Statistics, Politics, and Policy

Volume 2, *Issue* 1 2011 *Article* 7

Sharper *p*-Values for Stratified Election Audits

Michael J. Higgins, *University of California, Berkeley*
Ronald L. Rivest, *Massachusetts Institute of Technology*
Philip B. Stark, *University of California, Berkeley*

Recommended Citation:

Higgins, Michael J.; Rivest, Ronald L.; and Stark, Philip B. (2011) "Sharper *p*-Values for Stratified Election Audits," *Statistics, Politics, and Policy*: Vol. 2: Iss. 1, Article 7.
DOI: 10.2202/2151-7509.1031
Available at: http://www.bepress.com/spp/vol2/iss1/7

Sharper *p*-Values for Stratified Election Audits

Michael J. Higgins, Ronald L. Rivest, and Philip B. Stark

Abstract

Vote-tabulation audits can be used to collect evidence that the set of winners of an election (the outcome) according to the machine count is correct — that it agrees with the outcome that a full hand count of the audit trail would show. The strength of evidence is measured by the *p*-value of the hypothesis that the machine outcome is wrong. Smaller *p*-values are stronger evidence that the outcome is correct.

Most states that have election audits of any kind require audit samples stratified by county for contests that cross county lines. Previous work on *p*-values for stratified samples based on the largest weighted overstatement of the margin used upper bounds that can be quite weak. Sharper *p*-values can be found by solving a 0-1 knapsack problem. For example, the 2006 U.S. Senate race in Minnesota was audited using a stratified sample of 2–8 precincts from each of 87 counties, 202 precincts in all. Earlier work (Stark 2008b) found that the *p*-value was no larger than 0.042. We show that it is no larger than 0.016: much stronger evidence that the machine outcome was correct.

We also give algorithms for choosing how many batches to draw from each stratum to reduce the counting burden. In the 2006 Minnesota race, a stratified sample about half as large — 109 precincts versus 202 — would have given just as small a *p*-value if the observed maximum overstatement were the same. This would require drawing 11 precincts instead of 8 from the largest county, and 1 instead of 2 from the smallest counties. We give analogous results for the 2008 U.S. House of Representatives contests in California.

KEYWORDS: post-election audits, knapsack problem

Author Notes: We are grateful to Luke Miratrix, James J. Higgins, and Katherine McLaughlin for helpful conversations and to anonymous editors for helpful comments. Hua Yang was extremely helpful coding a version of the branch and bound algorithm. We relied heavily on the elec package in R, originally written by Luke Miratrix and extended by Hua Yang.

1 Introduction

Votes are often tallied by machines, but—at least in many jurisdictions—the correct electoral outcome of an election is defined to be the outcome that a full hand count of the audit trail would show. There are many reasons a hand count might show a different electoral outcome than a machine count, including defects in the hardware or software of the machines, accidental misconfiguration, voter error, pollworker error, or malfeasance. Even if the vote tabulation machines function "correctly," the machine interpretation of a voter-marked paper ballot may differ from how a human would interpret the ballot in a hand count.

In post-election audits, also known as "vote-tabulation" audits, batches of ballots are selected and counted by hand. The hand-count subtotals are compared with the machine-count subtotals for each audited batch, and any differences between the machine count and hand count are noted. Most mandated post-election audits stop here.

In contrast, *risk-limiting audits* (Stark, 2008a,b, 2009a,b,c, Miratrix and Stark, 2009) guarantee a large chance of a full hand count whenever the machine outcome is wrong, no matter why the outcome is wrong. A full hand count reveals the true outcome (by definition), thereby correcting the machine outcome if the machine outcome was wrong. The *risk* is the largest chance that the audit will fail to correct an outcome that is wrong.

Risk-limiting audits generally proceed by taking an initial sample that is big enough to give strong evidence that the outcome is correct, provided the sample does not find much error in the machine count. If the initial sample does not turn out to give strong evidence (because it finds too much error), the sample is enlarged. This continues until either there is sufficiently strong evidence that the outcome is correct, or until all the votes have been counted by hand.

Evidence is measured by the p-value of the hypothesis that the machine-count outcome is incorrect. The p-value is the maximum chance that the audit would reveal "as little" error as it did reveal, on the assumption that the machine outcome is wrong. The maximum is taken over all ways that the outcome could be wrong. Smaller p-values are stronger evidence. A risk-limiting audit stops short of a full hand count only if the p-value becomes less than the risk limit α. This approach to auditing amounts to a sequential test of the hypothesis that the outcome is wrong. Defining "as little" amounts to specifying the test statistic for the hypothesis test. Many test statistics lead to tractable p-value calculations; see, e.g., Stark (2009c).

Risk-limiting audits are widely considered best practice[1] and have been endorsed by the American Statistical Association, The Brennan Center for Justice, Common Cause, the League of Women Voters, and Verified Voting, among others. California AB 2023, passed in 2010, requires a pilot of risk-limiting audits in 2011. Colorado Revised Statutes §1-7-515 calls for risk-limiting audits by 2014. As of this writing, there have been ten risk-limiting audits: nine in California (two in Marin County, three in Yolo County, and one each in Orange, Monterey, San Luis Obispo, and Santa Cruz counties), and one in Boulder County, Colorado. California and Colorado received grants from the Election Assistance Commission in 2011 to develop and implement risk-limiting audits.

Risk-measuring audits are related to risk-limiting audits. They do not necessarily expand until the p-value is small. But they quantify the evidence that the machine outcome is correct by reporting the p-value of the hypothesis that the machine outcome is wrong.

States with election audit laws generally require each jurisdiction to audit the votes cast in a simple random sample of precincts. For example, California Elections Code §15360 requires each county to take a 1% sample of precincts and hand count all ballots within those precincts; if this misses any contest in any county, the sample is augmented to include at least one precinct with each contest. Minnesota Elections Law S.F. 2743 (2006) requires a sample of 2, 3, or 4 precincts from each county, depending on the size of the county. This results in a stratified random sample for contests that cross jurisdictional boundaries: The strata are jurisdictions. Even when the law does not require it, there may be logistical reasons to use stratified samples. For instance, scheduling the audit may be easier if batches of ballots cast in-person are audited separately from batches of vote-by-mail ballots and from batches of provisional ballots. Audit samples might also be stratified by the machine used to cast or count votes.

The first work on risk-limiting audits (Stark, 2008a) addressed stratified samples, developing a crude upper bound on the p-value when the test statistic is the maximum observed margin overstatement across audited batches (more generally, the maximum of monotone transformations of the overstatements in each audited batch). This paper constructs sharper bounds on the p-value for stratified samples for the same family of test statistics. The improvement, which can be substantial (the sharper p-value is just over 1/3 of the crude upper bound on the p-value for the 2006 U.S. Senate race in Minnesota), is largest when the sampling fractions vary across strata.

This paper also gives methods to choose sample sizes within strata to reduce the p-value for a given sample size and presumed value of the test statistic. This can

[1] See `http://electionaudits.org/principles.html` (last visited 23 September 2011).

substantially reduce the counting burden of a risk-limiting audit when the machine outcome is correct.

2 Audits using stratified simple random samples

2.1 Notation and framework

If a and b are real numbers, $a \vee b$ denotes the maximum of a and b and $a \wedge b$ denotes their minimum. For instance, $(1 \vee 2) = 2$ and $(1 \wedge 0) = 0$. The symbol $\equiv$ denotes a definition: $f(x) \equiv x^2$ defines $f(x)$ to be x^2. For any proposition s,

$$\mathbf{1}(s) \equiv \begin{cases} 1, & \text{if } s \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

For example, $\mathbf{1}(1 > 0) = 1$ and $\mathbf{1}(1 > 2) = 0$. If $a \equiv (a_j)_{j=1}^N$ and $b \equiv (b_j)_{j=1}^N$ are vectors of the same length N, the *inner product* of a and b is

$$a \cdot b \equiv \sum_{j=1}^{N} a_j b_j. \tag{2}$$

The sum of an empty list is defined to be zero and the product of an empty list is defined to be one: $\sum_{j=1}^{0} a_j \equiv 0$, $\prod_{j=1}^{0} a_j \equiv 1$. The product $0 \times \infty \equiv 0$ and the exponential $0^0 \equiv 1$. The minimum of any function over an empty domain is ∞, and the maximum of a function over an empty domain is $-\infty$.

"Apparent outcome" and "machine outcome" are synonymous, as are "apparent vote total" and "machine vote total." "Hand-count outcome," "correct outcome," and "true outcome" mean the same thing, as do "hand-count vote total" and "actual vote total." An apparent winner wins according to the machine count; a true winner would win according to a full hand count. The apparent outcome is correct if the apparent winners are the true winners.

We consider auditing one contest at a time. There are I candidates in the contest. The contest is of the form "vote for up to W candidates," and there are W apparent winners and $I - W$ apparent losers. (In more general scenarios, which we do not consider here, the voter may vote for a number of candidates that differs from the number of winners to be determined by the election.) The ballots are grouped into N batches spread across C strata, which are numbered 1 through C. There are N_c batches in stratum c. The kth batch in stratum c is denoted (k, c).

The total number of ballots cast in batch (k, c) is b_{kc}. The apparent vote total for candidate i in batch (k, c) is v_{kci}. The actual vote total for candidate i in batch

(k,c) is a_{kci}. The values of b_{kc} and v_{kci} are known for every batch, but a_{kci} is known only if batch (k,c) is audited. The apparent vote total for candidate i is

$$V_i \equiv \sum_{c=1}^{C} \sum_{k=1}^{N_c} v_{kci}.$$

The actual vote total for candidate i is

$$A_i \equiv \sum_{c=1}^{C} \sum_{k=1}^{N_c} a_{kci}.$$

Let $\mathbf{I}_W$ denote the apparent winners of the contest and $\mathbf{I}_L$ denote the apparent losers. Note that $\#\mathbf{I}_W = W$. We assume that there is no loser whose apparent vote total was equal to that of any winner. As a practical matter, such ties are rare in large contests. But if there were a tie, a risk-limiting audit would demand a full hand count, which is not the most interesting case statistically.

The apparent margin in votes between candidate $w \in \mathbf{I}_W$ and candidate $\ell \in \mathbf{I}_L$ is

$$V_{w\ell} = V_w - V_\ell > 0.$$

The true margin in votes between candidates w and ℓ is

$$A_{w\ell} = A_w - A_\ell.$$

The apparent outcome is correct if every winner actually got more votes than every loser: if for all $w \in \mathbf{I}_W$ and $\ell \in \mathbf{I}_L$,

$$A_{w\ell} > 0, \tag{3}$$

or equivalently, if

$$V_{w\ell} - A_{w\ell} = \sum_{c=1}^{C} \sum_{k=1}^{N_c} [v_{kcw} - v_{kc\ell} - (a_{kcw} - a_{kc\ell})] < V_{w\ell}. \tag{4}$$

The apparent outcome is wrong if and only if [4] fails for some $w \in \mathbf{I}_W$ and $\ell \in \mathbf{I}_L$.

