The Pennsylvania State University

The Graduate School

The Eberly College of Science

CALCULATING CELL BOUNDS IN CONTINGENCY TABLES BASED ON CONDITIONAL FREQUENCIES

A Thesis in

Statistics and Operations Research

by

Byran Jay Smucker

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

December 2007

The thesis of Byran Jay Smucker was reviewed and approved* by the following:

Aleksandra B. Slavković
Assistant Professor of Statistics
Thesis Advisor

Susan H. Xu
Professor of Management Science and Supply Chain Management

Runzi Li
Associate Professor of Statistics
Chair of Graduate Studies, Department of Statistics

*Signatures are on file in the Graduate School.

Abstract

Suppose we have data in the form of a contingency table. Certain information, such as marginal totals or conditional probabilities, along with sample size, can allow us to deduce bounds on each contingency table cell. If these intervals are too narrow - especially for cells with low counts - this could pose a privacy risk. In this thesis, we deduce and prove closed-form solutions for the linear relaxation bounds given full and partial conditional probabilities. We also calculate these bounds via linear programming and calculate sharp integer bounds via integer programming. We show that there can be large differences in the width of these bounds, suggesting that using the linear relaxation is likely not an acceptable shortcut to estimating the sharp bounds. We collect the run-times for these different examples, and compare several available optimization software packages. For the largest example, we compute the sharp integer bounds but find that this is prohibitively time-consuming for most practical usages. We also calculate bounds for a 2×2 table given any of three odds ratios and sample size, via quadratic constrained programming and mixed integer quadratic constrained programming and show that in this example the bounds are wider than those induced by the conditional probabilities and sample size.

Table of Contents

LIST OF TABLES **viii**

1 Introduction **1**

1.1 Statistics, Bounds and Optimization 3

1.1.1 Minimum Trace Factor Analysis 3

1.1.2 Controlled Rounding 3

1.1.3 Cell Suppression 4

1.1.4 Causal Inference 4

1.2 Cell Bound Calculation for Contingency Tables 5

1.3 Contributions of this Thesis 7

1.4 Computing 7

1.5 Notes on Examples Used 8

1.6 Optimization Methods 8

1.6.1 Linear Programming 9

1.6.2 Mixed Integer Programming 12

1.6.3 Quadratic Constrained Programming and Mixed Integer Quadratic Constrained Programming 14

2 Calculating Cell Bounds Given Conditional Probabilities: 2-way Tables 16

2.1 Setting and Notation 16

2.2 Cell Bounds for a 2×2 Table, Given Conditional Probabilities and Sample Size 18

2.2.1 Formulation of Optimization Problems for a 2×2 Example 18

2.2.2 Exact Formulas for Linear Relaxation Bounds Given Conditional Probabilities 22

2.2.3 Results for 2×2 Table 23

2.3 Cell Bounds for a 4×4 Table, Given Conditional Probabilities and Sample Size 24

2.3.1 Formulation of Optimization Problems for 4×4 Example 25

2.3.2 Results for 4×4 Example 29

2.4 Summary 29

3 Calculating Cell Bounds Given Conditional Probabilities: Multi-way Tables 30

3.1 Setting and Notation 30

3.2 Closed-Form Bounds for k-way Tables Given Conditional Probabilities . . . 32

3.2.1 Closed-Form Bounds for k-way Tables Given Full Conditionals . . . 32

3.2.2 Closed-Form Bounds for k-way Tables Given Partial Conditionals . . 33

3.3 Cell Bounds for a $3 \times 3 \times 3$ Table, Given Conditional Probabilities and Sample Size 35

3.3.1 Formulation of Optimization Problems for 3-way Example 36

3.3.2 Results for 3-way Example 38

3.4 Cell Bounds for a 4-way Example, Given Conditional Probabilities and Sample Size 38

3.4.1 Formulation of Optimization Problems: 4-way Example 39

3.4.2 Results for 4-way Example 40

3.4.3 4-way Example Given Partial Conditionals: $P(Trt|Center, Status)$. 41

3.4.4 4-way Example Given Partial Conditionals: $P(Center, Status|Trt)$. 44

3.4.5 4-way Example Given Partial Conditionals: $P(Response|Trt)$ 45

3.4.6 4-way Example Given Partial Conditionals: $P(Response|Center, Status)$ 46

3.5 Cell Bounds for a 6-way Table Given Conditional Probabilities and Sample Size 48

3.5.1 Formulation of Optimization Problems for 6-way Example 49

3.5.2 Results for 6-way Example 49

3.6 Cell Bounds for 8-way Table Given Conditional Probabilities and Sample Size 49

3.6.1 Formulation of Optimization Problems for 8-way Example 50

3.6.2 Results for 8-way Example 51

4 Discussion of Results **58**

4.1 Integer Programs 58

4.2 Linear Relaxations 59

4.3 Discussion of Method Performance (Including Genetic Algorithms) 60

5 Calculation of Cell Bounds Given Odds Ratios **62**

5.1 Odds Ratios 62

5.2 Calculating Bounds Given Odds Ratios and Sample Size 63

5.3 Analytical Bounds for Some Cells in Linear Relaxation 63

5.4 Results for MIQCP's and QCP's Given Odds Ratios and Sample Size . . . 65

5.5 Discussion of Cell Bound Results Given Odds Ratio Information 65

6 Conclusions **66**

6.1 Future Work 67

A IP/LP Formulations to Larger Multi-way Tables **68**

B Formulation of MIQCP's for α_1 and α_2 **69**

List of Tables

2.1 Basic 2-way Table 18

2.2 IP and LP Results for 2-way Table 23

2.3 2×2 Example: Comparison of Run-times for 3 Methods 24

2.4 4×4 Table. Delinquent Children Data 24

2.5 IP and LP Results for 4×4 Table 29

2.6 4×4 Example: Comparison of Run-times for 3 Methods 29

3.1 Data Describing Attitude of White Christians Toward Abortion in 1972 . . 35

3.2 IP and Linear Relaxation Results for Abortion Data 39

3.3 3-way Example: Comparison of Run-times for 3 Methods 39

3.4 Clinical Trial Data 40

3.5 Full Conditionals for Clinical Trial Data 40

3.6 IP and LP Relaxation bounds for Clinical Trial Data given Full Conditionals and Sample Size 41

3.7 4-way Example: Comparison of Run-times for 3 Methods 41

3.8 $Treatment|Center, Status$ Conditionals and Counts for Clinical Trial Data 42

3.9 IP and LP bounds for Clinical Trial Data, given $T|CS$ Conditional Probabilities and Original Data 43

3.10 $Center, Status|Treatment$ Conditionals and Counts for Clinical Trial Data 44

3.11 IP and LP bounds for Clinical Trial Data, given $CS|T$ Conditional Probabilities and Original Data 45

3.12 $Response|Treatment$ Conditionals and Counts for Clinical Trial Data . . . 45

3.13 IP and LP bounds for Clinical Trial Data, given $R|T$ Conditional Probabilities and Original Data 46

3.14 $Response|Center, Status$ Conditionals for Clinical Trial Data 47

3.15 IP and LP bounds for Clinical Trial Data, given $R|CS$ Conditional Probabilities and Original Data 48

3.16 Czech Autoworkers Data 53

3.17 Full Conditional Probabilities for Czech Autoworkers Data 54

3.18 IP Results for Czech Autoworkers Data, Given Full Conditionals and Sample Size, Using Original Data 55

3.19 6-way Example: Comparison of Run-times for 3 Methods 55

3.20 CPS Variables and Number of Levels 56

3.21 Select Data for CPS Example 56

3.22 Select IP/LP Results for CPS Data, Given Full Conditionals and Sample Size 57

3.23 8-way Example: Comparison of Run-times for 3 Methods 57

5.1 MIQCP and QCP Bounds for 2way Table Given α and Sample Size 65

5.2 MIQCP and QCP Bounds for 2-way Table Given α_1 and Sample Size . . . 65

5.3 MIQCP and QCP Bounds for 2-way Table Given α_2 and Sample Size . . . 65

Chapter 1

Introduction

A convenient and common way to display categorical data are via contingency tables. Social or government agencies often collect such data with intent to release it as public information so that it can be used for inference, the results of which could be used to affect policy decisions or further research. However, if too much information is released, privacy of individuals could be compromised, privacy that has likely been guaranteed upon collection of the data. Thus, there must be a trade-off between releasing the data and maintaining privacy.

There are many ways in which data privacy can be violated, as well as many ways to determine whether a violation has occurred. One way is the concern of this paper, and involves what has been called the *feasibility interval* [48], that is, the bounds on a cell that can be induced by given information. If these feasibility intervals are too narrow - or if the table is uniquely identified because the lower and upper bounds are the same - the risk of a disclosure could be high, particularly in cells with small counts.

We are particularly interested in the cell bounds that can be calculated when we are given conditional probabilities. This is an important question because although much categorical data is expressed in terms of contingency tables, agencies such as the Bureau of Labor

Statistics sometimes release rates or percentages representing proportions of individuals who fall in a certain category given some other characteristics ([42], p. 7). What information can be extracted from these conditional probabilities? This is the question which we explore in this thesis.

Another quantity which has an impact in statistical analysis involving categorical data is the odds ratio. This quantity comes up in areas such as logistic regression and log-linear models. Since this is a parameter of interest, there may be instances in which we are given odds ratios to summarize or help summarize a contingency table. Thus, it is of interest to determine the information we can extract from this quantity, and we do so in chapter 5.

In the first chapter of this thesis we do a literature review of some connections between Operations Research and Statistics, as well as cell bounds in statistical disclosure limitation, and outline the contributions of this work. We then discuss the optimization methods utilized in this work. In the second chapter we discuss 2-way tables, using two examples to illustrate formulation of the optimization problems which use conditional probability information to lead to the calculation of cell bounds. We also give two simple results for the calculation of the lower and upper bounds based on the linear relaxation. In the third chapter, we extend these results to k-way tables given both full and partial conditional probabilities, and explore four more datasets which are represented as k-way tables. In chapter 4 we give a short, general discussion of our results in chapters 2 and 3, as well as a comparison of the optimization methods used to calculates the bounds. In chapter 5 we use three different odds ratios to formulate an optimization problem which result in bounds on a 2-way contingency table. We conclude our discussion in chapter 6.

1.1 Statistics, Bounds and Optimization

1.1.1 Minimum Trace Factor Analysis

Operations Research (O.R.) and in particular optimization has long played a role in several areas in statistics. One prominent area has been the role of nonlinear and semidefinite programming in Minimum Trace Factor Analysis. This area attempts to find a diagonal matrix, D, which, when subtracted from a symmetric, positive-semidefinite matrix S will minimize the trace of $S - D$, subject to the constraint that $S - D$ must be positive semidefinite.

There are many variations of this problem, and it has been studied extensively (see [4], [27], [41], [47], and [46]). Specifically, it finds applications in determining the greatest lower bound for the reliability of a test. Reliability shows how consistently a variable is measured when a test or survey is administered, and when certain strong assumptions cannot be legitimately made it cannot be explicitly measured, though lower bounds on it can be developed, using optimization methods.

1.1.2 Controlled Rounding

The controlled rounding problem is the problem of preserving the additive row and column totals in a two-way array while optimally rounding real-valued entries to adjacent integers. Cox et al. [16] solved the controlled rounding problem using a transportation problem formulation, a linear programming formulation whose structure allows it to be solved extremely efficiently (see [3]). The transportation problem can be described as follows: Say you have n manufacturing facilities, $a_1, ..., a_n$, each of which produce r_i products. There are also m stores, $b_1, ..., b_n$, each of which requires c_i of the products. Between each manufacturing facility and store there is a cost d_{ij}. How should the products be transported from the manufacturing facilities to the stores to minimize cost?

Subsequently, Causey et al. [12] formulated several statistical problems as controlled round-

ing problems (the Controlled Selection problem; raking), including a direct application of this method to statistical disclosure limitation (SDL): given a 2-way array with small counts which could lead to a disclosure, apply controlled rounding to some integer base (say 5 or 10), which would result in an array with less disclosure risk.

Later, Cox and George [18] extended this idea to tables with subtotals using a capacitated transshipment formulation, where the transshipment problem is like the transportation problem except there are intermediate "distribution centers" which have no demand but can take products and route them to stores (again, see [3]).

1.1.3 Cell Suppression

Cell suppression of tabular data is a disclosure limitation method which suppresses cells (disclosure cells) in a table which would result in a disclosure, and subsequently suppresses other cells (called complementary cells) which could allow one to determine the disclosure cells. The problem is to select all complementary cells necessary to prevent a disclosure, but leave as many cells as possible to release as much information as possible.

To solve this problem optimally is difficult, though linear programming has been used by the U.S. Census Bureau ([51]). Cox [17] has also studied this problem, and used network theory to solve it optimally under the "minimum-number-of-complementary-suppressions" criteria.

1.1.4 Causal Inference

One area which has utilized bounding, as well as optimization to calculate these bounds, is causal inference. Manski [36] used the bounding idea in the context of the selection problem, which attempts to estimate a regression relationship in the presence of incomplete data. He does not use optimization, but instead computes bounds on the conditional expectation using simple facts about conditional probability and a property of expectations. He expands

upon this, applying his method to calculating bounds on treatment effects, tightening them for more specific situations [37].

Balke and Pearl [1] develop bounds on counterfactual probabilities by developing a deterministic function that relates the variable corresponding to the counterfactual consequent with that corresponding to the counterfactual antecedent. Of course, this function is only deterministic if we know values for all unknown variables affecting the consequent that might also affect the antecedent. Using this idea, they develop an expression for the counterfactual probabilities, along with some expressions denoting constraints. They then use linear or nonlinear programming to determine the bounds. This work utilizes optimization techniques in the same way that we do, in that we maximize and minimize using information that we have to get upper and lower bounds on probabilities (or in our case, cell counts). The setting in which we develop these optimization problems differs (causal inference versus contingency tables), however, and further, counterfactual probabilities are inherently unobservable, while the probabilities in our settings have been observed but are unknown to us. In a later paper [2], they apply this methodology to the treatment effect problem that Manski tackled before. Their bounds are tighter and more general.

1.2 Cell Bound Calculation for Contingency Tables

Calculating cell bounds for the entries of contingency tables given marginal totals has a long history, and goes back to Bonferroni [8], Frechet [28], and Hoeffding [32] in their work on bounds for cumulative distribution functions given univariate marginals (Fienberg, [23], [24]). Given an $I \times J$ table with total sample size (n_{++}) and marginal totals (n_{i+} and n_{+j}), these bounds, called *Fréchet bounds*, have the following form for the ij^{th} cell:

$$min\{n_{i+}, n_{+j}\} \geq n_{ij} \geq max\{0, n_{i+} + n_{+j} - n_{++}\}$$

Work has been done on generalizations of these bounds, i.e. bounds for k-way contingency tables. Given marginal totals for k-way tables, Dobra and Fienberg [20] developed theory

and explicit formulas for the bounds when the table can be represented as a decomposable graph, a construct in which the expected counts in the cells of the table can be written as functions of the marginals. They extended these results to the case in which the graph is reducible, though when the table cannot be represented as a graph (which is often the case), other methods such as linear programming must be employed.

They further extended this idea to general k-way tables [21] by generalizing the "shuttle algorithm" originally developed by Buzzigoli and Giustic [10] for 3-way tables. This algorithm exploits hierarchical relationships within the table, and sequentially updates the bounds for cells until they cannot be further improved. This algorithm gives sharp bounds for any k-way contingency table given marginal totals, and for a large example (a table with 2^{16} entries), they completed the calculations in less than one hour.

