Real Democracy: Post-Election Audits for Range Voting

Berj Chilingirian, Eric Huppert, Zara Perumal
MIT CSAIL, {berjc, ehuppert, zperumal}@mit.edu

May 11, 2016

***Abstract*—The election system of the United States of America is flawed in two ways. First, it relies on a voting scheme that does not enforce true democracy. Second, it relies on electronic voting systems to properly tabulate ballots despite such systems being susceptible to bugs and malicious behavior. To address the first issue, we advocate for the use of range voting, a scheme that allows a voter to express their true and complete preferences. To address the second concern, we propose four new post-election, ballot-polling audits for range voting contests. We compare our auditing procedures to existing strategies via a simulated error rate analysis. As a consequence of this research, we propose a model for generating range voting contests and implement the `ElectionEngine`, an open source election simulation platform. Finally, we discuss the tradeoffs present in choosing post-election audits.**

I. Introduction

The 2016 presidential election has arguably been one of the most entertaining elections in recent years [1]. At one point the contest consisted of six democratic and 17 republican candidates, including controversial businessman Donald Trump and former Secretary of State Hilary Clinton [2]. Nevertheless, the field has narrowed to three candidates and come November, the general public will make a pivotal decision on the future direction of the United States of America. A large portion of voters across the country will travel to their local polling sites and cast their choice as a traditional paper ballot [3]. These paper ballots will then be tabulated by an electronic voting system and the result of the election will be published.

The nature of this election brings to light two issues that threaten the integrity of our voting process. First, the field of candidates is thinned by a sequence of primary contests and caucuses whose order may determine the fate of a candidate before that candidate has the chance to compete in every state. This effect may be attributed to the United State's simple plurality system in which voters may choose only one candidate to support, with no way to distinguish a second option from a last choice. By prioritizing logistics over the voter's voice, this phenomenon weakens the democracy of the election process [4].

Second, the general public relies entirely on the honesty and correctness of voting system software to report the true election outcome. It could be, however, that *Trump Technologies* and/or *Clinton Cryptosystems* has software/firmware/hardware incorporated into voting systems that influences the election outcome.[1] Moreover, considering the prevalence of bugs in industry software (15-50 errors per 1000 lines of code) [5], it could be that the voting system has a bug that causes it to report the wrong outcome. In either case, the democracy of the election process is threatened.

To address the first concern, the United States government may consider replacing the traditional simple plurality scheme with a range voting scheme. In range voting, each voter *scores* every candidate from $[0, m]$ where m is the maximum score (e.g. 100). In this manner, a voter may specify their preference for each candidate independent of their preferences for other candidates. To determine a winner, the mean/median of each candidate's score across all ballots can be computed and the candidate with the highest score is declared the election winner. Range voting gives the voter the power to express their complete preferences and has numerous favorable properties compared to simple plurality voting [6].

Of course, the democracy delivered by range voting may still be violated by un/intentional malicious voting system software. This problem, however, may be mitigated by a post-election audit in which paper ballots are polled at random and used to either verify the election outcome or signal a manual recount. Numerous auditing procedures have been proposed in the literature [7], including both scheme-specific [8] (e.g. simple plurality-specific auditing procedures) and black-box procedures that work for any social choice function [9]. As the voting system is not trusted, an audit may provide *evidence* that the reported outcome is correct [10].

Despite the plethora of research in election audits, there has been no work in ballot-polling audits specifically for range voting. This paper proposes several ballot-polling audits for range voting and demonstrates their performance via simulation. Our problem statement is as follows.

Problem Statement: Can we develop post-election audits for range voting that outperform existing black-box methods?

In solving this problem, our paper makes the following contributions.

- **Comparison of Black-Box Audits:** We replicate and extend the results in [9] by comparing several black-box

[1] To our knowledge, neither *Trump Technologies* nor *Clinton Cryptosystems* are real corporations.

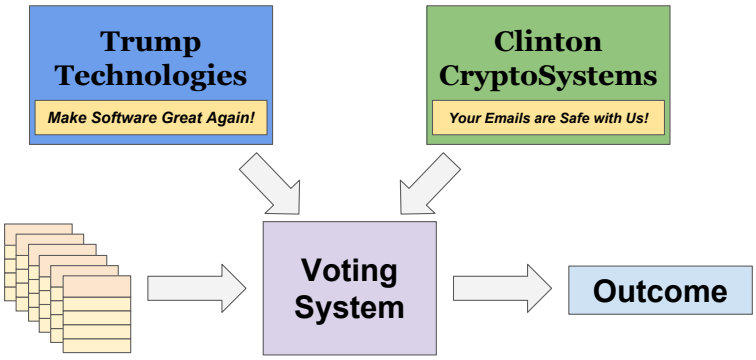

Fig. 1. **Post-Election Audit Threat Model.** We cannot trust that the software in our electronic voting system produces the actual outcome for every contest. For instance, it could be that the software vendor of the system is biased towards a specific political party and/or that the software contains bugs.

auditing schemes.

