Topics at the interface of optimization and statistics

by

Tu Nguyen

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

July, 2020

Abstract

Optimization has been an important tool in statistics for a long time. For example, the problem of parameter estimation in a statistical model, either by maximizing a likelihood function [82, Chapter 8] or using least squares approach [46, Chapters 5-6], reduces to solving an optimization problem. Not only has optimization been utilized in solving traditional statistical problems, it also plays a crucial role in more recent areas such as statistical learning. In particular, in most statistical learning models, one learns the best parameters for the model through minimizing some cost function under certain constraints.

In the past decade or so, there has been an increasing trend in going to reverse direction: Using statistics as a powerful tool in optimization. As learning algorithms become more efficient, researchers have focused on finding ways to apply learning models to improve the performance of existing optimization algorithms [91, 65, 11]. Following their footsteps, in this thesis, we study a recent algorithm for generating cutting planes in mixed integer linear programming problems and show how one can apply learning algorithms to improve the algorithm.

In addition, we use the decision theory framework to evaluate whether the solution given by the sample average approximation, a commonly used method to solve stochastic programming problems, is "good". In particular, we show that the sample average solution is admissible for an uncertain linear objective over a fixed compact set and for a convex quadratic function with an uncertain linear term over box constraints when the dimension $d \leq 4$.

Finally, we combine tools from mixed integer programming and Bayesian statistics to solve the catalog matching problem in astronomy, which tries to associate an object's de-

tections coming from independent catalogs. This problem has been studied by many researchers [26, 27, 88]. However, the most current algorithm to tackle the problem, as in [88], is only shown to work with 3 catalogs. In this thesis, we extend this algorithm to allow for matching across a higher number of catalogs. In addition, we introduce a new algorithm that is more efficient and scales much better with large number of catalogs.

Primary Reader: Amitabh Basu

Secondary Reader: Dan Naiman

Acknowledgements

In all honesty, writing the acknowledgement is one of the harder parts in writing this thesis. Expressing my gratitude to any of the following people in a couple of sentences is almost impossible, as there are not enough words to adequately express my appreciation for what they have done for me.

It took me some time to ponder who I should put first. Is it the person who has helped me the most throughout my journey or should it be one that has influenced me most? Fortunately, there is a common answer to all those questions: my advisor, Amitabh Basu! So, thank you, Amitabh, for your endless support since the day we started working together. Thank you for your patience in guiding me along this path. As I did not have any formal background in optimization before coming to Hopkins, you really took your time to instruct me from the most basic concepts. Finally, thank you for being an outstanding role model. Your curiosity and your desire to learn new subjects that are outside of your area of expertise encourage me to always broaden my knowledge.

I would like to thank my qualifying exams committee members and dissertation defense panels for their suggestions and feedbacks. In particular, thank you, Dan, for your formal introduction to the field of statistics through your class and for asking me insightful questions during my GBO exam. Your questions made me aware of the holes I have in my knowledge. To Tamas, thank you for being such a collaborator. Working with you in the catalog matching project not only helps me solidifying my programming skills but also teaches me a lesson that the presentation of the work is just as important as the work itself.

At Hopkins, I have also had the opportunity to learn from many amazing faculty members. I want to especially thank faculty members John Miller, James Fill, Donniell Fishkind,

Avanti Athreya, Steve Hanke, Helyette Geman, and Fred Torcaso for letting me pick your brain. In addition, to Kristin, Lindsay, Sandy, Ann, Seanice, and Sonjala, thank you for being the unsung heroes in the department. Your work has made my life at Hopkins easier.

To admission committee members, whom I do not know, thank you for giving me a chance to study at Hopkins. I graduated from a not-so-stellar college and honestly did not expect to get accepted here. Thank you for believing in me and I hope I proved you right!

I feel very fortunate to call so many wonderful PhD students in the department my friends. To William, Wayne, Lingyao, Ke, Andrew, Hsi-Wei, Tianyu, Chu-chi, thank you being there with me through thick and thin. Without you all, my experience would not be as great and enjoyable.

Finally, I would like to thank my family back home, epsecially my mom and my grandma, for their constant support and encouragement. They are the greatest source of my strength. Finally, this would not feel complete without mentioning my girlfriend. Thank you for always being by my side, for comforting me through hard times, and for listening to all my crazy research ideas. You know how much I love you.

This thesis is dedicated to my whole family, especially my parents and my grandparents.

Simply thinking about you gives me all the strength to keep going.

Contents

List of Figures

Chapter 1

Introduction and Motivation

Optimization and Statistics are two branches of applied mathematics that deal with different types of problems and have different applications. In particular, optimization is a great tool in decision making, such as how many products should be shipped from each factory to each warehouse or how much should investors allocate their capital between stocks and bonds. Statistics, on the other hand, allows one to analyze data and make inference about future values. An example is by studying past weather patterns, one could give a prediction on how likely it is to rain tomorrow.

Optimization methods has traditionally been used in many important areas of statistics, such as parameter estimation through maximizing the likelihood function and regression problems. Recently, with the increase in popularity of statistical learning, many statistical learning algorithms have been used to improve the performance of existing optimization methods (see [11, 19, 71, 60], and [1] for some examples of works in this direction). This dissertation was inspired by this new development. We explore how statistics can be utilized in optimization methods. In addition, this disseration also shows how one could combine techniques in both statistics and optimization to solve a problem in astronomy. In the rest of this chapter, we give a formal introduction to the fields of optimization and statistics.

1.1 Optimization

Optimization is a branch of applied mathematics that studies techniques to solve problems where one wants to find the values of a set of variables, or inputs, that maximize or minimize some objective function of interest. The objective is typically a function of this collection of variables, which one has control over. This kind of problem is central to any decision making process; hence optimization applications are ubiqutious, ranging from economics, finance, and manufacturing to engineering, scheduling, and transportation.

There are three main components in an optimization problem: The variables, the objective function, and the constraints. The first two are already discussed in the previous paragraph. The last component, constraints, is a set of equalities and inequalities on the variables that controls what values these variables can take. The region defined by these constraints is called the *feasible set*. A general mathematical optimization problem can now be defined.

Definition 1.1. (Optimization Problem). *Given a function* $f : \mathbb{R}^n \to \mathbb{R}$ *and a feasible set* $S \in \mathbb{R}^n$*, the optimization problem is to solve* $\min_{x \in S} f(x)$.

Depending on the function f and the feasible set S, we can classify an optimization problem into many types. It is important to know what category one's particular problem falls into because algorithms for solving optimization problems are often tailored to a particular type of problem. Providing a taxonomy of optimization is intractable: One can get to a smaller and more specialized class of problems by combining the different types of constraints and objective function. We only give examples of some important types for illustrative purposes.

1. When $S = \mathbb{R}^n$, i.e. there are no constraints, the problem is an *unconstrained optimization* problem. "Line search methods" and "trust region methods" are two most popular techniques to solve this type of problem (see [80, Chapters 2-6]). On the other hand, when $S \neq \mathbb{R}^n$, we have a *constrained optimization* problem.

2. When f depends on some random variables, the problem is a *stochastic optimization* problem (see [17, 87]). Otherwise, we have a *deterministic optimization* problem.

3. When the variables can only take on values from a discrete set, the problem is a *combinatorial optimization* problem. Some classic examples of this type is Traveling Saleman Problem, Maximum Matching, and Maximum Flow - Minimum Cut Problem (see [34]). In addition, when some of the variables are restricted to take integer values, we have a *mixed integer program*, whereas if all variables must be integers, it is said to be a *pure integer program*). Moreover, if the objective function and constraints are linear, we have a mixed (or pure) linear integer program. In this thesis, whenever we talk about mixed (or pure) integer program, unless otherwise noted, we always assume the objective function is linear.

4. When f is a convex function and S is a convex set, the problem is a *convex optimization* problem. With this special structure, problems of this type can be solved efficiently using interior-point methods (see [20]). The convexity assumptions are very powerful: they make the problems "easier" to solve. In fact, for unconstrained optimization problem, if f is a convex function, "line search" and "trust region" methods give better convergence rates. Moreover, if we impose that f be a linear function and S be defined by linear constraints, we have a well-understood class of problem - *linear programming* problem - that can be solved with Dantzig's simplex method (see [86]).

In this dissertation, we are particularly interested in mixed integer linear programming problem and stochastic optimization problems.

1.2 Statistics

Statistics can be defined as the science of developing and studying methods for collecting, summarizing, analyzing, and drawing conclusions from empirical data. Roughly speaking, statistics can be broken down into three main areas: Sampling methods, descripive statistics, and inferential statistics. The first area studies how best to design studies to collect data. The second is concerned with summarizing the data. Finally, the last area gives us tools to generalize beyond the data and make inference about the population from which the set of data is drawn from. Below, we give some examples of real-life problems where statistics

are useful.

1. A pharmaceutical company claims that its newly developed drug is more effective at treating certain disease than currently available drugs. Researchers at this company must carefully design clinical trials to test this hypothesis.

2. A quality control manager at a manufacturing plant that produces thousands of products every day need to conduct studies to make sure the products are up to company standards.

3. Government officials create a poll to estimate the proportion of residents that support a new regulation.

4. When people upload their photo on Facebook, their friends who appear on the photo are automatically tagged. This face detection feature is an application of deep learning, which is a method of statistical learning.

The last example is just one application of an ever growing area, which is statistical learning. In essence, statistical learning studies methods for modeling and understanding large and complex datasets. Even though statistical learning models have been around for some time, in the last decade, the popularity of these techniques has seen a very sharp rise with many different applications. With the advent of computers and improvement in computational power, vast amount of data are being generated in many different areas, resulting in more efficient and accurate learning of many learning models. As a result, statistical learning models have seen widespread applications in many areas, such as finance, healthcare, and entertainment.

Note 1.1. *The author is aware that there are subtle differences between "machine learning" and "statistical learning", where the former is concerned with algorithms to learn from data and the latter emphasizes statistical models underlying these algorithms. However, for convenience, in this disseratation we use both these two terms interchangeably to mean extract meaningful patterns and trends from the data, and often abbreviate them simply as* ***learning****.*

1.3 Contributions and Outline of the thesis

In the next chapter, we present preliminary background material used throughout the thesis. References are provided for any results presented without proofs. Our contributions are organized into Chapters 4-6. In particular, in Chapter 4, we apply machine learning techniques to improve the performance of cutting planes for mixed-integer linear programming problems. Chapter 5 discusses whether a commonly employed method to solve stochastic programming problems - the *Sample average approximation* method - is "good" to use in a statistical sense. Finally, in chapter 6, we design an algorithm to match large number of astronomical catalogs in crowded regions of the sky by formulating an integer linear program.

Chapter 2

Optimization Background

This chapter provides notation and some background material required for the research presented in this thesis. We start with a discussion on the complexity of problem classes. We then review convexity and present known results in convex analysis in Section 2.2. We cover mixed integer linear programming in Section 2.3, which provides important context for Chapter 4 and 6.

2.1 Complexity Classes

In Section 1.1, we claim that some problems are easier to solve than others. So what exactly does it mean to be an easy problem? In this section, we will give a rather informal introduction to computational complexity theory. Readers interested in a more formal coverage of the subject should see [2] and [50].

We define an "easy" problem as one that can be solved in an amount of time that is polynomial in the size of the input. This notion of problem complexity is often known as the "Cobham–Edmonds thesis" [50]. In particular, we denote the class of decision problems that can be solved in polynomial time as P. It is important to point out that there are many types of computational problems; however, we will just focus on decision problems - problems with a Yes or No answer - because they are easier to work with and that most other problems can be reduced to decision problems. Therefore, for a decision problem of class P, an yes or no answer can be decided in polynomial time.

For many problems, though, there are no known efficient algorithms to solve them in polynomial time. However, if someone gives us an answer together with a proof, we could verify if the solution is correct by checking the proof. We also do not want to take too long to check the proof and verify the answer: it should be done in a reasonable time. This gives rise to the notion of NP class, a class of decision problems for which a given yes solution has a so-called certificate, or proof, that can be verified in polynomial time. From the definition, it is clear that $P \subseteq NP$. However, it is still an open question whether the converse inclusion is true, or that $P = NP$.

At this point, some readers might be mistaken that this seems to imply NP problems are hard. We would like to emphasize that is not true. An NP problem just means it is easy to verify. In addition, there are many NP problems that are also P (since $P \subseteq NP$). So, how can we say that a problem is somewhat "difficult" to solve? To do that, we need another notion known as *NP-complete*. We say a problem is *NP-complete* if it is in NP and any problem in NP can be reduced to it in polynomial time. Thus, we could say that *NP-complete* problems are the hardest NP problems and that if we could solve these problems efficiently, we could solve all NP problems efficiently.

Finally, we comment on the complexity of several optimization problems of interest:

- Many convex programming problems are in P [79].
- Linear programming is in P [64].
- Mixed-integer linear programming is *NP-complete* [47].

2.2 Convex Analysis

As mentioned in Example 4 of Section 1.1, convexity is a strong assumption that could speed up and improve optimization methods. Here we formally define convex set and convex function.

Definition 2.1. (Convex Set). *A set $S \subseteq \mathbb{R}^d$ is called a* **convex set** *if for all $\mathbf{x}, \mathbf{y} \in S$ and every $\lambda \in [0,1], \lambda x + (1-\lambda)y \in S$. This means a line segment joining any two points in S will lie in S.*

Definition 2.2. (Convex Function). *A function* $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ *is called a* **convex function** *if*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y),$$

for all $x, y \in \mathbb{R}^d$ *and* $\lambda \in (0,1)$. *This means a line segment between any two points on the graph of the function lies above the graph.*

Examples of a convex set, nonconvex set, convex function, nonconvex function are shown in Figures 2.1-2.4 respectively.

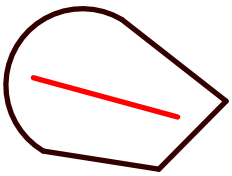

Figure 2.1: Example of a convex set

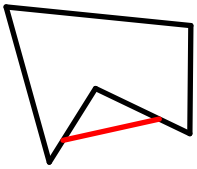

Figure 2.2: Example of a nonconvex set

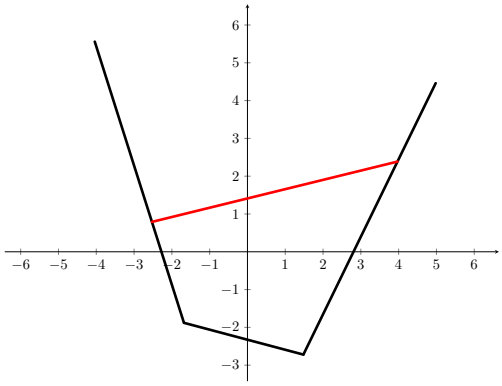

Figure 2.3: Example of a convex function

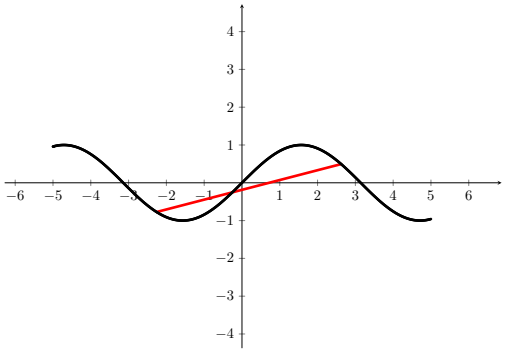

Figure 2.4: Example of a nonconvex function

These two definitions are the starting point for a rich field called *Convex Analysis*, which is the foundation for understanding convex optimization. Below, we define some basic concepts in convex analysis which are used throughout the thesis. We refer the reader interested in this subject to [94], [4, Chapter 2], or [83, Chapters 1-4] for a more comprehensive treatment.

Definition 2.3. [1]

[1] Thanks to the lecture notes of Dr. Amitabh Basu in his Introduction to Convexity class.

(i) Convex hull: Let $S \subseteq \mathbb{R}^d$. *The* convex hull *of* S *is* $\text{conv(S)} := \{\sum_{i=1}^{k} \lambda_i \mathbf{x}_i : \mathbf{x}_i \in S, 0 \leq \lambda \leq 1, \sum_{i=1}^{k} \lambda_i = 1\}$. *This also implies that* conv(S) *is the smallest, with respect to set inclusion, convex set containing* S. *See Figure 2.5 for an illustration.*

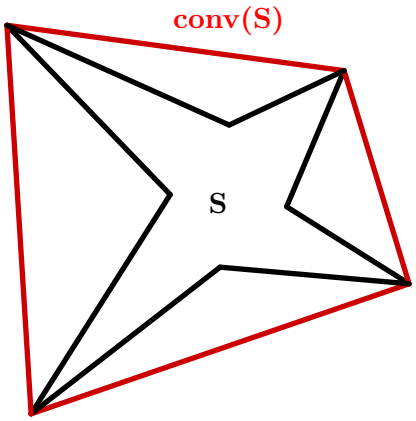

Figure 2.5: Example of the convex hull of a set S

(ii) Conical hull:Let $S \subseteq \mathbb{R}^d$. *The* conical hull *of* S *is* $\text{cone(S)} := \{\sum_{i=1}^{k} \lambda_i \mathbf{x}_i : \mathbf{x}_i \in S, \lambda \geq 0\}$. *See Figure 2.6 for an illustration.*

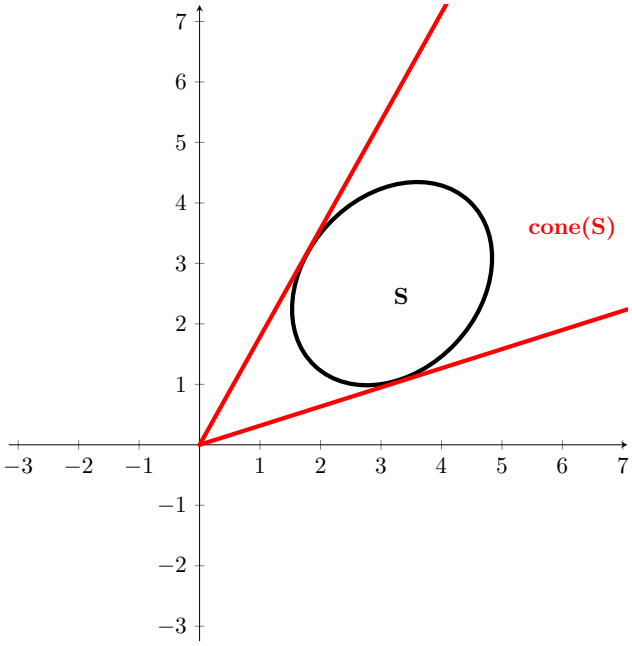

Figure 2.6: Example of the conical hull of a set S

(iii) Affine hull: Let $S \subseteq \mathbb{R}^d$. *The affine hull of* S *is* $\text{aff(S)} := \{\sum_{i=1}^{k} \lambda_i \mathbf{x}_i : \mathbf{x}_i \in S, \sum_{i=1}^{k} \lambda_i = 1\}$.

(iv) *Closure of a set S: The smallest (topologically) closed set containing S. Denoted by cl(S).*

(v) *Polar of a set X: Let $X \in \mathbb{R}^d$. The* polar *of X is defined as $X^\circ := \{\mathbf{y} \in \mathbb{R}^d : \langle \mathbf{x}, \mathbf{y} \rangle \leq 1,\ \forall \mathbf{x} \in X\}$*

(vi) *Relative interior of a convex set C: The set of all $\mathbf{x} \in C$ for which $\exists \varepsilon > 0$ such that $\forall \mathbf{y} \in$ aff(C), $\mathbf{x} + \varepsilon(\frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|}) \in C$. Denoted by relint(C).*

(vii) *Relative boundary of a convex set C: relbd(C) := cl(C) \ relint(C).*

(viii) *Dimension of an affine subspace $X \subseteq \mathbb{R}^d$: Let $\mathbf{x} \in X$. Then dim(X) is defined as the dimension of the linear subspace $X - \{\mathbf{x}\}$.*

(ix) *Dimension of a convex set C: Defined as the dimension of its affine hull: dim(C) = dim(aff(C))*

(x) *Normal cone of a convex set C: The* normal cone *of C at x is $N_C(x) := \{\mathbf{r} : \langle \mathbf{r}, \mathbf{x}-\mathbf{y} \rangle \geq 0,\ \forall \mathbf{y} \in C\}$.*

(xi) *Halfspace: A set of the form $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{a}, \mathbf{x} \rangle \leq b\}$ for some $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.*

(xii) *Polyhedron: An intersection of finitely many halfspaces. In other words, it is a set of the form $\{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \leq b\}$ for $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$. If $A \in \mathbb{Q}^{m \times d}$ and $b \in \mathbb{Q}^m$, we call it a rational polyhedron.*

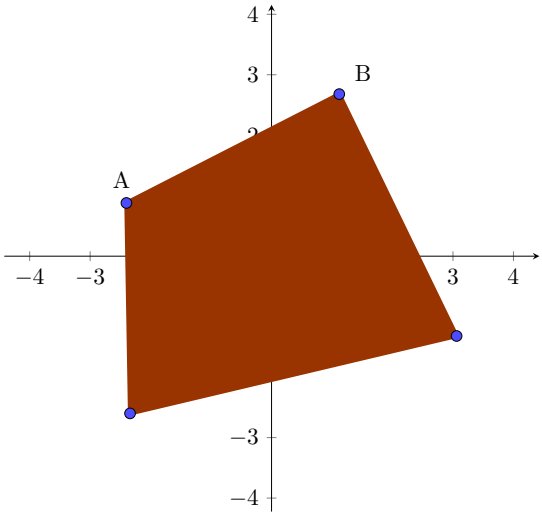

Figure 2.7: Example of a polyhedron

(xiii) Polytope: A polyhedron that is bounded.

(xiv) Face of a convex set: Given a convex set C, a convex subset $F \subseteq C$ is called a face *of C if for any $\mathbf{x} \in F$ and $\mathbf{x}^1, \mathbf{x}^2 \in C$, $\frac{\mathbf{x}^1+\mathbf{x}^2}{2} = \mathbf{x}$ implies that $\mathbf{x}^1, \mathbf{x}^2 \in F$. In addition, a face of dimension 0 is called an* extreme point *or a* vertex *of C. A face of dimension 1 is called an* edge*. For example, in Figure 2.7, A is a vertex of the polyhedron and the line segment AB is an edge.*

(xv) Exposed face: A face $F \subseteq C$ is called an exposed face *if $\exists \mathbf{a} \in \mathbb{R}^d$ and $\delta \in \mathbb{R}$ such that $C \subseteq \{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{a}, \mathbf{x} \rangle \leq \delta\}$ and $F = C \cap \{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{a}, \mathbf{x} \rangle = \delta\}$.*

Remark 2.1. *Besides the definition in (xiv), we could define the face of a convex set in 2 other ways:*

1. *A convex subset $F \subseteq C$ is called a* face *of C if for any $\mathbf{x} \in F$ and $\mathbf{x}^1, \mathbf{x}^2 \in C, \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 = \mathbf{x}$ implies that $\mathbf{x}^1, \mathbf{x}^2 \in F$.*

2. *A convex subset $F \subseteq C$ is called a* face *of C if any line segment in C with a relative interior point in F lies entirely in F, i.e.,*

$$\mathbf{x}^1, \mathbf{x}^2 \in C, \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \in F, 0 < \lambda < 1 \Rightarrow [\mathbf{x}^1, \mathbf{x}^2] \subset F.^2$$

We will now prove that these three definitions are equivalent.

Proof of Remark. In the order of their appearance, let's denote these three definitions as F1, F2, and F3 face.

1. Prove that F is an F1 face of C if and only if F is an F2 face of C:

   ($\Leftarrow$) Suppose F is an F2 face of C. Assume $\mathbf{x} \in F$ and $\mathbf{x}^1, \mathbf{x}^2 \in C$ such that $\mathbf{x} = \frac{1}{2}\mathbf{x}^1 + \frac{1}{2}\mathbf{x}^2$. Since F is an F2 face, this implies $\mathbf{x}^1, \mathbf{x}^2 \in F$ (because of definition of F2 face with $\lambda = \frac{1}{2}$). Hence, F is also an F1 face of C.

   ($\Rightarrow$) Suppose F is an F1 face of C. Assume $\mathbf{x} \in F, \mathbf{x}^1, \mathbf{x}^2 \in C$, and $\lambda \in (0,1)$ such that $\mathbf{x} = \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$. We need to show $\mathbf{x}^1, \mathbf{x}^2 \in F$. Without loss of generality, assume

[2] See Section 1.6.2 in [94].

$\lambda \in (0, \frac{1}{2}]$. Observe that if $\lambda = \frac{1}{2}$, we are done by the definition of F1 face. Now consider the line segment $\ell = [\mathbf{x}^1, \mathbf{x}^2] = \text{conv}(\mathbf{x}^1, \mathbf{x}^2)$ and the point $\mathbf{y} = 2\mathbf{x} - \mathbf{x}^2 = 2\lambda\mathbf{x}^1 + (1 - 2\lambda)\mathbf{x}^2$. Since $\lambda \in (0, \frac{1}{2})$, the coefficients of $\mathbf{x}^1$ and $\mathbf{x}^2$ are in $(0, 1)$ and they add up to 1. Hence, $\mathbf{y} \in \ell$ and thus, $\mathbf{y} \in C$. By rearranging the term, we also have $\mathbf{x} = \frac{\mathbf{y}+\mathbf{x}^2}{2}$ and by definition of F1 face, $\mathbf{y}, \mathbf{x}^2 \in F$. Now since $\mathbf{y} \in \ell$, we can write $\mathbf{y} = \lambda_1\mathbf{x}^1 + (1 - \lambda_1)\mathbf{x}^2$ for some $0 < \lambda_1 < 1$. If $\lambda_1 < \frac{1}{2}$, with a similar argument, we could consider a point $\mathbf{y}^1 = 2\mathbf{y} - \mathbf{x}^2$ and get $\mathbf{y}^1 \in \ell \cap F$. Again, we could then express $\mathbf{y}^1 = \lambda_2\mathbf{x}^1 + (1 - \lambda_2)\mathbf{x}^2$ and if $\lambda_2 < \frac{1}{2}$, we continue this construction until we find a point $\mathbf{y}^k \in \ell \cap F$ such that $\mathbf{y}^k = \lambda_{k+1}\mathbf{x}^1 + (1 - \lambda_{k+1})\mathbf{x}^2$ where $\lambda_{k+1} \geq \frac{1}{2}$. If $\lambda_{k+1} = \frac{1}{2}$, we get $\mathbf{x}^1 \in F$ by definition of F1 face because $\mathbf{y}^k \in F$. Otherwise, if $\lambda_{k+1} > \frac{1}{2}$, we could simply switch the roles of $\mathbf{x}^1, \mathbf{x}^2$ in the above argument to get $\mathbf{x}^1 \in F$. Hence, F is an F2 face of C.

2. Prove that F is an F2 face of C if and only if F is an F3 face of C:

   ($\Leftarrow$) Suppose F is an F3 face of C. Assume $\mathbf{x} \in F, \mathbf{x}^1, \mathbf{x}^2 \in C$, and $\lambda \in (0, 1)$ such that $\mathbf{x} = \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$. We need to show $\mathbf{x}^1, \mathbf{x}^2 \in F$. This is quite trivial, since from the definition of F3 face, the line segment $[\mathbf{x}^1, \mathbf{x}^2] \subset F$, and in particular, $\mathbf{x}^1, \mathbf{x}^2 \in F$.

   ($\Rightarrow$) Suppose F is an F2 face of C. Let $\mathbf{x}^1, \mathbf{x}^2 \in C$ and $\mathbf{x} = \lambda\mathbf{x}^1 + (1 - \lambda)\mathbf{x}^2 \in F$ for some $\lambda \in (0, 1)$. We need to show that $[\mathbf{x}^1, \mathbf{x}^2] \subset F$. This follows from the fact that by the definition of F2 face, $\mathbf{x}^1, \mathbf{x}^2 \in F$, and because F is convex, the line segment $[\mathbf{x}^1, \mathbf{x}^2]$ is also in F.

$\diamond$

Definition 2.4. (The smallest face containing a point). *Observe that for any point* $\mathbf{x}$ *in a convex set* C*, there is always at least one face containing* $\mathbf{x}$*, namely* C*. It is a simple exercise to verify by definition that the intersection of any two faces is also a face. Consequently, it is natural to define the smallest face, with respect to a convex set* C*, containing a point* $\mathbf{x}$ *as the intersection of all faces of* C *containing* $\mathbf{x}$*. This smallest face will be denoted by* $F_{\mathbf{x}}$*. Formally,* $F_{\mathbf{x}} = \bigcap(F : F \text{ is a face of } C, \mathbf{x} \in F)$.

Proposition 2.1. *$F_{\mathbf{x}}$ is the union of $\mathbf{x}$ and all $\mathbf{y} \in C$ such that there is a line segment $[\mathbf{y}, \mathbf{z}] \subset C$ with $\mathbf{x}$ as a relative interior point.*

Proof. Let's denote the aforementioned union as F. We will prove that $F = F_{\mathbf{x}}$ by showing each one is a subset of the other.

- Prove $F \subseteq F_{\mathbf{x}}$: Let $\mathbf{x}^1 \in F$, thus $\mathbf{x}^1 \in C$. From the definition of F, there exists a $\mathbf{x}^2 \in C$ such that $\mathbf{x} = \lambda \mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$ and $[\mathbf{x}^1, \mathbf{x}^2] \subset F$. But then since $\mathbf{x} \in F_{\mathbf{x}}$ and $F_{\mathbf{x}}$ is a face of C, $\mathbf{x}^1, \mathbf{x}^2 \in F_{\mathbf{x}}$ by using the F3 definition of face above (see Remark 2.1). Hence, $F \subseteq F_{\mathbf{x}}$.

- Prove $F_{\mathbf{x}} \subseteq F$: Since $F_{\mathbf{x}}$ is the smallest face containing $\mathbf{x}$, it suffices to prove that F is a face of C. By appropriate translation, we could assume that $\mathbf{x} = 0$. First, we will prove that F is convex. Let $\mathbf{x}^1, \mathbf{x}^2 \in F$. By definition of F and that $\mathbf{x} = 0$, we have $\mathbf{x}^1 = -\lambda_1 \mathbf{y}^1$ and $\mathbf{x}^2 = -\lambda_2 \mathbf{y}^2$ for some $\mathbf{y}^1, \mathbf{y}^2 \in C, \lambda_1, \lambda_2 > 0$. Let $\mathbf{z} = \alpha \mathbf{x}^1 + (1-\alpha)\mathbf{x}^2 = -\alpha\lambda_1 \mathbf{y}^1 - (1-\alpha)\lambda_2 \mathbf{y}^2$. We want to prove there exists a point $\mathbf{z}' \in C$ such that $\mathbf{z} = -\lambda' \mathbf{z}'$ for some $\lambda' > 0$. To see this, we could define

$$\mathbf{z}' = \frac{-1}{\alpha\lambda_1 + (1-\alpha)\lambda_2}\mathbf{z} = \frac{\alpha\lambda_1}{\alpha\lambda_1 + (1-\alpha)\lambda_2}\mathbf{y}^1 + \frac{(1-\alpha)\lambda_2}{\alpha\lambda_1 + (1-\alpha)\lambda_2}\mathbf{y}^2 \in C,$$

  thus $\mathbf{z} \in C$. Finally, we need to show that F is a face. Let $\mathbf{x}^1, \mathbf{x}^2 \in C, \mathbf{y} = \lambda \mathbf{x}^1 + (1-\lambda)\mathbf{x}^2 \in F$. Since $\mathbf{y} \in F, \exists \mathbf{z} \in C$ such that $\mathbf{y} = -\alpha \mathbf{z}$. Rearranging terms, we have

$$\mathbf{x}^1 = \frac{1}{\lambda}\mathbf{y} - \frac{1-\lambda}{\lambda}\mathbf{x}^2 = -\frac{\alpha}{\lambda}\mathbf{z} - \frac{1-\lambda}{\lambda}\mathbf{x}^2.$$

  Observe that

$$-\frac{\lambda}{\alpha + (1-\lambda)}\mathbf{x}^1 = \frac{\alpha}{\alpha + (1-\lambda)}\mathbf{z} + \frac{1-\lambda}{\alpha + (1-\lambda)}\mathbf{x}^2 \in C.$$

  Hence, $\mathbf{x}^1 \in F$. Repeating the same argument, we obtain $\mathbf{x}^2 \in F$, thus $[\mathbf{x}^1, \mathbf{x}^2] \subset F$. Hence, F is a face and $F_{\mathbf{x}} \subseteq F$.

□

Remark 2.2. *Proposition 2.1 implies that given a convex set C and a point $\mathbf{x} \in C$, F is the smallest face containing $\mathbf{x}$ if $\mathbf{x} \in$ relint(F).*

Proposition 2.2. *relint(C) is nonempty for any nonempty convex set $C \subseteq \mathbb{R}^d$.*

Proof. Let $\dim(\text{aff}(C)) = k-1$, so there exists k affinely independent points $\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^k \in C$. We will show that $\mathbf{x} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{x}^i \in \text{relint}(C)$. Let $\mathbf{y} \in$ aff(C). Then $\mathbf{y} = \sum_{i=1}^{k} \lambda_i \mathbf{x}^i$, where $\sum_{i=1}^{k} \lambda_i = 1$. Thus,

$$\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}) = (1-\alpha)\mathbf{x} + \alpha\mathbf{y} = \sum_{i=1}^{k}(\frac{1-\alpha}{k} + \alpha\lambda_i)\mathbf{x}^i.$$

Observe that the sum of the coefficients is 1. Also, for α sufficiently small, these coefficients are positive. Hence, $\mathbf{x} + \alpha(\mathbf{y} - \mathbf{x}) \in C$, i.e. $\mathbf{x} \in \text{relint}(C)$. □

We are now ready to state an important theorem regarding faces of a polyhedron.

Theorem 2.1. (Faces of polyhedra). *Given a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^d : A\mathbf{x} \leq \mathbf{b}\}$ with $A \in \mathbb{R}^{m\times d}, \mathbf{b} \in \mathbb{R}^m$, $F \subseteq P$ such that $F \neq \emptyset, P$, then F is a* face *of P if and only if there exists a subset $I \in \{1, \cdots, m\}$ such that $F = \{\mathbf{x} \in P : A_I\mathbf{x} = \mathbf{b}_I\}$.*

Proof. ($\Leftarrow$) Let $F = \{\mathbf{x} \in P : A_I\mathbf{x} = \mathbf{b}_I\}$ for some subset $I \in \{1, \cdots, m\}$. Let $\mathbf{x}^1, \mathbf{x}^2 \in P$ and $\mathbf{x} \in F$ such that $\mathbf{x} = \frac{\mathbf{x}^1+\mathbf{x}^2}{2}$. For $i \in I$, we have

$$\mathbf{b}_i = \langle \mathbf{a}_i, \mathbf{x}\rangle = \frac{\langle \mathbf{a}_i, \mathbf{x}^1\rangle}{2} + \frac{\langle \mathbf{a}_i, \mathbf{x}^2\rangle}{2} \leq \frac{\mathbf{b}_i + \mathbf{b}_i}{2} = \mathbf{b}_i.$$

Thus, the inequality must be an equality and so $\langle \mathbf{a}_i, \mathbf{x}^1\rangle = \langle \mathbf{a}_i, \mathbf{x}^2\rangle = \mathbf{b}_i$ for all $i \in I$. Hence, $\mathbf{x}^1, \mathbf{x}^2 \in F$, and F is a face.

($\Rightarrow$) Let F be a face of P. By Proposition 2.2, since $F \neq \emptyset, \exists \mathbf{x}^* \in$ relint(F). From Remark 2.2, we have $F = F_{\mathbf{x}^*}$. Let's define $F' = \{\mathbf{x} \in P : A_I\mathbf{x} = \mathbf{b}_I\}$, for $I = \{i : \langle \mathbf{a}^i, \mathbf{x}^*\rangle = \mathbf{b}^i\}$. As shown above, F' is a face containing $\mathbf{x}^*$, thus $F \subseteq F'$ by definition of $F_{\mathbf{x}^*}$.

Now, let $\mathbf{x} \in F'$ and consider $\mathbf{y} = \mathbf{x}^* + \lambda(\mathbf{x}^* - \mathbf{x})$. For $\forall i \in I, \langle \mathbf{a}^i, \mathbf{x}^*\rangle = \langle \mathbf{a}^i, \mathbf{x}\rangle = \mathbf{b}^i$, thus $\langle \mathbf{a}^i, \mathbf{y}\rangle = \mathbf{b}^i$. On the other hand, for $\forall i \notin I, \langle \mathbf{a}^i, \mathbf{x}^*\rangle < \mathbf{b}^i$; hence, we could choose $\lambda > 0$ small enough so that $\langle \mathbf{a}^i, \mathbf{y}\rangle \leq \mathbf{b}^i$. This means $\mathbf{y} \in F'$ and $[\mathbf{x}, \mathbf{y}] \subset F' \subset P$. Since F is a

face of P containing $\mathbf{x}^*$, which in turn is an interior point of $[\mathbf{x}, \mathbf{y}], [\mathbf{x}, \mathbf{y}] \in F$. In particular, $\mathbf{x} \in F$ and so $F' \subseteq F$. This means $F = \{\mathbf{x} \in P : A_I \mathbf{x} = \mathbf{b}_I\}$. □

2.3 Mixed-integer Linear Programming

Although convex optimization is a very powerful tool, many interesting real-world problems cannot be formulated as a convex programming problem. This section discusses a well-studied area of nonconvex optimization known as mixed-integer linear programming. It has widespread applications in many different fields such as finance [78],[10], renewable energy [74], supply chain management [16], scheduling [99], and airline logistics [89].

This section starts with a discussion of linear programming, which plays an important role in solving a mixed-integer linear programming problem. We then review the developments of methods in mixed-integer linear programming.

2.3.1 Linear Programming

A linear program (LP) refers to the problem of optimizing a linear objective function subject to linear equality or inequalities constraints on the variables. A linear program is expressed in standard form as,

$$\begin{aligned} \min \quad & c^\top x \\ \text{subject to:} \quad & \\ & Ax = b \\ & x \geq 0, \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ are given.

Remark 2.3. *In many places, one can see a linear program of the following form:*

$$\begin{aligned} \textit{min} \quad & c^\top x \\ \textit{subject to:} \quad & \\ & Ax \leq b \\ & x \geq 0. \end{aligned}$$

Note that an inequality with a " $\geq$ *" sign might be converted to the form above simply by multiplying both sides by* -1*. Also, a maximization problem of the form* $\max c^\top x$ *could be turned into a minimization problem* $\min -c^\top x$*. Finally, it turns out that 2.1 is equivalent to the standard form as we could introduce a slack variable* s *and rewrite the formulation as,*

$$\begin{aligned} \textit{min} \quad & c^\top x \\ \textit{subject to:} \quad & \\ & Ax + s = b \\ & x \geq 0, s \geq 0, \end{aligned}$$

and we are back to the standard form.

On the other hand, we could also convert a LP formulation in standard form to 2.1 by expressing the equality as a pair of inequalities:

$$\begin{aligned} \textit{min} \quad & c^\top x \\ \textit{subject to:} \quad & \\ & Ax \leq b \\ & -Ax \leq -b \\ & x \geq 0. \end{aligned}$$

Remark 2.4. *In these LP formulations, there might be no restrictions of the sign of the variables. For example, if there exists a variable* x_i *that is not bounded, we could replace it with* $x_i^+ - x_i^-$ *and add the constraints* $x_i^+, x_i^- \geq 0$ *to get back to the standard form.*

The most commonly used method to solve a linear programming problem is the simplex method, which was invented by Dantzig [35]. The basic idea of the simplex method is the following:

- Start at some vertex of the polyhedron defined by the constraints $\{Ax = b, x \geq 0\}$.
- While there exists a neighboring vertex with lower value of $c^\top x$, move to it. If no such vertex exists, stop.

We now explain in more detail how this works. First, let's assume that $\text{rank}(A) = m$, so it is of full row rank. Otherwise, there are redundant constraints that can be removed to make the matrix full row rank. This also implies that $n \geq m$. Hence, we could pick a set of indices $B \in [n]$ called *basis* corresponding to m linearly independent columns of the matrix A and define $N = [n] \backslash B$. We will now use the notations A_B, A_N to denote the submatrices formed by taking the columns of A indexed by B and N respectively. Similarly, x_B, x_N are the subvectors of x indexed by B and N. The variables corresponding to index set B, i.e. those variables in x_B, are called *basic variables.* The others are referred to as *nonbasic variables.* We could now rewrite the matrix constraint as $A_B x_B + A_N x_N = b$. Observe that since A_B is invertible by construction, setting $x_N = 0$ gives us $x_B = A_B^{-1} b$. The vector x defined that way is referred to as *a basic solution.* If x is also feasible (i.e. $x_B \geq 0$), it is called a *basic feasible solution* (BFS).