Significantly less work has been done examining bounds induced by given conditional probabilities. Work done by Slavkovic [42], Slavkovic and Feinberg [43], and Feinberg and Slavkovic [26] has begun to examine the cell bounds induced by conditional probabilities in conjunction with given marginals, as well as conditionals alone, using both mathematical programming (linear and integer) and Markov bases. In the next section, we discuss this work further, as well as our contribution.

As discussed above, when marginals and/or conditionals are given, a natural way to obtain bounds is via linear programming (1.6.1) or integer programming (1.6.2). Integer programming (IP) requires an integer solution and thus calculates the sharpest bounds possible based on given marginal and/or conditional information. The linear relaxation of the integer program (LP) does not pose the integer constraint. As might seem obvious, IP is much more computationally intensive than LP, so many may want to use LP as an approximation to the tight IP bounds. The maximal gap between an IP and its linear relaxation has been studied and applied to the statistical disclosure setting [33] and theoretically has been shown by Sullivant [44] to be exponentially large, showing that it is dangerous in the case of given marginals to use the linear relaxation as an approximation to the sharp integer bounds.

1.3 Contributions of this Thesis

In this thesis, we calculate cell bounds for a variety of examples given conditional probability information and sample size, using an integer/linear programming formulation. We improve upon the formulation proposed by Slavkovic [42] by requiring the marginals upon which we condition to be nonzero while allowing individual cells to be zero. Restricting the counts in individual cells to be greater than zero (as Slavkovic did) is unnecessarily restrictive, because while a zero marginal would result in a division by zero when calculating a conditional probability, there need not be any such restriction upon individual cells.

Further, for a 2×2 example we calculate sharp cell bounds given a particular odds ratio, as well as the linear relaxation of the problem. This is accomplished using mixed integer quadratic constrained programming (MIQCP) and quadratic constrained programming (QCP).

In terms of statistical disclosure limitation, computed sharp cell bounds given conditional information could trigger the realization that a disclosure is likely, in the case that the feasibility interval is narrow. Additionally, one might think that the linear relaxation to the integer program might be a good approximation to the sharp bound. We show empirically that there can be large gaps between the two, though while the linear relaxation bounds are fairly easy to calculate, the sharp bounds may be much more difficult.

1.4 Computing

We will use a variety of optimization software programs to calculate these bounds, and will evaluate the run-time effectiveness of each in comparison to one another. The approaches we will use include: a standard commercial solver, Cplex [53]; MATLAB's `linprog` from its Optimization Toolbox [56]; and an implementation of the freeware `lp_solve` system in R ([55] and [57]). We also experimented with MATLAB's Genetic Algorithm capability.

To solve the MIQCP's and QCP's, we used BARON (Branch and Reduce Optimization Navigator) ([52] and [45]). In the case of both Cplex and BARON, the solvers were accessed through the optimization modeling language GAMS (General Algebraic Modeling Language) ([54]).

Unless otherwise specified, all bounds have been calculated on a system which includes Quad 2.6 GHz AMD Opteron Processors and 32 GB of ECC RAM[1].

1.5 Notes on Examples Used

We will solve a variety of problems of varying sizes: a 2×2 table (called the Download Data), a 4×4 table (Delinquent Children Data), a three-way table (Abortion Data), a four-way table (Clinical Trial Data), a six-way table (Czech Autoworkers Data), and an 8-way table (CPS Data).

These datasets will induce optimization problems with anywhere from 4 to 2,880 decision variables (note that these are not random variables), and a similar range on the number of constraints. Though each dataset is different, the formulation of the mathematical program based on given full conditionals will be similar. There will be (1) a constraint requiring that the sum of each cell's count is equal to the sample size, (2) a set of constraints which are derived from the conditional probability information at hand, (3) a set of constraints requiring that each margin be at least 1, and (4) the lower bound on each variable will be set at zero.

1.6 Optimization Methods

In this paper, we will be solving mathematical programs to calculate cell bounds for contingency tables of varying sizes given certain information. These mathematical programs

[1] http://gears.aset.psu.edu/hpc/systems/hammer/

will be one of four types: linear programs (LP), mixed integer programs (IP), quadratically constrained programs (QCP), or mixed integer quadratically constrained programs (MIQCP).

1.6.1 Linear Programming

A linear program consists of a linear objective function, optimized subject to linear constraints. It can be represented in standard form as:

$$\begin{aligned} &\textit{Minimize} && \mathbf{cx} \\ &\textit{subject to} && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{1.1}$$

where there are n variables and m constraints, $\mathbf{c}$ is a row vector of length n, $\mathbf{x}$ is a column vector of length n, $\mathbf{A}$ is a $m \times n$ matrix, and $\mathbf{b}$ is a column vector of length m.

Constraints can be either equality or inequality constraints, but inequality constraints can be easily reformulated as equality constraints by introducing additional variables ([3], p.4).

In the context of this paper, we use linear programming to calculate upper and lower bounds on contingency tables without restricting those bounds to be integer. Thus, these bounds are called linear relaxation bounds.

In practice, there are two classes of methods that are generally used to solve linear programs: some variation of the Simplex Method and Interior Point methods.

The Simplex Method

The classic Simplex Method [19] is the seminal algorithm that launched linear programming into viability, enabling linear programs to be solved that were previously too difficult.

This exploits a property of linear optimization problems constrained by convex sets: any optimal solution must lie at an extreme point, where extreme points are points within the convex set that cannot be represented as convex combinations of two other points which lie in the set. If the constraint set is composed of linear constraints, extreme points can be thought of as "corner points." Thus, the search space can be made dramatically smaller and the linear program can be solved efficiently.

It has been shown that in a worst-case scenario, the Simplex Algorithm has exponential-time behavior. That is, the computational effort needed to solve a linear program via the Simplex Method cannot be bounded by a polynomial function [3]. However, in practice, the Simplex Method is still a competitive algorithm, and is widely employed by software today.

Interior Point Methods

Interior point methods were first introduced by Karmarkar in 1984 [34]. Karmarkar's algorithm was the third major theoretical advance in solving linear programs, following the Simplex Method [19] and the Ellipsoid Algorithm (see [6]), though the latter, while promising theoretically, was ineffective in practice ([3], p. 392). Karmarkar's algorithm, both because it was polynomial-time and because it was purported to be practically successful in solving linear programs ([49],1) set off a firestorm of research activity capitalizing on the interior point idea.

As the name implies, Interior Point Methods are algorithms which search through the interior of the feasible region of the linear program. Mehrotra's algorithm ([38], also see [49]) is a predictor-corrector primal-dual interior point method, and is a classic interior point algorithm. The following is a sketch of the idea of the algorithm.

Given a set of linear constraints, a curve called the Central Path can be defined within the feasible region which converges to the optimal solution. Primal-dual interior point methods use this Central Path in some way to guide them to the solution. In particular, Methotra's

algorithm utilizes a predictor-corrector construct, the search direction at each iteration consisting of 1) a predictor direction, i.e. a direction that most directly moves toward the optimal solution; 2) a centering parameter chosen at each iteration, which determines how closely the predictor direction is followed; and 3) a corrector direction, which keeps the search close to a trajectory which will lead to the optimal solution. ([49],pp.7-16, 94-95).

Many variations of this algorithm, along with other methods based on searching the interior of the feasible region have been proposed. While they have not completely supplanted the Simplex Method, it is safe to say that they represent the cutting edge in the solution of linear programs.

LP Methods We Utilize

Cplex Cplex [53] is a commercial optimization software package which has a variety of algorithms which will solve Linear Programs. However, several of these, including the Network Optimizer and Sifting Optimizer are for specific types of problems that do not concern this research.

For small problems, and problems that are not sparse, Cplex can use a dual or primal simplex algorithm [3], while for very large, sparse problems the most effective method is a primal-dual logarithmic barrier algorithm ([49], 37-40), an interior point method which uses this barrier function to define the Central Path (see Section 1.6.1).

Since Cplex is a commercial package, it performs very well compared to other packages. It can solve problems faster and can solve larger problems that other programs cannot. The downside, of course, is that to utilize this software there is a financial investment involved.

MATLAB's Optimization Toolbox This commercial software package can use a function called `linprog`[2]to solve linear programs. For smaller problems, it uses a variant of the

classic Simplex Method [19], an active set method, to solve smaller linear programs [5].

To solve larger optimization problems, `linprog` utilizes an interior point method [50] which is adapted from Mehrotra's predictor-corrector algorithm ([38] and Section 1.6.1).

Again, this is a commercial package, and perhaps more readily available than Cplex. Additionally, since it is part of a MATLAB toolbox, it may be easier to use since MATLAB's programming language is fairly common. However, it does not seem to perform as well as Cplex.

R implementation of lp_solve `lp_solve` [55] is freeware developed by Michel Berkelaar and its R implementation was authored by Sam Buttrey [57]. It utilizes a simplex method algorithm to solve linear programs.

For statisticians, this is the most readily accessible of the LP software we used, and is suitable for small problems. However, for larger problems, model specification becomes cumbersome and it failed to solve some of the larger LP problems we ran. `lp_solve` can be used in different software platforms, but the implementation in R seems somewhat unstable.

1.6.2 Mixed Integer Programming

Mixed integer programs can be formulated in the same way as linear programs (1.1), with the exception that some or all of the variables are required to be integer. In our case, we utilize pure integer programs in which we require all of our variables to be integer:

[2] http://www-ccs.ucsd.edu/matlab/toolbox/optim/linprog.html

$$\begin{aligned} &Minimize && \mathbf{cx} \\ &subject\ to && \mathbf{Ax} = \mathbf{b} \\ & && \mathbf{x} \in \mathbb{Z}^{+} \end{aligned} \tag{1.2}$$

where $\mathbb{Z}^{+}$ represents the nonnegative integers and the nonnegativity constraint in (1.1) is eliminated because it would be redundant.

Branch and Bound Methods

Branch and bound methods divide the feasible region of the IP into subregions. Then, they calculate upper and lower bounds for each of these subregions. When a subregion has a lower bound that is larger than the upper bound of another subregion, the subregion can be discarded, and this is called pruning. Subregions are sometimes solved, i.e. the upper and lower bounds calculated for them are the same. When all of the subregions are solved or pruned, the optimal solution is apparent. (See [40] for a technical discussion of this method in the IP context).

Branch and Cut Methods

Branch and cut methods (a combination of branch and bound and something called the cutting plane method, see [40]) solve Problem (1.2) without the integer constraints to develop a lower bound for the IP. If the linear relaxation does not produce an integral solution, the algorithm searches for a constraint with excludes the current solution, but does not exclude any of the integer solutions in the feasible region (this is the 'cut' part of the algorithm). This process repeats until either an integral solution is found or it proves too difficult or impossible to find a new constraint. If the latter is the case, then the branching part of the process commences and the problem is broken into subproblems, where the above process

is repeated.

IP Methods We Utilize

Cplex utilizes a branch and cut algorithm to solve integer programs, while `lp_solve` employs a branch and bound method. Unfortunately, MATLAB's Optimization Toolbox does not have IP capabilities.

1.6.3 Quadratic Constrained Programming and Mixed Integer Quadratic Constrained Programming

Quadratic Constrained Programming (QCP) is a class of optimization problems in which the constraints include quadratic terms. A QCP is a special case of a nonlinear program where both the objective function and constraints can be nonlinear functions. A Mixed Integer Quadratic Constrained Program (MIQCP) is a QCP with integer constraints on some of the variables. In this paper, we are concerned with solving small QCP's and MIQCP's which can be represented generally as:

$$\begin{aligned} &Minimize && \mathbf{cx} \\ &subject\ to && g(\mathbf{x}) \leq 0 \\ & && h(\mathbf{x}) = 0 \\ & && \mathbf{x} \in X \end{aligned} \tag{1.3}$$

where $g : X \to \mathbb{R}^a$ and $h : X \to \mathbb{R}^b$ are quadratic functions, a is the number of inequality constraints, and b is the number of equality constraints. If we are solving a QCP, $X \in \mathbb{R}^n$; if we are solving a MIQCP, $X \in \mathbb{Z}$.

Since we only solve very small QCP and MIQCP's, picking a robust, efficient piece of software for these problems was not critical. Because of its availability in the mathematical

modeling language we used (GAMS, see http://www.gams.com/), we utilized BARON (see [52] and for a technical discussion, [45]) to solve QCP's and MIQCP's.

Chapter 2

Calculating Cell Bounds Given Conditional Probabilities: 2-way Tables

In this chapter we consider $I \times J$ tables, first using a simple 2×2 example (Download Data) to demonstrate how to formulate the integer and linear programming problems which result in cell bounds, then using that formulation to prove a simple theorem and state an observation about the bounds of the linear relaxation. The chapter concludes with a 4×4 example (Delinquent Children Data).

2.1 Setting and Notation

Let X and Y be two random variables and $O = \{o_{ij}\}$ be the $I \times J$ table (matrix) of observed counts with sample size N. The joint probability distribution of these two random variables can be represented as $P = \{p_{ij}\}, i = 1, ..., I, j = 1, ..., J$, where $p_{ij} = P(X = i, Y = j)$.

Further, the marginal probability distributions for X and Y are

$$p_{i\cdot} = \sum_{j=1}^{J} p_{ij} = P(X = i)$$

and

$$p_{\cdot j} = \sum_{i=1}^{I} p_{ij} = P(Y = j)$$

respectively, and conditional probability distributions are $C = \{c_{ij}\}$ and $D = \{d_{ij}\}$ where

$$c_{ij} = \frac{p_{ij}}{p_{\cdot j}} = P(X = i|Y = j) \tag{2.1}$$

and

$$d_{ij} = \frac{p_{ij}}{p_{i\cdot}} = P(Y = j|X = i) \tag{2.2}$$

for $i = 1, ..., I, j = 1, ..., J$.

Note that these probability distributions involve true parameters, and under the assumption of multinomial sampling the observed counts are just estimators of those parameters. We are in particular interested in the estimated (observed) conditional probabilities and will represent them as $\hat{C} = \{\hat{c}_{ij}\}$ and $\hat{D} = \{\hat{d}_{ij}\}$ with $\hat{c}_{ij} = \frac{o_{ij}}{o_{\cdot j}}$ and $\hat{d}_{ij} = \frac{o_{ij}}{o_{i\cdot}}$.

An important note: As we have stated, the observed counts for the ij^{th} cell are represented by o_{ij} while in our integer and linear programs, the decision variables (those variables which can be varied subject to constraints) used to define cell bounds are represented by n_{ij}. One can think of the observed counts as fixed (as they are a realization from the joint probability distribution P), while the n_{ij}'s can vary with relation to the optimization programs.

2.2 Cell Bounds for a 2×2 Table, Given Conditional Probabilities and Sample Size

In a fictitious example taken from Slavkovic ([42], p. 57), suppose we have a sample of 25 male students and 25 female students and we ask them whether they have ever illegally downloaded mp3's on the internet. Thus X=gender and Y=illegally downloaded? with $i = 1, 2$ (male, female), $j = 1, 2$ (yes, no). These data are summarized in Table 2.1.