- **Four New Ballot-Polling Audits for Range Voting:** We propose four new ballot-polling audits for range voting contests. Two of the four proposed audits are considered black-box audits.
- **Model for Range Voting Contests:** We propose a way to model range voting contests using Yee-Pictures [11].
- **Error Rate Analysis and Comparison to Existing Audits:** We compare our proposed audits to existing techniques by measuring their error rates on simulated contests.
- **Open Source Election Simulation Platform:** We have developed `ElectionEngine`, an open source election simulation platform for anyone to write and test their own auditing schemes on both simple plurality and range voting contests.

The rest of this paper is outlined as follows. Section 2 presents background information on our notation and previous work on post-election audits. Section 3 presents the threat model faced by post-election audits. Section 4 describes range voting and our proposed method for generating a range voting contest. Section 5 describes the four post-election auditing procedures we developed for range voting. Section 6 presents a simulation-based comparison of the performance of our methods against the performance of other black-box methods. Section 7 discusses these results and section 8 concludes. Finally, section 9 mentions possible future directions for the work presented.

II. Background

To develop an auditing procedure for a range voting contest, we must first understand what is meant by an election contest and review previous work in contest auditing.

A. Preliminaries

An election consists of a set of contests. Each contest consists of a set of candidates and a set of ballots making up the voter *profile* for that contest [9]. At the end of an election, each contest publishes a *reported* outcome, R. The reported outcome is not necessarily equal to the *actual* outcome, A, of the contest, or the outcome computed manually from the contest's profile [12]. For example, it could be that the reported outcome is incorrect due to a software bug in the vote tabulation system.

B. Post-Election Audits

Post-election audits provide evidence that the reported outcome of an election contest corresponds to the actual outcome of that contest. Such audits sample the contest's profile repeatedly to build evidence for the reported outcome. A post-election audit may conclude by either (1) producing strong evidence that the reported outcome of an election contest is equal to the actual outcome of that contest or (2) reporting the actual outcome of the election contest by manually recounting all ballots. An audit that concludes with (2) implies that the evidence collected by the audit was not in favor of the reported outcome.

Post-election audits come in two forms.

1) **Ballot-Polling Audit:** A *ballot-polling audit* derives evidence from the paper ballots of a contest.
2) **Comparison Audit:** A *comparison audit* derives evidence by comparing paper ballots to their electronic representations in voting machines.

While comparison audits often inspect fewer ballots than ballot-polling audits, they require the voting system to track a correspondence between the electronic representation of a ballot and its paper form [9]. Thus, we only consider ballot-polling audits in this paper. It is worth noting, however, that a comparison audit for range voting has been proposed [13].

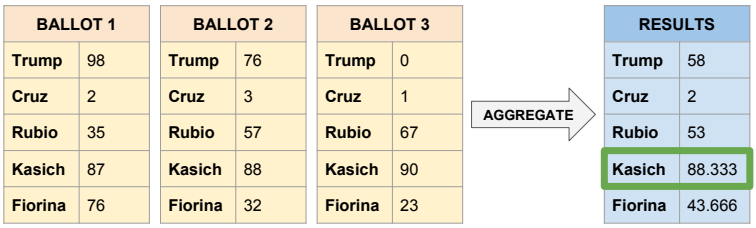

Fig. 2. **Mean Range Voting Contest.** In range voting, a voter assigns a score in the range from $[0, m]$ for all candidates. Then, the contest outcome is determined by aggregating the scores of each candidate across all ballots and declaring the candidate with the highest aggregated score as the winner. Here the "aggregator" is the mean of a candidate's score across all ballots.

C. Risk-Limiting Audits

A *risk-limiting audit* provides statistical assurance of the audit's *risk* of accepting an incorrect result as an upper bound α where $0 < \alpha < 1$ [14]. DiffSum is an example of a risk-limiting audit that uses a simple stopping rule to determine whether enough evidence has been collected to validate or overturn the reported outcome of a simple plurality contest. The stopping rule, shown below, takes the number of votes for R, a, and the number of votes for the strongest loser (i.e. second place), b, and determines whether their squared difference is greater than their sum times c where c controls risk.

$$(a-b)^2 > c \cdot (a+b)$$

Risk-limiting audits like [12] sacrifice generality for a theoretical bound on risk.