Let e^H_{kc} denote a measure of the difference between the machine count and the hand count in batch (k,c). The value of e^H_{kc} is known only if batch (k,c) is audited. We call the values e^H_{kc} "differences" because they are functions of

$$\{v_{kci} - a_{kci}\}_{k=1}^{N_c}\,{}_{c=1}^{C}\,{}_{i=1}^{I}.$$

The vector $e^H \equiv (e^H_{kc})_{k=1}^{N_c}\,{}_{c=1}^{C}$ is the *true allocation of differences*. We require e^H_{kc} to be defined so that there exists a known constant μ for which:

$$\text{If the apparent election outcome is wrong, then } \sum_{c=1}^{C} \sum_{k=1}^{N_c} e^H_{kc} \geq \mu. \tag{5}$$

The difference e^H_{kc} (and the resulting constant μ) can be defined many ways. A reasonable choice is the *maximum relative overstatement* (MRO) introduced by Stark (2008b):

$$e^H_{kc} \equiv \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kc\ell} - (a_{kcw} - a_{kc\ell})}{V_{w\ell}}. \tag{6}$$

For the MRO, [5] holds with $\mu = 1$.

Testing statistically whether

$$\sum_{c=1}^{C} \sum_{k=1}^{N_c} e^H_{kc} \geq \mu \tag{7}$$

generally requires an *a priori* upper bound ω_{kc} for e^H_{kc}, for each batch (k,c), known before the audit begins. Stark (2008b) shows that if difference is measured by the MRO,

$$e^H_{kc} \leq \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kc\ell} + b_{kc}}{V_{w\ell}} \equiv \omega_{kc}. \tag{8}$$

Without loss of generality, we assume that within each stratum c, the batches are ordered so that

$$\omega_{kc} \geq \omega_{k'c} \quad \text{if} \quad k < k'. \tag{9}$$

An *allocation of differences* or *allocation* is a vector

$$e = (e_{kc})_{k=1}^{N_c}{}_{c=1}^{C} \in \mathbb{R}^N \text{ such that } e_{kc} \leq \omega_{kc}, \quad k = 1, \ldots, N_c, \; c = 1, \ldots, C. \tag{10}$$

Let $\mathbf{E}$ be the set of all such allocations, and let

$$\mathbf{E}_\mu \equiv \left\{ e \in \mathbf{E} : \sum_{c=1}^{C} \sum_{k=1}^{N_c} e_{kc} \geq \mu \right\}. \tag{11}$$

If the apparent outcome is wrong, $e^H \in \mathbf{E}_\mu$.

2.2 Computing the *p*-value

This section sets out the precise problem we solve: finding a sharper (but still conservative) p-value for the null hypothesis[2] that the apparent outcome is incorrect from a stratified random sample. Let $\mathbf{J}^{n_c}_c$ be a simple random sample of n_c elements

[2] http://xkcd.com/892/ (last visited 23 September 2011).

from $\{(1,c),\ldots,(N_c,c)\}$, and let $\{\mathbf{J}_c^{n_c}\}_{c=1}^C$ be independent random samples. Let $\vec{n} \equiv (n_c)_{c=1}^C$, and let

$$\mathbf{J}_{\vec{n}} \equiv \bigcup_{c=1}^{C} \mathbf{J}_c^{n_c}.$$

Then $\mathbf{J}_{\vec{n}}$ is a stratified random sample of batches. We want to test the hypothesis that $e^H \in \mathbf{E}_\mu$ using

$$T \equiv \max_{(k,c)\in\mathbf{J}_{\vec{n}}} e^H_{kc} \tag{12}$$

as the test statistic. If T is surprisingly small on the assumption that $e^H \in \mathbf{E}_\mu$, we will conclude that the outcome is correct.

Instead of using the maximum MRO as the test statistic, we could use the maximum of a set of more general monotone transformations of the observed differences: Let $\{w_{kc}\}_{k=1}^{N_c}{}_{c=1}^{C}$ be a set of N monotone increasing functions. We could base the audit on the test statistic

$$T_w \equiv \max_{(k,c)\in\mathbf{J}_{\vec{n}}} w_{kc}(e^H_{kc}),$$

where e^H_{kc} is not necessarily the MRO. For instance, in Section 5, we consider *taint*. Using the maximum of monotone transformations of the observed differences as the test statistic leads to tractable probability calculations for a stratified sample; in contrast, using the sum of the observed differences does not. For discussion, see Stark (2008a). To simplify the exposition, we focus on the MRO. Section C lists the other changes to definitions required to use more general monotone weight functions.

The hypothesis $e^H \in \mathbf{E}_\mu$ does not completely specify the sampling distribution of T. That distribution depends on all components of e^H. We only know e^H_{kc} if batch (k,c) is audited, so to have a rigorous test, we assume the worst: If the maximum difference in the sample is t, then e^H is the element of $\mathbf{E}_\mu$ that maximizes the probability that $T \le t$. Let $e \in \mathbf{E}$ be an allocation of differences. Define

$$P_{\mathbf{J}_{\vec{n}}}(e) \equiv P_{\mathbf{J}_{\vec{n}}}(e;t) \equiv P\left(\max_{(k,c)\in\mathbf{J}_{\vec{n}}} e_{kc} \le t\right). \tag{13}$$

This is the probability that the maximum observed difference in the stratified random sample of batches $\mathbf{J}_{\vec{n}}$ will be no greater than t if the allocation of differences is e; that is, $\Pr_e\{T \le t\}$.

Suppose that, for the actual audit sample, the maximum observed difference is $T = t$. Then the *exact p-value* of the hypothesis that the apparent outcome is wrong is

$$P_{\#} = P_{\#}(t;\vec{n}) \equiv \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e;t). \tag{14}$$

Any $P_+ = P_+(t;\vec{n})$ for which

$$P_+ \geq P_{\#} \tag{15}$$

is a *conservative p-value*.

We now compute $P_{\mathbf{J}_{\vec{n}}}(e;t)$ for an arbitrary $e \in \mathbf{E}$ and $t \in \mathbb{R}$. For $e \in \mathbf{E}$, let

$$\mathbf{G}(e) = \mathbf{G}(e;t) \equiv \{(k,c) : e_{kc} > t\} \tag{16}$$

be the set of batches with difference greater than t, and let

$$\#_c\mathbf{G}(e) \equiv \#\{k : (k,c) \in \mathbf{G}(e)\}$$

be the number of batches within stratum c with difference greater than t.

Let $e \in \mathbf{E}$. If $N_c - \#_c\mathbf{G}(e) < n_c$, then a simple random sample of size n_c from $\{(1,c), \ldots, (N_c,c)\}$ is guaranteed to contain a batch with difference $e_{kc} > t$, so $P_{\mathbf{J}_c^{n_c}}(e) = 0$. If $N_c - \#_c\mathbf{G}(e) \geq n_c$, the probability that $\mathbf{J}_c^{n_c}$ does not contain any batch with difference $e_{kc} > t$ is

$$P_{\mathbf{J}_c^{n_c}}(e) = \frac{\binom{N_c - \#_c\mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}}.$$

The samples from different strata are drawn independently, so the probability that a stratified random sample of batches does not include any batch with $e_{kc} > t$ is

$$P_{\mathbf{J}_{\vec{n}}}(e) = \begin{cases} \displaystyle\prod_{c=1}^{C} \frac{\binom{N_c - \#_c\mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}}, & N_c - \#_c\mathbf{G}(e) \geq n_c, \quad c = 1, \ldots, C, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The exact p-value $P_{\#}$ [14] is the maximum of $P_{\mathbf{J}_{\vec{n}}}(e)$ over all allocations $e \in \mathbf{E}_\mu$.

For large cross-jurisdictional contests, finding the exact p-value by brute force is prohibitively expensive. The following sections show that [14] has special structure that allows us to find the exact p-value quickly.

3 Stratified audits and the 0-1 knapsack problem

You are packing a knapsack with food for a camping trip. You have available N food items, each of which has a weight and a caloric value. You want to pack the combination of food items that has at least M calories and weighs the least. This is a version of the 0-1 knapsack problem (KP), an NP-complete problem (Karp, 2010) with a long history and large literature (Pisinger, 1995, Pisinger and Toth, 1998).

We show in this section that there is a "small" set $\tilde{\mathbf{E}}_\mu$ such that

$$P_\# \equiv \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e) = \max_{e \in \tilde{\mathbf{E}}_\mu} P_{\mathbf{J}_{\vec{n}}}(e). \tag{18}$$

We then show that maximizing $P_{\mathbf{J}_{\vec{n}}}$ over allocations in $\tilde{\mathbf{E}}_\mu$ can be couched as KP.[3] Even though the problem is NP-complete, the maximum can be found in a matter of seconds, even for large, multi-jurisdictional contests. Good upper bounds can be calculated even faster.

3.1 Characterizing optimal allocations of differences

Recall that $P_{\mathbf{J}_{\vec{n}}}(e)$, the chance that the maximum difference in a stratified sample with sample sizes $\vec{n}$ is no larger than t, depends on e only through $(\#_c \mathbf{G}(e))_{c=1}^C$, the number of batches in each stratum that have differences greater than t. Smaller values of $\#_c \mathbf{G}(e)$ lead to bigger values of $P_{\mathbf{J}_{\vec{n}}}(e)$.

Given an allocation e, we can produce another allocation $\tilde{e}$ that has at least as much difference in each stratum and for which $P_{\mathbf{J}_{\vec{n}}}(\tilde{e}) \geq P_{\mathbf{J}_{\vec{n}}}(e)$ by concentrating the difference in each stratum c in the batches k that have the largest upper bounds ω_{kc}. That is, $\tilde{e}$ has at least as much total difference as e an is at least as likely to produce a sample with no difference greater than t.

The values $\kappa_c(e)$, defined below, limit how far this can go: An allocation must have at least $\kappa_c(e)$ batches in stratum c with difference exceeding t to have at least as much difference in stratum c as the allocation e has. For $e \in \mathbf{E}$, let

$$\kappa_c(e) \equiv \min \left\{ k' \geq 0 : \sum_{k=1}^{k'} \omega_{kc} + \sum_{k'+1}^{N_c} (\omega_{kc} \wedge t) \geq \sum_{k=1}^{N_c} e_{kc} \right\}.$$

For any $e \in \mathbf{E}$, let $\tilde{e} \equiv (\tilde{e}_{kc})_{k=1}^{N_c}{}_{c=1}^{C}$ be the vector with components

$$\tilde{e}_{kc} \equiv \begin{cases} \omega_{kc}, & k \leq \kappa_c(e), \\ \omega_{kc} \wedge t, & \text{otherwise.} \end{cases}$$

[3] Rivest (2007) shows that when batches are audited independently, finding

$$\max_{e \in \mathbf{E}_\mu} P\left(\text{ Not auditing any batch } (k,c) \text{ with difference } e_{kc} > 0 \right)$$

can be cast as KP. However, stratified random sampling does not select batches independently.

