 | . | 0.05 | 0.38 | . | . | . | . | . |
| 503 | . | . | . | 0.05 | . | . | . | . | . |
| 598 | . | . | . | . | . | . | . | . | 0.01 |
| 599 | . | . | . | . | . | . | . | . | 0.11 |
| 600 | . | . | . | . | . | . | . | . | 0.40 |
| 601 | . | . | . | . | . | . | . | . | 0.38 |
| 602 | . | . | . | . | . | . | . | . | 0.10 |
| 603 | . | . | . | . | . | . | . | . | 0.01 |

Applying our algorithm to the distributions above gives winning probabilities as follows:

$$A : 72.0\% \qquad B : 0\% \qquad C : 28\%$$

confirming our intuition that A and C each have a chance of winning after the recount, but B does not.

We can apply the same analysis to the two races in the 2022 Alaska election that were recounted.

**Example 4. 2022 Alaska State House District 15**

This was an election between Republicans Thomas McKay and David Eibeck, and Democrat Denny Wells, with around 7000 votes cast. We will denote these candidates respectively by M, E, and W. In the initial count for the election, Eibeck was eliminated in the first round by a wide margin, but McKay won the second round by only 7 votes. Applying the same analysis as above allows us to compute the likely results of a runoff:

$$M : 92.9\% \qquad E : 0\% \qquad W : 7.1\%$$

and we conclude that McKay is very likely to win in the runoff, with some small chance for Wells to win.

In fact, McKay did win the runoff, winning the second round by 9 votes.

[4] Many intermediate rows of zeros have been omitted from this table.

**Example 5. 2022 Alaska State Senate District E**

The 2022 race for Alaska's State Senate District E is notable because the results, as reported by the media, seemed to indicate a very close margin of loss for one candidate, who lost in the first round by only 14 votes out of about 16000 cast. The state conducted an official recount, but in fact, the election's results were not close.

This was an election between Republican Cathy Giessel, Republican Roger Holland, and Democrat Roselynn Cacy. We will denote these candidates respectively by G, H, and C. The typical public presentation of the results reported the following first-round totals (before the recount):

$$G : 5652 \qquad H : 5532 \qquad C : 5518$$

Here, we see Cacy behind Holland by only 14 votes, so Cacy was eliminated. Giessel went on to win the second round by a significant margin.

Despite the very slight margin between Cacy and Holland in the first round, a cursory analysis of the votes indicates that the election was not at all close. Even if Cacy had survived to the second round, she would have lost by significant margins to either Giessel or Holland, who receive far more transferred votes from one another since they are both Republicans.

Nevertheless, Cacy requested a recount. We can apply the method described above to calculate the expected results after the recount, and unsurprisingly, we obtain probabilities:

$$G : 100\% \qquad H : 0\% \qquad C : 0\%$$

# 3 The IRV algorithm

Here we review the details of how exactly the results of an IRV election are computed. We assume voters choose from a set of $N$ candidates $\mathcal{C} = \{A_1, \ldots, A_N\}$. Each vote consists of an ordered list $v_1 \ldots v_k$ for $0 \leq k \leq N$ where the $v_i$ are distinct elements of $\mathcal{C}$.

When a voter indicates $v_1 \ldots v_k$ on their ballot, this means that the candidate $v_1$ is this voter's top choice, the next candidate $v_2$ is their second choice in case $v_1$ is eliminated, and so on until $v_k$ is their last choice. If $k < N$, then this voter has left some candidates out of their vote entirely, indicating that they do not want to support those candidates under any circumstances.

In combinatorics, an ordered subset of a given set $\mathcal{C}$ is called a *subset permutation* of $\mathcal{C}$. We write $\Gamma(\mathcal{C})$ for the set of subset permutations of $\mathcal{C}$. The number of subset permutations of a set of $N$ elements is given by the formula:

$$\#\Gamma(\mathcal{C}) = N! \sum_{k=0}^{N} \frac{1}{k!} \tag{3}$$ ?subsetpermcount?

For various $N$, this is OEIS sequence A000522 [13]. Since $\sum_{k=0}^{\infty} \frac{1}{k!}$ equals the number $e$, the size of $\Gamma(\mathcal{C})$ grows asymptotically like $eN!$ as $N$ increases.

The election results are computed recursively in rounds. In a typical round, we count the total number of times that each remaining candidate appears as some voter's top choice. The candidate receiving the fewest of these top choice rankings is eliminated from the election.

If there is a tie among candidates receiving the least amount, then special rules are invoked to resolve the tie, which differs from place to place in real-world IRV use cases. Often, in theoretical contexts, the convention is that all candidates involved in the tie are simultaneously eliminated. In our analysis throughout the paper we consider the chance of an exact tie to be negligible, so we will assume throughout that no round ends in an exact tie.

For voters whose votes look like $v_1 \dots v_k$ with $k < N$, it is possible for all of $v_1, \dots, v_k$ to be eliminated before the end of the election. In that case, this vote is called "exhausted" and does not contribute to any totals in the remaining rounds.

In a real-world IRV election, the election may be "called" whenever one candidate is ranked as a top choice on a majority of the remaining (non-exhausted) ballots. In that case the majority candidate is mathematically guaranteed to win any future rounds, and so the ballot counting can stop before all of the rounds are computed.

For our mathematical discussion, it will be most convenient to run the rounds to their completion in all cases. Thus, in an election with $N$ candidates (assuming no ties in any round), we will always have $N$ rounds, where the final round consists of a single candidate who is declared the winner.

# 4 Discrete convolution

The fundamental computational tool used in our algorithm is the discrete convolution of discrete probability distributions. By *discrete probability distribution* we mean a function $f : B \to \mathbb{R}$ where $B$ is a finite set and $f(b) \in [0, 1]$ for all $b \in B$ and

$$\sum_{b \in B} f(b) = 1.$$

For us, these values $f(b)$ will always represent probabilities associated with some random variable $X$, and so we will sometimes write $f(b)$ as $P(X = b)$.

Given two discrete probability distributions $f, g : B \to \mathbb{R}$, the *discrete convolution* of $f$ and $g$ is defined as:

$$f * g(k) = \sum_{i \in B} f(i)g(k - i).$$

If $f$ and $g$ are probability distributions of two independent random variables $X$ and $Y$, then the above becomes:

$$\begin{aligned} f * g(k) &= \sum_{i \in B} P(X = i) \cdot P(Y = k - i) \\ &= \sum_{i \in B} P(X = i \text{ and } Y = k - i) = P(X + Y = k). \end{aligned}$$

This is the fundamental way in which we use the convolution: given two independent random variables, the probability distribution of their sum equals the discrete convolution.

We can also compute a convolution of more than two distributions:

$$f * g * h(k) = \sum_{i,j \in B} f(i)g(j)h(k-i-j) \tag{4}$$ ?badsum?

The sum above requires a twofold iteration, which may be undesirable computationally. Fortunately, the convolution satisfies an associative property:

$$f * g * h = (f * g) * h = f * (g * h),$$

so we can find $f * g * h$ by first finding $f * g$, and then convolving $f * g$ with $h$, which is typically far more efficient than using the double sum of (4).

