Single-Ballot Risk-Limiting Audits Using Convex Optimization

Stephen Checkoway
UC San Diego

Anand Sarwate
UC San Diego

Hovav Shacham
UC San Diego

Abstract

We take an information-theoretic approach to sequential election auditing. By comparing how far an empirical distribution of audited votes diverges from any distribution in which the reported outcome is incorrect, we gain a high degree of confidence in the outcome when our procedure confirms the reported results.

1 Introduction

Post-election audits are a standard method of providing some assurance that the reported outcome of an election actually reflects the voters' intent. When votes are cast on paper ballots, post-election audits allow verification that the election outcome is correct, independent of any misbehavior by ballot scanning and tabulation machinery (see, e.g., [11]). Many electoral jurisdictions now require some sort of post-election audit such as California's hand recount of 1% of the precincts.

We believe that audit procedures must have three properties to be useful in real elections to (statistically) guarantee that the final outcome is correct.

- First, audits must be *risk-limiting* [25]. If the audit certifies the outcome reported in the initial count, then either that outcome is correct or a bad event occurred — namely the audit failed to find enough evidence that the reported outcome is incorrect when it actually is. An audit is risk-limiting if its design provides a statistical bound on the probability that the bad event occurred.

- Second, audits must be *resilient to error*. Some audits are designed to find a discrepancy: a single miscounted ballot. Such audits allow efficient certification of an election when some fraction of the ballots are counted and no discrepancies are uncovered. However, our analysis of data from the 2008 Minnesota Senate race suggests that a small but nonzero fraction of ballots are miscounted in real elections; when such discrepancies are few compared to the reported election margin they should not lead to a full hand count.

- Third, audits must be *feasible*. LA County — admittedly an outlier — requires counting seven days a week for almost the entire 28 days allowed by California law to complete its 1% manual count [17]. When elections are extremely close or the initial count is incorrect, it is natural to expect that every ballot must be counted by hand, but audits that mandate manual counts of 20% of the ballots even in an election with a wider margin of victory could not realistically be implemented.

An audit may be risk-limiting without being resilient to error. For example, audits designed to look for a single miscounted ballot are clearly not resilient to error; however, if after finding a miscount, they proceed to a full hand count then they *are* risk-limiting.

Our contribution. In this paper, we propose an auditing scheme that is risk-limiting, resilient to error, and feasible. Unlike many previous auditing schemes, ours operates at the level of ballots, not precincts. As we discuss below, this means that it requires a mechanism for identifying individual ballots, a nontrivial change from today's election procedures. In exchange, our scheme is able to provide strong statistical risk limits while counting many fewer ballots than current precinct-based auditing schemes.

As discussed below, we believe that our risk analysis can be improved, giving rise to even more efficient auditing schemes using our techniques. Even so, the efficiency afforded by ballot-based audits like ours or like Stark's [26] suggests an important open problem: Do comparably efficient precinct-based auditing schemes exist? The current lack of such schemes provides a strong argument for investing in the infrastructure needed to support ballot-based auditing.

At the core of our algorithm is a simple but powerful idea. Each ballot has a reported value, which we know from the initial count, as well as an actual value, which we would determine by means of a hand count. We are concerned about *consequential* errors: ones in which the actual winner differs from the reported one. Suppose we knew that our adversary, who is trying to steal the election, would be using a specific joint distribution M of actual and reported values: for each ballot, he has deter-

mined what its reported values will be based on its actual value. By sampling and counting some fraction of the ballots, we obtain an approximation $\hat{M}$ to the underlying distribution. The question we ask is: *Assuming the underlying distribution is M, how likely is a random sample of ballots to yield the approximation $\hat{M}$?* (The probability is computed over the choice of ballots to count.) If this probability is sufficiently small, say less than some bound ξ, then this is evidence that the underlying distribution is not M, so the adversary is not stealing the election, and we can certify the election. The risk of miscertification is the probability, assuming that the underlying distribution actually is M, of drawing a sample $\hat{M}$ so very unlikely (i.e., below the bound ξ) that we certify the election. By setting ξ appropriately, we can guarantee that this probability is at most α, the desired miscertification risk level.

In the real world, of course, adversaries are not so helpful as to tell us the distribution of their malfeasance. To rule out any consequential tampering, we would have to compare against *all* distributions of actual and reported values in which the actual winner differs from the reported one. Unfortunately, there are far too many such distributions that we would have to consider — the number grows like n^8 for a two-candidate election with n ballots. In general, the exponent grows with the square of the number of candidates.

Surprisingly, by using tools from information-theoretic statistics and from convex optimization, we are able to do just what we have argued is difficult: given sampled ballots, rule out *all* distributions of actual and reported values in which the actual winner differs from the reported one, using a computationally efficient procedure. This is because the set of distributions against which we must compare is convex, and we can minimize our test statistic (Kullback-Leibler divergence) over this set using convex optimization procedures.

Building on our techniques, we propose a simple and concrete audit procedure. Ballots are counted in batches selected using simple random selection with replacement. From all of the ballots selected, we construct an empirical distribution $\hat{M}$ of the ballots sampled and compute a measure of the discrepancy between $\hat{M}$ and all possible (real) distributions M of ballots for which the reported outcome is incorrect. Based on this discrepancy measure, we either stop the audit and confirm the result or we count another batch.

Using data from the 2008 Minnesota Senate race recount, we are able to construct synthetic datasets with realistic rates of optical scan errors, on which we evaluate our scheme by simulations. The results are promising and highlight the power of ballot-based auditing and of convex optimization. However, the simulations also show that our analysis is far too conservative: With our proposed parametrization, our observed miscertification rate is much less than α, which conversely means that we count more ballots than are necessary to certify elections. We believe that our analysis can be improved, making possible an even more efficient audit procedure.

Related work. In 1975, Roy Saltman [21] proposed a method by which one could gain confidence in an election's outcome. This problem did not receive much attention for several decades. In recent years interest has been renewed and the problem of providing strong guarantees for the outcome's correctness has been studied along two orthogonal axes.

The first axis concerns exactly what an audit that confirms the reported outcome guarantees. The earlier work focused on finding evidence of a single miscounted vote (see [10] and the references therein). If no evidence is found after counting some specified number of ballots, then the outcome is correct with high probability. Unfortunately, in any election using paper ballots and optical-scan hardware, some ballots will be miscounted and once a single miscount has been discovered, these audit procedures provide no guarantee about the correctness of the reported outcome.

In contrast to finding a single error, Stark [22] proposed the first complete audit procedure that specifies what to do when miscounts are discovered. Rather than being concerned with finding evidence of a single miscount, Stark's procedure looks for evidence that the reported outcome is incorrect — a so-called material error. Follow up work produced procedures that are easier to follow and statistically more powerful. As would be expected, Stark's procedures require significantly more ballots to be counted than the earlier work focused on finding a single error.

The second axis of study concerns the size of each sample to be audited. Most auditing procedures operate at the granularity of a precinct as that is the granularity at which most results are tabulated. The traditional organization of elections into precincts makes this a natural model. Neff [18], Johnson [15], Calandrino et al. [4], and Sturton et al. [27] note that the statistical power of post-election audits would be greatly increased by reducing the unit of an audit to a single ballot. (Intermediate sub-precinct audit units, such as individual voting machines, appear to provide no such gain in statistical power.) The downside to ballot-based auditing is that, to perform it, one needs a way to associate an electronic record of a ballot — the cast vote record (CVR) — with the physical ballot, for example, by printing a unique serial number on each ballot as they are being counted [4] or by weighing stacks of ballots [27].

The efficiency of any ballot-based auditing scheme depends on being able to efficiently select arbitrary bal-

lots to count based on the CVRs selected by the auditing algorithm. This issue has been studied by Calandrino et al. [4] and Sturton et al. [27]. However, in the absence of real-world experience with single-ballot methods, it is unclear how expensive finding each ballot will be in practice.

The first attempt at an error-tolerant, risk-limiting audit scheme that proceeds in stages was described by Johnson [15]. Unfortunately, Johnson's analysis is not truly risk limiting when multiple stages are used. For a discussion of this issue in the context of our approach, see Section 3.3.

Concurrently with our work, Stark proposes another ballot-based auditing scheme [26]. Stark's scheme uses different mathematical tools than ours and is not directly comparable. However, for similar risk guarantees, his approach appears to use fewer ballots to certify some elections which contain few errors. Nevertheless, we believe that with a less conservative analysis of our risk (see Section 3), our approach will be statistically more powerful (see Section 4.2).

Assumptions and limitations. Because in this paper we study a scenario considerably different from those studied earlier, it is worth stating the basic assumptions underlying our mathematical model and algorithm.

- We assume that each counted ballot has an associated index and for any index, we can efficiently retrieve and examine both the CVR and the physical ballot. In particular, we assume that we can sample a ballot *uniformly from all ballots cast in the election*.
- We assume that examining a ballot reveals perfectly the actual value of the ballot.