Note that

$$\tilde{e} \in \mathbf{E} \text{ and } \tilde{\tilde{e}} = \tilde{e}. \tag{19}$$

By definition of κ_c,

$$\sum_{k=1}^{N_c} \tilde{e}_{kc} \geq \sum_{k=1}^{N_c} e_{kc}.$$

Hence,

$$\text{if } e \in \mathbf{E}_\mu \text{ then } \tilde{e} \in \mathbf{E}_\mu. \tag{20}$$

By [9],

$$\text{if } k < k',\ [\omega_{kc} - (\omega_{kc} \wedge t)] \geq [\omega_{k'c} - (\omega_{k'c} \wedge t)]. \tag{21}$$

It follows from the rearrangement theorem (Hardy, Littlewood, and Pólya, 1952), and the fact that $e_{kc} \leq \omega_{kc}$ that

$$\begin{aligned}
\sum_{k=1}^{\#_c\mathbf{G}(e)} \omega_{kc} \quad + & \sum_{\#_c\mathbf{G}(e)+1}^{N_c} (\omega_{kc} \wedge t) \\
= & \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)]\mathbf{1}(k \leq \#_c\mathbf{G}(e)) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\
\geq & \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)]\mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\
\geq & \sum_{k=1}^{N_c} [e_{kc} - t]\mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} t\mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} e_{kc}\mathbf{1}(e_{kc} \leq t) \\
= & \sum_{k=1}^{N_c} e_{kc}.
\end{aligned} \tag{22}$$

Thus, $\kappa_c(e) \leq \#_c\mathbf{G}(e)$, so for $c = 1, \ldots, C$,

$$\#_c\mathbf{G}(\tilde{e}) = \kappa_c(e) \leq \#_c\mathbf{G}(e).$$

It follows from [17] that

$$P_{\mathbf{J}_{\vec{n}}}(\tilde{e}) \geq P_{\mathbf{J}_{\vec{n}}}(e). \tag{23}$$

That is, compared with e, $\tilde{e}$ has at least as much difference and at least as large a chance of yielding a sample with no difference larger than t: It does at least as much damage to the election outcome and is at least as hard to detect using a stratified random sample.

Since [19], [20], and [23] hold for all $e \in \mathbf{E}$, it follows that

$$\max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(\tilde{e}) = \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e). \tag{24}$$

Thus, if we define

$$\tilde{\mathbf{E}} \equiv \{\tilde{e} : e \in \mathbf{E}\}, \tag{25}$$

and let

$$\tilde{\mathbf{E}}_\mu \equiv \tilde{\mathbf{E}} \cap \mathbf{E}_\mu, \tag{26}$$

then [18] holds for this definition of $\tilde{\mathbf{E}}_\mu$.

The set of allocations $\tilde{\mathbf{E}}_\mu$ is much smaller than the original set $\mathbf{E}_\mu$. Maximizing $P_{\mathbf{J}_{\vec{n}}}$ over allocations in this smaller set can be reduced to KP, as we now show.

3.2 Maximizing $P_{\mathbf{J}_{\vec{n}}}$ as a 0-1 knapsack problem

We write the 0-1 knapsack problem more precisely. There are N items. Item j has value $u_j \geq 0$ and cost $q_j \geq 0$. The value and cost are analogous to the caloric value and weight in the example in section 3. We want to find the combination of items that has minimal total cost among all combinations with total value above some threshold. In the example of section 3, this is like finding the combination of food items that has minimal total weight among all combinations with total caloric value above some threshold. Let $M \geq 0$ and let

$$\mathbf{X} \equiv \left\{ (x_j)_{j=1}^N : x_j \in \{0,1\} \right\}.$$

Define $x \equiv (x_j)_{j=1}^N$, $u \equiv (u_j)_{j=1}^N$, and $q \equiv (q_j)_{j=1}^N$. The 0-1 knapsack problem (KP) is to find

$$\lambda \equiv \min_{x \in \mathbf{X}} \{ q \cdot x : u \cdot x \geq M \}.$$

Recall that the minimum of a function over an empty domain is ∞, so if $\{x \in \mathbf{X} : u \cdot x \geq M\}$ is empty, $\lambda = \infty$. A vector $x^\dagger \in \mathbf{X}$ satisfying

$$q \cdot x^\dagger = \lambda \quad \text{and} \quad u \cdot x^\dagger \geq M$$

is called an *exact solution*; λ is the *exact value*. Finding λ can be expensive; often it is substantially easier to find a lower-bound $\lambda^- \leq \lambda$, an *approximation* to the exact value.

We show below that finding the exact p-value $P_\#$ also amounts to solving KP. To do so, we relate the constraint $u \cdot x \geq M$ to the condition $e \in \mathbf{E}_\mu$ and the objective function $q \cdot x$ to $P_{\mathbf{J}_{\vec{n}}}$. Moreover, we show that it is not necessary to search all of $\mathbf{X}$ for the minimum: We find a much smaller set $\tilde{\mathbf{X}} \subset \mathbf{X}$ for which

$$\log P_{\#} = \log \max_{e \in \tilde{\mathbf{E}}_\mu} \{P_{\mathbf{J}_{\vec{n}}}(e)\} = \min_{e \in \tilde{\mathbf{E}}_\mu} \{-\log(P_{\mathbf{J}_{\vec{n}}}(e))\} = \min_{y \in \tilde{\mathbf{X}}} \{q \cdot y : u \cdot y \geq M\}. \quad (27)$$

We then show that

$$\lambda = \min_{x \in \mathbf{X}} \{q \cdot x : u \cdot x \geq M\} = \min_{y \in \tilde{\mathbf{X}}} \{q \cdot y : u \cdot y \geq M\}. \quad (28)$$

Hence, any algorithm for solving KP can find the exact p-value $P_{\#} = e^{-\lambda}$. But algorithms that restrict the search to vectors $x \in \tilde{\mathbf{X}}$ can be faster than algorithms that search all of $\mathbf{X}$.

Variables: It is helpful to switch between doubly-indexed terms and singly-indexed terms. The double index k, c corresponds to the single index

$$j = \mathrm{j}(k,c) \equiv k + \sum_{c' < c} N_{c'}, \quad k = 1, \ldots, N_c, \; c = 1, \ldots, C. \quad (29)$$

Conversely, the single index j corresponds to the double index k, c with

$$c = \mathrm{c}(j) \equiv \min \left\{ d : \sum_{i=1}^{d} N_d \geq j \right\}, \quad k = \mathrm{k}(j) \equiv j - \sum_{d=1}^{\mathrm{c}(j)-1} N_d, \quad (30)$$

Recall that $\mathbf{G}(e)$ is the set of batches (k,c) for which $e_{kc} > t$ [16]. For $e \in \mathbf{E}$, define

$$g_{kc}(e) \equiv \mathbf{1}((k,c) \in \mathbf{G}(e)), \quad (31)$$

$$g(e) \equiv (g_{kc}(e))_{k=1}^{N_c}\,{}_{c=1}^{C} \in \mathbf{X}, \quad (32)$$

and

$$\tilde{\mathbf{X}} \equiv \left\{ y \in \mathbf{X} : y = g(e) \text{ for some } e \in \tilde{\mathbf{E}} \right\}. \quad (33)$$

Constraint: Let

$$u_{kc} \equiv \omega_{kc} - (\omega_{kc} \wedge t). \quad (34)$$

Note that

$$u_{kc} = 0 \quad \text{if and only if} \quad \omega_{kc} \leq t. \quad (35)$$

By [21],

$$u_{kc} \geq u_{k'c} \quad \text{if} \quad k < k'. \quad (36)$$

Let

$$M \equiv \left[\mu - \sum_{c=1}^{C} \sum_{k=1}^{N_c} \omega_{kc} \wedge t \right] \vee 0. \tag{37}$$

Observe that if $M = 0$, then

$$\omega \wedge t \equiv (\omega_{kc} \wedge t)_{k=1}^{N_c} {}_{c=1}^{C} \in \tilde{\mathbf{E}}_\mu$$

and

$$P_{\mathbf{J}_{\vec{n}}}(\omega \wedge t) = 1.$$

Thus, if $M = 0$, then the exact p-value $P_\# = 1$: There is an allocation of difference that causes the election outcome to be wrong, and for which the probability is 100% that the sample will not contain any batch with difference greater than t.

Subtracting $\sum_{c=1}^{C} \sum_{k=1}^{N_c} (\omega_{kc} \wedge t)$ from both $\sum_{c=1}^{C} \sum_{k=1}^{N_c} e_{kc}$ and μ shows that for $e \in \tilde{\mathbf{E}}$, $e \in \tilde{\mathbf{E}}_\mu$ if and only if

$$u \cdot g(e) \geq M. \tag{38}$$

Thus,

$$\{g(e) : e \in \tilde{\mathbf{E}}_\mu\} = \{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}. \tag{39}$$

We assume $\{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}$ is non-empty; otherwise, [39] shows that the apparent outcome must be correct, so the p-value is 0.

Objective function: Choose $e \in \mathbf{E}$. If for $c = 1, \ldots, C$, $N_c - \#_c\mathbf{G}(e) \geq n_c$, then

$$\begin{aligned} P_{\mathbf{J}_{\vec{n}}}(e) &= \prod_{c=1}^{C} \frac{\binom{N_c - \#_c\mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}} = \prod_{c=1}^{C} \prod_{k=1}^{\#_c\mathbf{G}(e)} \frac{\binom{N_c - k}{n_c}}{\binom{N_c - k + 1}{n_c}} \\ &= \prod_{c=1}^{C} \prod_{k=1}^{\#_c\mathbf{G}(e)} \frac{N_c - n_c - k + 1}{N_c - k + 1}. \end{aligned} \tag{40}$$

If instead there exists c such that $N_c - \#_c\mathbf{G}(e) < n_c$, then $P_{\mathbf{J}_{\vec{n}}}(e) = 0$: If the true allocation is e, the sample is guaranteed to contain a batch with difference greater than t. Combining this with [40] shows that for any $e \in \mathbf{E}$,

$$P_{\mathbf{J}_{\vec{n}}}(e) = \prod_{c=1}^{C} \prod_{k=1}^{\#_c\mathbf{G}(e)} \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \tag{41}$$

Let

$$p_{kc} \equiv \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \tag{42}$$

Note that

$$p_{kc} \geq p_{k'c} \quad \text{if} \quad k < k'. \tag{43}$$

Recall our convention that $0^0 = 1$. If $e \in \tilde{\mathbf{E}}$, then

$$P_{\mathbf{J}_{\vec{n}}}(e) = \prod_{c=1}^{C} \prod_{k=1}^{\#_c \mathbf{G}(e)} p_{kc} = \prod_{c=1}^{C} \prod_{k=1}^{N_c} p_{kc}^{g_{kc}(e)}. \tag{44}$$

That is, for allocations $e \in \tilde{\mathbf{E}}$, batch (k,c) has a *fixed* contribution p_{kc} to $P_{\mathbf{J}_{\vec{n}}}$. This is the key to writing $P_\#$ as KP. Let

$$q_{kc} \equiv \begin{cases} -\log(p_{kc}), & p_{kc} > 0, \\ \infty, & p_{kc} = 0. \end{cases} \tag{45}$$

Note that $q_{kc} \geq 0$ for all batches (k,c). By [43],

$$q_{kc} \leq q_{k'c} \quad \text{if} \quad k < k'. \tag{46}$$

From [44] and [45], for $e \in \tilde{\mathbf{E}}$,

$$-\log(P_{\mathbf{J}_{\vec{n}}}(e)) = q \cdot g(e). \tag{47}$$

Equations [39] and [47] yield equation [27]. We prove [28] in appendix A.

4 Approximate and exact solutions to KP

Dynamic programming algorithms and branch and bound algorithms can solve KP (Pisinger and Toth, 1998). Appendix B describes a branch and bound algorithm for finding $P_\#$ that restricts the search to $\tilde{\mathbf{X}}$ to improve efficiency. That algorithm can calculate the exact p-value in a matter of seconds, even for large elections. R code is available in the CRAN archive in the package `elec.strat`.

