

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Applications of Integer Programming Methods to Solve Statistical Problems

Permalink

<https://escholarship.org/uc/item/1q55c77q>

Author

Higgins, Michael

Publication Date

2013

Peer reviewed|Thesis/dissertation

Applications of Integer Programming Methods to Solve Statistical Problems

By

Michael James Higgins

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Jasjeet Sekhon, Co-chair

Deborah Nolan, Co-chair

Kevin Quinn

Spring 2013

Applications of Integer Programming Methods to Solve Statistical Problems

Copyright 2013
by
Michael James Higgins

Abstract

Applications of Integer Programming Methods to Solve Statistical Problems

by

Michael James Higgins

Doctor of Philosophy in Statistics

University of California, Berkeley

Jasjeet Sekhon, Co-chair

Deborah Nolan, Co-chair

Many problems in statistics are inherently discrete. When one of these problems also contains an optimization component, integer programming may be used to facilitate a solution to the statistical problem. We use integer programming techniques to help solve problems in the following areas: optimal blocking of a randomized controlled experiment with several treatment categories and statistical auditing using stratified random samples.

We develop a new method for blocking in randomized experiments that works for an arbitrary number of treatments. We analyze the following problem: given a threshold for the minimum number of units to be contained in a block, and given a distance measure between any two units in the finite population, block the units so that the maximum distance between any two units within a block is minimized. This blocking criterion can minimize covariate imbalance, which is a common goal in experimental design. Finding an optimal blocking is an NP-hard problem. However, using ideas from graph theory, we provide the first polynomial time approximately optimal blocking algorithm for when there are more than two treatment categories. In the case of just two such categories, our approach is more efficient than existing methods. We derive the variances of estimators for sample average treatment effects under the Neyman-Rubin potential outcomes model for arbitrary blocking assignments and an arbitrary number of treatments.

In addition, statistical election 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 than those found by previous work can be found by solving a 0-1

knapsack problem. We also give algorithms for choosing how many batches to draw from each stratum to reduce the counting burden.

To Mom, Dad, and Tracie.
Thank you for your infinite love and support.

Contents

List of Figures	iv
List of Tables	vi
Acknowledgments	vii
1 Introduction	1
1.1 Optimal blocking	2
1.2 Election auditing	2
2 Optimal Blocking by Minimizing the Maximum Within-Block Distance	4
2.1 Introduction	4
2.2 Optimal blocking	5
2.2.1 Optimal blocking as a graph theory problem	6
2.2.2 Notation and framework	7
2.2.3 An algorithm for approximately optimal partitions	10
2.2.4 Improvements to approximately optimal partitions	13
2.3 Blocking estimators under Neyman-Rubin model	14
2.3.1 Notation and preliminaries	14
2.3.2 Parameters of interest and estimators	15
2.3.3 Estimating the variance	18
2.3.4 Comparing block randomization and complete randomization	20
3 Sharper p-Values for Stratified Election Audits	23
3.1 Introduction	23
3.2 Audits using stratified simple random samples	25
3.2.1 Notation and framework	25
3.2.2 Computing the p -value	27
3.3 Stratified audits and the 0-1 knapsack problem	29
3.3.1 Characterizing optimal allocations of differences	29
3.3.2 Maximizing $P_{J_{\bar{n}}}$ as a 0-1 knapsack problem	31
3.4 Approximate and exact solutions to KP	34
3.5 Results: comparing p -values	36

3.6	Selecting sample sizes	37
3.6.1	Sample sizes proportional to stratum size	39
3.6.2	first.r and next.r	40
3.6.3	Constructing sample size vectors in $\mathbf{N}(\alpha, t^*)$	41
3.7	Sample sizes for Minnesota and California contests	41
3.7.1	The 2006 Minnesota U.S. Senate race	41
3.7.2	The 2008 California U.S. House of Representatives races	42
3.8	Conclusions and Future Work	47
Bibliography		48
A Appendix for Chapter 2		50
A.1	Proof of Lemmas 6 and 7	50
A.2	Proof of Lemma 10	59
A.3	Proof of Theorem 11	63
B Appendix for Chapter 3		67
B.1	Proof of [3.28]	67
B.2	Branch and bound description	68
B.3	More general monotone weight functions	71

List of Figures

- 2.1 The figure on the left is a plot of heights and weights for twelve hypothetical people. The figure on the right is a plot of these same twelve people when viewed as vertices in a graph; all information about the original heights and weights is summarized by the values of the edge weights. 6
- 2.2 The figure on the left gives the best matched-pair blocking for the graph of twelve heights and weights in Figure 2.1. The figure on the right shows the optimal blocking with size threshold $t = 2$ for the same graph. Edge weights are Mahalanobis distances of heights and weights between the units. Note that the maximum within-block distance for the graph on the right is smaller than that for the graph on the left. 8
- 2.3 We demonstrate our approximate algorithm for the graph in figure 2.1. Each block of the partition obtained by the algorithm is denoted by a unique color. The upper-left graph is the bottleneck subgraph used in Step 5. The highlighted vertices in the upper-right graph are a maximal independent set of vertices for this subgraph. The lower-right graph demonstrates blooming from the maximal independent set in Step 6. The remaining unassigned vertex is assigned to the “green” block in the lower-right graph. The size threshold $t = 2$, and edge weights are Mahalanobis distances of heights and weights between units. Note that, for these twelve observations, the approximate algorithm yields the desired optimal blocking in figure 2.2. In general, this will not be the case. 12
- 3.1 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). 38
- 3.2 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. 43

-
- 3.3 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. 45
- 3.4 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. 46

List of Tables

- 3.1 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. 37
- 3.2 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 3.1 (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. 42
- 3.3 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`. 44

Acknowledgments

I am deeply indebted to my co-chairs, Professors Jasjeet Sekhon and Deborah Nolan, for their guidance, assistance, and many helpful conversations over the last few years. I am absolutely convinced that they are, in fact, saints, and this dissertation could not have been completed without their help.

I would like to thank my dad, Professor Emeritus Jim Higgins, for proofreading papers, discussing statistical problems, and sending his support over Skype. Sorry for worrying you so much these past seven years. I would also like to thank my mom, Mary Higgins, my brother, Professor Eric Higgins, and my fiance, Tracie Vittore, for their equally unwavering infinite amount of support.

I would like to thank Professor Philip Stark for introducing me to my current research agenda and for teaching me a lot about statistics and academic writing. I am grateful to Professor Ron Rivest, Hua Yang, Katie McLaughlin, and John McCarthy for their help with various research projects. I would also like to thank my outside member, Professor Kevin Quinn.

I would like to thank my colleagues, including (but not limited to) Derek Bean, Adam Bloniarz, Margaret Boittin, Miguel DeFigueiredo, Megan Goldman, Chris Haulk, John Henderson, Chingway Lim, James Long, Luke Miratrix, Peter Ralph, Karl Rohe, Sean Ruddy, Dave Shilane, and Allan Sly, for their many fruitful conversations and their help in keeping me sane throughout graduate school. Brad Luen deserves special mention for his innumerable “best-of” lists, for his encyclopedic knowledge of Bay Area restaurants, and for being a tremendous friend.

I would like to thank the Kansas State Wildcats football and basketball teams for giving me something to ardently cheer for while at UC Berkeley. Go Cats!

Finally, I would like to thank God, whose existence is now proven—my completion of this dissertation can only be considered a divine miracle.

This dissertation was typeset using the ucastrothesis L^AT_EX template.

Chapter 1

Introduction

Statistics features many problems that are discrete in nature. Statistical sampling problems often begin with the assumption that an integer number of units is drawn from a finite population, with attributes of units considered fixed. Many problems in experimental design examine how to efficiently assign a finite number of units to pre-selected treatment categories. Nonparametric inference problems may involve computing ranks and counting the number of permutations of treatment assignments that would produce a test statistic as large or larger than the one observed under an exact null hypothesis. Frequently in causal inference, subjects in one treatment group are matched to one or more subjects in another treatment group based on the units' propensity scores or observed values of selected pretreatment covariates to improve accuracy of treatment effect estimates. Statistical clustering problems investigate methods to best partition a finite set of units.

The aforementioned areas have all benefitted from the use of integer programming techniques. *Integer programming* is a branch of combinatorial optimization that focuses on methods for optimizing objective functions over integer domains. Recognizing a statistical problem as an integer programming problem (e.g. as a graph partitioning problem or a knapsack problem) may facilitate the solution of the statistical problem.

We give two novel instances of solving a statistics problem through the use of integer programming. In chapter 2, we borrow from graph partitioning literature to obtain efficient methods for blocking units before applying treatment in an experiment. Our blocking method is applicable for experiments with an arbitrary number of treatment categories and an arbitrary number of replications of each treatment within each block. When treatment is randomized within each block, our blocking method will ensure some level of covariate balance between treatment groups. In chapter 3, we consider statistical election audits in which batches of ballots are selected for audit by a stratified random sample (which is commonly how contests that span multiple counties are audited). We show that a p -value for the null hypothesis that the machine count outcome is different from what a full hand count would show can be obtained by solving a 0-1 knapsack problem. This observation allows for a reduction in the audit workload (compared to previous methods) while preserving the same confidence in the election outcome, and can further be exploited to obtain optimal sample sizes for audits.

1.1 Optimal blocking

In randomized controlled experiments where treatment is completely randomized, especially in those with very few units, there may be a non-negligible probability of poor balance on a covariate that is highly predictive of response. When this happens, estimates of treatment effects may be inaccurate unless the covariate imbalance is taken into account. For example, consider a medical trial where patients are given either a medication or a placebo. If very sick people are disproportionately assigned to the placebo, and only healthy people receive the medication, it may be very difficult to accurately assess the true effectiveness of the medication. Moreover, current methods for adjusting for covariate imbalance when estimating treatment effects (e.g. post-stratification) may be susceptible to “data-mining” problems; it may be preferable to ensure that randomization is restricted to ensure balance on important covariates (Lock Morgan & Rubin 2012).

In chapter 2, based on work with Jasjeet Sekhon, we propose a method for blocking units before treatment is assigned. Given a pre-specified threshold for the minimum number of units to be contained in a block, and given a measure of dissimilarity between any two experimental units, our method selects a blocking so that the maximum dissimilarity between any two units within the same block is “close” to minimal. In this way, our blocking method will ensure a level of covariate balance between treatment groups. The method can be applied to experiments with an arbitrary many treatments, with arbitrarily many replications of each treatment within each block, and can be used for both small and fairly large experiments (up to about 10,000 units at this current time, with hopes of millions of units soon).

Our blocking method first uses an observation made by Paul Rosenbaum that units in an experiment can be viewed as vertices on a graph (Rosenbaum 1989). We then view blocking as a graph partitioning problem. The condition of minimizing the maximum within-block dissimilarity is closely related to the idea of bottleneck subgraphs (Gonzalez 1985; Hochbaum & Shmoys 1986). We exploit methods from this literature to develop an efficient blocking algorithm.

We then conclude the chapter with a discussion on estimation the sample average treatment effect of under the Neyman-Rubin potential outcomes model (Splawa-Neyman et al. 1990; Rubin 1974; Holland 1986). Under this model, we derive variances for two unbiased estimate of this parameter: the difference-in-means estimate and the Horvitz-Thompson estimate. We also discuss conservative estimates of these variances.

1.2 Election auditing

Statistical auditing uses techniques like statistical sampling and hypothesis testing to make rigorous statements about the amount of discrepancy contained in a set of units under audit. Often, this statistical rigor requires minimal assumptions to be made on the distribution of the discrepancy across the units; when the discrepancy is sufficiently large and deliberate, it will be allocated so as to make detection by an audit difficult. Thus, statistical auditing problems can become very hard very quickly.

There is a large literature on the subject of statistical financial auditing (Fienberg et al. 1977;

Guthrie et al. 1989; Wendell & Schmee 1996; Talens 2005; Financial Audit Manual 2008). More recently, many of the ideas in financial auditing have also been used in election auditing (Aslam et al. 2007; Stark 2008a,b, 2009a,b,c; McLaughlin & Stark 2011; Miratrix & Stark 2009). In chapter 3, originally published as Higgins et al. (2011) and reprinted with permission from De Gruyter¹, we discuss how to test whether an election outcome is incorrect when batches of ballots are drawn by a stratified random sample—a common way a contest is audited when it spans multiple counties.

The core of the paper presents an exact substantive test of details when units are selected using a simple random sample. A *substantive test of details* is a type of audit that tests whether the aggregate discrepancy in a set of audited units is *material*—greater than some prespecified threshold. In election auditing, a discrepancy in a batch of ballots is a difference between the reported vote total and the hand-count vote total for that batch, and the aggregate discrepancy is material when the reported winners differ from what a full hand count would show.

Although the chapter focuses on election audits, our method provides a novel test that is applicable for all types of audits (including financial audits). Our test only assumes that the discrepancy of a unit can be bounded from above before drawing a sample—in financial auditing, it is commonly assumed that the maximum discrepancy in a unit is no more than the book value of the unit; in election auditing, a count of the number of ballots can be used to obtain an upper bound on the maximum discrepancy. This work currently provides the sharpest one-sided nonparametric test for the population mean when units are drawn using stratified random sampling, and is currently the only such test that incorporates information about the values of the responses when deriving the sampling distribution of the test statistic.

We show that computing a p -value for this substantive test is equivalent to solving a 0–1 knapsack problem (KP). KP is one of Karp’s 21 NP-Complete problems (Karp 2010). However, the size of most auditing problems, and special structure related to stratified random sampling of units, allows for p -values to be compute in fractions of a second. Properties of KP can be exploited to derive good upper-bounds for this p -value very quickly and to improve methods for selecting sample sizes for an audit.

¹This article can be found at:

<http://www.degruyter.com/view/j/spp.2011.2.issue-1/issue-files/spp.2011.2.issue-1.xml>

Chapter 2

Optimal Blocking by Minimizing the Maximum Within-Block Distance

2.1 Introduction

Since each additional observation in an experiment sometimes comes at considerable cost, it is desirable to find more efficient estimators than the simple difference-in-means estimator to measure treatment effects. *Blocking*, which is when experimenters first partition their units into *blocks* and then randomize treatment within these blocks, can greatly reduce variance compared to the simple-difference estimator if the strata differ from each other. This idea goes back to Fisher (1926), and Neyman offers a treatment of blocking under his potential outcomes model in the appendix of Neyman (1935). In recent work, Abadie & Imbens (2008) and Imai (2008) analyze matched pairs under the Neyman model, Imai et al. (2009) do so for matched pairs of clusters, and Imbens (2011) considers various designs. We extend some of these results to the case of an arbitrary number of treatments. More importantly, we offer the first polynomial time approximately optimal blocking algorithm for when there are more than two treatment categories.¹ In the absence of such a computationally feasible blocking method, some researchers have turned to other approaches, such as re-randomization (Lock Morgan & Rubin 2012).

Consider a controlled experiment with n units and two or more treatments. Each unit is assigned to exactly one treatment. After applying treatment, a response for each unit is measured. These responses are used to make statements about the effectiveness of the treatments. Pretreatment covariates may be measured for each unit before treatment is assigned. Suppose the units are partitioned into blocks on the basis of some subset of covariates. The collection of blocks is called a *blocking*, the covariates used to create these groups are called *block covariates*, and the number of units contained in a block is called the *size* of the block.

In this paper, we consider the following method for blocking: Choose a set of block covariates. Select a metric that measures the similarity of the values of block covariates between any two

¹ In the case of only two treatments, researchers sometimes use optimal nonbipartite matching (e.g., Greevy et al. 2004; Lu & Rosenbaum 2004; Lu et al. 2011). Moore (2012) discusses a number of blocking methods such as optimal-greedy blocking.

units—e.g. Mahalanobis distance. Choose a threshold for the minimum block size. Block units so that the size of each block is at least this threshold, and so that the *maximum within-block distance*—the maximum distance between any two units within a block—is minimized. We will call any set of blocks that satisfy these criteria an *optimal blocking*.

When the goodness of a blocking is determined by how small the maximum within-block distance is, setting a threshold for the minimum block size seems preferable to fixing the number of units in a block (as is done, for example, in matched-pair designs), especially in small experiments. For instance, consider an experiment with a treatment group and a control group and six subjects, three men and three women. A matched pair design will necessarily pair one of the men to one of the women, which could lead to increased variability in estimated treatment effects if the sex of subject strongly affects response. An optimal blocking with a minimum block size of two, however, may have one block containing the three men and one block containing the three women.

Setting this threshold also seems preferable to fixing the number of blocks in the experiment, which can create very small blocks. Imbens (2011) advocates applying each treatment to at least two units within each block to estimate conditional variances. An optimal blocking with minimum block size equal to two times the number of treatments will allow for each treatment to be applied twice within each block. This treatment assignment suggestion may not be possible when fixing the number of blocks.

Finding an optimal blocking is an NP-hard problem (Hochbaum & Shmoys 1986). However, in small experiments, an optimal blocking can be found by brute force. Moreover, we give a polynomial time algorithm that can find a blocking with maximum within-block distance within a factor of four of the smallest maximum within-block distance possible.

We then discuss estimation of the sample average treatment effect (SATE) under the Neyman-Rubin Causal Model when units are blocked before treatment assignment. We derive variances for two unbiased estimators of the SATE—the difference-in-means estimator and the Horvitz-Thompson estimator—and give conservative estimates of these variances. We then give a description of when blocking is beneficial in controlled experiments.

2.2 Optimal blocking

Rosenbaum (1989) observed a connection between optimal matching in causal inference and matching in graph theory. We solve our optimal blocking problem by making a similar connection. Each experimental unit can be viewed as a vertex in a graph. If two units can be placed in the same block, there is a corresponding edge drawn between these two vertices in the graph. Weights are assigned to each edge in the graph; if two units have similar values of block covariates, the edge joining the corresponding vertices in the graph has a small weight. Thus, our desired optimal blocking, or approximately optimal blocking, can be found using existing machinery in graph theory.

