CONVEX ALGEBRAIC GEOMETRY WITH APPLICATIONS TO POWER SYSTEMS, STATISTICS AND OPTIMIZATION

by

Julia Lindberg

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2022

Date of Final Oral Examimation: 5/6/2022

The dissertation is approved by the following members of the Final Oral Committee:

B. Lesieutre, Professor, Electrical and Computer Engineering (ECE)

J. Rodriguez, Assistant Professor, Mathematics

N. Boston, Professor Emeritus, ECE and Mathematics

L. Roald, Assistant Professor, ECE

D. Molzahn, Assistant Professor, ECE

ABSTRACT

Many important problems in engineering are large scale and nonlinear – two things that are inherently at odds. As a result, it is desirable to make use of underlying structure to reduce the computational complexity. This work considers problems in power systems, statistics and optimization that have underlying algebraic structure and exploits this structure to answer relevant questions.

The first part of this thesis focuses on the power flow equations. Using techniques from numerical algebraic geometry, we design a novel algorithm that dramatically increases the speed with which we can find all solutions to the power flow equations. We study the real solution sets for various families of networks and show that there exist electrical parameter values for which the number of complex solutions is achieved by all real solutions. Finally, we find outer approximations of the convex hull of the real solutions to these equations.

The next chapter studies data center geographic load shifting. We propose a metric to allow data center operators to shift load independently of collaboration with independent system operators. We demonstrate the superiority of our metric over other commonly used metrics using a years worth of data.

The next part of this thesis focuses on a classical problem in statistics, density estimation for Gaussian mixture models (GMMs). We study the variety stemming from the moment equations of GMMs and provide upper bounds on the number of complex solutions to these equations. We apply these results to GMMs in $\mathbb{R}^n$ and design a homotopy algorithm that performs density estimation where the number of paths scales linearly in n.

We conclude by considering problems in optimization. We first study maximum likelihood estimation where the statistical model is defined by system of sparse polynomial equations. We

give an expression for the number of critical points for this problem and show that the number of critical points is determined by the Newton polytopes of the model. Next, we study the Shor relaxation of quadratic programs. We characterize the geometry of the set of objective functions for which this relaxation is exact and apply these results to the quadratic binary programs.

ACKNOWLEDGMENTS

I would like to thank my thesis advisors Bernie Lesieutre and Jose Rodriguez for all of the time, patience and wisdom they have gifted me. I am also grateful to Nigel Boston for the guidance and support during the initial years of my graduate studies as well as to Line Roald for serving as both a collaborator and role model. I would also like to thank Daniel Molzahn for serving on my committee and providing thoughtful feedback and insightful comments. I am truly lucky to have been surrounded by a group of faculty to teach me how to produce high quality research while having fun and fostering an environment of endless curiosity.

I owe a great deal of gratitude to all of my former teachers and professors whose early encouragement has lead to this dissertation. Specifically, to Andrea Harris, Shamgar Gurevich, Amir Assadi and Alexandra Kjuchukova for giving me the confidence during my undergraduate years to continue on to my graduate studies.

I would like to thank all of my family and friends who kept me motivated and sane during these past five years. Especially to those in the ECE GSA who have been able to commiserate and provide respite from the stress of deadlines and exams as well as to my lab mates and fellow graduate students. Finally, a special thanks to my family for the endless love and support. Thank you Mom, Dad, Evan, Amanda and Hervin, and most importantly, my precious Giselle.

TABLE OF CONTENTS

Page

Appendix

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Appendix
Figure Page

NOMENCLATURE

$\mathbb{R}[x_1, \ldots, x_n]$	Ring of real-valued polynomials in n variables
$\mathcal{PSD}_n$	Cone of $n \times n$ positive semidefinite matrices
$\mathbb{C}^*$	The set $\mathbb{C} \backslash \{0\}$
C_n	Cyclic graph on n vertices
K_n	Complete graph on n vertices
Conv$\{a, b\}$	Convex hull of points a and b
Vol_n	The n dimensional Euclidean volume
MVol(P, Q)	The mixed volume of P and Q
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathbf{i}$	$\sqrt{-1}$
$\mathbb{E}[X]$	Expected value of random variable X

Chapter 1

Introduction

Exploiting algebraic structure in problems arising in the sciences is at the core of applied algebraic geometry. Most of these problems involve polynomial systems of equations whose solutions have some type of physical meaning. The work in this thesis considers problems arising in power systems engineering, statistics and optimization that have underlying algebraic structure and utilizes this structure to develop novel solutions to these problems. The work of this thesis splits into three self contained sections.