The solution to KP can be bounded from below in $O(N)$ time by solving the linear knapsack problem (LKP), the continuous relaxation of KP (Pisinger and Toth, 1998). We use this *LKP bound*, $\lambda_{\text{LKP}} \leq \lambda$, in the bound step of the branch-and-bound algorithm in appendix B. We call $P_{\text{LKP}} \equiv e^{-\lambda_{\text{LKP}}} \geq P_\#$ the *LKP conservative p-value*. For some election audits, P_{LKP} is almost exactly equal to $P_\#$, the exact p-value.

LKP relaxes the constraint that each item either is or is not in the knapsack to the constraint that between 0 and 100% of each item is in the knapsack: The discrete set $\{0,1\}$ is replaced by the continuous set $[0,1]$. Define

$$\mathbf{X}^{rel} \equiv \left\{ (x_j)_{j=1}^N : x_j \in [0,1] \right\} \supset \mathbf{X}. \tag{48}$$

The LKP is to find

$$\lambda_{\text{LKP}} \equiv \min_{x \in \mathbf{X}^{rel}} \{ q \cdot x : u \cdot x \geq M \}. \tag{49}$$

The value λ_{LKP} is the LKP bound. Since $\mathbf{X} \subset \mathbf{X}^{rel}$, $\lambda_{\text{LKP}} \leq \lambda$, and

$$P_{\text{LKP}} \equiv \exp(-\lambda_{\text{LKP}}) \geq P_{\#}.$$

LKP can be solved using linear programming, but Dantzig (1957) shows that the value of λ_{LKP} can be obtained very simply, as follows. Sort the ratios

$$r_{kc} \equiv \frac{q_{kc}}{u_{kc}} \tag{50}$$

into increasing order, and put the cost vector q into the corresponding order. Find the smallest B so that the sum of the values of the first B batches is at least M. The LKP bound is the sum of the first $B-1$ components of the cost vector q and a fraction of the B^{th} component of q.

We now explain the LKP bound in more detail. Equations [36] and [46] show that

$$r_{kc} \leq r_{k'c} \ \text{ if } \ k < k'. \tag{51}$$

Recall [29] and [30], the mappings between double indices and single indices. Let $\pi : \{1,2,\ldots,N\} \to \{1,2,\ldots,N\}$ be a permutation such that

$$r_{\pi(1)} \leq r_{\pi(2)} \leq \cdots \leq r_{\pi(n)}. \tag{52}$$

That is, π maps j to the index of the j^{th} smallest value of $(r_j)_{j=1}^N$. For instance, if $\pi(1) = j$, then $r_j = \min(r_i)_{i=1}^N$. The inverse of π, denoted π^{-1}, maps j to the rank of r_j. For instance, if $r_j = \min(r_i)_{i=1}^N$, then $\pi^{-1}(j) = 1$.

If there are ties among the ratios $(r_j)_{j=1}^N$, we impose two additional conditions on π:

1. When $r_{kc} = r_{k'c}$ and $k < k'$,

$$\pi^{-1}(\mathrm{j}(k,c)) < \pi^{-1}(\mathrm{j}(k',c)). \tag{53}$$

2. When $r_{kc} = r_{k^*c^*}$, $c \neq c^*$, and $N_c > N_{c^*}$,

$$\pi^{-1}(\mathrm{j}(k,c)) < \pi^{-1}(\mathrm{j}(k^*,c^*)). \tag{54}$$

The first condition, together with [51] and [52], ensures that π preserves the order of batches within a stratum. The second condition breaks ties between ratios in different strata by putting the ratio in the stratum with fewer batches first.

For any $j' \in \{1,\ldots,N\}$ with $u_{\pi(j')} > 0$,

$$\left(\mathbf{1}[\pi^{-1}(j) \leq j']\right)_{j=1}^{N} \in \tilde{\mathbf{X}}. \tag{55}$$

That is, any allocation that assigns as much difference as possible to batches with the smallest ranks and difference $\omega \wedge t$ to larger ranks is in $\tilde{\mathbf{E}}$. To see this, consider the allocation e^* with components

$$e^*_{\pi(j)} = \begin{cases} \omega_{\pi(j)}, & 1 \leq \pi^{-1}(j) \leq j', \\ \omega_{\pi(j)} \wedge t, & \text{otherwise.} \end{cases}$$

By [52] and [53], if $\pi^{-1}(\mathrm{j}(k',c)) \leq j'$ and $k < k'$, then $\pi^{-1}(\mathrm{j}(k,c)) < j'$. That is, if $e^*_{k'c} = \omega_{k'c}$ and $k < k'$, then $e^*_{kc} = \omega_{kc}$. Thus, $e^* \in \tilde{\mathbf{E}}$. By [33] and [35],

$$g(e^*) = \left(\mathbf{1}[\pi^{-1}(j) \leq j']\right)_{j=1}^{N} \in \tilde{\mathbf{X}}.$$

Define

$$B \equiv \begin{cases} 1, & M = 0 \text{ and } u_{\pi(1)} = 0, \\ N \wedge \min\left\{B' > 0 : \sum_{j=1}^{B'} u_{\pi(j)} \geq M\right\}, & \text{otherwise.} \end{cases}$$

Then B is the smallest number of batches that must have difference greater than t for the election outcome to be wrong, if those differences are allocated in the order π. Note that, if $\{q \cdot y : u \cdot y \geq M\}$ is non-empty then $u_B > 0$. Dantzig (1957) shows that

$$\lambda_{\text{LKP}} = \begin{cases} 0, & M = 0 \text{ and } u_{\pi(1)} = 0, \\ \sum_{j=1}^{B-1} q_{\pi(j)} + \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}} \cdot q_{\pi(B)}, & \text{otherwise.} \end{cases} \tag{56}$$

The vector $x^{rel} \in \mathbf{X}^{rel}$ that attains this maximum has components

$$x^{rel}_{\pi(j)} \equiv \begin{cases} 1, & j < B, \\ \dfrac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}, & j = B, \\ 0, & \text{otherwise.} \end{cases} \tag{57}$$

Observe that $u \cdot x^{rel} = M$, and $\lambda_{\text{LKP}} = 0$ when $M = 0$. If $\sum_{j=1}^{B} u_{\pi(j)} = M$, then x^{rel} actually solves KP, not just LKP:

$$\text{If } \sum_{j=1}^{B} u_{\pi(j)} = M \text{ then } \lambda = \sum_{j=1}^{B} q_{\pi(j)}. \tag{58}$$

Note that $\left(\mathbf{1}[\pi^{-1}(j) \le B)]\right)_{j=1}^{N} \in \tilde{\mathbf{X}}$. If $\{q \cdot y : u \cdot y \ge M\}$ is non-empty, then $\sum_{j=1}^{B} u_{\pi(j)} \ge M$, and so

$$\lambda_{\text{LKP}}^{+} \equiv \sum_{j=1}^{B} q_{\pi(j)}$$

is an upper bound for λ: LKP lets us bracket the value of KP. Observe that

$$\lambda_{\text{LKP}} + \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} = \lambda_{\text{LKP}}^{+}. \tag{59}$$

Thus,

$$\lambda - \lambda_{\text{LKP}} \le \lambda_{\text{LKP}}^{+} - \lambda_{\text{LKP}} = \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} \le q_{\pi(B)}, \tag{60}$$

and so

$$\frac{\exp(-\lambda_{\text{LKP}})}{\exp(-\lambda)} \le \frac{1}{p_{\pi(B)}}. \tag{61}$$

That is, P_{LKP} is guaranteed to be within a factor of $1/p_{\pi(B)}$ of the exact p-value $P_{\#}$.

5 Results: comparing p-values

This section gives exact and conservative p-values for the hypothesis that the apparent outcome of the 2006 U.S. Senate race in Minnesota was wrong. Amy Klobuchar was the apparent winner; Mark Kennedy was the runner-up. There were a total of 2,217,818 ballots cast in 4,123 precincts spanning 87 counties. Klobuchar's reported margin of victory over Kennedy was 443,196 votes.

Many Minnesota counties are small; only ten had more than 75 precincts in 2006. Counties audited 2 to 8 precincts selected at random, depending on the size of the county. Hennepin County, which has the most precincts (426), audited 8 precincts. In all, 202 precincts were audited. For more information about the election and audit, see Halvorson and Wolff (2007).

Table 1: Conservative and exact p-values for the hypothesis that the apparent outcome of the 2006 U.S. Senate race in Minnesota was wrong, based on Minnesota's audit of a stratified random sample of 202 precincts. Values are given for two test statistics: maximum MRO and maximum taint. Column 2: conservative p-value using the method of Stark (2008b). Column 3: LKP conservative p-value. Column 4: exact p-value obtained by solving KP.

	Stark	P_{LKP}	$P_{\#}$
MRO	0.042	0.01591	0.01590
Taint	0.047	0.01892	0.01890

We consider tests based on two measures of difference: MRO and *taint*. The taint of a batch is the difference in the batch expressed as a fraction of the maximum possible difference in the batch. Taint is related to MRO through a weight function w_{kc}: If e_{kc} is the MRO in batch (k,c), the taint in batch (k,c) is

$$w_{kc}(e_{kc}) = \frac{e_{kc}}{\omega_{kc}}.$$

The largest overstatement of Klobuchar's margin over Kennedy in the audit sample was 2 votes, so the maximum MRO was 2/443,196. The largest taint found by the audit was 9.17×10^{-3}, a one vote overstatement of Klobuchar's margin in a precinct in Cottonwood county containing 149 ballots. For MRO,

$$M = 1 - \sum_{c=1}^{87} \sum_{k=1}^{N_c} (\omega_{kc} \wedge (2/443196)).$$

For taint,

$$M = 1 - \sum_{c=1}^{87} \sum_{k=1}^{N_c} (\omega_{kc} \times 9.17 \times 10^{-3}).$$

Table 1 gives conservative p-values using the method of Stark (2008b) and LKP, and the exact p-value obtained by solving KP. The exact p-values are less than half the conservative values based on the method in Stark (2008b). The LKP conservative p-value is nearly equal to the exact p-value.

Figure 1 shows conservative and exact p-values corresponding to some possible values of the maximum MRO and maximum taint. The LKP conservative p-values are essentially identical to the exact p-values; both are much smaller than the conservative p-value based on the method of Stark (2008b).

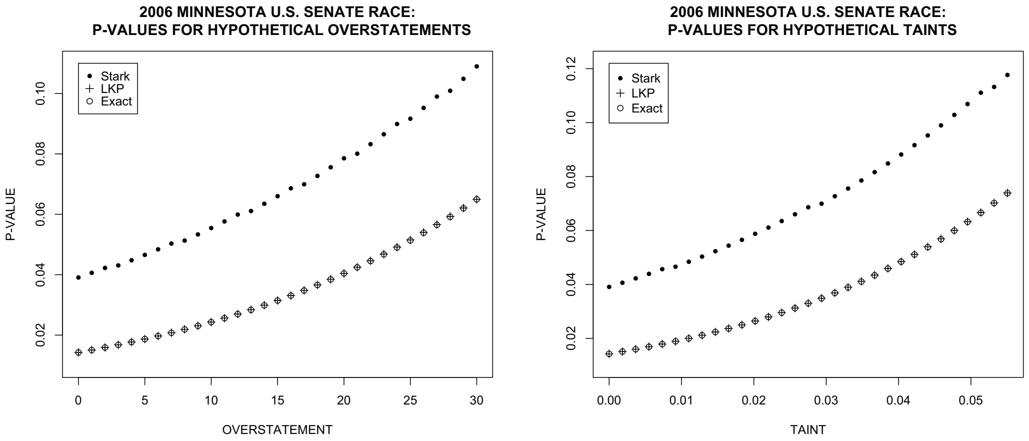

Figure 1: Exact and conservative p-values for hypothetical maximum observed overstatements (left) and hypothetical maximum observed taints (right) for the 2006 Minnesota Senate race. The LKP conservative p-values (P_{LKP}) are nearly identical to the exact p-values ($P_{\#}$). Both are substantially smaller than bounds using the method in Stark (2008b).

