

Ahead of the Count: An Algorithm for Probabilistic Prediction of Instant Runoff (IRV) Elections

Nicholas Kapoor and P. Christopher Staecker *

May 14, 2024

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.

1 Introduction

The American electorate is antsy. The build-up of an election season is palpable when polls close on election night; the need to know the winner is insatiable. 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 state laws around counting the vote, tallies are produced at varying rates, in some cases taking several days.

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

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 US.

Initiatives to replace FPTP elections with IRV elections have had 27 consecutive wins on municipal ballots [9], and this movement is quickly becoming one of the largest voting reforms in the United States. As IRV becomes a larger part of American politics, new 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.

The 2019 New York City Mayoral and City Council Democratic primaries were one of the first major uses of IRV in the US in multiple races on the same Election Day. Jelvani and Marian [3] 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 Bhattacharyya and Dey [1], 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 the question of how large the sample must be in order 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.

In this paper, we present an algorithm which calculates the set of possible outcomes of an IRV election and assigns probabilities to each one.

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	.

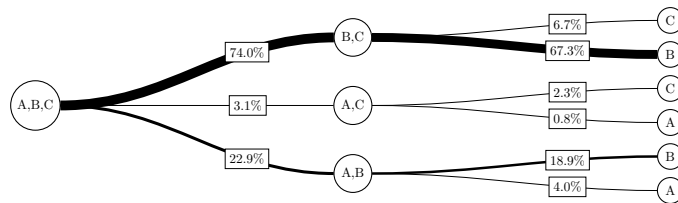
We note that our method makes no assumptions on the shape of the individual probability distributions. There seems to be some debate in the literature on the question of which distribution functions will most accurately model voting results. The obvious candidate is the normal distribution, but some have suggested that this is unrealistic (Gelman, Katz, and Tuerlinckx 2002; Dow and Enderbsy 2004; Bovens and Biesbart 2011). 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.

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 follows:

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

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.

¹For 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.

This tree provides a probabilistic analog of similar diagrams from Jelvani and Marian [3], for example, their Figure 1 which calculates which eliminations are mathematically possible but does not indicate their probabilities.

2 Sample applications of our algorithm

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 [2] and has survived a court challenge in the Alaska Supreme Court [5]. However, a ballot measure to repeal IRV elections in Alaska may make the 2024 ballot [10].

We will present some examples modeled on real data from the 2022 General Election in the state of 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

2.1 Winner prediction based on partial vote counts

Example 1. 2022 Alaska House District 18.

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

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 full 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)

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 which are still uncounted.

²<https://bit.ly/Alaska2022GE>

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 ℓ , we set the probability distribution function f_ℓ equal to a discretized normal distribution with mean and standard deviation in proportion to the mean and standard deviation of the already-counted ballots.

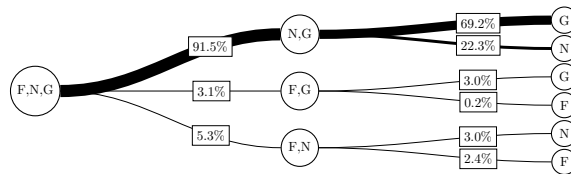
In this case, our probability distributions are as follows:

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

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

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

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

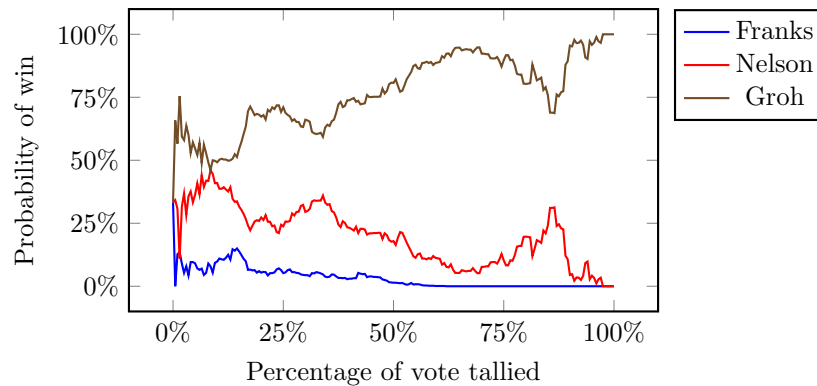


Figure 1: Win probabilities for Alaska House 18, as votes are tallied

realistic imitation of an election night count in progress, we will simulate a 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 exactly 3 candidates, we can also visualize the same data in the ternary plot of Figure 2.

