Sharper p-Values For Stratified Election Audits

Michael J. Higgins[*]
Ronald L. Rivest[†]
Philip B. Stark[‡]

Abstract

Vote-tabulation audits can be used to collect evidence that the set of winners (the outcome) of an election 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.

[*]Department of Statistics, University of California, Berkeley
[†]Department of Electrical Engineering and Computer Science, MIT
[‡]Department of Statistics, University of California, Berkeley

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, 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. Most mandated post-election audits stop here.

In contrast, *risk-limiting audits* [Stark, 2008a,b; Miratrix and Stark, 2009; Stark, 2009a,c,b] 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 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.[2] As of this writing, there have been nine risk-limiting audits: eight in California (two in Marin County, three in Yolo County, and one each in Orange, Monterey,

[1]See `http://electionaudits.org/principles.html`.
[2]The definition of "risk-limiting" in the Colorado statute is incomplete.

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-limiting audits frame auditing as a sequential test of the hypothesis that the machine outcome is wrong. To reject that hypothesis is to conclude that the machine outcome is correct. Risk-limiting audits expand to a full hand count unless the sample gives strong evidence that the machine outcome is correct. The strength of the evidence is measured by the p-value of the hypothesis that the machine outcome is wrong: Smaller p-values are stronger evidence. 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. Defining "as little" amounts to specifying the test statistic for the hypothesis test. It can be defined in many ways; see, e.g., Stark [2009c]. A risk-limiting audit stops short of a full hand count only if the p-value becomes less than the risk limit α.

Risk-measuring audits are related to risk-limiting audits. They do not necessarily expand when the p-value is large. 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.[3] 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 provisional ballots. Audit samples might also be stratified by the type of machine used to count ballots.

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 observed value of the test statistic. When used as part of a risk-limiting audit, this can give large reductions in the expected counting burden when the machine outcome is correct.

[3] 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. 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.

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. For instance $f(x) \equiv x^2$ defines $f(x)$ to be x^2. If $a \equiv (a_j)_{j=1}^N$ and $b \equiv (b_j)_{j=1}^N$ are vectors of the same length N, then

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

"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.

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 (reported) vote total for candidate i in batch (k,c) is v_{kci}. The actual (what an audit would show) 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 $\#\mathbf{I}_W = W$. The apparent margin in votes between candidate $w \in \mathbf{I}_W$ and candidate $\ell \in \mathbf{I}_L$ is

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

(We are ignoring ties at the winner-loser boundary: There is no loser who was reported to have as many votes as any winner.) 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 $\ell \in \mathbf{I}_L$,

$$A_{w\ell} > 0, \tag{2}$$

or equivalently, if

$$V_{w\ell} - A_{w\ell} = \sum_{c=1}^{C} \sum_{k=1}^{N_c} [v_{kcw} - v_{kc\ell} - (a_{kcw} - a_{kc\ell})] < V_{w\ell}. \tag{3}$$

The apparent outcome is wrong if and only if (3) 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, if the apparent election outcome is wrong,

$$\sum_{c=1}^{C} \sum_{k=1}^{N_c} e^H_{kc} \geq \mu. \tag{4}$$

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]:

$$e^H_{kc} \equiv \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kc\ell} - (a_{kcw} - a_{kc\ell})}{V_{w\ell}}. \tag{5}$$

For the MRO, (4) holds with $\mu = 1$ if the apparent outcome is wrong.

If inequality (4) does not hold, the apparent outcome must be correct. Testing (4) statistically 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,

$$e^H_{kc} \leq \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kc\ell} + b_{kc}}{V_{w\ell}} \equiv \omega_{kc}. \tag{6}$$

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

$$\omega_{kc} \geq \omega_{k'c} \quad \text{if} \quad k < k'. \tag{7}$$

An *allocation of differences* or *allocation* is a vector $e = (e_{kc})_{k=1}^{N_c}\,{}_{c=1}^{C}$ that satisfies

$$e_{kc} \leq \omega_{kc}, \quad k = 1, \ldots, N_c, \; c = 1, \ldots, C. \tag{8}$$

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

$$\mathbf{E}_\mu \equiv \left\{ e \in \mathbf{E} : \sum_{c=1}^{C} \sum_{k=1}^{N_c} e_{kc} \geq \mu \right\}. \tag{9}$$

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[4] that the apparent outcome is incorrect, for stratified random samples. Let $\{\mathbf{J}^{n_c}_c\}_{c=1}^{C}$ be independent random samples, where $\mathbf{J}^{n_c}_c$ is a simple random sample of n_c elements from $\{(1,c), \ldots, (N_c,c)\}$. Let $\vec{n} \equiv (n_c)_{c=1}^{C}$, and let

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

Then $\mathbf{J}_{\vec{n}}$ is a stratified random sample of batches. Let $e = (e_{kc})_{k=1}^{N_c}\,{}_{c=1}^{C} \in \mathbf{E}$ be an allocation of differences. Define[5]

$$P_{\mathbf{J}_{\vec{n}}}(e) = P_{\mathbf{J}_{\vec{n}}}(e;t) \equiv P\left(\max_{(k,c) \in \mathbf{J}_{\vec{n}}} e_{kc} \leq t \right). \tag{10}$$

This is the probability that the maximum observed difference over the stratified random sample of batches $\mathbf{J}_{\vec{n}}$ will be no greater than t if the allocation of differences is e. We want to test the hypothesis that $e^H \in \mathbf{E}_\mu$ using

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

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.