We emphasize that at present no deployed voting system is configured to support ballot-based auditing. Making use of our audit techniques would require substantial changes to election procedures. We view our results as an additional demonstration of the statistical power of ballot-based auditing compared to traditional precinct-based auditing, and hope that this demonstration will spur the development and deployment of voting systems that support ballot based auditing (cf. [4, 27]).

With each ballot having an index, sampling consists of drawing random entries from a table and then finding the corresponding CVR and paper ballot. Drawing random entries from a table is a straight-forward procedure using any random number generator, but see Cordero et al. [6], Hall [12], Calandrino et al. [5], Rescorla [20], and Heninger [13] for caveats on specific methods.

A drawback of our scheme is that although the convex optimization computation at its heart can be efficiently carried out, it is opaque from the viewpoint of election observers. Given the data from a sample count of ballots, the computation is deterministic, and so can be carried out on multiple computers. Thus, a transcript of the computations performed during an audit would enable outside observers to verify that the computation proceeded as intended.

We chose to implement our auditing scheme in MATLAB because of MATLAB's excellent support for numerical computation and its mature Optimization Toolbox.[1] The downside to our choice is that MATLAB is closed-source and expensive, making our prototype implementation less immediately useful to voting officials. We believe that it is also possible to implement the required algorithms in open-source, freely-available software for numerical computation such as GNU Octave or even more general math software such as Sage. It should also be possible to write a customized procedure for performing the specific minimization problem in our algorithm using standard libraries.

Any implementation used for a real election audit should include a transcript of its computation for independent verification that the procedure was correctly followed. A discussion of this issue is outside the scope of this paper.

2 A mathematical model for single-ballot sampling

We consider a slightly simplified model for sequentially auditing ballots after an election. We consider elections of the form "vote for 1 candidate." To model a vote for no candidate we introduce a fake candidate. Most models consider election auditing by precinct, in which entire precincts are counted and the sums are compared to the reported outcomes from that precinct. Here we pool all ballots cast into one collection and audit batches of ballots sampled uniformly from this combined pool. This procedure allows us to use classical tools from probability theory to analyze the results of the audit.

Mathematically, we model a set of C **candidates** as a set $\mathcal{X} = \{0, 1, 2, \ldots, C\}$ with a ballot cast for 0 signifying no vote or "white ballot." We focus on the case of $C = 2$, which is a contest between two candidates. A description of how to extend our results to more candidates is given in Section 5. In the course of the election, n ballots are cast. We model the n ballots as a set $X^n = \{X_1, X_2, \ldots, X_n\}$ where for each $i \leq n$, the variable X_i takes values in $\mathcal{X}$. Thus $X_i = 2$ corresponds to the ith ballot being a vote for Candidate 2, and $X_i = 0$ corresponds to a vote for no candidate.

The **true outcome** of the election is the fractions of the ballots in X^n that were cast for the different candidates. We write the fractions as a $C+1$-dimensional vec-

Table 1: Notation

Symbol	Meaning
α	maximum probability of miscertifying an election (risk level)
γ	upper bound on the probability of error for each stage
Δ_t	smallest value of divergence between $\hat{M}_t$ and any $R \in \mathcal{R}$
ξ_t	per-round threshold parameter
$b(\alpha, m)$	naïve bound on the number of ballots needed to detect one miscount
$\mathcal{B}_t$	set of ballots sampled in the batch t
C	number of candidates
$\mathcal{D}_x$	set of outcomes for which x is the winner
k_t	number of ballots in a batch t
K_t	number of ballots sampled through batch t: $K_t = \sum_{j=1}^{t} k_j$
M	empirical joint distribution of ballots
$\hat{M}_t$	empirical joint distribution of sampled ballots
p	true fractions of votes for each candidate
q	reported fractions of votes for each candidate
$\mathcal{R}$	set of joint distributions where the true winner is not the reported winner
T	maximum number of batches to be sampled
$\mathcal{X}$	set of candidates (including null candidate)
X_i	true value of ith ballot
Y_i	recorded value of ith ballot

tor $p = (p(0), p(1), p(2))$ and refer to p as the **empirical distribution** of X^n. Letting $\mathbf{1}(X_i = x) = 1$ when $X_i = x$ and 0 when $X_i \neq x$, we can write the following equation for $p(x)$:

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(X_i = x). \tag{1}$$

The **true winner** of the election — the candidate who won the most votes — is

$$w_{\text{true}}(p) = \operatorname*{argmax}_{x \in \mathcal{X} \setminus \{0\}} p(x). \tag{2}$$

Unfortunately, errors (potentially caused by fraud) occur when the ballots are counted, and the initial election results may not reflect the true outcome p. We say the ballot i is recorded as a variable Y_i that also takes values in $\mathcal{X}$, where Y_i may be different from X_i. The **reported outcome** of the election is the empirical distribution of the set $Y^n = \{Y_1, Y_2, \ldots, Y_n\}$ given by

$$q(y) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(Y_i = y). \tag{3}$$

The **reported winner** of the election is

$$w_{\text{reported}}(q) = \operatorname*{argmax}_{y \in \mathcal{X} \setminus \{0\}} q(y). \tag{4}$$

An **audit** is a procedure for sampling the ballots, recording the true values, and deciding whether the reported winner w_{reported} is the same as the true winner w_{true}, or whether a full hand-count of the ballots is required to decide this fact. If the auditor decides that $w_{\text{reported}} = w_{\text{true}}$, we say that she **certifies** the reported outcome. Suppose the audit samples some set $A \subseteq \{1, 2, \ldots, n\}$ of the ballots. We assume that an auditor looking at ballot i can determine X_i perfectly; Y_i is precisely the CVR and thus is already known. The auditor has to decide on the basis of $\{(X_i, Y_i) : i \in A\}$ whether or not to certify the election.

We are interested in **risk-limiting audits**. In a risk-limiting audit with risk level α, ballots are sampled randomly and we have the guarantee: If $w_{\text{reported}} \neq w_{\text{true}}$, then the audit will require a full hand-count with probability (over the choice of the sample) at least $1 - \alpha$. Another way of phrasing this is that the audit is conservative in the following sense: If the true result of the outcome is different than the reported outcome, then the probability that the audit certifies the election is smaller than α. Setting the value α is a policy question; values between 1% and 25% have been studied in prior work. The number of ballots to be sampled depends on the parameters α and C, and on the reported outcome q of the election.

To see why the number of ballots required depends on q, consider the following two scenarios involving two candidates. In the first, the reported outcomes are 20% for Candidate 1 and 80% for Candidate 2, and in the second, they are 49.9% for Candidate 1 and 50.1% for Candidate 2. In the first case, in order for the winner to not be Candidate 2, there must have been massive irregularities in the counting, so that $X_i \neq Y_i$ for more than 30% of the votes. Even a small subset of the ballots sampled

would show that the true values X_i are different from Y_i. However, if even a small sample shows very little irregularity, the auditor can be quite certain that a full count would still result in Candidate 2 winning. In the second scenario, even a small number of irregularities could result in the election flipping. An auditor would require a larger sample to have the same certainty Candidate 2 was the true winner.

The auditor's decision whether to certify the election depends on the values reported by the audit. We therefore consider a **sequential auditing** scheme that samples additional ballots until the auditor is assured that the risk of miscertification is low [23]. The audit operates in steps, which we index by $t = 1, 2, \ldots, T$. At the tth step, the auditor samples a **batch** of k_t ballots $\mathcal{B}_t \subset \{1, 2, \ldots, n\}$, with replacement,[2] and computes a test statistic using all of the ballots $\mathcal{A}_t = \bigcup_{i=0}^{t} \mathcal{B}_i$ audited so far. This statistic is used to bound the probability that the reported outcome is incorrect; the auditor compares this statistic to a threshold chosen as a function of α, the desired risk level. If the test statistic exceeds the threshold it certifies the election; if not, it moves to step $t+1$ and samples another batch of ballots.

Our auditing scheme is based on estimating the **empirical joint distribution** of the true and reported ballot values. This is a $(C+1)$-by-$(C+1)$ matrix M in which the element $M(x,y)$ located in the xth row and yth column is the fraction of the total ballots for which the true vote was $X_i = x$ but it was recorded as $Y_i = y$. The true joint distribution is therefore

$$M(x,y) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(X_i = x, Y_i = y). \tag{5}$$

The **marginal distributions** on X and Y are p and q respectively:

$$p(x) = \sum_{y \in \mathcal{X}} M(x,y), \tag{6}$$

$$q(y) = \sum_{x \in \mathcal{X}} M(x,y). \tag{7}$$

We would like to use the evidence from the audit to estimate the true joint distribution M. For simplicity, consider the tth step of the audit so that $K_t = \sum_{j=1}^{t} k_j$ ballots have been sampled thus far. From the t batches sampled we can construct an estimate $\hat{M}_t$ of M by

$$\hat{M}_t(x,y) = \frac{1}{K_t} \sum_{j=0}^{t} \sum_{i \in \mathcal{B}_j} \mathbf{1}(X_i = x, Y_i = y). \tag{8}$$

This is the empirical joint distribution of the sampled ballots. The double summation emphasizes that in the rare event that the same ballot is sampled in $\mathcal{B}_j$ and $\mathcal{B}_{j'}$, it is counted in $\hat{M}_t$ for each time it is sampled.