Claim 2.1. *There always exists a BFS for a feasible LP formulation.*

Proof of Claim. Let $\overline{x} \geq 0$ be a feasible point. Define the set of index $I = \{i : \overline{x}_i > 0\}$. We consider two cases:

Case 1. The columns of A_I are linearly independent. If $|I| < m$, we could extend it to a basis B with $|B| = m$ by finding columns in $A_{[n]\backslash I}$ that are linearly indepedent to the columns of A_I. Afterwards, we get the matrix A_B whose columns are linearly independent. Let $N = [n] \backslash B$. Since $I \subseteq B, \overline{x}_N = 0$. In addition, we have $b = A\overline{x} = A_B \overline{x}_B + A_N \overline{x}_N = A_B \overline{x}_B$. Thus, $\overline{x}_B = A_B^{-1} b$, i.e. $\overline{x}$ is a basic solution. Since it is feasible by assumption, it is a BFS.

Case 2. The columns of A_I are linearly dependent. Then exists $u_I \neq 0$ such that $A_I u_I = 0$. Define $N = [n] \backslash I$ and $u_N = 0$. Then we have $u = [u_I, u_N] \in \mathbb{R}^n$. For $\lambda \in \mathbb{R}$, $A(\overline{x} + \lambda u) = A\overline{x} + \lambda A_N u_N + \lambda A_I u_I = b + 0 + 0 = b$. Also, since $\overline{x}_N = u_N = 0, (\overline{x} + \lambda u)_N = 0$ and so $\overline{x} + \lambda u$ has at least as many zero entries as $\overline{x}$. Also, since $\overline{x}_I > 0$, we could choose λ small enough so that $(\overline{x} \pm \lambda u)_I \geq 0$, thus both $\overline{x} + \lambda u$ and $\overline{x} - \lambda u$ are feasible. In particular, we could choose $\lambda^* > 0$ such that either $(\overline{x} + \lambda^* u)_i = 0$ or $(\overline{x} - \lambda^* u)_i = 0$ for some $i \in I$. Hence, one of $\overline{x} \pm \lambda^* u$ has one more zero entry than $\overline{x}$ does. Without loss of generality, assume it is $\overline{x} + \lambda^* u$. Now, consider $\overline{x} + \lambda^* u$ as our new feasible point. If it falls into Case 1,

we are done. Otherwise, repeat the process in Case 2. Since the number of nonzero entries in the new feasible point, i.e. the set I, decreases by one through every iteration, eventually we will have a feasible point that falls into Case 1, where the columns of A_I are linearly independent. $\diamond$

From Claim 2.1, we can assume that we have a BFS to start with. Let B be the basis associated with this BFS. Let's rewrite the LP as follows,

$$\begin{aligned} \min \quad & c_B^\top x_B + c_N^\top x_N \\ \text{s.t.} \quad & A_B x_B + A_N x_N = b \\ & x_B, x_N \geq 0 \end{aligned}$$

Hence, for any solution x, $x_B = A_B^{-1}b - A_B^{-1}A_N x_N$ and the objective function, $c^\top x$ can be expressed as follows:

$$\begin{aligned} c^\top x &= c_B^\top x_B + c_N^\top x_N \\ &= c_B^\top (A_B^{-1}b - A_B^{-1}A_N x_N) + c_N^\top x_N \\ &= c_B^\top A_B^{-1} b + (c_N^\top - c_B^\top A_B^{-1} A_N) x_N \end{aligned}$$

For ease of exposition, define $\bar{c}_N = (c_N^\top - c_B^\top A_B^{-1} A_N) x_N$. It is clear that if there exists a $i \in N$ such that $\bar{c}_N < 0$, we can decrease the objective by increasing x_i from 0. However, we cannot increase it arbitrarily. As x_B depends on x_N, we need to make sure that $x_B \geq 0$. This is essentially one iteration of the Simplex method: First, find a $i \in N$ such that $\bar{c}_i < 0$. We then increase x_i as much as possible while keeping the components of x_B nonnegative. We will stop increasing x_i when one of the x_B reaches 0. At this point, a basic variable becomes 0 and is removed from the basis. On the other hand, the previously nonbasic variable x_i is now positive and is included in the basis. We repeat this procedure, until there is no $i \in N$ such that $\bar{c}_i < 0$. At this point, we have reached the optimal solution since from above, we now have the objective value $c^\top x = c_B^\top A_B^{-1} b + (c_N^\top - c_B^\top A_B^{-1} A_N) x_N \geq c_B^\top A_B^{-1} b$.

Remark 2.5. *From the description of the Simplex method, it is clear that we have a decision*

to make at every iteration: What variable to enter the basis and what variable to be removed from the basis. A systematic approach to make that decision is called a "pivoting rule". If we are not careful, we might be stuck in an infinite loop, or "cycling" [48]. There are many pivot rules, such as Dantzig"s rule, Bland's rule, steepest edge rule, random edge rule and greatest descent rule (see [92] for a survey of pivot rules). Among these, Bland's rule prevents cycling [18]. However, in the worse case, all known pivot rules require an exponential number of iterations. For example, in 1972, Klee and Minty [66] gave an example, the Klee-Minty Cube, showing that the Dantzig's rule has exponential time complexity. However, as pointed out earlier, in 1979, Khachiyan [64] came up with a polynomial time algorithm to solve linear programming problems. That being said, the Simplex method performs remarkably well in practice and is still commonly used in many modern LP solvers.

2.3.2 Mixed-integer Linear Programming

A mixed-integer linear program (MILP) has the standard form

$$\begin{aligned} \min \quad & c^\top x \\ \text{subject to:} \quad & \\ & Ax = b \\ & x \geq 0 \\ & x_i \in \mathbb{Z},\ \forall i \in I, \end{aligned} \tag{2.2}$$

where $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, I \subseteq [n]$ are given.

In the special case where $I = [n]$, we have a *pure integer linear program (PILP)*.

In the formulation above, we could be more specific by separating the integer variables and the real-valued variables as follows,

$$\begin{aligned} \min \quad & c^\top x + h^\top y \\ \text{subject to:} \quad & \\ & Ax + Gy = b \\ & x \geq 0 \ \text{ integral} \\ & y \geq 0, \end{aligned} \tag{2.3}$$

where $A \in \mathbb{Q}^{m\times n}, G \in \mathbb{Q}^{m\times p}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, h \in \mathbb{Q}^p$.

In this thesis, we will focus on MILP problem of this form.

Remark 2.6. *The feasible set*

$$S = \{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : Ax + Gy = b\} \tag{2.4}$$

is often called a ***mixed integer linear set****.*

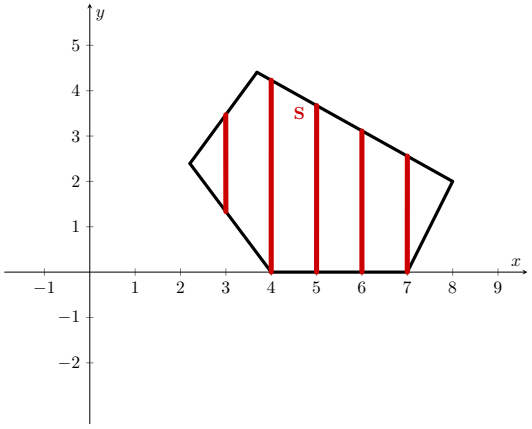

Figure 2.8: Example of a mixed integer linear set

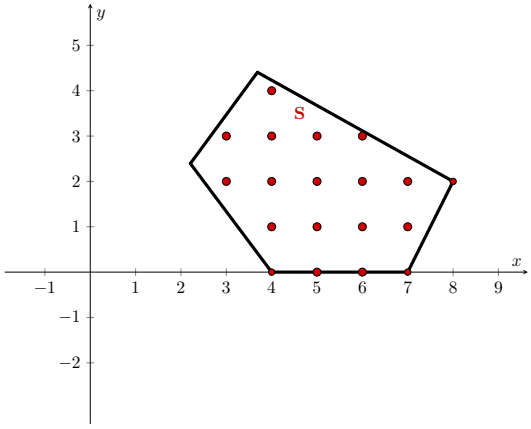

Figure 2.9: Example of a pure integer linear set

Remark 2.7. *Similar to the LP formulation, there are several variations to the MILP formulation that are all equivalent to each other, such as*

- *maximizing, instead of minimizing, the objective function,*
- *the matrix constraint is an inequality, i.e.* $Ax + Gy \leq b$*,*
- *the variables are not required to be non-negative.*

We now discuss techniques to solve a MILP problem. First, we should point out that unlike an LP problem where one can easily find the optimal solution at one of the feasible set's vertices, this is not generally true for MILP, as the feasible set S is not even convex and is hard to formulate. Hence, even though it is true that the optimal solution is found at an extreme point of the convex hull of all feasible points in S, it is not clear how we could compute this convex hull. An idea is to approximate this convex hull by sets that we

know how to compute. In particular, we introduce the notion of a natural *linear relaxation* of the set S.

Definition 2.5. (Linear relaxation of a mixed integer linear set). *For a mixed integer linear set S given by Equation 2.4, a natural linear relaxation is obtained by relaxing the integrality constraint on x, i.e.,*

$$P := \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^p_+ : Ax + Gy = b\}. \tag{2.5}$$

The corresponding linear programming relaxation of MILP is $\min\{c^\top x + h^\top y : (x, y) \in P\}$.

Remark 2.8. *It is generally not true that the optimal solution of 2.3 is a vertex of the polyhedron P as it might not satisfy the integrality constraint. However, what we can say is, since $S \subseteq P$, the optimal value of MILP is no better than the optimal value of its linear relaxation. This means the optimal value to the LP is a lower bound on the optimal value of MILP.*

For ease of exposition, let's assume that the solution to our MILP is finite, with optimal value z^* and optimal solution (x^*, y^*). As described above, to solve MILP, we start by solving its LP relaxation. Assume the optimal solution to the LP relaxation is (x^0, y^0) with objective value z^0. From Remark 2.8, we know that $z^0 \leq z^*$. Hence, it is clear that if $x^0 \in \mathbb{Z}^n$, we are done and (x^0, y^0) will be the solution to MILP. Now, we describe three approaches to deal with the case when $x^0_i \notin \mathbb{Z}$ for some $i \in [n]$.

Branch and Bound algorithm

The main idea of the branch and bound algorithm is that at each node $N_i \in \mathcal{L}$ corresponding to a MILP with feasible set S_i, we can solve its LP relaxation and see if the solution satisfies the integrality constraint. If it does, we have an upper bound on the objective value (the objective value of the original MILP can not be worse than this). If it does not, we break the feasible set S_i into two smaller sets S_{i_1}, S_{i_2} by adding one of the constraints $x_j \leq \left\lfloor x^i_j \right\rfloor$ and $x_j \geq \left\lceil x^i_j \right\rceil$ to each set. Observe that (S_{i_1}, S_{i_2}) is a partition of S_i

Algorithm 1 Branch and Bound Algorithm

Input: Given $A \in \mathbb{Q}^{m\times n}, G \in \mathbb{Q}^{m\times p}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, h \in \mathbb{Q}^p, I \subseteq [n]$, minimize $c^\top x + h^\top y$ over the mixed integer set S as defined in 2.4. Initialize $\mathcal{L} = \{N_0\}, \underline{z} = \infty, (x^*, y^*) = \emptyset$.

1: **while** $\mathcal{L} \neq \emptyset$ **do**
2: Choose a node $N_i \in \mathcal{L}$.
3: Solve LP_i associated with N_i. If it is infeasible, remove node N_i from $\mathcal{L}$ and go to line 2. Else, let (x^i, y^i) and z^i be the optimal solution and optimal value of LP_i.
4: **if** $z_i \geq \underline{z}$ **then** remove node N_i from $\mathcal{L}$ and go to line 2.
5: **end if**
6: **if** x^i is integral **then**, set $(x^*, y^*) = (x^i, y^i)$ and $\underline{z} = z_i$. Remove node N_i from $\mathcal{L}$ and go to line 2
7: **else**
8: Find an index $j \in [n]$ such that $x^i_j \notin \mathbb{Z}$.
9: Define two new feasible set S_{i_1}, S_{i_2} that are similar to S^i, except that in S^{i_1}, there is an additional constraint $x_j \leq \left\lfloor x^i_j \right\rfloor$ and in S^{i_2}, there is an additional constraint $x_j \geq \left\lceil x^i_j \right\rceil$.
10: Add these two mixed integer sets to $\mathcal{L}$ as nodes N_{i_1} and N_{i_2}, remove node N_i from $\mathcal{L}$, and go back to line 2.
11: **end if**
12: **end while**

and so we did not discard any feasible solution in N_i. The advantage is $S_{i_1} \cup S_{i_2}$ is a better approximation to $\text{conv}(S_i)$ and so we get closer to the optimal value of N_i. Also, note that if the optimal value to the LP relaxation of a node N_i is at least as large as the current upper bound of optimal value to original MILP (line 4), we do not have to explore further as any solution from this node will be no better than what we currently have.

By repeating this process of either branching (breaking into two sub-MILPs) or leaving it as it is, we can generate a search tree of MILP nodes. The leaf nodes of the search tree have linear programming relaxations whose optimal solution either satisfies the integrality constraints, or is worse than the value of some other leaf node. The best of the integral leaf node solutions is selected as the global optimal solution.

Remark 2.9. *There are several important decisions to make in the branch and bound algorithm, such as heuristics for a good upper bound $\underline{z}$, or what variables to branch on. Discussions on these issues are provided in [33, Chapter 9].*

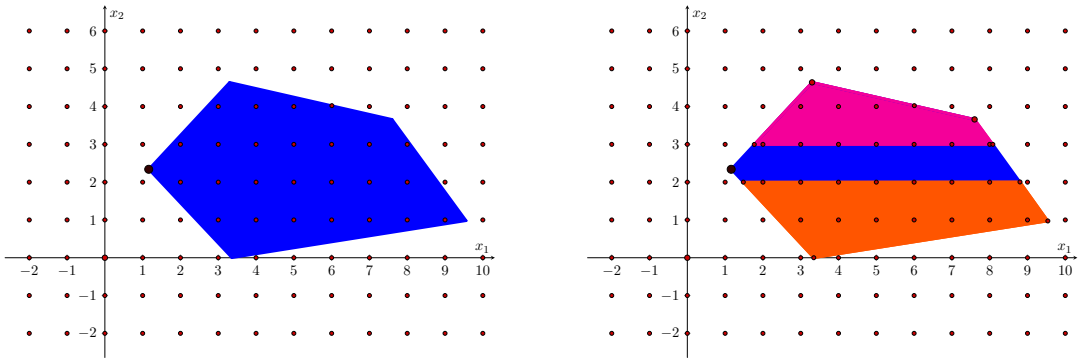

Figure 2.10: Let P be the blue shaded polyhedron and $S = P \cap \mathbb{Z}^2$. We have a MILP: $\min_{x \in S} x_1$. Since the LP relaxation's solution, shown as the black dot, does not satisfy the integrality constraint, we do branching on x_2. Two new mixed integer linear sets, S_1 and S_2, are created by adding the constraint $x_2 \leq 2$ and $x_2 \geq 3$ to P respectively.

Cutting plane algorithm

To understand the idea behind the cutting plane method, we first give the following definition of a valid inequality.

Definition 2.6 (Valid inequality)**.** *Let $S \in \mathbb{R}^n$ and let $a \in \mathbb{R}^n, \delta \in R$. We say that $\langle a, x \rangle \leq \delta$ is a* valid inequality *for S if $S \subseteq \{x : \langle a, x \rangle \leq \delta.\}$*

The cutting plane method for MILP can now be defined as the process of iteratively generating valid inequalities for its mixed integer linear set S, usually by separating the solution to the LP relaxation from S. In doing so, we are removing the "redundant" region of the LP relaxation's feasible set and getting a better approximation to $\text{conv}(S)$.

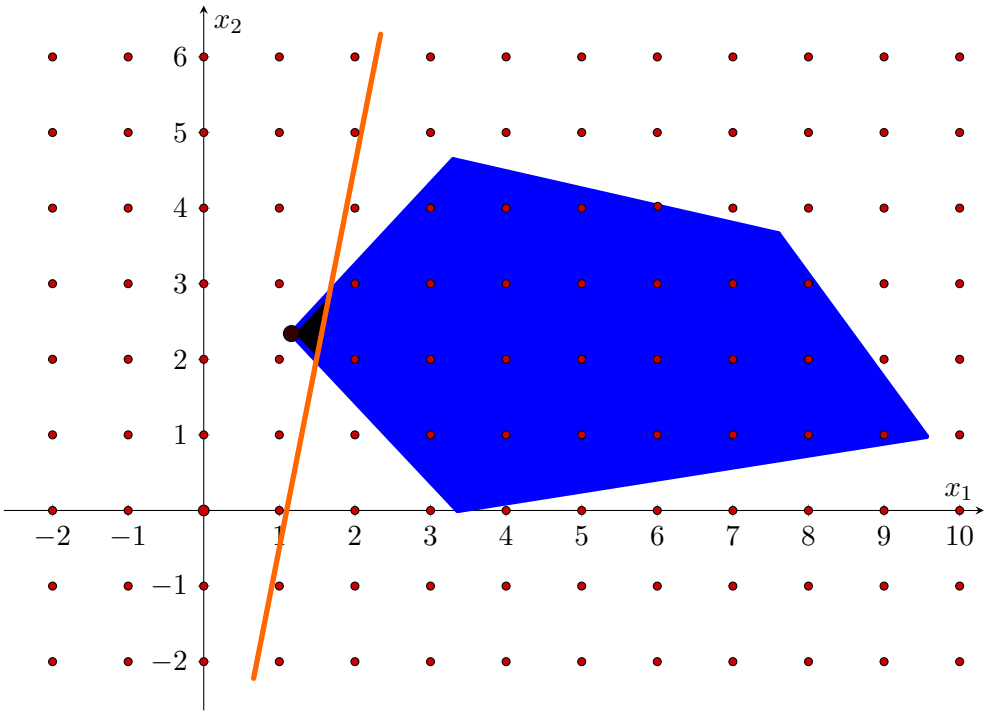

Figure 2.11: Let P be the shaded polyhedron and $S = P \cap \mathbb{Z}^2$. We have a MILP: $\min_{x \in S} x_1$. The cutting plane which cuts off the LP relaxation's solution, represented by the black dot, is plotted in orange. Observe that this cutting plane did not remove any feasible point of S.

Algorithm 2 Cutting Plane Algorithm

Input: Given $A \in \mathbb{Q}^{m \times n}, G \in \mathbb{Q}^{m \times p}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, h \in \mathbb{Q}^p, I \subseteq [n]$, minimize $c^\top x + h^\top y$ over the mixed integer set S as defined in 2.4. Initialize LP_0 with the LP relaxation of the original MILP problem and (x', y') with its optimal solution.

1: **while** $x' \notin \mathbb{Z}^n$ **do**

2: Add a cutting plane to LP_i to get a new LP, LP_{i+1}. Solve the new LP to get a solution (x^{i+1}, y^{i+1}). Set $(x', y') = (x^{i+1}, y^{i+1})$.

3: **end while**

Remark 2.10. *There are a couple of issues left untouched in the algorithm above. For example,*

- *There are infinitely many cutting planes that cut off (x^i, y^i) from S. To decide on which cutting plane to use is an important question in mixed integer linear programming. Usually, there is a tradeoff between the time to generate a cutting plane and the effectiveness of the cutting plane.*

- *Instead of using just a single cutting plane, we could generate many and use them all*

to better approximate conv(S.)

Conforti et al. [33] study these questions in Chapters 5 - 7.

Branch and cut algorithm

Finally, we discuss the branch and cut algorithm, which is a hybrid version of the two previous algorithms. This method is used exclusively in almost of commercial software for solving mixed integer linear programs. As can be seen in the algorithm below, besides the choices mentioned in Remark 2.9 and 2.10, during the branch and cut algorithm, we have to make another important decision, which is whether to "cut" or to "branch", as shown in line 8 of Algorithm 3.

Algorithm 3 Branch and Cut Algorithm

Input: Given $A \in \mathbb{Q}^{m\times n}, G \in \mathbb{Q}^{m\times p}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, h \in \mathbb{Q}^p, I \subseteq [n]$, minimize $c^\top x + h^\top y$ over the mixed integer set S as defined in 2.4. Initialize $\mathcal{L} = \{N_0\}, \underline{z} = \infty, (x^*, y^*) = \emptyset$.

1: **while** $\mathcal{L} \neq \emptyset$ **do**
2: Choose a node $N_i \in \mathcal{L}$.
3: Solve LP_i associated with N_i. If it is infeasible, remove N_i from $\mathcal{L}$ and go to line 2. Else, let (x^i, y^i) and z^i be the optimal solution and optimal value of LP_i.
4: **if** $z_i \geq \underline{z}$ **then** , remove N_i from $\mathcal{L}$ and go to line 2.
5: **end if**
6: **if** x^i is integral **then**, set $(x^*, y^*) = (x^i, y^i)$ and $\underline{z} = z_i$. Remove N_i from $\mathcal{L}$ and go to line 2.
7: **end if**
8: Method $\leftarrow$ "Branch" or "Cut".
9: **if** Method $\leftarrow$ "Branch" **then**
10: Find an index $j \in [n]$ such that $x^i_j \notin \mathbb{Z}$.
11: Define two new feasible set S_{i_1}, S_{i_2} that are similar to S^i, except that in S^{i_1}, there is an additional constraint $x_j \leq \left\lfloor x^i_j \right\rfloor$ and in S^{i_2}, there is an additional constraint $x_j \geq \left\lceil x^i_j \right\rceil$.
12: Add these two mixed integer sets to $\mathcal{L}$ as nodes N_{i_1} and N_{i_2}, remove N_i from $\mathcal{L}$, and go back to line 2.
13: **else**
14: Add a cutting plane, which separates the solution (x^i, y^i) from S, to the node N_i. Then go to line 3.
15: **end if**
16: **end while**

In this thesis, we will not study the branching aspect. Instead, we will focus on cutting plane procedures, which has been an active area of research since the 1970s. In particular,

our main contribution is that whenever a decision needs to be made within the branch-and-cut algorithm whether to add a cut, we can help to decide if it is worth it or not to use a particular family of cutting planes, and if so which cuts from this family to add.

2.4 Cut Generating Functions

The cutting plane method was first introduced by Gomory. In [51], he came up with the so-called fractional cuts to solve a pure integer programming problem; later, in 1960, he proposed the mixed integer inequalities to solve a general mixed integer linear programming problem [53]. While Gomory's fractional cuts for solving pure integer program requires the addition of slack variables to convert the problem into standard form, Chvátal [32] found another method to work with feasible set of the form $Ax \leq b$. It turns out that these two cuts are equivalent; hence, they are commonly known as Chvátal-Gomory cuts. Since then, many other types of cutting planes have been introduced in the literature, such as split inequalities, lift-and-project inequalities, clique inequalities, and cover inequalities. A comprehensive survey of different types of cutting planes can be found in [33, Chapters 5-6].

A large class of cutting planes can be described in an unifying framework with the introduction of cut generating functions. We first start with the definition of a mixed-integer set.

2.4.1 Mixed-integer set

Recall from Section 2.3 that to solve a MILP, one can first solve its LP relaxation using the Simplex method. The final tableaux form of the LP relaxation can be described by a system of equations $x + Rs + Py = b$, where x is the vector of basic (non-negative) variables, s is the vector of (non-negative) continuous variables, and y is the vector of (non-negative) integer variables. We will work with the relaxation of this set by dropping the nonnegativity constraints on all the basic variables x, as suggested in [52]. This is known in the literature as the *corner polyhedron*; see the surveys [5, 6, 7]. Note that under this relaxation, we can drop the constraints associated with the continuous basic variables x_i because these variables now only appear in one equation. Hence, we can assume that x is a vector of

integer variables. The model for this relaxation can be reexpressed as

$$\{(s, y) \in \mathbb{R}^k_+ \times \mathbb{Z}^l_+ : Rs + Py \in b + \mathbb{Z}^n\}. \tag{2.6}$$

Definition 2.7 (Mixed-integer set $X(R, P)$)**.** *The set*

$$X(R, P) := \{(s, y) \in \mathbb{R}^k_+ \times \mathbb{Z}^l_+ : Rs + Py \in b + \mathbb{Z}^n\} \tag{2.7}$$

is called a mixed-integer set*, where $k, l \in \mathbb{Z}_+, n \in N, R \in \mathbb{R}^{n\times k}$ and $P \in \mathbb{R}^{n\times l}$. Note that we allow $k = 0$ or $l = 0$, but not both.*

Remark 2.11. *In practice, not all the rows of the corner polyhedron are used. Instead, we only pick a subset of the rows. However, observe that the mixed-integer set obtained from this subset of the rows is still of the form in 2.7. Hence, all of the theory of cut generating functions can be applied in this context.*

In one extreme, the GMI cuts (see Remark 2.14), the most commonly used cutting planes in practice, are single row cuts. However, one would expect that using more than 1 *row may provide benefits because more information from the problem is used to generate the cuts.*

We will now comment on the significance of the mixed-integer set $X(R, P)$. Let's assume that the solution to the LP relaxation does not satisfy the integrality constraint, i.e. $b \notin \mathbb{Z}^n$. Recall from the cutting plane approach that we would like to find a valid inequality that cuts off this solution. The LP solution, in fact, corresponds to the origin $(0, 0)$ (i.e. by setting all the coordinates of s and y to 0) in the formulation 2.7 because all the non-basic variables of the LP solution are 0. In addition, since $b \notin \mathbb{Z}^n, 0 \notin X(R, P)$. Hence, the cutting plane approach seeks to seperate the point $(0, 0)$ from $X(R, P)$.

Remark 2.12. *The set $X(R, P)$ in 2.7 is a special case of the general mixed-integer set $X_S(R, P) := \{(s, y) \in \mathbb{R}^k_+ \times \mathbb{Z}^l_+ : Rs + Py \in S\}$ when $S = b + \mathbb{Z}^n$. Readers are referred to [5, Section 4.2] for treatment of the general S case.*

2.4.2 Cut Generating Function Pairs

We now give the definition of a cut generating function pair, which characterizes cuts for the mixed-integer set $X(R, P)$.

Definition 2.8 (Valid pair or Cut Generating Function pair)**.** *Fix $n \in \mathbb{N}$. Let $\psi : \mathbb{R}^n \mapsto \mathbb{R}$ and $\pi : \mathbb{R}^n \mapsto \mathbb{R}$. Then the pair (ψ, π) is called a valid pair or a cut generating function (CGF) pair if*

$$\sum_{i=1}^{k} \psi(r^i)s_i + \sum_{i=1}^{l} \pi(p^i)y_i \geq 1 \tag{2.8}$$

is a valid inequality for $X(R, P)$ for all k, l, R, P, where r^i is the i-th column of R and p^i is the i-th column of P. We want to emphasize that the pair (ψ, π) should give valid inequalities irrespective of $k, l, R,$ and P.

Remark 2.13. *The inequality 2.8 is indeed a valid inequality: It separates the origin $(0, 0)$ from $X(R, P)$ because $(0, 0)$ does not satisfy this inequality.*

Having seen the usefulness of such a CGF pair, we now demonstrate how such a pair could be generated. For this, we need to introduce a couple of concepts from convex geometry.

Definition 2.9 (Maximal $(b + \mathbb{Z}^n)$-free set)**.** *A closed, convex set $K \subseteq \mathbb{R}^n$ is said to be $(b + \mathbb{Z}^n)$-*free *if $int(K) \cap (b + \mathbb{Z}^n) = \emptyset$. It is called a* maximal $(b + \mathbb{Z}^n)$-free *set if there exists no other $(b + \mathbb{Z}^n)$-free set containing K.*

Definition 2.10 (Gauge function)**.** *Let $K \subseteq \mathbb{R}^n$ be a convex set such that $0 \in int(K)$. We define the* gauge function *of K as*

$$\psi_K(x) := \inf\{\lambda > 0 : \frac{x}{\lambda} \in K\},$$

for all $x \in R^n$.

We now consider a special case of the mixed-integer set $X(R, P)$: when $l = 0$, all the variables are continuous and we have $C(R) := \{s \in \mathbb{R}^k_+ : Rs \in b + \mathbb{Z}^n\}$. Note that we can think of $C(R)$ as $X(R, 0)$. Since in $C(R)$, there are only continuous variables, a cut

generating function pair reduces to a single function. To be specific, we call $\psi : \mathbb{R}^n \mapsto \mathbb{R}$ a *valid function* or a *cut generating function* for $C(R)$ if

$$\sum_{i=1}^{k} \psi(r^i)s_i \geq 1 \tag{2.9}$$

is a valid inequality for $C(R)$.

We want to study $C(R)$ and its cut generating functions because CGF pairs are generally built upon cut generating functions for $C(R)$, as will be shown later. We now gives a couple of definitions that will lead to a formula for finding cut generating functions for $C(R)$.

The following result gives the relationship between a $(b+\mathbb{Z}^n)$-free set and a valid function for $C(R)$.

Theorem 2.2. *Let $K \subseteq \mathbb{R}^n$ be a closed convex set such that $0 \in int(K)$. Then, ψ_K is a cut generating function, where ψ_K is the gauge function of K, if and only if K is $(b + \mathbb{Z}^n)$-free set.*

Proof. See Lemma 4.3 in [5]. □

Hence, the gauge function of a maximal $(b + \mathbb{Z}^n)$-free convex set will give us a cut generating function. In addition, another reason we prefer working with a maximal $(b+\mathbb{Z}^n)$-free set is that it has a nice characterization: Every maximal $(b+\mathbb{Z}^n)$-free set is a polyhedron. In particular, it is of the form $\{x \in \mathbb{R}^n : a_i x \leq 1,\ \forall i \in I\}$ for some finite set I. Moreover, its gauge function is $\psi(r) = \max_{i \in I} a_i r$ (See Theorem 4.5 in [5]). Hence, given a maximal $(b + \mathbb{Z}^n)$-free set, we have a formula to compute a valid function for $C(R)$.

We are now ready to discuss CGF pairs for the more general model $X(R, P)$. Let ψ be a cut generating function for $C(R)$. Then, for any $s, y \in X(R, P)$, one can see that

$$\sum_{i=1}^{k} \psi(r^i)s_i + \sum_{i=1}^{l} \psi(p^i)y_i \geq 1$$

because one can treat

$$R' = \begin{bmatrix} R & P \end{bmatrix}, s' = \begin{bmatrix} s \\ y \end{bmatrix}$$

and so because $R's' = Rs + Py \in b + \mathbb{Z}^n$ and ψ is a cut generating function for $C(R)$ where $R = R'$, we have the desired inequality, which implies that (ψ, ψ) is a valid CGF pair for $X(R, S)$.

Hence, one could obtain a "trivial" CGF pair as (ψ, ψ) for ψ being the gauge function of some maximal $(b + \mathbb{Z}^n)$-free convex set whose interior contains 0. However, in practice, it might not be beneficial to simply using the "trivial" CGF pair. A more common procedure to generate a CGF pair (ψ, π) for the mixed integer model $X(R, P)$ is to start by computing a valid function ψ for the corresponding continuous model $C(R)$, and then find π such that (ψ, π) is a valid pair. Any function π that makes (ψ, π) a valid pair is known as a *lifting* function of ψ. The reason we prefer this approach for finding CGF pairs to simply using the "trivial" CGF pair is that CGF pairs (ψ, π) from this procedure generally generate a stronger cut, in the sense that it cuts off more redundant parts from the feasible region of the LP relaxation.

We now show a result to generate a CGF pair from this procedure.

Definition 2.11 (Trivial Lifting [8])**.** *Let ψ be a valid function. Then,*

$$\widetilde{\psi}(x) := \inf_{z \in \mathbb{Z}^n} \psi(x + z)$$

is called the trivial lifting *of ψ.*

Theorem 2.3. *[39] Let K be a maximal $(b + \mathbb{Z}^n)$-free convex set. Then $(\psi_K, \widetilde{\psi_K})$ is a valid pair.*

2.4.3 Example

We now give an example to show how these valid pairs can be used to generate a cutting plane for a mixed integer linear program.

$$\begin{aligned} \min \quad & x_1 - 2x_2 \\ \text{subject to:} & \\ & 2x_1 + 5x_2 \leq 7 \\ & 3x_1 - 2x_2 \leq 2 \\ & x_1 \in \mathbb{R}_+, x_2 \in \mathbb{Z}_+ \end{aligned} \tag{2.10}$$

We can introduce slack variables to get the standard form.

$$\begin{aligned} \min \quad & x_1 - 2x_2 \\ \text{subject to:} & \\ & 2x_1 + 5x_2 + s_1 = 7 \\ & 3x_1 - 2x_2 + s_2 = 2 \\ & x_1 \in \mathbb{R}_+, x_2, s_1, s_2 \in \mathbb{Z}_+ \end{aligned} \tag{2.11}$$

The final simplex tableaux of its LP relaxation is

$$x_2 + 0.4x_1 + 0.2s_1 = 1.4$$
$$s_2 + 3.8x_1 + 0.4s_1 = 4.8,$$

which gives the optimal solution $(x_1, x_2) = (0, 1.4)$. Hence, the integrality constraint is violdated for x_2 and we need to find a cutting plane that cuts off this solution from the original feasible set. We now show how one can use the tools from the previous section to generate such a cutting plane.

Since the integrality constraint on x_2 is violated, we will focus on the equation

$$x_2 + 0.4x_1 + 0.2s_1 = 1.4. \tag{2.12}$$

In the language of the previous section, $s = x_1, y = s_1, b = [1.4] = 0.4$, where $[\cdot]$ denotes the fractional part. Let $K = [-0.6, 0.4]$, which is a maximal $(b + \mathbb{Z})$-free set. Let ψ be the

gauge of K. One can verify that

$$\psi(r) = \max\left(\frac{r}{[b]}, \frac{-r}{1-[b]}\right) = \max\left(\frac{r}{[0.4]}, \frac{-r}{0.6}\right)$$

and the trivial lifting $\widetilde{\psi}$ of ψ is given by

$$\widetilde{\psi}(r) = \min\left(\frac{[r]}{[b]}, \frac{1-[r]}{1-[b]}\right) = \min\left(\frac{[r]}{0.4}, \frac{1-[r]}{0.6}\right).$$

Applying the inequality $\psi(x_2) + \widetilde{\psi}(s_1) \geq 1$ to 2.12 gives us

$$x_1 + 0.5s_1 \geq 1.$$

Expressing this cut in the original variables x_1, x_2 gives us the inequality

$$x_2 \leq 1.$$

Moreover, one can check that using the "trivial" valid pair (ψ, ψ) also gives the same cut as using $(\psi, \widetilde{\psi})$.

Finally, adding this cut and solving the updated LP gives us the optimal solution for the original MILP: $(x_1, x_2) = (0, 1)$.

Remark 2.14. *The cut generated in the example above is known as* Gomory mixed integer (GMI) cut*, which is the most popular cutting plane used in modern day softwares for solving MILPs. We now present the definition of the GMI cut. Consider a standard MILP:*

$$\begin{aligned} &\textit{min} && c^\top x \\ &\textit{subject to:} && \\ & && Ax = b \\ & && x \geq 0 \\ & && x_i \in \mathbb{Z},\ \forall i \in I, \end{aligned} \tag{2.13}$$

where $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, I \subseteq [n]$ are given.

Then, with B being a basis and N being an index set of nonbasic variables, a final

simplex tableau for a LP relaxation is of the form

$$x_i + \sum_{j \in N} \overline{a}_i j x_j = \overline{b}_i, \;\; i \in B,$$

which gives the optimal solution of $x_i^* = \overline{b}_i$ *for* $i \in B$ *and* $x_j^* = 0$ *for* $j \in N$. *This is also optimal for the original MILP if* $\overline{b}_i \in \mathbb{Z}$ *for* $i \in B \cap I$. *If not, there exists an* $i \in B \cap I$ *such that* $\overline{b}_i \notin \mathbb{Z}$. *Let* $f_0 = \overline{b}_i - \lfloor \overline{b}_i \rfloor$ *and* $f_j = \overline{a}_{ij} - \lfloor \overline{a}_{ij} \rfloor$ *for* $j \in N$, *where* $\lfloor \cdot \rfloor$ *is the floor function. The GMI cut obtained from the row of the tableau associated with* x_i *is*

$$\sum_{\substack{j \in N \cap I \\ f_j \leq f_0}} \frac{f_j}{f_0} x_j + \sum_{\substack{j \in N \cap I \\ f_j > f_0}} \frac{1 - f_j}{1 - f_0} x_j + \sum_{\substack{j \in N \cap C \\ \overline{a}_{ij} \geq 0}} \frac{\overline{a}_{ij}}{f_0} x_j - \sum_{\substack{j \in N \cap C \\ \overline{a}_{ij} < 0}} \frac{\overline{a}_{ij}}{1 - f_0} x_j,$$

where $C = [n] \backslash I$ *is the set of continuous variables.*

Notice that there might be multiple basic variables x_i *whose integrality constraint is violated, which corresponds to different rows of the simplex tableau. The authors in [36] and in [3] show that the choice of row to use for generating GMI cuts can have an impact of their usefulness.*

Finally, one can verify that by applying the valid pair $(\psi, \widetilde{\psi})$ *introduced in the previous example to the row of the tableau associated with* x_i, *one would get back the GMI cut.*

Chapter 3

Statistics Background

Following the spirit of Chapter 2, this chapter presents the relevant background in statistics that are used in the thesis. In particular, we review methods in point estimation, the decision theory framework, the notion of admissibility, Bayes theorem, and statistical learning theory.

3.1 Point Estimation Methods

We start the section by providing a framework to understand a statistical problem in general.

Definition 3.1 (Statistical model)**.** *Assume we have a random experiment whose outcomes come from a sample space Ω. In addition, we observe a random vector $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)$. When $\omega \in \Omega$ is the outcome, we call the realization $\boldsymbol{X}(\omega)$, denoted by $(X_1, X_2, \cdots, X_n)$, the* data*. Finally, we assume that the probability distribution of $\boldsymbol{X}$ comes from a family $\mathcal{P}$ of distributions. $\mathcal{P}$ is also known as the* statistical model*.*

Remark 3.1. *There are three classes of statistical models: parametric, semiparametric, and nonparametric. In this dissertation, we are only interested in parametric model, where there is a parameter space Θ that specifies distribution in $\mathcal{P}$. In that case, our model can be expressed as $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$.*

Example 3.1. *Suppose we have n i.i.d. random variables $X_1, X_2, \cdots, X_n$ from a normal distribution $\mathcal{N}(\mu, \sigma^2)$, where (μ, σ^2) is unknown. Then this is a parametric model with the parameter space $\Theta = \{(\mu, \sigma^2) : \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+\}$.*

Definition 3.2 (Statistics)**.** *Assume a random vector takes value in a sample space* $\mathcal{X}$*. Then a* statistic T is a mapping $T : \mathcal{X} \mapsto \mathcal{T}$, where $\mathcal{T}$ is some space - usually a Euclidean space.

The problem of point estimation can now be stated as follows. Assume we have a parametric model where a random variable X has an unknown distribution in a family $\{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$. By writing $\boldsymbol{\theta}$ in vector form, we would like to emphasize that there can be multiple parameters that specify the distributions in family $\mathcal{P}$. Suppose further that we do not know some of the parameters $\{\boldsymbol{\theta}_i, i \in I\}$ for some subset I, but we have access to a sample $X_1, X_2, \cdots, X_n$ of X. The goal is to choose a statistic $T(X_1, X_2, \cdots, X_n)$ that gives the best estimate of $\boldsymbol{\theta}_I$, which is the subvector of $\boldsymbol{\theta}$ indexed by I. T is called an *estimator* of $\boldsymbol{\theta}_I$ and the realization $T(x_1, x_2, \cdots, x_n)$ is called an *estimate*.

Example 3.2. *Let* $X_1, X_2, \cdots, X_n$ *be a random sample from Poisson's distribution with unknown parameter* λ*. Then an example of an estimator of* λ *is*

$$T(X_1, X_2, \cdots, X_n) = \frac{\sum_{i=1}^{n} X_i}{n}.$$

There are many desirable properties for a "good" estimator, such as

- Unbiased: An estimator T is called an *unbiased* estimator of θ if and only if $\mathbb{E}(T) = \theta,\ \forall \theta \in \Theta$.