The hypothesis $e^H \in \mathbf{E}_\mu$ does not completely specify the sampling distribution of t: That distribution depends on all the 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—that e^H is the element of $\mathbf{E}_\mu$ that maximizes

[4] `http://xkcd.com/892/`

[5] For monotone weight functions w_{kc} instead of the MRO,

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

the probability that t is small. The *exact p-value* of the hypothesis that the apparent outcome is wrong is

$$P_\# = P_\#(t;\vec{n}) \equiv \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e;t) \tag{11}$$

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

$$P_+ \geq P_\# \tag{12}$$

is a *conservative p-value*.

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

$$\mathbf{G}(e) = \mathbf{G}(e;t) \equiv \{(k,c) : e_{kc} > t\} \tag{13}$$

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

$$P_{\mathbf{J}_{\vec{n}}}(e) = \begin{cases} \prod_{c=1}^{C} \frac{\binom{N_c - \#_c\mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}}, & N_c - \#_c\mathbf{G}(e) \geq n_c \\ & c = 1,\ldots,C, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The exact p-value $P_\#$ (11) is the maximum of $P_{\mathbf{J}_{\vec{n}}}(e)$ over all allocations $e \in \mathbf{E}$.

For large cross-jurisdictional contests, finding the exact p-value by brute force is prohibitively expensive. The following sections show that (11) has special structure that allows us to find the exact p-value quickly even for large multi-jurisdictional contests.

3 Stratified audits and the 0-1 knapsack problem

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

$$P_\# \equiv \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e) = \max_{e \in \tilde{\mathbf{E}}_\mu} P_{\mathbf{J}_{\vec{n}}}(e). \tag{15}$$

We then show that maximizing $P_{\mathbf{J}_{\vec{n}}}$ over allocations in $\tilde{\mathbf{E}}_\mu$ can be couched as a 0-1 knapsack problem. The 0-1 knapsack problem is *NP*-complete [Karp, 2010]. However, even for large, multi-jurisdictional contests, the maximum can be found in a matter of seconds; 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}. The values $\kappa_c(e)$, defined below, limit how far we 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[6,7]

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

If $\omega_{kc} > \omega_{k'c}$ then

$$[\omega_{kc} - (\omega_{kc} \wedge t)] \geq [\omega_{k'c} - (\omega_{k'c} \wedge t)]. \tag{16}$$

Let $\mathbf{1}(k \in S)$ be the vector whose components are zero except when k is an element of the indicator function of the set S; the dimension of the vector is to be understood from context. It follows from the rearrangement theorem [Hardy et al., 1952], (16), and the fact that $e_{kc} \leq \omega_{kc}$

[6] For general monotone weight functions w_{kc} instead of the MRO,

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

[7] $\sum_{k=1}^{0} \omega_{kc} \equiv 0, \sum_{N_c+1}^{N_c} (\omega_{kc} \wedge t) \equiv 0.$

that

$$
\begin{aligned}
& \sum_{k=1}^{\#_c \mathbf{G}(e)} \omega_{kc} + \sum_{\#_c \mathbf{G}(e)+1}^{N_c} (\omega_{kc} \wedge t) \\
= & \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)] \mathbf{1}(k \le \#_c \mathbf{G}(e)) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\
\ge & \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)] \mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\
\ge & \sum_{k=1}^{N_c} [e_{kc} - t] \mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} (e_{kc} \wedge t) \\
= & \sum_{k=1}^{N_c} e_{kc}. \qquad (17)
\end{aligned}
$$

Thus, $\kappa_c(e) \le \#_c \mathbf{G}(e)$.

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[8]

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

Note that

$$
\tilde{e} \in \mathbf{E} \ \text{ and } \ \tilde{\tilde{e}} = \tilde{e}. \qquad (18)
$$

By definition of κ_c, $\sum_{k=1}^{N_c} \tilde{e}_{kc} \ge \sum_{k=1}^{N_c} e_{kc}$. Hence,

$$
\text{if } e \in \mathbf{E}_\mu \ \text{ then } \ \tilde{e} \in \mathbf{E}_\mu. \qquad (19)
$$

Moreover, for $c = 1, \ldots, C$, $\#_c \mathbf{G}(\tilde{e}) = \kappa_c(e) \le \#_c \mathbf{G}(e)$. Thus,

$$
P_{\mathbf{J}_{\vec{n}}}(\tilde{e}) \ge P_{\mathbf{J}_{\vec{n}}}(e). \qquad (20)
$$

Compared with the allocation e, the allocation $\tilde{e}$ 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 (18), (19), and (20) hold for all $e \in \mathbf{E}$, it follows that

$$
\max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(\tilde{e}) = \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e). \qquad (21)
$$

Thus, if we define

$$
\tilde{\mathbf{E}} \equiv \{\tilde{e} : e \in \mathbf{E}\}, \qquad (22)
$$

and let

$$
\tilde{\mathbf{E}}_\mu \equiv \tilde{\mathbf{E}} \cap \mathbf{E}_\mu, \qquad (23)
$$

then (15) 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 a 0-1 knapsack problem, as we now show.

[8] For general monotone weight functions w_{kc} instead of the MRO,

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

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

The 0-1 knapsack problem can be described with the following analogy. A shoplifter wishes to steal items with a total value of at least \$$M$ while minimizing his chance of being caught. Each item is described by two numbers: its monetary value and its contribution to the chance that the thief will be caught if he takes the item. We call the latter number the *cost* of the item, even though it is a contribution to a probability. For example, an expensive watch might be stored in a very secure location: It would have high value and high cost, because to take it would put the thief at high risk of being caught. Conversely, a pack of chewing gum has little monetary value, but might be easy to steal without getting caught: It would have low value and low cost. The 0-1 knapsack problem is to decide which items to steal.