The last part we need is to model the assumption that the winner of the election is different than reported. We illustrate this for the simple two-candidate election with $\mathcal{X} = \{0, 1, 2\}$, and describe the model for more candidates in Section 5. The winner of the election is different than reported if the true outcome p has a winner $w_{\text{true}}(p)$ not equal to $w_{\text{reported}}(q)$. Let $\mathcal{D}_c$ be the set of distributions d such that the winner of the election is c — that is, the set of potential values for p. For example, in a two-candidate election, we have

$$\mathcal{D}_2 = \left\{ \begin{array}{l} d(x) \geq 0 \ \ \forall x \in \mathcal{X}, \\ d : d(0) + d(1) + d(2) = 1, \\ d(2) > d(1) \end{array} \right\}. \tag{9}$$

The first two conditions say that d is a probability distribution and the third says that Candidate 2 is the winner.

Suppose that $w_{\text{reported}}(q) = 1$. Then, since $\mathcal{D}_2$ contains the set of vote distributions for which the true winner was Candidate 2, the set of possible joint distributions such that the reported outcome is q but $w_{\text{true}}(p) = 2$ is

$$\mathcal{R} = \left\{ \begin{array}{ll} R(x,y) \geq 0 \ \ \forall x,y \in \mathcal{X}, & \displaystyle\sum_{x,y \in \mathcal{X}} R(x,y) = 1, \\ R : & \\ \displaystyle\sum_{x \in \mathcal{X}} R(x,y) = q(y), & \displaystyle\sum_{y \in \mathcal{X}} R(x,y) \in \mathcal{D}_2 \end{array} \right\}. \tag{10}$$

This is the set of joint distributions on (X,Y) pairs such that the Y-marginal agrees with the reported values q and the X-marginal does not agree with the reported outcome. The set $\mathcal{R}$ represents all possible values for the true underlying distribution M defined in (5) such that the reported winner was Candidate 2 but the true winner was Candidate 1.

We can phrase the auditing criteria mathematically using the notation we have just defined. If the true joint distribution $M \in \mathcal{R}$, then our auditing procedure should result in a full hand count with probability at least $1 - \alpha$. If the true joint distribution $M \notin \mathcal{R}$ then the outcome of the election is correct, and we would like a test which uses as few ballots as possible to determine this.

3 An algorithm for ballot-based auditing

Since our auditing algorithm is risk-limiting, we need to control the probability of certification when the reported outcome of the election is wrong. In the notation of the previous section, we want

$$\mathbb{P}(\text{certify} \mid M \in \mathcal{R}) < \alpha. \tag{11}$$

For a sequence of values $\mathbf{z} = (z_1, z_2, \ldots, z_K) \in \mathcal{Z}^K$ from a finite set $\mathcal{Z}$, define the **type** of $\mathbf{z}$ as the probability distribution

$$P_{\mathbf{z}}(z) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}(z_i = z). \tag{12}$$

Algorithm A Sequential auditing procedure

Given: reported outcome q, parameters α, C, T, and $k_1, k_2, \ldots, k_T$.
Output: true winner w.
$\mathit{Certify} \leftarrow 0$, $t \leftarrow 0$, $\gamma \leftarrow 1-(1-\alpha)^{1/T}$.
$K_t \leftarrow \sum_{j=1}^{t} k_j$ for $0 \le t \le T$.
$\hat{M}_0(x,y) \leftarrow 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{X}$.
while $\mathit{Certify} = 0$ and $t \le T$ **do**
  $t \leftarrow t+1$.
  Draw k_t indices $\mathcal{B}_t$ with replacement uniformly from $\{1,2,\ldots,n\}$.
  $\hat{M}_t(x,y) \leftarrow (K_{t-1}/K_t)\hat{M}_{t-1}(x,y) + (1/K_t)\sum_{i\in\mathcal{B}_t} \mathbf{1}(X_i = x, Y_i = y)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{X}$.
  $\Delta \leftarrow \min_{R\in\mathcal{R}} D(\hat{M} \parallel R)$.
  Compute ξ per Equation (36).
  if $\Delta > (1/K_t)\log(f(\hat{M})/\xi)$ **then**
    $\mathit{Certify} \leftarrow 1$.
  end if
end while
if $\mathit{Certify} = 1$ **then**
  Output $w \leftarrow w_{\text{reported}}$.
else
  Count all ballots and compute true distribution $M \leftarrow (1/n)\sum_{i=1}^{n} \mathbf{1}(X_i = x, Y_i = y)$.
  Output $w \leftarrow \text{argmax}_x \sum_{y\in\mathcal{X}} M(x,y)$.
end if

For our purposes, $\mathcal{Z} = \mathcal{X} \times \mathcal{X}$, the set of pairs of possible actual and reported votes for a ballot. Thus, if all of the ballots cast in an election are arranged in some sequence of (x_i, y_i) pairs, then the empirical joint distribution M is the type of the sequence. Using $\mathcal{Z}$ instead of $\mathcal{X} \times \mathcal{X}$ simplifies the notation in much of what follows and will be used frequently.

Let $\mathcal{P}_K$ be the set of types with denominator K — that is, for every $P \in \mathcal{P}_K$ and every $z \in \mathcal{Z}$, the probability $P(z)$ is an integer multiple of $1/K$ — then for any $P \in \mathcal{P}_K$, define the **type class** of P to be the set of sequences of length K with type P:

$$T(P) = \{\mathbf{z} \in \mathcal{Z}^K \ : \ P_{\mathbf{z}} = P\}. \tag{13}$$

As a final piece of notation, for a probability distribution R, and a sequence $\mathbf{z} \in \mathcal{Z}^K$ drawn i.i.d., let $R^K(\mathbf{z}) = \prod_{i=1}^{K} R(z_i)$ be the probability of drawing the sequence $\mathbf{z}$ according to the distribution R.

If we consider a round of the auditing algorithm as a decision procedure $A_t(\hat{M}) \in \{0,1\}$ such that $A_t(\hat{M}) = 1$ if it certifies the election after drawing a sample of ballots of size K_t with type $\hat{M}_t$, then the probability of this round certifying is

$$\mathbb{P}(\text{certify} \mid M \in \mathcal{R}) = \sum_{\hat{M}_t} M^{K_t}\big(T(\hat{M}_t)\big) \cdot A_t(\hat{M}_t) \tag{14}$$

$$= \sum_{\hat{M}_t \in \mathcal{C}_t} M^{K_t}\big(T(\hat{M}_t)\big), \tag{15}$$

where $\mathcal{C}_t$ is the set of types on which A_t certifies (independent of the behavior of the other rounds).

The sum in (15) is impossible to calculate directly since we do not know M — at least not without counting all of the ballots. Instead, we can bound the sum in two steps. The first step is to get a bound on $M^{K_t}\big(T(\hat{M}_t)\big)$ given that M is in $\mathcal{R}$. The second step is to slice the certification region $\mathcal{C}_t$ into two pieces, compute bounds on their sizes, and compute bounds on the probability of sampling from each of the regions.

Bounding the probability of an audit sample. For any known probability distribution R, a standard result in information theory [7, Theorem 11.1.2] tells us we can compute the probability of any sequence $\mathbf{z}$:

$$R^K(\mathbf{z}) = \exp\big(-K[D(P_{\mathbf{z}} \parallel R) + H(P_{\mathbf{z}})]\big). \tag{16}$$

The first term in the exponent is the Kullback-Leibler (KL) divergence, which is defined as

$$D(P \parallel R) = \sum_{z\in\mathcal{Z}} P(z)\log\frac{P(z)}{R(z)}, \tag{17}$$

where the logarithm is base e. The KL-divergence is a measure of how close two distributions are,[3] and appears frequently in the literature on hypothesis testing. If P and R are far apart, so that $\sum_z |P(z) - R(z)|$ is large, then $D(P \parallel R)$ is large as well. The second term in the exponent is the standard notion of Shannon entropy of a distribution

$$H(P) = \sum_{z\in\mathcal{Z}} P(z)\log\frac{1}{P(z)}. \tag{18}$$

By summing over all elements in the type class of $P \in \mathcal{P}_K$, we can compute the probability of $T(P)$,

$$\begin{aligned} R^K\big(T(P)\big) &= |T(P)| \exp\big(-K[D(P \parallel R) + H(P)]\big) \\ &= f(P) \exp\big(-K \cdot D(P \parallel R)\big) \end{aligned} \tag{19}$$

where

$$f(P) = |T(P)| \exp\big(-K \cdot H(P)\big) \leq 1. \tag{20}$$

The size of the type class is a simple multinomial coefficient

$$|T(P)| = \binom{K}{K \cdot P(z_1),\ \ldots,\ K \cdot P(z_{|\mathcal{Z}|})} \tag{21}$$

where $\{z_1, z_2, \ldots, z_{|\mathcal{Z}|}\} = \mathcal{Z}$.