D. Black-Box Audits

In contrast to risk-limiting audits, black-box methods tradeoff a theoretical bound on risk for applicability to *any* social choice function f (e.g. simple plurality, instant-runoff voting (IRV), range voting, etc.) [9]. Without theoretical guarantees, black-box audits are measured by their error rate, or the frequency with which they accept an incorrect outcome. There have been several notable black-box audits proposed in the literature.

1) T-Pile Audit: The *T-pile* audit draws a random sample of S ballots without replacement from the contest's profile and distributes them into T equally sized piles p_i for $i \in [1, T]$. If $R = f(p_i)$ for all $i \in [1, T]$, the audit ends by confirming the reported outcome R. Else, the size of S is increased and the audit continues. Rivest and Stark recommend $T = 7$ and a sampling schedule detailed in appendix A [9].

2) Bootstrap Audit: The *bootstrap* audit draws a random sample of S ballots without replacement from the contest's profile and creates T "variant" samples v_i for $i \in [1, T]$ each with size $|S|$. Each v_i is constructed by sampling S with replacement. If $R = f(v_i)$ for all $i \in [1, T]$ and $R = f(S)$, the audit ends by confirming the reported outcome R. Else, the size of S is increased and the audit continues. Rivest and Stark recommend $T = 100$ and the same sampling schedule as used by the T-pile audit [9].

3) Bayes Audit: The *bayes* audit draws a single random ballot at a time, adds it to a growing sample S, and then asks: for each possible outcome of the contest, what is the probability of that outcome being the actual outcome if the audit continues for ballots similar to those in S? This question is answered via a number of simulations that introduce variance in the vote tallies for the given sample. The Bayes audit terminates when R has a winning probability greater than or equal to $1 - \alpha$ where α represents the risk of the audit (although it has yet to be shown that the Bayes audit is risk-limiting). If an outcome not equal to R is found with a probability greater than or equal to $1-\alpha$ then a full manual recount is performed [16]. We implement the Bayes audit using gamma variates (see http://people.csail.mit.edu/rivest/bayes/bayes.js for details).

III. Threat Model

To develop sound post-election audits for range voting, we must understand the threats that they face. As shown in Figure 1, we do not trust the software of the electronic voting system to publish the actual contest outcome due to either intentional or unintentional bugs. We can, however, trust that the paper ballots of the contest have not been tampered with and are available to our post-election audit.

IV. Range Voting Model

Range voting is an election scheme in which each voter scores each candidate of a contest from $[0, m]$ where m is the maximum score (e.g. 100). Then, the outcome of the contest is computed by aggregating the scores of each candidate across all ballots and reporting the candidate with the highest aggregated score.

We are interested in auditing two specific aggregation procedures.

1) **Mean:** A candidate's aggregated score in the contest is the mean of the scores on all ballots.

2) **Median:** A candidate's aggregated score in the contest is the median of the scores on all ballots.

There are numerous variations on range voting not considered in this paper. This includes using a truncated mean as an aggregation procedure as well as allowing voter's to indicate they have "no preference" for a given candidate on their ballot [6]. A mean range voting contest is shown in Figure 2.

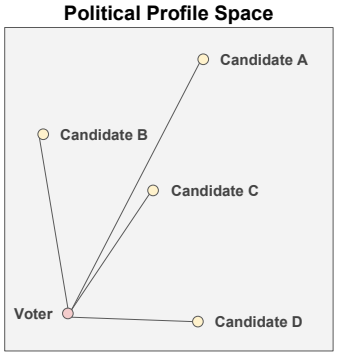

Fig. 3. **Yee-Pictures.** A Yee-Picture is a visualization of the space of possible political profiles where any point in the space is a specific political view. Note that the closer two points are in the space, the closer their political views are to one another's.

In order to test the efficacy of our proposed audits, we must run a large number of simulated range voting contests. This requires a procedure for generating a profile of range voting ballots. To do this, we propose using Yee-Pictures [11]. A Yee-Picture is a finite, two-dimensional plane where point (x, y) represents the political profile of a person. The closer two points are on the plane, the more similar their political views are. Then, we may use the following procedure to generate n ballots for an k-candidate range voting election.

1) Randomly sample k points on the plane representing the political views of the k candidates.