We now write the knapsack problem more precisely. There are N items. Item j has value $u_j \ge 0$ and cost $q_j \ge 0$. Let $M \ge 0$ 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

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

A vector $x \in \mathbf{X}$ satisfying

$$
q \cdot x = \lambda \quad \text{and} \quad u \cdot x \ge M
$$

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

We will show that finding the exact p-value $P_\#$ amounts to solving KP.[9] To do so, we relate the constraint $u \cdot x \ge M$ 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

$$
\min_{e \in \tilde{\mathbf{E}}_\mu} \{ -\log(P_{\mathbf{J}_{\vec{n}}}(e)) \} = \min_{y \in \tilde{\mathbf{X}}} \{ q \cdot y : u \cdot y \ge M \}. \qquad (24)
$$

We then show that

$$
\min_{x \in \mathbf{X}} \{ q \cdot x : u \cdot x \ge M \} = \min_{y \in \tilde{\mathbf{X}}} \{ q \cdot y : u \cdot y \ge M \}. \qquad (25)
$$

[9] 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.

Any algorithm for solving KP can find the exact p-value. 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

$$j = k + \sum_{c'<c} N_{c'}, \quad k = 1, \ldots, N_c, \; c = 1, \ldots, C. \tag{26}$$

Conversely, the single index j corresponds to the double index k,c with[10]

$$c = \min\left\{ d : \sum_{i=1}^{d} N_d \geq j \right\}, \quad k = j - \sum_{d=1}^{c-1} N_d, \tag{27}$$

Recall that $\mathbf{G}(e)$ is the set of batches (k,c) for which $e_{kc} > t$ (13). For $e \in \mathbf{E}$, define

$$g_{kc}(e) \equiv \mathbf{1}((k,c) \in \mathbf{G}(e)), \tag{28}$$

$$g(e) \equiv (g_{kc}(e))_{k=1}^{N_c} {}_{c=1}^{C} \in \mathbf{X}, \tag{29}$$

and

$$\tilde{\mathbf{X}} \equiv \{y \in \mathbf{X} : y = g(e) \text{ for some } e \in \tilde{\mathbf{E}}\}. \tag{30}$$

Constraint: Let[11]

$$u_{kc} \equiv \omega_{kc} - (\omega_{kc} \wedge t). \tag{31}$$

Note that

$$u_{kc} = 0 \quad \text{if and only if} \quad \omega_{kc} \leq t. \tag{32}$$

By (7) and (16),

$$u_{kc} \geq u_{k'c} \quad \text{if} \quad k < k'. \tag{33}$$

Let[12]

$$M \equiv \max\left(\left[\mu - \sum_{c=1}^{C} \sum_{k=1}^{N_c} \omega_{kc} \wedge t \right], 0 \right). \tag{34}$$

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.

For $e \in \tilde{\mathbf{E}}$, note that $e \in \tilde{\mathbf{E}}_\mu$ if and only if [13,14]

$$u \cdot g(e) \geq M. \tag{35}$$

Thus,[15]

$$\{g(e) : e \in \tilde{\mathbf{E}}_\mu\} = \{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}. \tag{36}$$

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

$$\begin{aligned} P_{\mathbf{J}_{\vec{n}}}(e) &= \prod_{c=1}^{C} \frac{\binom{N_c - \#_c\mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}} \\ &= \prod_{c=1}^{C} \prod_{\#_c\mathbf{G}(e)>0} \prod_{k=1}^{\#_c\mathbf{G}(e)} \frac{\binom{N_c-k}{n_c}}{\binom{N_c-k+1}{n_c}} \\ &= \prod_{c=1}^{C} \prod_{\#_c\mathbf{G}(e)>0} \prod_{k=1}^{\#_c\mathbf{G}(e)} \frac{N_c - n_c - k + 1}{N_c - k + 1}. \end{aligned} \tag{37}$$

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. It follows from (37) that for any $e \in \mathbf{E}$,

$$P_{\mathbf{J}_{\vec{n}}}(e) = \prod_{c=1}^{C} \prod_{\#_c\mathbf{G}(e)>0} \prod_{k=1}^{\#_c\mathbf{G}(e)} \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \tag{38}$$

Let

$$p_{kc} \equiv \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \tag{39}$$

Note that

$$p_{kc} \geq p_{k'c} \quad \text{if} \quad k < k'. \tag{40}$$

If $e \in \tilde{\mathbf{E}}$, then[16]

$$P_{\mathbf{J}_{\vec{n}}}(e) = \prod_{c=1}^{C} \prod_{\#_c\mathbf{G}(e)>0} \prod_{k=1}^{\#_c\mathbf{G}(e)} p_{kc} = \prod_{c=1}^{C} \prod_{k=1}^{N_c} p_{kc}^{g_{kc}(e)}. \tag{41}$$

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[17]

$$q_{kc} \equiv -\log(p_{kc}); \tag{42}$$

[10] $\sum_{d=1}^{0} N_d \equiv 0$.

[11] For general monotone weight functions w_{kc} instead of the MRO, $u_{kc} \equiv \omega_{kc} - (\omega_{kc} \wedge w_{kc}^{-1}(t))$

[12] For general monotone weight functions w_{kc} instead of the MRO, $M \equiv \mu - \sum_{c=1}^{C} \sum_{k=1}^{N_c} (\omega_{kc} \wedge w_{kc}^{-1}(t))$.