In fact, the summations can be avoided altogether using the property:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$$

where $\mathcal{F}$ is the discrete Fourier transform, and the right side is the pointwise product. This formula can be used to compute discrete convolutions more efficiently using fast Fourier transform algorithms. [5]

# 5 The algorithm for computing winning probabilities

?⟨algorithmsection⟩? Our input data consists of probability distributions of random variables representing the total number of votes received for each possible ranking $R \in \Gamma(\mathcal{C})$. Throughout, we make the fundamental assumption that these random variables are independent. See Section 8.2 for a detailed discussion of the independence assumption.

Let $\mathcal{E}$ represent an election among a set of $N$ candidates $\mathcal{C} = \{A_1, \ldots, A_N\}$. We assume that we are given data which consists of a set of discrete probability distributions $\{f_R \mid R \in \Gamma(C)\}$, where $f_R$ is the probability distribution describing the probabilities of various vote totals for the ranking $R$.

We assume these probability distributions all have the same discrete domain $B = b_0\mathbb{N} = \{0, b_0, 2b_0, \ldots\}$ for some positive natural number $b_0$. This set $B$ is called the *set of buckets*, and the number $b_0$ is called the *bucket size*. The example probability distributions in the introduction use a bucket size of 100. In that example, for instance, we have $f_{AC}(200) = 0.07$, which should be interpreted to mean that there is a 7% chance that the number of votes for the ranking $AC$ is between 200 and 300. Using a smaller bucket size will allow for more specific modeling of the probability distributions, but this comes with increased computational cost since every convolution must sum over the set of buckets.

[5] The naïve summation requires $O(n)$ operations, while the fast Fourier transform is $O(\log n)$ where $n$ is the size of $B$.

The examples presented in Section 2 use various bucket sizes according to the number of votes cast in the various elections.

Our goal is to compute the vector of winning probabilities, which we denote $W(\mathcal{E}) \in [0,1]^N$, where coordinate $i$ of $W(\mathcal{E})$ is the probability that candidate $A_i$ wins the election. As stated above, we assume there will be no exact tie in any round.

We compute $W(\mathcal{E})$ by considering all possible sequences of eliminations and their respective probabilities. For some ordered list of candidates, $\ell = v_1 \dots v_k \in \Gamma(\mathcal{C})$, let $\mathcal{E}^\ell$ denote the election round obtained from $\mathcal{E}$ after eliminating the candidates $v_1, \dots, v_k$ in order (with $v_1$ eliminated first). In this context, we refer to $\ell \in \Gamma(\mathcal{C})$ as an *elimination order*.

We also use elimination order superscripts to unambiguously denote the random variables representing vote totals in each round. Given an elimination order $\ell \in \Gamma(\mathcal{C})$, rankings in the round $\mathcal{E}^\ell$ should not use candidates appearing in $\ell$ (since those candidates have been eliminated). Thus a typical ranking in round $\mathcal{E}^\ell$ is taken from the set $\Gamma(\mathcal{C} \setminus \ell)$, where $\mathcal{C} \setminus \ell$ is the set of all candidates from $\mathcal{C}$ which do not appear in $\ell$.

For some $\ell \in \Gamma(\mathcal{C})$ and some ranking $R \in \Gamma(\mathcal{C} \setminus \ell)$, let $R^\ell$ denote the random variable of the number of votes received for the ranking $R$ in the round obtained by eliminating candidates in order according to $\ell$.

For example, in an election with $\mathcal{C} = \{A, B, C\}$, the number of votes for the ranking $AB$ in the first round is represented simply by the random variable $AB$, while the number of votes for the ranking $AB$ in a round after eliminating $C$ is represented by $AB^C$.

For each such random variable $R^\ell$, we write $f_R^\ell$ for its probability distribution. Thus, for example, if the probability that the ranking $AB$ receives 300 votes in the first round is 25%, then this would be written as $f_{AB}(300) = .25$ or $P(AB = 300) = .25$.

Recall that we assume at the outset that the various random variables $R \in \Gamma(\mathcal{C})$ are independent. When $\ell \in \Gamma(\mathcal{C})$, we will want the various random variables $R^\ell \in \Gamma(\mathcal{C} \setminus \ell)$ also to be independent. We give a proof of the following lemma in the appendix:

?⟨indeplem⟩? **Lemma 6.** Let the random variables $\{R \mid R \in \Gamma(\mathcal{C})\}$ be independent, and let $\ell \in \Gamma(\mathcal{C})$. Then the random variables $\{R^\ell \mid R \in \Gamma(\mathcal{C} \setminus \ell)\}$ are independent.

Our computation of $W(\mathcal{E})$ is done recursively by computing $W(\mathcal{E}^\ell)$ for all possible elimination orderings $\ell$. The recursion terminates when $\#\ell = N - 1$, in which case $\ell$ includes all elements of $\mathcal{C}$ except for a single candidate $A_i$. This represents the final round in which only candidate $A_i$ remains, so the vector $W(\mathcal{E}^\ell)$ will equal 1 in coordinate $i$, and 0 in all other coordinates.

Now we describe how to compute $W(\mathcal{E}^\ell)$ when $\#\ell < N - 1$. Let $E_{A_i}^\ell$ denote the probability that $A_i$ will be eliminated in election round $\mathcal{E}^\ell$. Then, the desired vector of winning probabilities is computed as a weighted average:

$$W(\mathcal{E}^\ell) = \sum_{A_i \in \mathcal{C} \setminus \ell} E_{A_i}^\ell W(\mathcal{E}^{\ell A_i}), \tag{5}$$ ?recursion?

where $\ell A_i$ is the elimination ordering consisting of the candidates of $\ell$ followed by $A_i$. Since the superscript on $\mathcal{E}$ is longer on the right side above, eventually, all superscripts will reach length $N-1$, and the algorithm will terminate. Our true goal is to compute $W(\mathcal{E})$, which is done recursively as:

$$W(\mathcal{E}) = \sum_{A_i \in \mathcal{C}} E_{A_i} W(\mathcal{E}^{A_i}). \tag{6}$$ ?winningformula?

To accomplish the computation in (5), there are two ingredients: Computation of the elimination probabilities $E^{\ell}_{A_i}$, and computation of the probability distributions for $\mathcal{E}^{\ell A_i}$, given the probability distributions for $\mathcal{E}^{\ell}$. In both cases, we fundamentally use the independence of the random variables representing the vote counts. We are assuming from the outset that the vote totals in round one are independent, and the required independence will continue to hold in subsequent rounds by Lemma 6.

Now, we describe how to compute the various ingredients of (5).

## 5.1 Computing distributions for $\mathcal{E}^{\ell A}$, given those for $\mathcal{E}^{\ell}$

The easiest part is deriving the required random variables and probability distributions for $\mathcal{E}^{\ell A}$, given the random variables and probability distributions for $\mathcal{E}^{\ell}$.

We must describe how a given ranking $R^{\ell A}$ in $\mathcal{E}^{\ell A}$ relates to the various rankings in $\mathcal{E}^{\ell}$. Since $A$ is eliminated when we move from round $\mathcal{E}^{\ell}$ to $\mathcal{E}^{\ell A}$, the IRV procedure stipulates that the votes for some ranking $R^{\ell A}$ will equal the sum of all rankings from round $\mathcal{E}^{\ell}$ which become $R$ when $A$ is eliminated. Symbolically, this is written as follows:

$$R^{\ell A} = \sum_{\substack{X \in \Gamma(\mathcal{C} \setminus \ell) \\ \rho_A(X) = R}} X^{\ell},$$

where $\rho_A(X)$ denotes removal of $A$ from $X$. Because a sum of random variables corresponds to a convolution of probability distributions, we have:

$$f^{\ell A}_R = \mathop{\Large *}_{\substack{X \in \Gamma(\mathcal{C} \setminus \ell) \\ \rho_A(X) = R}} f^{\ell}_X. \tag{7}$$ ?dconvolution?