If the test statistic is maximum MRO, the exact p-value is less than 0.05 if the largest overstatement less than than 26 votes. The conservative p-value from the method of Stark (2008b) is less than 0.05 only if the largest overstatement is less than 8 votes. If the test statistic is the maximum taint, the exact p-value is less than 0.05 if the observed maximum taint is less than 0.040; while the conservative p-value using the method of Stark (2008b) is less than 0.05 only if the observed maximum taint is less than 0.011: KP and LKP give substantially more powerful tests.

6 Selecting sample sizes

So far, we have assumed that the sample sizes in each stratum were given in advance, for instance, by law. Finding the best sample sizes—those that can confirm correct outcomes with the least hand counting—seems to be computationally intractable, but it is not hard to improve on the sample sizes used in Minnesota, for instance. In this section we pose optimization problems to define "optimal" sample sizes and give several methods for selecting sample sizes. Section 7 shows that selecting sample sizes to be proportional to the number of batches, which is how

California currently sets sample sizes, performs well in examples using data from real elections.

Recall that, for any choice of sample sizes $\vec{n} = (n_c)_{c=1}^C$, $\mathbf{J}_{\vec{n}}$ is a stratified random sample that selects n_c batches from stratum c, $c = 1, \ldots, C$. For fixed $\alpha > 0$ and $t^* > 0$, let $\mathbf{N}(\alpha, t^*)$ denote the set of all sample sizes $\vec{n}$ such that, if the maximum observed difference is t^* or less, the exact p-value obtained using sample sizes $\vec{n}$ will be less than α. That is,

$$\mathbf{N} = \mathbf{N}(\alpha, t^*) \equiv \left\{ \vec{n} = (n_c)_{c=1}^C : \begin{array}{l} P_{\#}(t^*; \vec{n}) \leq \alpha, \\ n_c \in \{0, 1, \ldots, N_c\}, c = 1, \ldots, C. \end{array} \right\}$$

We define a vector of sample sizes $\vec{n}^\dagger = (n_c^\dagger)_{c=1}^C$ to be *optimal* (for α and t^*) if

$$\sum_{c=1}^C n_c^\dagger = \min \left\{ \sum_{c=1}^C n_c : (n_c)_{c=1}^C \in \mathbf{N}(\alpha, t^*) \right\}. \tag{62}$$

By this definition, a vector of sample sizes is optimal if it minimizes the number of batches that must be counted to confirm the outcome at risk limit α on the assumption that the value of the test statistic turns out to be no larger than t^*. There can be more than one optimal vector of sample sizes.

There are other sensible definitions of optimality. If the vector of sample sizes is $\vec{n}$, the expected number of ballots that need to be hand counted is

$$\sum_{c=1}^C \frac{n_c}{N_c} \sum_{k=1}^{N_c} b_{kc}. \tag{63}$$

We might define a vector of sample sizes to be optimal if it minimizes the expected number of ballots that must be counted to confirm the outcome at risk limit α, again on the assumption that the value of the test statistic turns out to be no larger than t^*. Or the expectation could allow t^* to be random (for instance, based on a hypothetical allocation of difference), and could take into account the costs of expanding the audit if the p-value is larger than α. If batches are about the same size, a sample size vector that minimizes the number of batches audited will also minimize the expected number of ballots audited. In practice, there are costs to retrieve batches of ballots and to hand-count the votes on each ballot in a batch, so defining optimality in terms of a weighted combination of the number of batches and the expected number of ballots is appealing; weights might depend on how a jurisdiction organizes its ballots, on labor costs, etc. The methods described below can be modified to work for these optimality criteria, but we focus on minimizing the number of batches.

Optimal sample size vectors can be found by brute force when the contest spans few counties and the margin of victory is large. We give three simple algorithms for finding sample sizes that can improve on statutory allocations even when a brute-force solution is impossible. The core of each algorithm takes the total sample size $n \equiv \sum_c n_c$ to be fixed and selects $\vec{n}$ to make $P_{\#}(t^*;\vec{n})$ small. The algorithms increment n until $P_{\#}(t^*;\vec{n}) \leq \alpha$.

6.1 Sample sizes proportional to stratum size

A simple rule for allocating the sample across strata is to take sample sizes proportional to stratum size (PSS). California Elections Code §15360 requires sample sizes that are close to PSS sample sizes: Each county audits a random sample of 1% of its precincts, plus one precinct for each contest not included in the 1% sample.

PSS does not take advantage of information about the amount of difference batches can contain. In some cases, PSS sample sizes are close to optimal. However, when strata are not similar—for example, when one stratum has a disproportionately high number of batches that can hold large differences—PSS sample sizes can be far from optimal.

When nN_c/N is an integer for all $c = 1,\dots,C$, the PSS sample sizes are $n_c = nN_c/N$. When nN_c/N is not an integer for some c, we might define PSS sample sizes to be $n_c = \lceil nN_c/N \rceil$. In that case, PSS sample sizes would satisfy $\sum_{c=1}^{C} n_c \geq n$. Alternatively, we might define PSS sample sizes to satisfy $\sum_{c=1}^{C} n_c = n$ as follows: Sort the ratios $\{f_{kc} \equiv (k-1)N/N_c\}$, $k = 1,\dots,N_c$, $c = 1,\dots,C$, in ascending order, listing f_{kc} before $f_{k^*c^*}$ if $f_{kc} = f_{k^*c^*}$ and $N_c > N_{c^*}$. Consider the smallest n such ratios. The sample size n_c is the number of those n ratios that came from stratum c. We use this latter definition of PSS sample sizes in section 7.

6.2 first.r and next.r

We now present two algorithms to find sample sizes—`first.r` and `next.r`—that use information about stratum sizes and the amount of difference individual batches can hold. This can produce sample sizes that are smaller than PSS sample sizes when strata are dissimilar.

The algorithms are related. Both start with an empty sample size vector $\vec{n} = (0)_{c=1}^{C}$ and increment the sample size in the stratum c that contains the batch with the largest value of r (in some pool of batches) until the total sample size is n. The difference between the algorithms is whether the batch with the largest value of r at one iteration is kept in the pool (`first.r`) or excluded from consideration in subsequent iterations (`next.r`). After each increment, the costs [45] are updated based on the current value of $\vec{n}$. The ratios $(r_j)_{j=1}^N$ are updated, and the permutation π that sorts these ratios into increasing order is found.[4] Both algorithms use π to determine which n_c to increment, but they use different rules to make that determination. The algorithms are as follows.

Step 1: (Initialize)

Set $\vec{n} = (n_c)_{c=1}^C = (0)_{c=1}^C$.

Compute $(u_j)_{j=1}^N$.

Set $\mathbf{S} = \{1, \ldots, N\}$.

Step 2: (Update q, r, and π)

Using the current value of $\vec{n}$, compute $(q_{kc})_{k=1}^{N_c}{}_{c=1}^{C}$.

Set $q_{kc} = \min(q_{kc}, \log(n_c + 1))$.

Compute $(r_j)_{j=1}^N$.

Find the permutation π satisfying [52], [53], and [54].

Step 3: (Choose which n_c to increment)

Find $j = \min\{j' : \pi(j') \in \mathbf{S}\}$.

Increment $n_{\mathrm{c}(\pi(j))}$ (see equation [30]).

Step 4: (Update the search set.)

If `next.r`, set $\mathbf{S} = \mathbf{S} \setminus \pi(j)$.

Else if `first.r`, do nothing.

Step 5: (Terminate?)

[4] `next.r` requires the cost to be defined slightly differently:

$$q_{kc} \equiv -\log(p_{kc}) \wedge \log(n_c + 1).$$

This only matters if more than half of the batches in a stratum need to be sampled, which can occur in a closely contested race. The permutation π is not affected by this change, since $q_{kc} = \infty$ if and only if $k > N_c - n_c$. For $k \leq N_c - n_c$, by [46],

$$q_{kc} = -\log\left(\frac{N_c - n_c - k + 1}{N_c - k + 1}\right) \leq \log(n_c + 1)$$

with equality if and only if $k = N_c - n_c$. Thus, the ordering in [46] continues to hold.

If $\sum_{c=1}^{C} n_c < n$, go to Step 2.
Else stop.

By [52] and [53], we know that the minimum in Step 3 is one of only C values; this restriction can be exploited to decrease the computational time of the algorithm dramatically.

6.3 Constructing sample size vectors in $\mathbf{N}(\alpha, t^*)$.

Constructing a vector of sample sizes $\vec{n} \in \mathbf{N}(\alpha, t^*)$ is straightforward:
Step A: Set $n = 1$.
Step B: Given n, use PSS, `first.r`, or `next.r` to construct a vector of sample sizes $\vec{n}$ with $\sum_c n_c = n$.
Step C: Find the exact p-value $P_{\#}(t^*, \vec{n})$ on the assumption that the observed value of the test statistic is t^*. (A conservative p-value $P_{\#}(t^*, \vec{n}) \geq P_{\mathbf{J}_{\vec{n}}}(e; t^*)$ could be used instead of the exact p-value.)
Step D: If $P_{\#} > \alpha$, increment n and go to Step B. Otherwise, $\vec{n} \in \mathbf{N}(\alpha, t^*)$.

The next section gives numerical examples based on data from Minnesota and California.

7 Sample sizes for Minnesota and California contests

We use the data from the 2006 Minnesota Senate race to demonstrate how selecting sample sizes using PSS, `first.r`, or `next.r` can dramatically reduce the counting necessary for an audit. We then use data from the 2008 California U.S. House races to compare the performance of these methods.

7.1 The 2006 Minnesota U.S. Senate race

The statutory audit of the 2006 Minnesota election examined 202 precincts. As discussed in section 5, counties audited between 2 and 8 precincts each, depending on the size of the county. For the U.S. Senate contest, the largest observed overstatement of the margin in a single precinct was 2 votes; the corresponding exact p-value for the hypothesis that the apparent outcome is incorrect is 0.0159. The largest taint in a single precinct was 9.17×10^{-3}. The corresponding exact p-value is 0.0189.

Table 2: Statutory, PSS, `first.r`, and `next.r` sample sizes for the 2006 Minnesota Senate contest. Number of batches to audit and expected number of ballots to audit to obtain p-values no larger than the exact p-values in Table 1 (0.0159 for maximum MRO and 0.0189 for maximum observed taint), for the same observed values of the test statistics. PSS, `first.r`, and `next.r` all improve markedly on the statutory sample sizes.