[13] For $e \in \mathbf{E}$, $e \in \mathbf{E}_\mu$ only if (35) holds.

[14] Add $\sum_{c=1}^{C} \sum_{k=1}^{N_c} (\omega_{kc} \wedge t)$ to both sides of (35).

[15] We can assume that $\{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}$ is non-empty. Otherwise, by (36), it is impossible to have enough difference in the contest to make the apparent outcome and the correct outcome disagree; the p-value is 0.

[16] $p_{kc}^{g_{kc}(e)} \equiv 1$ when $p_{kc} = 0$ and $g_{kc}(e) = 0$.

[17] $q_{kc} = -\log(p_{kc}) \equiv \infty$ if $p_{kc} = 0$.

note that $q_{kc} \geq 0$ for all batches (k,c). By (40),

$$q_{kc} \leq q_{k'c} \quad \text{if} \quad k < k'. \tag{43}$$

From (41) and (42), for $e \in \tilde{\mathbf{E}}$,[18]

$$-\log(P_{\mathbf{J}_{\vec{n}}}(e)) = q \cdot g(e). \tag{44}$$

Equations (36) and (44) yield equation (24). We prove (25) in appendix A.

From (25), by monotonicity of $\exp(x)$,

$$\max_{e \in \tilde{\mathbf{E}}_\mu} P_{\mathbf{J}_{\vec{n}}}(e) = \exp(-\lambda). \tag{45}$$

Thus, finding $P_\#$ can be reduced to KP.

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 for exact solutions to $\tilde{\mathbf{X}}$ to improve efficiency. This algorithm can calculate the exact p-value in a matter of seconds, even for large elections. Code is available at

```
http://statistics.berkeley.edu/~stark/Code/
optStrat.r
```

Conservative p-values $P_+ \geq P_\#$ can be found even faster by solving the linear knapsack problem (LKP), the continuous relaxation of KP. The solution to LKP, the *LKP bound*, can be computed very quickly ($O(N)$ time by a partitioning algorithm [Pisinger and Toth, 1998]) and simply—in most cases with only a handheld calculator. We call the corresponding value of P_+ the *LKP conservative p-value*, P_{LKP}. For some election audits, P_{LKP} is almost exactly equal to $P_\#$, the exact p-value. (The branch and bound algorithm described in appendix B uses a variant of the LKP bound in the bound step.)

Define the continuous relaxation of $\mathbf{X}$:

$$\mathbf{X}^{rel} \equiv \left\{ (x_j)_{j=1}^N : x_j \in [0,1] \right\}. \tag{46}$$

The LKP is to find

$$\lambda_{\text{LKP}} \equiv \min_{x \in \mathbf{X}^{rel}} \{ q \cdot x : u \cdot x \geq M \}; \tag{47}$$

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

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

[18] $q_{kc}x_{kc} \equiv 0$ when $q_{kc} = \infty$ and $x_{kc} = 0$.

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

$$r_{kc} \equiv \frac{q_{kc}}{u_{kc}} \tag{48}$$

in increasing order, and find the smallest B so that the sum of the values of the first B batches is greater than M. The LKP bound is the sum of the first $B-1$ components and a fraction of the B^{th} component of the cost vector q with respect to the ordering of the ratios $(r_{kc})_{k=1}^{N_c}{}_{c=1}^{C}$.

We impose additional restrictions on the ordering of the ratios to simplify the descriptions of the algorithms in section 6.

Observe, by (33) and (43), that

$$r_{kc} \leq r_{k'c} \;\text{ if }\; k < k'. \tag{49}$$

Recall (26) and (27), the mappings between double indices and single indices. Choose a permutation $\pi : \{1,2,\ldots,N\} \to \{1,2,\ldots,N\}$ satisfying

1. $$r_{\pi(1)} \leq r_{\pi(2)} \leq \cdots \leq r_{\pi(n)}. \tag{50}$$

2. When $k < k'$ and[19] $r_{kc} = r_{k'c}$,
$$\pi\left(\sum_{c'<c} N_{c'} + k\right) < \pi\left(\sum_{c'<c} N_{c'} + k'\right). \tag{51}$$

3. When $r_{kc} = r_{k'c^*}$, $c \neq c^*$, and $N_c > N_{c^*}$,
$$\pi\left(\sum_{c'<c} N_{c'} + k\right) < \pi\left(\sum_{c'<c^*} N_{c'} + k'\right). \tag{52}$$

Note that for any $j' \in \{1,\ldots,N\}$ with $u_{j'} > 0$, the vector[20]

$$\left(\mathbf{1}[\pi(j) \leq j']\right)_{j=1}^N \in \tilde{\mathbf{X}}. \tag{53}$$

That is, putting ones in the early positions in π and zeros in the later positions corresponds to having differences greater than t in the batches (k,c) with the largest upper bounds ω_{kc} within each stratum c.

Define[21]

$$B \equiv \min\left\{ B' > 0 : \sum_{j=1}^{B'} u_{\pi(j)} > M \right\}$$

[19] By (49) and (50), (51) holds when $k < k'$ and $r_{kc} < r_{k'c}$.