For this paper, we focus on blockings where the size of each block is equal to or greater than some prespecified threshold. Of these blockings, we wish to find one which minimizes the maximum within-block distance. The algorithms presented only require the edge weights to satisfy

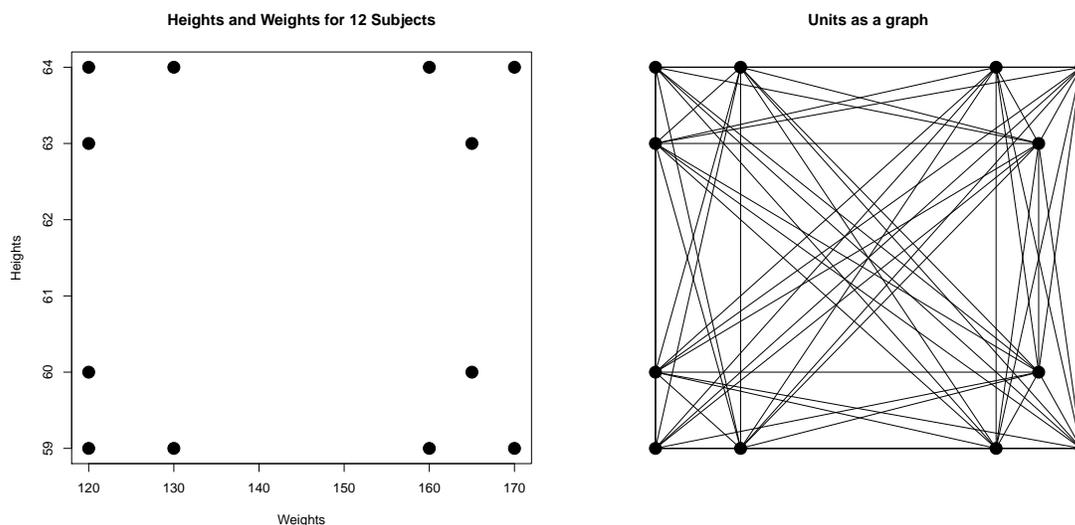


Figure 2.1: The figure on the left is a plot of heights and weights for twelve hypothetical people. The figure on the right is a plot of these same twelve people when viewed as vertices in a graph; all information about the original heights and weights is summarized by the values of the edge weights.

the triangle inequality (see Section 2.2.1); any metric applied to the block covariates can be minimized by our method. For example, edge weights may be Mahalanobis (or weighted Mahalanobis) distances of block covariates between units.

2.2.1 Optimal blocking as a graph theory problem

We begin by giving the following conventions for our notations. Lowercase letters will either denote constants or indices. Parameters of interest will always be denoted by greek letters. Estimates of these parameters are random variables denoted by caretted (\wedge) greek letters. Capital letters will either be used for graph theory notation or to denote random variables that are not parameter estimates. Vectors are denoted by bold lowercase letters. Sets are denoted by bold uppercase letters.

Let $G = (V, E)$ denote a graph, where V is a set of n vertices and E is a set of edges. Suppose that G is *complete*—between any two vertices $\{i\}, \{j\} \in V$, there is an edge $(i, j) \in E$ joining these two vertices. Suppose each edge (i, j) has a weight w_{ij} , and suppose that these weights satisfy the triangle inequality:

$$w_{ij} + w_{jk} \geq w_{ik} \quad \forall \{i\}, \{j\}, \{k\} \in V, \{i\} \neq \{j\} \neq \{k\}. \quad (2.1)$$

A *partition* of V is a collection of *blocks* of vertices (V_1, V_2, \dots, V_k) that separates the original set of vertices; each vertex $\{i\} \in V$ belongs to exactly one block V_ℓ . Formally, a partition (V_1, V_2, \dots, V_k) satisfies:

1. $V_\ell \subset V, \ell = 1, \dots, k$.
2. $V_\ell \cap V_{\ell'} = \emptyset$ when $\ell \neq \ell'$.
3. $\bigcup_{\ell=1}^k V_\ell = V$.

The *size* of a block V_ℓ , denoted $|V_\ell|$, is the number of vertices contained in that block. When vertices denote experimental units, each partition of V can be viewed as a blocking of units: two experimental units are in the same block if and only if their corresponding vertices are in the same block of the partition.

It follows that our original optimal blocking problem can be posed as the following optimal partition problem: Let \mathbf{V} denote the set of all partitions, let t denote a *size threshold*, and let

$$\mathbf{V}^t \equiv \{(V_1, \dots, V_k) \in \mathbf{V} : |V_\ell| \geq t, 1 \leq \ell \leq k\} \quad (2.2)$$

denote the set of partitions that have at least t vertices within each block. We are interested in finding a partition $\mathbf{v}^\dagger = (V_1^\dagger, V_2^\dagger, \dots, V_{k^\dagger}^\dagger) \in \mathbf{V}^t$ satisfying:

$$\max_{1 \leq \ell \leq k^\dagger} \max_{\{i\}, \{j\} \in V_\ell^\dagger} w_{ij} = \min_{(V_1, \dots, V_k) \in \mathbf{V}^t} \left(\max_{1 \leq \ell \leq k} \max_{\{i\}, \{j\} \in V_\ell} w_{ij} \right) \equiv \lambda. \quad (2.3)$$

That is, across all partitions that contain only blocks with t or more vertices, we are searching for the partition which minimizes the *maximum within-block edge weight*—the maximum weight of an edge that joins two vertices within the same block of the partition.

Finding such a partition \mathbf{v}^\dagger is NP-hard (Hochbaum & Shmoys 1986). However, when the set of vertices V is small, an optimal partition can be found by brute force. Also, as we now show, a polynomial-time algorithm can find an *approximately optimal* partition $(V_1^*, V_2^*, \dots, V_{k^*}^*)$ that has a maximum within-block edge weight of at most 4λ :

$$\max_{1 \leq \ell \leq k^*} \max_{\{i\}, \{j\} \in V_\ell^*} w_{ij} \leq 4\lambda. \quad (2.4)$$

In simulations, this approximately optimal partition can be found in a matter of seconds, even when the set of vertices is large.

To reduce ambiguity in future procedures and algorithms, we assume that there is an ordering on the vertex set V , and we make the convention that an edge between vertex $\{i\}$ and $\{j\}$ is written (i, j) if and only if $\{i\}$ is ordered before $\{j\}$. Under this convention, all procedures and algorithms are completely non-random given G . If edge weights are Mahalanobis distances between block covariates of units, for example, an ordering can be placed on vertices by ordering the distances from each unit's block covariates to the origin.

2.2.2 Notation and framework

We now introduce notation and structure that will be used to show approximate optimality of the partition produced by our algorithm. Our approach closely follows (Hochbaum & Shmoys 1986).

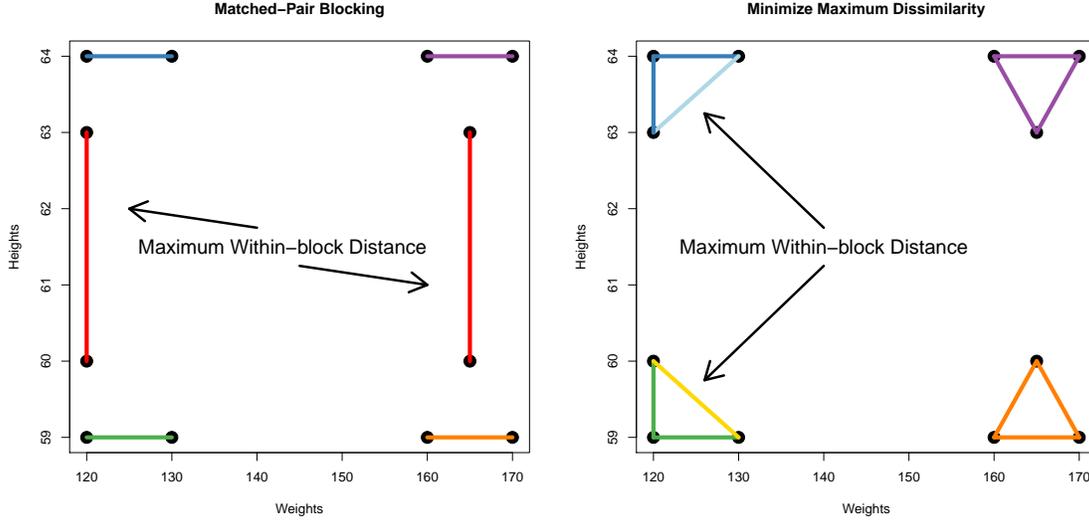


Figure 2.2: The figure on the left gives the best matched-pair blocking for the graph of twelve heights and weights in Figure 2.1. The figure on the right shows the optimal blocking with size threshold $t = 2$ for the same graph. Edge weights are Mahalanobis distances of heights and weights between the units. Note that the maximum within-block distance for the graph on the right is smaller than that for the graph on the left.

Let $G^* = (V, E^*)$ denote an arbitrary subgraph of G (Note: for this paper, all subgraphs G^* use the same vertex set V as the original complete graph G ; only edge sets E^* differ between subgraphs). Vertices $\{i\}$ and $\{j\}$ are *adjacent* in G^* if the edge $(i, j) \in E^*$. A set of vertices $\{\{i_1\}, \{i_2\}, \dots, \{i_m\}\} \subset V$ is *independent* in G^* if no vertices in V are adjacent to each other: $\nexists \{\ell\}, \{\ell'\} \in \{\{i_1\}, \{i_2\}, \dots, \{i_m\}\}$ such that $(i_\ell, i_{\ell'}) \in E^*$. This set is a *maximal independent set* in G^* if, for any additional vertex $\{i_{m+1}\} \in V$, the set $\{\{i_1\}, \dots, \{i_m\}, \{i_{m+1}\}\}$ is not independent: $\forall \{i_{m+1}\} \in V \setminus \{\{i_1\}, \{i_2\}, \dots, \{i_m\}\}, \exists \{\ell\} \in \{\{i_1\}, \{i_2\}, \dots, \{i_m\}\}$ such that $(i_{m+1}, i_\ell) \in E^*$.

The *degree* of a vertex $\{i\} \in V$ is the number of edges in E^* that have $\{i\}$ as an endpoint:

$$\deg(\{i\}) \equiv \#\{(i, j) \in E^* : j \in V, j \neq i\}. \quad (2.5)$$

Note that, in a complete graph on n vertices, each vertex has degree $n - 1$.

The r th power of G^* is a subgraph $(G^*)^r = (V, (E^*)^r)$ of G where the edge $(i_0, i_m) \in (E^*)^r$ if and only if there is a path from $\{i\}$ to $\{j\}$ in G^* with at most r edges:

$$(E^*)^r \equiv \{(i, j) : \exists i = i_0^*, i_1^*, \dots, i_{m-1}^*, i_m^* = j \text{ s.t. } (i_{q-1}^*, i_q^*) \in E^*, q = 1, \dots, m, m \leq r\}. \quad (2.6)$$

If $\{(i_{q-1}, i_q)\}_{q=1}^m$ is a path from $\{i\}$ to $\{j\}$ with m edges, then by (2.1),

$$w_{ij} \leq m \max_{1 \leq q \leq m} (w_{i_{q-1}i_q}). \quad (2.7)$$

Let $B_w(G) = (V, E_w)$ denote the *bottleneck subgraph of G of weight w* ; this subgraph has an edge $(i, j) \in E_w$ if and only if $w_{ij} \leq w$:

$$E_w \equiv \{(i, j) \in E : w_{ij} \leq w\}. \quad (2.8)$$

Since there are at most $n(n-1)/2$ different values of the edge weights w_{ij} , there are at most $n(n-1)/2$ different bottleneck subgraphs $B_w(G)$.

Given a partition $\mathbf{v} \in \mathbf{V}$, let $G(\mathbf{v}) = (V, E(\mathbf{v}))$ denote the *subgraph of G generated by \mathbf{v}* —the subgraph where $(i, j) \in E(\mathbf{v})$ if and only if vertices $\{i\}$ and $\{j\}$ are contained in the same block in the partition:

$$E(\mathbf{v}) \equiv \{(i, j) \in E : \exists V_\ell \in \mathbf{v} \text{ s.t. } \{i\}, \{j\} \in V_\ell\}. \quad (2.9)$$

Note that, if a block in the partition \mathbf{v} contains ℓ vertices, then every vertex $\{i\}$ contained in that block has $\deg(\{i\}) = \ell - 1$ in the subgraph $G(\mathbf{v})$.

We are now ready to state and prove some lemmas:

Lemma 1 *If $w \geq \lambda$, then the degree of each vertex in the bottleneck graph $B_w(G)$ is at least $t - 1$.*

Proof: Recall that each block of an optimal partition \mathbf{v}^\dagger has at least t vertices. Thus, in the subgraph $G(\mathbf{v}^\dagger)$, each vertex $\{i\} \in V$ has $\deg(\{i\}) \geq t - 1$.

By definition of λ , two vertices $\{i\}, \{j\}$ placed in the same block of \mathbf{v}^\dagger must have $w_{ij} \leq \lambda$. It follows that all edges (i, j) in the set $E(\mathbf{v}^\dagger)$ must have weight $w_{ij} \leq \lambda$. Thus, $G(\mathbf{v}^\dagger)$ is a subgraph of $B_\lambda(G)$; every edge in the set $E(\mathbf{v}^\dagger)$ is also in the set E_λ (and E_λ may have even more edges). Since, for each vertex $\{i\} \in V$, there are at least $t - 1$ edges in $E(\mathbf{v}^\dagger)$ that have $\{i\}$ as an endpoint, there must also be at least $t - 1$ edges in E_λ that have $\{i\}$ as an endpoint. That is, each vertex $\{i\} \in V$ has degree $\deg(\{i\}) \geq t - 1$ in $B_\lambda(G)$.

Similarly, if $w \geq \lambda$, then $B_\lambda(G)$ is a subgraph of $B_w(G)$, and by the same argument as above, each vertex $\{i\}$ in $B_w(G)$ has degree $\deg(\{i\}) \geq t - 1$. ■

The following corollary follows from this lemma:

Corollary 2 *If in $B_{w^*}(G)$ there is a vertex $\{i^*\}$ with $\deg(\{i^*\}) < t - 1$, then $w^* < \lambda$.*

Lemma 3 *All edges (i, j) in subgraph $(B_w(G))^r = (V, (E_w)^r)$ have weight $w_{ij} \leq rw$.*

Proof: Recall, the edge $(i, j) \in (E_w)^r$ if and only if there is a path of r or fewer edges connecting $\{i\}$ to $\{j\}$ in $B_w(G)$. Since all edges in $B_w(G)$ have weight at most w , it follows immediately from (2.7) that all edges (i, j) in $(B_w(G))^r$ have weight $w_{ij} \leq rw$. ■

Lemma 4 *Suppose that $G^* = (V, E^*)$ is a subgraph of a complete graph $G = (V, E)$. A maximal independent set of G^* can be found in polynomial time.*

We will prove this lemma by presenting a polynomial-time procedure that obtains a maximal independent set. Though this lemma holds for all arbitrary graphs G —not just for those that are complete—our procedure exploits the completeness of G . This procedure will be used as part of our algorithm to find an approximately optimal partition.

Proof:

1. (Initialize) Initialize the maximal independent set $\mathbf{M} = \emptyset$. Initialize $\mathbf{I} = \{1, 2, \dots, n\}$.
2. (Find maximum edge weight) Find an edge $(i^*, j^*) \in E^*$ satisfying:

$$w_{i^*j^*} = \max\{w_{ij} : (i, j) \in E^*\}. \quad (2.10)$$

3. (Order vertices in G^*) Set $i_{(1)} = i^*$, and for $k = 2, \dots, n$, set $i_{(k)}$ to the index j^* that corresponds to the $(k - 1)^{\text{th}}$ largest value of $(w_{i^*j})_{j \neq i^*}$.
4. (Find first non-connected vertex). If $\mathbf{I} = \emptyset$, set $s = \infty$. Otherwise, set $s = \min(\mathbf{I})$.
5. (Add vertex to \mathbf{M}) If $s < \infty$: Set $\mathbf{M} = \mathbf{M} \cup \{i_{(s)}\}$; Remove s from \mathbf{I} ; Remove all s^* with $(i_{(s)}, i_{(s^*)}) \in E^*$ from \mathbf{I} ; Go to Step 4.
If $s = \infty$: Stop.

We now show, by proof by contradiction, that when the procedure terminates, \mathbf{M} is a maximal independent set.

Suppose that \mathbf{M} is not a maximal independent set. Thus, \mathbf{M} is either not independent or not maximal. or there is a vertex in $\{\{1\}, \{2\}, \dots, \{n\}\} \setminus \mathbf{M}$ that is not adjacent to any other vertex in \mathbf{M} . If \mathbf{M} is not independent, then there are vertices $\{i_{(s)}\}, \{i_{(s^*)}\} \in \mathbf{M}$, with $\{i_{(s)}\}$ added to \mathbf{M} before $\{i_{(s^*)}\}$, such that $(i_{(s)}, i_{(s^*)}) \in E^*$. However, once $\{i_{(s)}\}$ is added to \mathbf{M} in Step 5, $\{s^*\}$ is subsequently removed from \mathbf{I} ; $i_{(s^*)}$ cannot be added to \mathbf{M} . Thus, \mathbf{M} is independent. If \mathbf{M} is not maximal, then there is a vertex $\{i_{(s^*)}\} \in \{\{1\}, \{2\}, \dots, \{n\}\} \setminus \mathbf{M}$ that is not adjacent to any other vertex in \mathbf{M} . Step 5 of the procedure does not eliminate s^* from \mathbf{I} . The subsequent iteration of Step 4 finds $\min(\mathbf{I}) \leq s^* < \infty$, and so, the procedure will not terminate. However, since \mathbf{I} is finite, and at least one element of \mathbf{I} is removed at each iteration of Step 5, the procedure must terminate. Thus, \mathbf{M} is maximal.

Note that there are $O(n^2)$ edges in the graph G . Step 2 takes $O(n^2)$ time. Step 3 takes $O(n \log n)$ time. Each iteration of Steps 4 and 5 takes $O(n)$ time, and these steps will be performed at most $O(n)$ times. Thus, this entire procedure requires, at most, $O(n^2)$ time. ■

2.2.3 An algorithm for approximately optimal partitions

We now give our algorithm for finding a partition $\mathbf{v}^* \in \mathbf{V}^t$ satisfying (2.4).

1. (Sort edge weights) Sort edge weights in increasing order. Denote the k th largest edge weight by $w_{(k)}$.
2. (Initialize) Set $k = \lceil (t - 1)n/2 \rceil$ (a subgraph on n vertices with fewer than $(t - 1)n/2$ edges will have at least one vertex with degree less than $t - 1$).
3. (Obtain bottleneck graph) Obtain the bottleneck subgraph $B_{w_{(k)}}(G)$.