		Statutory	PSS	`first.r`	`next.r`
Overstatement	Number of batches	202	122	109	110
	Expected ballots	90,691	59,611	55,787	56,940
Taint	Number of batches	202	122	108	109
	Expected ballots	90,691	59,611	55,228	55,851

To study the effectiveness of the statutory sampling rates, we find the sample sizes that would be required to get p-values at least as small for sampling vectors chosen using `first.r`, `next.r`, and the version of PSS that satisfies $\sum_{c=1}^{C} n_c = n$. The calculations assume that the observed value of the test statistic would be the same for all samples. The results are in Table 2, along with the expected number of ballots that would need to be tallied by hand.

All three new methods require auditing dramatically fewer batches and ballots than the statutory method: Selecting sample sizes more efficiently would reduce the number of batches by 80 (almost 40%) and would reduce the expected number of ballots to audit by one third (see equation [63]). The new methods draw more than 8 precincts from Hennepin county and only one precinct from the smallest counties, instead of two.

Figure 2 compares the total sample sizes and expected number of ballots to tally by hand for PSS, `first.r`, and `next.r` to get p-values no larger than 0.05, for observed maximum overstatements of 0 to 30 votes. The analogous graphs using taint as the test statistic are nearly identical.

`first.r` and `next.r` perform best in these examples: Only 100 batches need to be audited when the maximum overstatement is zero, and 113 batches or fewer need to be audited for a 30-vote overstatement of the margin. The total number of precincts and the expected number of ballots to audit are uniformly smaller for `first.r` and `next.r` than for PSS. The difference between `first.r` and `next.r` sample sizes and PSS sample sizes is greatest when the observed overstatement is large.

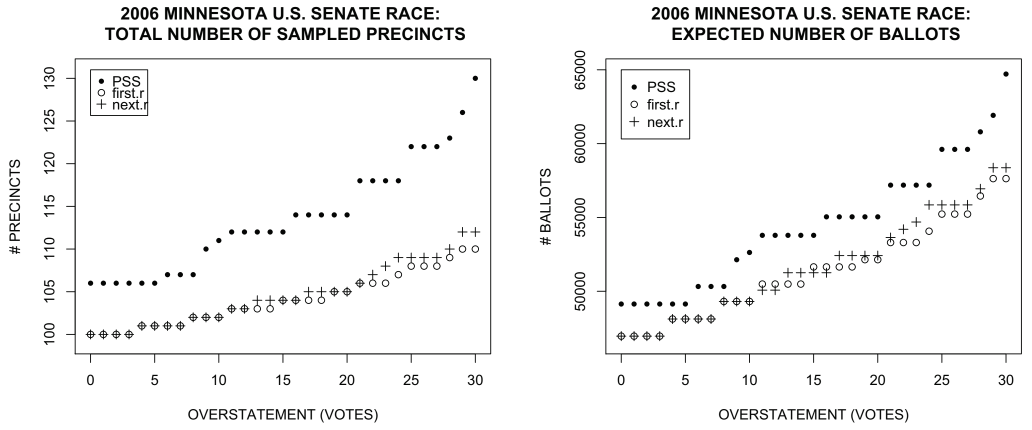

Figure 2: Number of batches to audit and expected number of ballots to audit to get p-values no larger than 0.05 for 2006 Minnesota Senate race, as observed maximum overstatements range from 0 to 30 votes, using sample size vectors selected by PSS, `first.r`, and `next.r`. In these simulations, PSS requires more auditing than `first.r` and `next.r`, which have nearly identical workloads.

7.2 The 2008 California U.S. House of Representatives races

The November 2008 election in California included 53 U.S. House of Representatives contests. The California Statewide Database (SWDB) has precinct-level voting data for these contests[5] The SWDB does not give the results of the statutory 1% audit.

Of these 53 contests, two had third-party candidates who received a substantial proportion of the vote; the SWDB did not provide vote totals for these third-party candidates. In nine of the contests, a single candidate was running unopposed. We omitted these 11 contests from our study.

Of the remaining 44 contests, 23 crossed county lines. Of those, 20 were contained in 5 counties or fewer, allowing us to find optimal sample size vectors by brute force.

We find PSS, `first.r`, `next.r`, and optimal sample sizes and expected ballots to audit to attain p-values no larger than 0.05 provided the audit does not uncover any overstatement of the margin (that is, sample size vectors in $\mathbf{N}(0.05, 0)$). We exclude precincts (k, c) with $\omega_{kc} = 0$, because differences in those precincts

[5] See `http://swdb.berkeley.edu/pub/data/G08/state/state_g08_sov_data_by_g08_svprec.dbf`.

could not have overstated the apparent margin. Table 3 lists the results, along with summary statistics such the number of counties and precincts in the contest and the margin of victory as a percentage of votes cast in the contest. Figures 3 and 4 plot the results.

PSS sample sizes are optimal in 8 contests and within 2 batches of optimal in 14 contests. Sample sizes from `first.r` are optimal in 9 contests and within 2 batches of optimal in 15 contests. Sample sizes from `next.r` are optimal in 12 contests and within 2 batches of optimal in 19 contests.

For 11 of the contests, PSS required auditing the most batches. For 10 contests, PSS had the largest expected number of ballots to audit. The PSS sample sizes were far from optimal for the District 11 and the District 44 contests.

`next.r` never required auditing the largest number of batches nor the largest expected number of ballots. However, it required auditing far more than the optimal number of batches and ballots in District 44.

All three approximate methods find sample sizes very quickly, even for large contests. Given a threshold value of the test statistic t^* and risk limit α, one can apply all three methods and choose whichever requires auditing the fewest batches or the fewest expected ballots. This is legitimate because the choice takes place before the sample is drawn. (In contrast, one cannot draw the samples all three ways and decide which of the samples to use after looking at the audit results—with “data snooping” of that kind, the nominal p-value could differ substantially from the true p-value.) For many contests, the methods perform similarly. The simplest—PSS—is typically quite good. For small contests, it can be close to optimal.

8 Conclusions and Future Work

Risk-limiting post-election audits guarantee that if the apparent outcome of a contest is wrong, there is a large chance of a full hand count to set the record straight. The risk is the maximum chance that the audit will not correct an apparent outcome that is wrong. A risk-limiting audit can be thought of as a hypothesis test: The null hypothesis is that the apparent outcome is wrong. A type I error corresponds to failing to correct a wrong outcome. The chance of a type I error is the risk. The p-value of the null hypothesis quantifies the evidence that the outcome is correct: Smaller p-values are stronger evidence.

Table 3: Summary of 20 multi-jurisdiction 2008 California U.S. House of Representative contests and audit workload for several methods of selecting sample sizes. Column 1: legislative district. Column 2: number of counties containing the contest. Column 3: number of precincts in the contest. Column 4: largest number of precincts in the contest in any single county. Column 5: total votes cast in the contest. Column 6: margin of victory as a percentage of valid votes cast. Columns 7–10: number of batches to audit if sample size vectors are selected using PSS, `first.r`, `next.r`, or optimally. The optimal choice is not unique. Columns 11–13: expected number of ballots to audit if sample size vectors are selected using PSS, `first.r`, or `next.r`.

	Contest summary					Precincts to audit				Expected ballots to audit		
CD	C	N	max	Votes	M	PSS	`f.r`	`n.r`	Opt	PSS	`f.r`	`n.r`
			N_c		(%)							
12	2	599	385	293,469	51.5	13	11	11	11	6,454	5,775	5,775
6	2	1,110	732	336,749	45.3	15	15	15	15	5,224	5,224	5,224
7	2	535	293	252,898	47.3	15	15	15	15	8,210	8,210	8,210
14	3	940	530	296,795	43.6	17	16	16	16	6,465	6,227	6,227
51	2	844	628	219,232	45.1	16	16	16	16	4,863	4,863	4,863
17	3	766	368	240,205	45.7	19	18	18	18	7,227	6,989	6,989
23	3	818	392	266,259	34.1	20	21	20	20	7,535	7,857	7,481
10	4	728	430	318,243	31.5	23	21	22	21	11,275	10,580	10,927
20	3	1,152	420	131,708	46.2	22	23	23	22	3,791	3,928	3,928
21	2	1,056	568	225,375	34.2	26	25	25	25	6,822	6,556	6,554
42	3	669	307	289,757	18.4	38	40	34	33	19,060	23,199	17,968
41	2	1,688	1,222	277,945	21.6	41	42	41	41	11,659	11,872	11,659
49	2	1,152	730	263,844	19.0	42	41	41	41	13,066	12,818	12,864
24	2	1,176	932	322,001	15.1	51	51	50	50	18,606	18,914	18,427
25	4	1,151	777	275,404	14.0	63	62	61	60	19,130	18,997	18,742
11	4	1,167	782	318,195	9.8	85	65	61	61	28,351	23,171	22,576
26	2	1,000	650	296,714	10.9	64	67	65	64	22,671	23,810	23,011
46	2	660	402	307,160	8.7	77	74	73	71	38,346	38,721	37,121
3	5	829	696	339,812	5.1	130	122	122	121	56,969	54,158	54,380
44	2	811	712	274,349	2.2	355	289	315	270	142,882	122,564	129,325

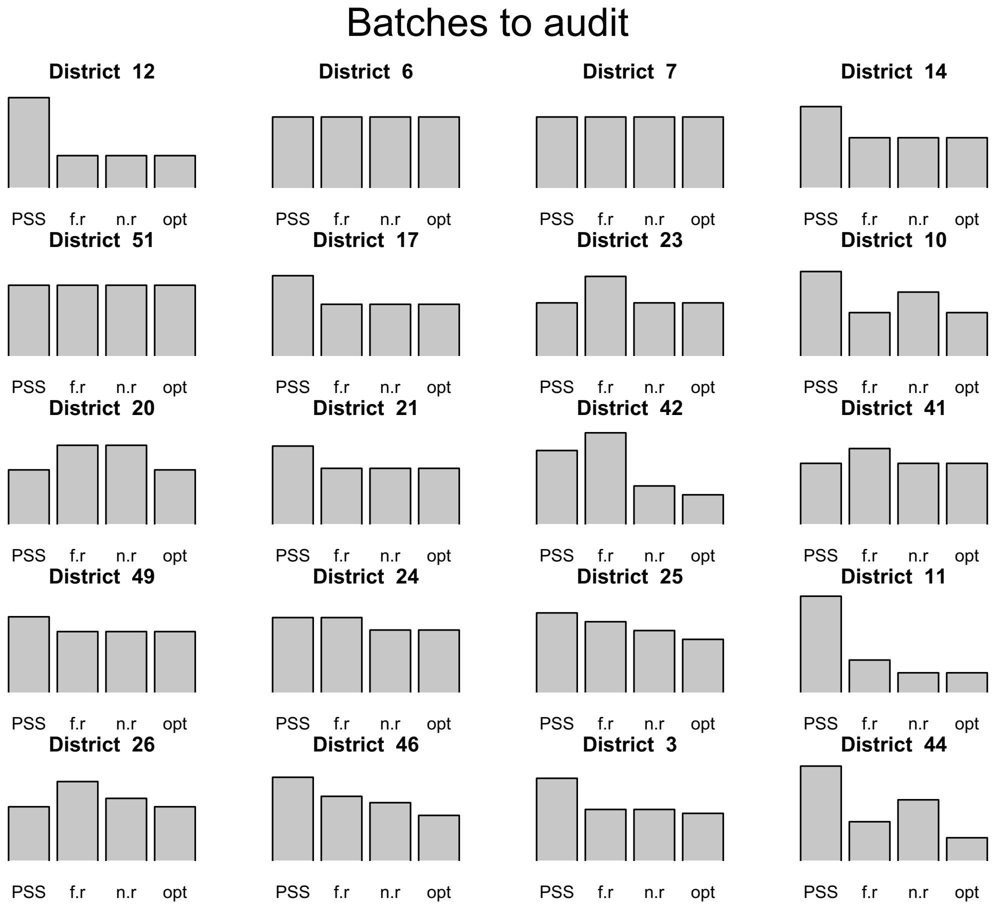

Figure 3: Number of batches to audit so that the p-value of the hypothesis that the outcome is incorrect will be less than $\alpha = 0.05$ if the sample finds no difference that overstated a margin. Bar graphs plot the ratio of the number of batches to audit for sample size vectors chosen using PSS, `first.r`, and `next.r` to the number of batches an optimal sample-size vector requires. `first.r` and `next.r` tend to require fewer batches than PSS. For many contests, the differences among methods are small.