1.1 Power systems engineering

Power systems engineering is a field in electrical engineering that studies the operation and planning of electric power networks. Critical to these studies is the understanding of *operating points* to electric power networks. Operating points are calculated as the real solutions to the power flow equations, a system of quadratic polynomial equations that encodes the underlying physics of electric power networks. Understanding these equations is a critical problem in this field, one which the first two chapters of this thesis makes gains towards.

1.2 Statistics

Statistics is a branch of math that aims to rigorously characterize empirical data. A classic problem in this domain is *density estimation*. Density estimation asks one to fit a density to a finite set of data. This problem has been studied in many contexts for many different families of

densities. The work in this thesis considers two techniques for density estimation: the method of moments and maximum likelihood estimation.

1.3 Optimization

Optimization problems are problems in math that seek to minimize an objective function to some set of constraints. These problems have applications in many other disciplines from machine learning and operations research to signal processing and the aerospace industry. Broadly speaking, if your optimization problem is *convex* then it can be solved efficiently to global optimality. Unfortunately, many important problems are nonconvex. Nonconvex optimization is an active field of research. An important tool used in nonconvex optimization is the use of convex relaxations. A particular relaxation of interest is the *semidefinite relaxation*. This relaxation is applied to polynomial optimization problems and has seen much success in applications. The work in this thesis studies when this relaxation is exact for a variety of quadratic polynomial optimization problems.

Chapter 2

Solving the power flow equations

A main focus of this thesis work is the *optimal power flow (OPF)* problem. This is a nonconvex, quadratically constrained, quadratic program that aims to optimally dispatch generation profiles of electric power networks subject to physical and engineering constraints. The *power flow equations* are a system of quadratic polynomial equations that comprise the nonconvex constraints. We begin by outlining the power flow equations.

2.1 The power flow equations

We model an n-node electric power network as a connected, undirected graph, $G = (V, E)$, where each vertex $v_m \in V$, $0 \leq m \leq n-1$, represents a node (bus) in the power network. There is an edge, e_{km} between vertices v_k and v_m if the corresponding nodes in the power network are connected. Each edge has a known complex admittance $b_{km} + \mathbf{i}g_{km}$ where $b_{km}, g_{km} \in \mathbb{R}$ and $\mathbf{i} = \sqrt{-1}$. We take the admittance $b_{km} + \mathbf{i}g_{km}$ to be zero if the vertices k and m are not connected. Each vertex v_k has an associated complex power injection $P_k + \mathbf{i}Q_k$, $P_k, Q_k \in \mathbb{R}$ where P_k models the active power and Q_k models the reactive power.

At each node, v_k, the relationship between the active and reactive power flows is captured by the nonlinear relations

$$P_k = \sum_{m=0}^{n-1} |V_k||V_m|(g_{km}\cos(\theta_k - \theta_m) + b_{km}\sin(\theta_k - \theta_m)) \tag{2.1.1}$$

$$Q_k = \sum_{m=0}^{n-1} |V_k||V_m|(g_{km}\sin(\theta_k - \theta_m) + b_{km}\cos(\theta_k - \theta_m)) \tag{2.1.2}$$

where V_k is the complex voltage magnitude and θ_k represents the complex voltage angle at node v_k. We fix v_0 to be the *slack bus*, meaning $\theta_0 = 0$. Equations (2.1.1)-(2.1.2) are the *power flow equations*.

We consider a power network where all nodes, except the slack node, have unknown reactive power injections but maintain constant voltage magnitude: Q_k is unknown while P_k and $|V_k|$, are known constants. These nodes are called PV nodes since the active power injection P_k and voltage magnitude $|V_k|$ are known. They model typical generator buses.

We can make this system algebraic by introducing the change of variables $x_k = |V_k|\cos(\theta_k)$ and $y_k = |V_k|\sin(\theta_k)$. Under this transformation $(2.1.1)$ becomes the system of $2(n-1)$ equations in $2(n-1)$ variables

$$P_k = \sum_{m=0}^{n-1} g_{km}(x_k x_m + y_k y_m) + b_{km}(x_m y_k - x_k y_m) \tag{2.1.3}$$

$$|V_k|^2 = x_k^2 + y_k^2. \tag{2.1.4}$$

Equations (2.1.3) and (2.1.4) are the power flow equations for the special case where all nodes are PV nodes. The *power flow problem* is to compute all of the real-valued solutions to the system of equations (2.1.3)-(2.1.4). At the slack node $(x_0, y_0) = (|V_0|, 0)$.