[20] Consider the allocation e^* with components

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

By (51), if $\pi(\sum_{c'<c} N_{c'} + k') \leq j'$ and $k < k'$, then $\pi(\sum_{c'<c} N_{c'} + k) < j'$. That is, if $e^*_{k'c} = \omega_{k'c}$ and $k < k'$, then $e^*_{kc} = \omega_{kc}$. By (32), $e^* \in \tilde{\mathbf{E}}$ and $g(e^*) = (\mathbf{1}[\pi(j) \leq j'])_{j=1}^N$. Thus, it follows that $(\mathbf{1}[\pi(j) \leq j'])_{j=1}^N \in \tilde{\mathbf{X}}$.

[21] $B \equiv 1$ when $M = 0$ and $u_{\pi(1)} = 0$.

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 $u_{\pi(B)} > 0$. Dantzig [1957] shows that[22]

$$\lambda_{\text{LKP}} = \sum_{j=1}^{B-1} q_{\pi(j)} + \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}} \cdot q_{\pi(B)}. \tag{54}$$

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

$$x^{rel}_{\pi(j)} \equiv \begin{cases} 1, & \pi(j) < B, \\ \dfrac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}, & \pi(j) = B, \\ 0, & \text{otherwise.} \end{cases} \tag{55}$$

Observe that $u \cdot x^{rel} = M$, and $\lambda_{\text{LKP}} = 0$ when $M = 0$. If $\sum_{j=1}^{B-1} u_{\pi(j)} = M$, then

$$x^{rel} = (\mathbf{1}[\pi(j) \le B-1)])_{j=1}^{N} \in \mathbf{X}$$

actually solves KP, not just LKP.

Since $\sum_{j=1}^{B} u_{\pi(j)} > M$ and $(\mathbf{1}[\pi(j) \le B)])_{j=1}^{N} \in \mathbf{X}$ it follows that

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

is an upper bound for λ.

Observe that

$$\lambda_{\text{LKP}} + \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} = \lambda^{+}_{\text{LKP}}. \tag{56}$$

Thus,

$$\begin{aligned} \lambda - \lambda_{\text{LKP}} &\le \lambda^{+}_{\text{LKP}} - \lambda_{\text{LKP}} \\ &= \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} \le q_{\pi(B)}, \end{aligned} \tag{57}$$

and so

$$\frac{\exp(-\lambda_{\text{LKP}})}{\exp(-\lambda)} \le \frac{1}{p_{\pi(B)}}. \tag{58}$$

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

[22] $\sum_{j=1}^{0} q_{\pi(j)} \equiv 0, \sum_{j=1}^{0} u_{\pi(j)} \equiv 0$.

$$\frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}} q_{\pi(B)} \equiv 0$$

when $M = 0, u_{\pi(1)} = 0$, and $q_{\pi(1)} \in [0, \infty]$.

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.[23]

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].

[23] For more information about the election and audit, see Halvorson and Wolff [2007].

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

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.

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: The new methods 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. The amount of hand counting required to confirm an outcome that is correct depends on these sample sizes. 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. The next section shows that selecting sample sizes to be proportional to the number of batches in each stratum performs quite well in actual elections.[24]

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 by sampling as in $\vec{n}$ will be less than α. That is,

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

[24]This is how California currently sets sample sizes.

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

$$\sum_{c=1}^C n_c^\dagger = \min \left\{ \sum_{c=1}^C n_c : (n_c)_{c=1}^C \in \mathbf{N}(\alpha, t^*) \right\}. \qquad (59)$$

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.

Other criteria for optimality make sense, too. For instance, we might define a vector of sample sizes to be optimal if it minimizes the expected number of ballots that must be counted[25] 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^*.[26] If batches are about the same size, a sample size vector that minimizes the number of batches will also minimize the expected number of ballots audited. In practice, there are costs associated with retrieving batches of ballots and with hand-tabulating 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 depending 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 will 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 the sample size vector selected by the core method satisfies $P_\#(t^*; \vec{n}) \le \alpha$.

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

[25]If the vector of sample sizes is $\vec{n}$, the expected number of ballots that need to be hand counted is

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

[26]We could define optimality allowing t^* to be random (for instance, based on a hypothetical allocation of difference), taking into account the costs of expanding the audit if the p-value is larger than α.