Example 2. 2022 US Senator from Alaska

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, but 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 is heavily dependent 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 that were counted early indicated a strong showing by Tshibaka. This early lead by Tshibaka quickly evaporated as more votes

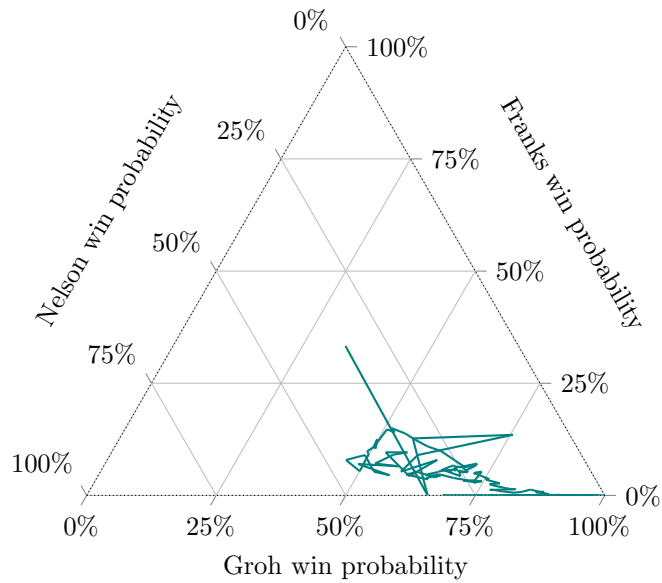


Figure 2: Ternary plot of data from Figure 1

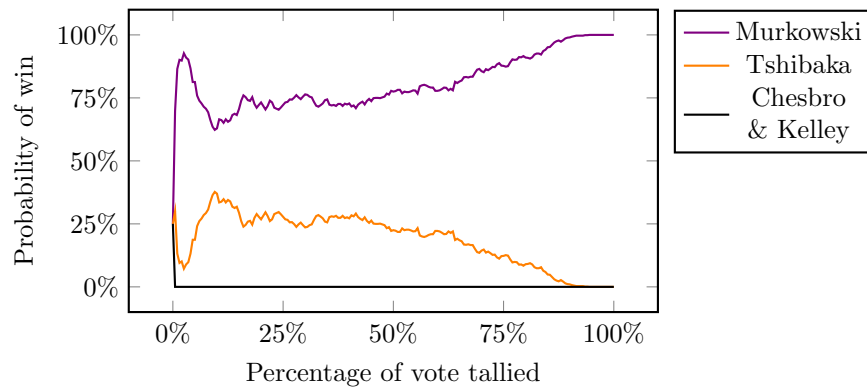


Figure 3: Win probabilities for Alaska US Senate race, as votes are tallied

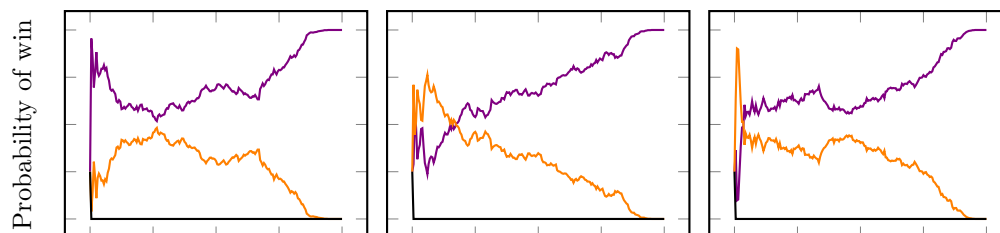


Figure 4: Alternative runs of Figure 3, using different randomized orderings of precinct portions

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

2.2 Recount predictions

Our algorithm can also be applied after the election results are known fully, in the case when an election 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 in order for the election outcome to become different [6]. In a 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 Magrino et al. [6] is shown to have factorial complexity in the number of candidates. Some authors have produced algorithms that run faster but give only approximate results [11, 1]. 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 [8]. This is comprised of 37 different 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.