	Download Yes	Download No	Total
Male	15	10	25
Female	5	20	25
Total	20	30	50

Table 2.1: Basic 2-way Table

From Table 2.1, using $\hat{d}_{ij} = \frac{o_{ij}}{o_{i1}+o_{i2}}$, we can calculate the following 2×2 matrix of row conditional probabilities:

$$\hat{D} = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}$$

2.2.1 Formulation of Optimization Problems for a 2×2 Example

In most statistical models which can help in the analysis of contingency table data, each population parameter is assumed to be greater than zero (i.e. no structural zeros). However, for a given sample we can certainly have a sampling zero. Because of this, instead of placing a lower bound of 1 on each cell, it seems reasonable to make the lower bound 0 and instead require that each margin have a count of at least one. This is necessary to satisfy the definition of conditional probability.

In cases in which we wish to condition upon margins which are zero, we must somehow deal with the fact that the resulting mathematical expression is undefined. People deal with this in a variety of ways, sometimes collapsing across categories in some of the independent

variables, for instance. In the 8-way example (Section 3.6) where we had this problem, we just define the conditional probability to be zero because collapsing some of our independent variables does not produce non-zero margins.

With this in mind, to calculate lower bounds on the ijth cell in the original table based on the row conditionals (matrix $\hat{D}$), the following optimization program can be constructed:

$$Min\ n_{ij} \tag{2.3}$$

$$s.t.\ n_{11} + n_{12} + n_{21} + n_{22} = 50 \tag{2.4}$$

$$\hat{d}_{11} = \frac{n_{11}}{n_{11} + n_{12}} \tag{2.5}$$

$$\hat{d}_{12} = \frac{n_{12}}{n_{11} + n_{12}} \tag{2.6}$$

$$\hat{d}_{21} = \frac{n_{21}}{n_{21} + n_{22}} \tag{2.7}$$

$$\hat{d}_{22} = \frac{n_{22}}{n_{21} + n_{22}} \tag{2.8}$$

$$n_{11} + n_{12} \geq 1 \tag{2.9}$$

$$n_{21} + n_{22} \geq 1 \tag{2.10}$$

$$n_{ij} \geq 0, \quad \forall i, j \tag{2.11}$$

$$n_{ij}\ integer\ \forall i, j \tag{2.12}$$

Remember that the n_{ij}'s are decision variables that can be varied subject to the specified constraints. In other words, these variables can take on any value such that the original conditional probabilities, as well as the other constraints in the model, are satisfied. The $\hat{d}_{ij}$'s are assumed known and calculated from the observed data O.

To calculate lower bounds for each cell, we must solve four optimization problems, each one having a different cell in the objective function. To calculate upper bounds on each cell, the objective function would be maximized instead of minimized.

We can linearize (2.5)-(2.8). For instance, (2.5) can be written as $-(1-\hat{d}_{11})n_{11}+\hat{d}_{11}n_{12} = 0$. This can be done for each of the four constraints, (2.5)-(2.8). However, note that if $\hat{d}_{11}$ is

known, then $\hat{d}_{12}$ is determined, and vice versa. Therefore, (2.6) and (2.8) can be eliminated as constraints altogether.

This results in the following integer program:

$$Min\ n_{ij} \tag{2.13}$$

$$s.t.\ n_{11} + n_{12} + n_{21} + n_{22} = 50 \tag{2.14}$$

$$-\hat{d}_{12}n_{11} + \hat{d}_{11}n_{12} = 0 \tag{2.15}$$

$$-\hat{d}_{22}n_{21} + \hat{d}_{21}n_{22} = 0 \tag{2.16}$$

$$n_{11} + n_{12} \geq 1 \tag{2.17}$$

$$n_{21} + n_{22} \geq 1 \tag{2.18}$$

$$n_{ij} \geq 0, \quad \forall i, j \tag{2.19}$$

$$n_{ij}\ integer\ \forall i, j \tag{2.20}$$

Removing the integral constraint results in the linear relaxation of the above IP:

$$Min\ n_{ij} \tag{2.21}$$

$$s.t.\ n_{11} + n_{12} + n_{21} + n_{22} = 50 \tag{2.22}$$

$$-\hat{d}_{12}n_{11} + \hat{d}_{11}n_{12} = 0 \tag{2.23}$$

$$-\hat{d}_{22}n_{21} + \hat{d}_{21}n_{22} = 0 \tag{2.24}$$

$$n_{11} + n_{12} \geq 1 \tag{2.25}$$

$$n_{21} + n_{22} \geq 1 \tag{2.26}$$

$$n_{ij} \geq 0, \quad \forall i, j \tag{2.27}$$

This optimization problem can be formulated equivalently but more succinctly in matrix

notation:

$$\begin{aligned} Min\ & \mathbf{cn} \\ s.t.\ & \mathbf{An} = \mathbf{b} \\ & \mathbf{Gn} \geq \mathbf{h} \\ & \mathbf{n} \geq \mathbf{0} \end{aligned} \tag{2.28}$$

where $\mathbf{c}$ is a row vector of length 4 consisting of 3 zeros, with a 1 in the ijth position, $\mathbf{n}$ is a column vector of length 4 consisting of the decision variables, $\mathbf{A}$ is a 3×4 matrix representing the left-hand side of the equality constraints, $\mathbf{b}$ is a column vector of length 3 representing the right hand side of the equality constraints, $\mathbf{G}$ is a 2×4 matrix representing the left-hand side of the inequality constraints, and $\mathbf{h}$ is a column vector of length 2 representing the right hand side of the inequality constraints.

Optimization problem (2.28) is a linear program, but if the additional constraints are added so that each variable must be integer-valued, it is an integer program.

For demonstration purposes, we will show the vectors and matrices here:

$$\mathbf{c} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{n} = \begin{bmatrix} n_{11} & n_{12} & n_{21} & n_{22} \end{bmatrix},$$

where the "1" in $\mathbf{c}$ corresponds to the particular cell being optimized.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -0.4 & 0.6 & 0 & 0 \\ 0 & 0 & -0.8 & 0.2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 50 \\ 0 \\ 0 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \mathbf{h} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

2.2.2 Exact Formulas for Linear Relaxation Bounds Given Conditional Probabilities

For the linear relaxation as we have formulated it, notice that the lower bounds for each cell are equal to the conditional probability for that cell. We will prove this result for the particular case of the 2×2 example, but it can easily be adapted to any contingency table - as we will show in Chapter 3 - since the LP associated with it has the same form.

Theorem 2.2.1 *Assume we have a 2-way contingency table. Based on the conditional probabilities $P(Y = j|X = i)$ and the sample size, we can construct a linear program of the form* (2.21)*. This linear program is minimized when $n_{ij} = \hat{d}_{ij}$. That is, the lower bound on the cell represented in the objective function is equal to its associated conditional probability.*

Proof We prove this result in the case of our 2-way example. Any other size of contingency table would have a linear program with the same structure, and could be proved similarly. Note the the lower bound for n_{ij} cannot be zero, because if it were, the other cell which defines its conditional distribution would be forced to zero by (2.23) or (2.24). This cannot happen because of constraints (2.25) and (2.26). Constraints (2.23) and (2.24) are derived from the the conditional probability relationship $\hat{d}_{ij} = \frac{n_{ij}}{\sum_j n_{ij}}$. Since (2.25) and (2.26) hold, and n_{ij} is minimized when its marginal is as small as possible, n_{ij} will be minimized when its marginal is 1, which forces n_{ij} to be precisely equal to its conditional probability, $\hat{d}_{ij}$. ∎

Additionally, a closed-form solution can be derived for the upper bounds. Recall that I is the number of categories in the first variable, J is the number of categories in the second, and N is the total sample size.

Theorem 2.2.2 *Assuming a 2-way contingency table and LP as in Theorem 2.2.1, and assuming none of the rows in the contingency table sum to zero, the linear program is maximized for the ij^{th} cell at $(N - (I - 1))\hat{d}_{ij}$.*

Proof Since we are maximizing n_{ij}, the marginal total for each of the rows (beside the i^{th} row) in the contingency table will be as small as possible, namely 1, as required by constraints (2.25) and (2.26). This is possible because each of the cells can have a value equal to their conditional probability, as shown in Theorem 2.2.1. Thus, for all but the i^{th} row, the marginal total is 1. So now there are $N-(I-1)$ counts to distribute among the J cells in row i. Recall that constraints (2.23) and (2.24) are derived from the given conditional probabilities (i.e. $\hat{d}_{ij} = \frac{n_{ij}}{\sum_j n_{ij}}$), the largest n_{ij} can be is the value which satisfies $\frac{n_{ij}}{N-(I-1)} = \hat{d}_{ij}$ which means n_{ij} is maximized at $n_{ij} = (N-(I-1))\hat{d}_{ij}$. ∎

To summarize, given the information as in Theorem 2.2.1, the linear relaxation bounds for the ij^{th} cell are given by

$$\hat{d}_{ij} \leq n_{ij} \leq (N-(I-1))\hat{d}_{ij} \tag{2.29}$$

We can extend these results to k-way tables as well, as we will do in Chapter 3.

2.2.3 Results for 2×2 Table

The results of the integer program and linear relaxation are listed in Table 2.2.

	Download Yes	Download No
Male	[3,27],[0.6,29.4]	[2,18],[0.4,19.6]
Female	[1,9],[0.2,9.8]	[4,36],[0.8,39.2]

Table 2.2: IP and LP Results for 2-way Table

We used three different methods to solve these problems, and the results are in Table 2.3. The time recorded is not the total time from when the program started to when it stopped. Instead, it is the time, as close as we could get it, that the actual solvers are working.

Method	IP Time (seconds)	LP Time (seconds)
Cplex	0.00	0.00
MATLAB's `linprog`	n/a	0.36
`lpsolve` in R	0.01	0.01

Table 2.3: 2×2 Example: Comparison of Run-times for 3 Methods

2.3 Cell Bounds for a 4×4 Table, Given Conditional Probabilities and Sample Size

These data are taken from Slavkovic ([42], Sec. 4.6.1), and show the number of juvenile delinquents broken down by county and education level, in Table 2.4. This time, X=county and Y=education level.

	Low	Medium	High	Very High
Alpha	15	1	3	1
Beta	20	10	10	15
Gamma	3	10	10	2
Delta	12	14	7	2

Table 2.4: 4×4 Table. Delinquent Children Data

Consider the case in which we are given an estimate of $P(Education\ Level|County)$ (namely $\hat{D}$), as well as the sample size. We calculate sharp integer bounds for the cells as well as bounds based on the linear relaxation of the integer program.

The matrix of row conditional probabilities,

$$\hat{D} = \begin{bmatrix} 0.75 & 0.05 & 0.15 & 0.05 \\ 0.364 & 0.182 & 0.182 & 0.272 \\ 0.12 & 0.40 & 0.40 & 0.08 \\ 0.343 & 0.40 & 0.20 & 0.057 \end{bmatrix}$$

2.3.1 Formulation of Optimization Problems for 4×4 Example

Similar to Section 2.2.1, an integer program can be constructed as follows (for $N = 135$):

$$\begin{aligned} &Min\ n_{ij} \\ &s.t.\ \sum_i \sum_j n_{ij} = N \\ &\quad \hat{d}_{ij} = \frac{n_{ij}}{\sum_k n_{ik}}\ \forall i,j \\ &\quad \sum_j n_{ij} \geq 1\ \forall i \\ &\quad n_{ij} \geq 0\ \forall i,j \\ &\quad n_{ij}\ integer\ \forall i,j \end{aligned} \tag{2.30}$$

Again, if we know all but one of the conditional probabilities the last one is determined, so we can eliminate four of these constraints. Then (2.30) can be rewritten:

$$\begin{aligned} \hat{d}_{ij} \sum_k n_{ik} - n_{ij} &= \hat{d}_{ij} n_{ij} - n_{ij} + \hat{d}_{ij} \sum_{k \neq j} n_{ik} \\ &= \hat{d}_{ij} \sum_{k \neq j} n_{ik} + (\hat{d}_{ij} - 1) n_{ij} \\ &= 0 \end{aligned}$$

Thus we have the following integer program:

$$Min\ n_{ij} \tag{2.31}$$

$$s.t.\ \sum_i \sum_j n_{ij} = N$$

$$\hat{d}_{ij} \sum_{k \neq j} n_{ik} + (\hat{d}_{ij} - 1) n_{ij} = 0,\ \forall i,\ j = 1, 2, 3 \tag{2.32}$$

$$\sum_j n_{ij} \geq 1\ \forall i$$

$$n_{ij} \geq 0\ \forall i, j$$

$$n_{ij}\ integer\ \forall i, j$$

where $\hat{d}_{ij}$ are elements of $\hat{D}$.

The **A** matrix (see 2.28) is given here:

$$A =$$

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-.25 & .75 & .75 & .75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.05 & -.95 & .05 & .05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
.15 & .15 & -.85 & .15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -.636 & .364 & .364 & .364 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .182 & -.818 & .182 & .182 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .182 & .182 & -.818 & .182 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.88 & .12 & .12 & .12 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4 & -.6 & .4 & .4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4 & .4 & -.6 & .4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.657 & .343 & .343 & .343 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4 & -.6 & .4 & .4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .2 & .2 & -.8 & .2
\end{bmatrix}$$

Because the decimal representations of the numbers in $\hat{D}$ must be rounded, this integer program is infeasible. However, if we consider the conditional probability in terms of the original data, we can construct an integer program that is feasible. Let $\hat{d}_{ij} = \frac{o_{ij}}{\sum_k o_{ik}} =$

$\frac{n_{ij}}{\sum_k n_{ik}}$. Linearizing the second equality leads to:

$$0 = o_{ij}\sum_k n_{ik} - \sum_k o_{ik}n_{ij} = o_{ij}n_{ij} - \sum_k o_{ik}n_{ij} + o_{ij}\sum_{k\neq j} n_{ik}$$
$$= o_{ij}\sum_{k\neq j} n_{ik} + (o_{ij} - \sum_k o_{ik})n_{ij}$$

$\forall i,\ j = 1, 2, 3$, where O is the observed cell counts in Table 2.4. Then, A can be rewritten using O:

$$A = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-5 & 15 & 15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -19 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 3 & -17 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -35 & 20 & 20 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & -45 & 10 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & 10 & -45 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -22 & 3 & 3 & 3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & -15 & 10 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 10 & -15 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -23 & 12 & 12 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & -21 & 14 & 14 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 7 & -28 & 7
\end{bmatrix}$$

In this case it can be reduced to:

$$A = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & 3 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -19 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 3 & -17 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -7 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -9 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 2 & -9 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -22 & 3 & 3 & 3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -3 & 2 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & -3 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -23 & 12 & 12 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -3 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -4 & 1
\end{bmatrix}$$

Thus, sharp integer bounds can be calculated for these cells using the following integer program:

$$\begin{aligned}
&Min\ n_{ij} \\
&s.t.\ \sum_i \sum_j n_{ij} = N \\
&\quad o_{ij} \sum_{k \neq j} n_{ik} + (o_{ij} - \sum_k o_{ik}) n_{ij} = 0,\ \forall i,\ j = 1, 2, 3 \\
&\quad \sum_j n_{ij} \geq 1\ \forall i \\
&\quad n_{ij} \geq 0\ \forall i, j \\
&\quad n_{ij}\ integer\ \forall i, j
\end{aligned} \tag{2.33}$$

Note that the simplification of coefficients in the matrix assumes knowledge of the marginal distribution, and thus under the assumptions of this section would not be available to an intruder. These bounds could only be calculated by the agency releasing the data.

2.3.2 Results for 4×4 Example

The results of the IP, Problem (2.33), as well as its linear relaxation are in Table 2.5. The run-time results of the three different optimization software packages are in Table 2.6.