4. (Identify suboptimality of $w_{(k)}$) If at least one vertex has degree less than $t-1$, then $\lambda > w_{(k)}$ (by Corollary 2); set $k = k + 1$ and go to Step 3. Otherwise, advance to Step 5. Note that, since $\lambda = w_{(\ell)}$ for some $\ell \in \{1, \dots, n\}$, and since all vertices $\{i\} \in V$ have $\deg(\{i\}) \geq t-1$ in $B_\lambda(G)$ (by Lemma 1), it follows that $w_{(k)} \leq \lambda$ when the algorithm advances to Step 5.
5. (Obtain maximal independent set) Using the procedure given in the proof for Lemma 4, find a maximal independent set \mathbf{M} in the subgraph $(B_{w_{(k)}}(G))^2 = (V, (E_{w_{(k)}})^2)$. Note that, for any two vertices $\{i\}, \{j\} \in \mathbf{M}$, there is no path in $B_{w_{(k)}}(G)$ from $\{i\}$ to $\{j\}$ of two or fewer edges.
6. (Bloom from the maximal independent set) For each $\{i\} \in \mathbf{M}$, form a block of vertices V_i^* comprised of vertex $\{i\}$ and all vertices adjacent to $\{i\}$ in $B_{w_{(k)}}(G)$. Step 4 guarantees that $\deg(\{i\}) \geq t-1$ in this subgraph, so $|V_i^*| \geq t$. Since no path of two edges or less in $B_{w_{(k)}}(G)$ connects any two vertices in \mathbf{M} , it follows that $V_i^* \cap V_j^* = \emptyset$ for all $\{i\}, \{j\} \in \mathbf{M}$.
7. (Assign remaining vertices) Some vertices may not be assigned to a block yet. These vertices are at most a path of two edges away from a vertex $\{i\} \in \mathbf{M}$ (otherwise, it would contradict \mathbf{M} being a maximal independent set). For each unassigned vertex $\{j\}$, choose an $\{i\} \in \mathbf{M}$ such that $(i, j) \in (E_{w_{(k)}})^2$ and w_{ij} is as small as possible; assign $\{j\}$ to V_i^* .

After Step 7, the blocks $\mathbf{v}^* = (V_i^*)_{i \in \mathbf{I}}$, form a partition of \mathbf{V} ; Steps 6 and Step 7 ensure that blocks are disjoint, and Step 7 forces every vertex to be assigned a block. We now prove our main theorem:

Theorem 5 *The algorithm above finds an approximately optimal partition $\mathbf{v}^* \in \mathbf{V}^t$ in polynomial time.*

Proof: By Step 6, every block V_i^* in the partition \mathbf{v}^* has $|V_i^*| \geq t$. Thus, $\mathbf{v}^* \in \mathbf{V}^t$. By Step 7, for any vertex $\{i\} \in \mathbf{M}$, and for any two vertices $\{j\}, \{\ell\} \in V_i^*$, there is a path of at most two edges from $\{j\}$ to $\{i\}$ and from $\{i\}$ to $\{\ell\}$ in $B_{w_{(k)}}(G)$. Combining these paths, we form a path of four edges or less that connects $\{j\}$ to $\{\ell\}$. That is, any two vertices in the same block of \mathbf{v}^* are at most a path of four edges away from each other in $B_{w_{(k)}}(G)$. Thus, $G(\mathbf{v}^*) = (V, E(\mathbf{v}^*))$ is a subgraph of $(B_{w_{(k)}}(G))^4$. By Lemma 3, and since $w_{(k)} \leq \lambda$, all edges $(j, \ell) \in E(\mathbf{v}^*)$ have weight $w_{j\ell} \leq 4w_{(k)} \leq 4\lambda$. That is, for any two vertices $\{j\}, \{\ell\}$ in the same block $V_i^* \in \mathbf{v}^*$, the edge (j, ℓ) has weight $w_{j\ell} \leq 4\lambda$. Approximate optimality of \mathbf{v}^* follows immediately.

Step 1 of the algorithm can be completed in $O(n^2 \log n)$ time. Each iteration of Step 4 requires at most $O(n^2)$ operations; there will be at most $O(n^2)$ iterations of this step. From Lemma 4, Step 5 requires $O(n^2)$ time. Steps 6 and 7 also require $O(n^2)$ time. Thus, the entire algorithm is performed in $O(n^4)$ time. ■

Although the partition \mathbf{v}^* produced by our algorithm satisfies (2.4), additional processing may yield a partition with an even smaller maximal within-block edge weight. We now give a simple procedure to find such a partition from a given \mathbf{v}^* .

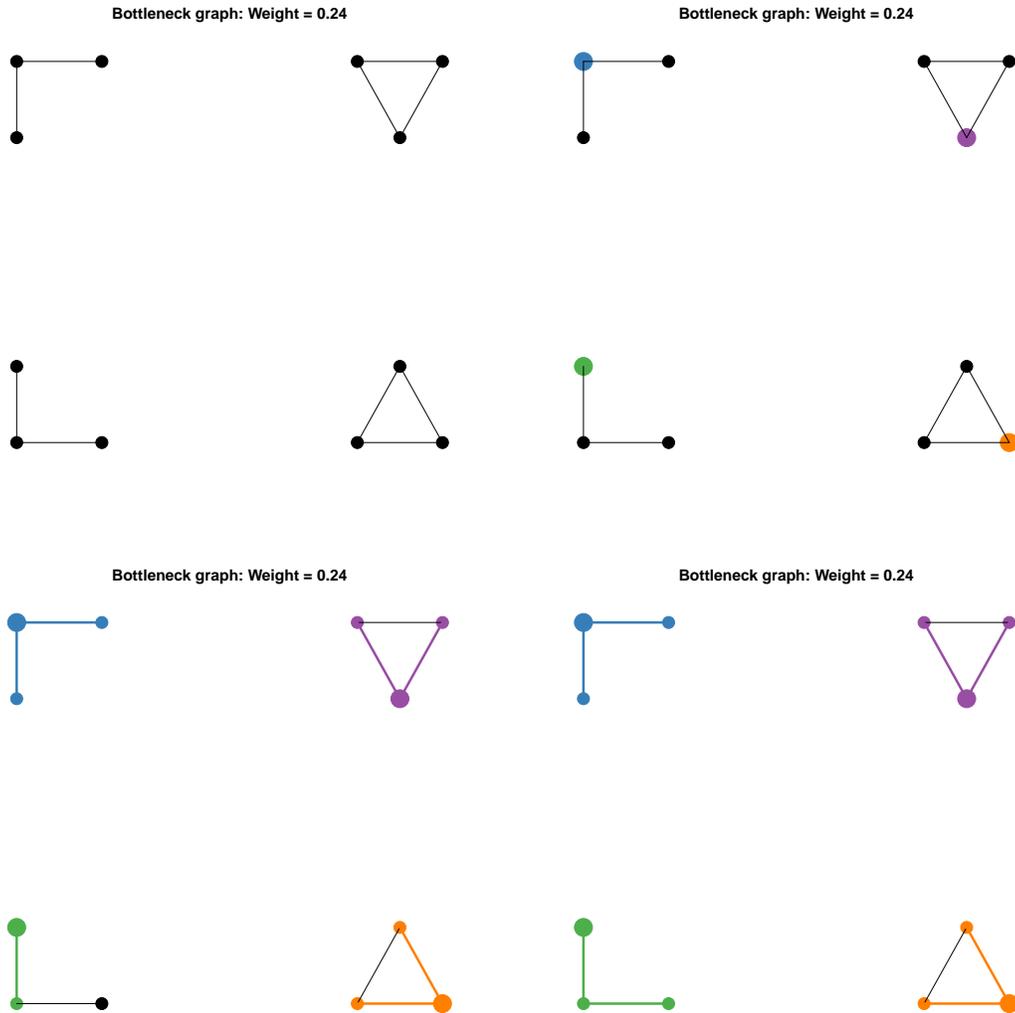


Figure 2.3: We demonstrate our approximate algorithm for the graph in figure 2.1. Each block of the partition obtained by the algorithm is denoted by a unique color. The upper-left graph is the bottleneck subgraph used in Step 5. The highlighted vertices in the upper-right graph are a maximal independent set of vertices for this subgraph. The lower-right graph demonstrates blooming from the maximal independent set in Step 6. The remaining unassigned vertex is assigned to the “green” block in the lower-right graph. The size threshold $t = 2$, and edge weights are Mahalanobis distances of heights and weights between units. Note that, for these twelve observations, the approximate algorithm yields the desired optimal blocking in figure 2.2. In general, this will not be the case.

2.2.4 Improvements to approximately optimal partitions

Given a partition \mathbf{v}^* obtained by the algorithm in 2.2.3, we now find an “improved” approximately optimal partition $\tilde{\mathbf{v}}^*$ by locating all blocks of size at least $2t$, and splitting them into smaller blocks, each with size less than $2t$.

Formally, for a block $V_i^* \in \mathbf{v}^*$, define

$$\left\lfloor \frac{|V_i^*|}{t} \right\rfloor \equiv m_i^*. \quad (2.11)$$

When $m_i^* = 1$, our procedure places the block V_i^* into $\tilde{\mathbf{v}}^*$. Otherwise, our procedure divides V_i^* into m_i^* smaller blocks, and places these smaller blocks into $\tilde{\mathbf{v}}^*$. The partition $\tilde{\mathbf{v}}^*$ is an improvement to \mathbf{v}^* in the sense that the maximum within-block edge weight of $\tilde{\mathbf{v}}^*$ will be as smaller or smaller than that of \mathbf{v}^* —our procedure ensures that $G(\tilde{\mathbf{v}}^*)$ is a subgraph of $G(\mathbf{v}^*)$.

1. (Initialize) Let k^* denote the number of blocks in \mathbf{v}^* . Initialize $i = 1$. Initialize $\tilde{\mathbf{v}}^* = \emptyset$.
2. (Obtain block from \mathbf{v}^*) Initialize $m = 1$. Set $\tilde{V}_i^* = V_i^*$.
3. (Find maximum edge weight for \tilde{V}_i^*) Find $\{j^*\}, \{\ell^*\}$ such that

$$w_{j^*\ell^*} = \max\{w_{j\ell} : \{j\}, \{\ell\} \in V_i^*\}. \quad (2.12)$$

4. (Create a block of size t) In $G(\mathbf{v}^*)$, find the $t - 1$ vertices $\{\ell\}$ that correspond to the $t - 1$ smallest values of $w_{j^*\ell}$. Denote these vertices by $\{\ell_{(1)}\}, \{\ell_{(2)}\}, \dots, \{\ell_{(t-1)}\}$. Create a block

$$\tilde{V}_{i,m}^* = \{j^*, \{\ell_{(1)}\}, \{\ell_{(2)}\}, \dots, \{\ell_{(t-1)}\}\}. \quad (2.13)$$

and remove all vertices in $\tilde{V}_{i,m}^*$ from \tilde{V}_i^* .

5. (Assign remaining vertices) If, after removing vertices from \tilde{V}_i^* , $0 < |\tilde{V}_i^*| < t$, place all remaining vertices in \tilde{V}_i^* in $\tilde{V}_{i,m}^*$.
6. (Increment m) If $m < m_i^*$, increment m and go to Step 3. Otherwise, set

$$\tilde{\mathbf{v}}^* = \tilde{\mathbf{v}}^* \bigcup_{j=1}^{m_i^*} \tilde{V}_{i,j}^*. \quad (2.14)$$

7. (Increment i) If $i < k^*$, increment i and go to Step 2. Otherwise, terminate.

2.3 Blocking estimators under Neyman-Rubin model

2.3.1 Notation and preliminaries

There are n units, numbered 1 through n . There are r treatments, numbered 1 through r . Each unit i has a vector of block covariates \mathbf{x}_i . A distance between block covariates (such as the Mahalanobis distance) between each pair of distinct units is measured.

Suppose the units are partitioned into b blocks (for example, by our algorithm), numbered 1 through b , with each block containing at least t^* units. For now, we assume $t^* \geq r$. Let n_c denote the number of units in block c . Assume that the units within each block c are ordered in some way: let (k, c) denote the k^{th} unit in block c . Let $\mathbf{n} \equiv (n_1, \dots, n_b)$.

Complete and block randomization

When we say that treatment is *completely randomized* we refer to the following randomization scheme: Each unit is assigned exactly one treatment.

1. When r divides n , each treatment is assigned to n/r units. Each of the

$$\prod_{i=0}^{r-1} \binom{n - i(n/r)}{n/r} \quad (2.15)$$

possible treatment assignments are equally likely.

2. Otherwise, let z denote the remainder of n/r . Of the r treatments, z treatments are replicated $\lfloor n/r \rfloor + 1$ times, and $r - z$ of the treatments are replicated only $\lfloor n/r \rfloor$ times. Each of the

$$\binom{r}{z} \prod_{i=0}^{z-1} \binom{n - i(\lfloor n/r \rfloor + 1)}{\lfloor n/r \rfloor + 1} \prod_{i=0}^{r-z-1} \binom{n - z(\lfloor n/r \rfloor + 1) - i\lfloor n/r \rfloor}{\lfloor n/r \rfloor} \quad (2.16)$$

possible treatment assignments are equally likely.

Treatment is *block randomized* if treatment is completely randomized within each block and treatment is assigned independently across blocks. Given a block c , let z_c denote the remainder of n_c/r .

The Neyman-Rubin Causal Model

Let T_{kcs} denote treatment indicators for each unit (k, c) :

$$T_{kcs} = \begin{cases} 1, & \text{unit } (k, c) \text{ receives treatment } s, \\ 0, & \text{otherwise.} \end{cases} \quad (2.17)$$

Let $\#T_{cs} = \sum_{k=1}^{n_c} T_{kcs}$ denote the number of units in block c that receive treatment s , and let $\#T_s = \sum_{c=1}^b \#T_{cs}$ denote the number of units in total assigned to s . Under block randomization, $\#T_{cs}$ has distribution

$$\#T_{cs} = \begin{cases} \lfloor n_c/r \rfloor + 1 & \text{with probability } z_c/r, \\ \lfloor n_c/r \rfloor & \text{with probability } (r - z_c)/r. \end{cases} \quad (2.18)$$

Note that, if $t^* \geq r$, then $\#T_{cs} \geq 1$. Under complete randomization, $\#T_s$ has distribution

$$\#T_s = \begin{cases} \lfloor n/r \rfloor + 1 & \text{with probability } z/r, \\ \lfloor n/r \rfloor & \text{with probability } (r - z)/r. \end{cases} \quad (2.19)$$

For the following calculations, we assume responses follow the Neyman-Rubin Causal Model (NRCM) (Splawa-Neyman et al. 1990; Rubin 1974; Holland 1986). Let y_{kcs} denote the *potential outcome* of unit (k, c) given treatment s —the hypothetical observed value of unit (k, c) had that unit received treatment s . Under the NRCM, the potential outcome y_{kcs} is non-random, and the value of this outcome is observed if and only if (k, c) receives treatment s ; exactly one of $\{y_{kcs}\}_{s=1}^r$ is observed. The observed response is:

$$Y_{kc} \equiv y_{kc1}T_{kc1} + y_{kc2}T_{kc2} + \cdots + y_{kcr}T_{kcr}. \quad (2.20)$$

Inherent in this equation is the *stable-unit treatment value assumption* (SUTVA): the observed Y_{kc} only depends on which treatment is assigned to unit (k, c) , and is not affected by the treatment assignment of any other unit (k', c') .

2.3.2 Parameters of interest and estimators

Given any two treatments s and t , we wish to estimate the *sample average treatment effect of s relative to t* (SATE_{st}), denoted μ_{st} . The SATE_{st} is a sum of differences of potential outcomes:

$$\delta_{st} = \frac{1}{n} \sum_{c=1}^b \sum_{k=1}^{n_c} (y_{kcs} - y_{kct}). \quad (2.21)$$

For this paper, we consider two estimators of the μ_{st} : the difference-in-means estimator:

$$\hat{\delta}_{st,\text{diff}} = \sum_{c=1}^b \frac{n_c}{n} \sum_{k=1}^{n_c} \left(\frac{y_{kcs}T_{kcs}}{\#T_{cs}} - \frac{y_{kct}T_{kct}}{\#T_{ct}} \right), \quad (2.22)$$

and the Horvitz-Thompson estimator:

$$\hat{\delta}_{st,\text{HT}} = \sum_{c=1}^b \frac{n_c}{n} \sum_{k=1}^{n_c} \left(\frac{y_{kcs}T_{kcs}}{n_c/r} - \frac{y_{kct}T_{kct}}{n_c/r} \right). \quad (2.23)$$

Before discussing the properties of these estimators, it will be helpful to break μ_{st} into block-level components. For any block c and distinct treatments s and t , define the following block-level means:

$$\mu_{cs} \equiv \frac{1}{n_c} \sum_{k=1}^{n_c} y_{kcs}, \quad (2.24)$$

$$\delta_{cst} \equiv \frac{1}{n_c} \sum_{k=1}^{n_c} y_{kcs} - y_{kct}. \quad (2.25)$$

Note that

$$\delta_{cst} = \mu_{cs} - \mu_{ct}, \quad (2.26)$$

$$\delta_{st} = \sum_{c=1}^b \frac{n_c}{n} \delta_{cst}. \quad (2.27)$$

The block-level variance of the potential outcomes for treatment s is:

$$\sigma_{cs}^2 = \sum_{k=1}^{n_c} \frac{(y_{kcs} - \mu_{cs})^2}{n_c} \quad (2.28)$$

$$= \sum_{k=1}^{n_c} \frac{y_{kcs}^2}{n_c} - \left(\sum_{k=1}^{n_c} \frac{y_{kcs}}{n_c} \right)^2. \quad (2.29)$$

The block-level covariance between the potential outcomes for treatments s and t is:

$$\gamma_{cst} = \sum_{k=1}^{n_c} \frac{(y_{kcs} - \mu_{cs})(y_{kct} - \mu_{ct})}{n_c} \quad (2.30)$$

$$= \sum_{k=1}^{n_c} \frac{y_{kcs} y_{kct}}{n_c} - \sum_{k=1}^{n_c} \frac{y_{kcs}}{n_c} \sum_{k=1}^{n_c} \frac{y_{kct}}{n_c}. \quad (2.31)$$

Two block-level estimators for μ_{cs} are:

$$\hat{\mu}_{cs,\text{diff}} \equiv \sum_{k=1}^{n_c} \frac{y_{kcs} T_{kcs}}{\#T_{cs}}, \quad (2.32)$$

$$\hat{\mu}_{cs,\text{HT}} \equiv \sum_{k=1}^{n_c} \frac{y_{kcs} T_{kcs}}{n_c/r}. \quad (2.33)$$

These estimators satisfy:

$$\hat{\delta}_{st,\text{diff}} = \sum_{c=1}^b \frac{n_c}{n} (\hat{\mu}_{cs,\text{diff}} - \hat{\mu}_{ct,\text{diff}}), \quad (2.34)$$

$$\hat{\delta}_{st,\text{HT}} = \sum_{c=1}^b \frac{n_c}{n} (\hat{\mu}_{cs,\text{HT}} - \hat{\mu}_{ct,\text{HT}}). \quad (2.35)$$

By linearity of expectations:

$$\mathbb{E}(\hat{\delta}_{st,\text{diff}}) = \sum_{c=1}^b \frac{n_c}{n} (\mathbb{E}(\hat{\mu}_{cs,\text{diff}}) - \mathbb{E}(\hat{\mu}_{ct,\text{diff}})), \quad (2.36)$$

$$\mathbb{E}(\hat{\delta}_{st,\text{HT}}) = \sum_{c=1}^b \frac{n_c}{n} (\mathbb{E}(\hat{\mu}_{cs,\text{HT}}) - \mathbb{E}(\hat{\mu}_{ct,\text{HT}})), \quad (2.37)$$

and by independence of treatment assignment across blocks under block randomization:

$$\text{Var}(\hat{\delta}_{st,\text{diff}}) = \sum_{c=1}^b \frac{n_c^2}{n^2} [\text{Var}(\hat{\mu}_{cs,\text{diff}}) + \text{Var}(\hat{\mu}_{ct,\text{diff}}) - 2\text{cov}(\hat{\mu}_{cs,\text{diff}}, \hat{\mu}_{ct,\text{diff}})], \quad (2.38)$$

$$\text{Var}(\hat{\delta}_{st,\text{HT}}) = \sum_{c=1}^b \frac{n_c^2}{n^2} [\text{Var}(\hat{\mu}_{cs,\text{HT}}) + \text{Var}(\hat{\mu}_{ct,\text{HT}}) - 2\text{cov}(\hat{\mu}_{cs,\text{HT}}, \hat{\mu}_{ct,\text{HT}})]. \quad (2.39)$$

Thus, when deriving expectations and variances of the difference-in-means and Horvitz-Thompson estimators, it suffices to derive expectations, variances, and covariances of their corresponding block-level estimators. We now make these derivations.