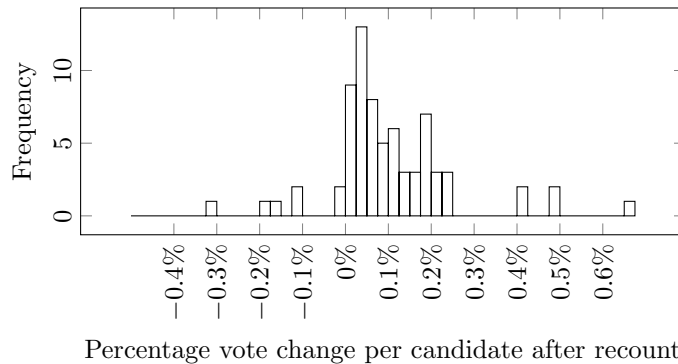


Figure 5: Histogram of historical vote shifts after recounts in US state elections, 2000–2023

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:

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

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

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

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

³In this case, we have 2401 total voters, and so our expectation is 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:

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

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:

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

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.

⁴Many 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 Republicans Cathy Giessel and 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):

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

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:

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

3 The IRV algorithm

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

When a voter indicates $v_1 \dots v_k$ 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:

$$\#\Gamma(\mathcal{C}) = N! \sum_{k=0}^N \frac{1}{k!} \tag{2}$$

For various N , this is OEIS sequence A000522 [7]. 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 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 differ 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 $f : B \rightarrow \mathbb{R}$ 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 $f, g : B \rightarrow \mathbb{R}$, 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:

$$f * g * h(k) = \sum_{i,j \in B} f(i)g(j)h(k-i-j) \quad (3)$$

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 (3).

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. Using fast Fourier transform algorithms, this formula can be used to compute discrete convolutions more efficiently.⁵

5 The algorithm for computing winning probabilities

Our input data consists of probability distributions of random variables representing the total number of votes that will be received for each possible ranking $R \in \Gamma(\mathcal{C})$. Throughout, we make the fundamental assumption that these random variables are independent. See Section 7.2 for a detailed discussion of the independence assumption.

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

We assume that these probability distributions all have the same discrete domain $B = b_0\mathbb{N} = \{0, b_0, 2b_0, \dots\}$ 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*. For example the example probability distributions presented 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

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

the set of buckets. The examples presented in Section 2 use various different 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 that 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}^ℓ 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}^ℓ should not use candidates appearing in ℓ (since those candidates have been eliminated). Thus a typical ranking in round \mathcal{E}^ℓ 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 ℓ .

For some $\ell \in \Gamma(\mathcal{C})$ and some ranking $R \in \Gamma(\mathcal{C} \setminus \ell)$, let R^ℓ 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 ℓ .

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^ℓ , we write f_R^ℓ 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 $R^\ell \in \Gamma(\mathcal{C} \setminus \ell)$ to also be independent. We give a proof of the following lemma in the appendix:

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 ℓ . The recursion terminates when $\#\ell = N - 1$, in which case ℓ 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, and 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}^ℓ . Then, the desired vector of winning probabilities is computed as a weighted average:

$$W(\mathcal{E}^\ell) = \sum_{A_i \in \mathcal{C} \setminus \ell} E_{A_i}^\ell W(\mathcal{E}^{\ell A_i}), \quad (4)$$

where ℓA_i is the elimination ordering consisting of the candidates of ℓ 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:

$$W(\mathcal{E}) = \sum_{A_i \in \mathcal{C}} E_{A_i} W(\mathcal{E}^{A_i}). \quad (5)$$

To accomplish the computation in (4), there are two ingredients: Computation of the elimination probabilities $E_{A_i}^\ell$, and computation of the probability distributions for $\mathcal{E}^{\ell A_i}$, given the probability distributions for \mathcal{E}^ℓ . 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 (4).