The formula above describes how to compute the probability distributions for round $\mathcal{E}^{\ell A}$, given those for round $\mathcal{E}^{\ell}$.

## 5.2 Computing elimination probabilities $E^{\ell}_{A_i}$

Now we discuss the more involved computation of the elimination probabilities $E^{\ell}_{A_i}$ for $A_i \in \mathcal{C} \setminus \ell$, which appear in (5). For each candidate $A_i \notin \ell$, let $T^{\ell}_{A_i}$ be the random variable giving the total number of votes for which $A_i$ appears in the

top position, as tallied in round $\mathcal{E}^\ell$. This $T^\ell_{A_i}$ is equal to the sum of the random variables of rankings $R^\ell$ in which $R$ is a ranking beginning with $A_i$. That is,

$$T^\ell_{A_i} = \sum_{\substack{R\in\Gamma(\mathcal{C}\setminus\ell)\\ a(R)=A_i}} R^\ell, \tag{8}$$ ?Tsum?

where $a(R) \in \mathcal{C}$ denotes the first-listed candidate of the ranking $R$.

Let $\tau^\ell_{A_i}$ be the probability distribution function of $T^\ell_{A_i}$. Again, since a sum of random variables corresponds to a convolution of probability distributions, we have:

$$\tau^\ell_{A_i} = \mathop{\Large *}_{\substack{R\in\Gamma(\mathcal{C}\setminus\ell)\\ a(R)=A_i}} f^\ell_R. \tag{9}$$ ?tconvolution?

Let $\kappa^\ell_{A_i}$ be the cumulative version of $\tau^\ell_{A_i}$, that is:

$$\kappa^\ell_{A_i}(k) = \sum_{\substack{b\in B\\ b\le k}} \tau^\ell_{A_i}(b).$$

Thus, $\kappa^\ell_{A_i}(k)$ is the probability that there are at most $k$ votes which rank $A_i$ in the top position. Accordingly, $1-\kappa^\ell_{A_i}(k)$ is the probability that there are fewer than $k$ votes which rank $A_i$ in the top position.

In a round with no ties, some candidate $A_i$ will be eliminated if there are $j$ votes placing $A_i$ in the top position while there are more than $j$ votes placing each of the other candidates in the top position. This probability is:

$$\hat{E}^\ell_{\{A_i\}} = \sum_k P(T^\ell_{A_i} = k \text{ and } T^\ell_{A_j} > k \text{ for all } j \neq i).$$

The above is unwieldy as written, but our independence assumption allows it to be computed easily. We provide proof of the following formula in the appendix.

?⟨elimproblem⟩? **Lemma 7.** With the notation above, we have:

$$\hat{E}^\ell_{\{A_i\}} = \sum_{k\in B}\left(\tau^\ell_{A_i}(k)\cdot\prod_{j\neq i}(1-\kappa^\ell_{A_j}(k))\right). \tag{10}$$ ?elimprobeq?

The quantity $\hat{E}^\ell_{\{A_i\}}$ is slightly less than the desired elimination probability $E^\ell_{A_i}$ appearing in (5) because it represents the probability that $A_i$'s top-ranked vote bucket is less than all others. But if the vote counts are close together, a candidate may be eliminated even if its top-ranked vote bucket equals the bucket of some other candidate.

Recall that we consider the likelihood of actual on-the-nose ties to be negligible. But we do need to consider cases in which candidates' totals land in the same bucket. We refer to this as a "bucket-tie". In the context of our probabilistic model, this represents not an actual tie but a situation where the margin is "too close to call" in terms of our discrete probability distributions.

To handle the case where two or more candidates are bucket-tied for the least number of rankings in the top position, let $S \subseteq \{1, \ldots, N\}$ be the (unordered) set indexing the bucket-tied candidates so that $A_i$ is among those in the bucket-tie for least number of top rankings if and only if $i \in S$. Then let:

$$\hat{E}_S^\ell = \sum_{k \in B} \left( \prod_{i \in S} \tau_{A_i}^\ell(k) \cdot \prod_{i \notin S} (1 - \kappa_{A_i}^\ell(k)) \right). \tag{11}$$ ?tieelimprobeq?

The above directly generalizes (10), and represents the probability that all candidates indexed in the set $S$ are bucket-tied for the least amount of top rankings in this round.

Since we cannot predict the eventual resolution of a bucket-tie, we will perform each of the possible elimination outcomes and weigh them equally in the probability $E_A$. For example, if our set of candidates is $\mathcal{C} = \{A_1, A_2, A_3\}$, then the total probability for elimination of $A$ in the first round is:

$$E_{A_1} = \hat{E}_{\{A_1\}} + \frac{1}{2}\hat{E}_{\{A_1,A_2\}} + \frac{1}{2}\hat{E}_{\{A_1,A_3\}} + \frac{1}{3}\hat{E}_{\{A_1,A_2,A_3\}},$$

where $\hat{E}_{\{A_1\}}$ is the probability that $A_1$ alone is eliminated unambiguously, $\hat{E}_{\{A_1,A_2\}}$ is the probability that there is a bucket-tie between $A_1$ and $A_2$, etc. Each of the terms above can be computed by (11)

In general, the proper formula for the elimination probability $E_{A_i}^\ell$ is as follows:

$$E_{A_i}^\ell = \sum_{S \subseteq \mathcal{C} \setminus \ell} \frac{1}{\#S} \hat{E}_S^\ell \tag{12}$$ ?realelimprobeq?

We have now described methods for computing each ingredient of the main formula (6), which completes the description of the algorithm.

# 6 A detailed example calculation

Before presenting an example, we give a result that can be used to simplify a given election.

It is not hard to see that a vote in which a voter ranks all $N$ choices is equivalent to the same vote if the last choice is blank.

?⟨undervotelem⟩? **Lemma 8.** In an election with $N$ candidates, a vote $V = v_1 \ldots v_N$ is equivalent to the vote $V' = v_1 \ldots v_{N-1}$.

*Proof.* By "equivalent," we mean that substituting the vote $V'$ for $V$ will not change the first-place vote totals in any round so the round-by-round results of the election will be unchanged.

The votes $V$ and $V'$ are the same in each of their first $N-1$ entries. This means that the only way for $V$ and $V'$ to be tabulated differently in some rounds is if all candidates $v_1, \ldots, v_{N-1}$ have already been eliminated. But this can only occur in round $N$, which is the final round in which the winner has already been determined, so exchanging $V$ for $V'$ does not affect the result. □

The result above means that, for a given election between $N$ candidates, we can immediately combine any votes of the form $v_1 \dots v_N$ with those of the form $v_1 \dots v_{N-1}$. In this way, we may assume that all voters have ranked at most $N-1$ candidates on their ballot, and none have ranked all $N$ candidates. This decreases the number of possible voter rankings by $N!$, which is a significant reduction. The number of voter rankings listing fewer than $N$ candidates is given by $N! \sum_{k=0}^{N-1} \frac{1}{k!}$, which is OEIS sequence A002627 [13].

Lemma 8 was used implicitly throughout all the examples in Section 2. For example in (1) we have combined the actual vote totals for FNG and FN into a single column labeled FN.