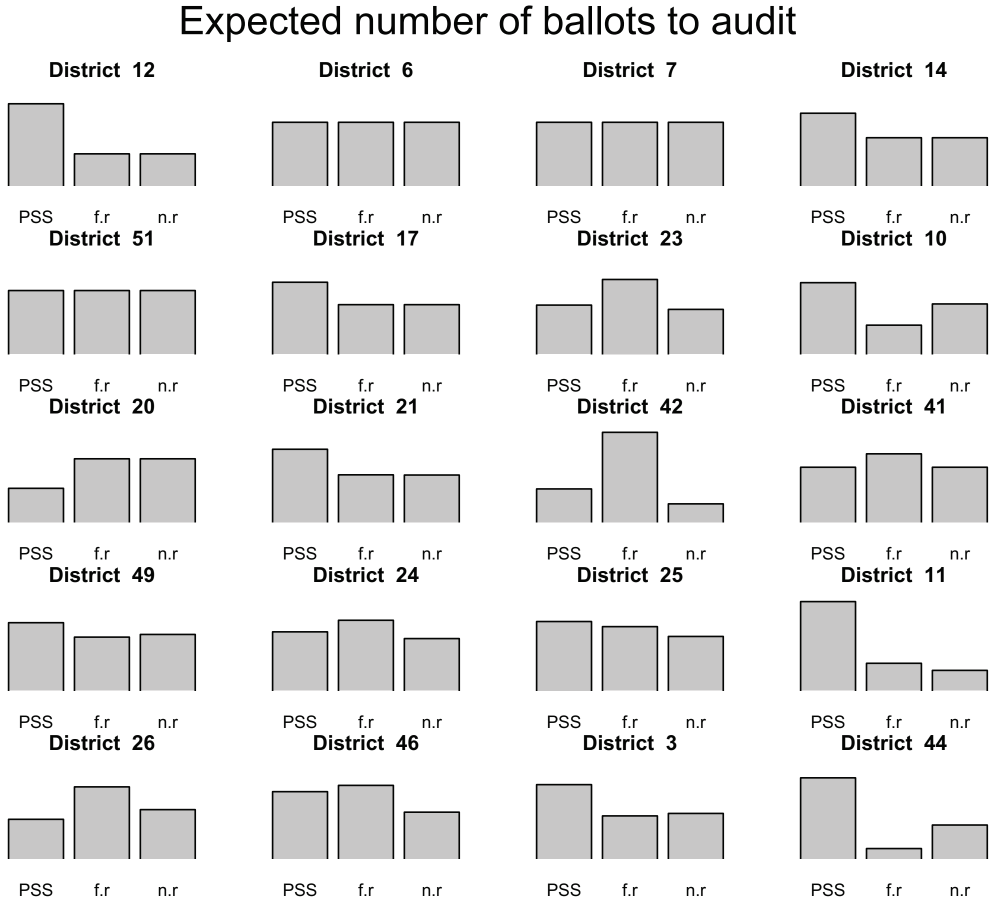

Figure 4: Expected number of ballots to audit so that the p-value of the hypothesis that the outcome is incorrect will be less than $\alpha = 0.05$ if the sample finds no difference that overstated a margin). Bar graphs plot the ratio of the expected number of ballots for PSS and `first.r` to the expected number of ballots for `next.r`. `first.r` and `next.r` tend to require fewer ballots than PSS. For many contests, the differences among methods are small.

Previous work on risk-limiting audits using stratified samples found upper bounds on *p*-values that were extremely conservative when sampling fractions varied widely across strata. We have shown here how to find a sharp *p*-value based on a stratified sample by solving a 0-1 knapsack problem (KP). KP can be solved efficiently using a branch and bound algorithm. The linear knapsack problem (LKP) bound gives an inexpensive upper bound on the *p*-value that is almost sharp: For the 2006 U.S. Senate contest in Minnesota, the exact *p*-value found by KP is nearly identical to the LKP conservative *p*-value, and both are dramatically smaller than conservative *p*-value computed using the method in Stark (2008a,b).

Sampling rates within strata have a large effect on workload. We show that in Minnesota, an audit could have obtained the same *p*-value by sampling 80 fewer precincts and counting a third fewer ballots, if the maximum difference observed by the audit remained the same. Simulations based on the 2008 U.S. House of Representatives contests in California suggest that choosing sample sizes to be proportional to the number of batches in each stratum can be close to optimal. Minnesota's stratification is far from proportional.

The legal requirement to use stratification makes some aspects of auditing more complex, and some simpler. It would be interesting to study how stratification affects the cost of audits and to understand when stratification increases statistical efficiency. McLaughlin and Stark (2011) compare the expected number of ballots that must be audited for proportionally stratified, optimally stratified, and unstratified audits using data from the 2008 U.S. House of Representatives contests in California. If MRO is the test statistic, optimal stratification can entail less hand counting than unstratified audits, depending on contest details. However, even optimal stratification tends to have a higher hand-counting workload than methods that sample batches with probability proportional to the amount of difference each batch can hold and that use a better test statistic than the maximum MRO.

It might be possible to reduce the audit workload for stratified audits (when the outcome is correct) by using a test statistic other than the maximum MRO or a maximum of monotone functions of the MRO. So far, there seems no analytically tractable, more powerful alternative for stratified random samples, but this is an area of active research.

In contrast, workload can be reduced dramatically (when the outcome is correct) by using smaller audit batches (Neff, 2003, Stark, 2010, McLaughlin and Stark, 2011). Unfortunately, most current vote tabulation systems do not report subtotals for batches smaller than precincts. Improving "data plumbing" to allow smaller batches to be audited—ideally, individual ballots—would be a powerful contribution to election integrity.

A Proof of [28]

Choose $x \in \mathbf{X}$, and let

$$\#_c x \equiv \sum_{k=1}^{N_c} x_{kc}$$

and

$$K_c(x) \equiv \min\left\{ k' \geq 0 : \sum_{k=1}^{k'} u_{kc} \geq \sum_{k=1}^{N_c} u_{kc} x_{kc} \right\}.$$

By [36] and the rearrangement theorem (Hardy et al., 1952), $K_c(x) \leq \#_c x$.

Let $\tilde{x} \equiv (\tilde{x}_{kc})_{k=1}^{N_c}\,{}_{c=1}^{C}$ be the vector with components

$$\tilde{x}_{kc} \equiv \begin{cases} 1, & k \leq K_c(x), \\ 0, & \text{otherwise}. \end{cases} \tag{64}$$

If $\tilde{x}_{kc} = 1$, then $u_{kc} > 0$, and by [35], $\omega_{kc} > t$. Let e^* be the allocation with components $e^*_{kc} = \omega_{kc}$ if $\tilde{x}_{kc} = 1$ and $e^*_{kc} = \omega_{kc} \wedge t$ if $\tilde{x}_{kc} = 0$. Then $e^* \in \tilde{\mathbf{E}}$ and $g(e) = \tilde{x}$. Hence,

$$\tilde{x} \in \tilde{\mathbf{X}}. \tag{65}$$

By definition of $K_c(x)$,

$$u \cdot \tilde{x} = \sum_{c=1}^{C} \sum_{k=1}^{N_c} u_{kc} \tilde{x}_{kc} \geq \sum_{c=1}^{C} \sum_{k=1}^{N_c} u_{kc} x_{kc} = u \cdot x. \tag{66}$$

Since $K_c(x) \leq \#_c x$ and $q_{kc} \geq 0$, it follows from [46] and the rearrangement theorem (Hardy et al., 1952) that

$$\begin{aligned} q \cdot \tilde{x} &= \sum_{c=1}^{C} \sum_{k=1}^{N_c} q_{kc} \mathbf{1}(k \leq K_c(x)) \leq \sum_{c=1}^{C} \sum_{k=1}^{N_c} q_{kc} \mathbf{1}(k \leq \#_c x) \\ &\leq \sum_{c=1}^{C} \sum_{k=1}^{N_c} q_{kc} x_{kc} = q \cdot x. \end{aligned} \tag{67}$$

By [65], [66], and [67], for any $x \in \mathbf{X}$ satisfying $u \cdot x \geq M$, there is a $y \in \tilde{\mathbf{X}}$ such that $u \cdot y \geq M$ and $q \cdot y \leq q \cdot x$. Equation [28] follows immediately.

B Branch and bound description

We describe a branch and bound algorithm for finding exact p-values by finding a vector $x^\dagger \in \tilde{\mathbf{X}} \subset \mathbf{X}$ that satisfies

$$q \cdot x^\dagger = \lambda = \min\{q \cdot x : u \cdot x \geq M,\ x \in \mathbf{X}\}.$$

The exact p-value is $P_\# = \exp(-q \cdot x^\dagger)$.

The branching step recursively splits the minimization problem into subproblems that fix the components of x corresponding to the first m elements of π (that is, they assign differences to the batches with the smallest values of r) and leave the remaining components free. Each branch is thus characterized by a vector $y^{\bullet m} \in \{0,1\}^m$, where m is the number of fixed components. For a given branch $y^{\bullet m}$, define x^{m0} to be the vector in $\mathbf{X}$ for which

$$x^{m0}_{\pi(j)} = \begin{cases} y^{\bullet m}_j, & j = 1, \ldots, m \\ 0, & \text{otherwise.} \end{cases}$$

That is, the components of x^{m0} corresponding to the smallest m values of r are equal to the corresponding values of $y^{\bullet m}$ and the rest of its components are zero. We call the elements $x^{m0}_{\pi(j)}$, $j = 1, \ldots, m$, the *fixed components* of x^{m0}, and the remaining $N - m$ elements the *free components*. Note that if $x^{m0} \notin \tilde{\mathbf{X}}$, then no $x \in \mathbf{X}$ with $x_{\pi(j)} = y^{\bullet m}_j$ is in $\tilde{\mathbf{X}}$.