2) Randomly sample n points on the plane representing the political views of the n voters. The score attributed by a voter to a candidate is the Euclidean distance between the voter and the candidate's points on the plane divided by the diagonal of the plane (or the maximum possible distance between any two political views) times m, the maximum score for the given range voting contest.

$$score_{voter}(candidate_i) = \frac{m \cdot EuclidDist(voter, candidate_i)}{|Diagonal|}$$

This is an *honest* range voting model in the sense that each voter scores each candidate with respect to their true political opinion. In contrast, a *dishonest* range voting model would permit a voter to score their "favorite" candidate with the highest possible score and all other candidates with a score of 0. Despite this distinction, we believe that auditing an honest range voting model is more difficult than auditing a dishonest model. This is because the margins between candidates' aggregated scores are closer in the honest model than in the dishonest model. Thus, we adopt the honest range voting model for our simulations.

It is also worth noting that the dimensionality of a voter's political view may be increased to introduce additional complexity in the space of political views. For simplicity, our simulations use only two dimensions, as shown in Figure 3.

V. Range Voting Audits

We propose four new post-election, ballot-polling audits for range voting. Note that all our audits adopt the sampling schedule described in appendix A. Figure 4 gives a visual interpretation of our proposed auditing procedures. It is worth noting that both our SubSim and informed T-pile audits are considered black-box audits.

A. DiffSumScore Audit

The *DiffSumScore* audit is simply the application of the DiffSum stopping rule described in §2-C to range voting scores. The DiffSumScore audit procedure works as follows.

1) Sample s_i ballots at random from the contest's profile and add them to the growing sample S.

2) Compute the aggregated score of each candidate for S. If R does not have the largest aggregated score from S, return to step 1 with an increased sample size s_{i+1}.

3) Call a the aggregated score for R and b the aggregated score for second place. If
$$(a - b)^2 > c \cdot (a + b)$$
the audit ends and R is confirmed. If the expression is not true, the audit continues by increasing the sample size to s_{i+1}.

Thus, the DiffSumScore audit terminates when either the reported outcome is confirmed or a manual recount has been performed. The author of DiffSum suggests performing a full recount after sampling 4 percent of the contest profile without satisfying the stopping condition [15].

B. SubSim Audit

The *SubSim* audit repeatedly subsamples a random sample of the contest's profile and then simulates a number of contests on this subsample. The SubSim audit procedure works as follows.

1) Sample s_i ballots at random from the contest's profile and add them to the growing sample S.