Since we are trying to bound the chance of certification given $M_t \in \mathcal{R}$, we can evaluate $R^{K_t}\big(T(\hat{M}_t)\big)$ exactly for each $R \in \mathcal{R}$. To be conservative, we want to find the largest probability over all possible true vote distributions $R \in \mathcal{R}$. This corresponds to finding the smallest value of the KL-divergence $D(\hat{M}_t \parallel R)$ over $R \in \mathcal{R}$:

$$\Delta_t = \min_{R \in \mathcal{R}} D(\hat{M}_t \parallel R). \tag{22}$$

The minimum value can be computed using any constrained convex optimization routine such as MATLAB's `fmincon` function; see Appendix A.

Therefore, when $M \in \mathcal{R}$,

$$M^{K_t}\big(T(\hat{M}_t)\big) \leq f(\hat{M}_t) \exp(-K_t \Delta_t). \tag{23}$$

This forms the basis of our certification test: If

$$f(\hat{M}_t) \cdot \exp(-K_t \Delta_t) < \xi_t, \tag{24}$$

for some ξ_t to be determined later, then the algorithm certifies the election.

Slicing the certification region. The second step to bounding the sum in (15) is to bound the size of the certification region $\mathcal{C}_t$. Unlike the bounds in the previous step — which are as tight as they can be up to neglecting the discrete nature of $\mathcal{R}$ in the optimization — the bound in the second step is *extremely* loose. This looseness represents the major opportunity for improving on our results. See Section 4.2 for evidence of just how much improvement potentially remains.

Recall that $\mathcal{Z} = \mathcal{X} \times \mathcal{X}$. One very simple bound on the size of $\mathcal{C}$ is the number of types with denominator K_t,

$$|\mathcal{P}_{K_t}| = \binom{K_t + |\mathcal{Z}| - 1}{|\mathcal{Z}| - 1}. \tag{25}$$

With this, we can bound the sum in (15) by

$$\mathbb{P}(\text{certify} \mid M \in \mathcal{R}) \leq |\mathcal{P}_{K_t}| f(\hat{M}_t) \exp(-K_t \Delta_t). \tag{26}$$

This bound is perfectly valid and an audit procedure that works can be constructed around it, but we can do better.

Pinsker's inequality [7], another standard result from information theory, states that the l_1 distance between two distributions $\|P - R\|_1 = \sum_{z \in \mathcal{Z}} |P(z) - R(z)|$ is related to the KL-divergence by

$$D(P \parallel R) \geq \frac{1}{2} \|P - R\|_1^2. \tag{27}$$

This justifies our assertion earlier assertion that if P and R are far apart then the divergence is large. We can use Pinsker's inequality to partition the certification region $\mathcal{C}_t$ into two pieces, $\mathcal{C}_t^1$ and $\mathcal{C}_t^2$.

Let $\mathcal{G}(\delta) = \{P \in \mathcal{P}_{K_t} \ : \ \|P - M\|_1 \leq \delta\}$. For any $\hat{M}_t \in \mathcal{G}(\delta_t)$, either the algorithm does not certify, or $f(\hat{M}) \exp(-K_t \Delta_t) < \xi_t$. Note that $|\mathcal{G}(\delta_t)|$ is an upper bound on the size of $\mathcal{C}_t^1 = \mathcal{C}_t \cap \mathcal{G}(\delta_t)$ — the set of distributions on which the algorithm certifies and has l_1 norm at most δ_t from M. We can bound the size of $\mathcal{G}(\delta_t)$ by a volume argument (see Appendix B) to get

$$|\mathcal{G}(\delta_t)| \leq \frac{(2\delta_t K_t + 2|\mathcal{Z}|)^{|\mathcal{Z}|-1}}{(|\mathcal{Z}|-1)!}. \tag{28}$$

Let $\mathcal{C}_t^2 = \mathcal{C}_t \setminus \mathcal{C}_t^1 \subset \mathcal{G}(\delta_t)^C$. Then for any distribution $\hat{M}_t \notin \mathcal{G}(\delta_t)$, we have

$$\delta_t < \|\hat{M} - M\|_1 \leq \sqrt{2D(\hat{M} \parallel M)}. \tag{29}$$

and thus

$$f(\hat{M}_t) \exp\big(-K_t \cdot D(\hat{M}_t \parallel M)\big) < \exp(-K\delta_t^2/2). \tag{30}$$

As before, we can bound the size of $\mathcal{C}_t^2$ by $|\mathcal{P}_{K_t}|$. This is, of course, a massive overstatement of the size of $\mathcal{C}_t^2$, but each distribution in $\mathcal{C}_t^2$ has an exponentially small probability and our size bound is polynomial so the contribution to the risk from $\mathcal{C}_t^2$ can be controlled with a reasonable choice of δ_t.

Putting this all together, the risk r_t of miscertification in round t is bounded by

$$\begin{aligned} r_t &= \mathbb{P}(\text{certify} \mid M \in \mathcal{R}) && (31) \\ &= \sum_{\hat{M}_t \in \mathcal{C}_t^1} M^{K_t}\big(T(\hat{M}_t)\big) + \sum_{\hat{M}_t \in \mathcal{C}_t^2} M^{K_t}\big(T(\hat{M}_t)\big) && (32) \\ &< \xi_t |\mathcal{C}_t^1| + e^{-K\delta_t^2/2} |\mathcal{C}_t^2| && (33) \\ &< \xi_t \cdot \frac{(2\delta_t K_t + 2|\mathcal{Z}|)^{|\mathcal{Z}|-1}}{(|\mathcal{Z}|-1)!} + e^{-K_t \delta_t^2/2} |\mathcal{P}_{K_t}|. && (34) \end{aligned}$$

We want the maximum risk of miscertification in each round to be at most a constant γ so, for a given number of ballots K_t sampled up to round t, we need to pick parameters ξ_t and δ_t such that $r_t < \gamma$. One way to do this is

to let the second term in (34) be equal to some ε-fraction of γ. Solving for δ_t, we get

$$\delta_t = \sqrt{\frac{2}{K_t} \log \frac{|\mathcal{P}_{K_t}|}{\varepsilon\gamma}}. \tag{35}$$

Now we set the first term in (34) equal to $(1-\varepsilon)\gamma$ and solve for ξ_t:

$$\xi_t = \frac{(1-\varepsilon)\gamma(|\mathcal{Z}|-1)!}{(2\delta_t K_t + 2|\mathcal{Z}|)^{|\mathcal{Z}|-1}}. \tag{36}$$

Plugging (36) into our certification test gives us our per-round test statistic:

$$\Delta_t > \frac{1}{K_t} \log \frac{f(\hat{M}_t)}{\xi_t}. \tag{37}$$

Thus far we have described an algorithm which takes batches of size k_t and after each batch computes an estimate $\hat{M}_t$. The last piece is to ensure a full hand-count if the audit cannot certify the election. We set a number T and say that if T total batches have been audited and (24) (equivalently, (37)) has still not been satisfied, then we do a full hand-count of all of the ballots. Note that under our assumptions, a full hand-count reveals the true outcomes p of the election.

3.1 A naïve bound

A simple lower bound on the number of ballots that need to be drawn in order to detect at least one ballot that has an error, given that the election outcome is incorrect, depends only on the margin m (as a fraction of n). Since the presence of an $m/2$ fraction of votes that change the margin by two is enough to change the outcome of the election, the probability of sampling b ballots (with replacement) and not seeing any errors is $(1-m/2)^b$. We can set this equal to α and solve for b to determine the minimum number of ballots any algorithm *searching for errors* with a risk level α must sample if no errors are detected:

$$b(\alpha, m) = \frac{\log \alpha}{\log(1-m/2)}. \tag{38}$$

We call $b(\alpha, m)$ the naïve bound.

Since this bound is computed by throwing away information, in principle, it is possible for an auditing algorithm to sample fewer than $b = b(\alpha, m)$ ballots and certify the election with risk level α. Despite this, in the normal case, an auditing algorithm will require more ballots so b is a useful reference point when setting the batch sizes.

3.2 Computing the threshold statistics

It is important in a sequential scheme that the statistics be efficiently computable at each time step. Computing $\hat{M}_t$ is simple, since it just involves updating the counts. Computing δ_t and $\log(f(\hat{M}_t)/\xi_t)$ is straight forward.

We must show that Δ_t in (22) has a unique minimum and is efficiently computable. We need two facts: firstly, that the set $\mathcal{R}$ is convex, and secondly, that $D(\hat{M} \parallel R)$ is a convex function in R. The first fact follows directly from the definition, since $\mathcal{R}$ is defined by a set of linear constraints. The second fact is standard [8, p. 50]. Therefore the threshold value Δ_t is efficiently computable because it involves minimizing a convex function over a convex set, which can be done by standard techniques; see Appendix A. For two-candidate elections it is a 9-dimensional problem and the minimization takes a less than a second in MATLAB using built-in functions. The complexity scales quadratically with the number of candidates for a single election. For auditing multiple contests, further evaluation is needed, but modern software systems routinely handle hundreds of such larger-scale optimization problems in under a minute. For context, the time spent counting the ballots to use in the audit dwarfs the computation time by orders of magnitude.