Each branch $y^{\bullet m}$ satisfies one of four sets of conditions:

1. If $x^{m0} \in \tilde{\mathbf{X}}$ and $u \cdot x^{m0} \geq M$, then no vector x that agrees with with the fixed components of x^{m0} can have $q \cdot x < q \cdot x^{m0}$. In this case, x^{m0} is kept as a potential solution, the value of $q \cdot x^{m0}$ is saved, and the branch is not split further.
2. If $u \cdot x^{m0} < M$ and there is no $x \in \tilde{\mathbf{X}}$ that agrees with the fixed components of x^{m0} and has at least one additional component equal to 1, there is no way that splitting the branch will lead to a feasible element of $\tilde{\mathbf{X}}$. In this case, the branch is pruned.
3. Solving LKP for the free components shows that all vectors $x \in \mathbf{X}$ derived from this branch that satisfy $u \cdot x \geq M$ have a value of $q \cdot x$ greater than the smallest value saved in step 1. In this case, the branch is pruned.
4. If the branch does not satisfy any of conditions (1)–(3), it is split into two branches by extending $y^{\bullet m}$ to make two $\{0,1\}^{m+1}$-vectors, one with $m + 1^{\text{st}}$ component equal to 0 and the other with $m + 1^{\text{st}}$ component equal to 1. If no element of $\tilde{\mathbf{X}}$ matches the resulting fixed components, the corresponding branch is pruned.

Branches can be split at most 2^N times, so eventually each branch is pruned or satisfies condition set (1). Once that has happened, the solution to the original problem is the vector that satisfies condition set (1) and has the smallest value. We now explain the calculations in more detail.

The test in condition set (1) needs no explanation. The test in condition set (2) and the pruning in condition set (4) rely on a set of indicator variables $z \equiv$

$(z_c)_{c=1}^C$ for each branch. Initially, $z = (1)_{c=1}^C$. For any j with $y_j^{\bullet m} = 0$, $z_{\mathrm{c}(\pi(j))}$ is set to 0. If $z = (0)_{c=1}^C$ and $u \cdot x^{m0} < M$, the branch satisfies condition set (2) and is pruned.

Suppose a branch $y^{\bullet m}$ satisfies condition set (4). If $z_{\mathrm{c}(\pi(m+1))} = 0$, then the branch with 1 in its $m+1^{\mathrm{st}}$ component is pruned, because it can never lead to an element of $\tilde{\mathbf{X}}$.

We now discuss the lower bound used in condition set (3). For any vector $a \in \mathbb{R}^N$, and for any $m \in \{1,\ldots,N\}$, define ${}_m a \equiv (a_{\pi(j)})_{j=1}^m$. For any vector $y^{\bullet m} \in \{0,1\}^m$, define

$$\lambda^y \equiv \min\{q \cdot x : x \in \mathbf{X},\ {}_m x = y^{\bullet m}, u \cdot x \geq M\}.$$

That is, λ^y is the smallest value of $q \cdot x$ for vectors $x \in \mathbf{X}$ that satisfy $u \cdot x \geq M$ and have components $x_{\pi(j)} = y_j^{\bullet m}$, $j = 1,\ldots m$, or ∞ if no vector satisfies those constraints. This is the smallest value that can be obtained along the branch $y^{\bullet m}$.

If ${}_m u \cdot y^{\bullet m} \geq M$, then $\lambda^y = {}_m q \cdot y^{\bullet m}$. If ${}_m u \cdot y^{\bullet m} < M$, we can find a lower bound for λ^y by solving LKP in $\mathbb{R}^{N-m}$:

$$\lambda^y_{\mathrm{LKP}} \equiv \min\{q \cdot x : x \in \mathbf{X}^{rel},\ {}_m x = y^{\bullet m}, u \cdot x \geq M\} \leq \lambda^y.$$

For any $y^{\bullet m} \in \{0,1\}^m$, define

$$B^y \equiv (N+1) \wedge \left\{ B' \geq 1 :\ {}_m u \cdot y^{\bullet m} + \sum_{j=m+1}^{m+B'} u_{\pi(j)} \geq M \right\}.$$

Note that $B^y = 1$ when ${}_m u \cdot y^{\bullet m} > M$. When $B = N+1$, $\lambda^y_{\mathrm{LKP}} = \infty$. When $B \leq N$, the explicit solution for λ^y_{LKP} (Dantzig, 1957) is

$$\lambda^y_{\mathrm{LKP}} =_m q \cdot y^{\bullet m} + \sum_{j=m+1}^{m+B^y-1} q_{\pi(j)} + 0 \vee \left(M - {}_m u \cdot y^{\bullet m} - \sum_{j=m+1}^{m+B^y-1} u_{\pi(j)} \right) \frac{q_{\pi(m+B^y)}}{u_{\pi(m+B^y)}}.$$

Note that

$$M - \left({}_m u \cdot y^{\bullet m} + \sum_{j=m+1}^{m+B^y-1} u_{\pi(j)} \right) \leq 0$$

if and only if ${}_m u \cdot y^{\bullet m} \geq M$. If no $x \in \mathbf{X}$ with components $x_{\pi(j)} = y_j^{\bullet m}$, $j = 1,\ldots m$ satisfies $u \cdot x \geq M$, then $\lambda^y_{\mathrm{LKP}} = \infty$ and the branch $y^{\bullet m}$ is pruned.

We now give pseudo-code for a recursive branch and bound algorithm.

Initialize:
$x = (0)_{j=1}^N$

```
$z = (1)_{j=1}^{C}$.
$m = 0$.
$x^{\dagger'} = \text{NULL}$.
$\lambda^{\min} = \infty$.
```

The first three variables (x, z and m) are local; $x^{\dagger'}$ and $\lambda^{\min}$ are global.

When the algorithm stops, $x^{\dagger'} = x^{\dagger}$ and $\lambda^{\min} = \lambda$.

```
**BaB**$(x, z, m)$:
If $m \neq 0$:

    Set $y^{\bullet m} = {}_m x$.
    If ${}_m u \cdot y^{\bullet m} \geq M$:
        Subproblem can be trivially solved.
        If $\lambda^{\min} > {}_m q \cdot y^{\bullet m}$:
            Set $\lambda^{\min} = {}_m q \cdot y^{\bullet m}$.
            Set $x^{\dagger'} = x$.
        Return.
    Else If $z = (0)_{j=1}^{C}$:
        The only branches that lead to elements of $\tilde{\mathbf{X}}$ have $x_{\pi(m')} = 0, \forall m' > m$.
        Return.
    Else If $\lambda_{\text{LKP}}^{y} > \lambda^{\min}$ :
        This branch does not contain the minimum $\lambda$.
        Return.

If $z_{\text{c}(\pi(m+1))} = 1$:

    Set $x_{\pi(m+1)} = 1$.
    **BaB**$(x, z, m+1)$.
    Set $x_{\pi(m+1)}$ to 0 and $z_{\text{c}(\pi(m+1))}$ to 0.

**BaB**$(x, z, m+1)$.
Return.
```

C More general monotone weight functions

As mentioned above, the derivations generalize from the maximum MRO to the maximum of more general monotone weight functions of the observed differences by changing various definitions, as follows.

The test statistic T_w becomes the maximum of the weighted observed differences:

$$T_w \equiv \max_{(k,c) \in \mathbf{J}_{\vec{n}}} w_{kc}(e^H_{kc}).$$

The probability that the sample will show a maximum weighted difference no greater than any fixed value t if the allocation of difference is e is

$$P_{\mathbf{J}_{\vec{n}}}(e) \equiv P\left(\max_{(k,c) \in \mathbf{J}_{\vec{n}}} w_{kc}(e_{kc}) \leq t \right).$$

To construct an outcome-changing difference that is as hard as possible to detect, we rely on

$$\mathbf{G}(e) = \mathbf{G}(e;t) \equiv \{(k,c) : w_{kc}(e_{kc}) > t\}.$$

Within each stratum, instead of using condition [9], order the batches so that if $k > k'$ then

$$[\omega_{kc} - (\omega_{kc} \wedge w^{-1}_{kc}(t))] \geq [\omega_{k'c} - (\omega_{k'c} \wedge w^{-1}_{k'c}(t))].$$

Define

$$\kappa_c(e) \equiv \min \left\{ k' \geq 0 : \sum_{k=1}^{k'} \omega_{kc} + \sum_{k'+1}^{N_c} (\omega_{kc} \wedge w^{-1}_{kc}(t)) \geq \sum_{k=1}^{N_c} e_{kc} \right\},$$

$$\tilde{e}_{kc} \equiv \begin{cases} \omega_{kc}, & k \leq \kappa_c(e), \\ \omega_{kc} \wedge w^{-1}_{kc}(t), & \text{otherwise}, \end{cases}$$

$$u_{kc} \equiv \omega_{kc} - (\omega_{kc} \wedge w^{-1}_{kc}(t)),$$

and

$$M \equiv \mu - \sum_{c=1}^{C} \sum_{k=1}^{N_c} (\omega_{kc} \wedge w^{-1}_{kc}(t)).$$

Then the proofs go through *mutatis mutandis*.

References

Dantzig, G. (1957): “Discrete-variable extremum problems,” *Operations Research*, 5, 266–277.

Halvorson, M. and L. Wolff (2007): “Report and analysis of the 2006 post-election audit of Minnesotas voting systems,” `http://ceimn.org/files/CEIMNAuditReport2006.pdf`, retrieved 30 May 2011.

Hardy, G., J. Littlewood, and G. Pólya (1952): *Inequalities*, Cambridge, United Kingdom: Cambridge University Press, second edition.

Karp, R. (2010): “Reducibility among combinatorial problems,” in M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, eds., *50 Years of Integer Programming 1958–2008*, New York: Springer, 219–241.

McLaughlin, K. and P. Stark (2011): “Workload estimates for risk-limiting audits of large contests,” `http://statistics.berkeley.edu/~stark/Preprints/workload11.pdf`, retrieved 9 July 2011.

Miratrix, L. and P. Stark (2009): “The trinomial bound for post-election audits,” *IEEE Transactions on Information Forensics and Security*, 4, 974–981.

Neff, C. (2003): “Election confidence: A comparison of methodologies and their relative effectiveness at achieving it,” `http://www.verifiedvoting.org/downloads/20031217.neff.electionconfidence.pdf`, retrieved 6 March 2011.

Pisinger, D. (1995): *Algorithms for knapsack problems*, Ph.D. thesis, University of Copenhagen, Denmark.

Pisinger, D. and P. Toth (1998): “Knapsack problems,” in D.-Z. Du and P. Pardalos, eds., *Handbook of Combinatorial Optimization*, volume 1, Kluwer, Dordrecht, The Netherlands, 299–428.

Rivest, R. (2007): “On auditing elections when precincts have different sizes,” `http://people.csail.mit.edu/rivest/Rivest-OnAuditingElectionsWhenPrecinctsHaveDifferentSizes.pdf`, retrieved 30 May 2011.

Stark, P. (2008a): “Conservative statistical post-election audits,” *Ann. Appl. Stat*, 2, 550–581, URL `http://arxiv.org/abs/0807.4005`.

Stark, P. (2008b): “A sharper discrepancy measure for post-election audits,” *Ann. Appl. Stat.*, 2, 982–985, URL `http://arxiv.org/abs/0811.1697`.

Stark, P. (2009a): “CAST: Canvass audits by sampling and testing,” *IEEE Transactions on Information Forensics and Security*, 4, 708–717.

Stark, P. (2009b): “Efficient post-election audits of multiple contests: 2009 California tests,” http://ssrn.com/abstract=1443314, 2009 Conference on Empirical Legal Studies.

Stark, P. (2009c): “Risk-limiting post-election audits: *P*-values from common probability inequalities,” *IEEE Transactions on Information Forensics and Security*, 4, 1005–1014.

Stark, P. (2010): “Risk-limiting vote-tabulation audits: The importance of cluster size,” *Chance*, 23, 9–12.