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

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}^ℓ .

We need to describe how a given ranking $R^{\ell A}$ in $\mathcal{E}^{\ell A}$ relates to the various rankings in \mathcal{E}^ℓ . Since A is eliminated when we move from round \mathcal{E}^ℓ 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}^ℓ 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:

$$f_R^{\ell A} = \bigstar_{\substack{X \in \Gamma(\mathcal{C} \setminus \ell) \\ \rho_A(X) = R}} f_X^\ell. \quad (6)$$

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

5.2 Computing elimination probabilities $E_{A_i}^\ell$

Now we discuss the more involved computation of the elimination probabilities $E_{A_i}^\ell$ for $A_i \in \mathcal{C} \setminus \ell$ which appear in (4). For each candidate $A_i \notin \ell$, let $T_{A_i}^\ell$ 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}^ℓ . This $T_{A_i}^\ell$ is equal to the sum of the random variables of rankings R^ℓ in which R is a ranking beginning with A_i . That is,

$$T_{A_i}^\ell = \sum_{\substack{R \in \Gamma(\mathcal{C} \setminus \ell) \\ a(R) = A_i}} R^\ell, \quad (7)$$

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

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

$$\tau_{A_i}^\ell = \bigstar_{\substack{R \in \Gamma(\mathcal{C} \setminus \ell) \\ a(R) = A_i}} f_R^\ell. \quad (8)$$

Let $\kappa_{A_i}^\ell$ be the cumulative version of $\tau_{A_i}^\ell$, that is:

$$\kappa_{A_i}^\ell(k) = \sum_{\substack{b \in B \\ b \leq k}} \tau_{A_i}^\ell(b).$$

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

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

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

Lemma 7. With the notation above, we have:

$$\hat{E}_{\{A_i\}}^\ell = \sum_{k \in B} \left(\tau_{A_i}^\ell(k) \cdot \prod_{j \neq i} (1 - \kappa_{A_j}^\ell(k)) \right). \quad (9)$$

The quantity $\hat{E}_{\{A_i\}}^\ell$ is slightly less than the desired elimination probability $E_{A_i}^\ell$ appearing in (4), 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 a true 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 $S \subseteq \{1, \dots, N\}$ 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:

$$\hat{E}_S^\ell = \sum_{k \in B} \left(\prod_{i \in S} \tau_{A_i}^\ell(k) \cdot \prod_{i \notin S} (1 - \kappa_{A_i}^\ell(k)) \right). \quad (10)$$

The above directly generalizes (9), 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 are unable to 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 recall that $\hat{E}_{\{A_1\}}$ 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 (10)

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

$$E_{A_i}^\ell = \sum_{S \subseteq \mathcal{C} \setminus \ell} \frac{1}{\#S} \hat{E}_S^\ell \quad (11)$$

We have now described methods for computing each ingredient of the main formula (5), 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 left blank.

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

Proof. By “equivalent” we mean that substituting the vote V' for V will not change the first-place vote totals in any round, so that 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 of the candidates v_1, \dots, 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, and so exchanging V for V' has no effect on the result. \square

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 [7].

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.

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	(12)
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	.	

(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 (8) to compute the probability distributions τ_A, τ_B , and τ_C , which represent probabilities for each candidate to receive a certain number of votes in the top position. By (8) we have:

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

Computing the appropriate convolutions of the columns of (12) produces the following values for these distributions, and we also compute their cumulative distributions:

Votes	τ_A	τ_B	τ_C	κ_A	κ_B	κ_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 (10):

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

and we find the elimination probabilities using (11):

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

The above elimination probabilities will be used in (5). 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 (12), this time representing the probabilities after elimination of A. These are obtained using (6). We have:

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

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

Votes	f_{\emptyset}^A	f_B^A	f_C^A	f_{BC}^A	f_{CB}^A
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_{\emptyset}^A . These votes arise from any first round votes for the ranking A, who has now been eliminated.