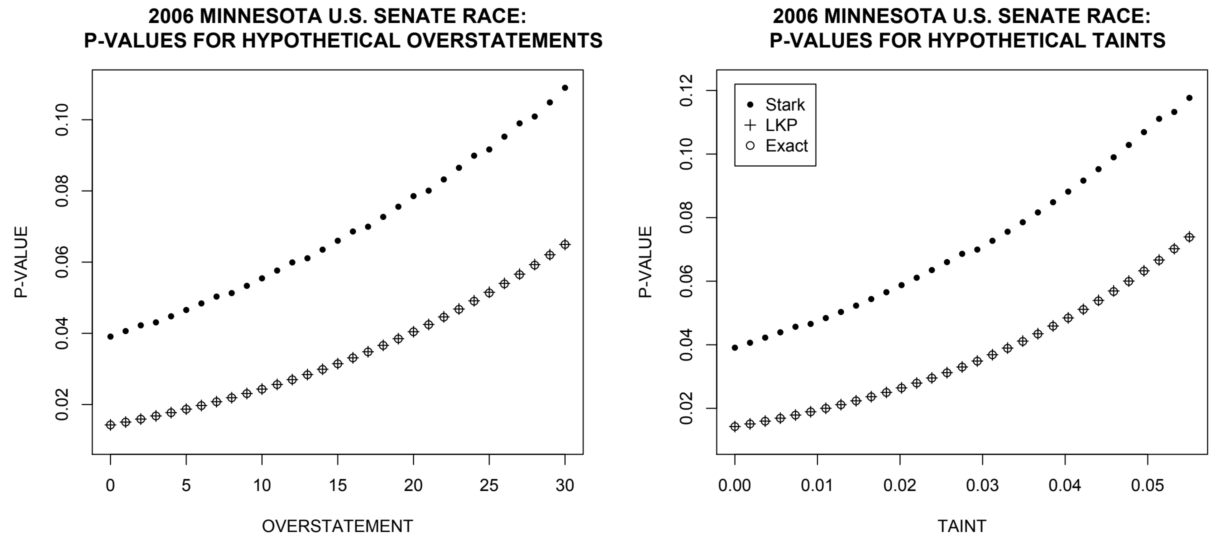

Figure 1: Exact and conservative p-values for hypothetical maximum observed overstatements (left) and maximum observed taints (right) for the 2006 Minnesota Senate race. The LKP conservative p-values (P_{LKP}) are nearly identical to the exact p-values ($P_{\#}$). Both are substantially smaller than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

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 $n\frac{N_c}{N}$ is an integer for all $c = 1, \ldots, C$, the PSS sample sizes are $n_c = n\frac{N_c}{N}$. When $n\frac{N_c}{N}$ is not an integer for some c, define PSS sample sizes to be $n_c = \lceil n\frac{N_c}{N} \rceil$. PSS sample sizes satisfy[27] $\sum_{c=1}^{C} n_c \geq n$.

6.2 first.r and next.r

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

The algorithms are similar. 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 (42) are updated based on the current value of $\vec{n}$.[28] The ratios $(r_j)_{j=1}^{N}$ are updated, and the permutation π that sorts these ratios into increasing order is found.[29] 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 $\vec{n} = (n_c)_{c=1}^{C} = (0)_{c=1}^{C}$.

[27] We could instead define PSS sample sizes $(n_c)_{c=1}^{C}$ to satisfy $\sum_{c=1}^{C} n_c = n$, as follows: For $k = 1, \ldots, N_c$, $c = 1, \ldots, C$, let $f_{kc} \equiv (k-1)\frac{N}{N_c}$. Let $f^{(j)}$ denote the jth smallest value of $(f_{kc})_{k=1}^{N_c}{}_{c=1}^{C}$. If $f_{kc} = f_{k'c'}$, let f_{kc} be ordered before $f_{k'c'}$ if $N_c > N_{c'}$. Set

$$n_c = \#\left\{ f_{kc} \in \left\{ f^{(1)}, \ldots, f^{(n)} \right\} \right\}.$$

[28] The `next.r` algorithm requires the cost to be defined slightly differently:

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

This only becomes an issue if more than half of the batches in a stratum need to be sampled, which can occur in a closely contested race.

[29] The permutation π is not affected by the modified definition of q_{kc} in 28, since $q_{kc} = \infty$ if and only if $k > N_c - n_c$. For $k \leq N_c - n_c$, by (43),

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

with equality if and only if $k = N_c - n_c$. Thus, the ordering in (43) continues to hold.

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 (50), (51), and (52).

Step 3: (Choose which n_c to increment)

Find $j = \min\{j' : \pi(j') \in \mathbf{S}\}$.
Using (27), find the double index (k,c) corresponding to the single index $\pi(j)$.
Increment n_c for that c.

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?)

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

By (50) and (51), 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[30] $P_\#(\vec{n}, t^*)$ on the assumption that the observed value of the test statistic is t^*.
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.

[30] A conservative p-value $P_+(\vec{n}, t^*) \geq P_{\mathbf{J}_{\vec{n}}}(e; t^*)$ could be used instead of the exact p-value.

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.

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 PSS,[31] `first.r`, or `next.r`. The calculations assume that the observed value of the test statistic would be the same. The results are in Table 2, along with the expected number of ballots to tally by hand.

All three methods require auditing dramatically fewer batches and ballots: 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.[32] The 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.

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.[33] The SWDB does not include California 1% audit findings.

[31] We use the approximate PSS sample sizes as detailed in footnote 27.
[32] See footnote 25.
[33] Data available at `http://swdb.berkeley.edu/pub/data/G08/state/state_g08_sov_data_by_g08_svprec.dbf`

		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

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.