3.3 Setting the parameters

The procedure described above has several parameters which must be set in order to implement our procedure. The most important of these parameters is the threshold γ used in (36). Our bound says that that if (37) is satisfied, then the sample $\hat{M}_t$ occurs with probability less than γ for every distribution in $\mathcal{R}$. Thus with probability γ, we make the wrong decision. Since we are sampling up to T times, the chance that our procedure results in a full hand-count when the reported winner is incorrect is at least $(1-\gamma)^T$ (see Stark [23]). By setting $1-\alpha = (1-\gamma)^T$ and solving for γ, we get a conservative setting

$$\gamma = 1 - (1-\alpha)^{1/T}. \tag{39}$$

The other parameters that we must set are the batch sizes k_t, the maximum number of total batches T, and the fraction ε. For concreteness, in the remainder of the paper, we set $\varepsilon = 0.01$ and $T = 5$. Since we desire not to count too many ballots yet at the same time count enough that we can certify most correct elections without a full hand count, we let $k_1 = 9b$ and $k_t = b$ for $t > 1$ where $b = b(\alpha, m)$ is the naïve bound given in Section 3.1; thus $K_t = (9+t)b$. In this way, we count at most $13b$ ballots before certifying the election or going to a full hand count. With a tighter analysis of the auditing algorithm, the number of ballots counted could be reduced, perhaps significantly reduced; see Section 4.2.

Our complete scheme is Algorithm A.

3.4 Example

We illustrate our algorithm via a numerical example. Suppose we have an election with 100,000 votes cast, and the true and reported votes were according to the following table.

	reported vote		
true vote	None	Candidate 1	Candidate 2
None	1500	300	600
Candidate 1	400	46300	600
Candidate 2	100	200	50000

That is, 600 votes for Candidate 1 were reported for Candidate 2, 200 votes for Candidate 2 were reported for Candidate 1, 100 votes for Candidate 2 were reported as blank, and so on.

Dividing each element of this table by the sum, we obtain the true joint distribution of the election,

$$M = \begin{pmatrix} 0.015 & 0.003 & 0.006 \\ 0.004 & 0.463 & 0.006 \\ 0.001 & 0.002 & 0.500 \end{pmatrix}. \tag{40}$$

If we sum down each column of M we get the reported outcomes q and if we sum along each row we get the true outcomes p:

$$q = \begin{pmatrix} 0.020 & 0.468 & 0.512 \end{pmatrix}, \tag{41}$$

$$p = \begin{pmatrix} 0.024 & 0.473 & 0.503 \end{pmatrix}. \tag{42}$$

From the reported outcome q the winner was Candidate 2 by a 4.4% relative margin, but the true margin is 3.0%. So the outcome of the election is correct, but by a smaller amount than reported.

We can construct the set $\mathcal{D}_1$ of possible true outcomes for which Candidate 1 is the winner, as in (9). This is the set of $d = (d(0), d(1), d(2))$ where each entry is nonnegative, they add up to 1, and $d(1) > d(2)$. Then, we can write the set of joint distributions as in (10). This is the set of all matrices with nonnegative entries:

$$R = \begin{pmatrix} R(0,0) & R(0,1) & R(0,2) \\ R(1,0) & R(1,1) & R(1,2) \\ R(2,0) & R(2,1) & R(2,2) \end{pmatrix}, \tag{43}$$

such that summing each column is equal to q and summing each row is something in $\mathcal{D}_1$.

Suppose we want a risk of $\alpha = 0.01$, then if we run the algorithm for up to $T = 5$ rounds, (39) says we should set

$$\gamma = 1 - (1 - 0.01)^{1/5} \approx 0.0020. \tag{44}$$

From α and the reported margin $m = 4.4\%$, we can compute the naïve bound

$$b(1\%, 4.4\%) = \left\lceil \frac{\log(0.01)}{\log(1 - 0.044/2)} \right\rceil = 208. \tag{45}$$

Table 2: Sequence of example test statistic values.

t	K_t	$\log \frac{1}{\xi_t}$	Δ_t	$\frac{1}{K_t} \log \frac{f(\hat{M}_t)}{\xi_t}$
1	1872	50.63	0.0079	0.0184
2	2080	51.10	0.0086	0.0166
3	2288	51.53	0.0064	0.0151
4	2496	51.91	0.0257	0.0136

The five batch sizes are $k_1 = 1872$ and $k_t = 208$ for $t > 1$.

We begin by randomly sampling 1872 ballots for the first batch ($t = 1$) and calculate the empirical distribution $\hat{M}_1$ according to (8). Suppose we measure the following counts.

	reported vote		
true vote	None	Candidate 1	Candidate 2
None	39	6	19
Candidate 1	6	840	8
Candidate 2	3	0	951

Dividing by 1872 gives,

$$\hat{M}_1 = \begin{pmatrix} 0.0208 & 0.0032 & 0.0101 \\ 0.0032 & 0.4487 & 0.0043 \\ 0.0016 & 0.0000 & 0.5080 \end{pmatrix}. \tag{46}$$

Next, we calculate Δ_1 per (22), which is the minimum of $D(\hat{M}_1 \parallel R)$ over all $R \in \mathcal{R}$. We can find the minimizing R numerically using standard optimization tools; see Appendix A. The minimum is attained at $R = R_*$:

$$R_* = \begin{pmatrix} 0.0157 & 0.0024 & 0.0151 \\ 0.0034 & 0.4656 & 0.0144 \\ 0.0009 & 0.0000 & 0.4825 \end{pmatrix}, \tag{47}$$

and $\Delta_1 = 0.0079$, which is smaller than the threshold $(1/1872)\log(f(\hat{M}_1)/\xi_t) = 0.0184$. Therefore, we cannot certify the election yet and we draw another batch of ballots.

Continuing in this way, we get a sequence of Δ_t and $(1/K_t)\log(f(\hat{M}_t)/\xi_t)$ given in Table 2. After 2496 ballots are counted, $\Delta_4 > (1/2496)\log(f(\hat{M}_4)/\xi_4)$ and the auditing procedure certifies the election.

4 Evaluation

We experimentally evaluate our algorithm by simulating elections with varying parameters. Our simulations show that the statistical bounds on the risk of miscertification actually hold in practice; in fact, they show that with this analysis, the risk of miscertification is essentially zero. This is a check that both our math and implementation are correct. We also demonstrate that, with the parameters given in Section 3.3, our algorithm has sufficient

statistical power to certify elections without a full hand count. This is required because an auditing algorithm without sufficient power to certify any election without a full hand count will never miscertify, but is of no practical value.

4.1 Validating miscertification bounds

A post-election, risk-limiting auditing algorithm has two competing goals: control the probability of certifying an election where the reported vote totals are incorrect and minimize the number of ballots counted when the reported outcome is correct. As the first goal is paramount, we design our algorithm to have strong guarantees that if the election is certified then either it is correct or an event with probability at most α occurred — that event being that we failed to find enough evidence that the outcome was incorrect.

To experimentally verify our results, we consider several elections with $n = 100{,}000$ ballots between two candidates in which the reported results are incorrect. Up to five rounds of counting occur ($T = 5$) and the batch sizes are as in Section 3.3 so that after 5 rounds $13b$ ballots have been counted where $b = b(\alpha, m)$ is the naïve bound (Section 3.1). The risk level is set to $\alpha = 1\%$ and we vary the margin m between 0.5% and 5%. To run simulations, we need *some* joint distribution of votes for which the reported winner is not the actual winner; to that end, we let

$$M = \begin{pmatrix} 0 & .4m & 0 \\ 0 & .5-.4m & 0 \\ .4m & .2m & .5-.6m \end{pmatrix}. \qquad (48)$$

An audit for each value of m is simulated 10,000 times. If our analysis were tight, we would expect to see roughly an $\alpha = 1\%$ fraction of miscertification for each election. Instead, we see that not a single election was miscertified. As we'll see shortly, this is because our analysis requires us to count far more ballots than should really be necessary. As a result, our sample $\hat{M}$ differs from the true distribution M by only a small amount with high probability and it is easy to distinguish a correct election from an incorrect election.

4.2 How conservative is our analysis?

In Section 3, we gave an extremely conservative risk analysis. As a result, we were forced to audit a larger number of ballots than if we had a tighter analysis. In this section, we give evidence that there is substantial room for improvement.

Recall that in round t, the algorithm will certify the election if $f(\hat{M}_t)\exp(-K_t\Delta_t) < \xi_t$. Thus, the greater ξ_t, the more likely the election will be certified. To that end, we simulate 100,000 independent rounds for each of the eight sample size and election combinations described below. To be risk-limiting at a risk level $\alpha = 1\%$ for a five round audit, we set the per-round risk to $\gamma \approx 0.002$ and expect to see a γ-fraction of miscertifications for the incorrect election with an appropriately chosen ξ_t.