Now we can compute the distributions τ_B^A and τ_C^A using (8):

$$\begin{aligned}\tau_B^A &= f_{BC}^A * f_B^A \\ \tau_C^A &= f_{CB}^A * f_C^A\end{aligned}$$

We perform these convolutions, construct the cumulative distributions κ_B^A and κ_C^A , and then compute elimination probabilities using (10) and (11):

$$\begin{aligned}E_B^A &= 0.090 \\ E_C^A &= 0.909\end{aligned}$$

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 (4) we have:

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

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

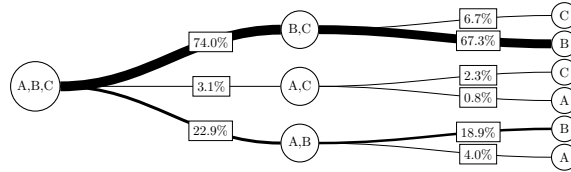
$$\begin{aligned} W(\mathcal{E}^B) &= E_A^B W(\mathcal{E}^{BA}) + E_C^B W(\mathcal{E}^{BC}) = (0.745, 0, 0.254) \\ W(\mathcal{E}^C) &= E_A^C W(\mathcal{E}^{CA}) + E_B^C W(\mathcal{E}^{CB}) = (0.823, 0.176, 0) \end{aligned}$$

and combining all the above in (5) 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, 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 that the most likely order of eliminations is A first, followed by C.

7 Limitations and future work

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

7.1 Computational complexity

The input data to our algorithm consists of one discrete probability distribution for each possible voter ranking. By (2), 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 , and 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 1500 vote buckets, computing the weighted elimination tree takes under 0.1 seconds on a typical consumer laptop. For 4 candidates it takes about 0.5 seconds. For 5 candidates it takes 14 seconds, and for 6 candidates, about 5 minutes.

7.2 The assumption of independence

Our algorithm fundamentally relies on the assumption that the probability distributions f_R for various $R \in \Gamma(\mathcal{C})$ are independent. This allows for convolutions to be used in (6) and (8). Without the assumption of independence, these convolutions would need to be replaced by some 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 in fact 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 simple 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 candidates who are politically similar. Thus for example if candidates A and B are politically similar, and opposed to C, then we expect for 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 which 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” which 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 mostly justified: for instance in Example 3 the shift in votes for the ranking AB after the recount should be independent from 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.)

7.3 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

⁶For example, Kendall’s (1938) τ -distance.

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 is more than 3.

Finally we point out that many questions still 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 will need to wrestle with these and other related questions if these probabilistic methods are to be used effectively.

Appendix: 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.

Lemma 10. Let $\{X_1, \dots, X_n, Y_1, \dots, Y_m\}$ be a set of independent random variables. Then the two sums $X_1 + \dots + X_n$ and $Y_1 + \dots + Y_m$ are independent.

Proof. The proof is a simple calculation. Using independence of $\{X_1, \dots, X_n, Y_1, \dots, Y_m\}$,

we have:

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

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

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 independent.

By induction on the length of ℓ , it suffices to prove the lemma in the case when ℓ 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 ρ_A denotes elimination of A from X . Let $\{X_1, \dots, X_s\} \in \Gamma(\mathcal{C})$ be this set of all such subset-permutations, and we have $S^A = \sum_i X_i$. Similarly let Y_1, \dots, 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 $\{X_1, \dots, X_s, Y_1, \dots, Y_t\}$, and we conclude that S^A and R^A are independent as desired. \square

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

$$\hat{E}_{\{A_i\}}^\ell = \sum_k \left(\tau_{A_i}^\ell(k) \cdot \prod_{j \neq i} (1 - \kappa_{A_j}^\ell(k)) \right).$$