Lemma 6 *Under block randomization, for $s \neq t$,*

$$\mathbb{E}(\hat{\mu}_{cs,\text{diff}}) = \mu_{cs}, \quad (2.40)$$

$$\text{Var}(\hat{\mu}_{cs,\text{diff}}) = \frac{(r-1)n_c(n_c+r-2z_c) + z_c(r-z_c)}{(n_c-1)(n_c-z_c)(n_c+r-z_c)} \sigma_{cs}^2, \quad (2.41)$$

$$\text{cov}(\hat{\mu}_{cs,\text{diff}}, \hat{\mu}_{ct,\text{diff}}) = \frac{-\gamma_{cst}}{n_c-1}. \quad (2.42)$$

Lemma 7 *Under block randomization, for $s \neq t$,*

$$\mathbb{E}(\hat{\mu}_{cs,\text{HT}}) = \mu_{cs}, \quad (2.43)$$

$$\text{Var}(\hat{\mu}_{cs,\text{HT}}) = \frac{n_c^2(r-1) - z_c(r-z_c)}{n_c^2(n_c-1)} \sigma_{cs}^2 + \frac{z_c(r-z_c)}{n_c^2} \left(\sum_{k=1}^{n_c} \frac{y_{kcs}}{n} \right)^2, \quad (2.44)$$

$$\text{cov}(\hat{\mu}_{cs,\text{HT}}, \hat{\mu}_{ct,\text{HT}}) = \frac{-\gamma_{cst}}{n_c-1} - \frac{z_c(r-z_c)}{(r-1)n_c^3(n_c-1)} \sum_{k=1}^{n_c} \sum_{\ell \neq k} y_{kcs} y_{\ell ct}. \quad (2.45)$$

These lemmas are proven in Appendix A.1.

From Lemmas 6 and 7, and using (2.36) and (2.39), we can show that both the difference-in-means estimate and the Horvitz-Thompson estimate of the SATE_{st} are unbiased, and we can compute the variance of these estimates.

Theorem 8 *Under block randomization, when $s \neq t$:*

$$\mathbb{E}(\hat{\delta}_{st,diff}) = \delta_{st}, \quad (2.46)$$

$$\begin{aligned} \text{Var}(\hat{\delta}_{st,diff}) &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left(\frac{(r-1)n_c(n_c+r-2z_c)}{(n_c-1)(n_c-z_c)(n_c+r-z_c)} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n-1} \right) \\ &\quad + \sum_{c=1}^b \frac{n_c^2}{n^2} \left(\frac{z_c(r-z_c)}{(n_c-1)(n_c-z_c)(n_c+r-z_c)} (\sigma_{cs}^2 + \sigma_{ct}^2) \right). \end{aligned} \quad (2.47)$$

Theorem 9 *Under block randomization, when $s \neq t$:*

$$\mathbb{E}(\hat{\delta}_{st,HT}) = \delta_{st}, \quad (2.48)$$

$$\begin{aligned} \text{Var}(\hat{\delta}_{st,HT}) &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left(\frac{r-1}{n_c-1} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n_c-1} \right) \\ &\quad - \sum_{c=1}^b \frac{n_c^2}{n^2} \left[\frac{z_c(r-z_c)}{n_c^2} \left(\frac{\sigma_{cs}^2 + \sigma_{ct}^2}{(n_c-1)} - \left(\sum_{k=1}^{n_c} \frac{y_{kcs}}{n_c} \right)^2 - \left(\sum_{k=1}^{n_c} \frac{y_{kct}}{n_c} \right)^2 \right) \right] \\ &\quad + \sum_{c=1}^b \frac{n_c^2}{n^2} \left(\frac{2z_c(r-z_c)}{(r-1)n_c^3(n_c-1)} \sum_{k=1}^{n_c} \sum_{\ell \neq k} y_{kcs} y_{\ell ct} \right). \end{aligned} \quad (2.49)$$

Note that, when r divides each n_c , then $\hat{\mu}_{st,diff} = \hat{\mu}_{st,HT}$ and

$$\begin{aligned} \text{Var}(\hat{\delta}_{st,diff}) &= \text{Var}(\hat{\delta}_{st,HT}) \\ &= \sum_{c=1}^b \frac{n_c^2}{n^2} \left(\frac{r-1}{n_c-1} (\sigma_{cs}^2 + \sigma_{ct}^2) + 2 \frac{\gamma_{cst}}{n_c-1} \right). \end{aligned} \quad (2.50)$$

2.3.3 Estimating the variance

Variance estimation of both the difference-in-means estimate and the Horvitz-Thompson estimate is complicated by the existence of the γ_{cst} term. This term is not directly unbiasedly estimable; such an estimate requires knowledge of potential outcomes under both treatment s and treatment t within a single unit. Only one potential outcome is observed for each unit.

We first discuss estimating the block-level variances $\text{Var}(\hat{\mu}_{cs,diff})$ and $\text{Var}(\hat{\mu}_{cs,HT})$. We then describe how to use these block-level estimates to obtain conservative estimates for $\text{Var}(\hat{\mu}_{st,diff})$ and $\text{Var}(\hat{\mu}_{st,HT})$.

Block-level variance estimates

Let

$$\hat{\sigma}_{cs,diff}^2 = \sum_{i=1}^n \frac{T_{is} \left(y_{is} - \sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2}{\#T_s - 1}. \quad (2.51)$$

Consider the following two variance estimates:

$$\widehat{\text{Var}}(\hat{\mu}_{cs,diff}) \equiv \frac{(r-1)n_c(n_c+r-2z_c) + z_c(r-z_c)}{n_c(n_c-z_c)(n_c+r-z_c)} \hat{\sigma}_{cs,diff}^2, \quad (2.52)$$

$$\begin{aligned} \widehat{\text{Var}}(\hat{\mu}_{cs,HT}) &\equiv \frac{r(r-1)}{n^2} \sum_{i=1}^n y_{is}^2 T_{is} \\ &\quad - \left(\frac{nr^2(r-1) - r^2z(r-z)}{n^4 - n^3r + n^2z(r-z)} \right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js}. \end{aligned} \quad (2.53)$$

These estimates are unbiased for their respective block-level variances.

Lemma 10 *Under block randomization:*

$$\mathbb{E} \left[\widehat{\text{Var}}(\hat{\mu}_{cs,diff}) \right] = \text{Var}(\hat{\mu}_{cs,diff}), \quad (2.54)$$

$$\mathbb{E} \left[\widehat{\text{Var}}(\hat{\mu}_{cs,HT}) \right] = \text{Var}(\hat{\mu}_{cs,HT}). \quad (2.55)$$

This lemma is proven in Appendix A.2.

Variance estimates of SATE_{st} estimates

We use the fact that

$$- \sqrt{\text{Var}(X)\text{Var}(Y)} \leq \text{cov}(X, Y) \leq \sqrt{\text{Var}(X)\text{Var}(Y)} \quad (2.56)$$

to obtain conservative variance estimates of $\hat{\mu}_{st,diff}$ and $\hat{\mu}_{st,HT}$. These estimates are conservative in the sense that we find an upper-bound for the variance of these estimates that is a function of block-level variances, and we plug in our estimates of the block-level variances from Section 2.3.3.

From (2.38), (2.39), (2.56), and the arithmetic mean-geometric mean inequality, it follows that:

$$\begin{aligned}
\text{Var}(\hat{\delta}_{st,\text{diff}}) &\leq \sum_{c=1}^b \frac{n_c^2}{n^2} \left[\text{Var}(\hat{\mu}_{cs,\text{diff}}) + \text{Var}(\hat{\mu}_{ct,\text{diff}}) + 2\sqrt{\text{Var}(\hat{\mu}_{cs,\text{diff}})\text{Var}(\hat{\mu}_{ct,\text{diff}})} \right] \\
&\leq \sum_{c=1}^b \frac{n_c^2}{n^2} \left[\text{Var}(\hat{\mu}_{cs,\text{diff}}) + \text{Var}(\hat{\mu}_{ct,\text{diff}}) + 2\frac{\text{Var}(\hat{\mu}_{cs,\text{diff}}) + \text{Var}(\hat{\mu}_{ct,\text{diff}})}{2} \right] \\
&\leq \sum_{c=1}^b \frac{2n_c^2}{n^2} [\text{Var}(\hat{\mu}_{cs,\text{diff}}) + \text{Var}(\hat{\mu}_{ct,\text{diff}})], \tag{2.57}
\end{aligned}$$

$$\begin{aligned}
\text{Var}(\hat{\delta}_{st,\text{HT}}) &\leq \sum_{c=1}^b \frac{n_c^2}{n^2} \left[\text{Var}(\hat{\mu}_{cs,\text{HT}}) + \text{Var}(\hat{\mu}_{ct,\text{HT}}) + 2\sqrt{\text{Var}(\hat{\mu}_{cs,\text{HT}})\text{Var}(\hat{\mu}_{ct,\text{HT}})} \right] \\
&\leq \sum_{c=1}^b \frac{2n_c^2}{n^2} [\text{Var}(\hat{\mu}_{cs,\text{HT}}) + \text{Var}(\hat{\mu}_{ct,\text{HT}})]. \tag{2.58}
\end{aligned}$$

These variance bounds can be estimated unbiasedly:

$$\widehat{\text{Var}}(\hat{\delta}_{st,\text{diff}}) \equiv \sum_{c=1}^b \frac{2n_c^2}{n^2} \left[\widehat{\text{Var}}(\hat{\mu}_{cs,\text{diff}}) + \widehat{\text{Var}}(\hat{\mu}_{ct,\text{diff}}) \right], \tag{2.59}$$

$$\widehat{\text{Var}}(\hat{\delta}_{st,\text{HT}}) \equiv \sum_{c=1}^b \frac{2n_c^2}{n^2} \left[\widehat{\text{Var}}(\hat{\mu}_{cs,\text{HT}}) + \widehat{\text{Var}}(\hat{\mu}_{ct,\text{HT}}) \right]. \tag{2.60}$$

On a final note, when blocks contain several replications of each treatment, and when potential outcomes satisfy some smoothing conditions with respect to the block covariates, a good estimate of the covariance γ_{cst} may be obtained. For details, see Abadie & Imbens (2008); Imbens (2011).

2.3.4 Comparing block randomization and complete randomization

We now describe precisely the conditions under which the variance of SATE_{st} estimates under block-randomized treatment assignment is smaller than those under complete randomization (without blocking). For the ease of demonstration, we assume in this subsection that r divides each n_c . After identifying these precise conditions, we then show that these conditions are met (in expectation) when the assignment of units into blocks of fixed size is completely randomized. Thus, unless block covariates are worse than random chance at predicting potential outcomes, blocking will only improve precision of SATE_{st} estimates. These results are a generalization of those found in Imai (2008).

Define the completely randomized estimator for the SATE_{st} as:

$$\hat{\delta}_{st,cs} \equiv \sum_{c=1}^b \sum_{k=1}^{n_c} \frac{y_{kcs}T_{kcs} - y_{kct}T_{kct}}{n/r}. \tag{2.61}$$

Define the following domain-level parameters:

$$\mu_s \equiv \frac{1}{n} \sum_{c=1}^b \sum_{k=1}^{n_c} y_{kcs}, \quad (2.62)$$

$$\sigma_s^2 = \sum_{c=1}^b \sum_{k=1}^{n_c} \frac{(y_{kcs} - \mu_s)^2}{n}, \quad (2.63)$$

$$\gamma_{st} = \sum_{c=1}^b \sum_{k=1}^{n_c} \frac{(y_{kcs} - \mu_s)(y_{kct} - \mu_t)}{n}. \quad (2.64)$$

Following the approach in Appendix A.1, we find that

$$\mathbb{E}(\hat{\delta}_{st,cr}) = \mu_{st}, \quad (2.65)$$

$$\begin{aligned} \text{Var}(\hat{\delta}_{st,cr}) &= \frac{r-1}{n-1}(\sigma_s^2 + \sigma_t^2) + 2\frac{\gamma_{st}}{n-1} \\ &= \sum_{c=1}^b \frac{n_c^2}{\sum_c n_c^2} \left(\frac{r-1}{n-1}(\sigma_s^2 + \sigma_t^2) + 2\frac{\gamma_{st}}{n-1} \right). \end{aligned} \quad (2.66)$$

Estimators under both block randomization and complete randomization are unbiased for the SATE_{st} . The variance under block randomization will be as small or smaller than that under complete randomization precisely when

$$\sum_{c=1}^b \frac{n_c^2}{\sum_c n_c^2} \left(\frac{r-1}{n-1}(\sigma_s^2 + \sigma_t^2) + 2\frac{\gamma_{st}}{n-1} \right) - \frac{n_c^2}{n^2} \left(\frac{r-1}{n_c-1}(\sigma_{cs}^2 + \sigma_{ct}^2) + 2\frac{\gamma_{cst}}{n_c-1} \right) \geq 0 \quad (2.67)$$

We can write this condition in terms of a comparison between block-level and sample-level variances. Let

$$\sigma_{c(s+t)}^2 = \frac{1}{n_c} \sum_{k=1}^{n_c} (y_{kcs} + y_{kct} - \mu_{cs} - \mu_{ct})^2 \quad (2.68)$$

$$\sigma_{s+t}^2 = \frac{1}{n} \sum_{c=1}^b \sum_{k=1}^{n_c} (y_{kcs} + y_{kct} - \mu_s - \mu_t)^2 \quad (2.69)$$

It follows that the variance under block randomization will be as small or smaller than that under complete randomization if and only if

$$\delta_{cr,blk} \equiv \sum_{c=1}^b n_c^2 \left(\frac{(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2}{(n-1)\sum_c n_c^2} - \frac{(r-2)(\sigma_{cs}^2 + \sigma_{ct}^2) + \sigma_{c(s+t)}^2}{(n_c-1)n^2} \right) \geq 0 \quad (2.70)$$

This formula gives some insight as to what properties of a blocking are helpful in reducing variance. Terms of $\delta_{cr,bl}$ will be positive (and thus, will favor estimates under block randomization) if and only if

$$\frac{(r-2)(\sigma_{cs}^2 + \sigma_{ct}^2) + \sigma_{c(s+t)}^2}{(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2} \leq \frac{(n-1) \sum_c n_c^2}{(n_c-1)n^2} \quad (2.71)$$

Since this fraction gets smaller as n_c gets larger, it follows that blocking helps most when the block-level variance in the largest-sized blocks is small.

We now show that, when units are randomly assigned to blocks, the variances of estimates of the $SATE_{st}$ under block randomization will not be greater in expectation than those under complete randomization. We say that an assignment of n units into blocks of sizes $\mathbf{n} = (n_1, \dots, n_b)$ is a *completely randomized blocking with block sizes \mathbf{n}* if each possible blocking with those block sizes is equally likely. Under completely randomized blocking, the block-level variances σ_{cs}^2 , σ_{ct}^2 , and $\sigma_{c(s+t)}^2$ are random variables; sample-level variances σ_s^2 , σ_t^2 , and σ_{s+t}^2 and block sizes are constants.

We now give the following result:

Theorem 11 *Under completely randomized blocking,*

$$\mathbb{E}(\delta_{cr,blk}) = 0. \quad (2.72)$$

That is, even when units are assigned to blocks randomly, the variance of estimates of the $SATE_{st}$ under block randomization will not be larger than that under complete randomization. When block covariates predict potential outcomes better than at random, blocking will increase the precision of $SATE_{st}$ estimates. A proof of this theorem is given in Appendix A.3.

Chapter 3

Sharper p -Values for Stratified Election Audits

3.1 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 mis-configuration, 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 & 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 practice¹ 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 substantially reduce the counting burden of a risk-limiting audit when the machine outcome is correct.

¹See <http://electionaudits.org/principles.html> (last visited 23 September 2011).

3.2 Audits using stratified simple random samples

3.2.1 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 ,

$$\mathbf{1}(s) \equiv \begin{cases} 1, & \text{if } s \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

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

$$a \cdot b \equiv \sum_{j=1}^N a_j b_j. \quad (3.2)$$

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 k th 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 $\ell \in \mathbf{I}_L$,

$$A_{w\ell} > 0, \quad (3.3)$$

or equivalently, if

$$V_{w\ell} - A_{w\ell} = \sum_{c=1}^C \sum_{k=1}^{N_c} [v_{kcw} - v_{kcl} - (a_{kcw} - a_{kcl})] < V_{w\ell}. \quad (3.4)$$

The apparent outcome is wrong if and only if [3.4] fails for some $w \in \mathbf{I}_W$ and $\ell \in \mathbf{I}_L$.

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

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

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

$$\text{If the apparent election outcome is wrong, then } \sum_{c=1}^C \sum_{k=1}^{N_c} e_{kc}^H \geq \mu. \quad (3.5)$$

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

$$e_{kc}^H \equiv \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kcl} - (a_{kcw} - a_{kcl})}{V_{w\ell}}. \quad (3.6)$$

For the MRO, [3.5] holds with $\mu = 1$.