- Minimum-variance unbiased: An estimator T is said to be a *minimum-variance unbiased* estimator of θ if it is an unbiased estimator that has the lowest variance, i.e. $\mathbb{V}(T) \leq \mathbb{V}(T'),\ \forall \theta \in \Theta$ for any unbiased estimator T'.

- Consistent: An estimator T_n is defined to be a *weakly consistent* estimator if for any $\varepsilon, \delta > 0, \exists n_0(\varepsilon, \delta)$ such that $\mathbb{P}(|T_n - \theta| \leq \varepsilon) > 1 - \delta$ for $n \geq n_0$ and for all $\theta \in \Theta$. This is saything that the estimate becomes more precise as the sample size increases.

- Efficient: Given two unbiased estimators for *theta*, T_1 and T_2, we say T_1 is more *efficient* than T_2 if $\mathbb{V}(T_1) < \mathbb{V}(T_2)$. Under certain regularity conditions, we can define the efficiency of an unbiased estimator T as $\text{eff}(T) = \frac{1/I(\theta)}{\mathbb{V}(T)}$, where $I(\theta) =$

$\mathbb{E}[(\frac{\partial}{\partial\theta}\log f(X;\theta))^2|\theta]$ is the Fisher information when this exists. Then, using the Cramér-Rao bound (see [81]), we have $\text{eff}(T) \leq 1$ for all unbiased estimator T.

Readers are referred to Chapter 1 in [85] for a more comprehensive survey of results relating to these properties.

There are many commonly used point estimation methods that provide estimators having some of these desirable properties, such as method of moments, method of maximum likelihood, method of minimum χ^2, and method of least squares. In this thesis, we are mostly concerned with the method of maximum likelihood as it is very popular and commonly used in practice, and often leads to efficient estimators.

Definition 3.3 (Method of Maximum Likelihood)**.** *Suppose random variables $X_1, X_2, \cdots, X_n$ have a joint density or frequency function $f(x_1, x_2, \cdots, x_n|\theta)$. Then, given observations $X_i = x_i$, we define the* likelihood *of θ as a function of $x_1, x_2, \cdots, x_n$ as*

$$L(\theta) = f(x_1, x_2, \cdots, x_n|\theta).$$

The maximum likelihood estimate *of θ, θ_{ML} is the value of θ that maximizes $L(\theta)$ assuming it exists and is unique.*

Example 3.3. *Assume $X_1, X_2, \cdots, X_n$ are i.i.d. $\mathcal{N}(\mu, \sigma^2)$ with unknown μ and σ. Then their joint density is simply*

$$L(\mu,\sigma) = f(x_1, x_2, \cdots, x_n|\mu,\sigma) = \prod_{i=1}^{n} f(x_i|\mu,\sigma) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left[\frac{x_i-\mu}{\sigma}\right]^2\right)$$

Thus, the log likehood is

$$l(\mu,\sigma) = -n\log\sigma - \frac{n}{2}\log 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

Taking the partial derivative of $l(\mu, \sigma)$ gives

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu)$$
$$\frac{\partial l}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i - \mu)^2.$$

Setting these partial derivatives to 0, we obtain $\mu^* = \overline{x}$ and $\sigma^* = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$. We now compute the second order partial derivatives to verify that μ^* and σ^* indeed maximize $l(\mu, \sigma)$:

$$\frac{\partial^2 l}{\partial \mu^2}(\mu^*, \sigma^*) = -\frac{n}{\sigma^{*2}} < 0$$
$$\frac{\partial^2 l}{\partial \sigma^2}(\mu^*, \sigma^*) = \frac{n}{\sigma^{*2}} - \frac{3}{\sigma^{*4}}\sum_{i=1}^{n}(x_i - \mu^*)^2 = -\frac{2n}{\sigma^{*2}}$$
$$\frac{\partial^2 l}{\partial \sigma \partial \mu}(\mu^*, \sigma^*) = -\frac{2}{\sigma^{*3}}\sum_{i=1}^{n}(x_i - \overline{x}) = 0.$$

In addition, one can see that

$$\frac{\partial^2 l}{\partial \mu^2}(\mu^*, \sigma^*) \times \frac{\partial^2 l}{\partial \sigma^2}(\mu^*, \sigma^*) - \left(\frac{\partial^2 l}{\partial \sigma \partial \mu}(\mu^*, \sigma^*)\right)^2$$
$$= \frac{2n^2}{\sigma^{*4}} - 0 > 0.$$

Hence, we can conclude that $\mu_{ML} = \overline{x}$ and $\sigma_{ML} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$.

3.2 Decision Theory Framework

In this section, we explain the decision theory framework for a statistical inference problem. In settings where one needs to make decisions under uncertainty, the framework allows one in principle to evaluate all the available choices one can take. To be specific, suppose we have a statistical model with a random vector $\boldsymbol{X}$ whose distribution is a member of a family $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$. In addition, assume $\boldsymbol{X}$ takes values in a sample space $\mathcal{X}$.

3.2.1 Components of the decision theory framework

We now discuss the different components of the decision theory framework.

State space. Θ is called the *state or parameter space* and $\theta \in \Theta$ is called a state of nature. This represents the uncertainty in the problem.

Action space. The *action space* $\mathcal{A}$ consists of all the actions, or decisions, a that the statistician can make. For example, in Example 3.2, if we want to estimate the value of λ, the action space is $[0, \infty)$.

Loss function. The *loss function* is a function $l : \Theta \times \mathcal{A} \to \mathbb{R}^+$. (θ, a) quantifies the loss incurred if the statistician takes action a and the true state of nature is θ. For an estimation problem, some of the most popular loss function are squared Euclidean distance loss function, absolute distance loss function, and maximum distance loss function.

Decision rule. A *decision rule* is a function $\delta : \mathcal{X} \to \mathcal{A}$. Using a rule δ means that the statistician, having observed data $\boldsymbol{X} = \mathbf{x}$, will take action $\delta(\mathbf{x})$. Notice that we can have lots of decision rules, in many cases an infinite number. We call the class of all decision rule the *decision space* and denote it by $\mathcal{D}$.

Risk function. As we have a great amount of decision rules, we need a procedure to measure the performance of each rule. Notice that when θ is the true state of nature and we observe $\boldsymbol{X} = \mathbf{x}$, then the loss is $l\left(\theta, \delta(\mathbf{x})\right)$. However, the problem is that θ is unknown. Also, we would like a decision rule that performs well across the values of $\mathbf{x}$. As a result, we will introduce the *risk function* as a measure of the performance of $\delta(\mathbf{x})$ by taking the average of the loss over the sample space:

$$R(\theta, \delta) = \mathbb{E}_\theta[l(\theta, \delta(\boldsymbol{X}))].$$

Importantly, this is a function of θ.

Example 3.4. *Assume* $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)$ *are i.i.d.* $\mathcal{N}(\mu, \sigma^2)$ *with unknown* $\mu \in \mathbb{R}$ *and known* σ*. Hence, the action space* $\mathcal{A} = \mathbb{R}$*. Assume the loss function* $l(\mu, a) = (\mu - a)^2$*. Then for the decision rule* $\delta(\boldsymbol{X}) = \overline{\boldsymbol{X}}$*, we have*

$$l\left(\mu, \overline{\boldsymbol{X}}\right) = \left(\overline{\boldsymbol{X}} - \mu\right)^2 = \frac{\sigma^2}{n}\left(\frac{\overline{\boldsymbol{X}} - \mu}{\sigma/\sqrt{n}}\right)^2.$$

Notice that $\left(\frac{\overline{\boldsymbol{X}}-\mu}{\sigma/\sqrt{n}}\right)^2$ *is a* χ_1^2 *distribution. Hence,* $R\left(\mu, \overline{\boldsymbol{X}}\right) = \mathbb{E}_\mu\left(\overline{\boldsymbol{X}} - \mu\right)^2 = \frac{\sigma^2}{n}$*, which is a constant.*

3.2.2 Comparing between decision rules

Now that we have a way to quantify the performance of a decision rule, we need to define some criteria for decision selection. Notice that it is not as simple as comparing between two real numbers, as the risk function for a particular decision rule gives a risk profile over all possible $\theta \in \Theta$. So for two decision rules δ and δ', one might perform better over another on a set of θ values but worse on another set of values. With that in mind, we introduce the minimal criterion for selecting a decision rule.

Definition 3.4 (Dominating and Admissible decision rules)**.** *A decision rule* δ *is said to* dominate *another decision rule* δ' *if and only if* $R(\theta, \delta) \leq R(\theta, \delta')$ *for all* θ *with strict inequality for some* θ*. A decision rule* δ *is called an* admissible *rule if it is not dominated by any other decision rule.*

Remark 3.2. *We say admissibility is a weak requirement for choosing a decision rule because there are admissible rules that are somewhat "useless". For example, in Example 3.4, we can define another decision rule* $\delta(\boldsymbol{X}) = c$*, for a fixed constant* c*. Then the risk is* $R(\mu, c) = (\mu - c)^2$*, which is* 0 *if* $\mu = c$ *is the true state of nature. Since no other rule can have a zero risk at* $\mu = c$*, this decision rule is admissible.*

Ideally, the best decision rule δ would be one that dominates every other rule, i.e. $R(\theta, \delta) \leq R(\theta, \delta')$ for all θ and δ' with strict inequality for some θ. However, such a rule typically does not exist. Hence, we need another criteria to compare between decision rules. There are two approaches for the decision selection problem. The first is by adding some

restrictions to narrow down the class of decision rules, such as unbiasedness or equivariance (see [42, 68]). The second approach is to use some global criteria over all $\theta \in \Theta$, such as

- Minimaxity: A rule δ is called *minimax* if it minimizes the value $\sup_\theta R(\theta, \delta)$, i.e. the maximum risk over all θ.

- Minimum Bayes risk: In this Bayesian point of view, instead of treating θ as a fixed unknown variable, we assume it is a realization of a random variable $\boldsymbol{\theta}$. As a result, P_θ is the conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{\theta} = \theta$. The risk function is $\mathbb{E}_{\boldsymbol{X}|\theta}\left[l(\boldsymbol{\theta}, \delta(\boldsymbol{X}))|\boldsymbol{\theta} = \theta\right]$. By taking the average of this risk over all θ, we get the *Bayes risk*:

$$r(\delta) = \mathbb{E}_\theta\left[R\left(\boldsymbol{\theta}, \delta\right)\right] = \mathbb{E}_\theta \mathbb{E}_{\boldsymbol{X}|\theta}\left[l(\boldsymbol{\theta}, \delta(\boldsymbol{X}))|\boldsymbol{\theta} = \theta\right] = \mathbb{E}\left[l\left(\boldsymbol{\theta}, \delta(\boldsymbol{X})\right)\right].$$

  Then the decision rule that minimizes $r(\delta)$ is called a *Bayes rule*.

Remark 3.3. *For the Bayes risk, assuming the prior distribution on θ is π, we normally write the Bayes risk for a particular decision rule δ as $r(\pi, \delta)$ to emphasize its dependent on the prior distribution.*

3.2.3 Improving a decision rule

We now show a method to improve upon a decision rule, which is known as the Rao-Blackwell procedure.

Definition 3.5 (Sufficient Statistic)**.** *Suppose $X \sim P_\theta, \theta \in \Theta$. Then a statistic $T(X)$ is a* sufficient statistic *for θ if the conditional distribution of X given $T(X) = t$ does not depend on θ. Intuitively, this means a sufficient statistic contains all the necessary information to estimate θ and so it can be used as a data-reduction tool.*

The following theorem by Fisher provides a nice way to verify whether or not a statistic is sufficient.

Theorem 3.1 (Factorization Theorem)**.** *Given a model with $X \sim P_\theta, \theta \in \Theta$, a statistic $T : \mathcal{X} \to \mathcal{T}$ is sufficient for θ if and only if there exists a function $g : \Theta \times \mathcal{T} \to [0, \infty)$ and*

a function $h : \mathcal{X} \to [0, \infty)$ *such that*

$$p_\theta(x) = g(T(x), \theta)h(x),$$

for all $x \in \mathcal{X}, \theta \in \Theta$.

Proof. See section 1.9 in [69]. □

Example 3.5 (Normal sufficient statistic)**.** *Let* $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ *be i.i.d.* $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

We have,

$$\begin{aligned}
p_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}_1, \cdots, \mathbf{x}_2) &\propto \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})\right) \\
&\quad \times \exp\left(-\frac{1}{2}\sum_{i=1}^n (\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})\right) \\
&\quad \times \exp\left(-\frac{n}{2}(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\sum_{i=1}^n \operatorname{Tr}\left((\mathbf{x}_i - \overline{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})\right)\right) \\
&\quad \times \exp\left(-\frac{n}{2}(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})\right)\right) \\
&\quad \times \exp\left(-\frac{n}{2}(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\sum_{i=1}^n \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \overline{\mathbf{x}})^\top (\mathbf{x}_i - \overline{\mathbf{x}})\right)\right) \\
&\quad \times \exp\left(-\frac{n}{2}(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right) \\
&= \frac{1}{\sqrt{|\boldsymbol{\Sigma}|^n}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}\right)\right) \\
&\quad \times \exp\left(-\frac{n}{2}(\overline{\mathbf{x}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\overline{\mathbf{x}} - \boldsymbol{\mu})\right),
\end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}} = \sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})^\top (\mathbf{x}_i - \overline{\mathbf{x}})$. *The third equality is due to the fact that* $\mathbf{x}^\top D\mathbf{x} =$

$\mathrm{Tr}(\mathbf{x}^\top D\mathbf{x})$ *for a diagonal matrix* D*. The fourth and fifth equality comes from the linear mapping and cyclic property of the trace, respectively.*

Using the Factorization Theorem, we can see that $T(\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n) = (\overline{\boldsymbol{X}}, \widehat{\boldsymbol{\Sigma}})$ *is sufficient for* $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

We are now ready to show the Rao-Blackwell theorem, which offers a procedure to improve an estimator.

Theorem 3.2 (Rao-Blackwell Theorem)**.** *Let* X *be a random variable with distribution in the family* $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ *and let* T *be a sufficient statistic for* θ*. Let* $\delta(X)$ *be an estimator of* θ *with finite expectation and risk. In addition, assume the loss function* $l(\theta, a)$ *is a convex function of* a*. Then, the estimator* $\delta^*(T) = \mathbb{E}[\delta(X)|T]$ *performs at least as well as* δ*, i.e.*

$$R(\theta, \delta^*) \leq R(\theta, \delta),$$

for all $\theta \in \Theta$*. Moreover, if the loss function is strictly convex, the inequality is strict.*

Proof. See Theorem 7.8 in [68, Section 1.7]. □

3.2.4 Admissibility results

We know from its definition that for an inadmissible rule, there exists another rule that performs, on average, at least as good as it does on very instance of θ. Hence, it is obvious that we are only interested admissible rules. However, it is generally hard to check by definition whether a rule is admissible. In this section, we provide some results that prove admissibility of a decision rule.

Theorem 3.3. *If* δ *is the unique Bayes estimator (almost surely for all* P_θ*), then it is admissible.*

Proof. Let δ_π be the unique Bayes estimator with respect to the prior distribution π. Suppose for the sake of contradiction that δ_π is inadmissible, i.e. there exists another estimator δ'_π that dominates δ_π. Hence, $R(\theta, \delta'_\pi) \leq R(\theta, \delta_\pi)$, with strict inequality for some θ. Then,

$$r(\pi, \delta'_\pi) = \int R(\theta, \delta'_\pi) d\pi(\theta) \leq \int R(\theta, \delta_\pi) d\pi(\theta) = r(\pi, \delta_\pi),$$

so δ'_π is also a Bayes estimator with respect to π, contradicting the uniqueness assumption.

□

Theorem 3.4. *Suppose δ_π is a Bayes estimator having finite Bayes risk with respect to the prior distribution π and support$(\pi) = \Theta$. Then δ_π is an admissible estimator if either of the following holds:*

- *Θ is finite.*
- *Θ is an open subset of $\mathbb{R}^n$ and $R(\theta, \delta)$ is continuous in θ, $\forall \delta \in \mathcal{D}$.*

Proof. We prove the two cases separately:

Discrete case: Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_d\}$ amnd δ_π is Bayes with respect to $\pi = \{\pi_1, \pi_2, \cdots, \pi_d\}$, where $\pi_i > 0,\ \forall i \in [d]$. Suppose to the contrary that there exists another estimator δ that dominates δ_π: $R(\theta, \delta) \leq R(\theta, \delta_\pi),\ \forall \theta \in \Theta$, with strict inequality for some θ. Then,

$$r(\pi, \delta) = \sum_{i=1}^{d} \pi_i R(\theta_i, \delta) < \sum_{i=1}^{d} \pi_i R(\theta_i, \delta_\pi) = r(\pi, \delta_\pi)$$

because $\pi_i > 0,\ \forall i \in [d]$. This contradicts the assumption that δ_π is Bayes with respect to π.

Continuous case: Again, we will prove by contradiction. Suppose that there exists another estimator δ that dominates δ_π: $R(\theta, \delta) \leq R(\theta, \delta_\pi),\ \forall \theta \in \Theta$, with strict inequality for some θ. Let's consider the following function of θ: $R(\theta, \delta_\pi) - R(\theta, \delta)$. By assumption, this function is continuous in θ and is positive for some $\theta_0 \in \Theta$. Hence, $\exists\ \varepsilon > 0$ such that $\forall \theta : |\theta - \theta_0| < \varepsilon, R(\theta, \delta_\pi) - R(\theta, \delta) > \eta > 0$. Let's define $\Omega = \{\theta : |\theta - \theta_0| < \varepsilon\}$. Then,

$$\begin{aligned} r(\pi, \delta_\pi) - r(\pi, \delta) &= \int [R(\theta, \delta_\pi) - R(\theta, \delta)] d\pi(\theta) \\ &= \int_\Omega [R(\theta, \delta_\pi) - R(\theta, \delta)] d\pi(\theta) + \int_{\Omega^C} [R(\theta, \delta_\pi) - R(\theta, \delta)] d\pi(\theta) \\ &\geq \int_\Omega [R(\theta, \delta_\pi) - R(\theta, \delta)] d\pi(\theta) \\ &> \eta \pi(\Omega) > 0, \end{aligned}$$

contradicting the assumption that δ_π is the minimizer of $r(\pi, \delta)$. □

We now present a powerful method, called Blyth's method, to verify the admissibility of an estimator.

Theorem 3.5. *[68, Theorem 7.13] Suppose Θ is open and the risk function $R(\theta, \delta)$ is continuous in θ for all $\delta \in D$. Let δ be an estimator and π_n be a sequence of (possibly improper) prior measures such that*

(i) $r(\pi_n, \delta) < \infty$ *for all* n,

(ii) for any nonempty open set $\Omega_0 \in \Omega, \exists\ B > 0, N > 0$ *such that*

$$\int_{\Omega_0} \pi_n(\theta) d\theta \geq B \quad \text{for all } n \geq N,$$

(iii) $r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n}) \to 0$ *as* $n \to \infty$.

Then, δ is admissible.

Proof. Proof by contradiction: Suppose that there exists another estimator δ' that dominates δ: $R(\theta, \delta') \leq R(\theta, \delta),\ \forall \theta \in \Theta$, with strict inequality for some θ. Since the risk functions are continuous, $\exists \Omega_0$ and $\varepsilon > 0$ such that $R(\theta, \delta) - R(\theta, \delta') > \varepsilon, \forall \theta \in \Omega_0$. Then, for all $n \geq N$, we have

$$\begin{aligned}
r(\pi_n, \delta) - r(\pi_n, \delta') &= \int [R(\theta, \delta) - R(\theta, \delta')] d\pi_n(\theta) \\
&= \int_{\Omega_0} [R(\theta, \delta) - R(\theta, \delta')] d\pi_n(\theta) + \int_{\Omega_0^C} [R(\theta, \delta) - R(\theta, \delta')] d\pi_n(\theta) \\
&\geq \int_{\Omega_0} [R(\theta, \delta_\pi) - R(\theta, \delta)] d\pi(\theta) \\
&> \varepsilon \int_{\Omega_0} d\pi_n(\theta) \\
&\geq \varepsilon B.
\end{aligned}$$

Hence, $r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n}) + r(\pi_n, \delta^{\pi_n}) - r(\pi_n, \delta') > \varepsilon B, \forall n \geq N$. But by assumption (iii), $r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n}) \to 0$. Thus, $\exists n_0$ such that for all $n \geq n_0, r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n}) < \frac{\varepsilon B}{2}$. Consequently, $\forall n \geq \max\{N, n_0\}, r(\pi_n, \delta^{\pi_n}) - r(\pi_n, \delta') > \frac{\varepsilon B}{2}$, which contradicts the assumption that δ^{π_n} is Bayes with respect to π_n. □

Remark 3.4. *In Theorem 3.5, we can combine assumptions (ii) and (iii) and rewrite the theorem as follows:*

Suppose Θ is open and the risk function $R(\theta, \delta)$ is continuous in θ for all $\delta \in D$. Let δ be an estimator and π_n be a sequence of (possibly improper) prior measures such that

(i) $r(\pi_n, \delta) < \infty$ for all n,

(ii) for any nonempty open set $\Omega_0 \in \Omega$,

$$\frac{r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n})}{\int_{\Omega_0} \pi_n(\theta) d(\theta)} \to 0 \text{ as } n \to \infty.$$

Then, δ is admissible.

The proof of this is very similar to the proof of the previous theorem.

From the previous theorem, we can see that a sufficient condition for an estimator to be admissible is that its Bayes risk is the limit of some sequence of Bayes risks of Bayes estimators. It turns out that the converse is also true under some assumptions, i.e. we can say that admissible estimator is the limit of some Bayes estimators. (See [68, Theorem 7.15] and [24, Theorem 4A.12]).

3.3 Statistical Learning Theory

Statistical learning theory provides a framework for machine learning by using tools from statistics and functional analysis. Statistical learning can be categorized into 3 main types: supervised learning, unsupervised learning, and reinforcement learning. In this thesis, we are mainly concerned with supervised learning, which is also the most widely used type of problems in practice. In particular, we will review classification algorithms in supervised learning.

Problem setup In a typical supervised learning problem, we are given a training set of input-output pairs $\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$, where the X_i is an input vector and Y_i is the output that corresponds to it. In classification problems, Y takes discrete values while in regression problems, it takes real values. Let $\mathcal{X}$ be the vector space of all

possible inputs and $\mathcal{Y}$ be the vector space of all possible outputs. We assume that each of the input-output pairs is a sample from a fixed but unknown probability distribution over the product space $\mathcal{X} \times \mathcal{Y}$, $p(X, Y)$. The goal of learning is to find a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(X)$ best approximates Y. In addition, we define a loss function $L(f(X), Y)$ that measures the error in prediction. Then, we introduce the concept of *risk* to measure the average loss over the unknown distribution:

$$R_{L,p}(f) = \mathbb{E}L(f(X), Y) = \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y)p(x, y)dxdy.$$

Hence, a good function is one that has small risk.

We define the *Bayes risk* to be the minimum of risk over all functions f : $R^*_{L,p} = \inf_f R_{L,p}(f)$, and so the function that has this risk would be our best function. For a classification problem, such a function is called a *Bayes classifier* and the Bayes risk is also known as *the Bayes error rate*. However, we do not know $p(X, Y)$, which means we are not able to calculate the true risk for any function f. Though, we can compute an approximation for the true risk by averaging the loss function over the training set:

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} L(f(X_i), Y_i),$$

which is called the *empirical risk*. Under some fairly general conditions, we can use the law of large numbers to see that for a fixed function f, the empirical risk converges to the true risk as the sample size increases. Hence, to find an f that minimize $R_{L,p}(f)$, one could try to minimize $R_{emp}(f)$ instead. However, simply minimizing the empirical risk over all possible functions is a bad idea as it will lead to *overfitting*, the phenomenon of learning a function that predicts well over the training set but does poorly when given new, unseen data. To deal with the problem of overfitting, we should first restrict f to some space of functions $\mathcal{H}$. In addition, when minimizing the empirical risk, we can include a regularization term, with Tikhonov regularization [93] being the most commonly used, that penalizes functions that fluctuate a lot over small regions of input space. It has been shown that using these two approaches, the solution obtained from the empirical risk minimization procedure will have

nice properties. Readers are referred to [75, 73, 77, 59] for a more comprehensive treatment of the subject.

We now give a brief introduction to some of the machine learning algorithms that are used in Chapter 4.

3.3.1 k Nearest Neighbor

k Nearest Neighbor (k-NN) is a memory-based algorithm that looks at the class of the k points in the training set $\mathcal{D}$ that are nearest to a test input x and outputs a label that is most common among these k points, or in other words, the most probable label. In particular, assuming there are c classes and k is fixed, then we have

$$p(y = c|\mathcal{D}, x) = \frac{1}{K} \sum_{i \in N_k(x,\mathcal{D})} \mathbb{1}_{(y_i=c)},$$

where $N_k(x, \mathcal{D})$ are the indices of k nearest (based on same distance metric) points to x in $\mathcal{D}$ and $\mathbb{1}_{(y_i=c)}$ is an indicator function that takes value of 1 when $y_i = c$ and 0 otherwise. Then, the output of k-NN algorithm is

$$\hat{y}(x) = \operatorname*{argmax}_c p(y = c|\mathcal{D}, x).$$

This is also known as a MAP, which stands for maximum a posteriori, estimate. We now give a formal justification of why this is a good estimate. Using the formal problem setup as discussed earlier, let the loss function be a $0 - 1$ loss, i.e.,

$$L(a, y) = \begin{cases} 0 & \text{if } a = y \\ 1 & \text{if } a \neq y \end{cases}.$$

Then, under the Bayesian approach, we can get the posterior expected loss

$$\mathbb{E}_{p(y|x)}[L(a, y)] = p(a \neq y|x) = 1 - p(y|x).$$

Hence, the optimal action is one that minimizes the posterior expected loss, thus maximizing $p(y|x)$.

3.3.2 Logistic Regression

Logistic regression is a supervised learning algorithm that is commonly used for binary classification. It can also be extended to multiclass classification. However, we only review material for the binary case as it is more related to our work and is easier to follow. In logistic regression, we assume the conditional probability of Y given X is a sigmoid function $p(x) = p(Y = 1|X = x) = \frac{1}{1+e^{-(\beta_0+\beta^\top x)}}$, where β is an unknown parameter. The goal of learning in this case is to estimate the parameter vector β, which can be done using the maximum likelihood approach. Before getting there, we would like to point out that from the definition of $p(x)$, one can easily verify that $\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta^\top x$.

The conditional likelihood of a single observation is $p(y^i|x^i; \beta_0, \beta) = p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$. Hence, the conditional likelihood of the whole dataset is

$$L(\beta_0, \beta) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}.$$

The log likelihood is thus

$$\begin{aligned}
l(\beta_0, \beta) &= \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \\
&= \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i)}{1 - p(x_i)}\right) + \log (1 - p(x_i)) \\
&= \sum_{i=1}^{n} y_i(\beta_0 + \beta^\top x_i) + \log\left(\frac{e^{-(\beta_0+\beta^\top x_i)}}{1 + e^{-(\beta_0+\beta^\top x_i)}}\right) \\
&= \sum_{i=1}^{n} y_i(\beta_0 + \beta^\top x_i) + \log\left(\frac{1}{1 + e^{\beta_0+\beta^\top x_i}}\right) \\
&= \sum_{i=1}^{n} y_i(\beta_0 + \beta^\top x_i) - \log\left(1 + e^{\beta_0+\beta^\top x_i}\right).
\end{aligned}$$

Taking partial derivatives with β_0 and components of β gives

$$\begin{aligned}
\frac{\partial l}{\partial \beta_0} &= \sum_{i=1}^{n} y_i - \frac{e^{\beta_0+\beta^\top x_i}}{e^{\beta_0+\beta^\top x_i}} \\
&= \sum_{i=1}^{n} y_i - p(x_i) \\
\frac{\partial l}{\partial \beta_j} &= \sum_{i=1}^{n} y_i x_{ij} - \frac{e^{\beta_0+\beta^\top x_i}}{e^{\beta_0+\beta^\top x_i}} x_{ij} \\
&= \sum_{i=1}^{n} y_i - p(x_i) x_{ij}
\end{aligned}$$

By setting these partial derivatives to 0, we could solve for (β_0, β) that maximizes the likelihood function. However, since these are transcendental equations, there is no closed-form solution. Hence, we will need to use numerical methods; since $-l(\beta_0, \beta)$ is a convex function, the traditional method to solve for β_0 and β is the Newton Raphson method.

3.3.3 Random Forests

We first describe the concept of a *decision tree*. Decision tree builds a classification model in a tree structure. It breaks down the input space into smaller and smaller regions while building the tree. The deeper the tree is, the more complex the tree becomes. The final tree will consists of decision nodes (intermediate nodes) and leaf nodes. A decision node has two (binary tree) or more (multiway tree) branches. The decisions at each node involve only a single feature, or input coordinate. We should note that the regions in the input space corresponding to 2 splitted branches need to form a partition of the region corresponding to the node. For example, for continuous variables, the splits always the form $x_i \leq t$ and $x_i > t$ for some real-valued t. Leaf nodes represent a classification decision.

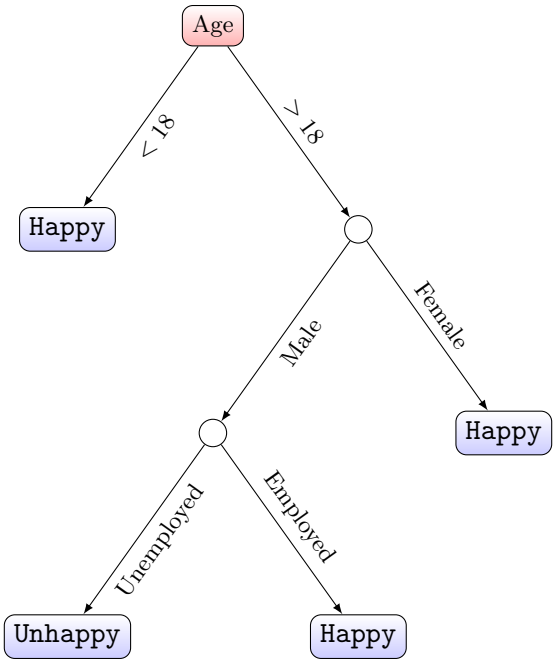

Figure 3.1: Example of a decision tree. The features are Age:$\{1, 2, 3, \cdots\}$, Sex: {Male, Female}, and Employment status: {Employed, Unemployed}. The target is {Happy, Unhappy}. All the light blue boxes correspond to leaf nodes, where a prediction is made.

More formally, suppose we are given a training set $\mathcal{D}$ with N observations (x_i, y_i), with $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $y_i \in [K]$. Suppose we have a partition of the input space into M regions $R_1, R_2, \cdots, R_M$. Let node m represents region R_m with N_m observations. At each leaf node m, we first estimate the class-conditional probabilities by

$$\hat{\pi}_{mk} = \frac{1}{N_m} \sum_{i: x_i \in \mathbb{R}_m} \mathbb{1}_{(y_i = k)}.$$

Then the predicted classification for node m, which is the same for every observation whose input x falls into region R_m, is

$$\hat{y}_m = \underset{k}{\text{argmax}}\ \hat{\pi}_{mk}.$$

We now explain how each node is splitted. Let R_L and R_R be the 2 regions corresponding to a potential node split with N_L and N_R points in the region respectively. Define $Q(R_L)$ and $Q(R_R)$ to be the node impurity measures. Then, we find the split that minimizes $N_L Q(R_L) + N_R Q(R_R)$. Some of the impurity measures that are often used in practice are:

- Misclassification rate: $Q_m = \frac{1}{N_m} \sum_{i:x_i \in \mathbb{R}_m} \mathbb{1}_{(y_i \neq k)} = 1 - \hat{\pi}_{mk}$.
- Gini index: $Q_m = \sum_{i=1}^{K} \hat{\pi}_{mk}(1 - \hat{\pi}_{mk})$.
- Entropy: $Q_m = -\sum_{i=1}^{K} \hat{\pi}_{mk} \log \hat{\pi}_{mk}$.

Readers are referred to [23] for an implementation of the decision tree method using CART algorithm.

A *random forest*, which was first introduced by Breiman [22], can then be defined as a collection of decision trees whose predictions are aggregated into one final result. In particular, each of these decision trees are learned based on a random subset of input features using random subset of the dataset $\mathcal{D}$. Assume we have trained T decision trees and for each input x, let $\hat{C}_t(x)$ be the class prediction of the tth tree at point x. Then the prediction given by the random forest is $\hat{C}_{RF}(x) = \text{mode}\{\hat{C}_t(x)\}_1^T$, i.e. the prediction with the majority vote from all trees.

This is an example of a more general technique called *bagging* [21], which aims to reduce the variance of an estimate by averaging the predictions from many different estimates. It is a powerful technique that is not peculiar to random forests: it can be applied to other classification algorithms to improve their performance.

3.3.4 Support Vector Machine

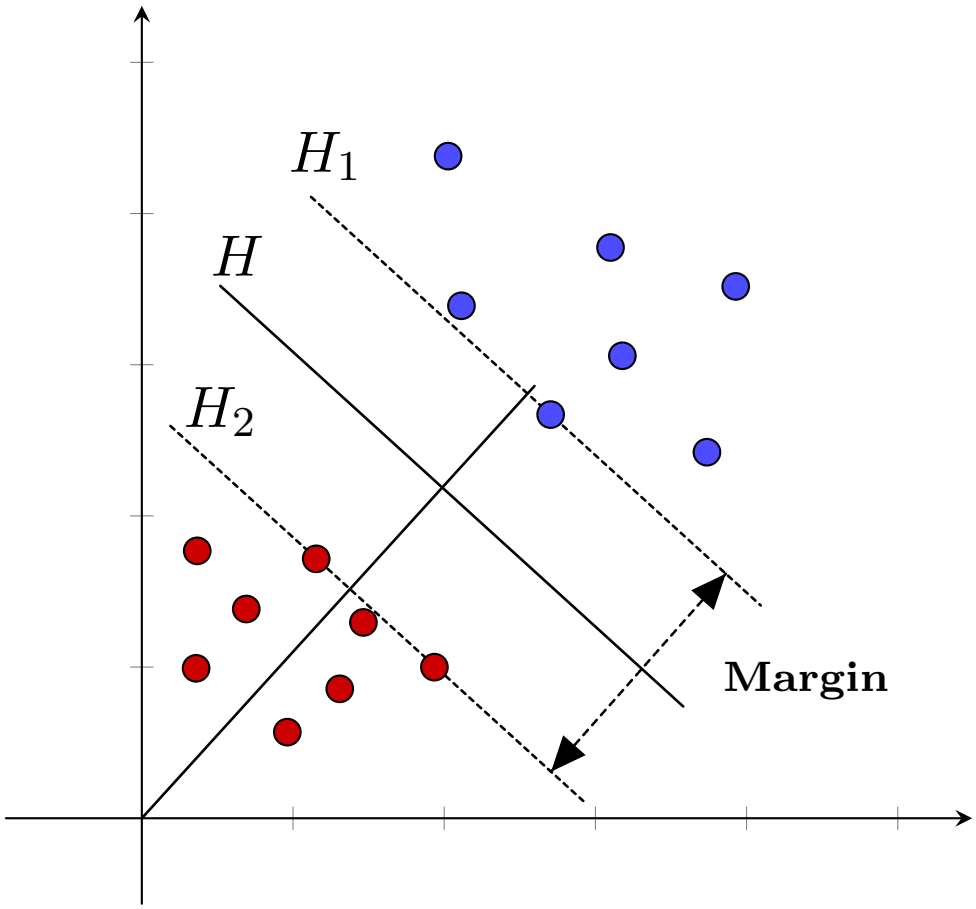

Figure 3.2: Illustration of Support Vector Machine Method. Points in blue are of class 1 ($y = 1$) while points in read are of class 2 ($y = -1$).

Assume we are given a training set $\mathcal{D}$ with N observations (x_i, y_i), with $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $y_i \in \{-1, 1\}$. Now, let's assume that the two classes are separable, i.e. we can find $\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p$ such that the hyperplane $H : x^\top \beta + \beta_0 = 0$ completely separates the two classes. Let H_1 and H_2 be two hyperplanes parallel to H such that the distances between the closest points from each class to H are the same as the distances between H and H_1, H_2. Observe that we can scale β and β_0 so that H_1 has the form $x^\top \beta + \beta_0 = 1$ and H_2 has the form $x^\top \beta + \beta_0 = -1$. We can define a classification rule as $f(x) = \text{sign}(x^\top \beta + \beta_0)$.

We now define the *margin* M to be the distance between H_1 and H_2. Since the distances between H_1 and H_2 are $\frac{|\beta_0 - 1|}{\|\beta\|}$ and $\frac{|\beta_0 + 1|}{\|\beta\|}$, respectively. Hence, $M = \frac{2}{\|\beta\|}$. The goal, then is to find β_0, β such that we have the largest margin between the 2 classes. Notice from Figure 3.3

that for any point with $y_i = 1, x_i^\top \beta + \beta_0 \geq 1$ and for any point with $y_1 = -1, x_i^\top \beta + \beta_0 \leq -1$.

The problem of finding the best hyperplane H can be formulated as follows.

$$\max_{\beta_0, \beta} M$$
$$\text{subject to } y_i(x_i^\top \beta + \beta_0) \geq 1,\ i = 1, 2, \cdots N.$$

Since $M = \frac{2}{\|\beta\|}$, we can rewrite this optimization problem as

$$\min_{\beta_0, \beta} \|\beta\|$$
$$\text{subject to } y_i(x_i^\top \beta + \beta_0) \geq 1,\ i = 1, 2, \cdots N.$$

Now let's tackle the general case when the two classes are not separable. While we still want to find a hyperplane that creates the largest margin between the 2 classes, we should also control the number of misclassifications. To accomplish that, we could rewrite the optimization problem as

$$\min_{\beta_0, \beta} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i$$
$$\text{subject to } \xi_i \geq 0, y_i(x_i^\top \beta + \beta_0) \geq 1 - \xi_i,\ i = 1, 2, \cdots N.$$

where ξ_i represents the relative distance by which the prediction $\hat{y}(x_i)$ is on the wrong side of the margin. In particular, when $0 < \xi_i < 1$, the observation is on the correct side of hyperplane but wrong side of the margin and when $\xi > 1$, the point is on the wrong side of the hyperplane and is misclassified. The extra term in the objective function represents a trade-off between having the largest margin between the 2 classes and having a small number of misclassifications. In particular, the parameter C measures how much we want to avoid misclassification. Note that when $C = \infty$, we are back to the separable case. As the objective function is quadratic and all the constraints are linear, this is a convex optimization problem; thus, it can be solved in polynomial time as pointed out in Section 2.1.

3.3.5 Neural Network

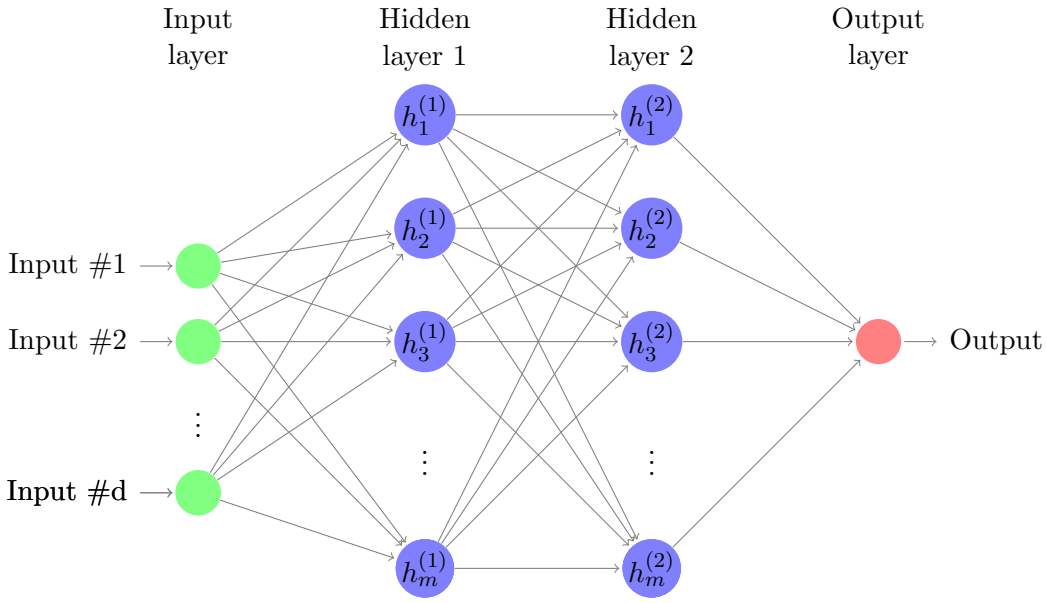

Figure 3.3: Illustration of a Neural Network with 2 hidden layers and m nodes per hidden layer.