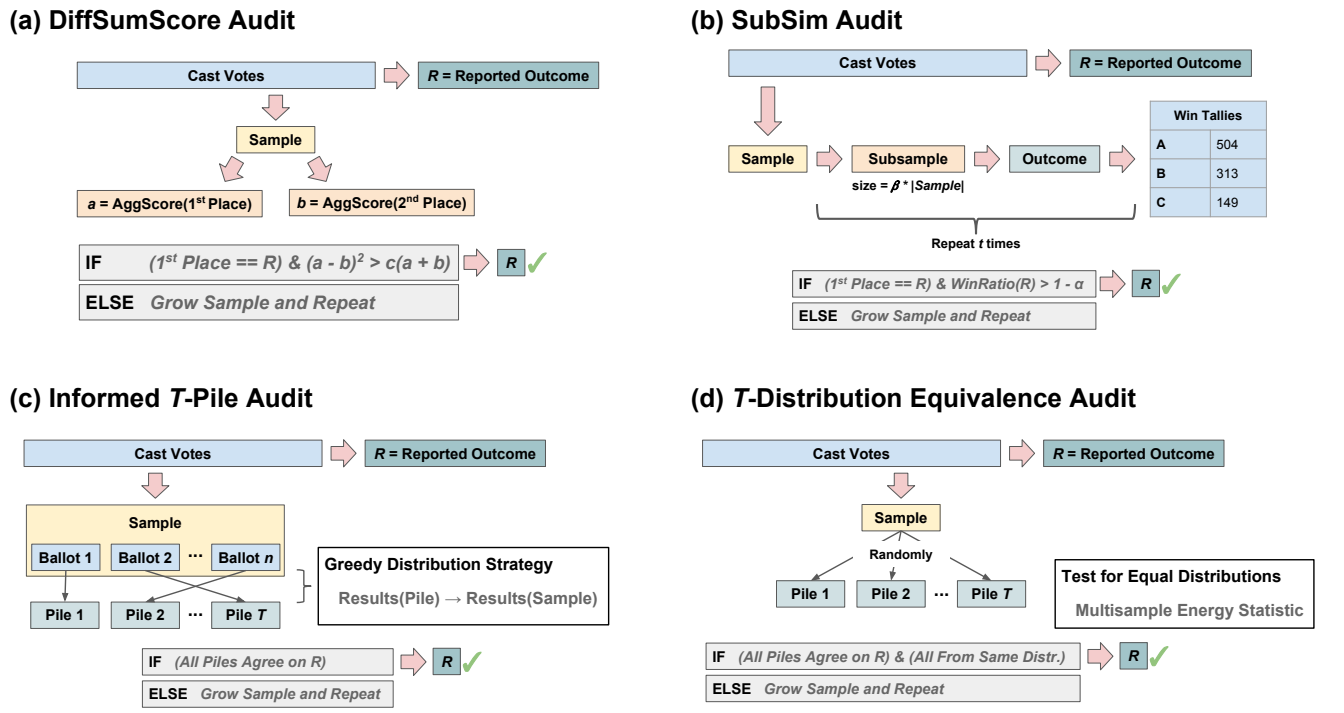

Fig. 4. **Visual Interpretation of Proposed Audits.** Note that all our audits have a similar structure. First, they sample from the cast votes. Then, they perform a computation on the information gathered from the sample. Finally, they check the stopping rule and terminate if it is satisfied, confirming the reported outcome; otherwise, they grow the sample and repeat.

2) Randomly subsample $\beta \cdot |S|$ ballots from S to form S'.

3) Compute the outcome of a contest with profile S' and record the winner.

4) Repeat from step 2 onwards t times.

5) After t simulated contests, check that the candidate with the highest number of wins is R. If not, increase the sample size to s_{i+1} and return to step 1.

6) Check that the frequency of R's wins is greater than $1 - \alpha$. If not, increase the sample size to s_{i+1} and return to step 1. Otherwise, terminate the audit and confirm R.

Thus, the SubSim audit terminates when either the reported outcome is confirmed or a manual recount has been performed. The effectiveness of the audit is a function of the parameters (β, t, α). Note, however, that the SubSim audit has a similar structure to the bootstrap audit in that both form "piles" with replacement. The distinction is in the stopping rule: the bootstrap audit terminates when agreement is reached amongst all piles while the SubSim audit terminates when the threshold frequency is satisfied.

C. Informed T-Pile Audit

The *informed T-pile* audit modifies the standard T-pile audit (see §2-D1) to distribute votes using a strategy that favors convergence. The informed T-pile audit works as follows.

1) Sample s_i ballots at random from the contest's profile and add them to the growing sample S.

2) Compute the aggregated score of S, $AggScore(S)$, resulting in an k-dimensional vector where k is the number of candidates and each entry in the vector is the aggregated score for the corresponding candidate in the sample S.

3) Define the benefit $B(p_i|b_j)$ of adding a ballot b_j to a pile p_i to be $EuclidDist(AggScore(S), AggScore(p_i))$ $-EuclidDist(AggScore(S), AggScore(p_i \cup \{ballot\})$. Then, for each ballot b_j added to S in this round, determine the pile p_i for $i \in [1, T]$ that receives the maximum benefit $B(p_i|b_j)$ from adding b_j to its pile and add b_j to p_i. Perform this procedure for a random ordering of ballots without replacement. Continue until all ballots for this round have been added to a pile and each pile has equal size.

4) If all piles p_i for $i \in [1, T]$ agree on R, terminate the audit and confirm R. Otherwise, increase the sample size to s_{i+1} and return to step 1.

Thus, the informed T-pile audit terminates when either the reported outcome is confirmed or a manual recount has been performed. The informed T-pile audit takes advantage of range

Two-Candidate Plurality Contests								
	T-Pile ($T = 7$)		Bootstrap ($T = 100$)		DiffSum ($c = 5$)		Bayes ($\alpha = 0.05$)	
p	ballots	error	ballots	error	ballots	error	ballots	error
0.7500	35.665	0.000	28.882	0.000	28.518	0.000	24.479	0.000
0.7000	50.967	0.000	39.865	0.001	38.682	0.000	29.554	0.000
0.6500	83.965	0.000	61.103	0.001	62.601	0.001	42.868	0.003
0.6000	188.335	0.002	129.864	0.001	134.932	0.004	78.995	0.011
0.5500	699.944	0.006	494.851	0.012	481.089	0.005	232.365	0.047
0.5400	1130.528	0.010	738.227	0.019	737.093	0.008	351.400	0.054
0.5300	1911.637	0.009	1332.898	0.035	1295.406	0.017	554.071	0.074
0.5200	4372.550	0.016	2785.566	0.040	2615.655	0.027	1080.744	0.110
0.5100	14539.804	0.050	10080.924	0.071	9330.286	0.059	2566.550	0.177
0.5050	41026.510	0.058	32677.987	0.080	30026.692	0.074	7152.985	0.235