Testing statistically whether

$$\sum_{c=1}^C \sum_{k=1}^{N_c} e_{kc}^H \geq \mu \quad (3.7)$$

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

$$e_{kc}^H \leq \max_{w \in \mathbf{I}_w, \ell \in \mathbf{I}_\ell} \frac{v_{kcw} - v_{kcl} + b_{kc}}{V_{w\ell}} \equiv \omega_{kc}. \quad (3.8)$$

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

$$\omega_{kc} \geq \omega_{k'c} \quad \text{if } k < k'. \quad (3.9)$$

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

$$e = (e_{kc})_{k=1}^{N_c} \Big|_{c=1}^C \in \mathbb{R}^N \text{ such that } e_{kc} \leq \omega_{kc}, \quad k = 1, \dots, N_c, \quad c = 1, \dots, C. \quad (3.10)$$

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

$$\mathbf{E}_\mu \equiv \left\{ e \in \mathbf{E} : \sum_{c=1}^C \sum_{k=1}^{N_c} e_{kc} \geq \mu \right\}. \quad (3.11)$$

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

3.2.2 Computing the p -value

This section sets out the precise problem we solve: finding a sharper (but still conservative) p -value for the null hypothesis² that the apparent outcome is incorrect from a stratified random sample. Let $\mathbf{J}_c^{n_c}$ be a simple random sample of n_c elements from $\{(1, c), \dots, (N_c, c)\}$, 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

$$T \equiv \max_{(k,c) \in \mathbf{J}_{\vec{n}}} e_{kc}^H \quad (3.12)$$

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 $\{w_{kc}\}_{k=1}^{N_c} \Big|_{c=1}^C$ be a set of N monotone increasing functions. We could base the audit on the test statistic

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

²<http://xkcd.com/892/> (last visited 23 September 2011).

where e_{kc}^H is not necessarily the MRO. For instance, in Section 3.5, 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 B.3 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_{kc}^H 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}_μ that maximizes the probability that $T \leq t$. Let $e \in \mathbf{E}$ be an allocation of differences. Define

$$P_{\mathbf{J}_{\vec{n}}}(e) \equiv P_{\mathbf{J}_{\vec{n}}}(e; t) \equiv P \left(\max_{(k,c) \in \mathbf{J}_{\vec{n}}} e_{kc} \leq t \right). \quad (3.13)$$

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, $\Pr_e\{T \leq t\}$.

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

$$P_{\#} = P_{\#}(t; \vec{n}) \equiv \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\vec{n}}}(e; t). \quad (3.14)$$

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

$$P_+ \geq P_{\#} \quad (3.15)$$

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

$$\mathbf{G}(e) = \mathbf{G}(e; t) \equiv \{(k, c) : e_{kc} > t\} \quad (3.16)$$

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 the set of batches $\{(1, c), \dots, (N_c, c)\}$ 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

$$P_{\mathbf{J}_{\vec{n}}}(e) = \begin{cases} \prod_{c=1}^C \frac{\binom{N_c - \#_c \mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}}, & N_c - \#_c \mathbf{G}(e) \geq n_c, \quad c = 1, \dots, C, \\ 0, & \text{otherwise.} \end{cases} \quad (3.17)$$

The exact p -value $P_{\#}$ [3.14] 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 [3.14] has special structure that allows us to find the exact p -value quickly.

3.3 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 & Toth 1998).

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

$$P_{\#} \equiv \max_{e \in \mathbf{E}_{\mu}} P_{\mathbf{J}_{\vec{n}}}(e) = \max_{e \in \tilde{\mathbf{E}}_{\mu}} P_{\mathbf{J}_{\vec{n}}}(e). \quad (3.18)$$

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.

3.3.1 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 and 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

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

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

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

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

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

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

Note that

$$\tilde{e} \in \mathbf{E} \text{ and } \tilde{\tilde{e}} = \tilde{e}. \quad (3.19)$$

By definition of κ_c ,

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

Hence,

$$\text{if } e \in \mathbf{E}_\mu \text{ then } \tilde{e} \in \mathbf{E}_\mu. \quad (3.20)$$

By [3.9],

$$\text{if } k < k', \quad [\omega_{kc} - (\omega_{kc} \wedge t)] \geq [\omega_{k'c} - (\omega_{k'c} \wedge t)]. \quad (3.21)$$

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

$$\begin{aligned} & \sum_{k=1}^{\#_c \mathbf{G}(e)} \omega_{kc} + \sum_{\#_c \mathbf{G}(e)+1}^{N_c} (\omega_{kc} \wedge t) \\ &= \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)] \mathbf{1}(k \leq \#_c \mathbf{G}(e)) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\ &\geq \sum_{k=1}^{N_c} [\omega_{kc} - (\omega_{kc} \wedge t)] \mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} (\omega_{kc} \wedge t) \\ &\geq \sum_{k=1}^{N_c} [e_{kc} - t] \mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} t \mathbf{1}(e_{kc} > t) + \sum_{k=1}^{N_c} e_{kc} \mathbf{1}(e_{kc} \leq t) \\ &= \sum_{k=1}^{N_c} e_{kc}. \end{aligned} \quad (3.22)$$

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

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

It follows from [3.17] that

$$P_{\mathbf{J}_{\bar{n}}}(\tilde{e}) \geq P_{\mathbf{J}_{\bar{n}}}(e). \quad (3.23)$$

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 [3.19], [3.20], and [3.23] hold for all $e \in \mathbf{E}$, it follows that

$$\max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\bar{n}}}(\tilde{e}) = \max_{e \in \mathbf{E}_\mu} P_{\mathbf{J}_{\bar{n}}}(e). \quad (3.24)$$

Thus, if we define

$$\tilde{\mathbf{E}} \equiv \{\tilde{e} : e \in \mathbf{E}\}, \quad (3.25)$$

and let

$$\tilde{\mathbf{E}}_\mu \equiv \tilde{\mathbf{E}} \cap \mathbf{E}_\mu, \quad (3.26)$$

then [3.18] 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}_μ . Maximizing $P_{\mathbf{J}_{\bar{n}}}$ over allocations in this smaller set can be reduced to KP, as we now show.

3.3.2 Maximizing $P_{\mathbf{J}_{\bar{n}}}$ 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 3.3. 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 3.3, 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

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

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 $P_{\mathbf{J}_{\bar{n}}}$. 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

$$\log P_\# = \log \max_{e \in \tilde{\mathbf{E}}_\mu} \{P_{\mathbf{J}_{\bar{n}}}(e)\} = \min_{e \in \tilde{\mathbf{E}}_\mu} \{-\log(P_{\mathbf{J}_{\bar{n}}}(e))\} = \min_{y \in \tilde{\mathbf{X}}} \{q \cdot y : u \cdot y \geq M\}. \quad (3.27)$$

We then show that

$$\lambda = \min_{x \in \mathbf{X}} \{q \cdot x : u \cdot x \geq M\} = \min_{y \in \tilde{\mathbf{X}}} \{q \cdot y : u \cdot y \geq M\}. \quad (3.28)$$

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

$$j = j(k, c) \equiv k + \sum_{c' < c} N_{c'}, \quad k = 1, \dots, N_c, \quad c = 1, \dots, C. \quad (3.29)$$

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

$$c = c(j) \equiv \min \left\{ d : \sum_{i=1}^d N_d \geq j \right\}, \quad k = k(j) \equiv j - \sum_{d=1}^{c(j)-1} N_d, \quad (3.30)$$

Recall that $\mathbf{G}(e)$ is the set of batches (k, c) for which $e_{kc} > t$ [3.16]. For $e \in \mathbf{E}$, define

$$g_{kc}(e) \equiv \mathbf{1}((k, c) \in \mathbf{G}(e)), \quad (3.31)$$

$$g(e) \equiv (g_{kc}(e))_{k=1}^{N_c} {}_c \in \mathbf{X}, \quad (3.32)$$

and

$$\tilde{\mathbf{X}} \equiv \left\{ y \in \mathbf{X} : y = g(e) \text{ for some } e \in \tilde{\mathbf{E}} \right\}. \quad (3.33)$$

Constraint: Let

$$u_{kc} \equiv \omega_{kc} - (\omega_{kc} \wedge t). \quad (3.34)$$

Note that

$$u_{kc} = 0 \quad \text{if and only if} \quad \omega_{kc} \leq t. \quad (3.35)$$

By [3.21],

$$u_{kc} \geq u_{k'c} \quad \text{if} \quad k < k'. \quad (3.36)$$

Let

$$M \equiv \left[\mu - \sum_{c=1}^C \sum_{k=1}^{N_c} \omega_{kc} \wedge t \right] \vee 0. \quad (3.37)$$

Observe that if $M = 0$, then

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

and

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

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

$$u \cdot g(e) \geq M. \quad (3.38)$$

Thus,

$$\{g(e) : e \in \tilde{\mathbf{E}}_\mu\} = \{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}. \quad (3.39)$$

We assume $\{y \in \tilde{\mathbf{X}} : u \cdot y \geq M\}$ is non-empty; otherwise, [3.39] shows that the apparent outcome must be correct, so the p -value is 0.

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

$$\begin{aligned} P_{\mathbf{J}_{\bar{n}}}(e) &= \prod_{c=1}^C \frac{\binom{N_c - \#_c \mathbf{G}(e)}{n_c}}{\binom{N_c}{n_c}} = \prod_{c=1}^C \prod_{k=1}^{\#_c \mathbf{G}(e)} \frac{\binom{N_c - k}{n_c}}{\binom{N_c - k + 1}{n_c}} \\ &= \prod_{c=1}^C \prod_{k=1}^{\#_c \mathbf{G}(e)} \frac{N_c - n_c - k + 1}{N_c - k + 1}. \end{aligned} \quad (3.40)$$

If instead there exists c such that $N_c - \#_c \mathbf{G}(e) < n_c$, then $P_{\mathbf{J}_{\bar{n}}}(e) = 0$: If the true allocation is e , the sample is guaranteed to contain a batch with difference greater than t . Combining this with [3.40] shows that for any $e \in \mathbf{E}$,

$$P_{\mathbf{J}_{\bar{n}}}(e) = \prod_{c=1}^C \prod_{k=1}^{\#_c \mathbf{G}(e)} \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \quad (3.41)$$

Let

$$p_{kc} \equiv \left(\frac{N_c - n_c - k + 1}{N_c - k + 1} \vee 0 \right). \quad (3.42)$$

Note that

$$p_{kc} \geq p_{k'c} \quad \text{if } k < k'. \quad (3.43)$$

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

$$P_{\mathbf{J}_{\bar{n}}}(e) = \prod_{c=1}^C \prod_{k=1}^{\#_c \mathbf{G}(e)} p_{kc} = \prod_{c=1}^C \prod_{k=1}^{N_c} p_{kc}^{g_{kc}(e)}. \quad (3.44)$$

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

$$q_{kc} \equiv \begin{cases} -\log(p_{kc}), & p_{kc} > 0, \\ \infty, & p_{kc} = 0. \end{cases} \quad (3.45)$$

Note that $q_{kc} \geq 0$ for all batches (k, c) . By [3.43],

$$q_{kc} \leq q_{k'c} \quad \text{if } k < k'. \quad (3.46)$$

From [3.44] and [3.45], for $e \in \tilde{\mathbf{E}}$,

$$-\log(P_{\mathbf{J}_{\bar{n}}}(e)) = q \cdot g(e). \quad (3.47)$$

Equations [3.39] and [3.47] yield equation [3.27]. We prove [3.28] in appendix B.1.

3.4 Approximate and exact solutions to KP

Dynamic programming algorithms and branch and bound algorithms can solve KP (Pisinger & Toth 1998). Appendix B.2 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 & 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.2. 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

$$\mathbf{X}^{rel} \equiv \{(x_j)_{j=1}^N : x_j \in [0, 1]\} \supset \mathbf{X}. \quad (3.48)$$

The LKP is to find

$$\lambda_{\text{LKP}} \equiv \min_{x \in \mathbf{X}^{rel}} \{q \cdot x : u \cdot x \geq M\}. \quad (3.49)$$

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

$$r_{kc} \equiv \frac{q_{kc}}{u_{kc}} \quad (3.50)$$

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 [3.36] and [3.46] show that

$$r_{kc} \leq r_{k'c} \text{ if } k < k'. \quad (3.51)$$

Recall [3.29] and [3.30], the mappings between double indices and single indices. Let

$\pi : \{1, 2, \dots, N\} \rightarrow \{1, 2, \dots, N\}$ be a permutation such that

$$r_{\pi(1)} \leq r_{\pi(2)} \leq \dots \leq r_{\pi(n)}. \quad (3.52)$$

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 $k < k'$,

$$\pi^{-1}(j(k, c)) < \pi^{-1}(j(k', c)). \quad (3.53)$$

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

$$\pi^{-1}(j(k, c)) < \pi^{-1}(j(k^*, c^*)). \quad (3.54)$$

The first condition, together with [3.51] and [3.52], 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 $j' \in \{1, \dots, N\}$ with $u_{\pi(j')} > 0$,

$$(\mathbf{1}[\pi^{-1}(j) \leq j'])_{j=1}^N \in \tilde{\mathbf{X}}. \quad (3.55)$$

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 [3.52] and [3.53], if $\pi^{-1}(j(k', c)) \leq j'$ and $k < k'$, then $\pi^{-1}(j(k, c)) < j'$. That is, if $e_{k'c}^* = \omega_{k'c}$ and $k < k'$, then $e_{kc}^* = \omega_{kc}$. Thus, $e^* \in \tilde{\mathbf{E}}$. By [3.33] and [3.35],

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

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

$$\lambda_{\text{LKP}} = \begin{cases} 0, & M = 0 \text{ and } u_{\pi(1)} = 0, \\ \sum_{j=1}^{B-1} q_{\pi(j)} + \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}} \cdot q_{\pi(B)}, & \text{otherwise.} \end{cases} \quad (3.56)$$

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

$$x_{\pi(j)}^{\text{rel}} \equiv \begin{cases} 1, & j < B, \\ \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}, & j = B, \\ 0, & \text{otherwise.} \end{cases} \quad (3.57)$$

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

$$\text{If } \sum_{j=1}^B u_{\pi(j)} = M \text{ then } \lambda = \sum_{j=1}^B q_{\pi(j)}. \quad (3.58)$$

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

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

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

$$\lambda_{LKP} + \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} = \lambda_{LKP}^+. \quad (3.59)$$

Thus,

$$\lambda - \lambda_{LKP} \leq \lambda_{LKP}^+ - \lambda_{LKP} = \left(1 - \frac{M - \sum_{j=1}^{B-1} u_{\pi(j)}}{u_{\pi(B)}}\right) q_{\pi(B)} \leq q_{\pi(B)}, \quad (3.60)$$

and so

$$\frac{\exp(-\lambda_{LKP})}{\exp(-\lambda)} \leq \frac{1}{p_{\pi(B)}}. \quad (3.61)$$

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

3.5 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 & Wolff (2007).

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

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

Table 3.1: 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

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 3.1 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 3.1 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).

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.

3.6 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

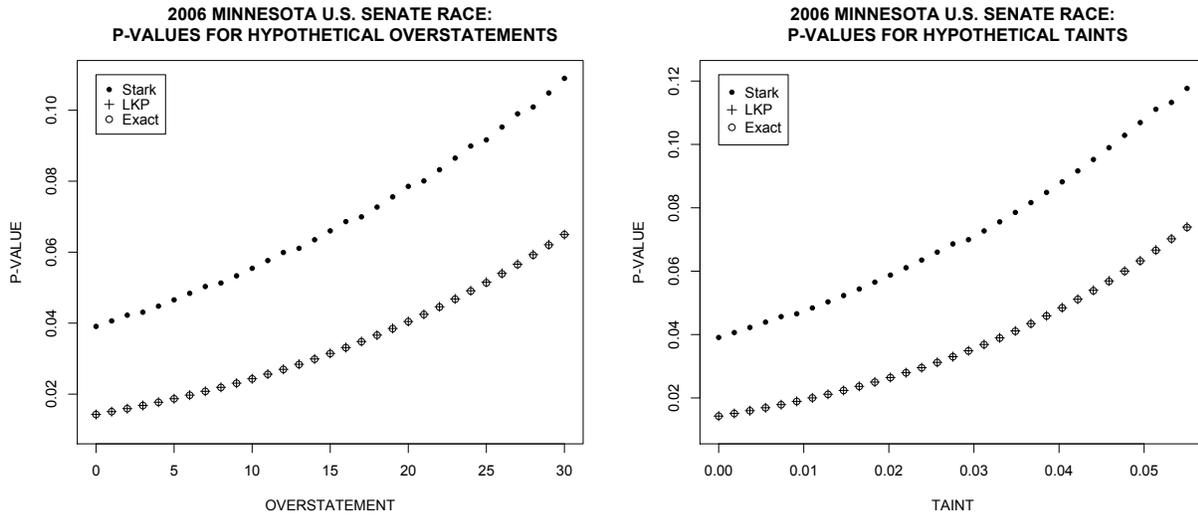


Figure 3.1: Exact and conservative p -values for hypothetical maximum observed overstaterments (left) and hypothetical maximum observed taints (right) for the 2006 Minnesota Senate race. The LKP conservative p -values (P_{LKP}) are nearly identical to the exact p -values ($P_{\#}$). Both are substantially smaller than bounds using the method in Stark (2008b).

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 3.7 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 , $c = 1, \dots, C$. 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,

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

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

$$\sum_{c=1}^C n_c^\dagger = \min \left\{ \sum_{c=1}^C n_c : (n_c)_{c=1}^C \in \mathbf{N}(\alpha, t^*) \right\}. \quad (3.62)$$

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

$$\sum_{c=1}^C \frac{n_c}{N_c} \sum_{k=1}^{N_c} b_{kc}. \quad (3.63)$$

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 $P_{\#}(t^*; \vec{n}) \leq \alpha$.

3.6.1 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 3.7.

3.6.2 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 [3.45] 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 $\mathbf{S} = \{1, \dots, N\}$.

Step 2: (Update q , r , and π)

Using the current value of \vec{n} , compute $(q_{kc})_{k=1}^{N_c} \quad C_{c=1}$.
 Set $q_{kc} = \min(q_{kc}, \log(n_c + 1))$.
 Compute $(r_j)_{j=1}^N$.
 Find the permutation π satisfying [3.52], [3.53], and [3.54].

Step 3: (Choose which n_c to increment)

Find $j = \min\{j' : \pi(j') \in \mathbf{S}\}$.
 Increment $n_{c(\pi(j))}$ (see equation [3.30]).

Step 4: (Update the search set.)

If `next.r`, set $\mathbf{S} = \mathbf{S} \setminus \pi(j)$.
 Else if `first.r`, do nothing.

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 [3.46],

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

with equality if and only if $k = N_c - n_c$. Thus, the ordering in [3.46] continues to hold.

Step 5: (Terminate?)

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

By [3.52] and [3.53], 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.

3.6.3 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.

3.7 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.

3.7.1 The 2006 Minnesota U.S. Senate race

The statutory audit of the 2006 Minnesota election examined 202 precincts. As discussed in section 3.5, 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 `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 3.2, 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

Table 3.2: 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 3.1 (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	<code>first.r</code>	<code>next.r</code>
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

(almost 40%) and would reduce the expected number of ballots to audit by one third (see equation [3.63]). The new methods draw more than 8 precincts from Hennepin county and only one precinct from the smallest counties, instead of two.

Figure 3.2 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.