Recall from our discussion on logistic regression method that after having trained the model, given a test input x, we can compute $p(Y = 1|X = x)$ and $p(Y = 0|X = x)$ and assign to it the class label that corresponds to the higher probability. Calculating the conditional probability $p(Y = k|X = x)$ in this case can be thought of as 2-step process: We first calculate the linear score function $\beta^\top x + \beta_0$ and then applies the sigmoid function to get a probability value. This can be extended to a multiclass classification problem as follows: For each x, we apply the learned linear score function to each class k and get a vector of linear scores in $\mathbb{R}^K$, asssuming there are K classes. Then, instead of using the sigmoid function, we use its more general version - the *softmax function* - which has the form:

$$\text{Softmax}(z_1, z_2, \cdots, z_K) = \left(\frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \cdots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right).$$

The values after applying the softmax function can be thought of as the conditional probability for each class and we can take the class with the highest probability as our prediction.

Neural Nework (or artificial neural network) can be thought of as a more general version of logistic regression as it allows us to learn highly nonlinear functions by apply nonlinear

score functions to the inputs instead of the linear score function as in logistic regression. A neural network is typically represented by a network diagram, with many layers and nodes in each layer. The first layer is the input layer where inputs are entered and the last layer contains all the possible outputs. All layers in between are called hidden layers. For ease of exposition, we now assume every hidden layer has m nodes.

Assuming the input space is in $\mathbb{R}^p$, the value of each node of the first hidden layer is given by

$$h_i^{(1)}(x) = \sigma((w^{(1)^\top} x)_i + b_i^{(1)}),$$

where σ is called the activation function - usually $\sigma(x) = \tanh x$ or $\sigma(x) = \max\{0, x\}$ - and $w^{(1)} \in \mathbb{R}^{m\times p}, b^{(1)} \in \mathbb{R}^m$ are parameters. Then, the value of the node of each subsequent hidden layer is

$$h_i^{(j)} = \sigma((w^{(j)^\top} h^{(j-1)})_i + b_i^{(j)}),\ j = 2, 3, \cdots L,$$

where $w^{(j)} \in \mathbb{R}^{m\times m}, b^{(j)} \in \mathbb{R}^m$, and L is the number of hidden layers in the network. Finally, for the last layer, we first apply an affine mapping: $a = w^{(D+1)^\top} h^{(D)} + b^{(D+1)}$, where $w^{(D+1)} \in \mathbb{R}^{m\times K}, b^{(D+1)} \in \mathbb{R}^K$. Afterwards, we apply the softmax function to get the K conditional probability $p(Y = k|x), k = 1, \cdots, K$. We can also see that the nonlinear score function can be expressed as a composition of functions (or layers):

$$f(x) = \left(a \circ h^{(D)} \circ \cdots \circ h^{(1)}\right)(x).$$

Let $\theta = (w^{(1)}, \cdots, w^{(D+1)}, b^{(1)}, \cdots, b^{(D+1)})$, we can write the log likelihood function as

$$l(\theta) = \sum_{i=1}^{n} \log\left[\text{Softmax}(f(x_i))_{y_i}\right],$$

where the y_i subscript means taking the y_i entry in the $\mathbb{R}^K$ vector. We then find θ that maximizes this likelihood function, using a technique called backpropagation. Readers are referred to [54, Chapter 6] for more detail on how to implement this algorithm.

Intuitively, we can see that the more layers and nodes in each layer a neural network has, the more capable the network is to approximate any function. Networks with multiple

hidden layers are given a special name - *Deep neural network*. In addition, there are many classes of artificial neural networks such as convolutional neural network, recurrent neural network, and long short-term memory network, that are developed to meet different needs.

Chapter 4

Learning to cut

4.1 Introduction

As demonstrated in Section 2.4, one can obtain a cutting plane for solving an MILP by working with a maximal $(b + \mathbb{Z}^n)$-free convex set. However, there might be an infinite number of such sets so it is not clear how one could decide which set to use. Recently, the authors in [9] study a particular family of $(b+\mathbb{Z}^n)$-free convex set, the family of generalized cross-polyhedra, and find that this family gives a good approximation of the closure obtained by using all cut generating functions derived from all maximal $(b + \mathbb{Z}^n)$-free convex sets. In addition, the cuts obtained from this family can provide some tangible improvement on a reasonable fraction of problems over the commonly used GMI cuts, which was introduced in Remark 2.14. However, since the generation of cuts derived from this family is computationally more expensive compared to GMI cuts, one would want to invest the time and effort to generate these cuts only for those instances where we expect to get a significant advantage beyond GMI cuts.

Let $\mathcal{X}$ denote the set of all possible Mixed-Integer Linear Programming (or Pure-Integer Linear Programming) instances $(A, b, c, \mathcal{I})$ (for fixed dimensions for the matrix A and vectors b, c and cardinality of the set $\mathcal{I}$) in standard form:

$$
\begin{aligned}
\min \quad & c^\top x \\
\text{subject to:} \quad & \\
& Ax = b \\
& x \geq 0 \\
& x_i \in \mathbb{Z},\ \forall i \in I,
\end{aligned}
\tag{4.1}
$$

where $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, I \subseteq [n]$ are given.

The first goal is now to use machine learning algorithms to classify, just by looking at the instance spaces A, b, c, on which types of problems do the generalized crosspolyhedral cuts perform better. We attempt to achieve this by using 2 approaches. The first approach is to compute d features, which are mappings $F : \mathcal{X} \to \mathbb{R}^d$ that maps every instance in $\mathbb{X}$ to a feature vector. We then attempt to learn a classifier $C : \mathbb{R}^d \to \{0, 1\}$, where the value 1 means applying generalized crosspolyhedral cuts on the problem will give an improvement over the GMI cuts. The second approach is to learn directly from the instance space, i.e. a mapping $F : \mathcal{X} \to \{0, 1\}$. The technique we use to classify these instances is similar to the idea in Convolutional Neural Network (CNN), which is a class of deep neural networks.

In addition, we try to select the best parameters for these cuts, instead of simply generating them randomly as was done in [9]. In particular, the family of generalized crosspolytope can be parameterized by a tuple (f, μ), where f is the center of the generalized crosspolytope and μ can be thought of as a scaling factor. In the previous paper, (f, μ) was randomly generated. However, for this project, we attempt to learn a function $F(D, f, \mu) = y$, where D is the problem data and y is the improvement of the generalized crosspolyhedral cuts over GMI cuts. Having obtained this function F, we could use optimization routines to learn the best (f, μ) to give the maximum improvement, given an instance D.

4.2 Problem setup

Data generation. Our instance set/data set is generated from some (known) distribution $\mathcal{D}$ on $\mathcal{X}$. We first sample A, b, c from a known distribution to generate the pure integer instances. To make such an instance into a mixed-integer problem, we randomly choose

each variable to be continuous or integer with equal probability.

Cut generation procedure. Recall from Section 2.4 that to obtain valid inequalities for the set $X(R, P)$ as introduced in 2.7, we could start with a maximal $(b+\mathbb{Z}^n)$-free convex set K because (ψ, ψ), where ψ is the gauge function of K, is a valid pair. Recently, the authors in [9] suggest using *generalized cross-polyhedra*, which is a family of maximal lattice-free sets. This family can be parameterized by a tuple $(f, \mu) \in \mathbb{R}^N \times \mathbb{R}^N$, for some fixed natural number N (the number of rows used in generating the cuts). Recall from Remark 2.11 that N can range from 1 to the maximum number of rows whose corresponding basic variables violate their integrality constraint. More precisely, the procedure to compute these pairs is as follows:

1. Choose a natural number N as the number of rows to be taken from the final simplex tableaux to generate the cutting planes. This will also be the dimension of the generalized cross-polytope.

2. Randomly generate $f \in \mathbb{R}^N$, which is the center of the generalized cross-polytope.

3. Choose a random vector $\mu \in \mathbb{R}^N$ from the simplex $\sum_i \mu_i = 1$.

4. From the parameters f and μ, [9] defines a generalized cross-polytope B such that $\text{int}(B) \cap (b + \mathbb{Z}^N) = \emptyset$ and $0 \in \text{int}(B)$.

5. Then, $\sum_{i=1}^{k} \psi_B(r_i)s_i + \sum_{i=1}^{\ell} \widetilde{\psi_{b+B}}(p_i)y_i \geq 1$ is a valid inequality for (**??**), where $\psi_B(r)$ is the gauge function of B and $\widetilde{\psi_{b+B}}$ is the so-called *trivial lifting* of $b + B$. If $f = 0$, we call this an X cut. If not, we obtain a GX cut.

Main observations from [9] relevant for this thesis. A typical measure used to compute the performance of cuts is *gap closed* which is given by

$$\frac{\mathsf{cut} - \mathsf{LP}}{\mathsf{IP} - \mathsf{LP}},$$

where:

cut is the objective of LP relaxation after applying the cut,

LP is the objective of LP relaxation of MILP,

IP is the optimal value of MILP.

However the IP optimal value IP could be expensive to compute on our instances. So the authors use a different metric, which measures the *improvement* over GMI cuts using the new cuts. They define

$$\beta \quad := \quad \frac{\mathsf{Best} - \mathsf{GMI}}{\mathsf{GMI} - \mathsf{LP}}, \tag{4.2}$$

where:

LP is the objective of LP relaxation of MILP.

GMI is the objective of LP relaxation of MILP with GMI cuts on all rows whose corresponding basic variables do not satisfy the integrality constraint.

Best is the maximum objective of MILP with all the GX and X cuts as well as the GMI cuts.

The computational procedure was run with $N = 2, 5$, and 10 rows. In mixed-integer problems, $\beta \geq 10\%$ in roughly 10% of the set of mixed-integer problems. In pure-integer problems, $\beta \geq 5\%$ in roughly 5% of the set of pure-integer problems. This suggests that the problem of deciding whether a given instance will stand to gain from using this cut generation procedure or not is worth studying, given the fact that it is slower to compute these cuts than the GMI cut. In addition, even after narrowing down to the specialized family of generalized crosspolyhedra, they are not still able to address the "cut selection" problem in practice, as the procedure to select cutting planes from this family is through randomization.

This motivates two questions:

1. Can we find the best parameters (f, μ) for the subfamily of generalized cross-polytopes that would give us the highest improvement over the GMI cut, instead of just **randomly generated** these parameters as in [9]?

2. Given a new instance $(A, b, c, \mathcal{I})$, can we know if our cut generation procedure will give an improvement over the GMI cut or not? This is important because there is

a bit of overhead in computing the slightly more expensive cuts, specially given that we will also need to optimize the parameters for these cuts. This will address the question of "whether to deploy these cuts or not".

This chapter will attempt to address both of these questions.

4.3 Finding the best parameters for generalized crosspolyhedral cuts

As mentioned before, the authors in [9] used a very naive random sampling method for their family of generalized crosspolyhedra, which results in not quite encouraging computational results. The natural next step would be to find the best parameters for this family of cuts to maximize the effectiveness of these cuts. This addresses the "cut selection" problem, which is a concern in the previous paper.

We first learn a "scoring function" $\mathcal{F} : (rows, \mu, f) \rightarrow \mathbb{R}_{\geq 0}$ that measures the improvement over the GMI cut using our cut for each instance, where *rows* represents the rows of the final simplex tableaux used to compute the cuts. We then use some standard optimization methods to optimize $\mathcal{F}$ over μ and f, while holding *'rows'* fixed. The idea is to find the best parameters that will give the most improvement for a new instance, from which some rows have been pulled out for generating the cuts.

For the first step, to keep the input size fixed for the model, we only consider 2-row cuts, i.e., $N = 2$. The inputs variables for F are two random rows from the final simplex tableaux, μ, and f. In addition, instead of using β as our target variable, we use a modified version of this - we measure the percentage gain of the objective of LP relaxation with our cut over the objective of just LP relaxation, which we denote by $\beta_{modified}$. The reason is that for 2-row cuts, our experimental results show that $\beta = 0$ for more than 95% of the problems. The highly imbalanced data can make it harder for models to learn. Using $\beta_{modified}$, we hope to have a more balanced distribution for the target variable. We then create a deep neural network architecture to learn this mapping, as plotted below.

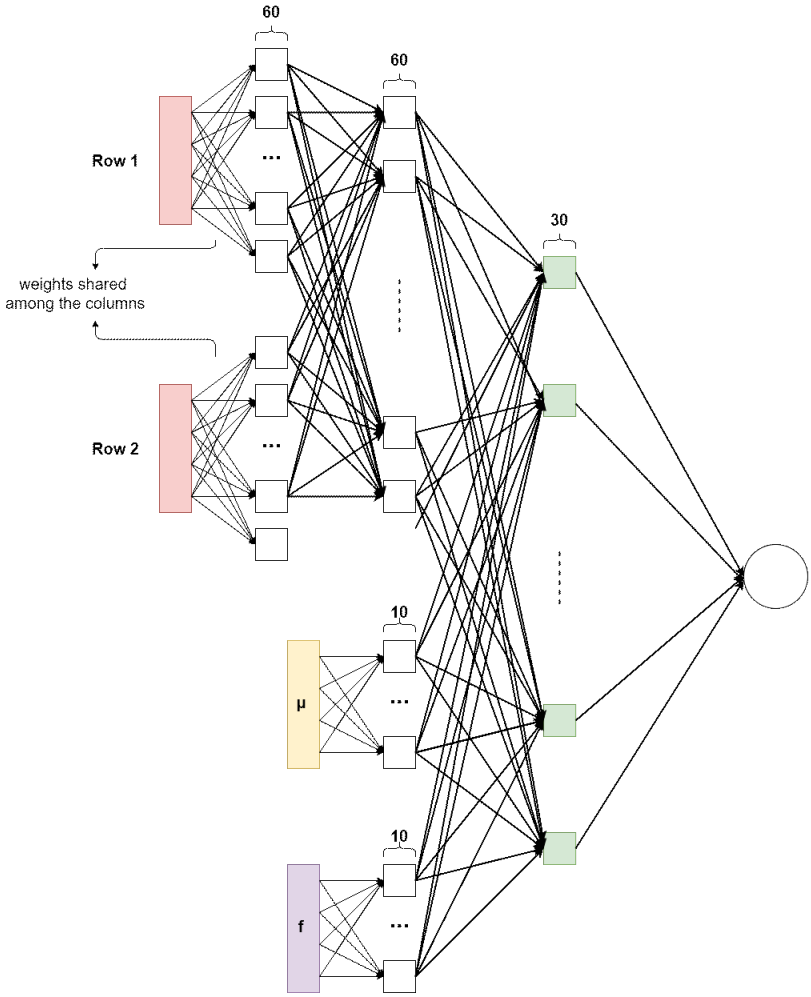

Figure 4.1: Deep Neural Network Structure. The numbers above each hidden layer represent the number of nodes in that layer. For example, there are 30 nodes in the last hidden layer.

Having learned this mapping $F(rows, \mu, f) \rightarrow \beta_{modified}$, we then use a constrained optimization technique in Scipy.optimize to find the best parameters for μ and f to maximize our target variable. We first use a set of instances with random (f, μ) to train the DNN model. In addition, we record the $\beta_{modified}$ on these random instances as $\beta^{original}_{modified}$. Having trained the DNN model, we then use it to find the optimized parameters for each instance. As a final step, we feed these optimized values for μ and f into our cut generation procedure to get the optimized values $\beta^{optimized}_{modified}$ and compare the average gain over $\beta^{original}_{modified}$. The idea is to see how much improvement do we make after using optimized parameters to generate the cuts. Furthermore, we repeat this whole computation for a completely new set of instances that was not used to train the DNN model. Doing so allows us to validate if our

scoring function F can generalize to unseen instances. The results for both the training set (with 19200 data points) and test set (with 4800 data points) are reported below.

Pure (Training)	Pure (Test)
124.62%	0.98%

Table 4.1: Average gain of $\beta_{modified}$ using optimized parameters for our cut

Having observed the big difference between the training set and the test set, it seems that our scoring function does not generalize well to new instances. Even for the training set, the variance for the **average gain of** $\beta_{modified}$ is enormous, with the median value of -0.45%.

With this new observation, we went back and took a closer look at the "scoring function". It turned out that the MSE obtained for both the training and validation sets during the training of our DNN model is virtually the same as the variance among the target variable itself. This means a simple strategy of predicting just the average value over the target variable does just as well as our model.

In an attempt to improve the model, we try two natural approaches:

- Varying the complexity of the model by either using a more complex model, with more hidden layers and neurons in each layer, or a simpler one.
- Add more features for the input layer. In particular, we also use the matrix A and vectors b, c.

Unfortunately, these approaches are not able to improve the current model, as the MSE value stays the same.

4.4 Classifying problem instances based on the effectiveness of generalized cross-polyhedral cuts

Since our attempt to find the best parameters for generalized crosspolyhedral cuts was not a successful one, for our next step of classifying problem instances based on the effectiveness of these cuts, we will just use randomized values, instead of optimized values as originally desired, for the parameters μ and f.

Label generation. A natural idea to answer the second problem using ideas from statistical and machine learning is to cast this as a binary classification problem. In particular, given any particular instance, we could run the randomized cut generation procedure as in [9] to obtain a value for β. Based on some predetermined thresholds for β, we label an instance as 0 or 1 (0 indicates no improvement, while 1 indicates some improvement). In particular, we consider 4 different problem scenarios: "Pure 0", "Mix 0", "Pure 5", "Mix 5" - the first word denotes whether the problem is a pure or mixed integer problem and the number represent the threshold used for β. Specifically, 0 means an instance will take the label 1 if $\beta > 0$ and 5 means an instance will take the label 1 if $\beta > 5\%$. In particular, for all these scenarios, we have 7174 observations in the training set and 1794 in the test set. Setting $\mathcal{Y} = \{0, 1\}$ we now obtain a joint distribution $\mathcal{K}$ on $\mathcal{X} \times \mathcal{Y}$. We denote the derived conditional distribution on $\mathcal{X}$ when the label is 0 as $\mathcal{D}_0$ and respectively the distribution $\mathcal{D}_1$ for label 1. Due to the randomization in our cut generation procedure, there is some inherent noise in this labeling mechanism. This could make our problem "unclassifiable". Hence, we first check whether this is a classifiable problem.

To do that, for every MILP (and PILP) instance we generate, we run the randomized cut generation process multiple times, and see if for "most" instances, the label 0 or 1 has dominant probability (say greater than 70%). If the majority of the problems are either 0 or 1 with high probability, we can be confident that the class labels depend on the problems themselves and not on our cut generation procedure. In that case, there is hope that we can learn from the features of the MILP (or PILP) instances to obtain their class labels.

Pure 0	Mix 0	Pure 5	Mix 5
95.70%	98.77%	99.32%	97.00%

Table 4.2: Proportion of instances with high probability of dominant class

Table 4.2 gives for each type of problem the proportion of instances where the dominant class has more than 70% chance of being labeled accordingly. From the table, we can see that for all 4 problems considered, more than 95% of the instances have a high probability of being labeled with the dominant class.

More rigorously, if we could find a map $C : \mathcal{X} \to \mathcal{Y}$ such that $\mathbf{P}_{(x,y)\sim\mathcal{K}}[C(x) \neq y]$ is

small, then this would be a good certificate. Even though we might not be able to construct such a map, we could "prove" the *existence* of such a thing by estimating the Bayes error rate (see Section 3.3 for a formal definition of the Bayes error rate).

To approximate the Bayes error rate, we run the randomized cut generation process 100 times for each problem instance. The dominant label out of these 100 times is taken as the instance's "true label". From there, we can count the number of True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), and consequently the accuracy rate and the error rate coming from this procedure. For example, if for all the instances in the dataset, the dominant class shows up 100% of the times, we have a perfect model and the error rate is 0. On the other hand, if for all the problems, the labels 0 or 1 have the same probability of coming up, the error rate is at its worst at 50%.

Pure 0	Mix 0	Pure 5	Mix 5
7.57%	2.82%	1.31%	5.84%

Table 4.3: Approximated Bayes Error Rates

With the low estimated Bayes error rates as reported in Table 4.3, we can be confident that the problem is classifiable. Having established this fact, the next natural question to ask is if this problem is "learnable", i.e., whether the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ are "sufficiently" different. To estimate $\mathcal{D}_0$ and $\mathcal{D}_1$, we generate lots of MILP (and PILP) instances from $\mathcal{D}$ and for each instance generate lots of random labels using the randomized procedure. Thus, we now have labeled instances from our joint distribution on $\mathcal{X} \times \mathcal{Y}$. If we look at all the data points with label 0, they will be samples from the distribution $\mathcal{D}_0$ and similarly for label 1. Now we can look at different statistics of these samples from $\mathcal{D}_0$ and $\mathcal{D}_1$ to see if there's a difference.

Since we have a multidimensional data comprising of the data for matrices A, b, and c, we use the Multivariate Nonparametric Cramer Test for the two sample problem to compare $\mathcal{D}_0$ and $\mathcal{D}_1$. The null hypothesis is that the 2 samples come from the same distribution. The p value for each problem is shown below.

It should be noted though that the results for the Mix 0 and Pure 5 cases should be taken with some skepticism because the Cramer Test is susceptible to the size of the dataset

Pure 0	Mix 0	Pure 5	Mix 5
0.43	0.00	0.00	0.99

Table 4.4: p-values of the Multivariate Nonparametric Cramer Test

and we have a relatively small dataset for these 2 problems due to the highly unbalanced distribution of the 2 classes.

With some caveats, the problem seems "learnable". We proceed to use some ideas from machine learning to do the classification problem.

Feature selection. We decide on a natural number d of features and a mapping $F : \mathcal{X} \to \mathbb{R}^d$ that maps every instance in $\mathcal{X}$ to a *feature vector*. This then induces a joint distribution $\hat{\mathcal{K}}$ on $\mathbb{R}^d \times \mathcal{Y}$, and corresponding marginal distributions $\hat{\mathcal{D}}_0$ and $\hat{\mathcal{D}}_1$. In particular, we have 105 features, the majority of which are often used in similar studies in the literature [see [98], [65]].

I. Problem Size Features:
1-2. Number of variables and constraints
II. Variable Type Features:
3-4. Number of Integer variables and percentage of Integer variables
III. Variable-Constraint Graph Features:
5-10. Variable node degree statistics: mean, std, min, max, 25th percentile, and 75th percentile
11-16. Constraint node degree statistics: mean, std, min, max, 25th percentile, and 75th percentile
IV. Variable Graph Features:
17-22. Node degree statistics: max, min, std, 25th percentile, and 75th percentile
23. Edge Density: number of edges in the VG divided by the number of edges in a complete graph having the same number of nodes
V. LP-Based Features:
24-30. Integer Slack 1 vector statistics: min, max, std, L2-norm, 25th percentile, 75th percentile, and number of nonzeros elements. Integer slack vector 1 is a vector of size equal to the number of integer variables in the problem. For each integer variable x_i, this contains the number x_i - np.floor(x_i)
31-36. Integer Slack 2 vector statistics: min, max, std, L2-norm, 25th percentile, and 75th percentile. Integer slack vector 2 is a vector of size equal to the number of variables in the problem. If x_i is a continuous variable, then the i-th coordinate of this vector is 0, else the i-th coordinate is x_i - np.floor(x_i)
37. Objective function value of LP solution
VI. Objective Function Features:
38. Standard deviation of normalized coefficients: c_i/m
39-40. Standard deviation of c_i/n_i and $c_i/\sqrt{n_i}$ where n_i denotes the number of nonzero entries in column i of A
VII. Linear Constraint Matrix Features
41-42. Distribution of normalized constraint matrix entries, Aij/b_i: mean and std (only of elements where $b_i \neq 0$
43-44. Variation coefficient of normalized absolute nonzero entries per row: mean and std
45-48. Min/max for ratios of constraint coeffs. to RHS: Min and Max ratios across positive and negative right-hand-sides
49-64. Min/max/mean/std for one-to-all coeff ratios: The statistics are over the ratios of a variable's coefficient, to the sum over all other variables' coefficients, for a given constraint. Four versions of these ratios are considered: positive (negative) coefficient to sum of positive (negative) coefficients
VIII. Single Probing Features
65-70. Presolving features: CPU times for presolving and relaxation, # of constraints, variables, nonzero entries in the constraint matrix, and clique table inequalities after presolving
71-83. Probing cut usage features: number of each of 12 different cut types, and total cuts applied
84-85. Number of iterations and number of nodes
IX. Geometric features:
86-107. Mean/std/coefficient of variation of norm of Aeq; Mean/variance/std/coefficient of variation of the inner products between the columns of the normalized matrix Aeq; Mean/variance/std/coefficient of variation of the angle between these columns

Table 4.5: Feature List

In order to check if our feature selection is good enough, we run a Kolgomorov-Smirnov 2 sample test for each of our features. Using a Bonferroni's corrected significant level of 0.05/105, the number of features that have significant differences between the distributions for class 0 and class 1 is shown in Table 4.6.

Pure 0	Mix 0	Pure 5	Mix 5
4	11	1	9

Table 4.6: Number of "significant" features

From the Kolgomorov-Smirnov test, our feature sets do not seem to capture all the relevant information from our instance space $\mathcal{X}$ as there are only a small number of significant features. However, there is some hope that one can derive a good classifier using these features if they indeed well separate the 2 classes. In addition, from the K-S test, we could only say that individually, these features cannot differentiate the two classes. However, it is possible that certain combinations of these features might be able to draw a distinction.

We can now run different standard classifiers. A classifier is simply a map $C : \mathbb{R}^d \to \mathcal{Y}$. This involves the following:

We have to decide on a family $\mathcal{C}$ of classifiers (k-NN, neural nets, etc.), and we have to decide on a rule that for every $n \geq \mathbb{N}$, and any sample $\{(x_i, y_i)\}_{i=1}^n$ from the distribution $\hat{\mathcal{K}}$, gives an element $C_n \in \mathcal{C}$.

We have decided to use the following families of classifiers:

- Logistic regression
- Random Forests
- Xgboost [30]
- SVM
- k-NN (k nearest neighbor)
- Neural Networks

The rule that we use is some penalized version of the empirical risk minimization principle, which was introduced in Section 3.3:

For any natural number n and n data samples $\{(x_i, y_i)\}_{i=1}^n$, we first solve the following minimization problem (exactly or approximately) for the family $\mathcal{C}$:

$$C_n(\lambda) = \arg\min_{C \in \mathcal{C}} \sum_{i=1}^{n} \ell(C, x_i, y_i) + \lambda r(C)$$

where $\ell : \mathcal{C} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a "loss function" that evaluates "how well the classifier C did on a given sample point (x, y)", $r : \mathcal{C} \to \mathbb{R}$ is a "regularization" term and $\lambda \in \mathbb{R}_+$ is a "regularization parameter". We finally find $C_n(\lambda)$ for many different values of λ and report the "best" one. In order to determine the best model (i.e. the best λ), we train each model on a training set and evaluate its performance on another cross validation set. The best model is the one with the best performance on the validation set. One can define the following reasonable measures to quantify a classifier's performance:

- Accuracy: This is simply $\mathbf{P}_{(x,y)\sim\hat{\mathcal{K}}}[C(x) = y]$. It might not be a good measure when the marginal probability of 0 or 1 labels are skewed.
- Balanced accuracy: $\frac{1}{2}\mathbf{P}_{x\sim\hat{\mathcal{D}}_0}[C(x) = 0] + \frac{1}{2}\mathbf{P}_{x\sim\hat{\mathcal{D}}_1}[C(x) = 1]$.
- Precision/Recall/F1-score for each label.

For our problem, the performance is measured using the balanced accuracy score, which is the average of recall obtained on each class. This is a better criterion than the traditional accuracy score as because for our unbalanced dataset, it punishes classifiers that always predict the majority class, which gives a high accuracy score just because of the structure of the classes. Also, it is intuitively easier to understand than the F1-score.

The best model from each family of classifiers is then compared against one another using a test set that is different from both the training and cross validation sets described above to obtain the best "overall" model for our classification problem.

The procedure for model building is described in the next section.

Heuristics used to boost model performance. As a point of reference, all the classifiers are run with the full feature set included. This serves as our base case.

However, it is well-known that redundant features can decrease the generalization performance on the test set. Hence, to improve our model, we employ some feature selection methods. In addition, to combat the problem of high imbalanced dataset, we also use some oversampling/undersampling techniques. In particular, the procedure for model building is outlined below, where the classifiers are run after each of these steps:

1. First, we plot the histograms for all the features to understand their distribution. We then calculate all the pairwise correlations and remove one feature from those pairs with high correlation (0.9 threshold). In addition, we also drop those features with unique value.

2. Next, we try 3 different oversampling/undersampling techniques:

    - Resampling: Resample the minority class with replacement, until we have the same number of data in each class
    - Synthetic Minority Over-sampling Technique (SMOTE): In essence, SMOTE is a technique to create synthetic data from the existing dataset to balance out the two classes. In particular, we create a new data point along the line connecting 2 points from the same class.
    - SMOTE + Edited Nearest Neighbor (ENN): Not only do we synthetically create more data for the minority class, we also reduce the size of the majority class using ENN. It removes examples whose class label differs from the class of at least half of 3 nearest neighbors.

3. We then try a more advanced feature selection method with our original dataset and then perform step (b) again.

The other feature selection method used is a package in Python called feature-selector, which is created by Will Koehrsen. Some feature selection methods included in this package are as follow:

- Remove features with only a unique value.

- Remove features with missing values: Drop features with more than 60% missing values.

- Remove highly correlated features: Identify pairs of features with a correlation coefficient magnitude of greater than 0.98 and drop one from the pair.

- Remove features with "low importance" using a gradient boosting machine (GBM) learning model. Using GBM, one can identify the relative importance of the features, then find and remove the least important features not required to achieve 99% of the cumulative feature importance.

It seems clear that the major difference between our own feature selection method and the "feature-selector" package is the addition of the Gradient boosting trees method. Hence, we will abbreviate this package as "GBT".

Hyperparameters in model building. It is well known that the majority of machine learning algorithms require hyperparameters, the most common of which is the regularization constant. These hyperparameters are determined by doing cross validation. Since an algorithm may have several hyperparameters and running a grid search over all of those requires sizable computing powers, we only look at a couple of those in cross validation. The hyperparameters considered for each kind of classifier are as follows:

- Logistic Regression: 'C' (regularization parameter); 'penalty' (penalty type - L1 or L2).

- Random Forest: 'n_estimators' (number of trees in the forest); 'max_depth' (maximum depth of a tree).

- Xgboost: 'learning_rate' (learning rate); 'n_estimators' (number of trees); 'min_child_weight' (minimum sum of weights of all observations required in a child); 'gamma' (minimum loss reduction required to make a node split); 'subsample' (fraction of observations to be randomly samples for each tree); 'colsample_bytree' (fraction of columns to be randomly samples for each tree); 'max_depth' (maximum depth of a tree).

- SVM: 'C' (regularization parameter).
- k-NN: 'n_neighbors' (number of neighbors).
- Neural Network: 'alpha' (regularization parameter); 'hidden_layer_sizes' (architecture of the network).

The results for all our models are included below. ROC plots are also shown in the appendix.

		LogReg	**RF**	**XGB**	**SVM**	***k*-NN**	**NeuralNet**
Pure 0	All features	**0.69**	0.66	0.69	0.61	0.50	0.64
	FS	**0.70**	0.66	0.69	0.62	0.66	0.70
	FS + RS	0.62	0.66	**0.66**	0.62	0.60	0.64
	FS + SMOTE	0.63	0.66	**0.67**	0.61	0.56	0.61
	FS + SMOTEENN	0.49	0.58	0.56	**0.59**	0.48	0.55
	GBT	**0.70**	0.66	0.70	0.62	0.66	0.69
	GBT + RS	0.60	**0.65**	0.65	0.62	0.61	0.61
	GBT + SMOTE	0.62	0.65	**0.67**	0.61	0.57	0.59
	GBT + SMOTEENN	0.48	**0.57**	0.56	0.60	0.49	0.54
Mix 0	All features	0.90	0.88	**0.90**	0.63	0.90	0.90
	FS	**0.90**	0.88	0.90	0.63	0.80	0.90
	FS + RS	0.66	0.84	0.84	0.65	0.85	**0.88**
	FS + SMOTE	0.67	0.81	**0.89**	0.65	0.73	0.87
	FS + SMOTEENN	0.48	0.74	**0.78**	0.61	0.60	0.77
	GBT	**0.90**	0.87	0.90	0.64	0.80	0.90
	GBT + RS	0.67	0.83	0.84	0.65	0.85	**0.87**
	GBT + SMOTE	0.67	0.82	**0.88**	0.64	0.72	0.86
	GBT + SMOTEENN	0.48	0.73	0.74	0.61	0.59	**0.76**
Pure 5	All features	0.94	0.93	**0.94**	0.22	0.93	0.92
	FS	0.94	0.93	**0.94**	0.51	0.89	0.94
	FS + RS	0.58	0.89	0.90	0.54	0.89	**0.91**
	FS + SMOTE	0.55	0.86	**0.93**	0.49	0.80	0.90
	FS + SMOTEENN	0.31	0.79	**0.85**	0.48	0.70	0.83
	GBT	0.94	0.93	**0.94**	0.53	0.89	0.94
	GBT + RS	0.58	0.88	**0.90**	0.53	0.89	0.90
	GBT + SMOTE	0.55	0.88	**0.93**	0.51	0.80	0.89
	GBT + SMOTEENN	0.29	0.80	**0.86**	0.49	0.69	0.81
Mix 5	All features	**0.60**	0.57	0.58	0.53	0.54	0.57
	FS	**0.60**	0.57	0.58	0.58	0.57	0.60
	FS + RS	**0.58**	0.57	0.57	0.58	0.54	0.55
	FS + SMOTE	**0.59**	0.56	0.57	0.58	0.53	0.54
	FS + SMOTEENN	0.51	0.53	0.52	**0.56**	0.52	0.52
	GBT	**0.60**	0.57	0.58	0.57	0.58	0.60
	GBT + RS	**0.58**	0.56	0.57	0.58	0.55	0.53
	GBT + SMOTE	**0.58**	0.57	0.58	0.58	0.53	0.54
	GBT + SMOTEENN	0.50	0.51	0.52	**0.57**	0.48	0.54

Table 4.7: Accuracy Scores of our Classifiers

		LogReg	**RF**	**XGB**	**SVM**	***k*-NN**	**NeuralNet**
Pure 0	All features	0.56	0.60	0.53	**0.61**	0.52	0.50
	FS	0.57	0.61	0.55	**0.63***	0.57	0.56
	FS + RS	0.62	0.61	0.60	**0.62**	0.54	0.55
	FS + SMOTE	**0.62**	0.62	0.59	0.61	0.53	0.55
	FS + SMOTEENN	0.59	0.61	0.60	**0.61**	0.55	0.57
	GBT	0.56	0.61	0.54	**0.62**	0.58	0.57
	GBT + RS	0.62	0.60	0.59	**0.62**	0.54	0.53
	GBT + SMOTE	**0.62**	0.61	0.60	0.62	0.55	0.56
	GBT + SMOTEENN	0.58	0.61	0.61	**0.62**	0.56	0.57
Mix 0	All features	0.50	0.52	0.50	**0.65**	0.51	0.50
	FS	0.51	0.53	0.50	**0.67***	0.55	0.51
	FS + RS	0.66	0.58	0.53	**0.67**	0.53	0.53
	FS + SMOTE	**0.65**	0.58	0.52	0.65	0.53	0.53
	FS + SMOTEENN	**0.66**	0.64	0.60	0.65	0.54	0.58
	GBT	0.51	0.52	0.50	**0.67**	0.55	0.50
	GBT + RS	**0.67**	0.58	0.55	0.67	0.53	0.54
	GBT + SMOTE	**0.66**	0.58	0.53	0.66	0.54	0.53
	GBT + SMOTEENN	0.64	0.64	0.63	**0.66**	0.54	0.59
Pure 5	All features	0.50	0.50	0.50	**0.50**	0.50	0.50
	FS	0.50	0.50	0.50	**0.50**	0.50	0.50
	FS + RS	0.50	0.50	**0.51**	0.51	0.50	0.49
	FS + SMOTE	0.47	0.49	**0.51**	0.49	0.49	0.50
	FS + SMOTEENN	0.49	0.49	**0.51**	0.49	0.50	0.49
	GBT	0.50	0.50	0.50	**0.51**	0.50	0.50
	GBT + RS	0.51	0.49	0.49	**0.51**	0.50	0.48
	GBT + SMOTE	0.48	0.49	0.49	0.50	**0.52***	0.49
	GBT + SMOTEENN	0.50	0.49	0.49	0.49	**0.51**	0.49
Mix 5	All features	**0.57**	0.56	0.56	0.53	0.51	0.50
	FS	0.58	0.57	0.55	**0.58**	0.55	0.56
	FS + RS	**0.59**	0.55	0.55	0.58	0.52	0.53
	FS + SMOTE	**0.59**	0.56	0.57	0.58	0.53	0.53
	FS + SMOTEENN	0.55	0.56	0.55	**0.57**	0.53	0.53
	GBT	**0.58**	0.57	0.56	0.58	0.55	0.58
	GBT + RS	**0.58**	0.55	0.55	0.58	0.53	0.51
	GBT + SMOTE	0.59	0.57	0.57	**0.59***	0.53	0.54
	GBT + SMOTEENN	0.54	0.54	0.55	**0.58**	0.50	0.56

Table 4.8: Balanced Accuracy Scores of our Classifiers

The best balanced accuracy score across the problems is 67%. Comparing with the balanced accuracy scores for the approximated Bayes classifier described above (reported in Table 4.9), our results look very far from ideal in comparison. That being said, to the

best of our knowledge, we have tried some of the more advanced techniques available so it indicates that this is a hard problem to learn. Since this is a relatively new approach, we hope to provide a baseline for future researchers to compare to.

Pure 0	Mix 0	Pure 5	Mix 5
0.94	0.98	0.89	0.93

Table 4.9: Balanced Accuracy Scores for "Approximated Bayes" classifier

Important observation: In the "Mix 5" scenario, doing SMOTE , either by itself or with feature selection methods, improves the performance across the classification methods. This suggests that we are suffering from the high skewness in class distribution and that we could obtain a better result by getting more data.

Learning from the instance space. It is clear from the previous discussion that our feature set does not capture all the necessary information from the instance space. Consequently, to obtain a better classifier, we try to learn directly from the instance space, i.e. by using the values from matrices A, b, and c. The technique we use to classify these instances is similar to the idea in Convolutional Neural Network (CNN), which is a class of deep neural networks. Different from the classical neural network in which values from all input nodes are used to compute the value of each neuron, in a CNN, to each neuron, we can connect only a chunk of the input nodes. For our problem, the advantage of having that property is we can derive some attribute from each row or each column of the matrix A instead of treating all values in A the same way. To be more specific, CNN has a weight-sharing mechanism (embedded in something called "filter") that allows the input to have some forms of invariance in it. In our model, we utilize a similar weight-sharing mechanism, but applied in a fully connected neural network setting. The individual rows (and columns) will be trained in separate neural networks, but sharing the weights. They are then combined with the values in matrices b and c to make final prediction. The final label is taken to be that of the output node with a larger value. The structure of the network is graphed below in Figure 4.2.

The result, however, is not as encouraging as we expect. We believe the reason for this is we do not have enough data for CNN, which requires a large dataset like other deep neural

networks structure. Having an almost perfect training accuracy and a bad test accuracy somewhat confirms our hypothesis.