Fig. 5. **Results for Auditing Performance of Simple Plurality Contests.** The results report the average number of ballots checked and the error rates of four audit methods for 1000 simulated two-candidate contests of $100,000$ ballots. The p value denotes the fraction of the votes that went to the first place winner. It is clear that the closer p is to 0.50, the harder it is to audit the contest. These results are very similar to those shown in [9]. In addition, it is worth noting that the Bayes audit on average checks fewer ballots than the other audits, but has a much higher error rate for all values of p.

voting margin information to move each pile towards the distribution of the sample and to accelerate agreement across all piles. In this manner, the informed T-pile audit is less conservative than the standard T-pile audit.

D. T-Distribution Equivalence Audit

The *T-distribution equivalence* audit modifies the stopping rule of the standard T-pile audit (see §2-D1) such that the audit confirms the reported outcome R if all piles agree on R and all piles come from the same distribution. This distribution "equivalence" is computed using Székely *et al.*'s technique for comparing a set of multi-dimensional distributions via energy statistics [17].

1) Sample s_i ballots at random from the contest's profile and add them to the growing sample S.

2) Evenly and randomly distribute the new ballots to the T piles.

3) If all piles p_i for $i \in [1, T]$ agree on R and all piles come from the same distribution with $p_{value} < \alpha$, terminate the audit and confirm R. Otherwise, increase the sample size to s_{i+1} and return to step 1.

Thus, the T-distribution equivalence audit terminates when either the reported outcome is confirmed or a manual recount has been performed. The T-distribution equivalence audit takes advantage of range voting margin information to impose a stricter stopping rule on agreement. In this manner, the T-distribution equivalence audit is more conservative than the standard T-pile audit.

VI. Results

In order to test the performance of our proposed audits, we implemented the `ElectionEngine`, an open source election simulation platform (see appendix B for details). The experiments we ran on the `ElectionEngine` are two-fold. First, we replicated and extended the results presented in [9]. Second, we compared our auditing procedures against the black-box audits described in §2-D for range voting contests.

A. Comparison of T-Pile, Bootstrap, DiffSum, and Bayes

We used the experimental design described in [9] to test our implementations of the T-pile, bootstrap, and DiffSum audits. In addition, we included the Bayes audit and performed a comparison of its performance against the other audit methods. To our knowledge, this is the first comparison of the Bayes audit with the T-pile and bootstrap audits.

We ran 1000 trials of a simulated two-candidate simple plurality contest with $100,000$ ballots. All audits used the sampling schedule described in appendix A. The number of votes allocated to the winner of the two-candidate contest was varied by p where p is the fraction of the $100,000$ ballots given to the winner of the contest. Both the average number of ballots counted and the error rate (fraction of contests for which the audit confirmed the wrong candidate) were reported. The results are shown in Figure 5.

B. Experimental Results for Range Voting Audits

We ran 1000 trials of a simulated four-candidate simple plurality contest, a four-candidate mean range voting contest, and a four-candidate median range voting contest, each with $100,000$ ballots. All audits used the sampling schedule described in appendix A. The simple plurality contests were generated by distributing the votes to the four candidates in a 4:3:2:1 ratio as done in [9]. The range voting contests were generated by the procedure described in §4. As before, the average number of ballots counted and the error rate were reported. The results are shown in Figure 6. Note that the T-distribution equivalence audit was not included in the results. Its absence is explained in §7.