3.7.2 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 contests⁵ 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 could not have overstated the apparent margin. Table 3.3 lists the results, along with summary statistics such the number of counties and precincts in the

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

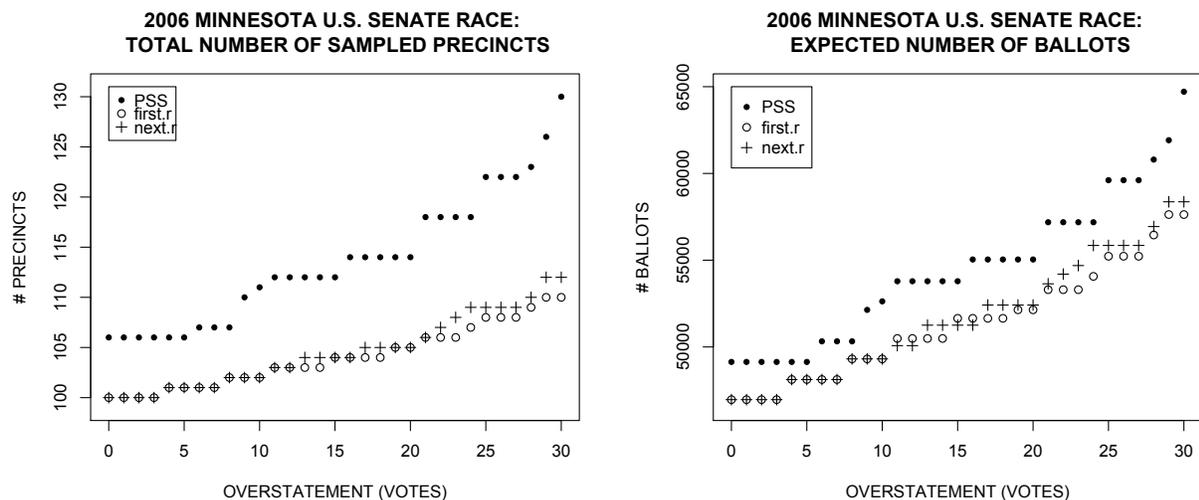


Figure 3.2: Number of batches to audit and expected number of ballots to audit to get p -values no larger than 0.05 for 2006 Minnesota Senate race, as observed maximum overstatements range from 0 to 30 votes, using sample size vectors selected by PSS, `first.r`, and `next.r`. In these simulations, PSS requires more auditing than `first.r` and `next.r`, which have nearly identical workloads.

contest and the margin of victory as a percentage of votes cast in the contest. Figures 3.3 and 3.4 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.

Table 3.3: 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`.

CD	Contest summary					Precincts to audit				Expected ballots to audit		
	C	N	max N_c	Votes	M (%)	PSS	f.r	n.r	Opt	PSS	f.r	n.r
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

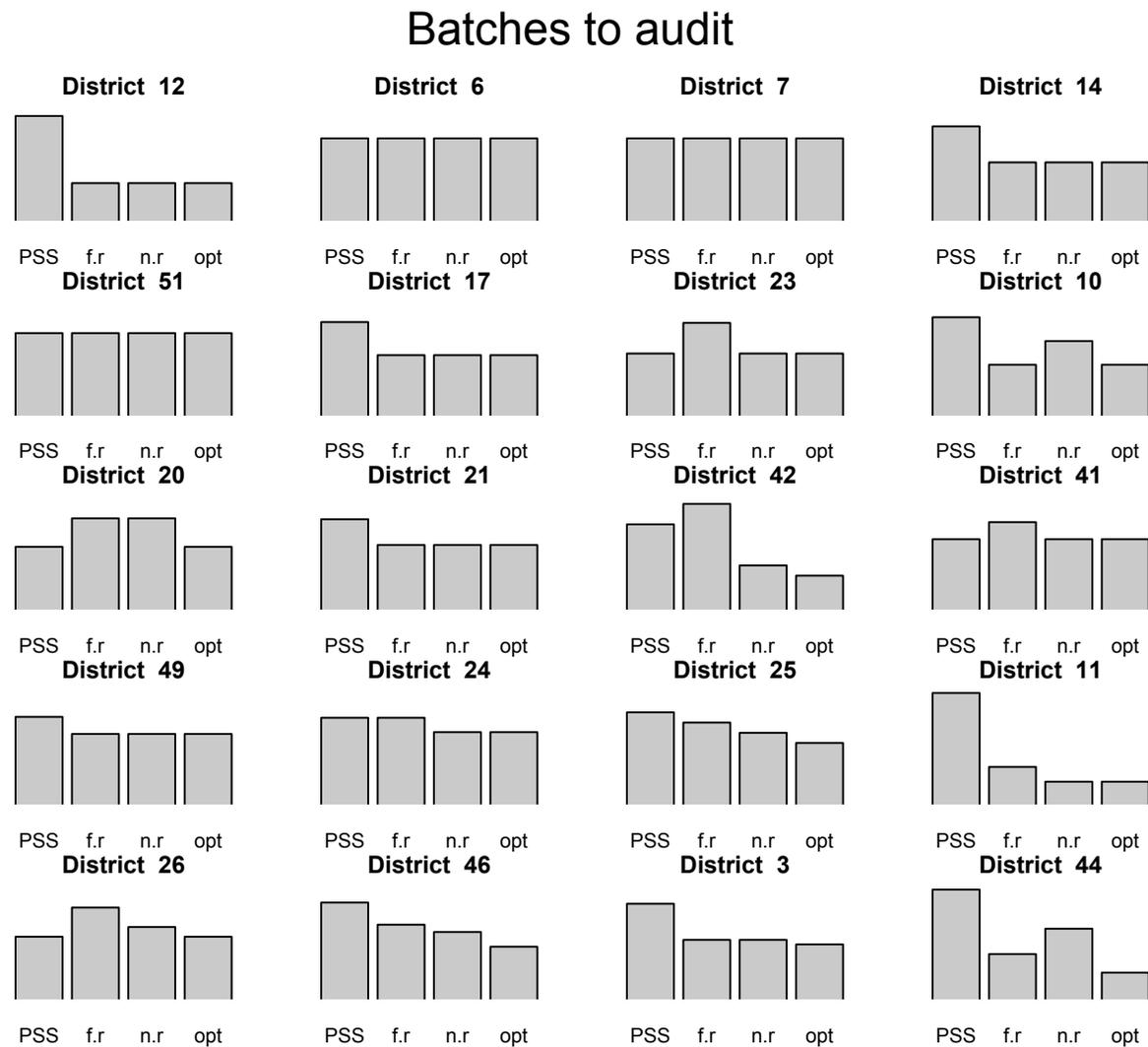


Figure 3.3: Number of batches to audit so that the p -value of the hypothesis that the outcome is incorrect will be less than $\alpha = 0.05$ if the sample finds no difference that overstated a margin. Bar graphs plot the ratio of the number of batches to audit for sample size vectors chosen using PSS, `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.

Expected number of ballots to audit

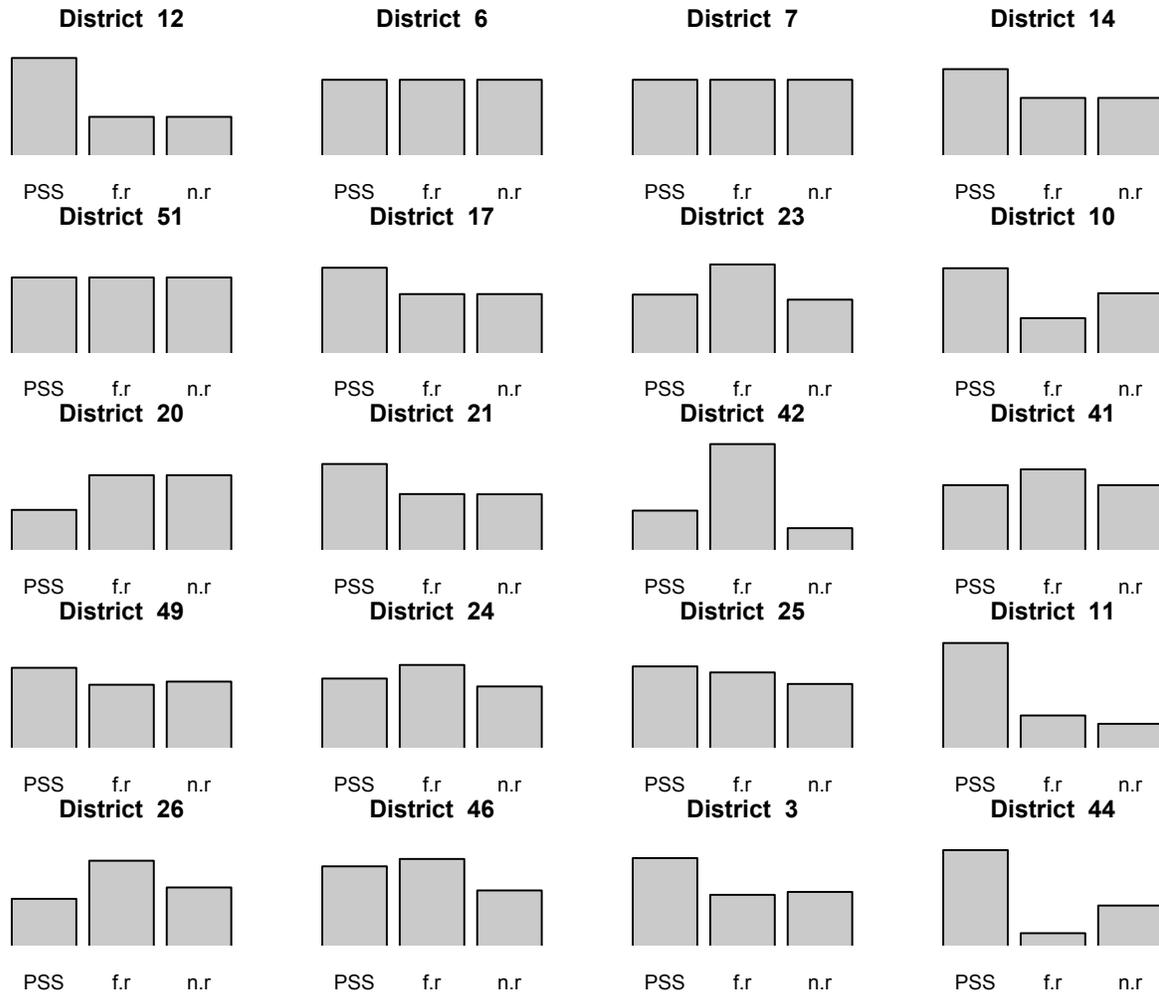


Figure 3.4: Expected number of ballots to audit so that the p -value of the hypothesis that the outcome is incorrect will be less than $\alpha = 0.05$ if the sample finds no difference that overstated a margin). Bar graphs plot the ratio of the expected number of ballots for PSS and first .r to the expected 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.

3.8 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 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 & 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 & 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.

Bibliography

- Abadie, A., & Imbens, G. 2008, *Annales d'Economie et de Statistique*, No. 91-92, 175
- Aslam, J. A., Popa, R. A., & Rivest, R. L. 2007, in *Proc. 2007 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT07)*
- Cochran, W. 1977, *Sampling techniques* (New York, NY: Wiley)
- Dantzig, G. 1957, *Operations Research*, 5, 266
- Fienberg, S. E., Neter, J., & Leitch, R. A. 1977, *Journal of the American Statistical Association*, 72, 295
- Financial Audit Manual. 2008, *Financial Audit Manual*, United States Government Accountability Office, volume 1. <http://www.gao.gov/new.items/d08585g.pdf>
- Fisher, R. A. 1926, *Journal of the Ministry of Agriculture of Great Britain*, 33, 503
- Gonzalez, T. F. 1985, *Theoretical Computer Science*, 38, 293
- Greevy, R., Lu, B., Silber, J. H., & Rosenbaum, P. 2004, *Biostatistics*, 5, 263
- Guthrie, D., et al. 1989, *Statistical Science*, 4, 2
- Halvorson, M., & Wolff, L. 2007, *Report and analysis of the 2006 post-election audit of Minnesota voting systems*, <http://ceimn.org/files/CEIMNAuditReport2006.pdf>, retrieved 30 May 2011
- Hardy, G., Littlewood, J., & Pólya, G. 1952, *Inequalities*, 2nd edn. (Cambridge University Press), 324
- Higgins, M., Rivest, R., & Stark, P. 2011, *Statistics, Politics, and Policy*, 2, article 7
- Hochbaum, D., & Shmoys, D. 1986, *Journal of the ACM (JACM)*, 33, 533
- Holland, P. W. 1986, *Journal of the American statistical Association*, 81, 945
- Imai, K. 2008, *Statistics in medicine*, 27, 4857
- Imai, K., King, G., & Nall, C. 2009, *Statistical Science*, 24, 29
- Imbens, G. W. 2011, working Paper
- Karp, R. 2010, in *50 Years of Integer Programming 1958–2008*, ed. M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, & L. A. Wolsey (New York: Springer), 219
- Lock Morgan, K., & Rubin, D. B. 2012, *Annals of Statistics*, 40, 1263
- Lohr, S. 1999, *Sampling: Design and Analysis* (Pacific Grove, CA: Duxbury Press)
- Lu, B., Greevy, R., Xu, X., & Beck, C. 2011, *The American Statistician*, 65, 21
- Lu, B., & Rosenbaum, P. 2004, *Journal of computational and graphical statistics*, 13, 422
- McLaughlin, K., & Stark, P. 2011, *Workload Estimates for Risk-Limiting Audits of Large Con-*

- tests, <http://statistics.berkeley.edu/~stark/Preprints/workload11.pdf>, retrieved 9 July 2011
- Miratrix, L., & Stark, P. 2009, *IEEE Transactions on Information Forensics and Security*, 4, 974
- Miratrix, L. W., Sekhon, J. S., & Yu, B. 2013, *Journal of the Royal Statistical Society, Series B*, 75, 369
- Moore, R. T. 2012, *Political Analysis*, 20, 460
- 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
- Neyman, J. 1935, *Supplement of Journal of the Royal Statistical Society*, 2, 107–180
- Pisinger, D. 1995, PhD thesis, University of Copenhagen, Denmark
- Pisinger, D., & Toth, P. 1998, in *Handbook of Combinatorial Optimization*, ed. D.-Z. Du & P. Pardalos, Vol. 1 (Kluwer, Dordrecht, The Netherlands), 299
- Rivest, R. 2007, On auditing elections when precincts have different sizes, <http://people.csail.mit.edu/rivest/Rivest-OnAuditingElectionsWhenPrecinctsHaveDifferentSizes.pdf>, retrieved 30 May 2011
- Rosenbaum, P. 1989, *Journal of the American Statistical Association*, 84, 1024
- Rubin, D. B. 1974, *Journal of Educational Psychology*; *Journal of Educational Psychology*, 66, 688
- Splawa-Neyman, J., Dabrowska, D., & Speed, T. 1990, *Statistical Science*, 5, 465
- Stark, P. 2008a, *Ann. Appl. Stat.*, 2, 550
- . 2008b, *Ann. Appl. Stat.*, 2, 982
- . 2009a, *IEEE Transactions on Information Forensics and Security*, 4, 708
- . 2009b, Efficient post-election audits of multiple contests: 2009 California tests, <http://ssrn.com/abstract=1443314>, 2009 Conference on Empirical Legal Studies
- . 2009c, *IEEE Transactions on Information Forensics and Security*, 4, 1005
- . 2010, *Chance*, 23, 9
- Talens, E. 2005, PhD thesis, University of Groningen, The Netherlands
- Wendell, J., & Schmee, J. 1996, *Journal of the American Statistical Association*, 91, 825

Appendix A

Appendix for Chapter 2

A.1 Proof of Lemmas 6 and 7

To help the reader, we suppress the block index in the following derivations, identifying units by a single index. The variance calculations for the difference-in means estimators follow Miratrix et al. (2013) closely. These proofs also uses methods found in Cochran (1977) and Lohr (1999).

Note that the following expectations hold under complete randomization:

$$\begin{aligned}\mathbb{E}\left(\frac{T_{is}}{\#T_s}\right) &= \mathbb{E}\left[\mathbb{E}\left(\frac{T_{is}}{\#T_s} \mid \#T_s\right)\right] \\ &= \mathbb{E}\left(\frac{\#T_s}{n}\right) = \mathbb{E}\left(\frac{1}{n}\right) = \frac{1}{n}\end{aligned}\tag{A.1}$$

$$\begin{aligned}\mathbb{E}\left(\frac{T_{is}}{(\#T_s)^2}\right) &= \mathbb{E}\left[\mathbb{E}\left(\frac{T_{is}}{(\#T_s)^2} \mid \#T_s\right)\right] \\ &= \mathbb{E}\left(\frac{\#T_s}{n(\#T_s)^2}\right) = \mathbb{E}\left(\frac{1}{n\#T_s}\right) = \frac{1}{n}\mathbb{E}\left(\frac{1}{\#T_s}\right)\end{aligned}\tag{A.2}$$

$$\begin{aligned}\mathbb{E}\left(\frac{T_{is}T_{jt}}{\#T_s\#T_t}\right) &= \mathbb{E}\left[\mathbb{E}\left(\frac{T_{is}T_{jt}}{\#T_s\#T_t} \mid \#T_s, \#T_t\right)\right] \\ &= \mathbb{E}\left(\frac{\#T_s\#T_t}{n(n-1)}\right) = \mathbb{E}\left(\frac{1}{n(n-1)}\right) = \frac{1}{n(n-1)}\end{aligned}\tag{A.3}$$

We first compute the expectation of the block level estimator $\hat{\mu}_{s,\text{diff}}$.

$$\mathbb{E}(\hat{\mu}_{s,\text{diff}}) = \mathbb{E}\left(\sum_{i=1}^n \frac{y_{is}T_{is}}{\#T_s}\right) = \sum_{i=1}^n y_{is}\mathbb{E}\left(\frac{T_{is}}{\#T_s}\right) = \sum_{i=1}^n \frac{y_{is}}{n} = \mu_s\tag{A.4}$$