In practice $g_{km} \ll b_{km}$, so we assume that the power network is $lossless$, meaning $g_{km} = 0$ for all nodes k, m. In addition, we assume that the voltage magnitude $V_k = 1$ for all k. For $k = 1, \ldots, n-1$, this turns (2.1.3)-(2.1.4) into the system

$$\begin{aligned} P_k &= \sum_{m=0}^{n-1} b_{km}(x_k y_m - x_m y_k) \\ 1 &= x_k^2 + y_k^2. \end{aligned} \tag{2.1.5}$$

The reference node in this case is then $x_0 = 1, y_0 = 0$. One special case of (2.1.5) is when there are zero active power injections, namely when $P_k = 0$ for $k = 1, \ldots, n-1$. This case is studied because it typically admits the most real-valued solutions[1]. Under the assumption of zero active power injections, the power flow equations for a graph on n vertices admits 2^{n-1} trivial solutions

[1]Certain rare examples have been found where this fails to be true [1]

corresponding to $y_k = 0, x_k = \pm 1$. This means finding all complex solutions to (2.1.5) is then reduced to finding all nontrivial solutions.

Work in this area can largely be categorized as

1. Finding all real solutions [2, 3, 4, 5, 6, 7, 8, 9]

2. Bounding the number of $\mathbb{C}^*$ solutions [10, 11, 12, 13]

3. Studying distributions of the number of real solutions [1, 14, 15]

We outline previous work done in each of these areas and present new contributions we made to each of these topics. Before getting to the details, we outline techniques from numerical algebraic geometry used to find all complex solutions to a polynomial system of equations.

2.2 Homotopy Continuation

Finding all complex solutions to a system of polynomial equations has received considerable attention in the past few decades. There exist many techniques to do this, and we outline two of the most popular techniques below.

2.2.1 Total degree and polyhedral homotopies

We briefly outline the main idea of homotopy continuation below but give [16, 17] as more detailed references. Say you would like to solve a system of polynomial equations

$$F(x) = \{p_1(x_1, \ldots, x_m), p_2(x_1, \ldots, x_m), \ldots, p_m(x_1, \ldots, x_m)\} = 0.$$

where we assume that the number of solutions to $F(x) = 0$ is finite. The main idea is to introduce a variable t and to construct a *homotopy* from a *start system* G to a *target system* F such that:

1. The solutions to $G(x) = 0$ are trivial to find,

2. There are no singularities along the path $t \in [0, 1)$, and

3. All isolated solutions of $F(x) = 0$ can be reached [18].

One such example of a homotopy, known as a *straight line homotopy* is :

$$H(x;t) = \gamma(1-t)G(x) + tF(x). \tag{2.2.1}$$

By using a random $\gamma \in \mathbb{C}$ each path for $t \in [0,1)$ avoids singularities almost surely, so condition 2 is easily met. This is referred to as the *gamma trick* [17]. Continuation methods are then used to track the solutions from $G(x) = 0$ to $F(x) = 0$ as t varies from 0 to 1. These methods are called *predictor-corrector methods* and are commonly used numerical techniques [19].

There are many choices for the start system $G(x)$. A *total degree* start system is

$$G(x) = \{x_1^{d_1} - 1, \ldots, x_n^{d_n} - 1\} = 0 \tag{2.2.2}$$

where d_i is the degree of p_i [17]. The number of solutions to $G(x) = 0$ is $d_1 \cdots d_n$, which is the Bezout bound. This means that you have to track $d_1 \cdots d_n$ paths in order to get all solutions to $F(x) = 0$. If $F(x) = 0$ has close to $d_1 \cdots d_n$ solutions this is a reasonable start system.

If $F(x)$ is sparse, the number of solutions to $F(x) = 0$ can be much less than $d_1 \cdots d_n$ so tracking $d_1 \cdots d_n$ paths is wasteful computation. In this case it is often more computationally efficient to use a *polyhedral* start system. These homotopy algorithms rely on the Bernstein-Kushnirenko-Khovanskii (BKK) bound [20, 21, 22], which gives an upper bound on the number of isolated $\mathbb{C}^* = \mathbb{C}\backslash\{0\}$ solutions for polynomial systems. This upper bound relates the number of zeros of $F(x) = 0$ to properties of convex polytopes associated with $F(x)$.

Definition 2.2.1. Let $p(x_1, \ldots, x_m) = \sum_{k=1}^{K} c_k x^{a_k}$ be a polynomial where the notation x^a is shorthand for $x^a = x_1^{a_1} \cdots x_m^{a_m}$. The *Newton polytope* of p is $\mathrm{Newt}(p) = \mathrm{Conv}\{a_1, \ldots, a_K\}$.

Example 2.2.2. Consider the system of polynomial equations

$$f_1(x_1, x_2) = 3x_1^2 + 2x_1x_2 - x_2 - 13x_1x_2^2 + 3$$
$$f_2(x_1, x_2) = 7x_1x_2^2 + 3x_2^2 - 2x_1x_2 - x_1 - 6x_1^2 + 6.$$

The Newton polytopes of f_1 and f_2 are $\mathrm{Newt}(f_1) = \mathrm{Conv}\{(2,0),(1,1),(0,1),(1,2),(0,0)\}$ and $\mathrm{Newt}(f_2) = \mathrm{Conv}\{(1,2),(0,2),(1,1),(1,0),(2,0),(0,0)\}$ where Conv$\{\}$ denotes the convex hull of the set of vertices. These polytopes are shown in Figure 2.1.