VII. Discussion

We expand on the results presented in [9] on black-box audits by including the Bayes audit. As seen in Figure 5, the

Four-Candidate Plurality and Range Voting Contests												
	T-pile ($T = 7$)		Bootstrap ($T = 100$)		Bayes ($\alpha = 0.05$)		DiffsumScore ($c = 5$)		SubSim ($0.7, 1000, 0.05$)		Informed T-Pile ($T = 7$)	
Contest Type	ballots	error	ballots	error	ballots	error	ballots	error	ballots	error	ballots	error
Plurality	555.716	0.001	377.300	0.004	201.488	0.014	351.148	0.007	80.234	0.054	54.957	0.038
Mean Range	3899.713	0.009	3006.216	0.012	744.758	0.157	95801.995	0.000	165.354	0.091	78.680	0.099
Median Range	5676.029	0.010	5866.507	0.013	1140.181	0.150	91003.759	0.001	373.714	0.121	262.815	0.099

Fig. 6. **Results for Auditing Performance of Plurality and Range Voting Contests.** The results report the average number of ballots checked and the error rates of six audit methods for 1000 simulated four-candidate contests of $100,000$ ballots. The simple plurality contests were generated by distributing votes to each candidate proportional to 4:3:2:1. The range voting contests were generated using the procedure described in §4. It is worth noting that the SubSim and informed T-pile audits produce a lower error rate than the Bayes audit and check fewer ballots than any other audit tested. The parameters of the SubSim audit correspond to (β, t, α) as described in §5-B.

Bayes audit checks fewer ballots than any other audit for all values of p. In addition, for $p < 0.7000$, the Bayes audit has a higher error rate than any other audit tested. This result is quite fascinating. The Bayes audit relies on the intuition of the Baye's rule in order to limit the risk of accepting an incorrect reported outcome. In contrast, the T-pile and bootstrap audits use "math-free" procedures to audit contests without any direct parameter for limiting risk. This result is a testament to the effectiveness of the T-pile and bootstrap audits as proposed in [9]. Otherwise, the replicated results in Figure 5 match those presented in [9].

We now discuss the results presented in Figure 6. First, we note that the T-distribution equivalence audit is absent from the results. In preliminary testing of the T-distribution equivalence audit we found that the dimensionality of each sample was too high for the algorithm to converge. More specifically, for the T-distribution equivalence audit to terminate, it must determine that each dimension of each pile of ballots is from the same population with high statistical significance. Unfortunately, the complexity of a four candidate election was too high for the audit to terminate for reasonable p-values. We continue to explore variations of this technique for auditing range voting contests.

The DiffSumScore audit had the lowest error rate for auditing range voting contests. Nevertheless, it checked more than 90 percent of the total ballots before confirming the reported outcome. This high number of ballots can be understood by the nature of range voting ballots. In simple plurality contests each ballot increases the aggregate score of a single candidate. In range voting contests, however, each ballot may increase the aggregate score of multiple candidates. Then, suppose a close range voting election between candidates A and B. Each round of the DiffSumScore audit will likely increase the aggregate scores of A and B proportional to one another. In this case there is no "progress" being made towards satisfying the stopping rule and the audit must check more ballots. This line of reasoning may explain why the DiffSumScore audit must check a high number of ballots before confirming the reported outcome.

Our two remaining audits, the SubSim and informed T-pile audits, checked fewer ballots and produced lower error rates for both the mean and median range voting simulations. The SubSim and informed T-pile audits also checked fewer ballots than both the T-pile and bootstrap audits (on the order of 15 to 20 times fewer ballots). Nevertheless, both the T-pile and bootstrap audits had lower error rates than the SubSim and informed T-pile audits (on the order of 10 times lower error rates). Again, this speaks to the elegance of Rivest and Stark's black-box methods: they are both "math-free" and accurate. With respect to each other, it seems as though the informed T-pile audit outperforms the SubSim audit, checking fewer ballots for approximately the same error rates.

Finally, it is worth noting that audits checked fewer ballots (often on the order of half the ballots) for a mean range voting contest as compared to a median range voting contest. We might attribute this phenomenon to the sensitivity of the median to a new data point as compared to the mean.

With these results in mind, we must question the tradeoff between the number of ballots checked and the error of accepting an incorrect reported outcome. This tradeoff may dictate the appropriate audit to use. For example, given a mean range voting contest, contest administrators may use the SubSim audit for contests in which they can afford to accept an increased chance of error for checking fewer ballots as compared to the T-pile audit. In contrast, contest administrators may choose the T-pile audit over our methods if the error rate must be minimized at all costs. In summary, we suspect that the resources available to the administrators of a contest may dictate the appropriate audit to use. We recommend, however, that auditing procedures be compared according to the average number of ballots they check divided by their error rate.