	Education Level			
County	Low	Medium	High	Very High
Alpha	[15,15],[0.75,99]	[1,1],[0.05,6.6]	[3,3],[0.15,19.8]	[1,1],[0.05,6.6]
Beta	[20,20],[0.36,48]	[10,10],[0.18,24]	[10,10],[0.18,24]	[15,15],[0.27,36]
Gamma	[3,3],[0.12,15.84]	[10,10],[0.4,52.8]	[10,10],[0.4,52.8]	[2,2],[0.08,10.56]
Delta	[12,12],[0.34,45.26]	[14,14],[0.4,52.8]	[7,7],[0.2,26.4]	[2,2],[0.06,7.54]

Table 2.5: IP and LP Results for 4×4 Table

Method	IP Time (seconds)	LP Time (seconds)
Cplex	0.06	0.00
MATLAB's `linprog`	n/a	0.69
`lpsolve` in R	0.50	0.05

Table 2.6: 4×4 Example: Comparison of Run-times for 3 Methods

2.4 Summary

In this chapter we have calculated sharp integer bounds for 2-way tables given the row conditionals D (2.2). We have also developed closed-form solutions for the corresponding linear relaxation bounds in Theorems 2.2.1 and 2.2.2. Similarly, if we were given the column conditionals C (2.1), we could calculate both the integer bounds and the linear relaxations in ways completely analogous to the methods employed in this chapter. We have demonstrated these methods using two examples.

Chapter 3

Calculating Cell Bounds Given Conditional Probabilities: Multi-way Tables

3.1 Setting and Notation

In this chapter we will explore k-way contingency tables, using a variety of examples: (1) a three-way example (Abortion Data); (2) a four-way example (Clinical Trial Data); (3) a six-way example (Czech Autoworkers Data); and (4) an eight-way example (CPS Data).

Let $X = \{X_1, ..., X_k\}$ be a vector of categorical random variables and let $\{i_1, i_2, ..., i_k\}$ be the index sets corresponding to each of the random variables, where $i_1 = 1, ..., I_1$ (I_1 is the number of categories in the first random variable), $i_2 = 1, ..., I_2$, all the way up to $i_k = 1, ..., I_k$. Define $f_X(x)$ as the joint density of these variables and define sets of indices I and J, s.t. $I, J \subset \{i_1, ..., i_k\}$ and $I \cup J \subseteq \{i_1, ..., i_k\}$. Also let X_I be the vector of random variables corresponding to those represented in I and X_J those represented in J. So X_I corresponds to those variables upon which we are conditioning, while X_J corresponds to

all the variables that are *not* being conditioned upon (typically the response variable). For instance, in a 4-way table, there would be four random variables, X_1,X_2,X_3, and X_4 with indices i_1, i_2, i_3, and i_4, where the last variable, X_4 corresponds to the response variable. In that case, we will most often be conditioning upon the first three variables (corresponding to X_1, X_2, and X_3 indices) so that $X_I = \{X_1, X_2, X_3\}$ and $X_J = \{X_4\}$.

Using this notation, the observed full conditional probabilities are represented by

$$\hat{d}_{IJ} = \frac{o_{IJ}}{o_{I.}} = \frac{o_{IJ}}{\sum_J o_{IJ}}$$

which is an estimate of the actual conditional probability $P(X_k|X_1 = i_1, ..., X_{k-1} = i_{k-1})$. This is a direct extension of the notation in Section 2.1.

Using this notation, we can write an integer problem based on the full conditionals in the same form:

$$\begin{aligned}
&Min\ n_{IJ} \\
&s.t.\ \sum_{I,J} n_{IJ} = N \\
&\quad o_{IJ} \sum_{h \neq J} n_{Ih} + (o_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, 2 \\
&\quad \sum_{J} n_{IJ} \geq 1\ \forall\, I \\
&\quad n_{I,J} \geq 0\ \forall I, J \\
&\quad n_{I,J}\ integer\ \forall\, I, J
\end{aligned}$$

The linear relaxation to this problem is below:

$$Min\ n_{IJ} \tag{3.1}$$

$$s.t.\ \sum_{I,J} n_{IJ} = N \tag{3.2}$$

$$o_{IJ} \sum_{h \neq J} n_{Ih} + (o_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, ..., i_k - 1 \tag{3.3}$$

$$\sum_{J} n_{IJ} \geq 1\ \forall\, I \tag{3.4}$$

$$n_{I,J} \geq 0\ \forall I, J$$

3.2 Closed-Form Bounds for *k*-way Tables Given Conditional Probabilities

3.2.1 Closed-Form Bounds for *k*-way Tables Given Full Conditionals

Theorem 2.2.1 can easily be extended to the case of k-way tables, because when considering full conditionals we have essentially collapsed the problem from a k-way table to a 2-way table with dimensions $I_1 I_2 \cdot ... \cdot I_{k-1} \times I_k$. The proof is very similar to that of Theorem 2.2.1.

Theorem 3.2.1 *Assume we have a* k*-way contingency table. Based on the full conditional probabilities, $\hat{d}_{IJ}$ and the sample size, N, we can construct a linear program of the form* (3.1)*. This linear program is minimized at $n_{IJ} = \hat{d}_{IJ}$. That is, the lower bound of the cell represented in the objective function is equal to its associated conditional probability.*

Proof Note that the lower bound for n_{IJ} cannot be zero, because if it were, the other cells which define its conditional distribution would be forced to zero by (3.3). This cannot happen because of constraints (3.4). The constraints (3.3) are derived from the the conditional probability relationship $\hat{d}_{IJ} = \frac{n_{IJ}}{\sum_J n_{IJ}}$. Since (3.4) holds, and n_{IJ} is minimized when its

marginal is as small as possible, n_{IJ} will be minimized when its marginal is 1, which forces n_{IJ} to be precisely equal to $\hat{d}_{IJ}$, its conditional probability. ∎

We can extend Theorem 2.2.2 similarly. Let R be the number of nonzero marginals (i.e. the number of rows which sum to a nonzero number) in the 2-way table that is constructed from the k-way table. If there are no nonzero marginals, $R = I_1 I_2 \cdot ... \cdot I_{k-1}$.

Theorem 3.2.2 *Assuming the a k-way contingency table and LP as in Theorem 2.2.1, the linear program is maximized at* $(N - (R-1))\hat{d}_{IJ}$ *for the* IJ^{th} *cell.*

Proof Since we are maximizing n_{IJ}, the marginal total for each of the rows (beside the I^{th} row) in the contingency table will be as small as possible, namely 1, as required by constraints (3.4). This is possible because each of the cells can have a value equal to their conditional probability, as shown in Theorem 3.2.1. Thus, for all but the i^{th} row, the marginal total is 1. So now there are $N - (R-1)$ counts to distribute among the cells in row I. Since constraints (3.3) are derived from the given conditional probabilities (i.e. $\hat{d}_{IJ} = \frac{n_{IJ}}{\sum_J n_{IJ}}$), the largest n_{IJ} can be is the value which satisfies $\frac{n_{IJ}}{N-(R-1))} = \hat{d}_{IJ}$ which means n_{IJ} is maximized at $n_{IJ} = (N-(R-1))\hat{d}_{IJ}$, which is what we wanted to show. ∎

Thus, the linear relaxation bounds for the IJ^{th} cell are given by

$$\hat{d}_{IJ} \leq n_{IJ} \leq (N - (R-1))\hat{d}_{IJ} \tag{3.5}$$

3.2.2 Closed-Form Bounds for k-way Tables Given Partial Conditionals

We can also develop linear relaxation bounds for partial conditionals. As in Section 3.1, let $X_I \subset X$ be the set of random variables being conditioned on, while $X_J \subset X$ is the set of random variables we are not conditioning on, though in this case $X_I \cup X_J \subset X$ (i.e. not all k of the variables are being considered in this conditional probability). Let $X_K \subset X$ be the

variables not being considered in the conditional probability. In this case, we collapse over the variables in X_K to get to a 2-way table (see Section 3.4 for examples of this).

Then, the conditional probabilities $P(X_J|X_I)$ are estimated from the observed counts using

$$\hat{d}_{IJ.} = \frac{\sum_K o_{IJK}}{\sum_J \sum_K o_{IJK}} \tag{3.6}$$

We define R to be the number of nonzero marginals in the collapsed table defined by the given partial conditionals. If there are no nonzero marginals, $R = \prod_{j:X_j \in X_J} I_j$.

Based upon this we can define the following linear program that will result in bounds on each cell in the original k-way table:

$$min\ n_{IJK} \tag{3.7}$$

$$s.t.\ \sum_I \sum_J \sum_K n_{IJK} = N$$

$$\frac{\sum_K o_{IJK}}{\sum_J \sum_K o_{IJK}} = \frac{\sum_K n_{IJK}}{\sum_J \sum_K n_{IJK}}\ \forall I, J \tag{3.8}$$

$$\sum_J \sum_K n_{IJK} \geq 1\ \forall I \tag{3.9}$$

$$n_{IJK} \geq 0\ \forall I, J, K \tag{3.10}$$

where constraint (3.8) can be linearized in the evident way, and comes from the partial conditional probability defined in (3.6). The integer program is the same, except integer constraints are added.

Theorem 3.2.3 *Assume we have a* k*-way contingency table. Let* $X_I \subset X$, $X_J \subset X$, $X_K \subset X$, *and* $X_I \cup X_J \cup X_K = X$. *Based on the partial conditionals,* $\hat{d}_{IJ.} = P(X_J|X_I)$, *an LP of the form (3.7) can be constructed. In this LP, the cell* n_{IJK} *is minimized at* $n_{IJK} = 0$ *and maximized at* $n_{IJK} = (N - (R-1))\hat{d}_{IJ.}$.

Proof We will first show the maximization, for cell n_{IJK}. Since we are maximizing, each

of the margins defined by (3.9) (besides the I^{th} row) should be as small as possible, at their lower bound of 1. Since R represents the number of margins, satisfying those constraints will take $R-1$ units, which leaves $N-(R-1)$ counts to be distributed among the cells in the I^{th} margin. Then by (3.8) and (3.6) we have the relationship $\frac{\sum_K n_{IJK}}{N-(R-1)} = \hat{d}_{IJ}$. which implies that for $\sum_K n_{IJK}$, the maximum value is $\sum_K n_{IJK} = (N-(R-1))\hat{d}_{IJ}$. But since there are no constraints on this sum (i.e. it does not define a margin in this situation), we can set any element of this sum to $(N-(R-1))\hat{d}_{IJ}$. and the rest of the cells to 0. Thus, $\sum_K n_{IJK} = (N-(R-1))\hat{d}_{IJ}. \Rightarrow n_{IJK} = (N-(R-1))\hat{d}_{IJ}$. and so n_{IJK} is maximized at that point.

However, note that the cell n_{IJK} could be set to 0 as well, letting some other cell or combination of cells in $\sum_K n_{IJK}$ be $(N-(R-1))\hat{d}_{IJ}$. Thus, the minimum value of n_{IJK} is 0. ∎

3.3 Cell Bounds for a 3 × 3 × 3 Table, Given Conditional Probabilities and Sample Size

Table 3.3 has data from the 1972 National Opinion Research Center General Society Survey regarding the attitude of white Christians toward abortion. This dataset is due to Haberman [31] and has a sample size of $N = 1055$.

		Attitudes		
Religion	Education	Positive	Mixed	Negative
North Protestant	≤ 8	9	16	41
	9 - 12	85	52	105
	≥ 13	77	30	38
South Protestant	≤ 8	8	8	46
	9 - 12	35	29	54
	≥ 13	37	15	22
Catholic	≤ 8	11	14	38
	9 - 12	47	35	115
	≥ 13	25	21	42

Table 3.1: Data Describing Attitude of White Christians Toward Abortion in 1972

3.3.1 Formulation of Optimization Problems for 3-way Example

Let $X_1 = Religion$ $(i_1 = 1, 2, 3)$, $X_2 = Education\ Level$ $(i_2 = 1, 2, 3)$, and $X_3 = Attitude$ $(i_3 = 1, 2, 3)$, and let o_{i_1,i_2,i_3} be the observed value in the appropriate cell. Thus, $I = \{i_1, i_2\}$ and $J = \{i_3\}$. Then, the full conditionals are $D = P(Attitude|Religion, Education) = p_{J|I}$.

If we are given the observed conditionals, $\hat{d}_{IJ}$, in addition to the sample size, we can construct an integer program to calculate sharp bounds on the cells just as we have done before (Sections 2.2.1 and 2.3.1), as shown below in components:

$$
\begin{aligned}
&Min\ n_{i_1 i_2 i_3} \\
&s.t.\ \sum_{i_1}\sum_{i_2}\sum_{i_3} n_{i_1 i_2 i_3} = N \\
&\quad \hat{d}_{i_1 i_2 i_3} \sum_{h \neq i_3} n_{i_1 i_2 h} + (\hat{d}_{i_1 i_2 i_3} - 1) n_{i_1 i_2 i_3} = 0,\ \forall i_1, i_2,\ i_3 = 1, 2 \\
&\quad \sum_{i_3} n_{i_1 i_2 i_3} \geq 1\ \forall i_1, i_2 \\
&\quad n_{i_1 i_2 i_3} \geq 0\ \forall i_1, i_2, i_3 \\
&\quad n_{i_1 i_2 i_3}\ integer\ \forall i_1, i_2, i_3
\end{aligned}
$$

To see how this optimization program was constructed, see Sections 2.2 and 2.3. This formulation is a direct extension of the formulations presented in those sections.

Using the index sets defined at this beginning of this section, we can rewrite the above IP

more concisely:

$$
\begin{aligned}
&Min\ n_{IJ} \\
&s.t.\ \sum_{I,J} n_{IJ} = N \\
&\quad \hat{d}_{IJ} \sum_{h \neq J} n_{Ih} + (\hat{d}_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, 2 \\
&\quad \sum_{J} n_{IJ} \geq 1\, \forall\, I \\
&\quad n_{I,J} \geq 0\, \forall I, J \\
&\quad n_{I,J}\ integer\, \forall\, I, J
\end{aligned}
$$

As in Section 2.3, the necessity of rounding the conditional probabilities renders the above IP infeasible. So instead we assume that we have access to the original data (as the data suppliers do) and, assuming the original data in Table 3.3 is denoted by O:

$$
\begin{aligned}
&Min\ n_{IJ} \\
&s.t.\ \sum_{I,J} n_{IJ} = N \\
&\quad o_{IJ} \sum_{h \neq J} n_{Ih} + (o_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, 2 \\
&\quad \sum_{J} n_{IJ} \geq 1\, \forall\, I \\
&\quad n_{I,J} \geq 0\, \forall I, J \\
&\quad n_{I,J}\ integer\, \forall\, I, J
\end{aligned}
$$

The linear relaxation to this problem is below:

$$
\begin{aligned}
&Min\ n_{IJ} \\
&s.t.\ \sum_{I,J} n_{IJ} = N \\
&\quad o_{IJ} \sum_{h \neq J} n_{Ih} + (o_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, 2 \\
&\quad \sum_{J} n_{IJ} \geq 1\, \forall\, I \\
&\quad n_{I,J} \geq 0\, \forall I, J
\end{aligned}
$$

Note that these IP's use general notation that can stretch to include any multi-way table simply by changing the sets I and J.

3.3.2 Results for 3-way Example

The results of the IP and linear relaxation problems defined in the previous section are in Table 3.3.2, and a comparison of the run-times for three methods is in Table 3.3. Note that for the integer program, the `lpsolve` in R failed to provide a solution.