?⟨sampleexample⟩? **Example 9.** We will give the details of the full calculation for the example outlined in Section 1, which is a hypothetical IRV election between three candidates, A, B, and C.

By Lemma 8, we may assume that no voter ranks all 3 candidates on their ballot in the first round. Thus, the possible rankings recorded on votes in this election are AB, AC, BA, BC, CA, CB, A, B, C, and $\emptyset$, where $\emptyset$ represents an exhausted vote or a vote for none of the listed candidates.

We will assume that the vote totals in the first round obey the following probability distributions:

| Votes | $f_A$ | $f_B$ | $f_C$ | $f_{AB}$ | $f_{AC}$ | $f_{BA}$ | $f_{BC}$ | $f_{CA}$ | $f_{CB}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.50 | 0.10 | 0.02 | 0.01 | 0.50 | . | . | 0.09 | 0.17 |
| 100 | 0.50 | 0.30 | 0.33 | 0.17 | 0.40 | . | 0.10 | 0.17 | 0.75 |
| 200 | . | 0.30 | 0.21 | 0.34 | 0.07 | 0.45 | 0.30 | 0.37 | 0.08 |
| 300 | . | 0.20 | 0.20 | 0.25 | 0.03 | 0.31 | 0.27 | 0.21 | . |
| 400 | . | 0.10 | 0.15 | 0.13 | . | 0.20 | 0.19 | 0.10 | . |
| 500 | . | . | 0.09 | 0.10 | . | 0.04 | 0.14 | 0.06 | . |

(13) ?simpleextable?

(Without Lemma 8, we would require 6 more columns of probability distributions describing $f_{ABC}, f_{BAC}$, etc.)

First, we determine the elimination probabilities for each candidate. As a preliminary step, we use (9) to compute the probability distributions $\tau_A, \tau_B$, and $\tau_C$, which represent probabilities for each candidate to receive a certain number of votes in the top position. By (9) we have:

$$\tau_A = f_{AB} * f_{AC} * f_A$$
$$\tau_B = f_{BA} * f_{BC} * f_B$$
$$\tau_C = f_{CA} * f_{CB} * f_C$$

Computing the appropriate convolutions of the columns of (13) produces the following values for these distributions, and we also compute their cumulative distributions:

| Votes | $\tau_A$ | $\tau_B$ | $\tau_C$ | $\kappa_A$ | $\kappa_B$ | $\kappa_C$ |
|---|---|---|---|---|---|---|
| 0 | 0.0025 | . | 0.0003 | 0.0025 | 0 | 0.0003 |
| 100 | 0.047 | . | 0.007 | 0.0495 | 0 | 0.0073 |
| 200 | 0.1639 | . | 0.039 | 0.2134 | 0 | 0.0463 |
| 300 | 0.2559 | 0.0045 | 0.095 | 0.4693 | 0.0045 | 0.1413 |
| 400 | 0.2336 | 0.0301 | 0.175 | 0.7029 | 0.0346 | 0.3163 |
| 500 | 0.1618 | 0.0867 | 0.1934 | 0.8647 | 0.1213 | 0.5097 |
| 600 | 0.0932 | 0.1525 | 0.1813 | 0.9579 | 0.2738 | 0.691 |
| 700 | 0.0337 | 0.1968 | 0.1466 | 0.9916 | 0.4706 | 0.8376 |
| 800 | 0.0069 | 0.199 | 0.0931 | 0.9985 | 0.6696 | 0.9307 |
| 900 | 0.0015 | 0.1577 | 0.0453 | 1 | 0.8273 | 0.976 |
| 1000 | . | 0.0998 | 0.0181 | 1 | 0.9271 | 0.9941 |
| 1100 | . | 0.0496 | 0.0055 | 1 | 0.9767 | 0.9996 |
| 1200 | . | 0.018 | 0.0004 | 1 | 0.9947 | 1 |
| 1300 | . | 0.0047 | . | 1 | 0.9994 | 1 |
| 1400 | . | 0.0006 | . | 1 | 1 | 1 |

Then we compute the quantities of (11):

$$\begin{aligned}
\hat{E}_{\{A\}} &= \sum_k \tau_A(k)(1-\kappa_B(k))(1-\kappa_C(k)) = 0.672 \\
\hat{E}_{\{B\}} &= \sum_k (1-\kappa_A(k))\tau_B(k)(1-\kappa_C(k)) = 0.016 \\
\hat{E}_{\{C\}} &= \sum_k (1-\kappa_A(k))(1-\kappa_B(k))\tau_C(k) = 0.167 \\
\hat{E}_{\{A,B\}} &= \sum_k \tau_A(k)\tau_B(k)(1-\kappa_C(k)) = 0.018 \\
\hat{E}_{\{A,C\}} &= \sum_k \tau_A(k)(1-\kappa_B(k))\tau_C(k) = 0.112 \\
\hat{E}_{\{B,C\}} &= \sum_k (1-\kappa_A(k))\tau_B(k)\tau_C(k) = 0.005 \\
\hat{E}_{\{A,B,C\}} &= \sum_k \tau_A(k)\tau_B(k)\tau_C(k) = 0.007
\end{aligned}$$

and we find the elimination probabilities using (12):

$$\begin{aligned}
\hat{E}_A &= \hat{E}_{\{A\}} + \frac{1}{2}\hat{E}_{\{A,B\}} + \frac{1}{2}\hat{E}_{\{A,C\}} + \frac{1}{3}\hat{E}_{\{A,B,C\}} = 0.740 \\
\hat{E}_B &= \hat{E}_{\{B\}} + \frac{1}{2}\hat{E}_{\{A,B\}} + \frac{1}{2}\hat{E}_{\{B,C\}} + \frac{1}{3}\hat{E}_{\{A,B,C\}} = 0.031 \\
\hat{E}_C &= \hat{E}_{\{C\}} + \frac{1}{2}\hat{E}_{\{A,C\}} + \frac{1}{2}\hat{E}_{\{B,C\}} + \frac{1}{3}\hat{E}_{\{A,B,C\}} = 0.229
\end{aligned}$$

The above elimination probabilities will be used in (6). We must also recursively compute $W(\mathcal{E}^A), W(\mathcal{E}^B)$, and $W(\mathcal{E}^C)$. We will present the details in

full for $W(\mathcal{E}^A)$. We want to produce a table similar to (13), this time representing the probabilities after elimination of A. These are obtained using (7). We have:

$$\begin{aligned} f^A_{BC} &= f_{BC} \\ f^A_{CB} &= f_{CB} \\ f^A_B &= f_{AB} * f_{BA} * f_B \\ f^A_C &= f_{AC} * f_{CA} * f_C \\ f^A_\emptyset &= f_A * f_\emptyset \end{aligned}$$

Computing these convolutions gives the desired table for the round $\mathcal{E}^A$:

| Votes | $f^A_\emptyset$ | $f^A_B$ | $f^A_C$ | $f^A_{BC}$ | $f^A_{CB}$ |
|---|---|---|---|---|---|
| 0 | 0.50 | . | . | . | 0.17 |
| 100 | 0.50 | . | 0.02 | 0.10 | 0.75 |
| 200 | . | . | 0.05 | 0.30 | 0.08 |
| 300 | . | 0.01 | 0.13 | 0.27 | . |
| 400 | . | 0.05 | 0.18 | 0.19 | . |
| 500 | . | 0.11 | 0.19 | 0.14 | . |
| 600 | . | 0.18 | 0.17 | . | . |
| 700 | . | 0.20 | 0.13 | . | . |
| 800 | . | 0.19 | 0.08 | . | . |
| 900 | . | 0.13 | 0.04 | . | . |
| 1000 | . | 0.08 | 0.02 | . | . |
| 1100 | . | 0.04 | . | . | . |
| 1200 | . | 0.01 | . | . | . |

Note that in this round, there is some expectation for a nonzero number of exhausted votes, indicated in the column labeled $f^A_\emptyset$. These votes arise from any first-round votes for the ranking $A$, which has now been eliminated.