We now derive the variance of this estimator. Observe that:

$$\begin{aligned}
\mathbb{E}(\hat{\mu}_{s,\text{diff}}^2) &= \mathbb{E}\left[\left(\sum_{i=1}^n \frac{y_{is}T_{is}}{\#T_s}\right)^2\right] = \mathbb{E}\left[\sum_{i=1}^n \frac{y_{is}^2T_{is}^2}{(\#T_s)^2} + \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is}y_{js}T_{is}T_{js}}{(\#T_s)^2}\right] \\
&= \mathbb{E}\left[\sum_{i=1}^n \frac{y_{is}^2T_{is}}{(\#T_s)^2} + \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is}y_{js}T_{is}T_{js}}{(\#T_s)^2}\right] \\
&= \sum_{i=1}^n y_{is}^2 \mathbb{E}\left(\frac{T_{is}}{(\#T_s)^2}\right) + \sum_{i=1}^n \sum_{j \neq i} y_{is}y_{js} \mathbb{E}\left(\frac{T_{is}T_{js}}{(\#T_s)^2}\right) \\
&= \frac{1}{n} \mathbb{E}\left(\frac{1}{\#T_s}\right) \sum_{i=1}^n y_{is}^2 + \sum_{i=1}^n \sum_{j \neq i} y_{is}y_{js} \left(\frac{1}{n(n-1)} - \frac{1}{n(n-1)} \mathbb{E}\left(\frac{1}{\#T_s}\right)\right) \\
&= \frac{1}{n} \mathbb{E}\left(\frac{1}{\#T_s}\right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\left(\sum_{i=1}^n y_{is}\right)^2 - \sum_{i=1}^n y_{is}^2\right) \left(\frac{1}{n(n-1)} - \frac{1}{n(n-1)} \mathbb{E}\left(\frac{1}{\#T_s}\right)\right) \\
&= \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is}\right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&\quad + \mathbb{E}\left(\frac{1}{\#T_s}\right) \left(\left(\frac{1}{n} + \frac{1}{n(n-1)}\right) \sum_{i=1}^n y_{is}^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is}\right)^2\right) \\
&= \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is}\right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&\quad + \mathbb{E}\left(\frac{1}{\#T_s}\right) \left(\frac{1}{n-1} \sum_{i=1}^n y_{is}^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is}\right)^2\right) \\
&= \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is}\right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&\quad + \mathbb{E}\left(\frac{1}{\#T_s}\right) \left(\frac{1}{n-1} \sum_{i=1}^n y_{is}^2 - \frac{n}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n}\right)^2\right)
\end{aligned}$$

(A.5)

Continuing on:

$$\begin{aligned}
& \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
& + \mathbb{E} \left(\frac{1}{\#T_s} \right) \left(\frac{1}{n-1} \sum_{i=1}^n y_{is}^2 - \frac{n}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) \\
= & \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
& + \mathbb{E} \left(\frac{1}{\#T_s} \right) \left(\frac{1}{n-1} \sum_{i=1}^n y_{is}^2 - \frac{n}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) \\
= & \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
& + \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \left(\sum_{i=1}^n y_{is}^2 - n\mu_s^2 \right) \\
= & \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 + \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n (y_{is} - \mu_s)^2
\end{aligned} \tag{A.6}$$

It follows from the property that $\text{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$ that:

$$\begin{aligned}
\text{Var}(\hat{\mu}_{s,\text{diff}}) &= \mathbb{E}(\hat{\mu}_s^2) - (\mathbb{E}(\hat{\mu}_s))^2 \\
&= \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&\quad + \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n (y_{is} - \mu_s)^2 - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{n}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&\quad + \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n (y_{is} - \mu_s)^2 - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{1}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 + \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n (y_{is} - \mu_s)^2 \\
&= \frac{1}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n (y_{is} - \mu_s)^2 - \left(\frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 - \frac{1}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) \\
&= \frac{n}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sum_{i=1}^n \frac{(y_{is} - \mu_s)^2}{n} - \left(\frac{1}{n-1} \sum_{i=1}^n \frac{y_{is}^2}{n} - \frac{1}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) \\
&= \frac{n}{n-1} \mathbb{E} \left(\frac{1}{\#T_s} \right) \sigma_s^2 - \frac{1}{n-1} \sigma_s^2 \\
&= \frac{n}{n-1} \sigma_s^2 \left(\mathbb{E} \left(\frac{1}{\#T_s} \right) - \frac{1}{n} \right) \tag{A.7}
\end{aligned}$$

Since, under complete randomization:

$$\begin{aligned}
\mathbb{E} \left(\frac{1}{\#T_s} \right) &= \frac{z}{r} \left(\frac{1}{\lfloor n/r \rfloor + 1} \right) + \left(1 - \frac{z}{r} \right) \left(\frac{1}{\lfloor n/r \rfloor} \right) \\
&= \frac{z \lfloor n/r \rfloor}{r(\lfloor n/r \rfloor)(\lfloor n/r \rfloor + 1)} + \frac{(r-z)(\lfloor n/r \rfloor + 1)}{r(\lfloor n/r \rfloor)(\lfloor n/r \rfloor + 1)} \tag{A.8}
\end{aligned}$$

it follows that:

$$\begin{aligned}
\text{Var}(\hat{\mu}_{s,\text{diff}}) &= \frac{n}{n-1} \sigma_s^2 \left(\mathbb{E} \left(\frac{1}{\#T_s} \right) - \frac{1}{n} \right) \\
&= \frac{n}{n-1} \sigma_s^2 \left(\frac{nr + r^2 - 2rz}{(n-z)(n+r-z)} - \frac{1}{n} \right) \\
&= \frac{n}{n-1} \frac{n^2r + nr^2 - 2nrz - (n-z)(n+r-z)}{n(n-z)(n+r-z)} \sigma_s^2 \\
&= \frac{nr(n+r-z) - nrz - (n-z)(n+r-z)}{(n-1)(n-z)(n+r-z)} \sigma_s^2 \\
&= \frac{n^2r + nr^2 - nrz - nrz - n^2 - nr + nz + nz + zr - z^2}{(n-1)(n-z)(n+r-z)} \sigma_s^2 \\
&= \frac{n^2r + nr^2 - 2nrz - n^2 - nr + 2nz + zr - z^2}{(n-1)(n-z)(n+r-z)} \sigma_s^2 \\
&= \frac{n^2(r-1) - 2nz(r-1) + nr(r-1) + z(r-z)}{(n-1)(n-z)(n+r-z)} \sigma_s^2 \\
&= \frac{(r-1)n(n+r-2z) + z(r-z)}{(n-1)(n-z)(n+r-z)} \sigma_s^2
\end{aligned} \tag{A.9}$$

Now we work on computing covariances of this estimator. Note that:

$$\begin{aligned}
&\mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \sum_{i=1}^n \frac{y_{it} T_{it}}{\#T_t} \right) \\
&= \mathbb{E} \left(\sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} T_{is} T_{jt}}{\#T_s \#T_t} \right) + \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} y_{it} T_{is} T_{it}}{\#T_s \#T_t} \right) \\
&= \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{jt} \mathbb{E} \left(\frac{T_{is} T_{jt}}{\#T_s \#T_t} \right) + \sum_{i=1}^n y_{is} y_{it} \mathbb{E} \left(\frac{T_{is} T_{it}}{\#T_s \#T_t} \right) \\
&= \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{jt} \frac{1}{n(n-1)} + 0 = \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{n(n-1)}
\end{aligned} \tag{A.10}$$

Now, recall the property $\text{cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$. It follows that:

$$\begin{aligned}
\text{cov}(\hat{\mu}_{s,\text{diff}}, \hat{\mu}_{t,\text{diff}}) &= \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_{is}} \sum_{i=1}^n \frac{y_{it} T_{it}}{\#T_{it}} \right) - \sum_{i=1}^n \frac{y_{is}}{n} \sum_{i=1}^n \frac{y_{it}}{n} \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{n(n-1)} - \sum_{i=1}^n \frac{y_{is}}{n} \sum_{i=1}^n \frac{y_{it}}{n} \\
&= \frac{1}{n(n-1)} \sum_{i=1}^n y_{is} \sum_{i=1}^n y_{it} - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is} y_{it} - \frac{1}{n^2} \sum_{i=1}^n y_{is} \sum_{i=1}^n y_{it} \\
&= \frac{1}{n^2(n-1)} \sum_{i=1}^n y_{is} \sum_{i=1}^n y_{it} - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is} y_{it} \\
&= \frac{-1}{n-1} \left(\sum_{i=1}^n \frac{y_{is} y_{it}}{n} - \sum_{i=1}^n \frac{y_{is}}{n} \sum_{i=1}^n \frac{y_{it}}{n} \right) = \frac{-\gamma_{st}}{n-1} \tag{A.11}
\end{aligned}$$

Note that the expressions for the variance and covariance of the difference-in-means estimator are slightly different than those in Miratrix et al. (2013) by a factor of $\frac{n-1}{n}$.

We now turn our attention to the Horvitz-Thompson estimator. Note the following expectations:

$$\mathbb{E}(T_{is} T_{js}) = \frac{n(n-r) + z(r-z)}{r^2 n(n-1)} \tag{A.12}$$

$$\mathbb{E}(T_{is} T_{jt}) = \frac{n^2(r-1) - z(r-z)}{r^2(r-1)n(n-1)} \tag{A.13}$$

Under complete randomization, the expectation of the Horvitz-Thompson estimator is:

$$\mathbb{E}(\hat{\mu}_{s,\text{HT}}) = \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{n/r} \right) = \sum_{i=1}^n \frac{y_{is} \mathbb{E}(T_{is})}{n/r} = \sum_{i=1}^n \frac{y_{is} (1/r)}{n/r} = \sum_{i=1}^n \frac{y_{is}}{n} = \mu_s \tag{A.14}$$

The variance of this estimator is derived as follows. Note that:

$$\begin{aligned}
\mathbb{E}(\hat{\mu}_{s,\text{HT}}^2) &= \mathbb{E} \left(\left(\sum_{i=1}^n \frac{y_{is} T_{is}}{n/r} \right)^2 \right) \\
&= \left(\frac{r^2}{n^2} \right) \left(\mathbb{E} \left(\sum_{i=1}^n y_{is}^2 T_{is}^2 \right) + \mathbb{E} \left(\sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js} \right) \right) \\
&= \left(\frac{r^2}{n^2} \right) \left(\mathbb{E} \left(\sum_{i=1}^n y_{is}^2 T_{is} \right) + \mathbb{E} \left(\sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js} \right) \right) \\
&= \left(\frac{r^2}{n^2} \right) \left(\sum_{i=1}^n y_{is}^2 \mathbb{E}(T_{is}) + \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \mathbb{E}(T_{is} T_{js}) \right) \\
&= \left(\frac{r^2}{n^2} \right) \left(\sum_{i=1}^n \frac{y_{is}^2}{r} + \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \frac{n(n-r) + z(r-z)}{r^2 n(n-1)} \right) \\
&= \left(\frac{r^2}{n^2} \right) \left[\sum_{i=1}^n \frac{y_{is}^2}{r} + \left(\frac{n(n-r) + z(r-z)}{r^2 n(n-1)} \right) \left(\left(\sum_{i=1}^n y_{is} \right)^2 - \sum_{i=1}^n y_{is}^2 \right) \right] \\
&= \left(\frac{r}{n^2} - \frac{n(n-r) + z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\frac{n(n-r) + z(r-z)}{n^3(n-1)} \right) \left(\sum_{i=1}^n y_{is} \right)^2 \\
&= \left(\frac{rn(n-1) - n(n-r) - z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\frac{n(n-r) + z(r-z)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2
\end{aligned} \tag{A.15}$$

It follows that:

$$\begin{aligned}
\text{Var}(\hat{\mu}_{s,\text{HT}}) &= \mathbb{E}(\hat{\mu}_{s,\text{HT}}^2) - (\mathbb{E}(\hat{\mu}_{s,\text{HT}}))^2 \\
&= \left(\frac{rn(n-1) - n(n-r) - z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\frac{n(n-r) + z(r-z)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \left(\frac{rn(n-1) - n(n-r) - z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\frac{n(n-r) + z(r-z) - n(n-1)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \left(\frac{rn^2 - nr - n^2 + nr - z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad + \left(\frac{n(1-r) + z(r-z)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \left(\frac{n^2(r-1) - z(r-z)}{n^2(n-1)} \right) \sum_{i=1}^n \frac{y_{is}^2}{n} \\
&\quad - \left(\frac{n(r-1) - z(r-z)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2
\end{aligned} \tag{A.16}$$

We can simplify even further:

$$\begin{aligned}
& \left(\frac{n^2(r-1) - z(r-z)}{n^2(n-1)} \right) \sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\frac{n(r-1) - z(r-z)}{n(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{r-1}{n-1} \left(\sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) - \frac{z(r-z)}{n^2(n-1)} \sum_{i=1}^n \frac{y_{is}^2}{n} + \frac{z(r-z)}{n(n-1)} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{r-1}{n-1} \left(\sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) - \frac{z(r-z)}{n^2(n-1)} \left(\sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \right) \\
&\quad + \left(\frac{z(r-z)}{n(n-1)} - \frac{z(r-z)}{n^2(n-1)} \right) \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{r-1}{n-1} \sigma_s^2 - \frac{z(r-z)}{n^2(n-1)} \sigma_s^2 + \frac{(n-1)z(r-z)}{n^2(n-1)} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{n^2(r-1) - z(r-z)}{n^2(n-1)} \sigma_s^2 + \frac{z(r-z)}{n^2} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \tag{A.17}
\end{aligned}$$

Now we compute the covariance. Note that:

$$\begin{aligned}
& \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{n/r} \sum_{i=1}^n \frac{y_{it} T_{it}}{n/r} \right) \\
&= \mathbb{E} \left(\sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} T_{is} T_{jt}}{(n/r)^2} \right) + \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} y_{it} T_{is} T_{it}}{(n/r)^2} \right) \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{(n/r)^2} \mathbb{E}(T_{is} T_{jt}) + \sum_{i=1}^n \frac{y_{is} y_{it}}{(n/r)^2} \mathbb{E}(T_{is} T_{it}) \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{(n/r)^2} \frac{n^2(r-1) - z(r-z)}{r^2(r-1)n(n-1)} + 0 \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} (n^2(r-1) - z(r-z))}{(r-1)n^3(n-1)} \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{n(n-1)} - \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} z(r-z)}{(r-1)n^3(n-1)} \tag{A.18}
\end{aligned}$$

Thus, using the covariance calculation from the difference-in-means estimator, we have:

$$\begin{aligned}
\text{COV}(\hat{\mu}_{s,\text{HT}}, \hat{\mu}_{t,\text{HT}}) &= \mathbb{E} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{n/r} \sum_{i=1}^n \frac{y_{it} T_{it}}{n/r} \right) - \sum_{i=1}^n \frac{y_{is}}{n} \sum_{i=1}^n \frac{y_{it}}{n} \\
&= \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt}}{n(n-1)} - \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} z(r-z)}{(r-1)n^3(n-1)} - \sum_{i=1}^n \frac{y_{is}}{n} \sum_{i=1}^n \frac{y_{it}}{n} \\
&= \frac{-\gamma_{st}}{n-1} - \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{jt} z(r-z)}{(r-1)n^3(n-1)} \tag{A.19}
\end{aligned}$$

This proves the two lemmas.

A.2 Proof of Lemma 10

To help the reader, we suppress the block index in the following derivations, identifying units by a single index.

Note that, under complete randomization and when $i \neq j$, the following expectations hold:

$$\begin{aligned}
\mathbb{E} \left(\frac{T_{is}}{\#T_s - 1} \right) &= \mathbb{E} \left[\mathbb{E} \left(\frac{T_{is}}{\#T_s - 1} \middle| \#T_s \right) \right] \\
&= \mathbb{E} \left(\frac{\frac{\#T_s}{n}}{\#T_s - 1} \right) = \frac{1}{n} \mathbb{E} \left(\frac{\#T_s}{\#T_s - 1} \right) \\
&= \frac{1}{n} \mathbb{E} \left(\frac{\#T_s - 1}{\#T_s - 1} \right) + \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \\
&= \frac{1}{n} + \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \tag{A.20}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E} \left(\frac{T_{is}}{\#T_s(\#T_s - 1)} \right) &= \mathbb{E} \left[\mathbb{E} \left(\frac{T_{is}}{\#T_s(\#T_s - 1)} \middle| \#T_s \right) \right] \\
&= \mathbb{E} \left(\frac{\frac{\#T_s}{n}}{\#T_s(\#T_s - 1)} \right) = \frac{1}{n} \mathbb{E} \left(\frac{\#T_s}{\#T_s(\#T_s - 1)} \right) \\
&= \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \tag{A.21}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E} \left(\frac{T_{is} T_{js}}{\#T_s(\#T_s - 1)} \right) &= \mathbb{E} \left[\mathbb{E} \left(\frac{T_{is} T_{js}}{\#T_s(\#T_s - 1)} \middle| \#T_s \right) \right] \\
&= \mathbb{E} \left(\frac{\frac{\#T_s}{n} \frac{\#T_s - 1}{n-1}}{\#T_s(\#T_s - 1)} \right) \\
&= \frac{1}{n(n-1)} \mathbb{E} \left(\frac{\#T_s(\#T_s - 1)}{\#T_s(\#T_s - 1)} \right) = \frac{1}{n(n-1)} \tag{A.22}
\end{aligned}$$

We first show that $\mathbb{E}(\hat{\sigma}_{s,\text{diff}}^2) = \frac{n}{n-1}\sigma_s^2$. The fact that $\mathbb{E}\left[\widehat{\text{Var}}(\hat{\mu}_{s,\text{diff}})\right] = \text{Var}(\hat{\mu}_{s,\text{diff}})$ follows immediately.