Pure 0	Mix 0	Pure 5	Mix 5
0.52	0.50	0.51	0.50

Table 4.10: Balanced accuracy scores for CNN

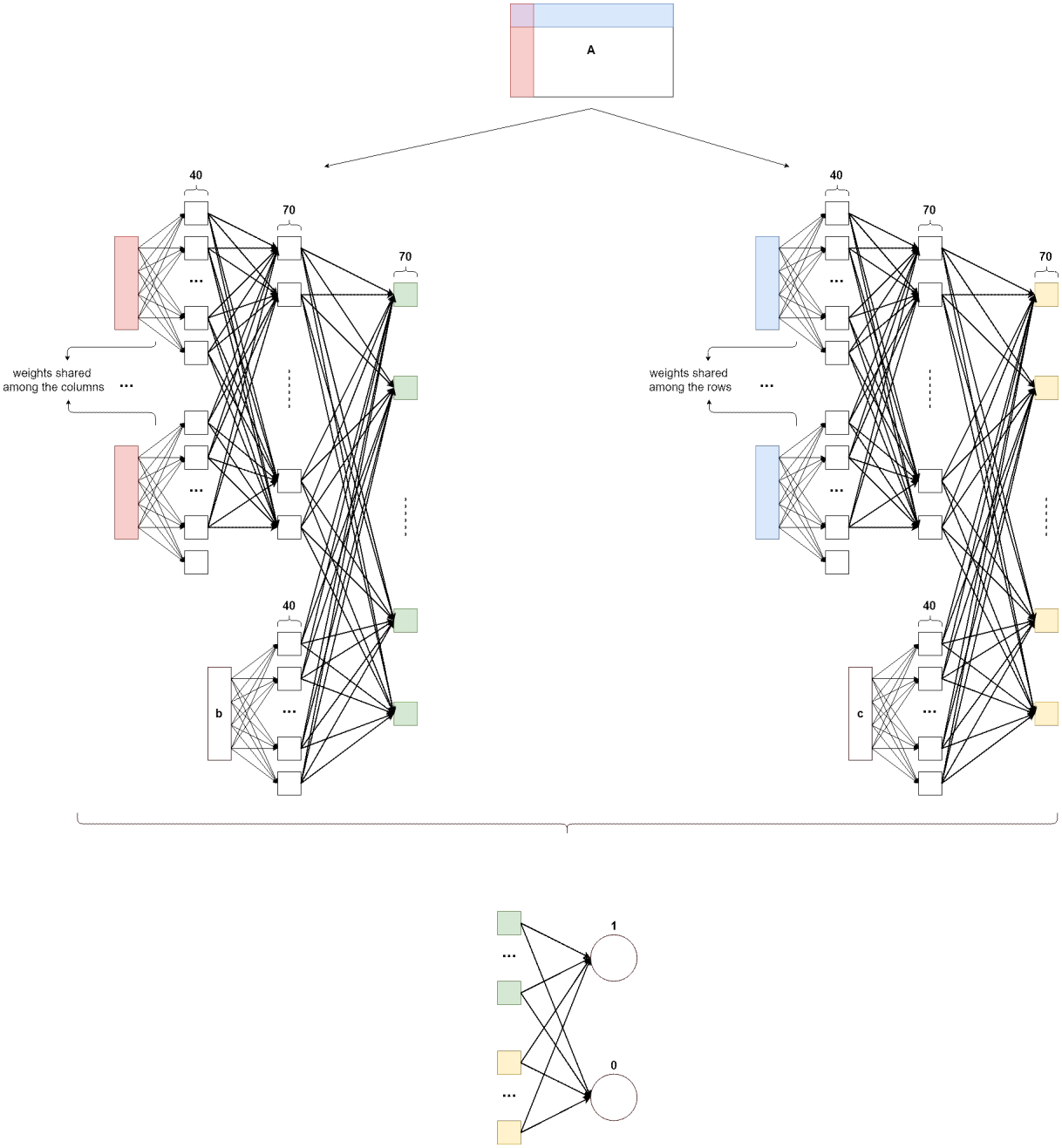

Figure 4.2: Structure of CNN-type neural network taking the raw A, b, c input

4.5 Future Work

The results from our classification set-up gives some hope that we might be able to identify problems where the cut generation procedure in [9] is helpful. However, in order to improve the accuracy of the classifier, we first need to learn the score function better. The reason is that the labels for the classification problem were obtained after doing a random run of the cut generation procedure, where μ and f were chosen randomly. Hence, if we were unlucky, even if these cuts would indeed improve upon the GMIT cut, we would get a label 0 instead. Hence, by having the best parameters (μ, f) for each problem, we will hopefully get a more consistent set of labels for the problem.

Unfortunately, our attempt at learning the score function is not fruitful. There are a couple of things that other researchers might try to get a better result:

- The set of *rows* from the final simplex tableaux that we used were randomly selected. By having a more systematic way to choose these rows, we might be able to improve the result. The authors in [36] and in [3] suggest that choosing a suitable set of equations for Gomory cuts can improve their performance, so we expect a similar phenomenon for our cuts.

- One might also wants to try other statistical learning methods to learn the score function, for example by using reinforcement learning. In fact, reinforcement learning has recently been utilized to improve certain optimization procedures [91, 60].

Chapter 5

Admissibility of Solution Estimators for Stochastic Optimization

5.1 Introduction

A large class of stochastic optimization problems can be formulated in the following way:

$$\min_{x \in X} \{f(x) := \mathbb{E}_{\xi}[F(x, \xi)]\}, \tag{5.1}$$

where $X \subseteq \mathbb{R}^d$ is a fixed feasible region, ξ is a random variable taking values in $\mathbb{R}^m$, and $F : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$. We wish to solve this problem with access to independent samples of ξ. The following are two classical examples:

1. Consider a learning problem with access to labeled samples $(z, y) \in \mathbb{R}^n \times \mathbb{R}$ from some distribution and the goal is to find a function $f \in \mathcal{F}$ in a finitely parametrized hypothesis class $\mathcal{F}$ (e.g., all neural network functions with a fixed architecture) that minimizes expected loss, where the loss function is given by $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$. One can model this using (5.1) by setting d to be the number of parameters for $\mathcal{F}$, $m = n + 1$, $X \subseteq \mathbb{R}^d$ is the subset that describes $\mathcal{F}$ via the parameters, and $F(f, (z, y)) = \ell(f(z), y)$.

2. When $d = m$, $F(x, \xi) = \|x - \xi\|^2$, $X = \mathbb{R}^d$ and ξ is distributed with mean μ, (5.1) becomes

$$\min_{x \in \mathbb{R}^d} \mathbb{E}_\xi[\|x - \xi\|^2] = \min_{x \in \mathbb{R}^d} \|x - \mathbb{E}[\xi]\|^2 + \mathbb{V}[\xi]$$

In particular, if one knows $\mu := \mathbb{E}[\xi]$, the optimal solution is given by $x = \mu$. Thus, this stochastic optimization problem becomes equivalent to the classical statistics problem of estimating the mean of the distribution of ξ, given access to independent samples.

We would like to emphasize our data-driven viewpoint on the problem (5.1). In particular, we will not assume detailed knowledge of the distribution of the random variable ξ, but only assume that it comes from a large family of distributions. More specifically, we will not assume knowledge of means or higher order moments, and certainly not the exact distribution of ξ. This is in contrast to some approaches within the stochastic optimization literature that proceed on the assumption that such detailed knowledge of the distribution is at hand. Such an approach would rewrite (5.1) by finding an analytic expression for the expectation $\mathbb{E}_\xi[F(x, \xi)]$ (in terms of the known parameters of the distribution of ξ), perhaps with some guaranteed approximation if an exact analysis is difficult. The problem then becomes a *deterministic* optimization problem, often a very complicated and difficult one, which is then attacked using novel and innovative ideas of mathematical optimization. See [17] for a textbook exposition of this viewpoint.

In contrast, as mentioned above, we will assume that the true distribution of ξ comes from a postulated large family of (structured) distributions, and we assume that we have access to data points $\xi^1, \xi^2, \ldots$ drawn independently from the true distribution of ξ. This makes our approach distinctly statistical and data-driven in nature. We "learn" or glean information about the distribution of ξ from the data, which we then use to "solve" (5.1). Statistical decision theory becomes a natural framework for such a viewpoint, to formalize what it even means to "solve" the problem after "learning" about the distribution from data. We briefly review relevant terminology from statistical decision theory below.

We do not mean to imply that a statistical perspective on stochastic optimization is new to this thesis. This is far from true; see [17, Chapter 9] and [87, Chapter 5] for detailed discussions of statistical approaches and methods in stochastic optimization. In [70] and [31],

the authors introduce a statistical decision theory perspective that is essentially the same as our framework. In recent parlance, "data-driven optimization" has been used to describe the statistical viewpoint and has a vast literature; some recent papers closely related to our work are [56, 40, 13, 41, 96], with [56, 40] particularly close in spirit to this thesis. Nevertheless, our perspective is different from previous work and follows in the footsteps of the inspirational paper of Davarnia and Cornuejols [37].

5.1.1 Statistical decision theory and admissibility

Statistical decision theory is a mathematical framework for modeling decision making in the face of uncertain or incomplete information. One models the uncertainty by a set of *states of nature* denoted by Θ. The decision making process is to choose an *action* from a set $\mathcal{A}$ that performs best in a given state of nature θ. To take our stochastic optimization setting, the set of states of nature is given by the family $\mathcal{D}$ of distributions that we believe the true distribution of ξ comes from, and the set of actions $\mathcal{A}$ is the feasible region X, i.e., select $x \in X$ that minimizes $f(x) := \mathbb{E}_{\xi\sim D}[F(x,\xi)]$. In the general framework of decision theory, one defines a *loss function* $\mathcal{L} : \Theta \times \mathcal{A} \to \mathbb{R}_+$ to evaluate the performance of an action $a \in \mathcal{A}$ against a state of nature $\theta \in \Theta$. The smaller $\mathcal{L}(\theta, a)$ is, the better $a \in \mathcal{A}$ does with respect to the state $\theta \in \Theta$[1]. In our setting of stochastic optimization, we take an action $\hat{x} \in X$. The natural way to evaluate its performance is via the so-called *optimality gap*, i.e., how close is $f(\hat{x})$ to the optimal value of (5.1). Therefore, the following is a natural loss function for stochastic optimization:

$$\begin{aligned}\mathcal{L}(D, x) &:= f(x) - f(x(D)) \\ &= \mathbb{E}_{\xi\sim D}[F(x,\xi)] - \mathbb{E}_{\xi\sim D}[F(x(D),\xi)],\end{aligned} \tag{5.2}$$

where $x(D)$ is an optimal solution to (5.1) when $\xi \sim D$.

The *statistical* aspect of statistical decision theory comes from the fact that the state θ

[1]We caution the reader that the use of the words "loss" and "risk" in statistical decision theory are somewhat different from their use in machine learning literature. In machine learning, the function $F(x,\xi)$ is usually referred to as "loss" and the function $f(x)$ is referred to as "risk" in (5.1). Thus Example 1. above becomes a "risk minimization" problem with an associated "empirical risk minimization (ERM)" problem when one replaces the expectation by a sample average.

is not revealed directly, but only through data/observations based on θ that can be noisy or incomplete. This is formalized by postulating a parameterized family of probability distributions $\mathcal{P} := \{P_\theta : \theta \in \Theta\}$ on a common *sample space* $\mathcal{X}$. After observing a realization $y \in \mathcal{X}$ of this random variable, one forms an opinion about what the possible state is and one chooses an action $a \in \mathcal{A}$. Formally, a *decision rule* is a function $\delta : \mathcal{X} \to \mathcal{A}$ giving an action $\delta(y) \in \mathcal{A}$ when data $y \in \mathcal{X}$ is observed. To take our particular setting of stochastic optimization, one observes data points $\xi^1, \xi^2, \ldots, \xi^n$ that are i.i.d. realizations of $\xi \sim D$; thus, $\mathcal{X} = \underbrace{\mathbb{R}^m \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m}_{n \text{ times}}$ with distributions $\mathcal{P} := \{\underbrace{D \times D \times \ldots \times D}_{n \text{ times}} : D \in \mathcal{D}\}$ on $\mathcal{X}$ parameterized by the states $D \in \mathcal{D}$.

Finally, one evaluates decision rules by averaging over the data, defining the *risk function*

$$\mathcal{R}(\theta, \delta) := \mathbb{E}_{y \sim P_\theta}[\mathcal{L}(\theta, \delta(y))].$$

One can think of the risk function as mapping a decision rule to a nonnegative function on the class of distributions $\mathcal{P}$, or alternatively, a nonnegative function on the parameter space Θ; this function is sometimes called *the risk of the decision rule*. A decision rule is "good" if its risk has "low" values. The "best" possible decision rule would be a δ^* such that for any other decision rule δ', $\mathcal{R}(\theta, \delta^*) \leq \mathcal{R}(\theta, \delta')$ for all $\theta \in \Theta$, i.e., δ^* has risk that pointwise dominates the risk of any other decision rule. Usually such universally dominating decision rules do not exist.

A basic criterion for choosing decision rules is then the following. We say that δ' *weakly dominates* δ if $\mathcal{R}(\theta, \delta') \leq \mathcal{R}(\theta, \delta)$ for all $\theta \in \Theta$. We say that δ' *dominates* δ if, in addition, $\mathcal{R}(\hat{\theta}, \delta') < \mathcal{R}(\hat{\theta}, \delta)$ for some $\hat{\theta} \in \Theta$. A decision rule δ is said to be *inadmissible* if there exists another decision rule δ' that dominates δ. A decision rule δ is said to be *admissible* if it is not dominated by any other decision rule. In-depth discussions of general statistical decision theory can be found in [12, 15, 68].

5.1.2 Admissibility of the sample average estimator and our results

We would like to study the admissibility of natural decision rules for solving (5.1). As explained above, we put this in the decision theoretical framework by setting the sample

space $\mathcal{X} = \underbrace{\mathbb{R}^m \times \ldots \times \mathbb{R}^m}_{n \text{ times}}$, where n is the number of i.i.d. observations one makes for $\xi \sim D$ for $D \in \mathcal{D}$, and $\mathcal{D}$ is a fixed family of distributions. A decision rule is now a map $\delta : \underbrace{\mathbb{R}^m \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m}_{n \text{ times}} \to X$. The class of distributions on $\mathcal{X}$ is $\mathcal{P} = \{\underbrace{D \times D \times \ldots \times D}_{n \text{ times}} : D \in \mathcal{D}\}$. The loss function is defined as in (5.2).

In this thesis, we wish to study the admissibility of the *sample average* decision rule δ_{SA} defined as

$$\delta_{SA}(\xi^1, \ldots, \xi^n) \in \arg\min \left\{ \frac{1}{n} \sum_{i=1}^{n} F(x, \xi^i) : x \in X \right\} \tag{5.3}$$

In other words, δ_{SA} reports an optimal solution with respect to the sample average of the objective. This is a standard procedure in stochastic optimization, and often goes by the name of *sample average approximation (SAA)*; in machine learning, it goes by the name of *Empirical Risk Minimization (ERM)*. To emphasize the dependence on the number of samples n, we introduce a superscript, i.e., δ^n_{SA} will denote the estimator based on the sample average of the objective from n observations. Moreover, for any $n \in \mathbb{N}$, let Δ^n be the set of all decision rules $\delta : \underbrace{\mathbb{R}^m \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m}_{n \text{ times}} \to X$ such that $\mathbb{E}_{\xi^1, \ldots, \xi^n}[\delta(\xi^1, \ldots, \xi^n)]$ exists.

Stein's paradox In turns out that there are simple instances of problem (5.1) where the sample average estimator is *inadmissible*. Consider the setting of Example 2 in the Introduction, where $m = d$, $F(x, \xi) = \|x - \xi\|^2$ and $X = \mathbb{R}^d$. We assume $\xi \sim N(\mu, I)$ with unknown μ; here I denotes the identity matrix. Thus, we assume that the true distribution of ξ is a normal with identity as covariance matrix; the mean μ is what is unknown. In the language of statistical decision theory that we have adopted, the states of nature are now parametrized by $\mu \in \mathbb{R}^d$. Let us calculate the exact form of the loss using (5.2). First recall from the calculation in Example 2 that $\mathbb{E}_{\xi \sim D}[F(x, \xi)] = \|x - \mathbb{E}[\xi]\|^2 + \mathbb{V}[\xi]$ and thus the optimal solution to minimize $\mathbb{E}_{\xi \sim D}[F(x, \xi)]$ is simply $x(D) = \mathbb{E}[\xi] = \mu$ with objective value

$\mathbb{V}[\xi]$. We then obtain

$$\begin{aligned}\mathcal{L}(\mu, x) &= \mathbb{E}_{\xi\sim D}[F(x,\xi)] - \mathbb{E}_{\xi\sim D}[F(x(D),\xi)] \\ &= \left(\|x - \mathbb{E}[\xi]\|^2 + \mathbb{V}[\xi]\right) - \mathbb{V}[\xi] \\ &= \|x - \mu\|^2.\end{aligned}$$

Thus, $x = \mu$ minimizes the loss when the state of nature is μ. Consequently, the problem becomes the classical problem of estimating the mean of a Gaussian from samples under "squared distance loss". Also, the sample average decision rule solves (5.3) which is the problem

$$\min\left\{\frac{1}{n}\sum_{i=1}^{n}\left\|x - \xi^i\right\|^2 : x \in \mathbb{R}^d\right\}$$

and therefore simply returns the empirical average of the samples, i.e., $\delta^n_{SA}(\xi^1, \ldots, \xi^n) = \overline{\xi}$ where $\overline{\xi} := \frac{1}{n}\sum_{i=1}^{n}\xi^i$ denotes the sample average. It is well-known that this sample average decision rule is *inadmissible* if $d \geq 3$; this was first observed by Stein [90] and is commonly referred to as *Stein's paradox* in statistics literature. The *James-Stein estimator* [62] can be shown to strictly dominate the sample average estimator; see [12, 68] for an exposition.

Our results We focus on two particular cases of the stochastic optimization problem (5.1):

1. $m = d$, $F(x, \xi) = \xi^T x$, $X \subseteq \mathbb{R}^d$ is a given compact (not necessarily convex) set, and ξ has a *Gaussian distribution* with *unknown* mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$ denoted by $\xi \sim N(\mu, \Sigma)$. In other words, we optimize an uncertain linear objective over a fixed compact set. Note that, along with linear or convex optimization, we also capture non-convex feasible regions like mixed-integer non-linear optimization or linear complementarity constraints.

2. $m = d$, $F(x, \xi) = \frac{1}{2}\|x\|^2 - \xi^T x$, $X \subseteq \mathbb{R}^d$ is a box constrained set, i.e., $X := \{x \in \mathbb{R}^d : \ell_i \leq x_i \leq u_i, \quad i = 1, \ldots, d\}$ ($\ell_i \leq u_i$ are arbitrary real numbers), and ξ has a *Gaussian distribution* with *unknown* mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$ denoted by $\xi \sim N(\mu, \Sigma)$. Here, we wish to minimize a convex quadratic function with an uncertain linear term over box constraints.

In the first case, we show that there is no "Stein's paradox" type phenomenon, i.e., the sample average solution is admissible for every $d \in \mathbb{N}$. For the second case, we show that the sample average solution is admissible for $d \leq 4$. Note that in the second situation above, $F(x,\xi) = \frac{1}{2}\|x-\xi\|^2 - \frac{1}{2}\|\xi\|^2$ and thus the problem of minimizing $\mathbb{E}[\frac{1}{2}\|x-\xi\|^2 + \frac{1}{2}\|\xi\|^2] = \mathbb{E}[\frac{1}{2}\|x-\xi\|^2] + \frac{1}{2}\mathbb{E}[\|\xi\|^2]$ is equivalent to the setting of Stein's paradox (since $\frac{1}{2}\mathbb{E}[\|\xi\|^2]$ is just a constant), except that we now impose box constraints on x. Thus, admissibility is recovered for $d = 3, 4$ with box constraints. While we are unable to establish it for $d \geq 5$, we strongly suspect that there is no Stein's paradox in *any dimension* once box constraints are imposed. The precise statements of our results follow.

Theorem 5.1. *Consider problem (5.1) in the setting where X is a given compact set and $F(\xi,x) = \xi^T x$, and $\xi \sim N(\mu,\Sigma)$ with unknown μ and Σ. The sample average rule now simply becomes*

$$\delta^n_{SA}(\xi^1,\ldots,\xi^n) \in \arg\min\{\overline{\xi}^T x : x \in X\} \tag{5.4}$$

where $\overline{\xi} := \frac{1}{n}\sum_{i=1}^n \xi^i$ denotes the sample average of the observed objective vectors. For any $n \in \mathbb{N}$, and any $\Sigma \in \mathbb{R}^{d\times d}$, we consider the states of nature to be parametrized by $\mu \in \mathbb{R}^d$. Then for every $n \in \mathbb{N}$ and $\Sigma \in \mathbb{R}^{d\times d}$, δ^n_{SA} is admissible within Δ^n.

Theorem 5.2. *Let $d \leq 4$. Consider problem (5.1) in the setting where $X := \{x \in \mathbb{R}^d : \ell_i \leq x_i \leq u_i,\ \ i = 1,\ldots,d\}$ ($\ell_i \leq u_i$ are arbitrary real numbers) and $F(\xi,x) = \frac{1}{2}\|x\|^2 - \xi^T x$, and $\xi \sim N(\mu,\Sigma)$ with unknown μ and Σ. The sample average rule now simply becomes*

$$\delta^n_{SA}(\xi^1,\ldots,\xi^n) \in \arg\min\left\{\frac{1}{2}\|x\|^2 - \overline{\xi}^T x : x \in X\right\} \tag{5.5}$$

where $\overline{\xi} := \frac{1}{n}\sum_{i=1}^n \xi^i$ denotes the sample average of the observed vectors. For any $n \in \mathbb{N}$, and any $\Sigma \in \mathbb{R}^{d\times d}$, we consider the states of nature to be parametrized by $\mu \in \mathbb{R}^d$. Then for every $n \in \mathbb{N}$ and $\Sigma \in \mathbb{R}^{d\times d}$, δ^n_{SA} is admissible within Δ^n.

We present two different proofs of Theorem 5.1. The first one, presented in Sections 5.3 and 5.3.2 uses a novel proof technique for admissibility, to the best of our knowledge. The second proof, presented in Section 5.4 uses the conventional idea of showing that the sample average estimator is the (unique) Bayes estimator under an appropriate prior. We

feel that the first proof technique could be useful for future research into the question of admissibility of solution estimators for stochastic optimization. The second method using Bayes estimators is easier to generalize to the quadratic setting of Theorem 5.2 and thus forms a natural segue into its proof presented in Section 5.5. .

5.1.3 Comparison with previous work

The statistical decision theory perspective on stochastic optimization presented here follows the framework of [37] and [38]. In particular, the authors of [37] consider admissibility of solution estimators in two different stochastic optimization problems: one where $X = \mathbb{R}^d$ and $F(x, \xi) = x^T Q x + \xi^T x$ for some fixed matrix positive definite matrix Q (i.e., unconstrained convex quadratic minimization), and the second one where X is the unit ball and $F(x, \xi) = \xi^T x$. ξ is again assumed to be distributed according to a normal distribution $N(\mu, I)$ with unknown mean μ. They show that the sample average approximation is *not* admissible in general for the first problem, and it is admissible for the second problem. Note that the second problem is a special case of our setting. In both these cases, there is a closed-form solution to the deterministic version of the optimization problem, which helps in the analysis. This is not true for the general optimization problem we consider here (even if we restrict X to be a polytope, we get a linear program which, in general, has no closed form solution).

Another difference between our work and [37] is the following. In [37], the question of admissibility is addressed within a smaller subset of decision rules that are "decomposable" in the sense that any decision rule is of the form $\tau \circ \kappa$, where $\kappa : \underbrace{\mathbb{R}^d \times \mathbb{R}^d \times \ldots \times \mathbb{R}^d}_{n \text{ times}} \to \mathbb{R}^d$ maps the data $\xi^1, \ldots, \xi^n$ to a vector $\hat{\mu} \in \mathbb{R}^d$ and then $\tau : \mathbb{R}^d \to X$ is of the form $\tau(\hat{\mu}) \in \arg\min\{\hat{\mu}^T x : x \in X\}$. In other words, one first estimates the mean of the uncertain objective (using any appropriate decision rule) and then uses this estimate to solve a deterministic optimization problem. In the follow-up work [38], the authors call such decision rules *Separate estimation-optimization (Separate EO) schemes* and more general decision rules as *Joint estimation-optimization (Joint EO) schemes.*

In this thesis, we establish admissibility of the sample average estimator within general

decision rules (joint EO schemes in the terminology of [38]). The only condition we put on the decision rules is that of integrability, which is a minimum requirement needed to even define the risk of a decision rule. Note that proving inadmissibility within separate EO schemes implies inadmissibility within joint EO schemes. On the other hand, establishing admissibility within joint EO schemes means defending against a larger class of decision rules. The general concept of joint estimation-optimization schemes also appears in [31, 70, 40], presented in slightly different vocabulary.

As mentioned before, the quadratic convex objective has been studied in statistics in the large body of work surrounding Stein's paradox, albeit not in the stochastic optimization language that we focus on here. Moreover, all of this classical work is for the unconstrained problem. To the best of our knowledge, the version with box constraints has not been studied before (but see [29, 28, 72, 63, 67, 44, 58, 14, 49, 55] and the book [45] for a related, but different, statistical problem that has received a lot of attention). It is very intriguing (at least to us) that in the presence of such constraints, admissibility is recovered for dimensions $d = 3, 4$; recall that for the unconstrained problem, the sample average solution is admissible only for $d = 1, 2$ and inadmissible for $d \geq 3$. Unfortunately, we are unable to resolve the admissibility question of the sample average solution for dimensions $d \geq 5$, but we strongly suspect that there is no Stein's paradox in *any dimension* once box constraints are imposed.

5.1.4 Admissibility and other notions of optimality

We end our discussion of the results with a few comments about other optimality notions for decision rules. In large sample statistics, one often considers the behavior of decision rules when $n \to \infty$ (recall n is the number of samples). A sequence of decision rules δ^n (each δ^n is based on n i.i.d samples) is said to be *asymptotically inadmissible* if there exists a decision rule sequence $\bar{\delta}^n$ such that $\lim_{n\to\infty} \frac{R(\theta,\bar{\delta}^n)}{R(\theta,\delta_n)} \leq 1$ for every θ, and for some $\hat{\theta}$ the limiting ratio is strictly less than 1. Admissibility for every $n \in \mathbb{N}$ does not necessarily imply asymptotic admissibility [15, Problem 4, page 200], and asymptotic admissibility does not imply admissibility for finite $n \in \mathbb{N}$, i.e., there can be decision rules that are inadmissible for every $n \in \mathbb{N}$ and yet be asymptotically admissible. Thus, the small-sample behaviour

(fixed $n \in \mathbb{N}$) and large sample behavior ($n \to \infty$) can be quite different. One advantage of proving asymptotic admissibility is that it also implies the *rate* of convergence of the risk (as a function of n) is optimal; such rules are called *rate optimal*. Unfortunately, our admissibility results about δ^n_{SA} do not immediately imply asymptotic admissibility or rate optimality. Standard techniques for proving rate optimality such as Hajek-Le Cam theory [95, Chapter 8] cannot be applied because regularity assumptions about the decision rules are not satisfied in our setting. For example, in the linear objective case discussed above, δ^n_{SA} has a degenerate distribution that is supported on the boundary of the feasible region X which rules out "asymptotic normality" or "local asymptotic minimaxity" results [95, Chapters 7, 8].

While we are unable to prove rate optimality for δ^n_{SA}, it is reasonably straightforward to show that δ^n_{SA} is *consistent* in the sense that $R(\theta, \delta^n_{SA}) \to 0$ as $n \to \infty$. This can be derived from consistency results in stochastic optimization literature [87, Chapter 5], but we present the argument in Appendix A.1 for completeness. In the linear objective case, the loss for δ^n_{SA} is given by $\mathcal{L}(\mu, \delta^n_{SA}) = \mu^T \delta^n_{SA} - \mu^T x(\mu) = (\mu - \bar{\xi})^T \delta^n_{SA} + \bar{\xi}^T \delta^n_{SA} - \mu^T x(\mu) \leq (\mu - \bar{\xi})^T \delta^n_{SA} + \bar{\xi}^T x(\mu) - \mu^T x(\mu)$ since δ^n_{SA} is the minimizer with respect to $\bar{\xi}$. Thus, $\mathcal{L}(\mu, \delta^n_{SA}) \leq (\mu - \bar{\xi})^T (\delta^n_{SA} - x(\mu)) \leq \|\mu - \bar{\xi})\| \cdot K$ by the Cauchy-Schwarz inequality, where K is the diameter of the compact feasible region X. Therefore, $R(\mu, \delta^n_{SA}) \leq K\mathbb{E}[\|\mu - \bar{\xi})\|]$. Since the sample average $\bar{\xi}$ has a normal distribution with mean μ and variance that scales like $O(1/n)$, $R(\mu, \delta^n_{SA}) \to 0$ with rate $O(1/\sqrt{n})$. A similar argument can be made in the quadratic objective case (see Appendix A.1). However, we are unable to show that $O(1/\sqrt{n})$ is the optimal rate in either case.

There is a large body of literature on *shrinkage estimators* in the unconstrained, quadratic objective setting. A relatively recent insight [97] shows that as $d \to \infty$ (recall d is the dimension), a certain class of shrinkage estimators (called *SURE estimators*) have risk functions that dominate any other shrinkage estimator's risk, and hence the sample average estimator's risk, with just a single sample. This potentially suggests that the phenomenon presented here, where admissibility of δ^n_{SA} is recovered for $d = 3, 4$, holds only for small dimensions and for large enough dimensions, the sample average estimator remains inadmissible. However, this is not immediate because of two reasons: 1) the value of d for

which the SURE estimator in [97] starts to dominate any other estimator depends on the parameter μ, and 2) the setting in [97] is still unconstrained optimization. In fact, as stated earlier, we strongly suspect that with compact constraints, admissibility of δ_{SA}^n holds for *all* dimensions. As an example, if box constraints are replaced with ball constraints, i.e., X is a ball around the origin of radius R, then the quadratic problem reduces to the linear case, and using Theorem 5.1, δ_{SA}^n can then be shown to be admissible for all d.

There are other notions of optimality of decision rules even in the small/finite sample setting. For example, the *minimax* decision rule minimizes the sup norm of the risk function, i.e., one solves $\inf_\delta \sup_\theta R(\theta, \delta)$. In general, admissibility does not imply minimaxity, nor does minimaxity imply admissibility. Of course, if a minimax rule is inadmissible, then the dominating rule is also minimax and is certainly to be preferred, unless computational concerns prohibit this. In many settings however (e.g., estimation in certain exponential and group families [68, Chapter 5]), minimax rules are also provably admissible and thus minimaxity is a more desirable criterion.

Generally speaking, admissibility is considered a weak notion of optimality because admissible rules may have undesirable properties like very high risk values for certain states of the world. Moreover, as noted above, inadmissible rules may have optimal large sample behavior. Nevertheless, it is useful to know if widely used decision rules such as sample average approximations satisfy the basic admissibility criterion, because if not, then one could use the dominating decision rule unless it is computationally much more expensive.

5.2 Technical Tools

We first recall a basic fact from calculus.

Lemma 5.1. *Let $F : \mathbb{R}^m \to \mathbb{R}$ be a twice continuously differentiable map such that $F(0) = 0$. Suppose $\nabla^2 F(0)$ is not negative semidefinite; in other words, there is a direction $d \in \mathbb{R}^d$ of positive curvature, i.e., $d^T \nabla^2 F(0) d > 0$. Then there exists $z \in \mathbb{R}^m$ such that $F(z) > 0$.*

We will briefly describe the argument of the proof. A complete proof is given in Appendix A.2.

Proof. If $\nabla F(0) \neq 0$, then there exist $\lambda > 0$ such that $F(z) > 0$ for $z = \lambda \nabla F(0)$ since $F(0) = 0$. Else, if $\nabla F(0) = 0$ then there exists $\lambda > 0$ such that $F(z) > 0$ for $z = \lambda d$, where d is the direction of positive curvature at 0. □

We will need the following result from statistics. See Example 3.5 for a more general result.

Proposition 5.1. *Let* $\mathcal{X} = \underbrace{\mathbb{R}^d \times \ldots \times \mathbb{R}^d}_{n \text{ times}}$ *and let* $\mathcal{P} = \{\underbrace{N(\mu, I) \times \ldots \times N(\mu, I)}_{n \text{ times}} : \mu \in \mathbb{R}^d\}$, *i.e.,* $(\xi^1, \ldots, \xi^n) \in \mathcal{X}$ *are i.i.d samples from the normal distribution* $N(\mu, I)$. *Then* $T(\xi^1 \ldots, \xi^n)$
$= \bar{\xi} := \frac{1}{n} \sum_{i=1}^n \xi^i$ *is a sufficient statistic for* $\mathcal{P}$.

We will also need the following useful property for the family of normal distributions $\{N(\mu, I) : \mu \in \mathbb{R}^m\}$. Indeed the following result is true for any *exponential family* of distributions; see Theorem 5.8, Chapter 1 in [68] for details.

Theorem 5.3. *Let* $f : \mathbb{R}^m \to \mathbb{R}^d$ *be any integrable function. The function*

$$h(\mu) := \int_{\mathbb{R}^m} f(y) e^{\frac{-n\|y-\mu\|^2}{2}} dy$$

is continuous and has derivatives of all orders with respect to μ, *which can be obtained by differentiating under the integral sign.*

In the rest of the chapter, for any vector v, v_j will denote the j-th coordinate, and for any matrix $A \in \mathbb{R}^{p \times q}$, A_{ij} will denote entry in the i-th row and j-th column.

We need one further result on the geometry of the hypercube which is easy to verify. We recall that for any closed, convex set $C \subseteq \mathbb{R}^d$ and any face F of C, the *normal cone at the face* F is defined to be the set of all vectors $c \in \mathbb{R}^d$ such that $F \subseteq \text{argmax}_{y \in C} c^T y$.

Lemma 5.2. *Let* $X = \{x \in \mathbb{R}^d : -u_i \leq x_i \leq u_i \;\; i = 1, \ldots, d\}$ *be a box centered at the origin. For any face* $F \subseteq X$ *of* X *(possibly with* $F = X$*), let* $I_F^+ \subseteq \{1, \ldots, d\}$ *be the subset of coordinates which are set to the bound* u_i *for all points in* F, $I_F^- \subseteq \{1, \ldots, d\}$ *be the subset of coordinates which are set to the bound* $-u_i$ *for all points in* F, *and* N_F *denote the normal cone at* F, *which is* $N_F = \{\mathbf{r} : \langle \mathbf{r}, \mathbf{y} - \mathbf{x} \rangle \geq 0, \; \forall \mathbf{y} \in F \text{ and } \forall \mathbf{x} \in X\}$ *(notice the*

similarity between this definition and the definition of normal cone at a point as introduced in Section 2.2). Then the following are true:

1. *For any face $F \subseteq X$,*

$$F + N_F = \left\{ x \in \mathbb{R}^d : \begin{array}{ll} x_i \geq u_i & i \in I_F^+ \\ x_i \leq -u_i & i \in I_F^- \\ -u_i \leq x_i \leq u_i & i \notin I_F^+ \cup I_F^- \end{array} \right\}.$$

2. *The interior of $F + N_F$ is disjoint from the interior of $F' + N_F'$ whenever $F \neq F'$ and we have the following decomposition of $\mathbb{R}^d$:*

$$\mathbb{R}^d = \bigcup_{F \text{ face of } X} F + N_F$$

The proof of this lemma is provided in Appendix A.3.

5.3 Proof of Theorem 5.1 (the scenario with linear objective)

5.3.1 When the covariance matrix is the identity

Proof of Theorem 5.1 when $\Sigma = I$. As introduced in the previous sections, $\overline{\xi}$ will denote the sample average of $\xi^1, \ldots, \xi^n$. Consider an arbitrary decision rule $\delta \in \Delta^n$. Consider the conditional expectation

$$\eta(y) = \mathbb{E}_{\xi^1,\ldots,\xi^n}[\delta(\xi^1, \ldots, \xi^n) | \overline{\xi} = y].$$

Observe that $\eta(y) \in \text{conv}(X)$ (i.e., the convex hull of X, which is compact since X is compact) since δ maps into X. Moreover, since $\overline{\xi}$ is a sufficient statistic for the family of normal distributions by Proposition 5.1, $\eta(y)$ does not depend on μ. This is going to be important below. To maintain intuitive notation, we will also say that δ_{SA}^n is given by $\delta_{SA}^n(\xi^1, \ldots, \xi^n) = \eta^*(\overline{\xi})$, where $\eta^*(y)$ returns a point in $\arg\min\{ y^T x \ : \ x \in X\}$. Note also

that for any action $x \in X$, (5.2) evaluates to

$$\mathcal{L}(\mu, x) = \mu^T x - \mu^T x(\mu),$$

where $x(\mu)$ denotes the optimal solution to the problem $\min\left\{\mu^T x : x \in X\right\}$. Using the law of total expectation,

$$\begin{aligned} R(\mu, \delta) &= \mathbb{E}_{\xi^1,\ldots,\xi^n}[\mathcal{L}(\mu, \delta(\xi^1, \ldots, \xi^n))] \\ &= \mathbb{E}_{\xi^1,\ldots,\xi^n}[\mu^T \delta(\xi^1, \ldots, \xi^n)] - \mu^T x(\mu) \\ &= \mathbb{E}_{\bar{\xi}}[\mathbb{E}_{\xi^1,\ldots,\xi^n}[\mu^T \delta(\xi^1, \ldots, \xi^n)|\bar{\xi}]] - \mu^T x(\mu) \\ &= \mathbb{E}_{\bar{\xi}}[\mu^T \eta(\bar{\xi})] - \mu^T x(\mu) \end{aligned}$$

If $\eta = \eta^*$ almost everywhere, then $R(\mu, \delta) = R(\mu, \delta^n_{SA})$ for all $\mu \in \mathbb{R}^d$, and we would be done. So in the following, we assume that $\eta \neq \eta^*$ on a set of strictly positive measure. This implies the following

Claim 5.1. *For all $y \in \mathbb{R}^d$, $y^T\eta(y) \geq y^T\eta^*(y)$ and the set $\{y \in \mathbb{R}^d : y^T\eta(y) > y^T\eta^*(y)\}$ is of strictly positive measure.*

Proof of Claim. Since X is compact, $\text{conv}(X)$ is a compact, convex set and $\min\{y^T x : x \in \text{conv}(X)\} = \min\{y^T x : x \in X\}$ for every $y \in \mathbb{R}^d$. Therefore, since $\eta(y) \in \text{conv}(X)$ and $\eta^*(y) \in \arg\min\{ y^T x : x \in X\}$, we have $y^T\eta(y) \geq y^T\eta^*(y)$ for all $y \in \mathbb{R}^d$.