Now we can compute the distributions $\tau^A_B$ and $\tau^A_C$ using (9):

$$\begin{aligned} \tau^A_B &= f^A_{BC} * f^A_B \\ \tau^A_C &= f^A_{CB} * f^A_C \end{aligned}$$

We perform these convolutions, construct the cumulative distributions $\kappa^A_B$ and $\kappa^A_C$, and then compute elimination probabilities using (11) and (12):

$$\begin{aligned} E^A_B &= 0.090 \\ E^A_C &= 0.909 \end{aligned}$$

Now since $\mathcal{E}^{AB}$ and $\mathcal{E}^{AC}$ have only one candidate remaining, we have $W(\mathcal{E}^{AB}) = (0, 0, 1)$ and $W(\mathcal{E}^{AC}) = (0, 1, 0)$. Then by (5) we have:

$$\begin{aligned} W(\mathcal{E}^A) &= E^A_B W(\mathcal{E}^{AB}) + E^A_C W(\mathcal{E}^{AC}) \\ &= 0.090(0, 0, 1) + 0.909(0, 1, 0) = (0, 0.909, 0.090). \end{aligned}$$

Repeating the computations above for $\mathcal{E}^B$ and $\mathcal{E}^C$ in this example gives:

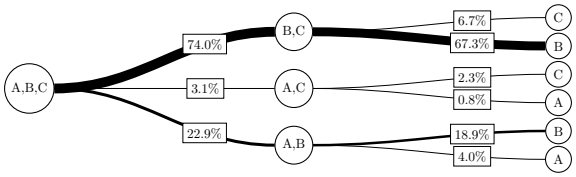
$$W(\mathcal{E}^B) = E_A^B W(\mathcal{E}^{BA}) + E_C^B W(\mathcal{E}^{BC}) = (0.745, 0, 0.254)$$
$$W(\mathcal{E}^C) = E_A^C W(\mathcal{E}^{CA}) + E_B^C W(\mathcal{E}^{CB}) = (0.823, 0.176, 0)$$

and combining all the above in (6) gives our final result of:

$$\begin{aligned} W(\mathcal{E}) &= E_A W(\mathcal{E}^A) + E_B W(\mathcal{E}^B) + E_C W(\mathcal{E}^C) \\ &= 0.740(0, 0.909, 0.090) + 0.031(0.745, 0, 0.254) + 0.229(0.823, 0.176, 0) \\ &= (0.048, 0.861, 0.089) \end{aligned}$$

So the chances of winning for A, B, and C, respectively, are 4.8%, 86.1%, and 8.9%.

Included in all of the intermediate calculations above are all the figures necessary to produce the weighted elimination tree:




From the diagram, we see that B is favored to win the election with a winning probability of 86.2%, and further the most likely order of eliminations is A first, followed by C.

# 7 Accuracy of the predictions

It is difficult to measure the accuracy of real-world election predictions. For example, if we predict that candidate A has a 75% chance of winning some election, the public will generally view that prediction as "correct" if A does indeed win and will view the prediction as "incorrect" if A loses. This is, of course, not an appropriate way to gauge the accuracy of the prediction. The appropriate way to measure this probability is to make many individual predictions and then ask: among all instances where a win probability of 75% was predicted, how many of these were an actual win?

In this spirit, we propose the following test for the accuracy of our method: Given a particular election from the Alaska 2022 dataset, we consider many different orderings of precinct portions, calculate various winning probabilities as in Figure 4, and then aggregate all of the resulting individual predictions, rounding to integer values for probabilities. Since the number of different orderings of precinct portions is very large, this allows for a very large sample of applications of our algorithm.

In aggregating all prediction results, we consider for each percentage $x$: among all instances where we predicted a win probability of $x$%, how many

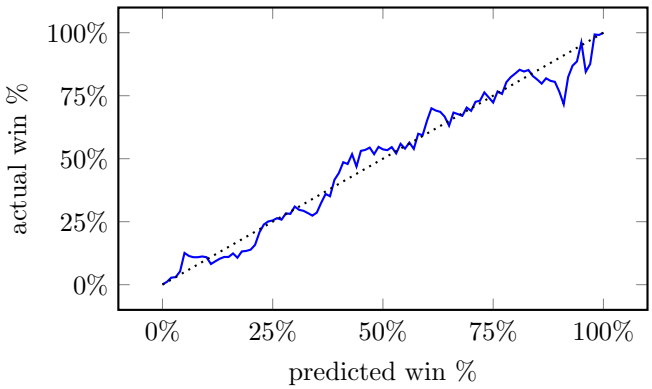



Figure 6: Predicted win probability vs actual win percentage across many applications of the algorithm in the AK House 18 race. Dotted line is $f(x) = x$, which would represent a "perfect" set of predictions.

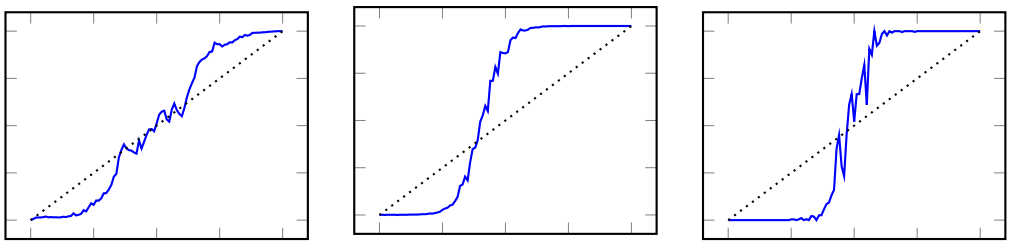
?⟨robustnessfig⟩?

represented an actual win? Results of these tests appear in Figure 6, which plots various predicted win probabilities vs the actual proportion of wins. The dotted line in the figure is $f(x) = x$, representing a "perfect" prediction algorithm in which a prediction of probability $x$ corresponds to an actual win proportion $x$. We call this *the ideal line*.

Figure 6 was produced using 200,000 individual applications of the algorithm using data from AK House 18. This consisted of producing 2,000 different random orderings of the precinct portions, each time calculating a set of predictions analogous to Figure 1. In this race, we see a good matching of algorithm predictions with the ideal line.

In Figure 7, we give similar charts for three other races from the 2022 Alaska data set. For these races, the fit to the ideal line is not as good. The general theme seems to be that the algorithm overestimates win percentages for eventual losers and underestimates win percentages for eventual winners. It is difficult to say why these races show worse predictions than in AK House 18.

We note that measuring the accuracy in this way reflects the accuracy of the entire scheme described for producing live election predictions as in Figure 1. Thus, Figures 6 and 7 judge the accuracy of our prediction algorithm together with the specific strategy of predicting future votes as normal distributions based on votes already received, with certain chosen means and standard deviations. It may be that the observed deviations from the ideal line and differences in accuracy from race to race reflect shortcomings of the live prediction strategy rather than our algorithm itself. See also our discussion below of our assumption of independence, the impacts of which may differ significantly from race to race.