Table 3: Values of $\hat{\xi}$ and $\hat{p}$ from Figure 1.

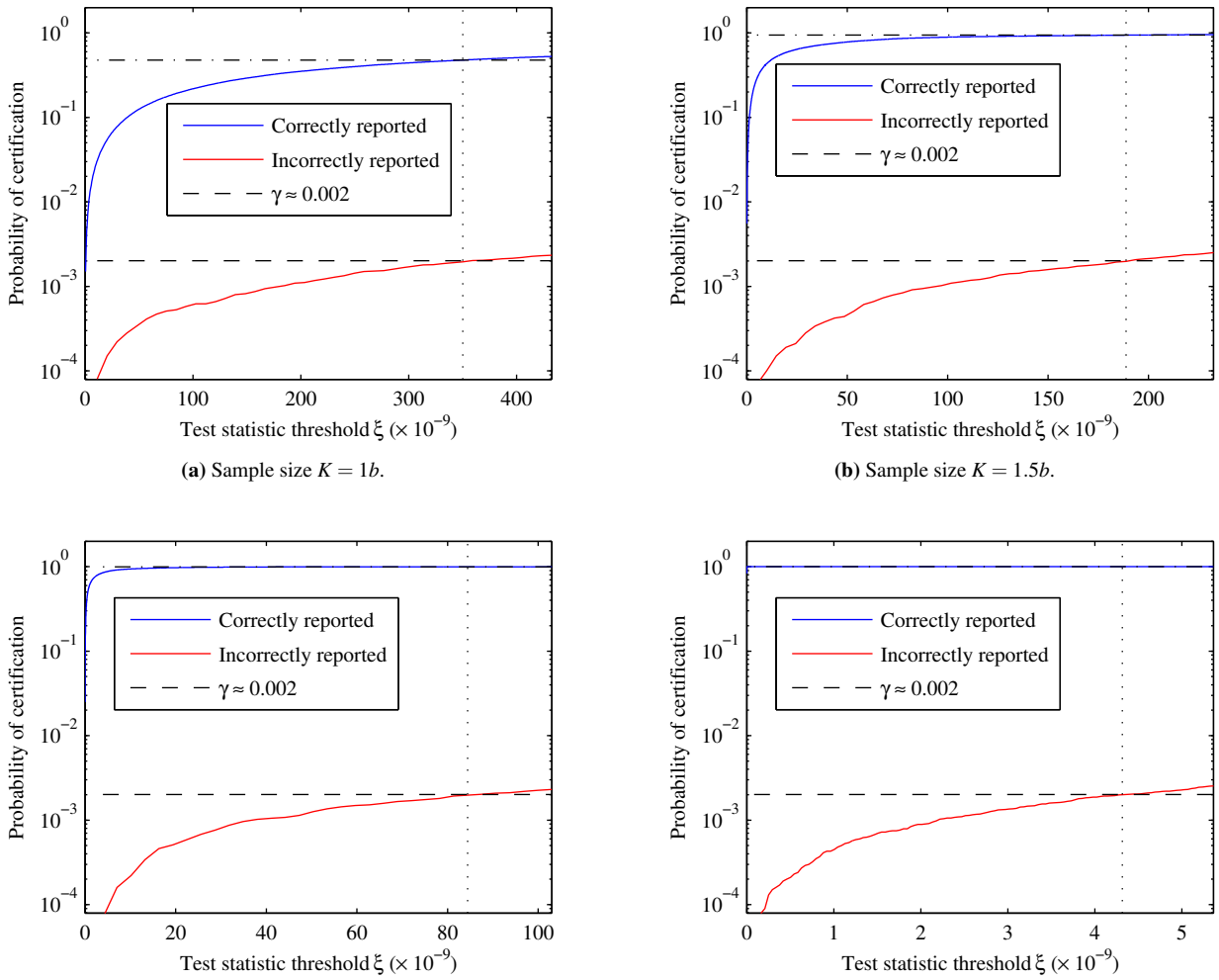
K	$\hat{\xi}$ ($\times 10^{-9}$)	$\hat{p}$ (%)
$1b$	350.07	47.72
$1.5b$	188.78	94.08
$2b$	84.41	99.76
$9b$	4.31	100.00

Figure 1 shows the measured probability of certifying two different elections — with a margin of 0.5% for four different sample sizes K: $1b$, $1.5b$, $2b$, and $9b$, where $b = b(1\%, 0.5\%)$ is the naïve bound — given the test statistic threshold value ξ. In each of the four figures, the top solid line is the measured probability of certifying an election where the reported outcome is correct: the election parameters are identical to the bidirectional errors condition in the next section. The bottom solid line is the measured probability of miscertifying an election in which the reported outcome is incorrect: the election parameters are identical to those of the previous section. The dashed line is the per-round risk level $\gamma \approx 0.002$ for a five round audit. The mixed dotted and dashed line ($\cdot - \cdot - \cdot$) is the measured probability $\hat{p}$ of certifying the correct election in this round, given a test statistic threshold of $\hat{\xi}$ — represented by the dotted, vertical line — corresponding to a per-round risk of γ. Table 3 shows the threshold $\hat{\xi}$ and probability $\hat{p}$ for each of the sample sizes.

Recall that Section 3.3 sets the initial batch size $k_1 = 9b$. Figure 1d and Table 3 show that the threshold $\hat{\xi}$ corresponding to a measured risk of miscertifying the second election with a sample of size $K = 9b$ is approximately $\hat{\xi} \approx 10^{-9}$. Using this threshold, the probability of certifying the first election 100%. For comparison, using our conservative analysis, we set $\xi_1 \approx 10^{-22}$.

These results suggest that sampling $9b$ ballots in the first round is far too many. Figures 1a, 1b, and 1c give evidence that a significant savings in terms of ballots counted can be gained by a tighter risk analysis.

One interesting side effect of our analysis being so conservative is that the risk level α can be made much smaller with very little change in the number of ballots required to certify correct elections. For example, changing from $\alpha = 1\%$ to $\alpha = 0.1\%$ requires essentially no change to the number of ballots that need to be counted. For comparison, Stark's procedure [26] requires twice as many ballots when moving from $\alpha = 1\%$ to $\alpha = 0.1\%$.



(a) Sample size $K = 1b$.

(b) Sample size $K = 1.5b$.

(c) Sample size $K = 2b$.

(d) Sample size $K = 9b$.

Figure 1: Measured certification rates for a round of auditing for a correctly reported election and an incorrectly reported election versus the threshold parameter ξ for four sample sizes.

4.3 Expected number of ballots counted

The second goal of a post-election, risk-limiting audit is to minimize the number of ballots counted when the outcome is correct — that is, to maximize the statistical power of our test. We simulate elections with correctly reported results not to validate our math but to experimentally determine the power of our algorithm. It is important to remember that the expected number of ballots counted during the audit is independent of the size of the election, except for quantization effects.

We consider auditing four kinds of elections differing in their error rates only. Our template is an election with two candidates who each receive 45% of the 100,000 votes cast. The template is then modified so that the difference in votes between the candidates is then increased to be an m fraction of the votes, where m ranges between .5% and 5%.

No errors. The first case we consider is when there are no errors. Thus, we have

$$M = \begin{pmatrix} .1 & 0 & 0 \\ 0 & .45 - m/2 & 0 \\ 0 & 0 & .45 + m/2 \end{pmatrix}. \qquad (49)$$

"Natural" unidirectional errors. In a roughly similar proportion to the error rates of the 2008 Minnesota Senate race, we add to the previous election 16 miscounts[4] of the form, "a vote for Candidate 1 was mistakenly counted for no one," and similarly for the other candidate. For example, the voter made a stray mark on the ballot and the

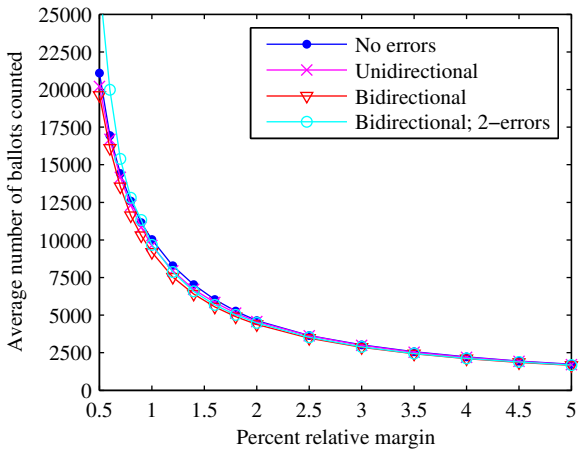


Figure 2: Average number of ballots counted vs. percent relative margin for four types of elections.

ballot was counted as being over voted. We have

$$M = \begin{pmatrix} .1 & 0 & 0 \\ \varepsilon & .45 - m/2 - \varepsilon & 0 \\ \varepsilon & 0 & .45 + m/2 - \varepsilon \end{pmatrix}, \quad (50)$$

where $\varepsilon = 16/100{,}000$.