Since $\text{conv}(X)$ is a compact, convex set, the set of $y \in \mathbb{R}^d$ such that $|\arg\min\{y^T x : x \in \text{conv}(X)\}| > 1$ is of zero Lebesgue measure. Let $S \subseteq \mathbb{R}^d$ be the set of $y \in \mathbb{R}^d$ such that $\arg\min\{y^T x : x \in \text{conv}(X)\}$ is a singleton, i.e., there is a unique optimal solution; so $\mathbb{R}^d \setminus S$ has zero Lebesgue measure. Let $D := \{y \in \mathbb{R}^d : \eta(y) \neq \eta^*(y)\}$. Since we assume that D has strictly positive measure, $D \cap S$ must have strictly positive measure. Consider any $y \in D \cap S$. Since $\min\{y^T x : x \in X\} = \min\{y^T x : x \in \text{conv}(X)\}$, we must have $\arg\min\{y^T x : x \in X\} \subseteq \arg\min\{y^T x : x \in \text{conv}(X)\}$. Since $y \in S$, $\arg\min\{y^T x : x \in \text{conv}(X)\}$ is a singleton and thus $\eta^*(y)$ is the unique optimum for $\min\{y^T x : x \in \text{conv}(X)\}$. Since $y \in D$, $\eta(y) \neq \eta(y^*)$, and therefore $y^T\eta(y) > y^T\eta^*(y)$. Thus, we have the second part of the claim. ◇

Now consider the function $F : \mathbb{R}^d \to \mathbb{R}$ defined by

$$F(\mu) := R(\mu, \delta) - R(\mu, \delta^n_{SA}). \tag{5.6}$$

To show that δ^n_{SA} is admissible, it suffices to show that there exists $\bar{\mu} \in \mathbb{R}^d$ such that $F(\bar{\mu}) > 0$. For any $\mu \in \mathbb{R}^d$, we have from above

$$\begin{aligned} F(\mu) &= R(\mu, \delta) - R(\mu, \delta^n_{SA}) \\ &= \mathbb{E}_y[\mu^T \eta(y)] - \mathbb{E}_y[\mu^T \eta^*(y)] \\ &= \mu^T \int_{\mathbb{R}^d} (\tfrac{n}{2\pi})^{n/2} \eta(y) e^{\frac{-n\|y-\mu\|^2}{2}} dy - \mu^T \int_{\mathbb{R}^d} (\tfrac{n}{2\pi})^{n/2} \eta^*(y) e^{\frac{-n\|y-\mu\|^2}{2}} dy \\ &= \mu^T \int_{\mathbb{R}^d} (\tfrac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y)) e^{\frac{-n\|y-\mu\|^2}{2}} dy \end{aligned}$$

where in the second to last equality, we have used the fact that $\overline{\xi}$ has distribution $N(\mu, \frac{1}{n} I)$. Note that the formula above immediately gives $F(0) = 0$. We will employ Lemma 5.1 on $F(\mu)$ to show the existence of $\bar{\mu} \in \mathbb{R}^d$ such that $F(\bar{\mu}) > 0$. For this purpose, we need to compute the gradient $\nabla F(\mu)$ and Hessian $\nabla^2 F(\mu)$. We alert the reader that in these calculations, it is crucial that $\eta(y)$ does not depend on μ (due to sufficiency of the sample average) and hence it is to be considered as a constant when computing the derivatives below. For ease of calculation, we introduce the following functions $E, G^1, \ldots, G^d : \mathbb{R}^d \to \mathbb{R}^d$:

$$\begin{aligned} E(\mu) &:= (\tfrac{n}{2\pi})^{n/2} \int_{\mathbb{R}^d} (\eta(y) - \eta^*(y)) e^{\frac{-n\|y-\mu\|^2}{2}} dy, \\ G^i(\mu) &:= (\tfrac{n}{2\pi})^{n/2} \int_{\mathbb{R}^d} y_i (\eta(y) - \eta^*(y)) e^{\frac{-n\|y-\mu\|^2}{2}} dy, \quad i = 1, \ldots, d. \end{aligned}$$

So $F(\mu) = \mu^T E(\mu)$. We also define the map $G : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ as

$$G(\mu)_{ij} = (G^i(\mu))_j.$$

Claim 5.2. *For any $\mu \in \mathbb{R}^d$, $\nabla F(\mu) = E(\mu) + nG(\mu)\mu - n(\mu^T E(\mu))\mu$. (Note that $G(\mu)\mu$ is a matrix-vector product.)*

Proof of Claim. This is a straightforward calculation. Consider the i-th coordinate of $\nabla F(\mu)$, i.e., the i-th partial derivative

$$\begin{aligned}
\frac{\partial F}{\partial \mu_i} &= \frac{\partial(\sum_j \mu_j E(\mu)_j)}{\partial \mu_i} \\
&= E(\mu)_i + \textstyle\sum_{j=1}^d \mu_j \frac{\partial E(\mu)_j}{\partial \mu_i} \\
&= E(\mu)_i + \textstyle\sum_{j=1}^d \mu_j \big(\int_{\mathbb{R}^d} (\frac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y))_j \frac{\partial(e^{\frac{-n\|y-\mu\|^2}{2}})}{\partial \mu_i} dy \big) \\
&= E(\mu)_i + \mu^T \textstyle\int_{\mathbb{R}^d} (\frac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y)) \frac{\partial(e^{\frac{-n\|y-\mu\|^2}{2}})}{\partial \mu_i} dy \\
&= E(\mu)_i + \mu^T \textstyle\int_{\mathbb{R}^d} (\frac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y)) e^{\frac{-n\|y-\mu\|^2}{2}} (n(y_i - \mu_i)) dy \\
&= E(\mu)_i + n\mu^T G^i(\mu) - n(\mu^T E(\mu))\mu_i
\end{aligned}$$

where in the third equality, we have used Theorem 5.3 and the fact that $\eta(y), \eta^*(y)$ do not depend on μ by sufficiency of the sample average (Proposition 5.1). The last expression above corresponds to the i-th coordinate of $E(\mu) + nG(\mu)\mu - n(\mu^T E(\mu))\mu$. Thus, we are done. ◇

Claim 5.3. $\nabla^2 F(0) = n(G(0)^T + G(0))$.

Proof of Claim. Let us compute $\frac{\partial^2 F}{\partial \mu_i \mu_j}$ using the expression for $\frac{\partial F}{\partial \mu_i}$ from Claim 5.2.

$$\begin{aligned}
\frac{\partial^2 F}{\partial \mu_i \mu_j} &= \frac{\partial(E(\mu)_i)}{\partial \mu_j} + n\frac{\partial(\mu^T G^i(\mu))}{\partial \mu_j} - n\frac{\partial((\mu^T E(\mu))\mu_i)}{\partial \mu_j} \\
&= \frac{\partial(E(\mu)_i)}{\partial \mu_j} + n(G^i(\mu))_j + n\mu^T \frac{\partial(G^i)}{\partial \mu_j} - n\frac{\partial(\mu^T E(\mu))}{\partial \mu_j}\mu_i - n(\mu^T E(\mu))\gamma_{ij}
\end{aligned}$$

where γ_{ij} denotes the Kronecker delta function, i.e., $\gamma_{ij} = 1$ if $i = j$ and 0 otherwise. At $\mu = 0$, the above simplifies to

$$\left.\frac{\partial^2 F}{\partial \mu_i \mu_j}\right|_{\mu=0} = \left.\frac{\partial(E(\mu)_i)}{\partial \mu_j}\right|_{\mu=0} + n(G^i(0))_j. \tag{5.7}$$

Let us now investigate $\frac{\partial(E(\mu)_i)}{\partial \mu_j}$. By applying Theorem 5.3 and the sufficiency of $\overline{\xi}$ again, we

obtain

$$\begin{aligned}
\frac{\partial(E(\mu)_i)}{\partial \mu_j} &= \int_{\mathbb{R}^d} (\tfrac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y))_i \frac{\partial(e^{\frac{-n\|y-\mu\|^2}{2}})}{\partial \mu_i} dy \\
&= \int_{\mathbb{R}^d} (\tfrac{n}{2\pi})^{n/2} (\eta(y) - \eta^*(y))_i e^{\frac{-n\|y-\mu\|^2}{2}} (n(y_j - \mu_j)) dy \\
&= n(\tfrac{n}{2\pi})^{n/2} \int_{\mathbb{R}^d} y_j (\eta(y) - \eta^*(y))_i e^{\frac{-n\|y-\mu\|^2}{2}} dy \\
&\quad -n(\tfrac{n}{2\pi})^{n/2} \mu_j \int_{\mathbb{R}^d} (\eta(y) - \eta^*(y))_i e^{\frac{-n\|y-\mu\|^2}{2}} dy \\
&= n(G^j(\mu))_i - n\mu_j E(\mu)_i
\end{aligned}$$

Therefore, at $\mu = 0$, we obtain that $\left.\frac{\partial(E(\mu)_i)}{\partial \mu_j}\right|_{\mu=0} = nG^j(0)_i$. Putting this back into (5.7), and using the definition of the matrix $G(\mu)$, we obtain

$$\left.\frac{\partial^2 F}{\partial \mu_i \mu_j}\right|_{\mu=0} = nG^j(0)_i + n(G^i(0))_j = n(G(0)_{ji} + G(0)_{ij}).$$

Thus, we obtain that $\nabla^2 F(0) = n(G(0)^T + G(0))$. ◇

Claim 5.4. *There exists a direction of positive curvature for $\nabla^2 F(0)$, i.e., there exists $r \in \mathbb{R}^d$ such that $r^T \nabla^2 F(0) r > 0$.*

Proof of Claim. Consider the trace $\mathrm{Tr}(\nabla^2 F(0))$ of the Hessian at $\mu = 0$. By Claim 5.3,

$$\begin{aligned}
\mathrm{Tr}(\nabla^2 F(0)) &= 2n \,\mathrm{Tr}(G(0)) \\
&= 2n \textstyle\sum_{i=1}^d (\frac{n}{2\pi})^{n/2} \int_{\mathbb{R}^d} y_i (\eta(y) - \eta^*(y))_i e^{\frac{-n\|y-\mu\|^2}{2}} dy \\
&= 2n(\tfrac{n}{2\pi})^{n/2} \int_{\mathbb{R}^d} y^T (\eta(y) - \eta^*(y)) e^{\frac{-n\|y-\mu\|^2}{2}} dy
\end{aligned}$$

By Claim 5.1, $y^T(\eta(y) - \eta^*(y)) \geq 0$ for any $y \in \mathbb{R}^d$ and $y^T(\eta(y) - \eta^*(y)) > 0$ on a set of strictly positive measure. Therefore, $\int_{\mathbb{R}^d} y^T(\eta(y) - \eta^*(y)) e^{\frac{-n\|y\|^2}{2}} dy > 0$.

Therefore, the trace of $\nabla^2 F(0)$ is strictly positive. Since the trace equals the sum of the eigenvalues of $\nabla^2 F(0)$ (see Section 1.2.5 in [61]), we must have at least one strictly positive eigenvalue. The corresponding eigenvector is a direction of positive curvature. ◇

As noted earlier, $F(0) = 0$. Combining Claim 5.4 and Lemma 5.1, there exists $\bar{\mu} \in \mathbb{R}^d$ such that $F(\bar{\mu}) > 0$. □

5.3.2 General covariance

The proof in the previous section focused on the family of normal distributions with the identity as the covariance matrix. We now consider any positive definite covariance matrix Σ for the normal distribution of ξ. In this case, we again consider the function $F(\mu)$ defined in (5.6) and prove that there exists $\bar{\mu}$ such that $F(\bar{\mu}) > 0$. We must adapt the calculations for $\bar{\xi} \sim N(\mu, \frac{1}{n}\Sigma)$. Let us denote the density function of $N(\mu, \frac{1}{n}\Sigma)$ by $g_{\mu,\Sigma}$, i.e.,

$$g_{\mu,\Sigma}(y) := \frac{1}{\sqrt{|\Sigma|}}\left(\frac{n}{2\pi}\right)^{n/2} \exp\left(-\frac{n}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right).$$

Redefining

$$\begin{aligned}
E(\mu) &:= \int_{\mathbb{R}^d}(\eta(y) - \eta^*(y))g_{\mu,\Sigma}(y)dy \\
&= \tfrac{1}{\sqrt{|\Sigma|}}\left(\tfrac{n}{2\pi}\right)^{n/2}\int_{\mathbb{R}^d}(\eta(y) - \eta^*(y))\exp\left(-\tfrac{n}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right)dy, \\
G^i(\mu) &:= \int_{\mathbb{R}^d} y_i(\eta(y) - \eta^*(y))g_{\mu,\Sigma}(y)dy \\
&= \tfrac{1}{\sqrt{|\Sigma|}}\left(\tfrac{n}{2\pi}\right)^{n/2}\int_{\mathbb{R}^d} y_i(\eta(y) - \eta^*(y))\exp\left(-\tfrac{n}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right)dy,\ i = 1, \ldots, d,
\end{aligned}$$

and letting $G(\mu)$ be the matrix with $G^i(\mu)$ as rows, and adapting the calculations from the previous section reveals that

$$\nabla^2 F(0) = n(\Sigma^{-1}G(0) + G(0)^T\Sigma^{-1}).$$

Claim 5.1 again shows that the trace

$$\mathrm{Tr}(G(0)) = \int_{\mathbb{R}^d} y^T(\eta(y) - \eta^*(y))g_{0,\Sigma}(y)dy > 0.$$

This shows that $G(0)$ has an eigenvalue λ with positive *real* part (since $G(0)$ is not guaranteed to be symmetric, its eigenvalues and eigenvectors may be complex). Let the corresponding (possibly complex) eigenvector be v, i.e., $G(0)v = \lambda v$ and $Re(\lambda) > 0$ (denoting the real part of λ). Following standard linear algebra notation, for any matrix/vector M, M^* will denote its Hermitian conjugate [61] (which equals the transpose if the matrix has

real entries). We now consider

$$\begin{aligned} v^*\nabla^2 F(0)v &= n(v^*(\Sigma^{-1}G(0) + G(0)^T\Sigma^{-1})v) \\ &= n(v^*\Sigma^{-1}G(0)v + v^*G(0)^T\Sigma^{-1}v) \\ &= n(v^*\Sigma^{-1}G(0)v + v^*G(0)^*\Sigma^{-1}v) \\ &= n(\lambda(v^*\Sigma^{-1}v) + \lambda^*(v^*\Sigma^{-1}v)) \\ &= 2n(v^*\Sigma^{-1}v)Re(\lambda) \end{aligned}$$

Since Σ is positive definite, so is Σ^{-1}. Therefore $v^*\Sigma^{-1}v > 0$ and we obtain that $v^*\nabla^2 F(0)v > 0$. Since $\nabla^2 F(0)$ is a symmetric matrix, all its eigenvalues are real and its largest eigenvalue γ_d is positive because

$$\gamma_d = \max_{x \in \mathbb{C}^d \setminus \{0\}} \frac{x^*\nabla^2 F(0)x}{x^*x} \geq \frac{v^*\nabla^2 F(0)v}{v^*v} > 0.$$

Thus, $\nabla^2 F(0)$ has a direction of positive curvature and Lemma 5.1 implies that there exists $\bar{\mu} \in \mathbb{R}^d$ such that $F(\bar{\mu}) > 0$.

5.4 An alternate proof for the linear objective based on Bayes' decision rules

To the best of our knowledge, our proof technique for admissibility from the previous sections is new. The conventional way of addressing admissibility uses Bayesian analysis. We provide an alternate proof for Theorem 5.1 using these ideas, which arguably gives a simpler proof. On the other hand, this alternate proof builds upon some well-established facts in statistics, and so is less of a "first principles" proof compared to the one presented in the previous sections. Moreover, as we noted earlier, the new technique of the previous proof might be useful for future admissibility investigations in stochastic optimization.

We now briefly review the relevant ideas from Bayesian analysis. Consider a general statistical decision problem with Θ denoting the states of nature, $\mathcal{A}$ denoting the set of actions, and $\{P_\theta : \theta \in \Theta\}$ the family of distributions on the sample space $\mathcal{X}$. Let $P^\star$ be any so-called *prior distribution* on the parameter space Θ. For any decision rule δ, one can

compute the expected risk, a.k.a., the *Bayes' risk*

$$r(P^\star, \delta) := \mathbb{E}_{\theta \sim P^\star}[R(\theta, \delta)].$$

A decision rule that minimizes $r(P^\star, \delta)$ is said to be a *Bayes' decision rule.*

Theorem 5.4. *[68, Chapter 5, Theorem 2.4] If a decision rule is the unique*[2] *Bayes' decision rule for some prior, then it is admissible.*

The following states that Gaussian distributions are *self-conjugate* [68, Example 2.2].

Theorem 5.5. *Let $d \in \mathbb{N}$ and let $\Sigma \in \mathbb{R}^{d\times d}$ be fixed. For the joint distribution on $(\xi, \mu) \in (\underbrace{\mathbb{R}^d \times \mathbb{R}^d \times \ldots \times \mathbb{R}^d}_{n times}) \times \mathbb{R}^d$ defined by $\xi|\mu \sim \underbrace{\mathcal{N}(\mu, \Sigma) \times \mathcal{N}(\mu, \Sigma) \times \ldots \times \mathcal{N}(\mu, \Sigma)}_{n times}$ and $\mu \sim \mathcal{N}(\mu_0, \Sigma_0)$, we have that $\mu|\xi \sim \mathcal{N}((\Sigma_0^{-1} + n\Sigma^{-1})^{-1}(\Sigma_0^{-1}\mu_0 + n\Sigma^{-1}\overline{\xi}), (\Sigma_0^{-1} + n\Sigma^{-1})^{-1})$.*

Alternate proof of Theorem 5.1. Consider the prior $P^\star$ to be $\mu \sim \mathcal{N}(0, \Sigma)$, then by Theorem 5.5, $\mu|\xi \sim \mathcal{N}\big(\frac{n}{n+1}\overline{\xi}, \frac{1}{n+1}\Sigma\big)$. In particular, the mean of μ, conditioned on the observation ξ is simply a scaling of the sample average $\overline{\xi}$. We now do a standard Bayesian analysis:

$$\begin{aligned} r(P^\star, \delta) &= \mathbb{E}_{\mu\sim\mathcal{N}(0,\Sigma)}[\mathbb{E}_{\xi\sim\mathcal{N}(\mu,\Sigma)^n}[\mathcal{L}(\mu, \delta)]] \\ &= \int_\mu \int_\xi (\mu^T\delta(\xi) - \mu^T x(\mu))p(\xi|\mu)p(\mu)d\xi d\mu \\ &= \int_\mu \int_\xi \mu^T\delta(\xi)p(\xi|\mu)p(\mu)d\xi d\mu - C \end{aligned}$$

where $x(\mu)$ again denotes the optimal solution to $\min\{\mu^T x : x \in X\}$, $p(\xi|\mu)$ denotes the conditional density function of $\xi|\mu$, $p(\mu)$ is the density function of the prior on μ, and the constant C equals $\int_\mu \int_\xi \mu^T x(\mu)p(\xi|\mu)p(\mu)d\xi d\mu = \int \mu^T x(\mu)p(\mu)d\mu$. To find the decision rule δ that minimizes $r(P^\star, \delta)$, we thus need to minimize $\int_\mu \int_\xi \mu^T\delta(\xi)p(\xi|\mu)p(\mu)d\xi d\mu$. We change the order of integration by Fubini's theorem, and rewrite

$$r(P^\star, \delta) = \int_\xi \int_\mu \mu^T\delta(\xi)p(\mu|\xi)p(\xi)d\mu d\xi.$$

[2] Here uniqueness is to be interpreted up to differences on a set of measure zero.

Consequently, given the observation ξ, we choose $\delta(\xi) \in X$ that minimizes the inner integral

$$\int_{\mu} \mu^T \delta(\xi) p(\mu|\xi) d\mu = \frac{n}{n+1} \bar{\xi}^T \delta(\xi).$$

Thus, we may set $\delta(\xi)$ to be the minimizer in X for the linear objective vector $\frac{n}{n+1}\bar{\xi}^T$, which is just a scaling of the sample average. Except for a set of measure zero, any linear objective has a unique solution as was noted in the proof of Claim 5.1. Thus, the Bayes' decision rule is unique and coincides with the sample average estimator δ^n_{SA}. We are done by Theorem 5.4.

□

5.5 Proof of Theorem 5.2 (the scenario with quadratic objective)

A more general version of Theorem 5.4 goes by the name of *Blyth's method*. Here, we state it as in [68] (see Exercise 7.12 in Chapter 5).

Theorem 5.6. *Let $\Theta \subseteq \mathbb{R}^d$ be any open set of states of nature. Suppose δ is a decision rule with a continuous risk function and $\{\pi_n\}_{n\in\mathbb{N}}$ is a sequence of prior distributions such that:*

1. *$r(\pi_n, \delta) = \int \mathcal{R}(\theta, \delta) d\pi_n < \infty$ for all $n \in \mathbb{N}$, where r is the Bayes risk.*
2. *For any nonempty open subset $\Theta_0 \subseteq \Theta$, we have*

$$\lim_{n\to\infty} \frac{r(\pi_n, \delta) - r(\pi_n, \delta^{\pi_n})}{\int_{\Theta_0} \pi_n(\theta) d\theta} = 0$$

where δ^{π_n} is a Bayes decision rule having finite Bayes risk with respect to the prior density π_n. Then, δ is an admissible decision rule.

Proof of Theorem 5.2. For simplicity of exposition, we consider X to be centered at 0, i.e., $\ell_i = -u_i$. The entire proof can be reproduced for the general case by translating the means of the priors to the center of axis-aligned box X. The calculations are much easier to read

and follow when we assume the origin to be the center. We will show that δ^n_{SA} satisfies the conditions of Theorem 5.6 under the priors

$$\mu \sim \mathcal{N}(0, \tau^2\Sigma), \quad \tau \in \mathbb{N}, \text{and } \Sigma \text{ is the covariance matrix of } \xi|\mu.$$

First we obtain a simple expression for the loss function $\mathcal{L}(\mu, x)$ for any action $x \in X$ under the state of nature $\mu \in \mathbb{R}^d$. As noted in Section 5.1.2, $\mathbb{E}_\xi[F(x,\xi)] = \mathbb{E}_\xi[\frac{1}{2}x^Tx - \xi^Tx] = \frac{1}{2}x^Tx - \mu^Tx = \frac{1}{2}\|\mu - x\|^2 - \frac{1}{2}\|\mu\|^2$. Let the minimum value of this for $x \in X$ be denoted by $B(\mu)$. Thus,

$$\begin{aligned} \mathcal{L}(\mu, x) &= \mathbb{E}_{\xi\sim\mathcal{N}(\mu,\Sigma)}[F(x,\xi)] - \mathbb{E}_{\xi\sim D}[F(x(D),\xi)] \\ &= \tfrac{1}{2}\|\mu - x\|^2 - \tfrac{1}{2}\|\mu\|^2 - B(\mu) \end{aligned}$$

Putting this into the numerator of the second condition in Theorem 5.6 and simplifying (which means that the terms $-\frac{1}{2}\|\mu\|^2 - B(\mu)$ cancel out), we need to show that for any open set $\Omega_0 \subseteq \mathbb{R}^d$,

$$\lim_{\tau\to\infty} \frac{\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \|\mu - \delta^n_{SA}(\xi)\|^2 \, p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu - \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \|\mu - \delta_{Bayes}(\xi)\|^2 \, p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu}{\int_{\Omega_0}\pi_\tau(\mu)d\mu} = 0 \tag{5.8}$$

where $p(\xi|\mu) = \mathcal{N}(\mu, \Sigma)$ denotes the conditional density of ξ given μ, and $\pi_\tau(\mu)$ is the marginal density of μ (of course, the marginal density $\pi_\tau(\mu)$ is nothing but the prior $\mathcal{N}(0, \tau^2\Sigma)$).

Next, let us see how the rule δ^n_{SA} given by (5.5) behaves. Minimizing $\frac{1}{2}\|x\|^2 - \overline{\xi}^T x$ is equivalent to minimizing $\frac{1}{2}\|x - \overline{\xi}\|^2$ since $\overline{\xi}$ can be regarded as constant for the optimization problem $\min_{x\in X} \frac{1}{2}\|x\|^2 - \overline{\xi}^T x$. Thus, δ^n_{SA} returns the closest point to $\overline{\xi}$ in X, i.e.,

$$\delta^n_{SA}(\xi) = \text{Proj}_X(\overline{\xi}) \tag{5.9}$$

where the notation $\text{Proj}_X(y)$ denotes the projection of the closes point in X to y.

Let us also see what the Bayes' rule δ_{Bayes} is, i.e., what value of $\delta_{Bayes}(\xi)$ minimizes

$$\begin{aligned}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta^n_{SA}(\xi)\right\|^2 p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta_{Bayes}(\xi)\right\|^2 p(\mu|\xi)m(\xi)d\mu d\xi \\ &= \int_{\mathbb{R}^d}\mathbb{E}_{\mu|\xi}\left\|\mu-\delta_{Bayes}(\xi)\right\|^2 m(\xi)d\xi.\end{aligned}$$

where we have again evaluated the Bayes' risks by switching the order of the integrals and using the conditional density

$$p(\mu|\xi)=\mathcal{N}\left(\frac{n\tau^2}{n\tau^2+1}\bar{\xi},\frac{\tau^2}{n\tau^2+1}\Sigma\right)\qquad\text{(by Theorem 5.5)},\tag{5.10}$$

and the marginal density of ξ is denoted by $m(\xi)$. Since

$$\mathbb{E}[\|Y-a\|^2]=\|\mathbb{E}[Y]-a\|^2+\mathbb{V}[Y]\tag{5.11}$$

for any random variable Y and constant $a\in\mathbb{R}$, one sees that

$$\delta_{Bayes}(\xi)=\mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\bar{\xi}\right).\tag{5.12}$$

Let us now consider the numerator and denominator of the left hand side in (5.8) separately.

Numerator The numerator is:

$$\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\|\mu-\delta^n_{SA}(\xi)\|^2\,p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu-\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\|\mu-\delta_{Bayes}(\xi)\|^2\,p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu$$

$$\begin{aligned}
=\;& \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu\\
&-\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu\\
=\;& \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 p(\xi|\bar\xi)p(\bar\xi|\mu)\pi_\tau(\mu)d\xi d\bar\xi d\mu\\
&-\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 p(\xi|\bar\xi)p(\bar\xi|\mu)\pi_\tau(\mu)d\xi d\bar\xi d\mu\\
=\;& \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 p(\bar\xi|\mu)\pi_\tau(\mu)d\bar\xi d\mu\\
&-\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 p(\bar\xi|\mu)\pi_\tau(\mu)d\bar\xi d\mu\\
=\;& \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 p(\mu|\bar\xi)m(\bar\xi)d\mu d\bar\xi\\
&-\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 p(\mu|\bar\xi)m(\bar\xi)d\mu d\bar\xi,
\end{aligned}$$

where the first equality follows from substituting (5.9) and (5.12), and the last equality follows from the standard trick in Bayesian analysis of switching the order of the integrals.

Since $\bar\xi|\mu\sim\mathcal{N}(\mu,\frac{1}{n}\Sigma)$ and $\pi(\mu)\sim\mathcal{N}(0,\tau^2\Sigma)$, it is a simple exercise to check that $\mu|\bar\xi\sim\mathcal{N}(\frac{n\tau^2}{n\tau^2+1}\bar\xi,\frac{\tau^2}{n\tau^2+1}\Sigma)$ and $m(\bar\xi)\sim\mathcal{N}(0,\frac{n\tau^2+1}{n}\Sigma)$. The formula for the numerator above can then be rewritten as

$$\int_{\mathbb{R}^d}\mathbb{E}_{\mu|\bar\xi}\left\|\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi-\int_{\mathbb{R}^d}\mathbb{E}_{\mu|\bar\xi}\left\|\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi$$

$$\begin{aligned}
=\;& \int_{\mathbb{R}^d}\mathbb{V}(\mu|\bar\xi)+\left\|\mathbb{E}_{\mu|\bar\xi}\mu-\mathrm{Proj}_X(\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi\\
&-\int_{R^d}\mathbb{V}(\mu|\bar\xi)+\left\|\mathbb{E}_{\mu|\bar\xi}\mu-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi\\
=\;& \int_{\mathbb{R}^d}\left\|\tfrac{n\tau^2}{n\tau^2+1}\bar\xi-\mathrm{Proj}_X(\bar\xi)\right\|^2-\left\|\tfrac{n\tau^2}{n\tau^2+1}\bar\xi-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi\\
=\;& \int_{\mathbb{R}^d}\left\|\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)-\mathrm{Proj}_X(\bar\xi)\right\|^2 m(\bar\xi)d\bar\xi\\
&-\int_{\mathbb{R}^d}2\left\langle\tfrac{n\tau^2}{n\tau^2+1}\bar\xi-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi),\mathrm{Proj}_X(\bar\xi)-\mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\bar\xi)\right\rangle m(\bar\xi)d\bar\xi,
\end{aligned}\tag{5.13}$$

where the first equality follows from (5.11), the second equality follows from the formula for the conditional density $p(\mu|\overline{\xi})$, and the last equality follows from the fact that

$$\|a-b\|^2 - \|a-c\|^2 = \|c-b\|^2 - 2\langle a-c, b-c\rangle, \quad \text{for any three vectors } a,b,c \in \mathbb{R}^d.$$

Claim 5.5. *Consider the box $X = \{x \in \mathbb{R}^d : -u_i \le x_i \le u_i \;\; i = 1,\dots,d\}$. Let F be any face of X and so $F' := \frac{n\tau^2+1}{n\tau^2}F$ is a face of $X' := \frac{n\tau^2+1}{n\tau^2}X$. Suppose $\overline{\xi} \in F' + N_{F'}$ where $N_{F'}$ denotes the normal cone at F' with respect to X' (see Lemma 5.2 and the discussion above it), then $\mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$ and $\mathrm{Proj}_X(\overline{\xi})$ both lie in F. Consequently,*

$$\left\langle \frac{n\tau^2}{n\tau^2+1}\overline{\xi} - \mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right), \mathrm{Proj}_X(\overline{\xi}) - \mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)\right\rangle = 0. \tag{5.14}$$

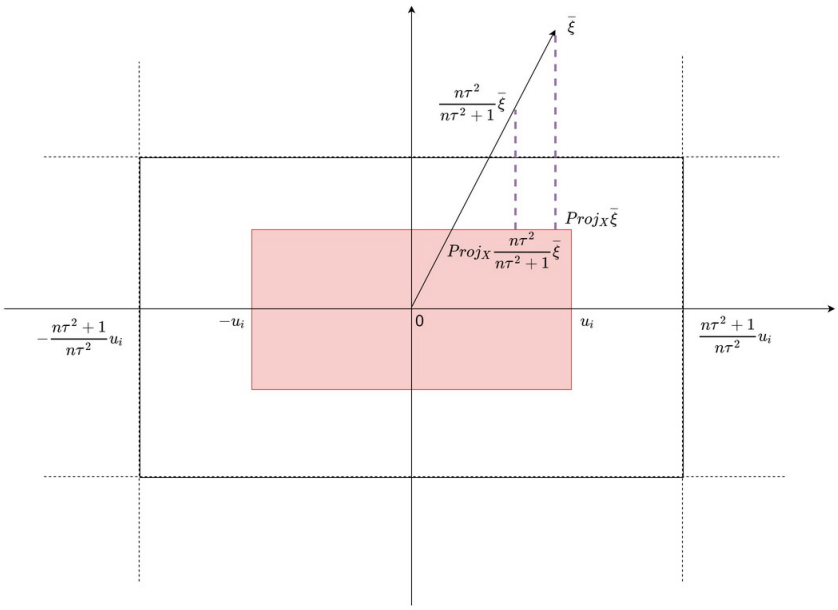

Figure 5.1: The scaling X' of X and the different regions $F' + N_{F'}$ for different faces F of X, with an illustration of Claim 5.5.

*Proof of Claim.*The general formula for the projection onto X is given by

$$\mathrm{Proj}_X(y) = p, \quad \text{where} \quad p_i = sign(y_i)min\{|y_i|, u_i\}. \tag{5.15}$$

Consider any face F of X and any $\overline{\xi} \in F' + N_{F'}$; see Figure 5.1. By Lemma 5.2 part 1.,

$|\overline{\xi}_i| \geq \frac{n\tau^2+1}{n\tau^2}|u_i|$ for all $i \in I_F^+ \cup I_F^-$. Therefore, $|\overline{\xi}_i| \geq \frac{n\tau^2}{n\tau^2+1}|\overline{\xi}_i| \geq |u_i|$ for all $i \in I_F^+ \cup I_F^-$. By the projection formula (5.15), for all $i \in I_F^+ \cup I_F^-$ the i-th coordinate of both $\mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$ and $\mathrm{Proj}_X(\overline{\xi})$ are equal to $sign(\overline{\xi}_i)u_i$. This shows that they both lie on F. By the geometry of projections, the vector $\frac{n\tau^2}{n\tau^2+1}\overline{\xi} - \mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$ is orthogonal to the face F that contains the projection $\mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$. This proves (5.14). $\diamond$

By appealing to (5.14), one can reduce (5.13) to

$$\int_{\mathbb{R}^d} \left\| \mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right) - \mathrm{Proj}_X(\overline{\xi}) \right\|^2 m(\overline{\xi}) d\overline{\xi} \tag{5.16}$$

Using Lemma 5.2 part 2., we decompose the above integral as follows:

$$(5.16) = \sum_{F \text{ face of } X} \int_{F'+N_{F'}} \left\| \mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right) - \mathrm{Proj}_X(\overline{\xi}) \right\|^2 m(\overline{\xi}) d\overline{\xi}.$$

where we have used the notation of Claim 5.5 for F'. Using the formula in Lemma 5.2 part 1., we may simplify the integral further by introducing some additional notation. For any face F of X, recall the notation I_F^+ and I_F^- from Lemma 5.2. Let $I_F^0 := \{1, \ldots, d\} \setminus (I_F^+ \cup I_F^-)$. We introduce the decomposition of any vector $y \in \mathbb{R}^d$ into y^+, y^-, y^0 where $y^+ \in \mathbb{R}^{I_F^+}$ denotes the restriction of the vector y onto the coordinates in I_F^+; similarly, $y^- \in \mathbb{R}^{I_F^-}$ is y restricted to I_F^-, and y^0 is y restricted to I_F^0. Denote the corresponding domains $D_F^+ := \{z \in \mathbb{R}^{I_F^+} : z_i \geq \frac{n\tau^2+1}{n\tau^2}u_i\}$, $D_F^- := \{z \in \mathbb{R}^{I_F^i} : z_i \leq -\frac{n\tau^2+1}{n\tau^2}u_i\}$ and $D_F^0 := \{z \in \mathbb{R}^{I_F^0} : -\frac{n\tau^2+1}{n\tau^2}u_i \leq z_i \leq \frac{n\tau^2+1}{n\tau^2}u_i\}$. By Lemma 5.2 part 1.,

$$\begin{aligned}
(5.16) &= \sum_{F \text{ face of } X} \int_{D_F^0} \int_{D_F^-} \int_{D_F^+} \left\| \mathrm{Proj}_X(\tfrac{n\tau^2}{n\tau^2+1}\overline{\xi}) - \mathrm{Proj}_X(\overline{\xi}) \right\|^2 m(\overline{\xi}) d\overline{\xi}^+ d\overline{\xi}^- d\overline{\xi}^0 \\
&\leq \sum_{F \text{ face of } X} \int_{D_F^0} \int_{D_F^-} \int_{D_F^+} \left(\tfrac{1}{(n\tau^2+1)^2} \textstyle\sum_{i \in I_F^0} \overline{\xi}_i^2 \right) m(\overline{\xi}) d\overline{\xi}^+ d\overline{\xi}^- d\overline{\xi}^0
\end{aligned}$$

where the inequality follows from Claim 5.5 which tells us that both $\mathrm{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$ and $\mathrm{Proj}_X(\overline{\xi})$ lie on F and therefore coincide on the coordinates in $I_F^+ \cup I_F^-$; thus, the coordinates in $I_F^+ \cup I_F^-$ vanish in the integrand. Moreover, on any remaining coordinate i that is not set

to the bound u_i or $-u_i$, the absolute difference in the coordinate is at most $|\frac{n\tau^2}{n\tau^2+1}\overline{\xi}_i - \overline{\xi}_i| = \frac{1}{n\tau^2+1}|\overline{\xi}_i|$, by the projection formula (5.15). Plugging in the formula for $m(\overline{\xi})$, we get

$$\begin{aligned}
(5.16) \quad &\leq \left(\frac{n}{2\pi(n\tau^2+1)\det(\Sigma)^{1/d}}\right)^{d/2} \times \\
&\sum_{F \text{ face of } X} \int_{D_F^0}\int_{D_F^-}\int_{D_F^+} \left(\frac{1}{(n\tau^2+1)^2}\sum_{i\in I_F^0}\overline{\xi}_i^2\right) e^{\frac{-n\overline{\xi}^T\Sigma^{-1}\overline{\xi}}{2(n\tau^2+1)}} d\overline{\xi}^+ d\overline{\xi}^- d\overline{\xi}^0 \\
&= \left(\frac{n}{2\pi(n\tau^2+1)\det(\Sigma)^{1/d}}\right)^{d/2} \times \\
&\sum_{F \text{ face of } X} \int_{D_F^0} \left(\frac{1}{(n\tau^2+1)^2}\sum_{i\in I_F^0}\overline{\xi}_i^2\right) \int_{D_F^+}\int_{D_F^-} e^{\frac{-n\overline{\xi}^T\Sigma^{-1}\overline{\xi}}{2(n\tau^2+1)}} d\overline{\xi}^+ d\overline{\xi}^- d\overline{\xi}^0 \\
&= \left(\frac{n}{2\pi(n\tau^2+1)\det(\Sigma)^{1/d}}\right)^{d/2} \times \\
&\sum_{F \text{ face of } X} \int_{D_F^0} \left(\frac{1}{(n\tau^2+1)^2}\sum_{i\in I_F^0}\overline{\xi}_i^2\right) \cdot C_F(n\tau^2+1)^{(d-\dim(F))/2} h(\overline{\xi}^0,\tau) d\overline{\xi}^0
\end{aligned}$$

where the integral is evaluated using standard Gaussian integral formulas and C_F is a constant independent of τ, $\dim(F)$ denotes the dimension of the face F and $h(\xi^0,\tau)$ is a continuous function which is upper bounded on the compact domain D_F^0 for every F by a universal constant independent of τ.

We now observe that if $I_F^0 = \emptyset$, i.e, F is a vertex of X, then the integrand is simply 0 (in other words, when $\overline{\xi}$ lies in the normal cone of a vertex v of X translated by $\frac{n\tau^2+1}{n\tau^2}v$, the projections $\text{Proj}_X\left(\frac{n\tau^2}{n\tau^2+1}\overline{\xi}\right)$ and $\text{Proj}_X(\overline{\xi})$ are both equal to v, and the integral vanishes). Therefore, we are left with the terms where the face has dimension at least 1. Thus, $(n\tau^2+1)^{(d-\dim(F))/2}$ can be upper bounded by $(n\tau^2+1)^{(d-1)/2}$. Since D_F^0 is a compact domain and h is a continuous function upper bounded by a universal constant independent of τ, we infer the following upper bound on the numerator

$$\left(\frac{n}{2\pi(n\tau^2+1)\det(\Sigma)^{1/d}}\right)^{d/2} \cdot C \cdot (n\tau^2+1)^{(d-5)/2}$$

where C is a constant independent of τ.

Denominator Using the density formula for $\pi_\tau(\mu)$, the denominator of (5.8) is

$$\frac{1}{[2\pi\tau^2]^{d/2}\det(\Sigma)^{\frac{1}{2}}}\int_{\Omega_0} e^{\frac{-\mu^T\Sigma^{-1}\mu}{2\tau^2}} d\mu.$$

Combining the formulas for the numerator and denominator in (5.8), we have:

$$\lim_{\tau\to\infty} \frac{\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta^n_{SA}(\xi)\right\|^2 p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu - \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta_{Bayes}(\xi)\right\|^2 p(\xi|\mu)\pi_\tau(\mu)d\xi d\mu}{\int_{\Omega_0}\pi_\tau(\xi)d\xi}$$

$$\leq \lim_{\tau\to\infty}\left(\frac{n\tau^2}{n\tau^2+1}\right)^{d/2}\cdot C(n\tau^2+1)^{(d-5)/2}\cdot\left(1/\int_{\Omega_0} e^{\frac{-\mu^T\Sigma^{-1}\mu}{2\tau^2}}d\mu\right)$$

As $\tau\to\infty$, $\int_{\Omega_0} e^{\frac{-\mu^T\Sigma^{-1}\mu}{2\tau^2}}d\mu$ approaches the volume of Ω_0 which is strictly positive and the first term $(\frac{n\tau^2}{n\tau^2+1})^{d/2}$ goes to 1. Moreover, since $d\leq 4$, the middle term $(n\tau^2+1)^{(d-5)/2}$ goes to zero. Consequently, we have:

$$\lim_{\tau\to\infty}\frac{\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta^n_{SA}(\xi)\right\|^2 p(\mu|\xi)m(\xi)d\mu d\xi - \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left\|\mu-\delta_{Bayes}\right\|^2 p(\mu|\xi)m(\xi)d\mu d\xi}{\int_{\Omega_0}\pi_\tau(\mu)d\mu}=0$$

By Theorem 5.6, δ^n_{SA} is admissible. □

5.6 Future Work

To the best of our knowledge, a thorough investigation of the admissibility of solution estimators for stochastic optimization problems has not been undertaken in the statistics or optimization literature. There are several avenues for continuing this line of investigation:

1. The most immediate question is whether the sample average solution for the quadratic objective subject to box constraints continues to be admissible for dimension $d\geq 5$. We strongly suspect this to be true, but our current proof techniques are not able to resolve this either way. Any attempt to adapt the dominance proof for the James-Stein estimators for μ breaks down because of the presence of the box constraints. The problem seems to be quite different, and significantly more complicated, compared to the unconstrained case that has been studied in classical statistics literature.

2. The next step, after resolving the higher dimension question, would be to consider general convex quadratic objectives $F(x,\xi)=x^TQx+\xi^Tx$ for some fixed positive

(semi)definite matrix Q and the constraint X to be a general compact, convex set, as opposed to just box constraints. We believe new ideas beyond the techniques introduced in this thesis are needed to analyze the admissibility of the sample average estimator for this convex quadratic program[3]. This problem is important in financial engineering where coherent risk measures can be modeled using the above $F(x, \xi)$.

3. One may also choose to avoid nonlinearities and stick to piecewise linear $F(x, \xi)$ and polyhedral X. Such objectives show up in the stochastic optimization literature under the name of *news-vendor type problems*. The current techniques of this thesis do not easily apply directly to this setting either. In fact, in the simplest setting for the news-vendor problem, one has a function $F : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ given by $F(x, \xi) = cx - p \min\{x, \xi\} + r \max\{0, x - \xi\}$ and $X = [0, U]$ for known constants $0 < r < c < p$ and some given bound $U > 0$. In this setting, the natural distributions for ξ are ones whose support is contained in the nonnegative real axis. As a starting point, one can consider the uniform distribution setting where the mean of the uniform distribution is unknown.