AK House 35 US Rep. AK Senate J

Figure 7: Predicted win probability vs actual win probability for some other races.

?⟨robustnessmorefig⟩?

# 8 Limitations and future work

In this section, we present some limitations of the present work and suggest some avenues for future work.

## 8.1 Computational complexity

The input data to our algorithm consists of one discrete probability distribution for each possible voter ranking. By (3), the number of these voter rankings grows asymptotically like $N!$ where $N$ is the number of candidates. Thus, the complexity of our algorithm is at least factorial in $N$, so we do not expect it to be practical when $N$ is large.

Our algorithm performs well when the number of candidates is 3 or 4. For an election with 3 candidates and domains using around 500 vote buckets, computing the weighted elimination tree takes under 0.1 seconds on a typical consumer laptop. For 4 candidates, it takes about 0.6 seconds. For 5 candidates, it takes 19 seconds, and for 6 candidates, about 12 minutes.

## 8.2 The assumption of independence

?⟨dependencesection⟩?

Our algorithm fundamentally relies on the assumption that the probability distributions $f_R$ for various $R \in \Gamma(\mathcal{C})$ are independent. This allows convolutions to be used in (7) and (9). Without the assumption of independence, these convolutions would need to be replaced by more complicated formulas to compute joint probabilities.

The extent to which this assumption is reasonable will depend on the specific application. For the examples presented in Section 2, we believe that some dependence does exist, although it is hard to judge how significant this may be. Therefore our algorithm for deriving winning probabilities should be regarded as a sort of opening salvo: we are modeling the votes in the most mathematically sim-

ple way available. Further work can improve on this basic model by introducing correlations between these distributions, which are application-specific.

For the case of predicting votes based on partial vote tallies as presented in Section 2.1, we suggest that correlations will exist having two main types.

*Political correlations:* It is natural to expect correlations between vote tallies for politically similar candidates. Thus for example, if candidates A and B are politically similar and opposed to C, then we expect the tallies for the rankings A and B to be positively correlated with each other and each to be negatively correlated with the ranking C. Accurately modeling these correlations will require detailed analysis of the voter perception of the various candidates, presumably based on polling or other metrics.

*Mathematical correlations:* We also expect purely formal correlations between various rankings that are mathematically similar. For example, in an election with candidates A,B,C,D,E, even without knowing any political information about these candidates, we can expect a correlation of the ranking ABCD with the ranking ABCE, because these two rankings are formally similar. We expect that these correlations could be modeled using one of the many "string metrics" that exist in the literature for quantifying the similarity between two strings.[6]

In the case of predicting recounts as in Section 2.2, the independence assumption seems primarily justified. For instance, in Example 3, the shift in votes for the ranking AB after the recount should be independent of the shift in votes for the ranking BA after the recount. There may, however, be a slight positive correlation in the absolute value of the vote shifts among all rankings when recounting ballots from the same polling place or jurisdiction. (For example, if a certain polling place reports many votes changed for ranking AB after the recount, then we may take this as a sign that they are generally error-prone in counting, and so we may expect higher rates of vote changes across all rankings.)

Though we will not attempt to present a model that takes these possible dependencies into account, we can measure to what extent any dependencies will affect the accuracy of the model. We will use the probability distributions presented in Example 1 (Alaska House District 18, with 50% of votes tallied), in which our algorithm calculated the following win probabilities:

$$F : 2.6\% \qquad N : 25.3\% \qquad G : 72.1\%$$

Intuitively, this means that if the election were run many times, with votes distributed independently according to the table (2), we expect that F will win these elections with a frequency of 2.6%, N with a frequency of 25.3%, and G with a frequency of 72.1%. Simulating 100,000 elections by choosing votes in each column independently according to (2) produces win frequencies of:

$$F : 2.1\% \qquad N : 25.8\% \qquad G : 72.1\% \tag{14}$$ ?indeq?

This provides a nice check on the correctness of our algorithm's implementation.

We can adapt the strategy above to measure how well the algorithm performs even in the context of dependence between our random variables. We will

[6] For example, the $\tau$-distance of [11].

again simulate 100,000 elections using (2), but this time, we will model strong dependencies in which we imagine F and G to be politically identical and both perfectly opposed to N. (In the real election, Frank and Groh are both Democrats, while N is a Republican.)

Specifically, each of the 100,000 random elections will be generated as follows: Choose randomly a number $x \in [0, 1]$, and choose the number of votes for F to be the bucket size at which the cumulative probability in column $f_F$ of (2) exceeds $x$. Since we imagine F and G to be politically identical, we use the same value $x$ to determine the number of votes for G using the column $f_G$. Since we are modeling N as perfectly opposed to F, we use the value $1 - x$ to determine the number of votes for N using the column $f_N$. For the rankings involving two candidates, we choose another random number $y \in [0, 1]$ (chosen independently from $x$), which we use to produce the vote counts for the rankings FN and GN, and we use the value $1 - y$ to choose vote counts for the rankings NF and NG. Finally we choose another random number $z \in [0, 1]$ which is used to determine the vote counts for the rankings FG and GF. Thus, we are modeling a set of votes in which the votes for F, FG, and FN are each independent, but votes for all other rankings are perfectly determined by these three.

Modeling dependencies in this way, we see the following proportions of wins:

$$F : 0\%, \qquad N : 30.6\%, \qquad G : 69.4\% \tag{15}$$ ?depeq?

Comparing (14) and (15) indicates some effect of the independence assumption in this race. It is hard to say which of (14) or (15) are more plausible results in this case.

## 8.3 Future Work

The most obvious avenue for future work will be to improve the model by allowing for correlations between the distributions. Hopefully, this can be done in a way that improves the accuracy of the predictions without significantly compromising computational efficiency.

It is also natural to consider if our basic model can be adapted to the context of multi-winner contests decided by Single Transferable Vote (STV). We expect that the general idea of computing weighted averages recursively can be applied to STV as well.

If ours or similar algorithms are used for real-time prediction of in-progress elections, there is a need to present the results visually in a thoughtful and informative way. We have shown several types of visualizations of our algorithm's output. A diagram such as Figure 1 provides a nice way to track predictions over time as more and more votes are counted. A single application of the algorithm can produce a weighted elimination tree, which includes more information but represents a prediction of only one snapshot in time. In elections between 3 candidates, a ternary chart as in Figure 2 may be useful. It would be nice to see more options for similar parametric diagrams when the number of candidates exceeds 3.

Finally we point out that many questions remain concerning the interpretations of the algorithm's results. For example, media organizations often want to "call the election" when they have determined that one candidate's probability of winning is overwhelming. It is unclear in the case of our algorithm when such a "call" is justified.

For example, in the data presented in Figure 3, we report a 92.7% chance of a win by Murkowski after counting only 2.5% of the vote. Obviously, it is not appropriate to call the election at that point. In the same example, Murkowski's probability of winning reaches 100% after 94.5% of the votes are counted. But probably, a call for Murkowski is justified earlier than 94.5% in this case. It would be interesting if more analysis could add flares to the lines in Figure 3 so that the election can be called with high confidence as soon as the flares are non-overlapping.

Political analysts must wrestle with these and other related questions if these probabilistic methods are to be used effectively.

# A Proofs of technical lemmas

We begin with a standard fact about disjoint sums of independent random variables. The proof is elementary, but we include it for the sake of completeness.