VIII. Conclusion

We propose and test four new post-election auditing procedures for range voting contests. We compare the performance of our audits to existing black-box schemes. Our experiments verified the efficacy of two of our auditing procedures: the SubSim and informed T-pile audits. We find that these audits outperform the Bayes audit and tradeoff checking far fewer ballots for increased error rates compared to the T-pile and bootstrap audits. In performing these experiments we also showed that the Bayes audit checks ballots for an increased error rate compared to the T-pile and bootstrap audits. In addition, we developed a model for generating range voting contests. Finally, we developed the `ElectionEngine`, an open source election simulation platform for anyone to write and test their own auditing schemes on both simple plurality and range voting contests. We hope this work encourages others to explore auditing procedures for strengthening the democracy of range voting contests.

IX. Future Work

The performance of both the SubSim and informed T-pile audits depend on their respective parameters. While tuning these parameters was outside the scope of this project, we believe that such efforts could improve the performance of both audits. In fact, it is our hypothesis that the SubSim audit may be tuned to produce error rates comparable to that of the T-pile and bootstrap audits while still checking far fewer ballots.

Appendix A
Sampling Schedule

We adopt the sampling schedule proposed by Rivest and Stark in [9]. We present it here for the reader's convenience.

The audit begins with a sample size of $s_0 = 21$ ballots. Then, given the previous sample of size s_i, the next sample grows by $\lceil r * s_i \rceil_d - s_i$ where r is the growth rate and d is the divisor. Here, $\lceil x \rceil_d$ refers to rounding x up to the next multiple of d. We use $r = 1.1$ and $d = 7$ as proposed by Rviest and Stark in [9].

Appendix B
The ElectionEngine

The `ElectionEngine` is an open source election simulation platform for researchers to implement and experiment on new and existing auditing procedures. The code may be found at https://github.com/berjc/election-engine.

Acknowledgment

We would like to thank Professor Ronald Rivest, Institute Professor at the Massachusetts Institute of Technology, for stimulating our interest in post-election audits and guiding us throughout our research.

References

[1] "2016 Election: 13 Jaw-Dropping Moments." *CNN*. Cable News Network, n.d. Web. 04 May 2016.

[2] "Presidential Candidates, 2016 - Ballotpedia." *Presidential Elections*. Ballotpedia, n.d. Web. 04 May 2016.

[3] Rivest, Ronald. "Auditability and Verifiability of Elections." 6.857: Computer and Network Security - Lecture 20. Massachusetts Institute of Technology, Cambridge. 20 Apr. 2016. Lecture.

[4] Rosenbaum, David E. "Relax, Nader Advises Alarmed Democrats, but the 2000 Math Counsels Otherwise." *The New York Times* (2004).

[5] McConnell, Steve. *Code complete*. Pearson Education, 2004.

[6] "Get Real Democracy." *RangeVoting.org*. The Center for Range Voting, n.d. Web. 4 May 2016.

[7] Lawrence Norden, Aaron Burstein, Joseph Lorenzo Hall, and Margaret Chen. Post-Election Audits: Restoring Trust in Elections. Technical report, Brennan Center for Justice and Samuelson Law, Technology & Public Policy Clinic, 2007.

[8] Lindeman, Mark, Philip B. Stark, and Vincent S. Yates. "BRAVO: Ballot-polling Risk-limiting Audits to Verify Outcomes." *EVT/WOTE*. 2012.

[9] R.L. Rivest and P.B. Stark, "Black-Box Post-Election Audits." Unpublished draft, Mar. 2016.

[10] Stark, Philip B., and David Wagner. "Evidence-Based Elections." *Security & Privacy, IEEE* 10.5 (2012): 33-41.

[11] Yee, Ka-Ping. "Voting Simulation Visualizations." *Voting Simulation Visualizations*. N.p., 8 Dec. 2006. Web. 04 May 2016.

[12] Stark, Philip B. "Super-Simple Simultaneous Single-Ballot Risk-Limiting Audits." *EVT/WOTE*. 2010.

[13] Sarwate, Anand, Stephen Checkoway, and Hovav Shacham. *Risk-Limiting Audits for Nonplurality Elections*. No. CS2010-0967. CALIFORNIA UNIV SAN DIEGO LA JOLLA, 2011.

[14] Mark Lindeman and Philip B. Stark. A Gentle Introduction to Risk-Limiting Audits. *IEEE Security and Privacy*, 10:42?49, 2012.

[15] Rivest, Ronald L. "DiffSum: Post-Election Risk-Limiting Audit in a Page." (2015).

[16] Rivest, Ronald L., and Emily Shen. "A Bayesian Method for Auditing Elections." *EVT/WOTE*. 2012.

[17] Székely, Gábor J., and Maria L. Rizzo. "Testing for Equal Distributions in High Dimension." *InterStat* 5 (2004): 1-6.