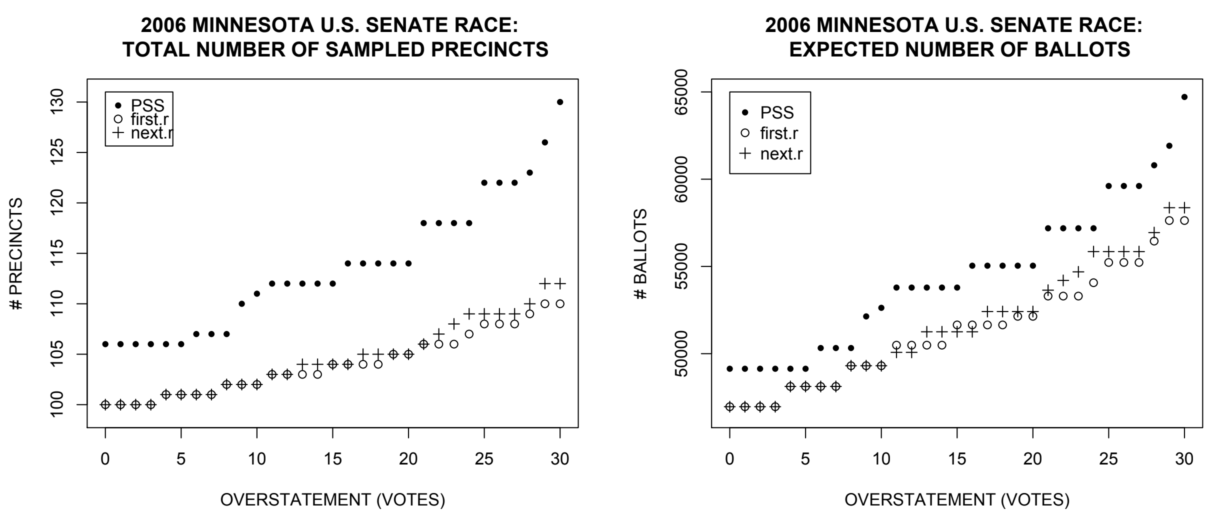

Figure 2: Number of batches to audit and expected number of ballots to audit to get p-values no larger than 0.05 for 2006 Minnesota Senate race, as observed maximum overstatements range from 0 to 30 votes, using sample size vectors selected by PSS, `first.r`, and `next.r`. In these simulations, PSS requires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

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 on the assumption that the audit does not uncover any overstatement of the margin (that is, sample size vectors in $N(0.05, 0)$).[34] 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. Figure 3 plots 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 or 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. 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.

Previous work on risk-limiting audits based on stratified samples found upper bounds on p-values that were weak 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), using the maximum relative overstatement of pairwise margins (MRO [Stark, 2008b]) as the test statistic. KP can be solved efficiently using a branch and bound algorithm. The linear knapsack problem (LKP) bound gives an extremely inexpensive upper bound on the p-value that is almost sharp: In examples for Minnesota and California, 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 for the 2006 U.S. Senate contest 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 observed error remained the same. 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 the process of setting up an audit a bit more complex, as we have seen in this paper. It would be interesting to study how much stratification adds to the logistical cost of audits and to understand the circumstances in which 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 error each batch can hold and that use a better test statistic than the 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 MRO or a monotone function 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 au-

[34] We exclude precincts (k, c) with $\omega_{kc} = 0$, because they cannot contribute overstatement errors.

	Race summary					Precincts to audit				Expected ballots to audit		
District	Strata	Precincts	Largest stratum	Votes	Margin (%)	PSS	`first`	`next`	Opt	PSS	`first`	`next`
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

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`.

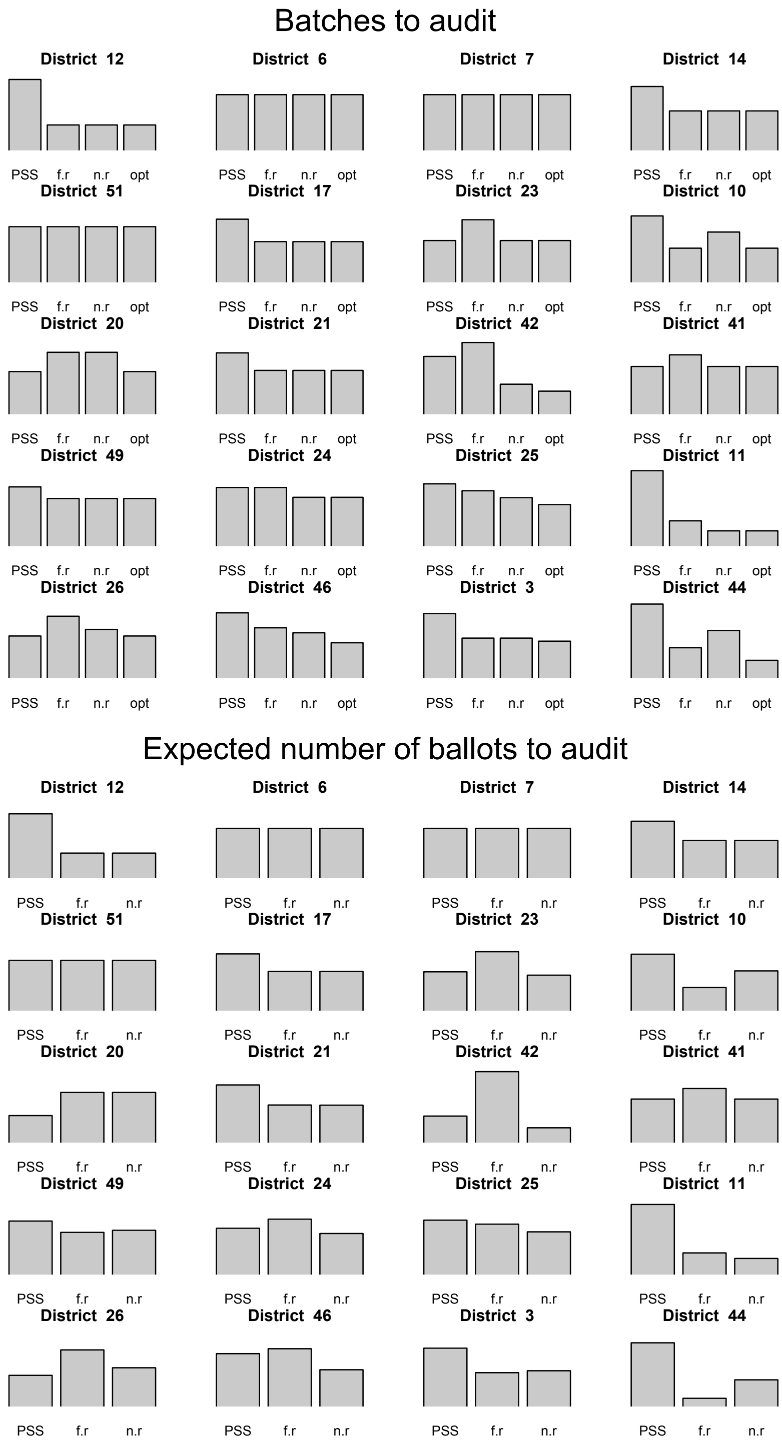

Figure 3: Number of batches to audit and expected number of ballots to audit for sample size vectors in $\mathbf{N}(0.05, 0)$. Bar graphs for the number of batches plot the ratio of the number of batches to audit for sample size vectors chosen using PSS, `first.r`, and `next.r` to the number of batches an optimal sample-size vector requires. Bar graphs for expected number of ballots to audit 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 batches than PSS; however, for many contests, the differences among methods are small.

dit 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.