3.4 Cell Bounds for a 4-way Example, Given Conditional Probabilities and Sample Size

This $2 \times 2 \times 2 \times 3$ dataset, with $N = 193$, is due to Koch [35], and shows the number of patients in a clinical trial for an analgesic drug that make recoveries. These patients are given one of two treatments, have one of two possible statuses, and are treated in one of two centers. There are small counts in this dataset, and even some sampling zeros, as can be seen in Table 3.4.

		Attitudes		
Religion	Education	Positive	Mixed	Negative
North Protestant	≤ 8	[9,9], [0.14,142.77]	[16,16], [0.24,253.82]	[41,41], [0.62,650.41]
	9 - 12	[85,85], [0.35,367.75]	[52,52], [0.21,224.98]	[105,105], [0.43,454.28]
	≥ 13	[77,77], [0.53,555.99]	[30,30], [0.21,216.62]	[38,38], [0.26,274.39]
South Protestant	≤ 8	[8,8], [0.13,135.10]	[8,8], [0.13,135.10]	[46,46], [0.74,776.81]
	9 - 12	[35,35], [0.30,310.55]	[29,29], [0.25,257.31]	[54,54], [0.46,479.14]
	≥ 13	[37,37], [0.50,523.50]	[15,15], [0.20,212.23]	[22,22], [0.30,311.27]
Catholic	≤ 8	[11,11], [0.17,182.81]	[14,14], [0.22,232.67]	[38,38], [0.60,631.52]
	9 - 12	[47,47], [0.24,249.79]	[35,35], [0.18,186.02]	[115,115], [0.58,611.19]
	≥ 13	[25,25], [0.28,297.44]	[21,21], [0.24,249.85]	[42,42], [0.48,499.70]

Table 3.2: IP and Linear Relaxation Results for Abortion Data

Method	IP Time (seconds)	LP Time (seconds)
Cplex	0.57	0.00
MATLAB's `linprog`	n/a	0.90
`lpsolve` in R	n/a	0.12

Table 3.3: 3-way Example: Comparison of Run-times for 3 Methods

3.4.1 Formulation of Optimization Problems: 4-way Example

Let $X_1 = Center$ $(i_1 = 1, 2)$, $X_2 = Status$ $(i_2 = 1, 2)$, $X_3 = Treatment$ $(i_3 = 1, 2)$, and $X_4 = Recovery$ $(i_4 = 1, 2, 3)$, and let $o_{i_1 i_2 i_3 i_4}$ be the observed value in the appropriate cell. Thus, $I = \{i_1, i_2, i_3\}$ and $J = \{i_4\}$.

Then, the observed full conditionals are $\hat{D} = P(Recovery|Center, Status, Treatment)$. For instance, $\hat{d}_{1111} = P(R = Poor|C = 1, S = 1, T = 1) = \frac{3}{3+20+5} = 0.107$ and $\hat{d}_{2213} = P(R = Excellent|C = 2, S = 2, T = 1) = \frac{4}{3+9+4} = 0.25$.

The formulation uses the same structure as the previous examples (see Section 3.3.1 and

			Recovery		
Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	3	20	5
		2	11	14	8
	2	1	3	14	12
		2	6	13	5
2	1	1	12	12	0
		2	11	10	0
	2	1	3	9	4
		2	6	9	3

Table 3.4: Clinical Trial Data

			Recovery		
Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	0.107	0.714	0.179
		2	0.333	0.424	0.243
	2	1	0.103	0.483	0.414
		2	0.25	0.542	0.208
2	1	1	0.5	0.5	0
		2	0.524	0.476	0
	2	1	0.188	0.562	0.25
		2	0.333	0.5	0.167

Table 3.5: Full Conditionals for Clinical Trial Data

Appendix A). Similar to early examples, the integer program formulation with the conditional probabilities is infeasible, so we must use the original data. The linear relaxation of this integer program can be constructed by removing the integer constraints.

3.4.2 Results for 4-way Example

The results of this IP, as well as its linear relaxation, are in Table 3.6.

We can also use the conditional probabilities (instead of the original data) for the linear relaxation, and the results are very similar, though slightly different due to the rounding issues that preclude us from solving the integer program with the conditional probabilities in the first place.

Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	[3,6], [0.11,19.93]	[20,40], [0.71,132.86]	[5,10], [0.18,33.21]
		2	11, [0.33,62]	14, [0.42,78.91]	8, [0.24,45.09]
	2	1	3, [0.10,19.24]	14, [0.48,89.79]	12, [0.41,76.97]
		2	[6,12], [0.25,46.5]	[13,26], [0.54,100.75]	[5,10], [0.21,38.75]
2	1	1	[1,18], [0.5,93]	[1,18], [0.5,93]	0
		2	11, [0.52,97.43]	10, [0.48,88.57]	0
	2	1	[3,9], [0.19,34.88]	[9,27], [0.56,104.63]	[4,12], [0.25,46.5]
		2	[2,12], [0.33,62]	[3,18], [0.5,93]	[1,6], [0.17,31]

Table 3.6: IP and LP Relaxation bounds for Clinical Trial Data given Full Conditionals and Sample Size

Table 3.7 compares the 3 methods with respect to run-time.

Method	IP Time (seconds)	LP Time (seconds)
Cplex	0.05	0.00
MATLAB's `linprog`	n/a	0.83
`lpsolve` in R	3.58	0.08

Table 3.7: 4-way Example: Comparison of Run-times for 3 Methods

In this example and in the 2×2 example, the sharp integer bounds do not determine the original table uniquely (evident because the lower bound does not equal the upper bound). In some of the earlier examples such as the 4×4 and *3*-way datasets, the original table was uniquely determined. The reason some examples allow the original counts to be uniquely determined is yet to be established.

3.4.3 4-way Example Given Partial Conditionals: $P(Trt|Center, Status)$

In addition to examining the cell bounds produced given the full conditionals, $P(R|CST)$, we can also look at conditionals involving subsets of the data. For these "small conditionals" we will resort to components notation, though each of the following linear programs is equivalent to the one defined in (3.7).

Suppose the agency releases the small conditional, $P(Treatment|Center, Status)$. This can be calculated from the original data:

$$P(T|CS) = \hat{d}_{i_1 i_2 i_3 \cdot} = \frac{\sum_{i_4} o_{i_1 i_2 i_3 i_4}}{\sum_{i_3} \sum_{i_4} o_{i_1 i_2 i_3 i_4}} \tag{3.11}$$

Referring to notation in Section 3.2.2, $I = \{i_1, i_2\}$, $J = \{i_3\}$, and $K = \{i_4\}$.

These partial conditionals are calculated and shown in Table 3.8, along with the actual counts for each of these cells.

Center	Status	1	2
1	1	0.459 [28]	0.541 [33]
	2	0.547 [29]	0.453 [24]
2	1	0.533 [24]	0.467 [21]
	2	0.471 [16]	0.529 [18]

Table 3.8: $Treatment|Center, Status$ Conditionals and Counts for Clinical Trial Data

Based on this partial conditional as the only given information, the following integer program can be constructed, which will produce sharp bounds on each cell (using original data instead of conditionals because of rounding issues):

$$\begin{aligned}
&min\ n_{i_1 i_2 i_3 i_4} \\
&s.t.\ \sum_{i_1} \sum_{i_2} \sum_{i_3} \sum_{i_4} n_{i_1 i_2 i_3 i_4} = N \\
&\quad - o_{i_1 i_2 2 \cdot} \sum_{i_4} n_{i_1 i_2 1 i_4} + o_{i_1 i_2 1 \cdot} \sum_{i_4} n_{i_1 i_2 2 i_4} = 0 \ \ \forall i_1, i_2 \qquad (3.12) \\
&\quad \sum_{i_3} \sum_{i_4} n_{i_1 i_2 i_3 i_4} \geq 1 \ \ \forall i_1, i_2 \qquad (3.13) \\
&\quad n_{i_1 i_2 i_3 i_4} \geq 0 \ \ \forall i_1, i_2, i_3, i_4 \\
&\quad n_{i_1 i_2 i_3 i_4} \ integer \ \ \forall i_1, i_2, i_3, i_4
\end{aligned}$$

Constraint (3.12) is derived from the $\hat{d}_{i_1 i_2 i_3 \cdot}$'s as well as from the fact that once we know

$\hat{d}_{i_1 i_2 1 \cdot}$, $\hat{d}_{i_1 i_2 2 \cdot}$ is known.

Also, the marginal constraints serve to deter the optimization from considering situations in which there would be a zero marginal, in which case the conditionals do not exist. So in the case of released $P(T|CS)$, the margin of interest is $n_{i_1 i_2 \cdot \cdot} = \sum_{i_3} \sum_{i_4} n_{i_1 i_2 i_3 i_4}$, as in (3.13).

The linear relaxation is the same as the above IP except that the integer constraints are removed.

Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	[0,28], [0,87.21]	[0,28], [0,87.21]	[0,28], [0,87.21]
		2	[0,33], [0,102.79]	[0,33], [0,102.79]	[0,33], [0,102.79]
	2	1	[0,29], [0,103.96]	[0,29], [0,103.96]	[0,29], [0,103.96]
		2	[0,24], [0,86.04]	[0,24], [0,86.04]	[0,24], [0,86.04]
2	1	1	[0,24], [0,101.33]	[0,24], [0,101.33]	[0,24], [0,101.33]
		2	[0,21], [0,88.67]	[0,21], [0,88.67]	[0,21], [0,88.67]
	2	1	[0,16], [0,89.41]	[0,16], [0,89.41]	[0,16], [0,89.41]
		2	[0,18], [0,100.59]	[0,18], [0,100.59]	[0,18], [0,100.59]

Table 3.9: IP and LP bounds for Clinical Trial Data, given $T|CS$ Conditional Probabilities and Original Data

Notice that these bounds are substantially wider than the bounds calculated when given the full conditionals because less information is inherent in the release of these small conditionals. Now, the information concerning the two cells with zero counts is lost. Also, since constraints (3.12) and (3.13) involve summing over the response variable, nothing restricts the IP or LP from setting lower bounds to zero, which it does for all cells.

Another interesting observation is that the sharp integer upper bound is the same as the counts in the marginal table defined by $Treatment \times [Center, Status]$. This seems to be consistent among all the partial conditionals that we looked at. Further, it is known that if we are given a marginal table we can deduce a lower bound of zero and an upper bound equal to the corresponding cell count in this marginal table. It appears as if the bounds given the partial conditionals are the same as those given the corresponding marginal table. This has not been formalized, but we consistently observe it in each of our subsequent

examples.

One other note: the LP bounds shown here are calculated using the original data. Similar bounds could be obtained using the conditional probabilities found in Table 3.8. These bounds are slightly, though not substantially, different due to the rounding that is forced upon the conditional probabilities.

3.4.4 4-way Example Given Partial Conditionals: $P(Center, Status|Trt)$

For this partial conditional, $P(CS|T) = \frac{\sum_{i_4} o_{i_1 i_2 i_3 i_4}}{\sum_{i_1}\sum_{i_2}\sum_{i_4} o_{i_1 i_2 i_3 i_4}} = \frac{o_{i_1 i_2 i_3 \cdot}}{o_{\cdot\cdot i_3 \cdot}}$. These are (rounded and) shown in Table 3.10, along with the actual counts. In this case, $I = \{i_3\}$, $J = \{i_1, i_2\}$, and $K = \{i_4\}$. Notice that as shown in Table 3.10, this has essentially been collapsed down into a two-way problem (see Tables 2.1 and 2.4). In fact, this is the case for any multi-way table (see, for instance, Table 3.17).

Treatment	11	12	21	22
1	0.289 [28]	0.299 [29]	0.247 [24]	0.165 [16]
2	0.344 [33]	0.25 [24]	0.219 [21]	0.187 [18]

Table 3.10: $Center, Status|Treatment$ Conditionals and Counts for Clinical Trial Data

$$
\begin{aligned}
&min\ n_{i_1 i_2 i_3 i_4} \\
&s.t.\ \sum_{i_1}\sum_{i_2}\sum_{i_3}\sum_{i_4} n_{i_1 i_2 i_3 i_4} = N \\
&\quad o_{\cdot\cdot i_3 \cdot}\sum_{i_4} n_{i_1 i_2 i_3 i_4} - o_{i_1 i_2 i_3 \cdot}\sum_{i_1}\sum_{i_2}\sum_{i_4} n_{i_1 i_2 i_3 i_4} = 0\ \ \forall i_1, i_2, i_3 \qquad (3.14) \\
&\quad \sum_{i_1}\sum_{i_2}\sum_{i_4} n_{i_1 i_2 i_3 i_4} \geq 1\ \ \forall i_3 \qquad (3.15) \\
&\quad n_{i_1 i_2 i_3 i_4} \geq 0\ \ \forall i_1, i_2, i_3, i_4 \\
&\quad n_{i_1 i_2 i_3 i_4}\ \ integer\ \ \forall i_1, i_2, i_3, i_4
\end{aligned}
$$

The linear relaxation is a linear program that is the same as the above integer program except that the integer constraints are removed.

Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	[0,28], [0,55.42]	[0,28], [0,55.42]	[0,28], [0,55.42]
		2	[0,33], [0,66.00]	[0,33], [0,66.00]	[0,33], [0,66.00]
	2	1	[0,29], [0,57.4]	[0,29], [0,57.4]	[0,29], [0,57.4]
		2	[0,24], [0,48.00]	[0,24], [0,48.00]	[0,24], [0,48.00]
2	1	1	[0,24], [0,47.51]	[0,24], [0,47.51]	[0,24], [0,47.51]
		2	[0,21], [0,42.00]	[0,21], [0,42.00]	[0,21], [0,42.00]
	2	1	[0,16], [0,31.67]	[0,16], [0,31.67]	[0,16], [0,31.67]
		2	[0,18], [0,36.00]	[0,18], [0,36.00]	[0,18], [0,36.00]

Table 3.11: IP and LP bounds for Clinical Trial Data, given $CS|T$ Conditional Probabilities and Original Data

Notice that the IP bounds are the same in this case as when $P(T|CS)$ is released, but the LP relaxation bounds are not nearly as wide. Also, the IP upper bounds are the same as the original marginal table of $(Treatment) \times (Center, Status)$ (Table 3.10).

3.4.5 4-way Example Given Partial Conditionals: $P(Response|Trt)$

In this case, $P(R|T) = \frac{\sum_{i_1}\sum_{i_2} o_{i_1 i_2 i_3 i_4}}{\sum_{i_1}\sum_{i_2}\sum_{i_4} o_{i_1 i_2 i_3 i_4}} = \frac{o_{..i_3 i_4}}{o_{..i_3}}$.

Treatment	Poor	Modest	Excellent
1	0.216 [21]	0.567 [55]	0.217 [21]
2	0.354 [34]	0.479 [46]	0.167 [16]

Table 3.12: $Response|Treatment$ Conditionals and Counts for Clinical Trial Data

$$min\ n_{i_1 i_2 i_3 i_4} \tag{3.16}$$

$$s.t.\ \sum_{i_1}\sum_{i_2}\sum_{i_3}\sum_{i_4} n_{i_1 i_2 i_3 i_4} = N$$

$$o_{..i_3.}\sum_{i_1}\sum_{i_2} n_{i_1 i_2 i_3 i_4} - o_{..i_3 i_4}\sum_{i_1}\sum_{i_2}\sum_{i_4} n_{i_1 i_2 i_3 i_4} = 0\ \ \forall i_3, i_4 \tag{3.17}$$

$$\sum_{i_1}\sum_{i_2}\sum_{i_4} n_{i_1 i_2 i_3 i_4} \geq 1\ \ \forall i_3 \tag{3.18}$$

$$n_{i_1 i_2 i_3 i_4} \geq 0\ \ \forall i_1, i_2, i_3, i_4$$

$$n_{i_1 i_2 i_3 i_4}\ \ integer\ \ \forall i_1, i_2, i_3, i_4$$

The linear relaxation is constructed similarly, minus the integer constraints. The results to both the IP and the linear relaxation are in Table 3.13. Again, notice that the upper bounds in this table correspond to the counts in the $Response \times Treatment$ marginal table (Table 3.12).

Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	[0,21], [0,41.57]	[0,55], [0,108.87]	[0,21], [0,41.57]
		2	[0,34], [0,68.00]	[0,46], [0,92.00]	[0,16], [0,32.00]
	2	1	[0,21], [0,41.57]	[0,55], [0,108.87]	[0,21], [0,41.57]
		2	[0,34], [0,68.00]	[0,46], [0,92.00]	[0,16], [0,32.00]
2	1	1	[0,21], [0,41.57]	[0,55], [0,108.87]	[0,21], [0,41.57]
		2	[0,34], [0,68.00]	[0,46], [0,92.00]	[0,16], [0,32.00]
	2	1	[0,21], [0,41.57]	[0,55], [0,108.87]	[0,21], [0,41.57]
		2	[0,34], [0,68.00]	[0,46], [0,92.00]	[0,16], [0,32.00]

Table 3.13: IP and LP bounds for Clinical Trial Data, given $R|T$ Conditional Probabilities and Original Data

3.4.6 4-way Example Given Partial Conditionals: $P(Response|Center, Status)$

For this partial conditional, $P(R|CS) = \frac{\sum_{i_3} o_{i_1 i_2 i_3 i_4}}{\sum_{i_3}\sum_{\ell} o_{i_1 i_2 i_3 \ell}} = \frac{o_{i_1 i_2 \cdot i_4}}{o_{i_1 i_2 \cdot \cdot}} = a_{i_1 i_2 i_4}$, where ℓ is an index with the same domain as i_4 ($\ell = 1, 2, 3$) and the elements of A is in Table .

Center	Status	Poor	Modest	Excellent
1	1	0.230 [14]	0.557 [34]	0.213 [13]
	2	0.170 [9]	0.509 [27]	0.321 [17]
2	1	0.511 [23]	0.489 [22]	0 [0]
	2	0.265 [9]	0.529 [18]	0.206 [7]

Table 3.14: $Response|Center, Status$ Conditionals for Clinical Trial Data

$$min\ n_{i_1i_2i_3i_4} \tag{3.19}$$

$$s.t.\ \sum_{i_1}\sum_{i_2}\sum_{i_3}\sum_{i_4} n_{i_1i_2i_3i_4} = N$$

$$o_{i_1i_2\cdot\cdot}\sum_{i_3} n_{i_1i_2i_3i_4} - o_{i_1i_2\cdot i_4}\sum_{i_3}\sum_{i_4} n_{i_1i_2i_3i_4} = 0\ \ \forall i_1, i_2, i_4 \tag{3.20}$$

$$\sum_{i_3}\sum_{i_4} n_{i_1i_2i_3i_4} \geq 1\ \ \forall i_1, i_2 \tag{3.21}$$

$$n_{i_1i_2i_3i_4} \geq 0\ \ \forall i_1, i_2, i_3, i_4$$

$$n_{i_1i_2i_3i_4}\ \ integer\ \ \forall i_1, i_2, i_3, i_4$$

Constraint (3.20) can be rewritten. This allows the problem to be formulated more easily in software such as MATLAB that solves the problem based on standard form matrices and vectors.

Recall that we are representing the observed conditional probabilities by $P(R|CS) = a_{i_1i_2i_4}$. Thus, we have as the basis for this constraint, $P(R|CS) = a_{i_1i_2i_4} = \frac{\sum_{i_3} n_{i_1i_2i_3i_4}}{\sum_{i_3}\sum_{\ell} n_{i_1i_2i_3\ell}}$. This can be rewritten

$$a_{i_1i_2i_4}\sum_{i_3}\sum_{\ell} n_{i_1i_2i_3\ell} - \sum_{i_3} n_{i_1i_2i_3i_4} = a_{i_1i_2i_4}\sum_{i_3} n_{i_1i_2i_3i_4} + a_{i_1i_2i_4}\sum_{i_3}\sum_{\ell\neq i_4} n_{i_1i_2i_3\ell} - \sum_{i_3} n_{i_1i_2i_3i_4}$$
$$= a_{i_1i_2i_4}\sum_{i_3}\sum_{\ell\neq i_4} n_{i_1i_2i_3\ell} - (1 - a_{i_1i_2i_4})\sum_{i_3} n_{i_1i_2i_3i_4} = 0$$

These are constraints $\forall i_1, i_2,\ i_4 = 1, 2$ (because $a_{i_1i_21} + a_{i_1i_22} = 1 - a_{i_1i_23}$).

Center	Status	Treatment	Poor	Modest	Excellent
1	1	1	[0,14], [0,43.61]	[0,34], [0,105.90]	[0,13], [0,40.49]
		2	[0,14], [0,43.61]	[0,34], [0,105.90]	[0,13], [0,40.49]
	2	1	[0,9], [0,32.26]	[0,27], [0,96.79]	[0,17], [0,60.94]
		2	[0,9], [0,32.26]	[0,27], [0,96.79]	[0,17], [0,60.94]
2	1	1	[0,23], [0,97.11]	[0,22], [0,92.89]	0,0
		2	[0,23], [0,97.11]	[0,22], [0,92.89]	0, 0
	2	1	[0,9], [0,50.29]	[0,18], [0,100.59]	[0,7], [0,39.12]
		2	[0,9], [0,50.29]	[0,18], [0,100.59]	[0,7], [0,39.12]

Table 3.15: IP and LP bounds for Clinical Trial Data, given $R|CS$ Conditional Probabilities and Original Data

3.5 Cell Bounds for a 6-way Table Given Conditional Probabilities and Sample Size

This 2^6 dataset is due to Edwards and Havranek [22] and includes data on 1841 Czech Autoworkers, recording prognostic factors for coronary heart disease. The variables were originally labeled A through F, but are renamed to remain consistent with our notation. They are defined as follows: X_1 indicates family anamnesis of coronary heart disease (indexed by i_1); X_2 shows whether the ratio of β to α lipoproteins is above or below 3 (indexed by i_2); X_3 records whether systolic pressure is above or below 140 (indexed by i_3); X_4 indicates whether or not the person is engaged in strenuous physical labor (indexed by i_4); X_5 indicates whether or not the person is engaged in strenuous mental work (indexed by i_5); and X_6 indicates whether or not the person smokes (indexed by i_6).

Table 3.16 has the data, O. Table 3.17 is the row conditionals, $\hat{D}$, and we can read the conditionals this table. For instance,

$\hat{d}_{111111} = P(X_6 = no|X_1 = neg, X_2 = \text{`}< 3\text{'}, X_3 = \text{`}< 140\text{'}, X_4 = no, X_5 = no) = 0.167$

$\hat{d}_{121212} = P(X_6|X_1 = neg, X_2 = \text{`}\geq 3\text{'}, X_3 = \text{`}< 140\text{'}, X_4 = yes, X_5 = no) = 0.615$

3.5.1 Formulation of Optimization Problems for 6-way Example

To clarify, and using the notation introduced at the beginning of this section, let $I = \{i_1, i_2, i_3, i_4, i_5\}$ and $J = \{i_6\}$. Then, using the same structure as in Appendix A, the integer and linear program can be constructed. The sample size is $N = 1841$. Again, because of rounding issues, we used the original data instead of the conditional probabilities.

3.5.2 Results for 6-way Example

The results of the integer program and its linear relaxation for the 6-way example are in Table 3.18. Note that `lpsolve` in R failed to provide a solution for the integer program.

3.6 Cell Bounds for 8-way Table Given Conditional Probabilities and Sample Size

This dataset comes from the Census Bureau's 1993 CPS, and includes 8 variables and a sample size of $48,842$. Table 3.20 gives the variables, number of levels, and their levels.

We will take Salary to be the response variable, and so the full conditionals will be

$$P(Salary|HrsWorked, Sex, Race, Marital\ Status, Employment, Age) \tag{3.22}$$

In the full conditionals for this dataset, we have many margins which are zero. This presents the question as to how to treat these rows, as most inferential procedures proceed under the assumptions that the margins are greater than zero. One possibility is to collapse certain variables to fewer categories, while another option would be to treat those variables as identical to zero.

From the perspective of the agency releasing the data, conditionals with zero marginals are undefined, and would be of little use to those who would want to do inference on the data. Thus, collapsing the table in such a way that no margins are zero is probably the best solution. However, for this dataset, significantly collapsing those variables with four and five categories still does not result in nonzero margins. Also, there seems to be no guidance in the Working Paper 22 by the Federal Committee on Statistical Methodology [58] that would inform this. Thus, we will just assume that any "conditional probability" from a row with a margin of zero is zero itself.

3.6.1 Formulation of Optimization Problems for 8-way Example

To facilitate our standard notation, let $I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$ and $J = \{i_8\}$.

Note that there are $2^4 * 3^2 * 4 * 5 = 2880$ decision variables (cells) and a similar amount of constraints. However, as discussed above, there are many zero marginals in this dataset. To handle this, we will set all cells in which there is a marginal zero to zero itself. We will not even optimize it: $n_{IJ} = 0$ if $\sum_J o_{IJ} = 0$.

$$
\begin{aligned}
& min\ n_{IJ} \forall\, IJ\ s.t. \sum_J o_{IJ} \geq 1 \\
& s.t.\ \sum_{IJ} n_{IJ} = N \\
& \quad - o_{I2} n_{I1} + o_{I1} n_{I2} = 0,\ \forall I \qquad (3.23) \\
& \quad \sum_J n_{IJ} \geq 1\, \forall I\ \ s.t \sum_J o_{IJ} \geq 1 \\
& \quad n_{IJ} \geq 0\, \forall IJ \\
& \quad n_{IJ}\ integer\, \forall IJ
\end{aligned}
$$

where (3.23) is a constraint which arises from the given full conditionals and is derived as

follows:

$$d_{IJ} = \frac{n_{IJ}}{\sum_J n_{IJ}} \Leftrightarrow \tag{3.24}$$

$$(d_{IJ} - 1)n_{IJ} + d_{IJ} \sum_{h \neq J} n_{Iu} = 0 \Leftrightarrow \tag{3.25}$$

$$(d_{I1} - 1)n_{I1} + d_{I1} \sum_{u \neq J} n_{I2} = 0 \Leftrightarrow \tag{3.26}$$

$$-d_{I2}n_{I1} + d_{I1}n_{I2} = 0 \tag{3.27}$$

where (3.26) and (3.27) come from the fact that the response variable, Salary, (which has the subscript t) has only two levels. Other than handling the marginal zeros, this is the same structurally as shown in Appendix A.

3.6.2 Results for 8-way Example

The results are too numerous to give in their entirety. Instead, we present an interesting subset of the results in Table 3.22. For convenience, we present the corresponding original data in Table 3.21.

These results are consistent with those from the other examples. Overall, the IP results are often significantly narrower than the bounds from the linear relaxation.

The IP lower bounds are often the same as the original data. Because of this, it would seem inadvisable to release the full conditional probabilities. However, we have found that for datasets of any significant size the sharp integer bounds cannot be calculated using the conditional probabilities because of rounding issues. In fact, the original counts must be given for the integer program to provide bounds. Since the agency releasing the data could check this before releasing the conditionals, they could potentially release the conditional probabilities knowing that the sharp bounds could not be deduced.

When the lower bounds are not the same as the original data, it seems to be because the

fraction which defines the conditional probabilities can be reduced. If this is the case, the IP lower bound is reduced by that same factor (see Section 4.1 for an example).

As we proved in Theorems 3.2.1 and 3.2.2, the LP lower bound for a particular cell is equal to the conditional probability associated with that cell. Additionally, using Table 3.20 we see that there are $3 * 4 * 5 * 2 * 2 * 2 * 3 = 1440$ rows in the collapsed *2*-way table. However, if the data are examined we find that 302 of the rows have zero marginals. Thus, $R = 1440 - 302 = 1138$, where R is as defined in Section 3.2.1. Thus, each LP upper bound can be calculated as

$$(N - (R - 1))\hat{d}_{IJ} = (48842 - 1137)\hat{d}_{IJ} = 47705\hat{d}_{IJ}$$

The comparison of optimization method run-time is in Table 3.23. Note that the IP was solved on a batch system using a single node with Dual 3.06 GHz Intel Xeon Processors or Dual 3.2 Ghz Intel Xeon Processors and 4 GB of ECC RAM. Details about this system can be accessed at *http://gears.aset.psu.edu/hpc/systems/lionxm/*. Also, neither the integer program or its linear relaxation was solved (or run) on the "hammer" system (see Section 1.4) in `lpsolve` in R.

X_1	X_2	X_3	X_4	X_5	X_6	no	yes	Total
neg	< 3	< 140	no	no		44	40	84
				yes		112	67	179
			yes	no		129	145	274
				yes		12	23	35
		≥ 140	no	no		35	12	47
				yes		80	33	113
			yes	no		109	67	176
				yes		7	9	16
	≥ 3	<140	no	no		23	32	55
				yes		70	66	136
			yes	no		50	80	130
				yes		7	13	20
		≥ 140	no	no		24	25	49
				yes		73	57	130
			yes	no		51	63	114
				yes		7	16	23
pos	< 3	< 140	no	no		5	7	12
				yes		21	9	30
			yes	no		9	17	26
				yes		1	4	5
		≥ 140	no	no		4	3	7
				yes		11	8	19
			yes	no		14	17	31
				yes		5	2	7
	≥ 3	<140	no	no		7	3	10
				yes		14	14	28
			yes	no		9	16	25
				yes		2	3	5
		≥ 140	no	no		4	0	4
				yes		13	11	24
			yes	no		5	14	19
				yes		4	4	8

Table 3.16: Czech Autoworkers Data

X_1	X_2	X_3	X_4	X_5	X_6	no	yes
neg	< 3	< 140	no	no		0.524	0.476
				yes		0.626	0.374
			yes	no		0.471	0.529
				yes		0.343	0.657
		≥ 140	no	no		0.745	0.255
				yes		0.708	0.292
			yes	no		0.619	0.381
				yes		0.4375	0.5625
	≥ 3	<140	no	no		0.418	0.582
				yes		0.515	0.485
			yes	no		0.385	0.615
				yes		0.35	0.65
		≥ 140	no	no		0.490	0.510
				yes		0.562	0.438
			yes	no		0.447	0.553
				yes		0.304	0.696
pos	< 3	< 140	no	no		0.417	.583
				yes		0.7	0.3
			yes	no		0.346	0.654
				yes		0.2	0.8
		≥ 140	no	no		0.571	0.429
				yes		0.579	0.421
			yes	no		0.452	0.548
				yes		0.714	0.286
	≥ 3	<140	no	no		0.7	0.3
				yes		0.5	0.5
			yes	no		0.36	0.64
				yes		0.4	0.6
		≥ 140	no	no		1	0
				yes		0.542	0.458
			yes	no		0.263	0.727
				yes		0.5	0.5