4. For learning problems, such as neural network training with squared or logistic loss, what can be said about the admissibility of the sample average rule, which usually goes under the name of "empirical risk minimization"? Is the empirical risk minimization rule an admissible rule in the sense of statistical decision theory? It is also possible that decision rules that take the empirical risk objective and report a *local* optimum can be shown to dominate decision rules that report the global optimum, under certain conditions. This would be an interesting perspective on the debate whether local solutions are "better" in a theoretical sense than global optima.

[3]When Q is the identity and X is a scaled and translated ℓ_2 unit norm ball, as opposed to a box, the problem becomes equivalent to minimizing a linear function over the ball. Theorem 5.1 then applies to show admissibility in every dimension. This special case is also analyzed in [37].

Chapter 6

Scalable N-way matching of astronomy catalogs

6.1 Motivation

For a long time, astronomers have identified celestial objects by looking at the same part of the sky (after adjusting for the Earth's rotation) and doing a cross matching between past and new observations. Nowadays, with the help of dedicated telescopes systemically scanning the same area that are able to capture various wavelength ranges of the electromagnetic spectrum, astronomers are able to generate lots of surveys that record different properties of objects in the observed area. The problem of associating an object's independent detections coming from these independent catalogs is known as *cross-identification* or *catalog matching*. At this point, we would like to explain some terminology from astronomy for readers not familiar with the subject. Throughout the chapter, we use "object" to denote a physical celestial entity that exists in the observable universe. In addition, "source" and "detection" are used interchangeably to represent an observation of some celestial object that was captured in a catalog by a telescope.

This cross-identification problem has been successfully addressed using the Bayesian formalism, see [27] and [26] for a review. More recently, Budavári and Basu [25] formulated the matching problem as a search for globally optimal associations using combinatorial

optimization, where the marginal likelihood of the entire matched catalog is maximized, and used the Hungarian algorithm ([76]) to solve it. After their proof of concept and efficient assignment approach for two catalogs, Shi et al. [88] extended the algorithm to multiple catalogs using Integer Linear Programming, or ILP for short. However, they only tested their approach with three catalogs. In addition, they only considered a rather special case, where every source shares the same value of the uncertainty of the source directions.

In this chapter, we extend their algorithm to allow for matching across a higher number of catalogs. Also, the extension is applicable to situations where the uncertainty of the source directions are different. For simplicity, we will call this method *CanILP*, as the method considers all *can*didates for possible associations across independent detections and uses *integer linear programming (ILP)* to find the association that maximizes the likelihood of a global matching. As we will discuss later, this naïve extension to the approach in [88] does not scale very well with large number of catalogs. We improve on the previous studies by introducing a novel formulation, hereafter referred to as *DirILP*, where we use ILP to directly assign detections to objects.

Section 6.2 describes the new approach, and Section 6.3 illustrates how the new method scales better with the number of input catalogs. Section 6.4 discusses a public software tool to solve the catalog matching problem. Section 6.5 concludes the study.

6.2 Our Approach

To quantify the associations among independent detections, a relatively recent approach was developed that uses a hierarchical Bayesian formalism. This probabilistic approach to cross-identification was developed in [26]. Here, we will give a summary of the framework.

Suppose there are C catalogs, indexed by $c \in \{1, \ldots, C\}$, with each catalog capturing N_c sources respectively. Let D_{ic} denote the astrometric data for source i in catalog c. We will use the notation (i, c) to denote source i in catalog c. Thus, every observation is tagged with a tuple (i, c). Associated with any such data point (i, c) is a likelihood function $\ell_{ic}(\omega) = p(D_{ic}|\omega)$, for the unknown direction ω which can be thought of as the location of the underlying object producing the data for the source.

Every association hypothesis is a partition of the data (the union of all sources in all catalogs) into subsets, with sources in the same subset hypothesized to represent the same underlying object. Hence, the number of subsets in the partition will constitute the number of hypothesized objects. With $N_{\rm obj}$ objects, we can index every object by an integer $o \in \{1, \ldots, N_{\rm obj}\}$. In addition, let S_o be the set of sources (i, c) associated with object o and C_o be the list of catalogs containing sources associated with object o.

The hierarchical Bayesian framework can be described as follows. We wish to evaluate a "probability" for a particular partitioning of the data $\{D_{ic} : (i, c)\ \text{source}\}$. An important feature of the hierarchical Bayesian framework is that the data from different associations are conditionally independent, given a partition P. More precisely, with each partition P, we associate a likelihood value that is factored across the associations in the partition:

$$\mathcal{L}(P) \equiv p(\{D_{ic}\}|P) = \prod_o \mathcal{M}_o, \tag{6.1}$$

where the marginal likelihood $\mathcal{M}_o$ for the associations of object o is

$$\mathcal{M}_o = \int d\omega\, \rho_{C_o}(\omega) \prod_{(i,c)\in S_o} \ell_{ic}(\omega). \tag{6.2}$$

In the formula above, $\rho_{C_o}(\omega)$ is the prior probability of location of the object producing sources in the association set C_o.

On the other hand, the marginal likelihood for non-association of the sources in S_o is

$$\mathcal{M}_o^{\rm NA} = \prod_{(i,c)\in S_o} \int d\omega\, \rho_c(\omega)\, \ell_{ic}(\omega), \tag{6.3}$$

where $\rho_c(\omega)$ is the prior probability of locations for orphans in catalog c. The Bayes factor is defined as

$$B_o \equiv \frac{\mathcal{M}_o}{\mathcal{M}_o^{\rm NA}}. \tag{6.4}$$

The larger the value of B_o, the more likely that the sources in S_o are from the same object. In particular, a value of $B_o > 1$ indicates that we should treat these sources as

being produced from the same object o rather than as orphans. Hence, the baseline to compare against is that every source comes from a distinct object. The goal is to choose the association hypothesis, i.e., the partition P, that has the highest Bayes factor. Thus, we wish to choose the one with maximum $\prod B_o$.

In order to calculate the Bayes factors B_o, we should first specify a distribution for the member likelihood function $\ell_{ic}(\omega)$. To describe directional uncertainty in astronomy observations, the Fisher distribution [43] is often assumed:

$$\ell_{ic}(\omega) \equiv f(x_{ic}; \omega, \kappa_{ic}) = \frac{\kappa_{ic}}{4\pi \sinh \kappa_{ic}} \exp\left(\kappa_{ic}\, \omega \cdot x_{ic}\right), \tag{6.5}$$

where x_{ic} is the observed direction of source (i, c), ω is the unit vector denoting the direction of the mode, and κ_{ic} is a concentration parameter. When $\kappa_{ic} \gg 1$, the Fisher distribution approximates a Gaussian distribution with standard deviation (in radians) for each coordinate σ_{ic} with $\kappa_{ic} = 1/\sigma_{ic}^2$ and the Bayes factor can be calculated analytically as shown in [27] as follows,

$$B_o = 2^{|S_o|-1} \frac{\prod_{ic} \kappa_{ic}}{\sum_{ic} \kappa_{ic}} \exp\left(-\frac{\sum_{ic} \sum_{i'c'} \kappa_{ic}\kappa_{i'c'}\psi_{ic,i'c'}^2}{4\sum_{ic} \kappa_{ic}}\right), \tag{6.6}$$

where (i, c) and (i', c') are all sources in subset S_o and $\psi_{ic,i'c'}$ is the angle between the directions for sources (i, c) and (i', c').

The next section will discuss how the task of finding an association hypothesis that has the maximum overall Bayes factor $\prod B_o$ can be carried out using an integer linear programming formulation.

6.2.1 CanILP: Optimal Selection of Candidates

Recall from the discussion above that our goal is to maximize the overall likelihood over every object o in a valid partition P, i.e. maximizing $\prod B_o$. This is equivalent to minimizing

$$-\sum_o \ln B_o. \tag{6.7}$$

Given a data set D of all (i, c) pairs for all catalogs c and items i in catalog c, we

introduce a binary variable x_T taking values in $\{0, 1\}$ for each nonempty subset $T \subseteq D$, with the interpretation that $x_T = 1$ indicates that the subset T is included in the partition. To ensure the validity of the partition, we require

$$\sum_{T \ni (i,c)} x_T = 1 \tag{6.8}$$

for every element $(i, c) \in D$. This forces every source (i, c) to be included in exactly one subset of the partition. However, note that for an orphan o, $B_o = 1$. Hence, these coefficients do not contribute to the objective function and we could simply remove those subsets T that have $|T| = 1$. From this, we could modify the above constraint to

$$\sum_{T \ni (i,c)} x_T \leq 1 \tag{6.9}$$

for every element $(i, c) \in D$. In the final solution, if a source (i, c) does not appear in any subset T, we treat it as an orphan. For example, in Figure 6.1, `Source (2,1)` is not included in any subset T, so, in the solution, we will include it as an orphan.

The final integer linear programming function is as follows,

$$\begin{gathered} \min \sum_T w_T x_T \\ \text{subject to } x_T \in \mathbf{Z} \text{ and } 0 \leq x_T \leq 1 \text{ for all } T, \\ \text{and } \sum_{T \ni (i,c)} x_T \leq 1 \text{ for all } (i, c) \in D. \end{gathered} \tag{6.10}$$

Note that the formulation above can be used to solve the matching problem given any number of catalogs C. However, in [88], it was only applied to the 3-catalog case. Also, the authors of [88] only consider the case when κ_{ic} values for all $(i, c) \in D$ are the same and equal to $\frac{1}{\sigma^2}$ for some $\sigma > 0$. In this paper, we attempt to extend and apply this approach to situations with a larger number of catalogs and where κ_{ic} is different for distinct sources. However, as shown later in Section 6.3, even by making use of parallel computing, given a

maximum time of 1 day, the maximum number of catalogs we could handle is 20 catalogs. Hence, we develop another formulation of the catalog matching problem to handle more catalogs.

6.2.2 DirILP: Optimal Direct Associations

For simplicity, we first discuss the special case where the astronometric uncertainty of each detection is the same, i.e., $\sigma_{ic}=\sigma$ for each detection (i, c).

Given a data set D, let N be the total number of detections in all catalogs considered. The number of astronomical objects these represent will be at most N, corresponding to the hypothesis that every detection comes from a different object. Our goal is to find a mapping that matches each source to one (and only one) object. This association between a source and an object means that the source is an observation of that object in the sky. Note that it is possible for multiple sources to get matched with the same object. This represents the hypothesis that all of these sources are observations of the object. To capture the matching between a source (i, c) and an object o, we introduce binary variables $\{x^o_{ic}\}$, where a given $x^o_{ic} = 1$ if the (i, c) detection is associated with object o, and 0 otherwise.

Figure 6.2 illustrates how this approach works. For example, the arrow from `Source (2,1)` to `Object 1` representing an association means that $x^1_{21} = 1$. Similarly, $x^3_{11} = 0$ means no association, hence there is no arrow between the corresponding entries.

A partition P can now be represented as $\{S_o : o \in \{1, 2, ..., N\}\}$, where $S_o := \{(i, c) : x^o_{ic} = 1\}$. Note that if for a given index $o \in \{1, \ldots, N\}$, $x^o_{ic} = 0$ for all (i, c), then $S_o = \emptyset$ meaning that there is no set corresponding to o in the association hypothesis.

Recall that the goal is to maximize the product of Bayes factors $\prod B_o$ (or to minimize $-\sum \ln B_o$) corresponding to these associations. Given an association S_o, assuming $\kappa_{ic} = \kappa$ for all source (i, c), Equation (8) in [88] gives us

$$B_o = 2^{|S_o|-1} \frac{\prod_{ic} \kappa}{\sum_{ic} \kappa} \exp\left(-\frac{\sum_{ic} \sum_{i'c'} \kappa^2 \psi^2_{ic,i'c'}}{4 \sum_{ic} \kappa}\right) \tag{6.11}$$

$$= 2^{|S_o|-1} \frac{\kappa^{|S_o|}}{|S_o|\kappa} \exp\left(-\frac{\kappa \sum_{ic} \sum_{i'c'} \psi^2_{ic,i'c'}}{4|S_o|}\right) \tag{6.12}$$

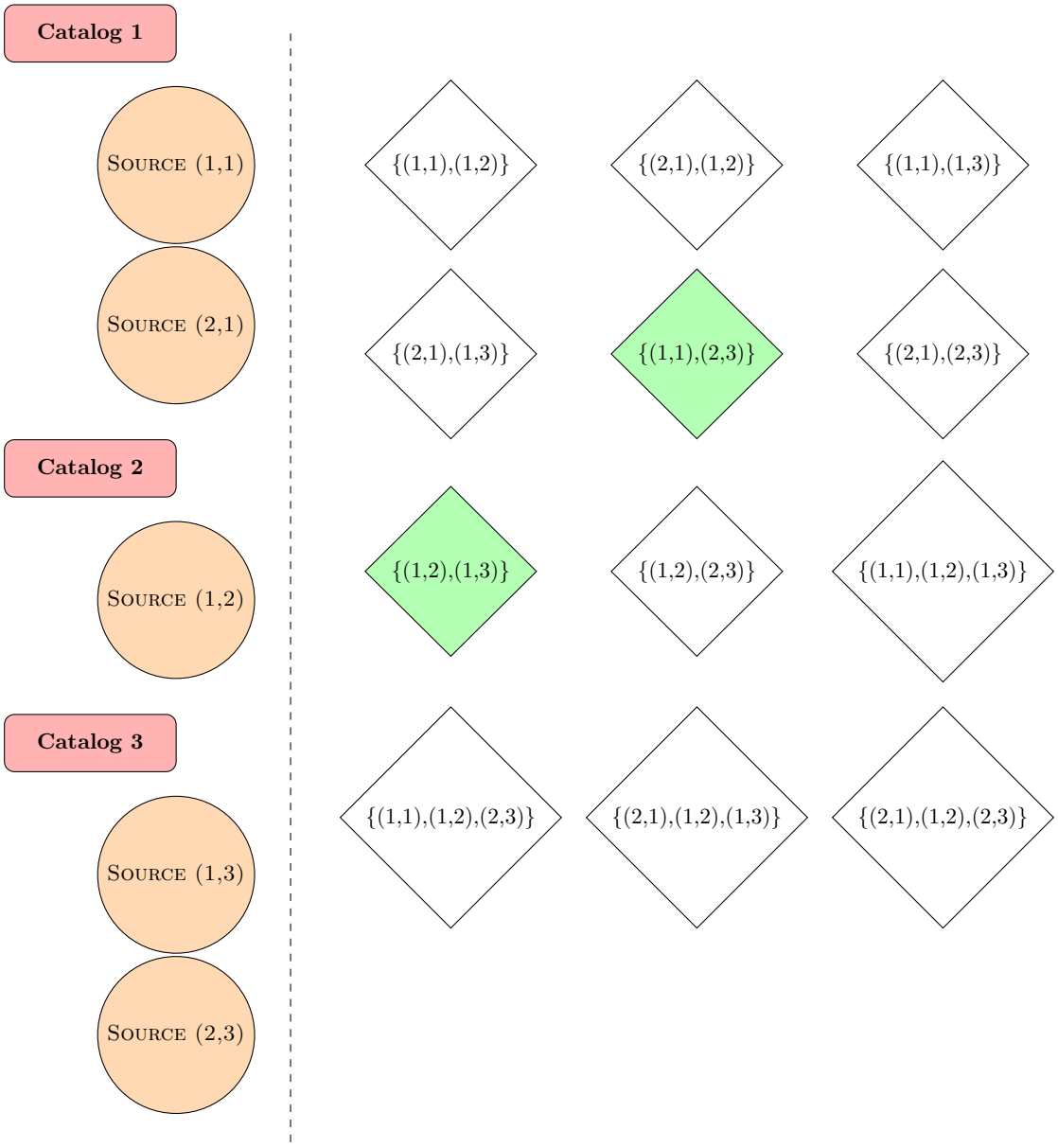

Figure 6.1: An illustration of CanILP. As can be seen on the left side, we assume there are 2 detections from Catalog 1 (`Sources (1,1)` and `(2,1)`), 1 detection from Catalog 2 (`Source (1,2)`) and 2 detections from Catalog 3 (`Sources (1,3)` and `(2,3)`). In CanILP, we list all candidates for possible associations across independent detections, which are shown on the right side. These are the x_T in the formulation. We then find the combinations of subsets that maximize the overall likelihood. Here, the solution given by CanILP indicates that the subsets $\{(1,1),(2,3)\}$ and $\{(1,2),(1,3)\}$ are included in the partition. These subsets, which are represented by a green color, correspond to the variables $x_{\{(1,1),(2,3)\}} = x_{\{(1,2),(1,3)\}} = 1$ in the model. On the other hand, all other variables $x_T = 0$. Notice that because `Source (2,1)` does not appear in any of these subsets, so we treat it as an orphan. As a result, the association outputted by CanILP is $\{\{(1,1),(2,3)\},\{(1,2),(1,3)\},\{(2,1)\}\}$.

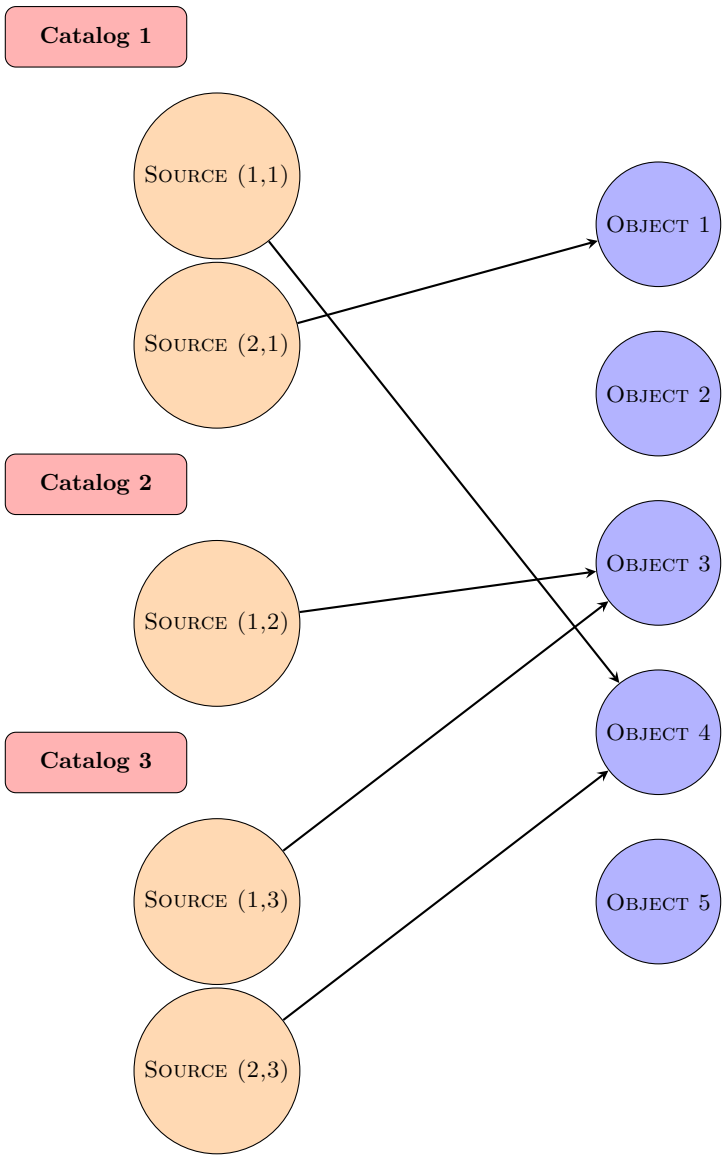

Figure 6.2: An illustration of DirILP. As in Figure 6.1, assume there are 2 detections from Catalog 1 (`Sources (1,1)` and `(2,1)`), 1 detection from Catalog 2 (`Source (1,2)`) and 2 detections from Catalog 3 (`Sources (1,3)` and `(2,3)`). In this case, the output of DirILP indicates that `Sources (1,1)` and `(2,3)` belong to the same object, that `Sources (1,2)` and `(1,3)` belong to the same object, and that `Source (2,1)` is an orphan. Notice that it is okay for an object to not have any source associated with it. The solution given by DirILP is $\{\{(1,1),(2,3)\},\{(1,2),(1,3)\},\{(2,1)\}\}$, which is the same as the one given by CanILP in Figure 6.1.

Hence,

$$-\ln B_o = \ln(2\kappa)(1-|S_o|) + \ln|S_o| + \frac{\sum\limits_{ic,i'c'} \kappa\psi^2_{ic,i'c'}}{4|S_o|} \tag{6.13}$$

We want to find the partition P that minimizes $-\sum_o \ln B_o$. Notice that as of now, there are still several non-linear terms in $-\ln B_o$ so it is not yet a linear objective. To make use of ILP method, we will first need to rewrite this as a linear function. To do that, we introduce the following variables, defined for each $k \in \{0, \ldots, C\}$:

$$z^o_k = \begin{cases} 1 & \text{if } \sum\limits_{ic} x^o_{ic} = k \\ 0 & \text{otherwise} \end{cases}. \tag{6.14}$$

This variable captures the number of sources getting matched to object o, or the cardinality of the subset S_o. When $z^{o'}_{k'} = 1$, there are k' hypothesized observations of object o' in the data. In addition, notice that at most 1 of the z^o_k can be 1. We also introduce

$$y^o_{ic,i'c'} = \begin{cases} 1 & \text{if } x^o_{ic} = x^o_{i'c'} = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{6.15}$$

This is an indicator variable that checks whether the sources (i, c) and (i', c') belong to the same object o. In particular, $y^{o'}_{ic,i'c'} = 1$ indicates the hypothesis that sources (i, c) and (i', c') are observations of object o'. We also have

$$t^o = \begin{cases} \frac{\sum \kappa\psi^2_{ic,i'c'} y^o_{ic,i'c'}}{4k} & \text{if } z^o_k = 1 \text{ for some } k \in [C] \\ 0 & \text{otherwise} \end{cases}, \tag{6.16}$$

where $[C]$ represents the set of numbers $\{1, 2, \cdots, C\}$.

This variable captures the last term in $-\ln B_o$ for a subset S_o. In particular, when $z^o_k = 1$ for some $k \in [C]$, i.e. $|S_o| = k$ by definition of z^o_k, we have

$$t^o = \frac{\sum\limits_{ic,i'c'} \kappa\psi^2_{ic,i'c'}}{4|S_o|} \tag{6.17}$$

as desired, where the summation goes over all (i, c) and (i', c') in S_o. On the other hand,

when $z_0^o = 1$, no detection is assigned to object o so this term should contribute nothing to the objective function. Next, we introduce

$$p^o = \begin{cases} \ln(2\kappa)(1 - |S_o|) & \text{if } z_0^o = 0 \\ 0 & \text{if } z_0^o = 1 \end{cases} . \tag{6.18}$$

This variable captures the first term in $-\ln B_o$ for a subset S_o. It plays a similar role as t^o, i.e. when $z_0^o = 1$, no detection is assigned to object o so this term should contribute nothing to the objective function. On the other hand, if some sources are matched to object $o, p^o = \ln(2\kappa)(1 - |S_o|)$ as desired.

Finally, we will linearize the term $\ln|S_o|$ by breaking the natural log function into finitely many affine linear pieces. We first introduce constants $a_1, a_2, \cdots, a_C$, where $a_p = \ln(p) - \ln(p-1)$, for $p = 2, \cdots, C$ and $a_1 = 0$. Then for each subset S_o, we define binary variables $w_1^o \geq w_2^o \geq \cdots \geq w_C^o$ and impose the constraint that $\sum_{p=1}^C w_p^o = \sum_{ic} x_{ic}^o = |S_o|$.

Using the new notation, we can now express $\ln|S_o|$ as a linear function of w_p^o: $\ln|S_o| = \sum_{p=1}^C a_p w_p^o$. To explain why this is the case, it is best to work with an example. Suppose 3 sources are matched with object o, so $|S_o| = 3$ and $\ln|S_o| = \ln 3$. Because $\sum_{p=1}^C w_p^o = |S_o| = 3$ and w_p^o are 0/1 variables with $w_1^o \geq w_2^o \geq \cdots \geq w_C^o$, we have $w_1^o = w_2^o = w_3^o = 1$ and $w_4^o = w_5^o = \cdots = w_C^o = 0$. Then, $\sum_{p=1}^C a_p w_p^o = a_1 + a_2 + a_3 = (0) + (\ln 2 - \ln 1) + (\ln 3 - \ln 2) = \ln 3$, which is exactly $\ln|S_o|$.

Our objective function now becomes

$$\min \sum_o \left(p^o + \sum_p a_p w_p^o + t^o \right), \tag{6.19}$$

which is linear in the variables p^o, w_p^o and t^o.

As can be seen in the definitions of these variables, there are certain relationships that still need to be modeled using linear constraints because ILP formulations only take linear constraints. The full ILP formulation is given in Appendix B.1, with detailed explanations for how the constraints model the relationships between the variables $x_{ic}^o, y_{ic,i'c'}^o, z_k^o, p^o, w_p^o$, and t^o.

6.2.3 DirILP in General

Next, we remove the assumption that every source has the same measure of uncertainty κ_{ic}.

From equation 6.6, we have,

$$\begin{aligned} -\ln B_o &= (1-|S_o|)\ln 2 - \sum_{ic} \ln \kappa_{ic} + \ln \sum_{ic} \kappa_{ic} + \\ &+ \frac{\sum_{ic}\sum_{i'c'} \kappa_{ic}\kappa_{i'c'}\psi^2_{ic,i'c'}}{4\sum_{ic}\kappa_{ic}}, \end{aligned} \tag{6.20}$$

where all the summations run over all (i,c) and (i',c') in S_o.

We use x^o_{ic}, z^o_k, and $y^o_{ic,i'c'}$ as defined in the special case of Section 6.2.2. We also introduce new variables to convert $-\ln B_o$ to a linear function.

We first try to linearize the term $\ln \sum_{ic} \kappa_{ic}$ using the same trick as when we linearize $\ln|S_o|$ in Section 6.2.2. We introduce constants $b_{min} \equiv b_1, b_2, b_3, \cdots$, where $b_{min} = \ln(\min_{ic\in D} \kappa_{ic})$; Also, define $b_{max} = \ln(C \times \max_{ic\in D} \kappa_{ic})$. Now, if we set an error threshold ε, then the $P \equiv \lceil \frac{b_{max}-b_{min}}{\varepsilon} \rceil$ constants b_p are defined as $b_p = b_{min} + (p-1)\times\varepsilon$, for $p = 1, 2, \cdots, P$. Then for each subset S_o, we define binary variables $\chi^o_1 \geq \chi^o_2 \geq \cdots \geq \chi^o_P$ and impose the constraint $\chi^o_1 \exp(b_1) + \sum_{p=2}^{P} \chi^o_p(\exp(b_p) - \exp(b_{p-1})) \geq \sum_{ic} \kappa_{ic} x^o_{ic}$. Using the new variables, we have $\ln \sum_{ic} \kappa_{ic} \approx \chi^o_1 b_1 + \sum_{p=2}^{P} \chi^o_p(b_p - b_{p-1}) = \chi^o_1 b_{min} + \varepsilon(\sum_{p=2}^{P} \chi^o_p)$, since $b_p - b_{p-1} = \varepsilon$ for all $p \geq 2$.

Again, we will give an example to demonstrate how these variables χ^o_p work. Assume after looking at the data, we determine that $b_{min} = 29$ and $b_{max} = 33$. If we let $\varepsilon = 0.5$, then the value of constants b_p are $\{29, 29.5, \cdots, 32.5, 33\}$. Now suppose there are 3 sources that are matched to an object o with associated κ_{ic} values of $5\times10^{12}, 8\times10^{12}$, and 10^{13}. Then the true value of $\ln \sum_{ic\in S_o} \kappa_{ic}$ is $\ln 2.3\times10^{13}$, which evaluates to 30.77. With the defined variables, the solution given by ILP is $\chi^o_1 = \chi^o_2 = \cdots = \chi^o_5 = 1$ and $\chi^o_6 = \cdots = \chi^o_9 = 0$ because $\chi^o_1 \exp(b_1) + \sum_{p=2}^{P} \chi^o_p(\exp(b_p) - \exp(b_{p-1})) = \exp(29) + \exp(29.5) - \exp(29) + \exp(30) - \exp(29.5) + \cdots + \exp(31) - \exp(30.5) = \exp(31) > 2.3\times10^{13}$, which satisfies the constraint $\chi^o_1 \exp(b_1) + \sum_{p=2}^{P} \chi^o_p(\exp(b_p) - \exp(b_{p-1})) \geq \sum_{ic} \kappa_{ic} x^o_{ic}$. Notice that setting the variables $\chi^o_6, \cdots, \chi^o_9 = 1$ will also satisfy the constraint. However, since we will

model our problem with a minimization objective, the optimal solution will force $\chi_1^o b_1 + \sum_{p=2}^{P} \chi_p^o (b_p - b_{p-1})$ to be as small as possible. Finally, notice that in this case the value of $\chi_1^o b_1 + \sum_{p=2}^{P} \chi_p^o (b_p - b_{p-1})$, which is used to approximate $\ln \sum_{ic \in S_o} \kappa_{ic}$, is 31, which is close to the true value of 30.77.

Next, we will linearize the last term in Equation 6.20 by first introducing the constant $c_{\min} = \min_{ic \in D} \kappa_{ic}$; also, define $c_{max} = C \times \max_{ic \in D} \kappa_{ic}$. Then by rounding these two values to the nearest 100, we can introduce grid points $0 \equiv c_0, c_1, c_2, \cdots, c_Q$, where c_1 is the nearest 100 of c_{min}, c_Q is the nearest 100 of c_{max}, and for all $i > 2$, $c_i = c_1 + 100(i-1)$. We then introduce

$$u_k^o = \begin{cases} 1 & \text{if } \sum_{ic} (\kappa_{ic})^{\approx^{100}} x_{ic}^o = c_k \\ 0 & \text{otherwise} \end{cases}, \tag{6.21}$$

where k ranges in $\{0, 1, \ldots, Q\}$ and the symbol "$\approx^{100}$" is defined as rounding to the nearest 100. This variable attempts to approximate $\sum_{ic \in S_o} \kappa_{ic}$, which appears in the denominator of the last term of Equation 6.20.

p^o and t^o are also very similar to the definitions in Section 6.2.2; however, we need to slightly modify them as follows:

$$t^o = \frac{\sum_{ic} \sum_{i'c'} \kappa_{ic} \kappa_{i'c'} \psi_{ic,i'c'}^2 y_{ic,i'c'}^o}{4 c_k}, \tag{6.22}$$

if $u_k^o = 1$ for some $k \in \{1, 2, \cdots, Q\}$, and $t^o = 0$ otherwise.

The reasoning for defining t^o this way is that if $u_0^o = 1$, $\sum_{(i,c)} (\kappa_{ic})^{\approx^{100}} x_{ic}^o = c_0 = 0$. This happens only when $x_{ic}^o = 0$ for all (i, c), i.e. no sources are matched to object o. Hence, t^o should not contribute to the objective function, hence the value of 0. On the other hand, if $u_k^o = 1$ for some $k > 0$, by definition of u_k, c_k is the best approximation to $\sum_{ic \in S_o} \kappa_{ic}$. Thus,

$$t^o = \frac{\sum_{ic} \sum_{i'c'} \kappa_{ic} \kappa_{i'c'} \psi_{ic,i'c'}^2 y_{ic,i'c'}^o}{4 c_k}, \tag{6.23}$$

as desired.

In addition, we modify p^o as follows,

$$p^o = \begin{cases} (1 - |S_o|) \ln 2 & \text{if } z_0^o = 0 \\ 0 & \text{if } z_0^o = 1 \end{cases} . \tag{6.24}$$

This variable serves a similar function as in the special case, which is to capture the first term in Equation 6.20.

The objective function can now be written as

$$\sum_o \left(p^o - \sum_{ic} x_{ic}^o \ln \kappa_{ic} + \chi_1^o b_{\min} + \varepsilon \sum_{p=2}^{P} \chi_p^o + t^o \right), \tag{6.25}$$

which is linear in all the variables involved.

There are certain relationships that still need to be modeled using linear constraints because ILP formulations only take linear constraints. The full ILP formulation is given in Appendix B.2, with detailed explanations for how the constraints model the relationships between the variables $x_{ic}^o, y_{ic,i'c'}^o, z_k^o, \chi_p^o, u_k^o, p^o$, and t^o.

6.3 Mock Objects and Simulations

We consider the idealized case where all the catalogs capture the same astrological properties of objects in the sky, i.e. they detect the same set of objects. As we generate 100 objects and assume there are C distinct catalogs, we expect to see $100 \times C$ sources and 100 $C-$way association sets. We will now show the catalog matching results using both of our approaches. The ILP programs in both approaches are solved using Gurobi, an optimization solver [57].

6.3.1 Special case: $\kappa_{ic} = \frac{1}{\sigma^2}$ for every detection (i, c)

CanILP formulation analysis. Observe that for the CanILP formulation in Section 6.2.1, we need to list all the possible valid subsets $T \subseteq D$. We could do this by sequentially adding catalogs one by one and considering sources from the new catalog. However, this evaluates to $101^C - 1$ subsets, which is exponential in terms of the number of catalogs C. Hence, we first try to reduce the number of possible subsets by observing that sources that

are far away cannot be from the same object. So we can impose some distance constraints on the sources that are put into the same candidate association set. In doing so, we should be careful not to discard potentially true associations later on because say two sources from the first 2 catalogs that are far away might not be a 2−way matching; but, if on the third catalog, there is a source lying in the middle of the path between these 2 sources, the 3 sources together might be a 3−way matching.

That being said, this suggests an idea for dividing the whole region of the sky that is of interest into different islands where the sources are clustered together so that instead of solving one big problem, we could break it into smaller problems and make use of parallel computing. Essentially, we first apply a single-linkage clustering algorithm to our dataset, which is done using the DBSCAN algorithm with parameters "min samples" = 2 and "eps" $= 5 \times \max_{ic \in D} \sigma_{ic}$. It turns out that for our simulation, most of these islands consist of only 1 source from each catalog. Hence, from now on, we will show the result for this scenario of having 1 source from each catalog. This situation is not peculiar to our simulation but is, in fact, observed in real data sets from multiple visits of the same part of the sky by the same telescope. Analysis for the multiple sources per catalog will be discussed later. As can be seen in Figure 6.3, even though we are able to handle more than 3 catalogs, the maximum number of catalogs we could analyze in a day is 20. The next paragraph discusses how far we could get using DirILP formulation.

DirILP formulation analysis. The main drawback from the previous approach is that the process of creating potential subsets T is exponential in terms of the number of catalogs. Even if we consider the island, the number of nonempty subsets in such an island will still be $2^C - 1$, so creating the variables for the ILP takes a tremendous amount of time.

DirILP formulation attempts to fix that problem by reducing the time complexity to create the variables for the ILP to something that is polynomial in the total number of sources. However, since this catalog matching problem is intrinsically difficult, we still have to tackle the exponential complexity of the problem somewhere else: this appears in the time needed to solve the ILP. We believe that with advances in the field of integer linear programming, the Gurobi solver will be able to solve this problem more efficiently. It turns

out using DirILP, we are able to tackle up to 60 catalogs. The comparison for the total running time between CanILP and DirILP is shown in Figure 6.3. In addition, we also include the set up time and optimization time for each formulation in Figures 6.4 and 6.5.

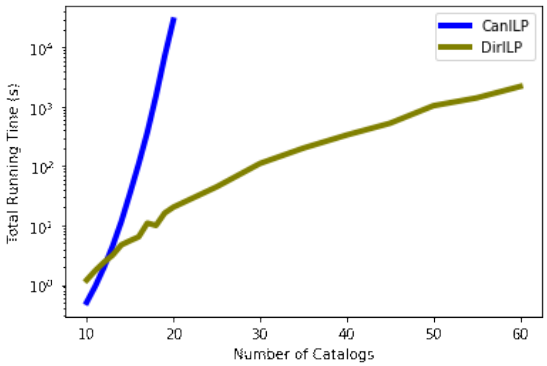

Figure 6.3: Total running time comparison between the two formulations for the special case (Log Scale). Notice that CanILP chokes when there are 20 catalogs.

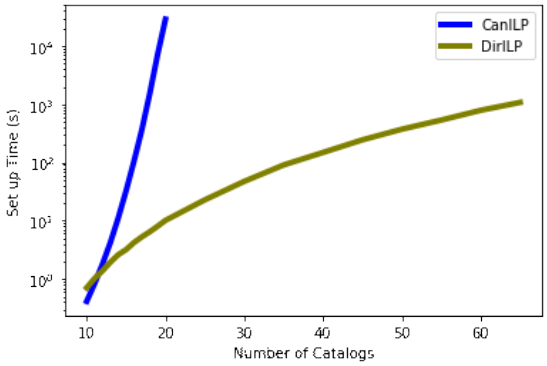

Figure 6.4: Set up time comparison between the two formulations for the special case (Log Scale)

Moreover, by including some heuristic constraints, such as imposing a time limit between incumbent solutions, on the Gurobi solver, we are able to push the DirILP further to handle 160 catalogs.

Finally, it is important to note that the associations given by CanILP and DirILP are the same and they match the ground truth perfectly. Hence, there is no difference in the accuracy of the matching between the two approaches. They only differ in their running time.

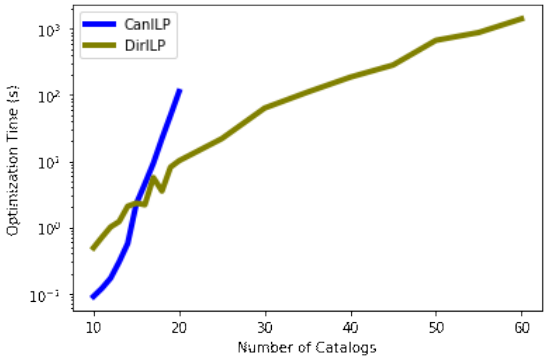

Figure 6.5: Optimization time comparison between the two formulations for the special case (Log Scale)

6.3.2 General case: κ_{ic} is different for every detection (i, c)

For the general case, both approaches still give all correct associations that match the ground truth. However, as in the special case, DirILP is still more efficient at solving the matching problem than CanILP, as shown in Figure 6.6. We should point out that even though in this general setting, the optimal value found in DirILP is just an approximation of the Bayes factor associated with the ground-truth matching, the values are still quite close to each other. More importantly, the associations obtained from DirILP still match the ground-truth associations. Figures 6.6 - 6.8 show the total running time, time to set up the ILP, and time for Gurobi to solve the ILP, for both CanILP and DirILP in this general case.

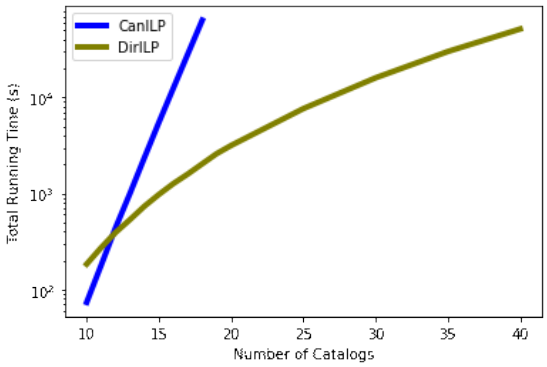

Figure 6.6: Total running time comparison between the two formulations for the general case (Log Scale). Notice that CanILP chokes when there are 18 catalogs.