9 Acknowledgments

We are grateful to Luke Miratrix, James J. Higgins, and Katherine McLaughlin for helpful conversations. 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.

A Proof of (25)

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

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

and

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

where we define $\sum_{k=1}^{0} u_{kc} \equiv 0$. By (33) 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

$$\tilde{x}_{kc} \equiv \begin{cases} 1, & k \leq K_c(x), \\ 0, & \text{otherwise.} \end{cases} \tag{60}$$

If $\tilde{x}_{kc} = 1$, then $u_{kc} > 0$, and by (32), $\omega_{kc} > t$. 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,

$$\tilde{x} \in \tilde{\mathbf{X}}. \tag{61}$$

By definition of $K_c(x)$,

$$u \cdot \tilde{x} = \sum_{c=1}^{C}\sum_{k=1}^{N_c} u_{kc}\tilde{x}_{kc} \geq \sum_{c=1}^{C}\sum_{k=1}^{N_c} u_{kc}x_{kc} = u \cdot x. \tag{62}$$

Since $K_c(x) \leq \#_c x$ and $q_{kc} \geq 0$, it follows from (43) and the rearrangement theorem [Hardy et al., 1952] that

$$\begin{aligned} q \cdot \tilde{x} &= \sum_{c=1}^{C}\sum_{k=1}^{N_c} q_{kc}\mathbf{1}(k \leq K_c(x)) \\ &\leq \sum_{c=1}^{C}\sum_{k=1}^{N_c} q_{kc}\mathbf{1}(k \leq \#_c x) \\ &\leq \sum_{c=1}^{C}\sum_{k=1}^{N_c} q_{kc}x_{kc} = q \cdot x. \end{aligned} \tag{63}$$

By (61), (62), and (63), 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 (25) 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 largest 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 components that have been assigned fixed values. For a given branch $y^{\bullet m}$, define x^{m0} to be the vector in $\mathbf{X}$ for which

$$x^{m0}_{\pi(j)} = \begin{cases} y^{\bullet m}_j, & j = 1, \ldots, m \\ 0, & \text{otherwise.} \end{cases} \tag{64}$$

That is, the components of x^{m0} corresponding to the largest 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 there exists some $x \in \tilde{\mathbf{X}}$ for which $x_{\pi(j)} = y^{\bullet m}_j$, $j = 1, \ldots, m$, then $x^{m0} \in \tilde{\mathbf{X}}$.

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, 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 every vector x derived from this branch has a value greater than the smallest value the algorithm has found so far. 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 (1). Once that has happened, the solution to the original problem is the vector that satisfies (1) and has the smallest value. We now explain the calculations in more detail.

The test in condition (1) needs no explanation. The test in condition (2) and the pruning in condition (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_c is set to zero for the stratum c corresponding to the index $\pi(j)$. If $z = (0)_{c=1}^C$ and $u \cdot x^{m0} < M$, the branch satisfies condition (2) and is pruned.

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

We now discuss the lower bound used in condition (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 $x_{\pi(j)} = y_j^{\bullet m}$, $j = 1,\ldots m$. 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 $\mathbb{R}^{N-m}$:

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

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

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

Note that $B^y = 1$ when ${}_m u \cdot y^{\bullet m} > M$. The explicit solution [Dantzig, 1957] is

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

where $\sum_{j=m+1}^m q_{\pi(j)} \equiv 0$ and $\sum_{j=m+1}^m u_{\pi(j)} \equiv 0$. 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 ${}_m u \cdot y^{\bullet m} \geq M$.

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

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

The first three variables 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 $\lambda^y_{\text{LKP}} > \lambda^{\min}$:

This branch does not contain the minimum λ.
Return.

Find c corresponding to the index $\pi(m+1)$.
If $z_c = 1$:

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

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

References

Dantzig, G. (1957). Discrete-variable extremum problems. *Operations Research*, 5(2):266–277.

Halvorson, M. and Wolff, L. (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., Littlewood, J., and Pólya, G. (1952). *Inequalities*. Cambridge University Press, second edition.

Karp, R. (2010). Reducibility among combinatorial problems. *50 Years of Integer Programming 1958-2008*, pages 219–241.

McLaughlin, K. and Stark, P. (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 Stark, P. (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. and Toth, P. (1998). *Handbook of Combinatorial Optimization*, volume 1, chapter Knapsack problems, pages 299–428. Kluwer, Dordrecht, The Netherlands.

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(2):550–581.

Stark, P. (2008b). A sharper discrepancy measure for post-election audits. *Ann. Appl. Stat.*, 2:982–985.

Stark, P. (2009a). CAST: Canvass audits by sampling and testing. *IEEE Transactions on Information Forensics and Security*, 4(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(3):9–12.