Table 3.17: Full Conditional Probabilities for Czech Autoworkers Data

X_1	X_2	X_3	X_4	X_5	X_6	no	yes
neg	< 3	< 140	no	no		[11,209],[0.52,948.1]	[10,190],[0.48,861.9]
				yes		[112,336],[0.63,1132.51]	[67,201],[0.37,677.49]
			yes	no		[129,258],[0.47,852.15]	[145,290],[0.53,957.85]
				yes		[12,132],[0.34,620.57]	[23,253],[0.66,1189.43]
		≥ 140	no	no		[35,315],[0.74,1347.87]	[12,108],[0.26,462.13]
				yes		[80,320],[0.71,1281.42]	[33,132],[0.29,528.58]
			yes	no		[109,327],[0.62,1120.97]	[67,201],[0.38,689.03]
				yes		[7,168],[0.44,791.88]	[9,216],[0.56,1018.12]
	≥ 3	<140	no	no		[23,161],[0.42,756.91]	[32,224],[0.58,1053.09]
				yes		[35,210],[0.51,931.62]	[33,198],[0.49,878.38]
			yes	no		[5,150],[0.38,696.15]	[8,240],[0.62,1113.85]
				yes		[7,133],[0.35,633.5]	[13,247],[0.65,1176.5]
		≥ 140	no	no		[24,192],[0.49,886.53]	[25,200],[0.51,923.47]
				yes		[73,219],[0.56,1016.38]	[57,171],[0.44,793.62]
			yes	no		[17,170],[0.45,809.74]	[21,210],[0.55,1000.26]
				yes		[7,119],[0.3,550.87]	[16,272],[0.7,1259.13]
pos	< 3	< 140	no	no		[5,160],[0.42,754.17]	[7,224],[0.58,1055.83]
				yes		[7,266],[0.7,1267]	[3,114],[0.3,543]
			yes	no		[9,135],[0.35,626.54]	[17,255],[0.65,1183.46]
				yes		[1,76],[0.2,362]	[4,304],[0.8,1448]
		≥ 140	no	no		[4,220],[0.57,1034.29]	[3,165],[0.43,775.71]
				yes		[11,220],[0.58,1047.89]	[8,160],[0.42,762.11]
			yes	no		[14,182],[0.45,817.42]	[17,221],[0.55,992.58]
				yes		[5,275],[0.71,1292.86]	[2,110],[0.29,517.14]
	≥ 3	<140	no	no		[7,266],[0.7,1267]	[3,114],[0.3,543]
				yes		[1,190],[0.5,905]	[1,190],[0.5,905]
			yes	no		[9,144],[0.36,651.6]	[16,256],[0.64,1158.4]
				yes		[2,152],[0.4,724]	[3,228],[0.6,1086]
		≥ 140	no	no		[1,380],[1,1810]	0
				yes		[13,208],[0.54,980.42]	[11,176],[0.46,829.58]
			yes	no		[5,100],[0.26,476.32]	[14,280],[0.74,1333.68]
				yes		[1,190],[0.5,905]	[1,190],[0.5,905]

Table 3.18: IP Results for Czech Autoworkers Data, Given Full Conditionals and Sample Size, Using Original Data

Method	IP Time (seconds)	LP Time (seconds)
Cplex	3.76	0.00
MATLAB's `linprog`	n/a	1.73
`lpsolve` in R	n/a	0.63

Table 3.19: 6-way Example: Comparison of Run-times for 3 Methods

Table 3.20: CPS Variables and Number of Levels

Variable	Num. of Levels	Levels	Index
Age	3	< 25,25-55,> 55	i
Employment	4	Government, Private, Self-employed, Other	j
Education	5	<HS, HS, College, Bachelor, Bachelor+	k
Marital Status	2	Married, Unmarried	ℓ
Race	2	Non-white, White	m
Sex	2	Male, Female	n
Hours Worked	3	< 40, 40, > 40	s
Salary	2	< 50, 50+	t

							Salary	
Age	Employment	Education	Marital Status	Race	Sex	Hours Worked	< 50	50+
> 55	Government	HS	Married	Non-white	Male	< 40	1	1
						> 40	0	1
						40	7	3
				White	Female	< 40	5	2
						> 40	2	0
						40	0	3
					Male	< 40	22	3
						> 40	10	4
						40	56	24
			Unmarried	Non-white	Female	< 40	8	0
						> 40	1	0
						40	8	0

Table 3.21: Select Data for CPS Example

							Salary	
Age	Employ	Educ	Marital	Race	Sex	Hrs Worked	< 50	50+
> 55	Gov	HS	Married	Non-white	Male	< 40	[1,8528], [0.50, 23852.50]	[1, 8528], [0.50, 23852.50]
						> 40	0	[1, 17056], [1, 47705]
						40	[7, 11942], [0.70, 33393.50]	[3, 5118], [0.30, 14311.50]
				White	Female	< 40	[5, 12185], [0.71, 34075.00]	[2, 4874], [0.29, 13630.00]
						> 40	[1, 17056], [1, 47705]	0
						40	0	[1, 17056], [1, 47705]
					Male	< 40	[22, 15026], [0.88, 41980.40]	[3, 2049], [0.12, 5724.60]
						> 40	[5, 12185], [0.71, 34075.00]	[2, 4874], [0.29, 13630.00]
						40	[7, 11942], [0.70, 33393.50]	[3, 5118], [0.30, 14311.50]
			Unmarried	Non-white	Female	< 40	[1, 17056], [1, 47705]	0
						> 40	[1, 17056], [1, 47705]	0
						40	[1, 17056], [1, 47705]	0

Table 3.22: Select IP/LP Results for CPS Data, Given Full Conditionals and Sample Size

Method	IP Time (seconds)	LP Time (seconds)
Cplex	164,614.73	115.93
MATLAB's `linprog`	n/a	1127.85

Table 3.23: 8-way Example: Comparison of Run-times for 3 Methods

Chapter 4

Discussion of Results

In surveying the results of the previous two chapters, it seems clear that sharp integer bounds produce significantly smaller bounds than the linear relaxation of the same optimization problem. Because of this, it does not seem safe to make disclosure decisions based upon the linear relaxation, even though we have shown that it can be calculated directly from the data.

4.1 Integer Programs

In some of our examples, the sharp integer bounds uniquely identified the counts in the original table. This occurred more often with smaller tables, but actually the most elementary example of all (the 2×2 table) did not yield a unique specification. At this point, we do not understand the underlying characteristics of a table that would produce a unique specification. Perhaps there is some kind of tradeoff between the sample size and the number of cells, though in our examples the ratio between these two quantities certainly does not suggest anything obvious in our examples.

Even if the original counts are not uniquely identified, many of the cells have sharp lower

bounds that are equal to the actual cell count. However, in the case in which the lower bound is less than the actual cell count, the actual cell count is a multiple of the lower bound. The factor by which the actual cell count is reduced is the same factor by which the fractional form of the conditional probability is reduced to in lowest fractional terms.

For instance, for the first cell in the *6*-way example (see Tables 3.17 and 3.18), the sharp lower bound is 11. Notice that the conditional probability for the first cell is $\hat{d}_{111111} = \frac{44}{44+40} = \frac{44}{84} = \frac{11}{21}$.

Further, the sharp upper bounds calculated via the integer programs seem to be an integer multiple of the lower bound and this multiple seems to be constant among rows. So for instance in the 6-way example, the multiple is 19 for the first row of the 6-way example, but 3 for the second row.

It remains to be explained whether this structure could somehow be exploited to allow lower bounds to be reliably calculated for these integer programs.

For the sharp integer bounds given partial conditional information, we have observed that the upper bounds are the same as the cell counts in the corresponding "small" marginal table. Furthermore, the bounds given the partial conditional information seem to be the same as the bounds given the corresponding marginal table. There is more work to be done to understand this relationship.

4.2 Linear Relaxations

For the linear relaxations as we have formulated them given full conditional probabilities, we have shown that the lower bounds for each cell are equal to the conditional probability for that cell (Theorems 2.2.1 and 3.2.1) and that the upper bounds can be directly calculated as well (Theorems 2.2.2 and 3.2.2). To calculate these upper bounds, we only need to know the given conditional probability, the given sample size, and the number of nonzero marginals

in the *2*-way table which is constructed from the k-way table (we called this quantity R in Section 3.2.1).

For instance, in the *6*-way example (see Tables 3.17 and 3.18), there are no zero marginals, so $R = 2^5 = 32$ since each of the five variables we are conditioning on has two categories. Then, each linear relaxation upper bound can be calculated by

$$(N - (R-1))\hat{d}_{IJ} = (1841 - 31)\hat{d}_{IJ} = 1810\hat{d}_{IJ}$$

and this can be checked using Tables 3.17 and 3.18.

We have also shown that given partial conditional probabilities still allows us to directly deduce linear relaxation bounds (Theorem 3.2.3). This is done in similar fashion to the full conditional case.

There are large discrepancies between the sharp IP bounds and the linear relaxation bounds. It does not seem as if LP bounds serve as good approximations to the IP bounds at all, and thus the bounds derived from the LP bounds are not useful for detecting whether there is a potential disclosure.

On the other hand, in all except the toy example in Section 2.2 the IP based on the released conditionals proved infeasible anyway. Therefore, it is likely that in practice releasing conditional probabilities would not allow intruders to calculate sharp integer bounds (and this could be checked by the agency before releasing the data).

4.3 Discussion of Method Performance (Including Genetic Algorithms)

Clearly, Cplex performs the best for both the integer programs as well as the linear relaxations. All problems, with the exception of the 8-way IP, were solved with ease using

this powerful commercial software. On the other hand, the `lp_solve` implementation in R could not solve several of the larger IP's, and would not even solve the 8-way LP. Matlab's Optimization Toolbox solved each of the LP's, but does not have IP solving capability. It seems clear then, that if large-scale optimization must be conducted, Cplex or another similar commercial package is likely preferrable to an add-on or freeware optimization package (though there could be more effective freeware packages out there).

It is also clear that even the best solvers may be pushed to or past their practical limit for integer programs of considerable size. In the 8-way example, not only was each optimization problem large, with thousands of constraints and variables, but to calculate the lower and upper bounds, the optimizations had to be repeated thousands of times. In the course of the Cplex 8-way IP calculation, all of the lower bounds were calculated in under 10 seconds, but a few of the upper bounds took hours to find an optimal solution.

We experimented with a Genetic Algorithm (GA) implementation in MATLAB as well. It solved the small 2×2 linear relaxation example, but it took between 20 and 47 seconds to do each of the required 8 optimizations. It would take at least that long to solve the integer program, as well as the larger problems, and so for all except the larger problems the GA would seem not be competitive with Cplex.

However, this is an area that potentially warrants more investigation, if bounds on large problems need to be calculated. One word of caution though: for a truly large problem, there will be thousands (maybe millions) of optimizations to be done. Even if the GA could solve each in a half a minute, the time required to solve each bound could still be prohibitively large.

Chapter 5

Calculation of Cell Bounds Given Odds Ratios

5.1 Odds Ratios

In a 2×2 table, the odds ratio is defined to be

$$\alpha = \frac{p_{11}p_{22}}{p_{12}p_{21}} \tag{5.1}$$

$$\hat{\alpha} = \frac{o_{11}o_{22}}{o_{12}o_{21}} \tag{5.2}$$

For a 2-way table we can define two additional odds ratios:

$$\alpha_1 = \frac{p_{11}p_{12}}{p_{21}p_{22}} \tag{5.3}$$

$$\hat{\alpha}_1 = \frac{o_{11}o_{12}}{o_{21}o_{22}} \tag{5.4}$$

$$\alpha_2 = \frac{p_{11}p_{21}}{p_{12}p_{22}} \tag{5.5}$$

$$\hat{\alpha_2} = \frac{o_{11}o_{21}}{o_{12}o_{22}} \tag{5.6}$$

5.2 Calculating Bounds Given Odds Ratios and Sample Size

Based on the odds ratio information only, we can construct a Mixed Integer Quadratically Constrained Program (MIQCP) and calculate sharp bounds on each cell, as well as a linear relaxation of this (see Section 1.6.3).

We return to the fictional example in Table 2.1. In this example, the observed $\hat{\alpha} = \frac{15 \cdot 20}{5 \cdot 10} = 6$. Then, we can construct the following MIQCP:

$$min\ n_{ij} \tag{5.7}$$

$$s.t.\ \sum_i \sum_j n_{ij} = 50 \tag{5.8}$$

$$\alpha n_{12} n_{21} - n_{11} n_{22} = 0 \tag{5.9}$$

$$n_{12} n_{21} \geq 1 \tag{5.10}$$

$$n_{ij} \geq 0\ \forall i, j$$

$$n_{ij}\ integer\ \forall i, j$$

Constraint (5.9) is derived from (5.1), and constraint (5.10) does not allow the denominator of the odds ratio to be zero. The linear relaxation of this problem is the optimization problem defined by (5.7) except the integer constraints are removed. We can derive similar MIQCP's using α_1 and α_2 (see Appendix B).

5.3 Analytical Bounds for Some Cells in Linear Relaxation

A closed-form solution to the linear relaxations of (5.7), (B.1), and (B.4) for some of the cells in the two-way table can be derived.

Theorem 5.3.1 *Assume $N \geq \alpha + 3$, where $\alpha \in \mathbb{R}$ is one of* (5.1), (5.3), *or* (5.5) *and N*

is the sample size. Then the solutions (lower and upper bounds) to (5.7)*,* (B.1)*, and* (B.4)*, for those cells in the numerators of the appropriate odds ratios, are given by*

$$n_{ij} = \frac{(N-2) \pm \sqrt{(N-2)^2 - 4\alpha}}{2}. \tag{5.11}$$

Proof We will use (5.7) and cell n_{11}, but the same logic applies to each of the odds ratios and each of the cells in their numerators. First, rewrite (5.9) for convenience as $\alpha n_{12} n_{21} = n_{11} n_{22}$. Let $a = n_{12} + n_{21}$ and $b = n_{12} n_{21}$. Then, $n_{11} + n_{22} = N - a$ because of (5.8) and since constraint (5.9) must hold, $n_{11} n_{22} = b\alpha$. Solving these two equations gives two solutions for n_{11}:

$$n_{11} = \frac{(N-a) \pm \sqrt{(N-a)^2 - 4b\alpha}}{2} \tag{5.12}$$

Now all that is left to be proved is that the roots of n_{11} (call them n_ℓ for the smaller root, and n_u for the larger one) are the upper and lower bounds when the denominator of the odds ratio is as small as possible ($n_{12} = n_{21} = 1$).

To prove that the larger root is maximized when $n_{12} = n_{21} = 1$, assume it is true, which means $a = 2$ and $b = 1$. In order for $(N-2) - \sqrt{(N-2)^2 - 4\alpha} \leq 2$, one can easily show that $\alpha \leq N - 3$, and if this is the case, then $n_\ell \leq 1$. Note also that $n_\ell + n_u = N - a$ which means $n_u = N - a - n_\ell$. Thus, when $a = 2$, $n_u = N - 2 - n_\ell$ which will be no smaller than $n_u = N - 3$. When $a \geq 3$, n_u will be no larger than $n_u = N - 3$. Therefore, n_u is less when $a \geq 3$ than when $a = 2$.