?⟨sumlemma⟩? **Lemma 10.** Let $\{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ be a set of independent random variables. Then the two sums $X_1 + \cdots + X_n$ and $Y_1 + \cdots + Y_m$ are independent.

*Proof.* The proof is a simple calculation. Using independence of $\{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$, we have:

$$\begin{aligned} P\left(\sum_i X_i = k \text{ and } \sum_j Y_j = \ell\right) &= \sum_{\substack{k_1+\ldots k_n=k \\ \ell_1+\cdots+\ell_m=\ell}} P(X_i = k_i \text{ and } Y_j = \ell_j) \\ &= \sum_{\substack{k_1+\ldots k_n=k \\ \ell_1+\cdots+\ell_m=\ell}} P(X_i = k_i)P(Y_j = \ell_j) \\ &= \sum_{k_1+\cdots+k_n=k} P(X_i = k_i) \sum_{\ell_1+\cdots+\ell_m=\ell} P(Y_j = \ell_j) \\ &= P\left(\sum_i X_i = k\right) P\left(\sum_j Y_j = \ell\right), \end{aligned}$$

and we have shown that $\sum_i X_i$ and $\sum_j Y_j$ are independent. □

Now we are ready to prove Lemmas 6 and 7.

*Proof of Lemma 6.* Let the random variables $\{R \mid R \in \Gamma(\mathcal{C})\}$ be independent, and let $\ell \in \mathcal{C}$, and we will show that the random variables $\{R^\ell \mid R \in \Gamma(\mathcal{C})\}$ are indepedent.

By induction on the length of $\ell$, it suffices to prove the lemma in the case when $\ell$ consists of a single candidate $\ell = A \in \mathcal{C}$. Take two subset permutations $S^A, R^A \in \Gamma(\mathcal{C} \setminus A)$ with $S^A \neq R^A$, and we must show that $S^A$ and $R^A$ are independent.

As a random variable, $S^A$ is the sum of all subset-permutations $X \in \Gamma(\mathcal{C})$ for which $\rho_A(X) = S$, where $\rho_A$ denotes elimination of $A$ from $X$. Let $\{X_1, \ldots, X_s\} \in \Gamma(\mathcal{C})$ be this set of all such subset-permutations, and we have $S^A = \sum_i X_i$. Similarly let $Y_1, \ldots, Y_t$ be the set of subset-permutations with $\rho_A(Y) = R$, and we have $R^A = \sum_i Y_i$.

Now we claim that we must have $X_i \neq Y_j$ for all $i, j$: if we had $X_i = Y_j$, then $S = \rho_A(X_i) = \rho_A(Y_j) = R$, and so $S^A = R^A$, but we have assumed that $S^A \neq R^A$. Since $X_i \neq Y_j$ for all $i, j$, and these random variables are all taken from the independent set $\{R \mid R \in \Gamma(\mathcal{C})\}$, the various $X_i$ and $Y_j$ are all independent.

Then we may apply lemma 10 to the set $\{X_1, \ldots, X_s, Y_1, \ldots, Y_t\}$, and we conclude that $S^A$ and $R^A$ are independent as desired. □

*Proof of Lemma 7.* Using notations from Section 5, we must show that

$$\hat{E}^\ell_{\{A_i\}} = \sum_k \left( \tau^\ell_{A_i}(k) \cdot \prod_{j \neq i} (1 - \kappa^\ell_{A_j}(k)) \right).$$

First, we show that the random variables $T^\ell_{A_i}$ for various $A_i$ are independent. We must show that:

$$P(T^\ell_{A_i} = m \text{ and } T^\ell_{A_j} = n) = P(T^\ell_{A_i} = m) P(T^\ell_{A_j} = n)$$

when $i \neq j$. Using (8), let $T^\ell_{A_i} = X^\ell_1 + \ldots X^\ell_s$ for $X \in \Gamma(\mathcal{C})$ where $a(X_k) = A_i$ for each $k$, and similarly let $T^\ell_{A_j} = Y^\ell_1 + \ldots Y^\ell_t$ for $Y \in \Gamma(\mathcal{C})$ where $a(Y_k) = A_j$ for each $k$. (Recall that $a(R)$ denotes the first-listed candidate in the ranking $R$.) Since $A_i \neq A_j$, each $X_k$ is different from each $Y_k$, and thus the set $\{X_1, \ldots, X_s, Y_1, \ldots, Y_t\}$ is independent by Lemma 6. This means that $T^\ell_{A_i}$ and $T^\ell_{A_j}$ are independent by Lemma 10.

Now for fixed $k$, we have:

$$\begin{aligned} P(T^\ell_{A_i} = k \text{ and } T^\ell_{A_j} > k \text{ for all } j \neq i) &= \sum_{u > k} P(T^\ell_{A_i} = k \text{ and } T^\ell_{A_j} = u \text{ for all } j \neq i) \\ &= \sum_{u > k} P(T^\ell_{A_i} = k) \prod_{j \neq i} P(T^\ell_{A_j} = u) = \tau^\ell_{A_i}(k) \prod_{j \neq i} \sum_{u > k} \tau^\ell_{A_j}(u) \\ &= \tau^\ell_{A_i}(k) \prod_{j \neq i} \left( 1 - \sum_{u \leq k} \tau^\ell_{A_j}(u) \right) = \tau^\ell_{A_i}(k) \prod_{j \neq i} \left( 1 - \kappa^\ell_{A_j}(k) \right) \end{aligned}$$

Summing the above over $k$ gives:

$$\begin{aligned}\hat{E}^{\ell}_{\{A_i\}} &= \sum_k P(T^{\ell}_{A_i} = k \text{ and } T^{\ell}_{A_j} > k \text{ for all } j \neq i) \\ &= \sum_k \tau^{\ell}_{A_i}(k) \prod_{j \neq i} \left(1 - \kappa^{\ell}_{A_j}(k)\right)\end{aligned}$$

as desired.

□

# B Choosing the number of vote buckets

?⟨bucketsection⟩? Our algorithm uses discrete probability distributions in which vote counts are grouped in "buckets" (these are the rows of (13), which uses 6 buckets). We expect for our computed probabilities to be more accurate when we increase the number of buckets, but the complexity of our algorithm increases linearly in the number of buckets, so there will be a tradeoff between accuracy and efficiency.

We have not attempted a detailed investigation of the "best" number of vote buckets to use. Still, it is easy to see graphically the effect of choosing different numbers of buckets in Figure 8. Here, we have compared plots similar to Figure 1 using various numbers of buckets, this time using data from the 2022 Alaska election for US Representative (candidates Begich, Palin, Peltola).

In each chart of Figure 8 labeled "using $b$ buckets," this means that we begin with initial (first-round) discrete probability distributions using $b$ vote buckets. As seen in Example 9, the number of buckets used in subsequent rounds will increase whenever convolutions are taken. In that example, we have 6 buckets in the first round, but convolutions result in 13 buckets in the round after A is eliminated.

Based on the plots from Figure 8, we have taken as a general rule that using 500 vote buckets is sufficient to get a good sense of the probabilities. Running similar tests in other races shows similar results: increasing the number of buckets beyond 500 seems to offer only minimal improvements.


# Funding

The authors did not utilize any external funding for this project.

# Acknowledgements

The authors would like to thank Dr. Laura Dumitrescu for helpful conversations.