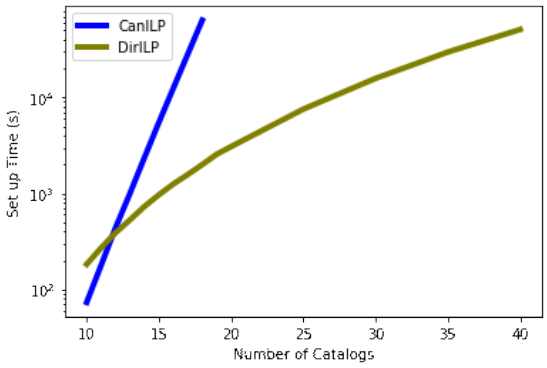

Figure 6.7: Set up time comparison between the two formulations for the general case (Log Scale)

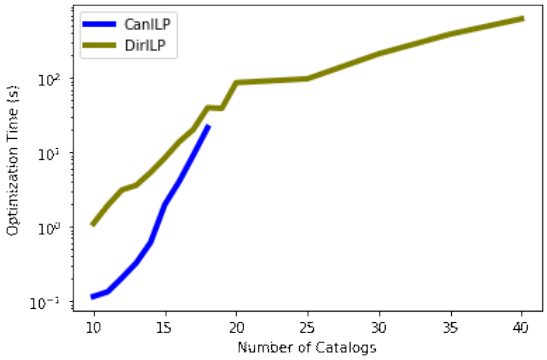

Figure 6.8: Optimization time comparison between the two formulations for the general case (Log Scale)

6.3.3 Multiple sources per catalog in each island

Recall that in the previous sections, we assume that in each island there is only one detection from each catalog, which is a reasonable assumption in many real-life situations. However, in this section, we would like to discuss scenarios when the uncertainty σ_{ic} is large or the sky is very dense. These scenarios will result in islands where there might be multiple detections from each catalog in an island. It turns out that in our simulation, CanILP and DirILP still give the correct association under this scenario. However, both methods run much slower than in the previous scenario. We give an example of the running time for the 2 methods when there are 2 detections from each catalog. One can see how much worse it can get when the number of detections from each catalog becomes larger. Figure 6.9 shows the total running time for both CanILP and DirILP when there are 2 detections from each catalog in an island.

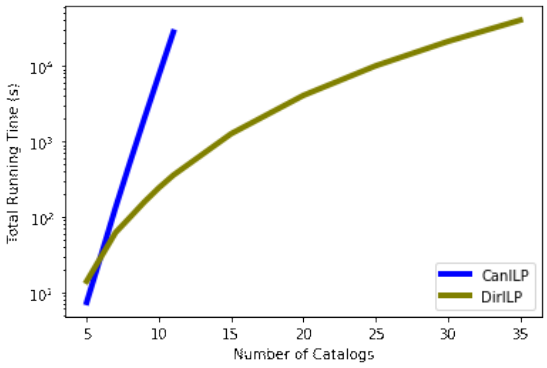

Figure 6.9: Total Running time comparison between the two formulations when there are 2 detections from each catalog in an island (Log Scale)

6.4 Software Tool

The software consists of two Python modules. The first module, *source_clustering.py*, takes a text file containing all Catalog IDs and, for each catalog, an Excel file with all the sources captured in that catalog together with their coordinates and uncertainty measure. It performs the DBSCAN algorithm to output different text files, each containing a list of (i, c) sources together with their coordinates and uncertainty measure σ_{ic}, corresponding to different islands.

After running the first module, users can then feed each of the outputted text file into the second module - *catalog_matching.py* to solve the catalog matching problem in an island. In addition, there is an optional argument for this module that allow users to pick which method they want to use to solve the problem: either "CanILP" or "DirILP".

The software can be found at the url: https://github.com/tunguyen52/Nway-matching.

6.5 Discussion and Conclusion

We now give a brief explanation for the shape of the curves in Figures 6.3 - 6.8. For CanILP, since the number of variables is exponential in terms of the number of catalogs, under the log scale as in Figures 6.4 and 6.7, the time to create these variables and set up the ILP as a function of the number of catalogs is represented by a straight line. On the other hand, for DirILP, we have a curve with decreasing gradient instead of a straight line because the number of variables and constraints in this method is polynomial in the number of catalogs.

The explanation for the curves in Figures 6.5 and 6.8 are similar because the amount of time to solve an ILP generally depends on the number of variables and constraints in the problem. That being said, the curves in these two figures look more jagged because of some randomness involved in the optimization procedure. Finally, as most of the time to solve the catalog matching problem is spent on setting up the ILP, the curves in Figures 6.3 and 6.6 are very much similar to their counterparts in Figures 6.4 and 6.7, respectively.

We have shown how our two methods, CanILP and DirILP, can be used to solve the catalog matching problem. We now discuss in what situations does one of the 2 methods is preferred over the other. DirILP has shown more potential in handling large number of catalogs. However, when there are not many catalogs, in particular, when we have less than 12 catalogs, it is a bit faster to use CanILP. This is true for both the special case and the general case. When there are more than 12 catalogs and if the uncertainty parameter σ_{ic} happens to be the same for every source (i, c), it is recommended to use the DirILP formulation designed for the special case in Section 6.2.2 as it is more efficient. On the other hand, the formulation in Section 6.2.3 should be used when we have a different σ_{ic} for every source.

Appendix A

Appendix to Chapter 5

A.1 Proof of the consistency of δ^n_{SA} in Section 5.1.4

A.1.1 Proof of consistency of δ^n_{SA} for the linear case

Given $\xi^1, \xi^2, \cdots, \xi^n$, define

$$\overline{\xi}_n = \frac{\sum \xi^i}{n}.$$

Then $\delta^n_{SA}(\xi^1, \xi^2, \cdots, \xi^n) = \arg\min\{\overline{\xi}^T x : x \in \mathbf{X}\}$. Also, recall that $x(\mu) = \arg\min\{\mu^T x : x \in \mathbf{X}\}$.

We now prove that $\mathbb{E}\mu^T(\delta^n_{SA}(\xi^1, \xi^2, \cdots, \xi^n) - x(\mu)) \to 0$ as $n \to 0$. Let $\varepsilon > 0$ be given.

Since $\mathbf{X}$ is compact, $\|x\|_2 \leq K$ for $\forall x \in \mathbf{X}$ and for some K. Now, $\forall x \in \mathbf{X}$, we can write $\overline{\xi}^T_n x = \mu^T x + (\overline{\xi}_n - \mu)^T x$. By Cauchy-Schwartz,

$$|\overline{\xi}^T_n x - \mu^T x| = |(\overline{\xi}_n - \mu)^T x| \leq \left\|\overline{\xi}_n - \mu\right\|_2 \cdot \|x\|_2 \leq \left\|\overline{\xi}_n - \mu\right\|_2 K$$
$$\mu^T x - \left\|\overline{\xi}_n - \mu\right\|_2 K \leq \overline{\xi}^T_n x \leq \mu^T x + \left\|\overline{\xi}_n - \mu\right\|_2 K$$

Plugging in δ^n_{SA}, $\overline{\xi}^T_n \delta^n_{SA} \geq \mu^T \delta^n_{SA} - \left\|\overline{\xi}_n - \mu\right\|_2 K \geq \mu^T x(\mu) - \left\|\overline{\xi}_n - \mu\right\|_2 K$ (by def. of δ^n_{SA}).

Plugging in $x(\mu)$, $\mu^T x(\mu) + \left\|\overline{\xi}_n - \mu\right\|_2 K \geq \overline{\xi}_n^T x(\mu) \geq \overline{\xi}_n^T \delta_{SA}^n$ (by def. of $x(\mu)$).

Therefore,

$$
\begin{aligned}
-\left\|\overline{\xi}_n - \mu\right\|_2 K \leq \overline{\xi}_n^T \delta_{SA}^n - \mu^T x(\mu) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K \\
-\left\|\overline{\xi}_n - \mu\right\|_2 K \leq \overline{\xi}_n^T \delta_{SA}^n - \mu^T \delta_{SA}^n + \mu^T \delta_{SA}^n - \mu^T x(\mu) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K \\
-\left\|\overline{\xi}_n - \mu\right\|_2 K \leq (\overline{\xi}_n - \mu)^T \delta_{SA}^n + \mu^T (\delta_{SA}^n - x(\mu)) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K
\end{aligned}
$$

Using the left inequality,

$$
\begin{aligned}
-\left\|\overline{\xi}_n - \mu\right\|_2 K \leq |(\overline{\xi}_n - \mu)^T \delta_{SA}^n| + \mu^T (\delta_{SA}^n - x(\mu)) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K + \mu^T (\delta_{SA}^n - x(\mu)) \\
-2\left\|\overline{\xi}_n - \mu\right\|_2 K \leq \mu^T (\delta_{SA}^n - x(\mu))
\end{aligned}
$$

Using the right inequality,

$$
\begin{aligned}
\mu^T (\delta_{SA}^n - x(\mu)) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K - (\overline{\xi}_n - \mu)^T \delta_{SA}^n \\
\mu^T (\delta_{SA}^n - x(\mu)) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K + |(\overline{\xi}_n - \mu)^T \delta_{SA}^n| \\
\mu^T (\delta_{SA}^n - x(\mu)) \leq \left\|\overline{\xi}_n - \mu\right\|_2 K + \left\|(\overline{\xi}_n - \mu)\right\|_2 \|\delta_{SA}^n\|_2 \\
\mu^T (\delta_{SA}^n - x(\mu)) \leq 2\left\|\overline{\xi}_n - \mu\right\|_2 K
\end{aligned}
$$

Hence, we can conclude that $|\mu^T(\delta_{SA}^n - x(\mu))| \leq 2\left\|\overline{\xi}_n - \mu\right\|_2 K$. Since $\overline{\xi}_n \sim N(\mu, \frac{1}{n}\Sigma), \overline{\xi}_n - \mu \sim N(0, \frac{1}{n}\Sigma)$ and we have $\mathbb{E}\left\|\overline{\xi}_n - \mu\right\|_2 = \sqrt{\frac{1}{n}\operatorname{Tr}(\Sigma)}$. As $n \to \infty$, the RHS and hence $\mathbb{E}\left\|\overline{\xi}_n - \mu\right\|_2 \to 0$. As a result, $\mathbb{E}\mu^T(\delta_{SA}^n - x(\mu)) \to 0$.

A.1.2 Proof of consistency of δ_{SA}^n for the quadratic case

Recall that $\delta_{SA}^n(\xi^1, \xi^2, \cdots, \xi^n) = \operatorname{Proj}_X \overline{\xi}_n$. In order to prove the asymptotic consistency of the sample average approximation rule, we need to show that $\mathbb{E}(\left\|\operatorname{Proj}_X \overline{\xi}_n - \mu\right\|_2 - \|\operatorname{Proj}_X \mu - \mu\|_2) \to 0$. This is equivalent to proving $\mathbb{E}\left\|\operatorname{Proj}_X \overline{\xi}_n - \operatorname{Proj}_X \mu\right\|_2 \to 0$ because by triangle inequality, we have $|\left\|\operatorname{Proj}_X \overline{\xi}_n - \mu\right\|_2 - \|\operatorname{Proj}_X \mu - \mu\|_2| \leq \left\|\operatorname{Proj}_X \overline{\xi}_n - \operatorname{Proj}_X \mu\right\|_2$ and so if $\mathbb{E}(\text{RHS}) \to 0$, we have our result.

From the property of orthogonal projection, we know that $\left\|\text{Proj}_X \bar{\xi}_n - \text{Proj}_X \mu\right\|_2 \leq \left\|\bar{\xi}_n - \mu\right\|_2$. Using same argument as in the linear case, $\mathbb{E}\left\|\bar{\xi}_n - \mu\right\|_2 \to 0$, which implies $\mathbb{E}\left\|\text{Proj}_X \bar{\xi}_n - \text{Proj}_X \mu\right\|_2 \to 0$ and we are done.

A.2 Proof of Lemma 5.1 in Section 5.2

We consider 2 cases:

Case 1: $\nabla F(0) \neq 0$. Define $\phi(t) = F(0 + t\nabla F(0))$. Since F is continuously differentiable, ϕ is continuously differentiable. In fact,

$$\phi'(t) = \nabla F(t)^T \nabla F(0).$$

Hence, $\phi'(0) = \|\nabla F(0)\|^2$, which is positive. By the continuity property of ϕ, there exists $\bar{\alpha} > 0$ such that $\phi'(\alpha) > 0, \forall \alpha \in [0, \bar{\alpha}]$. Let $\lambda \in (0, \bar{\alpha}]$. By the Mean Value Theorem,

$$\phi(\lambda) = \phi(0) + \phi'(\eta)\lambda$$

for some $\eta \in [0, \alpha] \subseteq [0, \bar{\alpha}]$. As a result, for $z = \lambda \nabla F(0), F(z) = F(\lambda \nabla F(0)) = \phi(\lambda) > 0$.

Case 2: $\nabla F(0) = 0$. By the hypothesis, $\exists d \in \mathbb{R}^d$ such that $d^T \nabla^2 F(0) d > 0$. Define $\phi(t) = F(0 + td)$. Then, $\phi'(t) = \nabla F(t)^T d$ and $\phi''(t) = d^T \nabla F^2(t) d$. Clearly, $\phi(0) = \phi'(0) = 0$ and $\phi''(0) > 0$. Hence, by the continuity property of ϕ'', there exists $\bar{\alpha} > 0$ such that $\phi''(\alpha) > 0, \forall \alpha \in [0, \bar{\alpha}]$. Let $\lambda \in (0, \bar{\alpha}]$. By the Mean Value Theorem,

$$\phi(\lambda) = \phi(0) + \phi'(0)\lambda + 1/2\phi''(\eta)\lambda^2$$

for some $\eta \in [0, \alpha] \subseteq [0, \bar{\alpha}]$. As a result, for $z = \lambda d, F(z) = F(\lambda d) = \phi(\lambda) > 0$.

A.3 Proof of Lemma 5.2 in Section 5.2

First, notice that X is a polyhedron given by $Ax \leq b$, where

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}, b = \begin{bmatrix} u_1 \\ u_1 \\ \cdots \\ u_n \\ u_n \end{bmatrix}.$$

Let $F \subseteq X$ be a face. We know from Theorem 2.1 that there exists a subset $I \in \{1, \cdots, 2d\}$ such that $F = \{\mathbf{x} \in P : A_I \mathbf{x} = \mathbf{b}_I\}$.

Lemma A.1. *$N_F = cone(a^i,\ i \in I)$, where a^i's are columns of matrix A.*

Proof. **Forward inclusion:** Let $r \in N_F$ and $y \in \text{relint}\,(F)$. Define $\text{cone}(X - y) := \{\lambda(x - y), x \in X, \lambda \geq 0\}$. Then, for all $v \in \text{cone}(X - y)$, we have

$$\langle r, v \rangle = \lambda \langle r, x - y \rangle \leq 0. \tag{A.1}$$

We now prove that $\text{cone}(X - y) = \{v \in \mathbb{R}^d : \langle a^i, v \rangle \leq 0,\ \forall i \in I\} =: S$.

Let $v \in \text{cone}(X - y)$, i.e. $v = \lambda(x - y)$, for some $\lambda \geq 0, x \in X$. Then, $\forall i \in I, \langle a^i, v \rangle = \lambda \langle a^i, x - y \rangle \leq 0$ since $\langle a^i, x \rangle \leq b_i$ and $\langle a^i, y \rangle = b_i$.

Conversely, let $v \in S$, thus $\langle a^i, v \rangle \leq 0,\ \forall i \in I$. We will prove that $\lambda v + y \in X$ for some $\lambda > 0$ and hence $v \in \text{cone}(X - y)$. Indeed, for $i \in I$ and for any $\lambda > 0$, we have $\langle a^i, \lambda v + y \rangle = \lambda \langle a^i, v \rangle + \langle a^i, y \rangle \leq b_i$ because $\langle a^i, v \rangle \leq 0$ and $\langle a^i, y \rangle = b_i$. For $j \in [2d] \backslash I, \langle a^j, \lambda v + y \rangle = \lambda \langle a^j, v \rangle + \langle a^j, y \rangle$. Since $\langle a^j, y \rangle < b_j$, we can choose a λ_j small enough so that $\lambda_j \langle a^j, v \rangle + \langle a^j, y \rangle \leq b_j$. By letting $\lambda = \min_j \{\lambda_j\}$, we have $\langle a^j, \lambda v + y \rangle \leq b_j,\ \forall j \in [2d] \backslash I$. Hence, $v \in \text{cone}(X - y)$.

From Equation A.1, we know that $r \in \text{cone}(X - y)^\circ$ by definition. From the result above, this means $r \in S^\circ$. In fact, using Lemma 6.45 in [84], $S^\circ = \text{cone}(a^i,\ i \in I)$. Hence, $r \in \text{cone}(a^i,\ i \in I)$.

Backward inclusion: Let $r \in \text{cone}(a^i,\ i \in I)$, thus $r = \sum_{i \in I} \lambda^i a^i,\ \lambda > 0$. Then for any

$x \in X$, we have

$$\langle r, y - x \rangle = \sum_{i \in I} \lambda^i \left(\langle a^i, y \rangle - \langle a^i, x \rangle \right) \geq 0$$

because for any $i \in I, \langle a^i, y \rangle = b_i$ and $\langle a^i, x \rangle \leq b_i$. Hence, by definition, $r \in N_F$.

For our polyhedron X, notice that every column of A has only 2 nonzero entries, corresponding to the inequalities $x_i \leq u_i$ and $-x_i \leq u_i$. Hence, for a face F, because at most 1 of the 2 inequalities can become an equality, $\forall i \in I, a^i = e_i$ or $a^i = -e^i$, where e_i is the unit vector in with 1 in the ith entry and 0 everywhere else. Now any $x \in F + N_F$ can be expressed as $y + \sum_{i \in I} \lambda_i a^i$, for some $y \in F, \lambda_i \geq 0$. Then, for $i \in I_F^+, x_i = u_i + \lambda_i \geq u_i$. For $i \in I_F^-, x_i = -u_i - \lambda_i \leq -u_i$. Finally, for $i \notin I_F^+ \cup I_F^-, x_i = y_i$, for some $-u_i \leq y_i \leq u_i$. Thus,

$$F + N_F = \left\{ x \in \mathbb{R}^d : \begin{array}{ll} x_i \geq u_i & i \in I_F^+ \\ x_i \leq -u_i & i \in I_F^- \\ -u_i \leq x_i \leq u_i & i \notin I_F^+ \cup I_F^- \end{array} \right\}.$$

Finally, one can see from their defintion that these sets $F + N_F$ for different face F form a partition of $\mathbb{R}^d$. □

Appendix B

Appendix to Chapter 6

B.1 DirILP Formulation - Special Case

The ILP Formulation for DirILP when $\kappa_{ic} = \frac{1}{\sigma^2}$ for every source (i, c) is given below.

The objective function we want to minimize is given by

$$\min \sum_o \left(\ p^o + \sum_p a_p w_p^o + t^o \right) \tag{B.1}$$

The following constraints restrict $x_{ic}^o, y_{ic,i'c'}^o, z_k^o, w_p^o$ to binary variables and t^o to have non-negative values.

$$x_{ic}^o, y_{ic,i'c'}^o, z_k^o, w_p^o \in \mathbb{Z} \text{ and } 0 \leq x_{ic}^o, y_{ic,i'c'}^o, z_k^o, w_p^o \leq 1, \ 0 \leq t^o, \ \ \forall (i,c), k, p, o. \tag{B.2}$$

The next equation ensures that all sources (i, c) need to belong to exactly one subset:

$$\sum_o x_{ic}^o = 1, \ \ \forall (i,c) \tag{B.3}$$

The following equation imposes that every subset takes no more than 1 source from each catalog.

$$\sum_i x_{ic}^o \leq 1, \ \ \forall o \in \{1, 2, ..., N\}, \ \ \forall c \in \{1, \ldots, C\} \tag{B.4}$$

The following set of constraints on $y^o_{ic,i'c'}$ is an implementation of the definition of $y^o_{ic,i'c'}$ in Section 6.2.2, which requires $y^o_{ic,i'c'} = 1$ only if $x^o_{ic} = x^o_{i'c'} = 1$:

$$y^o_{ic,i'c'} \geq x^o_{ic} + x^o_{i'c'} - 1, \tag{B.5}$$

$$y^o_{ic,i'c'} \leq x^o_{ic}, \tag{B.6}$$

$$y^o_{ic,i'c'} \leq x^o_{i'c'}, \tag{B.7}$$

for all $(i, c) \neq (i', c')$ and $\forall o$.

Since the cardinality of any subset from a partition P is between 0 and C, the following equation states that only 1 of z^o_k can take a value of 1.

$$\sum_{k=0}^{C} z^o_k = 1, \forall o, \tag{B.8}$$

The next constraint is the definition of w^o_p as described in Section 6.2.2.

$$w^o_1 \geq w^o_2 \geq \cdots \geq w^o_C \text{ and } \sum_p w^o_p = \sum_{ic} x^o_{ic}, \quad \forall o, \tag{B.9}$$

With the specific choice of the constant M as defined below, the equation that follows becomes redundant when $z^o_k = 0$ since RHS will be negative and so $t_o \geq 0$ becomes the enforcing constraint, and when $z^o_k = 1$, the minimization forces t^o to be equal to the first term of the RHS.

$$t^o \geq \frac{\sum \kappa \psi^2_{ic,i'c'} y^o_{ic,i'c'}}{4k} - (1 - z^o_k)M, \quad \forall o \text{ and } k \in \{1, 2, \cdots, C\}, \tag{B.10}$$

where $M = \left\lceil \sum_{ic,i'c' \in D} \frac{\kappa \psi^2_{ic,i'c'}}{4} \right\rceil$.

The following set of equations constitutes the definition of z_k^o.

$$\sum_{ic} x_{ic}^o \leq k z_k^o + C(1 - z_k^o) \tag{B.11}$$

$$\sum_{ic} x_{ic}^o \geq k z_k^o, \tag{B.12}$$

for all $k \in \{0, 1, 2, \cdots, C\}$ and for all o.

Finally, the last equation

$$p^o \geq \ln(2\kappa)(1 - \sum_p w_p^o) - \ln(2\kappa) z_0^o, \ \ \forall o, \tag{B.13}$$

ensures that for an empty subset S_o, $p^o = 0$, hence contributing nothing to the objective. This is because when $z_0^o = 1$ (nothing is assigned to subset S_o), $w_p^o = 0, \forall p$. As we are minimizing the objective function with respect to p^o, p^o will be set to 0. On the other hand, when $z_0^o = 0$, the constraint becomes $p^o \geq \ln(2\kappa)(1 - \sum_p w_p^o)$ and again, since we are minimizing, p^o will equal this value.

B.2 DirILP Formulation - General Case

Below, we give the ILP Formulation for DirILP when κ_{ic} is different for distinct sources (i, c). Some of these constraints are similar to the special case so we will only give explanations for the new constraints, which are shown after the ILP formulation.

$$\min \sum_o \left(\; p^o - \sum_{ic} x^o_{ic} \ln \kappa_{ic} + \chi^o_1 b_{\min} + \varepsilon \sum_{p=2}^{P} \chi^o_p + t^o \right)$$

$$\text{subject to } x^o_{ic}, y^o_{ic,i'c'}, z^o_k, \chi^o_p, u^o_k \in \mathbb{Z} \text{ and } 0 \le x^o_{ic}, y^o_{ic,i'c'}, z^o_k, \chi^o_p, u^o_k \le 1, \;\; 0 \le t^o,$$

$$\sum_o x^o_{ic} = 1, \;\; \forall (i,c), \tag{B.14}$$

$$\sum_i x^o_{ic} \le 1, \;\; \forall o \in \{1,2,...,N\}, \;\; \forall c \in \{1,\ldots,C\} \tag{B.15}$$

$$\begin{aligned} y^o_{ic,i'c'} &\ge \; x^o_{ic} + x^o_{i'c'} - 1 \\ y^o_{ic,i'c'} &\le x^o_{ic} \qquad\qquad , \;\; \forall (i,c) \ne (i',c') \text{ and } \forall o, \\ y^o_{ic,i'c'} &\le x^o_{i'c'} \end{aligned} \tag{B.16}$$

$$\sum_{k=0}^{C} z^o_k = 1, \forall o, \tag{B.17}$$

$$\sum_{k=0}^{Q} u^o_k = 1, \forall o, \tag{B.18}$$

$$\chi^o_1 \ge \chi^o_2 \ge \cdots \ge \chi^o_P \text{ and } \chi^o_1 \exp(b_1) + \sum_{p=2}^{P} \chi^o_p (\exp(b_p) - \exp(b_{p-1})) \ge \sum_{ic} \kappa_{ic} x^o_{ic}, \;\; \forall o, \tag{B.19}$$

$$t^o \ge \frac{\sum_{ic} \sum_{i'c'} \kappa_{ic} \kappa_{i'c'} \psi^2_{ic,i'c'} y^o_{ic,i'c'}}{4c_k} - (1 - u^o_k) M, \;\; \forall o \text{ and } k \in \{1,2,\cdots,Q\}, \tag{B.20}$$

$$\begin{aligned} \textstyle\sum_{ic} x^o_{ic} &\le k z^o_k + C(1 - z^o_k) \\ \textstyle\sum_{ic} x^o_{ic} &\ge k z^o_k \end{aligned} \quad , \;\; \forall k \in \{0,1,2,\cdots,C\} \text{ and } \forall o, \tag{B.21}$$

$$\begin{aligned} \textstyle\sum_{ic} (\kappa_{ic}) \approx^{100} x^o_{ic} &\le c_k u^o_k + M'(1 - u^o_k) \\ \textstyle\sum_{ic} (\kappa_{ic}) \approx^{100} x^o_{ic} &\ge c_k u^o_k \end{aligned} \quad , \;\; \forall k \in \{0,1,2,\cdots,Q\} \text{ and } \forall o, \tag{B.22}$$

$$p^o \ge (1 - \sum_{ic \in S_o} x^o_{ic}) \ln 2 - z^o_0 \ln 2, \;\; \forall o. \tag{B.23}$$

where $M = \left\lceil \frac{\max_{ic \in D} \kappa^2_{ic} \sum_{ic,i'c' \in D} \psi^2_{ic,i'c'}}{4 \min_{ic \in D} \kappa_{ic}} \right\rceil$ and $M' = C \max_{ic \in D} \kappa_{ic}$.

Equation B.18 combined with Equation B.22 says that the value of $\sum_{ic \in S_o} \kappa_{ic}$ will be approximately equal to c_k for some $k \in \{0,1,2,\cdots Q\}$. Equation B.19 is the definition of χ^o_p as described in Section 6.2.3. Again, this gives an approximation for $\sum_{ic \in S_o} \kappa_{ic}$.

List of notation

$\text{aff}(A)$	.....	the affine hull of A
$\text{conv}(A)$	.....	the convex hull of A
$\dim(A)$	.....	the dimension of A
$\text{relint}(A)$	.....	the relative interior of A
$[n]$	.....	the set of integers $\{1, \ldots, n\}$
$\mathbb{R}_+$	.....	the nonnegative real numbers
$\mathbb{R}^n_+$	.....	the vectors in $\mathbb{R}^n$ with nonnegative entries
$\mathbb{R}^{n\times k}$	.....	the collection of real-valued $n \times k$ matrices
$\mathbb{Z}_+$	.....	the set of nonnegative integers
$\mathbb{Z}^n_+$	.....	the set of vectors in $\mathbb{R}^n$ with nonnegative, integer entries
$A \times B$	.....	the Cartesian product of two sets A and B
$\text{Tr}(A)$	.....	Trace of matrix A
$\mathbf{1}_m$	.....	the vector of all 1's in $\mathbb{R}^m$
i.i.d.	.....	independent and identically distributed
$\mathcal{N}(\mu, \sigma^2)$	.....	the normal distribution with mean μ and variance σ^2
$\mathbb{E}(X)$	.....	the expected value of a random variable X
$\mathbb{V}(X)$	.....	the variance of a random variable X
$\mathbb{P}(A)$	.....	probability of event A

Bibliography

[1] Alvarez, A.M., Louveaux, Q., Wehenkel, L.: A machine learning-based approximation of strong branching. INFORMS Journal on Computing 29(1), 185–195 (2017)

[2] Arora, S., Barak, B.: Computational Complexity - A Modern Approach. Cambridge University Press (2009)

[3] Balas, E., Saxena, A.: Optimizing over the split closure. Mathematical Programming 113(2), 219–240 (2008)

[4] Barvinok, A.: A Course in Convexity, vol. 54. American Mathematical Society, Providence, Rhode Island (2002)

[5] Basu, A., Conforti, M., Di Summa, M.: A geometric approach to cut-generating functions. Mathematical Programming 151(1), 153–189 (2015)

[6] Basu, A., Hildebrand, R., Köppe, M.: Light on the infinite group relaxation I: Foundations and taxonomy. 4OR 14(1), 1–40 (2016)

[7] Basu, A., Hildebrand, R., Köppe, M.: Light on the infinite group relaxation II: Sufficient conditions for extremality, sequences, and algorithms. 4OR 14(2), 1–25 (2016)

[8] Basu, A., Paat, J.: Operations that preserve the covering property of the lifting region. SIAM Journal on Optimization 25(4), 2313–2333 (2015)

[9] Basu, A., Sankaranarayanan, S.: Can cut-generating functions be good and efficient? SIAM Journal on Optimization 29(2), 1190–1210 (2019)

[10] Benati, S., Rizzi, R.: A mixed integer linear programming formulationof the optimal mean/value-at-risk portfolio problem. European Journal of Operational Research (2007)

[11] Bengio, Y., Lodi, A., Prouvost, A.: Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operations Research (2020)

[12] Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer Science & Business Media (2013)

[13] Bertsimas, D., Gupta, V., Kallus, N.: Data-driven robust optimization. Mathematical Programming 167(2), 235–292 (2018)

[14] Bickel, P.J.: Minimax estimation of the mean of a normal distribution when the parameter space is restricted. The Annals of Statistics 9(6), 1301–1309 (1981)

[15] Bickel, P.J., Doksum, K.A.: Mathematical statistics: basic ideas and selected topics, volume I, vol. 117. CRC Press (2015)

[16] Bilgen, B., Ozkarahan, I.: A mixed-integer linear programming model for bulk grain blending and shipping. International Journal of Production Economics (2007)

[17] Birge, J.R., Louveaux, F.: Introduction to stochastic programming. Springer Science & Business Media (2011)

[18] Bland, R.G.: New finite pivoting rules for the simplex method. Mathematics of Operations Research (1977)

[19] Bonami, P., Lodi, A., Zarpellon, G.: Learning a classification of mixed-integer quadratic programming problems. In: van Hoeve, W.J. (ed.) Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 595–604. Springer International Publishing, Cham (2018)

[20] Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004), http://dx.doi.org/10.1017/CBO9780511804441

[21] Breiman, L.: Bagging predictors. Machine Learning (1996)

[22] Breiman, L.: Random forests. Machine Learning (2001)

[23] Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth Statistics/Probability (1984)

[24] Brown, L.D.: Fundamentals of Statistical Exponential Families: With Applications in Statistical Decision Theory. Hayward, CA: Institute of Mathematical Statistics (1986)

[25] Budavári, T., Basu, A.: Probabilistic cross-identification in crowded fields as an assignment problem. The Astronomical Journal (Oct 2016)

[26] Budavári, T., Loredo, T.J.: Probabilistic record linkage in astronomy: Directional cross-identification and beyond. Annual Review of Statistics and Its Application 2(1), 113–139 (Oct 2015)

[27] Budavári, T., Szalay, A.S.: Probabilistic cross-identification of astronomical sources. The Astrophysical Journal 679(1), 301–309 (2008)

[28] Casella, G., Strawderman, W.E.: Estimating a bounded normal mean. The Annals of Statistics pp. 870–878 (1981)

[29] Charras, A., Van Eeden, C.: Bayes and admissibility properties of estimators in truncated parameter spaces. Canadian Journal of Statistics 19(2), 121–134 (1991)

[30] Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)

[31] Chu, L.Y., Shanthikumar, J.G., Shen, Z.J.M.: Solving operational statistics via a bayesian analysis. Operations Research Letters 36(1), 110–116 (2008)

[32] Chvátal, V.: Edmonds polytopes and weakly hamiltonian graphs. Mathematical Programming 5(1), 29–40 (1973)

[33] Conforti, M., Cornuéjols, G., Zambelli, G.: Integer programming, vol. 271. Springer (2014)

[34] Cook, W.J., Cunningham, W.H., Pulleyblank, W., Schrijver, A.: Combinatorial Optimization (1997)

[35] Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press, Princeton, N.J. (1963)

[36] Dash, S., Günlük, O., Lodi, A.: Mir closures of polyhedral sets. Mathematical Programming 121(1), 33–60 (2010)

[37] Davarnia, D., Cornuéjols, G.: From estimation to optimization via shrinkage. Operations Research Letters 45(6), 642–646 (2017)

[38] Davarnia, D., Kocuk, B., Cornuéjols, G.: Bayesian solution estimators in stochastic optimization. http://www.optimization-online.org/DB_HTML/2017/11/6318.html (2018)

[39] Dey, S.S., Wolsey, L.A.: Two row mixed-integer cuts via lifting. Mathematical Programming 124(1–2), 143–174 (2010)

[40] Elmachtoub, A.N., Grigas, P.: Smart" predict, then optimize". arXiv preprint arXiv:1710.08005 (2017)

[41] Esfahani, P.M., Kuhn, D.: Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. Mathematical Programming 171(1-2), 115–166 (2018)

[42] Ferguson, T.S.: Mathematical statistics: A decision theoretic approach. Academic Press (1967)

[43] Fisher, R.: Dispersion on a sphere. Proceedings of the Royal Society of London Series A (1953)

[44] Fourdrinier, D., Marchand, É.: On bayes estimators with uniform priors on spheres and their comparative performance with maximum likelihood estimators for estimating bounded multivariate normal means. Journal of Multivariate Analysis 101(6), 1390–1399 (2010)

[45] Fourdrinier, D., Strawderman, W.E., Wells, M.T.: Shrinkage estimation. Springer (2018)

[46] Fox, J.: Applied Regression Analysis and Generalized Linear Models. SAGE Publications, Inc (2015)

[47] Garey, M., Johnson, D.: Computers and intractability. W. H. Freeman and Co., San Francisco, Calif. (1979), a guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences

[48] Gass, S.I., Vinjamuri, S.: Cycling in linear programming problems. Computers and Operations Research (2004)

[49] Gatsonis, C., MacGibbon, B., Strawderman, W.: On the estimation of a restricted normal mean. Statistics & probability letters 6(1), 21–30 (1987)

[50] Goldreich, O.: Computational complexity: a conceptual perspective. Cambridge University Press (2008)

[51] Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bull. Amer. Math. Soc. 64, 275–278 (1958)

[52] Gomory, R.E.: Some polyhedra related to combinatorial problems. Linear Algebra and Appl. 2, 451–558 (1969)

[53] Gomory, R.: An algorithm for the mixed integer problem. Tech. rep., DTIC Document (1960)

[54] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

[55] Gorman, J.D., Hero, A.O.: Lower bounds for parametric estimation with constraints. IEEE Transactions on Information Theory 36(6), 1285–1301 (1990)

[56] Gupta, V., Rusmevichientong, P.: Small-data, large-scale linear optimization with uncertain objectives. Available at SSRN 3065655 (2017)

[57] Gurobi Optimization, L.: Gurobi optimizer reference manual (2020), http://www.gurobi.com

[58] Hartigan, J.A.: Uniform priors on convex sets improve risk. Statistics & probability letters 67(4), 285–288 (2004)

[59] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer New York Inc. (2008)

[60] He, H., III, H.D., Eisner, J.M.: Learning to search in branch and bound algorithms. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3293–3301. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5495-learning-to-search-in-branch-and-bound-algorithms.pdf

[61] Horn, R., Johnson, C.: Matrix Analysis. John Wiley & Sons, second edn. (1985)

[62] James, W., Stein, C.: Estimation with quadratic loss. In: Proceedings of the fourth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 361–379 (1961)

[63] Karimnezhad, A.: Estimating a bounded normal mean relative to squared error loss function. Journal of Sciences, Islamic Republic of Iran 22(3), 267–276 (2011)

[64] Khachiyan, L.: A polynomial algorithm in linear programming. Doklady Akademii Nauk SSSR 244, 1093–1096 (1979)

[65] Khalil, E.B., Le Bodic, P., Song, L., Nemhauser, George L. & Dilkina, B.N.: Learning to branch in mixed integer programming. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16) (2016)

[66] Klee, V., Minty, G.J.: How good is the simplex algorithm? Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin) (1972)

[67] Kumar, S., Tripathi, Y.M.: Estimating a restricted normal mean. Metrika 68(3), 271–288 (2008)

[68] Lehmann, E.L., Casella, G.: Theory of point estimation. Springer Science & Business Media (2006)

[69] Lehmann, E.L., Romano, J.P.: Testing statistical hypotheses. Springer Science & Business Media (2006)

[70] Liyanage, L.H., Shanthikumar, J.G.: A practical inventory control policy using operational statistics. Operations Research Letters 33(4), 341–348 (2005)

[71] Lodi, A., Zarpellon, G.: On learning and branching: a survey. TOP: An Official Journal of the Spanish Society of Statistics and Operations Research 25(2), 207–236 (2017)

[72] Marchand, É., Perron, F.: Improving on the mle of a bounded normal mean. The Annals of Statistics 29(4), 1078–1093 (2001)

[73] Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press (2018)

[74] Morais, H., Kadar, P., Faria, P., Vale, Z.A., Khodr, H.: Optimal scheduling of a renewable micro-grid in an isolated load area usingmixed-integer linear programming. Renewable Energy (2010)

[75] Mukherjee, S., Niyogi, P., Poggio, T., Rifkin, R.: Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. Advances in Computational Mathematics (2006)

[76] Munkres, J.: Algorithms for the assignment and transportation problems (1957)

[77] Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press (2012)

[78] Myers, S.C., Pogue, G.A.: A programming approach to corporate financial management. The Journal of Finance (1973)

[79] Nesterov, Y.E., Nemirovski, A.S.: Interior-point polynomial algorithms in convex programming, SIAM Studies in Applied Mathematics, vol. 13. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1994), http://dx.doi.org/10.1137/1.9781611970791

[80] Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)

[81] Rao, C.R.: Information and the Accuracy Attainable in the Estimation of Statistical Parameters, pp. 235–247. Springer New York, New York, NY (1992), https://doi.org/10.1007/978-1-4612-0919-5_16

[82] Rice, J.A.: Mathematical Statistics and Data Analysis. Cengage Learning (2006)

[83] Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton, New Jersey (1970)

[84] Rockafellar, T.R., Wets, R.: Variational Analysis. Springer (1998)

[85] Sahu, P.K., Pal, S.R., Das, A.K.: Estimation and Inferential Statistics. Springer India (2015)

[86] Schrijver, A.: Theory of linear and integer programming. Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Ltd., Chichester (1986), a Wiley-Interscience Publication

[87] Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on stochastic programming: modeling and theory. SIAM (2009)

[88] Shi, X., Budavári, T., Basu, A.: Probabilistic cross-identification of multiple catalogs in crowded fields. The Astrophysical Journal 870(1), 51 (2019)

[89] Soolaki, M., Mahdavi, I., Mahdavi-Amiri, N., Hassanzadeh, R., Aghajani, A.: A new linear programming approach and genetic algorithm for solving airline boarding problem. Applied Mathematical Modelling (2012)

[90] Stein, C.: Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In: Proceedings of the third Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 197–206 (1956)

[91] Tang, Y., Agrawal, S., Faenza, Y.: Reinforcement learning for integer programming: Learning to cut. In: International Conference on Machine Learning (2020)

[92] Terlaky, T., Zhang, S.: Pivot rules for linear programming: A survey on recent theoretical developments. Annals of Operations Research (1993)

[93] Tikhonov, A.N.: Nonlinear Ill-Posed Problems. Springer Netherlands (1998)

[94] Tuy, H.: Convex Analysis and Global Optimization. Springer (2016)

[95] Van der Vaart, A.W.: Asymptotic statistics, vol. 3. Cambridge university press (2000)

[96] Van Parys, B.P., Esfahani, P.M., Kuhn, D.: From data to decisions: Distributionally robust optimization is optimal. arXiv preprint arXiv:1704.04118 (2017)

[97] Xie, X., Kou, S., Brown, L.D.: Sure estimates for a heteroscedastic hierarchical model. Journal of the American Statistical Association 107(500), 1465–1479 (2012)

[98] Xu, L., Hutter, F., Hoos, H. H. & Leyton-Brown, K.: Hydra-MIP: Automated algorithm configuration and selection for mixed integer programming. In: Proceedings of the 18th RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion. pp. 16 – 30 (2011)

[99] Yin, J., Yang, L., Tang, T., Gao, Z., Ran, B.: Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. Transportation Research Part B: Methodological (2017)

Curriculum Vitae

Tu Nguyen was born in Hanoi, Vietnam on February 5, 1993. In 2011, he came to the United States to attend Randolph College. In the spring of 2015, Tu graduated as *valedictorian* of his class with Bachelors of Science degree in Mathematics, Economics, and Physics. He then enrolled in the Applied Mathematics and Statistics Ph.D. program at Johns Hopkins University in the same year. During his time at Johns Hopkins, he completed his Master of Science in Engineering degree in Financial Mathematics in 2019. At Johns Hopkins, Tu is a recipient of the Acheson J. Duncan Fund for the Advancement of Research in Statistics Travel Award. In 2019, he worked as an Data Scientist Intern at Quicken Loans and he will come back with a full time Data Scientist position in the Fall of 2020.