To prove that the smaller root is minimized when $n_{12} = n_{21} = 1$, note that the form of (5.12) is such that as b increases, $\sqrt{(N-a)^2 - 4b\alpha}$ gets further away from $N - a$, so that $(N-a) - \sqrt{(N-a)^2 - 4b\alpha}$ is smallest when b is as small as possible. ∎

We have not provided results for those cells in the linear relaxation that are in the denominator of the odds ratio, or for the MIQCP. However, perhaps this could be extended to $I \times J$ or multi-way table in some way and a closed-form solution could serve as a shortcut in calculating linear relaxation bounds in these problems.

5.4 Results for MIQCP's and QCP's Given Odds Ratios and Sample Size

The results of both of the optimization problems (the MIQCP and its linear relaxation) for each of the odds ratios are in Tables 5.1, 5.2, and 5.3.

[3,36], [0.13,47.87]	[1,24], [0.02,45.08]
[1,24], [0.02,45.08]	[3,36], [0.13,47.87]

Table 5.1: MIQCP and QCP Bounds for 2way Table Given α and Sample Size

[2,33],[0.03,47.97]	[2,33],[0.03,47.97]
[2,28],[0.02,47.53]	[2,28],[0.02,47.53]

Table 5.2: MIQCP and QCP Bounds for 2-way Table Given α_1 and Sample Size

[1,27],[0.01,47.99]	[2,36],[0.02,48.75]
[1,27],[0.01,47.99]	[2,36],[0.02,48.75]

Table 5.3: MIQCP and QCP Bounds for 2-way Table Given α_2 and Sample Size

5.5 Discussion of Cell Bound Results Given Odds Ratio Information

The first thing to note about the results is that, given a particular odds ratio, the bounds for cells in the numerator of that odds ratio are the same, and similarly for those cells in the denominator. This is in contrast to the conditionals, which give information about each cell individually.

Based only on this single example, it seems as if the conditionals do indeed provide more information than any of the odds ratios by themselves. Indeed, both the integer and linear bounds induced by the conditional probability information (Table 2.2) are narrower than any of the bounds induced by any of the three odds ratios individually (Tables 5.1, 5.2, and 5.3).

Chapter 6

Conclusions

In this thesis we have formulated optimization problems that allow the calculation of sharp integer bounds as well as linear relaxation bounds given the sample size and either partial or full conditional information. We have also shown that linear programs are not necessary to calculate the linear relaxation bounds, as they can be calculated directly from the data. We see that there may be large gaps between the sharp bounds and those resulting from the linear relaxation. Though the calculation of the linear relaxation bounds are direct and thus much less time-intensive than the corresponding integer programming bounds, it is inadvisable in practice to substitute bounds from the linear relaxation for sharp bounds, because of these large gaps.

In calculating the sharp integer bounds, often the lower bound is equal to the actual cell count. When this is not the case, it is because of the structure of the data. If for a given cell the fraction used to calculate the conditional probability can be reduced, the lower bound will be reduced by the same factor.

We also formulated and solved optimization problems for a 2×2 table given odds ratios and sample size, and demonstrated at least in this example that the bounds are looser than those bounds derived given conditional information. Furthermore, we found for the linear

relaxation of this problem, the bounds for some of the cells can be calculated via a formula instead of solving the optimization problem.

6.1 Future Work

Still to be understood is why some contingency tables can be uniquely specified by their full conditional probabilities and others cannot. Additionally, though we have determined formulas which allow us to calculate linear relaxation bounds directly, the sharp integer bounds are not as well understood. They are evidently closely tied to the counts in the original table, because the original counts seem to always be a multiple of the lower bound, and the upper bound seems to be a multiple of the lower bound. Whether these relationships can be exploited to calculate these bounds more directly is yet to be explored.

Another issue that warrants more investigation is the relationship between the space of tables induced given partial conditional information and the space of tables induced by the corresponding marginal table. It seems in our examples that the bounds in both of these cases are the same.

We were able to calculate sharp integer bounds for an 8-way table, though the time to solve it was prohibitively large. What about tables that are even larger? Perhaps some sort of evolutionary optimization algorithm could be constructed that could solve this problem in a reasonable amount of time. This is yet to be fully explored.

Appendix A

IP/LP Formulations to Larger Multi-way Tables

Any IP formulation given full conditional information can be fit into the following framework, based on the notation developed in Section 3:

$$
\begin{aligned}
&Min\ n_{IJ} \\
&s.t.\ \sum_{I,J} n_{IJ} = N \\
&\quad o_{IJ} \sum_{h \neq J} n_{Ih} + (o_{IJ} - 1) n_{IJ} = 0,\ \forall\, I,\ J = 1, ..., K-1 \\
&\quad \sum_{J} n_{IJ} \geq 1\, \forall\, I \\
&\quad n_{I,J} \geq 0\, \forall I, J \\
&\quad n_{I,J}\ integer\, \forall\, I, J
\end{aligned}
$$

Note that this formulation assumes use of the original data instead of the conditionals. It also is an integer program, though the linear relaxation could be calculated by dropping the integer constraints.

Appendix B

Formulation of MIQCP's for α_1 and α_2

MIQCP based on α_1:

$$min\ n_{ij} \tag{B.1}$$

$$s.t.\ \sum_i \sum_j n_{ij} = 50$$

$$\alpha_1 n_{21} n_{22} - n_{11} n_{12} = 0 \tag{B.2}$$

$$n_{21} n_{22} \geq 1 \tag{B.3}$$

$$n_{ij} \geq 0\ \forall i, j$$

$$n_{ij}\ integer\ \forall i, j$$

We can also construct an MIQCP based on α_2:

$$min\ n_{ij} \tag{B.4}$$

$$s.t.\ \sum_i \sum_j n_{ij} = 50$$

$$\alpha_2 n_{12} n_{22} - n_{11} n_{21} = 0 \tag{B.5}$$

$$n_{12} n_{22} \geq 1 \tag{B.6}$$

$$n_{ij} \geq 0\ \forall i, j$$

$$n_{ij}\ integer\ \forall i, j$$

Bibliography

[1] Balke, A. and Pearl, J. "Counterfactual probabilities: Computational methods, bounds and applications." *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, Morgan Kauffman, 46-54.

[2] Balke, A. and Pearl, J. "Bounds on Treatment Effects From Studies with Imperfect Compliance." *Journal of the American Statistical Association*, 92:1171-1176 (1997).

[3] Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D. *Linear Programming and Network Flows.* John Wiley & Sons, Inc., Hoboken, NJ, USA, 3rd Edition (2005).

[4] Bentler, P. M. and Woodward, J. A. "Inequalities among lower bounds to reliability: with applications to test construction and factor analysis." *Psychometrika*, 45:249-267 (1980).

[5] Best, M.J. and Ritter, K., *Linear Programming: active set analysis and computer programs*, 1985, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

[6] Bland, R.G., Goldfarb, D., and Todd, M.J. "The Ellipsoid Method: A Survey." *Operations Research*,29(6):1039-1091 (1981).

[7] Boggs, P.T. and Tolle, J.W. "Sequential Quadratic Programming." *Acta Numerica*, *citeseer.ist.psu.edu/boggs95sequential.html* 1-51 (1995).

[8] Bonferroni, C. E. *Teoria statistica delle classi e calcolo delle probabilitá*, Vol. 8. Publicazioni del R. Instituto Superiore di Scienze Economiche e Commerciali di Firenze (1936).

[9] Branch, M.A., Coleman, T.F., Li, Y. "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems." *SIAM Journal on Scientific Computing*, 21(1):1-23 (1999).

[10] Buzzigoli, L. and Gusti, A. "An Algorithm to Calculate the Upper and Lower Bounds of the Elements of an Array Given its Marginals." in *Statistical Data Protection (SDP '98) Proceedings*, 131-147. Eurostat, Luxembourg.

[11] Byrd, R.H., Schnabel, R.B., Shultz, G.A. "Approximate Solution of the Trust Region Problem by Minimization Over Two-Dimensional Subspaces." *Mathematical Programming*, 40:247-263 (1988).

[12] Causey, B., Cox, L., and Ernst, L. "Applications of Transportation Theory to Statistical Problems." *Journal of the American Statistical Association*, 80(392):903-909 (1985).

[13] Coleman, T.F. and Y. Li. "An Interior, Trust-Region Approach for Nonlinear Minimization Subject to Bounds." *SIAM Journal on Optimization*, 6:418-445 (1996).

[14] Coleman, T.F. and Y. Li. "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds." *Mathematical Programming*, 67(2):189-224 (1994).

[15] Cox, L. "Suppression Methodology and Statistical Disclosure Control." *Journal of the American Statistical Association*, 75:377-385 (1980).

[16] Cox, L. and Ernst, L. "Controlled Rounding." *INFOR*, 20:423-432 (1982)

[17] Cox, L. "Network Models For Complementary Cell Suppression." *Journal of the American Statistical Association*, 90(432):1453-1462 (1995).

[18] Cox, L. and George, J. "Controlled Rounding for Tables with Subtotals." *Annals of Operations Research*, 20:141-157 (1989).

[19] Dantzig, G.B., Orden, A., and Wolfe, P. "Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Constraints." *Pacific Journal Math.*, 5:183-195 (1955).

[20] Dobra, A. and Fienberg, S. E. "Bounds for Cell Entries in Contingency tables given marginal totals and decomposable graphs." *Statistical Journal of the United Nations Economic Commission for Europe*, 18(4):363-371 (2001).

[21] Dobra, A. and Fienberg, S. E. "Bounds for Cell Entries in Contingency Tables Induced by Fixed Marginal Totals." *Statistical Journal of the United Nations ECE*, 18:363-371 (2003).

[22] Edwards, D., and Havranek, T. "A fast procedure for model search in multidimensional contingency tables." *Biometrika*, 72:339-351 (1985).

[23] Fienberg, S. E. "Fréchet and Bonferroni Bounds for Multi-Way Tables of Counts With Applications to Disclosure Limitation." in *Statistical Data Protection: Proceedings of the Conference*, Luxembourg: Eurostat, 115-129 (1999).

[24] Fienberg, S. E. "Contingency Tables and Log-Linear Models: Basic Results and New Developments," *Journal of the American Statistical Association*, 95(450):643-647 (2000).

[25] Fienberg, S., Makov, U., Meyer, M., and Steele, R. "Computing the exact distribution for a multi-way contingency table conditional on its marginal totals." In Saleh, P. (ed.), *Data Analysis from Statistical Foundations: A Festschrift in Honor of the 75th Birthday of D.A.S. Fraser*, 145-165. Nova Science Publishers, Huntington, NY (2001).

[26] Fienberg, S. E. and Slavkovic, A. B., "Preserving the Confidentiality of Categorical Statistical Data Bases When Releasing Information for Association Rules." *Data Mining and Knowledge Discovery*, 11:155-180 (2005).

[27] Fletcher, R. "A nonlinear programming problem in statistics (educational testing)." *SIAM Journal on Scientific and Statistical Computing*, 2:257-267 (1981).

[28] Fréchet, M. Les Probabilitiés, Associées a un Système dÉvénments Compatibles et Dépendants, Vol. Premiere Partie. Hermann & Cie, Paris (1940).

[29] Freund, R. and Mizuno, S. "Interior Point Methods: Current Status and Future Directions." *Optima*, 51:1-9 (1996).

[30] Gill, P.E., Murray, W., and Wright, M.H. *Practical Optimization*, Academic Press, London (1981).

[31] Haberman, S. J., *The Analysis of Qualitative Data*, vol. 1,2. Academic Press, Orlando (1978).

[32] Hoeffding, W. "Scale-invariant correlation theory." Schriften des Mathematischen Instituts und des Instituts für Angewandte Mathematik der Universität Berlin, Vol. 5(3):181233 (1940).

[33] Hosten, S. and Sturmfels, B. "Computing the Integer Programming Gap." *http://www.citebase.org/abstract?id=oai:arXiv.org:math/0301266*, 2003.

[34] Karmarkar, N. "A New Polynomial-Time Algorithm for Linear Programming." *Combinatorica*, 4:373-395 (1984).

[35] Koch, G., Amara, J., Atkinson, S., and Stanish, W. "Overview of categorical analysis methods." SAS-SUGI, 8:785-795 (1983).

[36] Manski, C. F. "Anatomy of the Selection Problem." *Journal of Human Resources*, 24:343-360 (1989).

[37] Manski, C. F. "Nonparametric bounds on treatment effects." *American Economic Review, Papers and Proceedings*, 80:319-323 (1990).

[38] Mehrotra, S. "On the Implementation of a Primal-Dual Interior Point Method." *SIAM J. Optimization*, 2(4):575-602 (1992).

[39] Moré, J.J. and Sorensen, D.C. "Computing a Trust Region Step." *J. Sci. Stat. Comput.*, 4(3) (1983).

[40] Nemhauser, George L. and Wolsey, Laurence A., *Integer and Combinatorial Optimization*, Wiley-Interscience (1988).

[41] Shapiro, A. "The Asymptotic Bias of Minimum Trace Factor Analysis, With Applications to the Greatest Lower Bound to Reliability." *Psychometrika*, 65(3):413-425 (2000).

[42] Slavkovic, A. B. "Statistical Disclosure Limitation Beyond the Margins: Characterization of Joint Distributions for Contingency Tables." *PhD Thesis*, Carnegie Mellon University (2004).

[43] Slavkovic, A. B. and Fienberg, S. E. "Bounds for cell entries in two-way tables given conditional relative frequencies." In Domingo-Ferrer, J. and Torra, V. (eds.), *Privacy in Statistical Databases– PSD '2004, Lecture Notes in Computer Science No. 3050*, 30-43. Springer-Verlag (2004).

[44] Sullivant, S. "Small Contingency Tables with Large Gaps." *Siam J. Discrete Math*, 18(4):787-793 (2005).

[45] Tawarmalani, M. and Sahinidis, N. V. "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study." *Mathematical Programming*, 99(3):563-591 (2004).

[46] Vandenberghe, L. and Boyd, S. "Semidefinite Programming." *SIAM Review*, 38(1):49-95 (1996).

[47] Watson, G. A. "Algorithms for minimum trace factor analysis." *SIAM J. on Matrix Analysis and Applications*, 13:1039-1053 (1992).

[48] Willenborg, L. and de Waal, T. *Statistical Disclosure Control in Practice*. Lecture Notes in Statistics **111** Springer, New York (1996).

[49] Wright, S.J. *Primal-Dual Interior-Point Methods*, SIAM (1997).

[50] Zhang, Y. "Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment." Technical Report TR96-01, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD (1995).

[51] Zayatz, L. "Using Linear Programming Methodology for Disclosure Avoidance Purposes." *Bureau of the Census Statistical Research Division Research Report Series No. RR-92/02*. U.S. Bureau of the Census Statistical Research Division (1992).

[52] Sahinidis, N. V. and Tawarmalani, M. "Baron 7.4: Global Optimization of Mixed-Integer Nonlinear Programs." *Users Manual*. Available at http://www.gams.com/dd/docs/solvers/baron.pdf.

[53] ILOG CPLEX 10.1 User's Manual. ILOG (2006).

[54] GAMS Users Guide. GAMS Development Corporation, Washington D.C. *http://www.gams.com/docs/gams/GAMSUsersGuide.pdf* (2007).

[55] `lp_solve` Reference Guide. *http://lpsolve.sourceforge.net/5.5/* (2007).

[56] Optimization Toolbox Users Guide, Mathworks Inc. (2001).

[57] Linear and Integer Programming, R Documentation. *http://rweb.stat.umn.edu/R/library/lpSolve/html/lp.html*.

[58] Statistical Policy Working Paper 22. Report on Statistical Disclosure Limitation Methodology, Federal Committee on Statistical Methodology (Second Version, 2005).