First, note that:

$$\begin{aligned}
& \sum_{i=1}^n T_{is} \left(y_{is} T_{is} - \sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 \\
&= \sum_{i=1}^n T_{is} (y_{is} T_{is})^2 - 2 \sum_{i=1}^n T_{is} \left(y_{is} T_{is} \sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right) + \sum_{i=1}^n T_{is} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 \\
&= \sum_{i=1}^n y_{is}^2 T_{is} - 2 \sum_{i=1}^n \left(y_{is} T_{is} \sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right) + \sum_{i=1}^n T_{is} \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 \\
&= \sum_{i=1}^n y_{is}^2 T_{is} - 2\#T_s \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 + \#T_s \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 \\
&= \sum_{i=1}^n y_{is}^2 T_{is} - \#T_s \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2 \tag{A.23}
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}(\hat{\sigma}_{s,\text{diff}}^2) &= \mathbb{E} \left(\frac{\sum_{i=1}^n T_{is} \left(y_{is} - \sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2}{\#T_s - 1} \right) \\
&= \mathbb{E} \left(\frac{\sum_{i=1}^n y_{is}^2 T_{is} - \#T_s \left(\sum_{i=1}^n \frac{y_{is} T_{is}}{\#T_s} \right)^2}{\#T_s - 1} \right) \\
&= \sum_{i=1}^n y_{is}^2 \mathbb{E} \left(\frac{T_{is}}{\#T_s - 1} \right) - \mathbb{E} \left(\frac{\left(\sum_{i=1}^n y_{is} T_{is} \right)^2}{\#T_{is} (\#T_{is} - 1)} \right) \tag{A.24}
\end{aligned}$$

Now:

$$\sum_{i=1}^n y_{is}^2 \mathbb{E} \left(\frac{T_{is}}{\#T_s - 1} \right) = \frac{1}{n} \sum_{i=1}^n y_{is}^2 + \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \sum_{i=1}^n y_{is}^2 \tag{A.25}$$

And:

$$\begin{aligned}
& \mathbb{E} \left(\frac{(\sum_{i=1}^n y_{is} T_{is})^2}{\#T_{is}(\#T_{is} - 1)} \right) \\
&= \mathbb{E} \left(\frac{\sum_{i=1}^n (y_{is} T_{is})^2}{\#T_{is}(\#T_{is} - 1)} \right) + \mathbb{E} \left(\frac{\sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js}}{\#T_{is}(\#T_{is} - 1)} \right) \\
&= \sum_{i=1}^n y_{is}^2 \mathbb{E} \left(\frac{T_{is}}{\#T_{is}(\#T_{is} - 1)} \right) + \sum_{j \neq i} y_{is} y_{js} \mathbb{E} \left(\frac{T_{is} T_{js}}{\#T_{is}(\#T_{is} - 1)} \right) \\
&= \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \sum_{i=1}^n y_{is}^2 + \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \\
&= \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \sum_{i=1}^n y_{is}^2 + \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 - \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2
\end{aligned} \tag{A.26}$$

So it follows that:

$$\begin{aligned}
\mathbb{E}(\hat{\sigma}_{s,\text{diff}}^2) &= \sum_{i=1}^n y_{is}^2 \mathbb{E} \left(\frac{T_{is}}{\#T_s - 1} \right) - \mathbb{E} \left(\frac{(\sum_{i=1}^n y_{is} T_{is})^2}{\#T_{is}(\#T_{is} - 1)} \right) \\
&= \frac{1}{n} \sum_{i=1}^n y_{is}^2 + \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \sum_{i=1}^n y_{is}^2 \\
&\quad - \frac{1}{n} \mathbb{E} \left(\frac{1}{\#T_s - 1} \right) \sum_{i=1}^n y_{is}^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 + \frac{1}{n(n-1)} \sum_{i=1}^n y_{is}^2 \\
&= \frac{1}{n-1} \sum_{i=1}^n y_{is}^2 - \frac{1}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 \\
&= \frac{n}{n-1} \sum_{i=1}^n \frac{y_{is}^2}{n} - \frac{n}{n-1} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 = \frac{n}{n-1} \sigma_s^2.
\end{aligned} \tag{A.27}$$

Now we show that $\mathbb{E} \left[\widehat{\text{Var}}(\hat{\mu}_{cs,HT}) \right] = \text{Var}(\hat{\mu}_{cs,HT})$. Note that the following expectations hold:

$$\begin{aligned}
\mathbb{E}((\#T_s)^2) &= \frac{z}{r} (\lfloor n/r \rfloor + 1)^2 + \left(1 - \frac{z}{r}\right) (\lfloor n/r \rfloor)^2 \\
&= \frac{z}{r} ((\lfloor n/r \rfloor)^2 + 2\lfloor n/r \rfloor + 1) + \left(1 - \frac{z}{r}\right) (\lfloor n/r \rfloor)^2 \\
&= (\lfloor n/r \rfloor)^2 + \frac{2z}{r} \lfloor n/r \rfloor + \frac{z}{r} = (\lfloor n/r \rfloor + z/r)^2 + (z/r - (z/r)^2) \\
&= (n/r)^2 + z/r(1 - z/r)
\end{aligned} \tag{A.28}$$

$$\begin{aligned}
\mathbb{E}((T_{is}T_{js})) &= \mathbb{E}[\mathbb{E}(T_{is}T_{js} | \#T_s)] = \mathbb{E}\left(\frac{\#T_s}{n} \frac{\#T_s - 1}{n-1}\right) \\
&= \frac{\mathbb{E}((\#T_s)^2) - \mathbb{E}(\#T_s)}{n(n-1)} \\
&= \frac{(n/r)^2 + z/r(1 - z/r) - n/r}{n(n-1)}
\end{aligned} \tag{A.29}$$

Starting from (A.16), we can write the block variance of the Horvitz-Thompson estimator as:

$$\begin{aligned}
\text{Var}(\hat{\mu}_{s,HT}) &= \left(\frac{n^2(r-1) - z(r-z)}{n^2(n-1)}\right) \sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\frac{n(r-1) - z(r-z)}{n(n-1)}\right) \left(\sum_{i=1}^n \frac{y_{is}}{n}\right)^2 \\
&= \left(\frac{n^2(r-1) - z(r-z)}{n^2(n-1)}\right) \sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\frac{n(r-1) - z(r-z)}{n^2(n-1)}\right) \sum_{i=1}^n \frac{y_{is}^2}{n} \\
&\quad - \left(\frac{n(r-1) - z(r-z)}{n^3(n-1)}\right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \\
&= \left(\frac{n(n-1)(r-1)}{n^2(n-1)}\right) \sum_{i=1}^n \frac{y_{is}^2}{n} - \left(\frac{n(r-1) - z(r-z)}{n^3(n-1)}\right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \\
&= \frac{r-1}{n^2} \sum_{i=1}^n y_{is}^2 - \left(\frac{n(r-1) - z(r-z)}{n^3(n-1)}\right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js}
\end{aligned} \tag{A.30}$$

It follows that:

$$\begin{aligned}
& \frac{r-1}{n^2} \sum_{i=1}^n \frac{y_{is}^2 T_{is}}{1/r} - \left(\frac{n(r-1) - z(r-z)}{n^3(n-1)} \right) \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{js} T_{is} T_{js}}{\frac{(n/r)^2 - (n/r) + z/r(1-z/r)}{n(n-1)}} \\
&= \frac{r(r-1)}{n^2} \sum_{i=1}^n y_{is}^2 T_{is} \\
&\quad - \left(\frac{n(r-1) - z(r-z)}{n^2((n/r)^2 - (n/r) + z/r(1-z/r))} \right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js} \\
&= \frac{r(r-1)}{n^2} \sum_{i=1}^n y_{is}^2 T_{is} - \left(\frac{nr^2(r-1) - r^2z(r-z)}{n^4 - n^3r + n^2z(r-z)} \right) \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} T_{is} T_{js}
\end{aligned} \tag{A.31}$$

is an unbiased estimate of the Horvitz-Thompson block-level variance.

A.3 Proof of Theorem 11

Consider a block of size n_c . Let \mathbf{S}_{n_c} denote an arbitrary subset of $\{1, \dots, n\}$ of size $|\mathbf{S}_{n_c}| = n_c$. Let $\mathbf{1}(i \in \mathbf{S}_{n_c})$ denote an indicator function: $\mathbf{1}(i \in \mathbf{S}_{n_c}) = 1$ if $i \in \mathbf{S}_{n_c}$; otherwise, $\mathbf{1}(i \in \mathbf{S}_{n_c}) = 0$. Possible values of the within-block variance σ_{cs}^2 are

$$\frac{1}{n_c} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) - \frac{1}{n_c^2} \left(\sum_{i \in \mathbf{S}_{n_c}} y_{is} \mathbf{1}(i \in \mathbf{S}_{n_c}) \right)^2 \tag{A.32}$$

and possible values of the variance $\sigma_{c(s+t)}^2$ are

$$\begin{aligned}
& \frac{1}{n_c} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) + y_{it}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) \\
& - \frac{1}{n_c^2} \left(\sum_{i \in \mathbf{S}_{n_c}} y_{is} \mathbf{1}(i \in \mathbf{S}_{n_c}) + y_{it} \mathbf{1}(i \in \mathbf{S}_{n_c}) \right)^2
\end{aligned} \tag{A.33}$$

Under completely randomized blocking, the probability that block c is comprised of the units in \mathbf{S}_{n_c} is $\binom{n}{n_c}$. Thus, the expectation of σ_{cs} is

$$\begin{aligned}
\mathbb{E}(\sigma_{cs}^2) &= \binom{n}{n_c}^{-1} \sum_{\mathbf{S}_{n_c}} \left(\frac{1}{n_c} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) - \frac{1}{n_c^2} \left(\sum_{i \in \mathbf{S}_{n_c}} y_{is} \mathbf{1}(i \in \mathbf{S}_{n_c}) \right)^2 \right) \\
&= \binom{n}{n_c}^{-1} \sum_{\mathbf{S}_{n_c}} \left(\frac{1}{n_c} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) - \frac{1}{n_c^2} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) \right) \\
&\quad - \binom{n}{n_c}^{-1} \sum_{\mathbf{S}_{n_c}} \frac{1}{n_c^2} \sum_{i \neq j \in \mathbf{S}_{n_c}} y_{is} y_{js} \mathbf{1}(i \in \mathbf{S}_{n_c}) \mathbf{1}(j \in \mathbf{S}_{n_c}) \\
&= \binom{n}{n_c}^{-1} \frac{n_c - 1}{n_c^2} \sum_{\mathbf{S}_{n_c}} \sum_{i \in \mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) \\
&\quad - \binom{n}{n_c}^{-1} \frac{1}{n_c^2} \sum_{\mathbf{S}_{n_c}} \sum_{i \neq j \in \mathbf{S}_{n_c}} y_{is} y_{js} \mathbf{1}(i \in \mathbf{S}_{n_c}) \mathbf{1}(j \in \mathbf{S}_{n_c}) \\
&= \binom{n}{n_c}^{-1} \frac{n_c - 1}{n_c^2} \sum_{i=1}^n \sum_{\mathbf{S}_{n_c}} y_{is}^2 \mathbf{1}(i \in \mathbf{S}_{n_c}) \\
&\quad - \binom{n}{n_c}^{-1} \frac{1}{n_c^2} \sum_{i=1}^n \sum_{j \neq i} \sum_{\mathbf{S}_{n_c}} y_{is} y_{js} \mathbf{1}(i \in \mathbf{S}_{n_c}) \mathbf{1}(j \in \mathbf{S}_{n_c}) \\
&= \binom{n}{n_c}^{-1} \frac{n_c - 1}{n_c^2} \sum_{i=1}^n y_{is}^2 \binom{n-1}{n_c-1} - \binom{n}{n_c}^{-1} \frac{1}{n_c^2} \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \binom{n-2}{n_c-2}
\end{aligned} \tag{A.34}$$

Now since

$$\frac{\binom{n-1}{n_c-1}}{\binom{n}{n_c}} = \frac{\frac{(n-1)!}{(n_c-1)!(n-n_c)!}}{\frac{n!}{n_c!(n-n_c)!}} = \frac{n_c}{n} \tag{A.35}$$

$$\frac{\binom{n-2}{n_c-2}}{\binom{n}{n_c}} = \frac{\frac{(n-2)!}{(n_c-2)!(n-n_c)!}}{\frac{n!}{n_c!(n-n_c)!}} = \frac{n_c(n_c-1)}{n(n-1)} \tag{A.36}$$

It follows that

$$\begin{aligned}
\sigma_{cs}^2 &= \binom{n}{n_c}^{-1} \frac{n_c - 1}{n_c^2} \sum_{i=1}^n y_{is}^2 \binom{n-1}{n_c-1} - \binom{n}{n_c}^{-1} \frac{1}{n_c^2} \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \binom{n-2}{n_c-2} \\
&= \frac{n_c n_c - 1}{n} \frac{1}{n_c^2} \sum_{i=1}^n y_{is}^2 - \frac{n_c(n_c - 1)}{n(n-1)} \frac{1}{n_c^2} \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \\
&= \frac{n_c - 1}{n(n_c)} \sum_{i=1}^n y_{is}^2 - \frac{n_c - 1}{n(n-1)n_c} \sum_{i=1}^n \sum_{j \neq i} y_{is} y_{js} \\
&= \frac{n_c - 1}{n(n_c)} \sum_{i=1}^n y_{is}^2 - \frac{n(n_c - 1)}{(n-1)n_c} \sum_{i=1}^n \sum_{j \neq i} \frac{y_{is} y_{js}}{n^2} \\
&= \frac{n_c - 1}{n(n_c)} \sum_{i=1}^n y_{is}^2 - \frac{n(n_c - 1)}{(n-1)n_c} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 + \frac{n_c - 1}{n(n-1)n_c} \sum_{i=1}^n y_{is}^2 \\
&= \frac{n_c - 1}{n(n_c)} \sum_{i=1}^n y_{is}^2 - \frac{n(n_c - 1)}{(n-1)n_c} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 + \frac{n_c - 1}{n(n-1)n_c} \sum_{i=1}^n y_{is}^2 \\
&= \frac{(n-1)(n_c - 1)}{n(n-1)n_c} \sum_{i=1}^n y_{is}^2 - \frac{n(n_c - 1)}{(n-1)n_c} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 + \frac{n_c - 1}{n(n-1)n_c} \sum_{i=1}^n y_{is}^2 \\
&= \frac{n(n_c - 1)}{n(n-1)n_c} \sum_{i=1}^n y_{is}^2 - \frac{n(n_c - 1)}{(n-1)n_c} \left(\sum_{i=1}^n \frac{y_{is}}{n} \right)^2 \\
&= \frac{n(n_c - 1)}{(n-1)n_c} \sum_{i=1}^n \frac{y_{is}^2}{n} - \frac{n_c(n_c - 1)}{n(n-1)} \left(\sum_{i=1}^n y_{is} \right)^2 = \frac{n(n_c - 1)}{(n-1)n_c} \sigma_s^2 \tag{A.37}
\end{aligned}$$

Likewise, substituting $y_{is} + y_{it}$ in for y_{is} , we obtain:

$$\sigma_{c(s+t)}^2 = \frac{n(n_c - 1)}{(n-1)n_c} \sigma_{s+t}^2 \tag{A.38}$$

Therefore

$$\begin{aligned}
&\mathbb{E} \left(\frac{n_c^2}{(n_c - 1)n^2} [(r-2)(\sigma_{cs}^2 + \sigma_{ct}^2) + \sigma_{c(s+t)}^2] \right) \\
&= \frac{n_c^2}{(n_c - 1)n^2} [(r-2)(\mathbb{E}(\sigma_{cs}^2) + \mathbb{E}(\sigma_{ct}^2)) + \mathbb{E}(\sigma_{c(s+t)}^2)] \\
&= \frac{n_c^2}{(n_c - 1)n^2} \left[(r-2) \left(\frac{n(n_c - 1)}{(n-1)n_c} \sigma_s^2 + \frac{n(n_c - 1)}{(n-1)n_c} \sigma_t^2 \right) + \frac{n(n_c - 1)}{(n-1)n_c} \sigma_{s+t}^2 \right] \\
&= \frac{n_c^2}{(n_c - 1)n^2} \frac{n(n_c - 1)}{(n-1)n_c} [(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2] \\
&= \frac{n_c}{n(n-1)} [(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2] \tag{A.39}
\end{aligned}$$

Finally, it follows that the expected difference in variances is:

$$\begin{aligned}
& \mathbb{E} \left[\sum_{c=1}^b \frac{n_c^2}{(n-1) \sum n_c^2} [(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2] \right] \\
& - \mathbb{E} \left[\sum_{c=1}^b \frac{n_c^2}{(n_c-1)n^2} [(r-2)(\sigma_{cs}^2 + \sigma_{ct}^2) + \sigma_{c(s+t)}^2] \right] \\
& = \sum_{c=1}^b \frac{n_c^2}{(n-1) \sum n_c^2} [(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2] \\
& \quad - \sum_{c=1}^b \frac{n_c}{n(n-1)} [(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2] \\
& = \frac{(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2}{n-1} \left(\frac{\sum n_c^2}{\sum n_c^2} - \frac{\sum n_c}{n} \right) \\
& = \frac{(r-2)(\sigma_s^2 + \sigma_t^2) + \sigma_{s+t}^2}{n-1} (0) = 0
\end{aligned} \tag{A.40}$$

That is, in expectation, the variance of estimates of the SATE under block randomization with random blocks be the same as those under complete randomization.

Appendix B

Appendix for Chapter 3

B.1 Proof of [3.28]

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 [3.36] 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

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

If $\tilde{x}_{kc} = 1$, then $u_{kc} > 0$, and by [3.35], $\omega_{kc} > t$. 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,

$$\tilde{x} \in \tilde{\mathbf{X}}. \quad (\text{B.2})$$

By definition of $K_c(x)$,

$$u \cdot \tilde{x} = \sum_{c=1}^C \sum_{k=1}^{N_c} u_{kc} \tilde{x}_{kc} \geq \sum_{c=1}^C \sum_{k=1}^{N_c} u_{kc} x_{kc} = u \cdot x. \quad (\text{B.3})$$

Since $K_c(x) \leq \#_c x$ and $q_{kc} \geq 0$, it follows from [3.46] and the rearrangement theorem (Hardy

et al. 1952) that

$$\begin{aligned}
q \cdot \tilde{x} &= \sum_{c=1}^C \sum_{k=1}^{N_c} q_{kc} \mathbf{1}(k \leq K_c(x)) \leq \sum_{c=1}^C \sum_{k=1}^{N_c} q_{kc} \mathbf{1}(k \leq \#_c x) \\
&\leq \sum_{c=1}^C \sum_{k=1}^{N_c} q_{kc} x_{kc} = q \cdot x.
\end{aligned} \tag{B.4}$$

By [B.2], [B.3], and [B.4], 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 [3.28] follows immediately.

B.2 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

$$x_{\pi(j)}^{m0} = \begin{cases} y_j^{\bullet m}, & j = 1, \dots, m \\ 0, & \text{otherwise.} \end{cases}$$

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 $x_{\pi(j)}^{m0}$, $j = 1, \dots, m$, 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 $x_{\pi(j)} = y_j^{\bullet m}$ 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 $y_j^{\bullet m} = 0$, $z_{c(\pi(j))}$ 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 $z_{c(\pi(m+1))} = 0$, 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 set (3). For any vector $a \in \mathbb{R}^N$, and for any $m \in \{1, \dots, N\}$, 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 $x_{\pi(j)} = y_j^{\bullet m}$, $j = 1, \dots, m$, 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} :

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

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 $B = N + 1$, $\lambda_{\text{LKP}}^y = \infty$. When $B \leq N$, the explicit solution for λ_{LKP}^y (Dantzig 1957) is

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

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 $x_{\pi(j)} = y_j^{\bullet m}$, $j = 1, \dots, m$ satisfies $u \cdot x \geq M$, then $\lambda_{\text{LKP}}^y = \infty$ 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 λ^{\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 λ .

Return.

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

Set $x_{\pi(m+1)} = 1$.

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

Set $x_{\pi(m+1)}$ to 0 and $z_{c(\pi(m+1))}$ to 0.

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

Return.

B.3 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:

$$T_w \equiv \max_{(k,c) \in \mathbf{J}_{\bar{n}}} w_{kc}(e_{kc}^H).$$

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

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

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 [3.9], order the batches so that if $k > k'$ then

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

Define

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

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

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

and

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

Then the proofs go through *mutatis mutandis*.