"Natural" bidirectional errors. Rather than consider what happens when the error is all of the form "votes for a candidate were counted as no-votes," we consider 48 ballots of that form for each candidate and 32 ballots of the form "no-votes were counted for a candidate" for each candidate. Thus, the reported totals for each (real) candidate are exactly as they are in the previous scenario, but the errors are more extreme. We have

$$M = \begin{pmatrix} .1 - 4\varepsilon & 2\varepsilon & 2\varepsilon \\ 3\varepsilon & .45 - m/2 - 3\varepsilon & 0 \\ 3\varepsilon & 0 & .45 + m/2 - 3\varepsilon \end{pmatrix}, \quad (51)$$

where ε is unchanged.

"Natural" bidirectional errors with 2-errors. Modifying the previous scenario, we introduce 8 votes for the first candidate reported for the second candidate and 8 votes for the second counted for the first. These so-called *2-errors* change the reported margin by two. Again, the reported totals for each candidate remain the same,

$$M = \begin{pmatrix} .1 - 4\varepsilon & 2\varepsilon & 2\varepsilon \\ 3\varepsilon & .45 - (m + 7\varepsilon)/2 & \varepsilon/2 \\ 3\varepsilon & \varepsilon/2 & .45 + (m - 7\varepsilon)/2 \end{pmatrix}, \quad (52)$$

where ε is unchanged.

For each margin fraction m and each of the error rates, we simulate 1000 audits with risk level $\alpha = .01$, $T = 5$ batches, and batch sizes given in Section 3.3. The (average) fractions of ballots counted vs. the margin as a fraction of $n = 100{,}000$ for each of the four error conditions are plotted in Figure 2.

For margins at least as large as 1.5%, Figure 2 shows that only a few thousand ballots need to be counted in order to confirm elections with roughly realistic error rates. For smaller margins, it is more difficult to certify elections. Nevertheless, for large elections with small margins, counting 25,000 ballots may be feasible.

5 Extensions

The models and examples discussed here were for simple elections with only two candidates. We briefly touch on some extensions to our analysis for use in more general elections: handling multiple candidates, auditing multiple contests, errors in auditing, and improving our thresholds.

We can easily modify our results to handle elections with more than two candidates. The only challenge, theoretically, is that the set $\mathcal{R}$ as we have defined it may not be convex. However, for each candidate $c \in \mathcal{X}$ different from w_{reported}, we can define sets

$$\mathcal{D}_c = \left\{ d : \begin{array}{l} d(x) \geq 0 \ \ \forall x \in \mathcal{X}, \\ \sum_i d(i) = 1, \\ d(w) > d(w_{\text{reported}}) \end{array} \right\} \quad (53)$$

$$\mathcal{R}_c = \left\{ R : \begin{array}{ll} R(x,y) \geq 0 \ \ \forall x,y \in \mathcal{X}, & \sum_{x,y \in \mathcal{X}} R(x,y) = 1, \\ \sum_{x \in \mathcal{X}} R(x,y) = q(y), & \sum_{y \in \mathcal{X}} R(x,y) \in \mathcal{D}_c \end{array} \right\}. \quad (54)$$

For each c we can compute $\Delta_t(c)$ for $\mathcal{R} = \mathcal{R}_c$ and then take the minimum of $\Delta_t(c)$ over all $c \neq w_{\text{reported}}$. This procedure corresponds to doing pairwise tests between the reported winner and all reported losers. In a single-race election with C candidates the dimension for the optimization grows quadratically with C, and we would have to do C such optimizations. From a numerical standpoint, our current analysis is probably too loose to handle more candidates as a number of terms contain the square of the number of candidates in the exponent; however, it may be the case that the additional information gained about the distributions is sufficient to account for what is lost by moving to multiple candidates. Much of this could be mitigated by an improved analysis, but more simulation is also needed to validate the scaling performance of our methods.

Previous authors have considered auditing multiple elections simultaneously (e.g., [24]). The statistical methods we use here can be adapted to more complex election outcomes by employing additional pairwise tests: for auditing K races simultaneously with a maximum of C candidates in any race, it would require K times more computation than a single race with C candidates. It may be possible to modify our proposed mathematical framework to instead increase the dimension of the optimization problem; we leave this for future work.

Another problem which can occur in practice is errors *in the auditing*. Mathematically, we can model this as some uncertainty about the accuracy of our estimate $\hat{M}$. If the auditing errors are on the order of the margin of the election, then we run the risk of miscertifying the election when we assume $\hat{M}_t$ is accurate. A way to fix this is to associate to $\hat{M}_t$ an "uncertainty" set and minimize $D(\hat{M}_t \parallel R)$ over both $R \in \mathcal{R}$ and $\hat{M}_t$ in the uncertainty set.

As our experiments show, our analysis is very conservative. Better bounds on the probability of miscertification, for example by better bounds on the sizes of $\mathcal{C}_1$ and $\mathcal{C}_2$ could lead to a nearly 10-fold reduction in the number of ballots needed to certify correct elections while still maintaining the risk level. This is the major open problem posed by our paper. Additionally, it may be possible to increase the value of γ used in our simulations. This would involve a more careful evaluation of how our threshold condition behaves after batch $t+1$ conditioned on the fact that the threshold was not satisfied at time t. Such an analysis has been done by Stark [25] but those techniques (based on martingales) do not appear to apply to our test statistic directly.

6 Conclusions

We have presented a risk-limiting, statistical, ballot-based auditing algorithm that is resilient to errors, based on information-theoretic statistics and convex optimization. Our auditing algorithm is more efficient than current precinct-based auditing schemes. Our simulations suggest that the analysis we rely on for parameter selection could be improved, allowing for more efficient auditing. We believe that our algorithm provides an argument for installing the infrastructure required to use ballot-based auditing in elections.

7 Acknowledgments

We are deeply indebted to David Wagner for pointing out a crucial error that invalidated the risk analysis in what was to have been the final version of this paper.

We thank the reviewers for their helpful comments, Eric Rescorla for numerous discussions and comments, and Nadia Heninger for providing an early draft of her paper.

This material is based upon work supported by the National Science Foundation under Grant No. 0831532, by a MURI grant administered by the Air Force Office of Scientific Research, and by the California Institute for Telecommunications and Information Technology (CALIT2) at UC San Diego. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Air Force Office of Scientific Research.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[2] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9 (4):877–900, 1999.

[3] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[4] J. A. Calandrino, J. A. Halderman, and E. W. Felten. Machine-assisted election auditing. In R. Martinez and D. Wagner, editors, *Proceedings of EVT 2007*. USENIX and ACCURATE, Aug. 2007.

[5] J. A. Calandrino, J. A. Halderman, and E. W. Felten. In defense of pseudorandom sample selection. In Dill and Kohno [9].

[6] A. Cordero, D. Wagner, and D. Dill. The role of dice in election audits – extended abstract. Presented at WOTE 2006, June 2006. Online: `http://www.eecs.berkeley.edu/~daw/papers/dice-wote06.pdf`.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, Hoboken, New Jersey, second edition, 2006.

[8] I. Csiszár and J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Akadémi Kiadó, Budapest, 1982.

[9] D. Dill and T. Kohno, editors. *Proceedings of EVT 2008*, July 2008. USENIX and ACCURATE.

[10] K. Dopp. History of confidence election auditing development (1975 to 2008) & overview of election auditing fundamentals. `http:`

//electionarchive.org/ucvAnalysis/US/paper-audits/History-of-Election-Auditing-Development.pdf, Mar. 2008.

[11] J. A. Halderman, E. Rescorla, H. Shacham, and D. Wagner. You go to elections with the voting system you have: Stop-gap mitigations for deployed voting systems. In Dill and Kohno [9].

[12] J. L. Hall. Research memorandum: On improving the uniformity of randomness with alameda county's random selection process. UC Berkeley School of Information, Mar. 2008. Online: http://josephhall.org/papers/alarand_memo.pdf.

[13] N. Heninger. Computational complexity and information asymmetry in election audits with low-entropy randomness. In Jones et al. [16].

[14] D. Jefferson, J. L. Hall, and T. Moran, editors. *Proceedings of EVT/WOTE 2009*, Aug. 2009. USENIX, ACCURATE, and IAVoSS.

[15] K. C. Johnson. Election certification by statistical audit of voter-verified paper ballots. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=640943, 2004.

[16] D. Jones, J.-J. Quisquater, and E. Rescorla, editors. *Proceedings of EVT/WOTE 2010*, Aug. 2010. USENIX, ACCURATE, and IAVoSS.

[17] Los Angeles County Registrar-Recorder/County Clerk. Los Angeles County 1% manual tally report, November 4, 2008 general election, 2008. http://www.sos.ca.gov/voting-systems/oversight/mcr/2008-11-04/los-angeles.pdf.

[18] C. A. Neff. Election confidence: A comparison of methodologies and their relative effectiveness at achieving it. http://www.votehere.net/old/papers/ElectionConfidence.pdf, 2003.