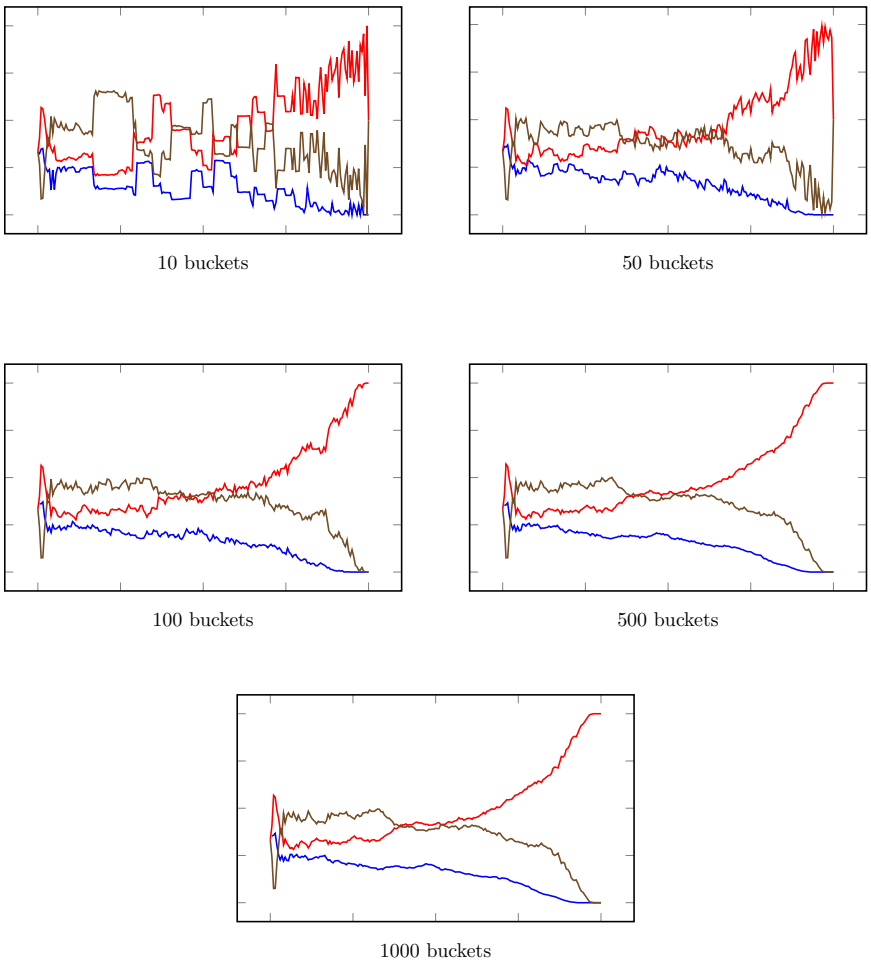



Figure 8: Comparison of results from AK US Rep. using different numbers of vote buckets

?⟨bucketfig⟩?

## Data Availability Statement

The replication code for this article is available at https://github.com/cstaecker/irvprob

## Competing Interests

The authors declare none.

## References


kohlhaas_2022 [1] Kohlhaas v. Alaska. Supreme Court of Alaska, 2022. URL: https://caselaw.findlaw.com/court/ak-supreme-court/1972157.html.

rcv_history [2] Ballotpedia. History of ranked-choice voting (rcv) ballot measures, n.d. Accessed: December 22, 2024. URL: https://ballotpedia.org/History_of_ranked-choice_voting_(RCV)_ballot_measures.

bhattacharyya_2021 [3] Arnab Bhattacharyya and Palash Dey. Predicting winner and estimating margin of victory in elections using sampling. *Artificial Intelligence*, 296:103476, 2021. doi:10.1016/j.artint.2021.103476.

blom_irv_margins [4] Michelle Blom, Vanessa Teague, Peter J. Stuckey, and Ron Tidhar. Efficient computation of exact irv margins. *Frontiers in Artificial Intelligence and Applications*, 285:480–488, 2016. doi:10.3233/978-1-61499-672-9-480.

bovens_measuring_2011 [5] Luc Bovens and Claus Biesbart. Measuring Voting Power for Dependent Voters through Causal Models. *Synthese (Dordrecht)*, 179(Suppl 1):35 – 56, April 2011. doi:https://doi.org/10.1007/s11229-010-9854-8.

Brooks_2020 [6] James Brooks. Ballot measure 2 eeks out narrow win with nearly all votes in. *Alaska Journal of Commerce*, 44(47):6, Nov 22 2020. Copyright - Copyright MCC Magazines, LLC d/b/a Alaska Magazine Nov 22, 2020; Last updated - 2023-11-07; SubjectsTermNotLitGenreText - United States–US; Alaska. URL: https://www-proquest-com.libdb.fairfield.edu:8443/trade-journals/ballot-measure-2-eeks-out-narrow-win-with-nearly/docview/2467349692/se-2.

dow_multinomial_2004 [7] Jay Dow and James Endersby. Multinomial probit and multinomial logit: a comparison of choice models for voting research. *Electoral Studies*, 23(1):107–122, March 2004. doi:10.1016/S0261-3794(03)00040-4.

everest [8] Floyd Everest, Michelle Blom, Philip B. Stark, Peter J. Stuckey, Vanessa Teague, and Damjan Vukcevic. Ballot-polling audits of instant-runoff voting elections with a dirichlet-tree model. In Sokratis et al. Katsikas, editor, *Computer Security. ESORICS 2022 International Workshops*, pages 525–540, Cham, 2023. Springer International Publishing.

gelman_mathematics_2002 [9] Andrew Gelman, Jonathan N. Katz, and Francis Tuerlinckx. The Mathematics and Statistics of Voting Power. *Statistical Science*, 17(4):420 – 435, 2002.

a_jelvani_identifying_2022 [10] Alborz Jelvani and Amelie Marian. Identifying possible winners in ranked choice voting elections with outstanding ballots. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 10(1):114–123, Oct. 2022. URL: https://ojs.aaai.org/index.php/HCOMP/article/view/21992, doi:10.1609/hcomp.v10i1.21992.

kendall_1938 [11] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. URL: http://www.jstor.org/stable/2332226.

magrino_2011 [12] Thomas R. Magrino, Ronald L. Rivest, Emily Shen, and David Wagner. Computing the margin of victory in IRV elections. In *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'11, page 4, USA, 2011. USENIX Association.

oeis [13] OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2024. Published electronically at http://oeis.org.

alaska_ballot_measure_recount [14] Office of the Lieutenant Governor of Alaska. Lieutenant governor nancy dahlstrom announces completion of ballot measure 2 recount, 2024. Accessed: December 22, 2024. URL: https://ltgov.alaska.gov/newsroom/2024/12/lieutenant-governor-nancy-dahlstrom-announces-completion-of-ballot-measure-2-recount/.

fairvote_2023 [15] Deb Otis. An analysis of statewide election recounts, 2000–2023. 2023. Accessed: 2024-04-29. URL: https://fairvote.org/report/election-recounts-2023/.

otis_ranked_nodate [16] Deb Otis. Ranked choice voting in 2023: A year in review. pages 1–18, 2023. URL: https://fairvote.org/report/2023-a-year-in-review/.

sarwate_2013 [17] Anand D. Sarwate, Stephen Checkoway, and Hovav Shacham. Risk-limiting audits and the margin of victory in nonplurality elections. *Statistics, Politics, and Policy*, 4(1):29–64, 2013. doi:doi:10.1515/spp-2012-0003.