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

$$P(T_{A_i}^\ell = m \text{ and } T_{A_j}^\ell = n) = P(T_{A_i}^\ell = m) P(T_{A_j}^\ell = n)$$

when $i \neq j$. Using (7), let $T_{A_i}^\ell = X_1^\ell + \dots X_s^\ell$ for $X \in \Gamma(\mathcal{C})$ where $a(X_k) = A_i$ for each k , and similarly let $T_{A_j}^\ell = Y_1^\ell + \dots Y_t^\ell$ 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 $\{X_1, \dots, X_s, Y_1, \dots, Y_t\}$ is independent by Lemma 6. This means that $T_{A_i}^\ell$ and $T_{A_j}^\ell$ are independent by Lemma 10.

Now for fixed k , we have:

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

Summing the above over k gives:

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

as desired. □

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.

Data Availability Statement

Replication code for this article is available at <https://github.com/cstaecker/irvprob>

Competing Interests

The authors declare none.

References

- [1] Arnab Bhattacharyya and Palash Dey. “Predicting winner and estimating margin of victory in elections using sampling”. In: *Artificial Intelligence* 296 (2021), p. 103476. ISSN: 0004-3702. DOI: 10.1016/j.artint.2021.103476.
- [2] James Brooks. “Ballot Measure 2 eeks out narrow win with nearly all votes in”. English. In: *Alaska Journal of Commerce* 44.47 (Nov. 2020). Copyright - Copyright MCC Magazines, LLC d/b/a Alaska Magazine Nov 22, 2020; Last updated - 2023-11-07; SubjectsTermNotLitGenreText - United States-US; Alaska, p. 6. URL: <https://www-proquest-com.libdb.fairfield.edu:8443/trade-journals/ballot-measure-2-eeks-out-narrow-win-with-nearly/docview/2467349692/se-2>.
- [3] Alborz Jelvani and Amelie Marian. “Identifying Possible Winners in Ranked Choice Voting Elections with Outstanding Ballots”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 10.1 (Oct. 2022), pp. 114–123. DOI: 10.1609/hcomp.v10i1.21992. URL: <https://ojs.aaai.org/index.php/HCOMP/article/view/21992>.
- [4] M. G. Kendall. “A New Measure of Rank Correlation”. In: *Biometrika* 30.1/2 (1938), pp. 81–93. ISSN: 00063444. URL: <http://www.jstor.org/stable/2332226> (visited on 05/04/2024).
- [5] *Kohlhaas v. Alaska*. Oct. 21, 2022. URL: <https://caselaw.findlaw.com/court/ak-supreme-court/1972157.html> (visited on 05/05/2024).
- [6] Thomas R. Magrino et al. “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. San Francisco, CA: USENIX Association, 2011, p. 4.
- [7] OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at <http://oeis.org>. 2024.
- [8] Deb Otis. *An analysis of statewide election recounts, 2000–2023*. Accessed: 2024-04-29. 2023. URL: <https://fairvote.org/report/election-recounts-2023/>.
- [9] Deb Otis. *Ranked Choice Voting in 2023: A year in review*. 2023, pp. 1–18. URL: <https://fairvote.org/report/2023-a-year-in-review/> (visited on 05/05/2024).
- [10] Iris Samuels. “Lawsuit asserts Alaska elections officials illegally certified ballot question on ranked choice voting”. In: *Anchorage Daily News* (Apr. 3, 2024). URL: <https://www.adn.com/politics/2024/04/02/lawsuit-asserts-alaska-elections-officials-illegally-certified-ballot-question-on-ranked-choice-voting/> (visited on 05/05/2024).

- [11] Anand D. Sarwate, Stephen Checkoway, and Hovav Shacham. “Risk-limiting Audits and the Margin of Victory in Nonplurality Elections”. In: *Statistics, Politics, and Policy* 4.1 (2013), pp. 29–64. DOI: [doi:10.1515/spp-2012-0003](https://doi.org/10.1515/spp-2012-0003).