[19] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, second edition, 2006.

[20] E. Rescorla. On the security of election audits with low entropy randomness. In Jefferson et al. [14].

[21] R. G. Saltman. Effective use of computing technology in vote-tallying. Technical Report Tech. Rep. NBSIR 75–687, National Bureau of Standards (Information Technology Division), Washington, D.C., USA, Mar. 1975. http://csrc.nist.gov/publications/nistpubs/NBS_SP_500-30.pdf.

[22] P. B. Stark. Conservative statistical post-election audits. *Ann. Appl. Stat.*, 2(2):550–581, Mar. 2008. doi: 10.1214/08-AOAS161. http://statistics.berkeley.edu/~stark/Preprints/conservativeElectionAudits07.pdf.

[23] P. B. Stark. CAST: Canvass audits by sampling and testing. *IEEE Transactions on Information Forensics and Security*, 4(4):708–717, Dec. 2009.

[24] P. B. Stark. Efficient post-election audits of multiple contests: 2009 California tests. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1443314, Aug. 2009.

[25] P. B. Stark. Risk-limiting postelection audits: Conservative *P*-values from common probability inequalities. *IEEE Transactions on Information Forensics and Security*, 4(4):1005–1014, Dec. 2009.

[26] P. B. Stark. Super-simple simultaneous single-ballot risk-limiting audits. In Jones et al. [16].

[27] C. Sturton, E. Rescorla, and D. Wagner. Weight, weight, don't tell me: Using scales to select ballots for auditing. In Jefferson et al. [14].

[28] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming*, 107(3): 391–408, 2006.

[29] X. Wang. Volumes of generalized unit balls. *Mathematics Magazine*, 78(5):390–395, Dec. 2005.

A An optimization program

The most computationally difficult part of Algorithm A is computing the minimization

$$\Delta = \min_{R \in \mathcal{R}} D(\hat{M} \parallel R). \tag{A.1}$$

Each time we need to compute this minimum, we form the function

$$\begin{aligned} f_{\hat{M}}(R) &= D(\hat{M} \parallel R) \\ &= \sum_{x,y \in \mathcal{X}} \hat{M}(x,y)\big(\log \hat{M}(x,y) - \log R(x,y)\big). \end{aligned} \tag{A.2}$$

Care must be taken to ensure that $R(x,y) \neq 0$ whenever $\hat{M}(x,y) \neq 0$. Similarly, if $\hat{M}(x,y) = 0$ for some $x,y \in \mathcal{X}$, then the summand corresponding to (x,y) should be zero.

Thus, $f_{\hat{M}}$ is a real-valued function of $(C+1)^2$ variables $R(0,0), R(0,1), \ldots, R(0,C), R(1,0), R(1,1), \ldots, R(C,C)$.

We want to minimize $f_{\hat{M}}(R)$ subject to the constraint $R \in \mathcal{R}$; (10). All of the constraints that define the feasible region $\mathcal{R}$ are linear equalities or inequalities:

$$\forall x, y \in \mathcal{X} \qquad R(x,y) \geq 0 \tag{A.3a}$$

$$\sum_{x,y \in \mathcal{X}} R(x,y) = 1 \tag{A.3b}$$

$$\forall y \in \mathcal{X} \qquad \sum_{x \in \mathcal{X}} R(x,y) = q(y) \tag{A.3c}$$

$$\sum_{y \in \mathcal{X}} R(l,y) \geq \sum_{y \in \mathcal{X}} R(w,y) \tag{A.3d}$$

where l is the reported loser and w is the reported winner. By treating R as a $(C+1)^2$-dimensional vector $\mathbf{r} = (R(x,y))_{x,y}$, these constraints can be expressed in matrix form $\mathbf{Ar} = \mathbf{b}$ and $\mathbf{A'r} \leq \mathbf{b'}$.

By Taylor's theorem in several variables, a (twice) differentiable function f can be written as

$$f(\mathbf{x}+\mathbf{h}) = f(\mathbf{x}) + \mathbf{h}^T \nabla f(\mathbf{x}) + \frac{1}{2}\mathbf{h}^T H(f)(\mathbf{x})\mathbf{h} + \cdots \tag{A.4}$$

where $\nabla f(\mathbf{x})$ and $H(f)(\mathbf{x})$ are the gradient and Hessian of f evaluated at $\mathbf{x}$, respectively. As a result, many optimization algorithms either require the gradient and Hessian of the objective function or perform better with access to them. For the case of $f_{\hat{M}}$, the gradient and the Hessian are taken with respect to the $(C+1)^2$ variables $R(x,y)$ and are easily computed,

$$\nabla f_{\hat{M}}(R) = \left(-\frac{\hat{M}(x,y)}{R(x,y)} \right)_{x,y} \tag{A.5}$$

$$H(f_{\hat{M}})(R) = \operatorname{diag}\left(\frac{\hat{M}(x,y)}{R(x,y)^2} \right)_{x,y} \tag{A.6}$$

where $\operatorname{diag}(\mathbf{v})$ is the diagonal matrix with the elements of $\mathbf{v}$ down the main diagonal.

Minimizing (A.2) subject to the constraints in (A.3) can be accomplished by using one of the standard constrained, nonlinear minimization algorithms such as interior point algorithms [2, 3, 28], [1, Chapter 11] or sequential quadratic programming algorithms [19, Chapter 18]. A custom solver can be written or an off-the-shelf numerical package such as MATLAB's Optimization Toolbox can be employed.

In general, with C candidates, the objective function $f_{\hat{M}}$ is minimized $C-1$ times where l in (A.3d) varies over the candidates other than the reported winner. The minimum value of all $C-1$ minimizations is thus the value Δ from (A.1).

B Bounding the certification region

To provide an (extremely loose) upper bound on the size of the certification region $\mathcal{C}^1$, we use Pinsker's inequality to relate the size of $\mathcal{C}^1$ to an upper bound on the size of $\mathcal{G}(\delta)$. Then, by treating $|\mathcal{Z}|-1$-dimensional difference vectors as a cube in $\mathbb{R}^{|\mathcal{Z}|-1}$ (with the l_1 metric), we can bound the number of such cubes which lie in a ball with a slightly larger radius δ'.

Lemma 1. *Fix $0 < \delta < 1$ and some distribution M. Let $\delta' = \delta + |\mathcal{Z}|/K$. Then for $\mathcal{G}(\delta) = \{P \in \mathcal{P}_K : \|P - M\|_1 \leq \delta\}$, the size of $\mathcal{G}(\delta)$ is bounded above by*

$$|\mathcal{G}(\delta)| \leq \frac{(2\delta' K)^{|\mathcal{Z}|-1}}{(|\mathcal{Z}|-1)!}. \tag{B.1}$$

Proof. Consider a distribution $P \in \mathcal{G}(\delta)$ and let $S = P - M$ as an element of $\mathbb{R}^{|\mathcal{Z}|}$. Denote by $\tilde{S}$ the first $|\mathcal{Z}|-1$ components of S.

By the definition of $\mathcal{G}(\delta)$, $\sum_z |\tilde{S}(z)| \leq \delta$ so $\tilde{S}$ lies in the closed δ-ball

$$\mathbb{B}(\delta) = \left\{ Q \in \mathbb{R}^{|\mathcal{Z}|-1} : \|Q\|_1 \leq \delta \right\}. \tag{B.2}$$

Since P and M are a probability distributions, the $|\mathcal{Z}|$th component of S is uniquely determined by the other components and thus the map sending $P \mapsto \tilde{S}$ is injective.

Now we compute the volume of a distribution and bound how many can lie in the ball. The reason for expanding the radius to δ' from δ is to ensure that the volume taken up by all distributions in $\mathcal{G}(\delta)$ is wholly contained in $\mathbb{B}(\delta')$. In $\mathbb{R}^{|\mathcal{Z}|}$, we can think of each distribution as occupying a cube of side length $1/K$ from P to $P + [0, 1/K)^{|\mathcal{Z}|}$. By truncating the last component, we see that this cube corresponds to $\tilde{S} + [0, 1/K)^{|\mathcal{Z}|-1}$. This cube is entirely contained within $\mathbb{B}(\delta')$ and furthermore, no element of the cube can correspond to a point in $\mathcal{P}_K$ other than P.

Therefore, $|\mathcal{G}(\delta)| \leq K^{|\mathcal{Z}|-1} \operatorname{Vol}(\mathbb{B}(\delta'))$. Since $\mathbb{B}(\delta')$ is an l_1 ball, its volume is [29]

$$\operatorname{Vol}(\mathbb{B}(\delta')) = \frac{(2\delta')^{|\mathcal{Z}|-1}}{(|\mathcal{Z}|-1)!}. \tag{B.3}$$

□

Notes

[1] `http://www.mathworks.com/products/optimization/`

[2] Thus, $\mathcal{B}_t$ is actually a multiset.

[3] Note that KL-divergence is not a proper metric [7].

[4] Since our sample election is roughly 30 times smaller than the 2008 Minnesota Senate race, we scale the roughly 500 errors to 16.