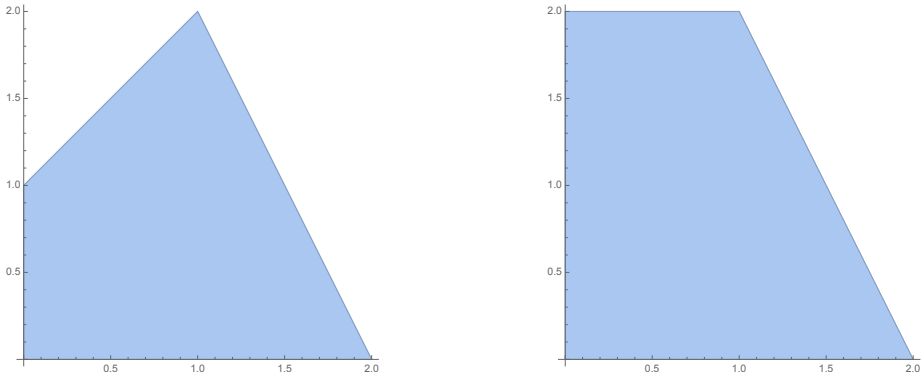


Figure 2.1: The Newton polytopes of f_1 (left) and f_2 (right) from Example 2.2.2.

Given s convex polytopes in $\mathbb{R}^n$, $K_1, \ldots, K_s$, we consider the standard $n-$dimensional Euclidean volume of a linear combination of these polytopes $v(\lambda_1, \ldots, \lambda_s) = \text{Vol}_n(\sum_{i=1}^{s} \lambda_i K_i)$ where the sum here refers to the Minkowski sum. The polynomial $v(\lambda_1, \ldots, \lambda_s)$ is homogeneous of degree n in $\lambda_1, \ldots, \lambda_s$.

Definition 2.2.3. Given s convex polytopes $K_1, \ldots, K_s$ in $\mathbb{R}^n$, the *mixed volume* of $K_1, \ldots, K_s$ is the coefficient in front of the $\lambda_1 \lambda_2 \cdots \lambda_s$ term in $v(\lambda_1, \ldots, \lambda_s)$. It is denoted $\text{MVol}(K_1, \ldots, K_s)$.

Lemma 2.2.4. [23] The mixed volume of $K_1, \ldots, K_s$ can be expressed using the inclusion exclusion formula

$$\text{MVol}(K_1, \ldots, K_s) = \sum_{J \subseteq [n]} (-1)^{n-|J|} \cdot \text{Vol}_n(K_J)$$

where $K_J = K_{i_1} + \ldots + K_{i_{|J|}}$ and $J = (i_1, \ldots, i_{|J|})$.

Example 2.2.5. Consider the polytopes $\text{Newt}(f_1)$ and $\text{Newt}(f_2)$ from Example 2.2.2. Using Lemma 2.2.4 we see

$$\text{MVol}(\text{Newt}(f_1), \text{Newt}(f_2)) = \text{Vol}_2(\text{Newt}(f_1) + \text{Newt}(f_2)) - \text{Vol}_2(\text{Newt}(f_1)) - \text{Vol}_2(\text{Newt}(f_2)).$$

Since in this case $n = 2$, the volume is just the area. This gives $\text{MVol}(\text{Newt}(f_1), \text{Newt}(f_2)) = 11.5 - 2.5 - 3 = 6$. We also observe that there are exactly 6 solutions to the system of equations defined in Example 2.2.2. This can be formally explained.

Theorem 2.2.6 (BKK Bound [20, 22, 21]). *The number of isolated* $\mathbb{C}^*$ *solutions to* $F(x) = 0$ *is less than or equal to* $\mathrm{MVol}(\mathrm{Newt}(p_1), \ldots, \mathrm{Newt}(p_m))$. *Moreover, for generic*[2] *choice of coefficients, the number of* $\mathbb{C}^*$ *solutions is exactly* $\mathrm{MVol}(\mathrm{Newt}(p_1), \ldots, \mathrm{Newt}(p_m))$.

Theorem 2.2.6 gives an upper bound on the number of isolated $\mathbb{C}^*$ solutions for any polynomial system. Moreover, in his paper [20], Bernstein gives conditions under which the coefficients of p_i, $i \in [m]$ are generic and this mixed volume is a strict upper bound. To do this we need a bit more notation.

Given a nonzero vector $w \in \mathbb{Z}^m$ and a polytope $P \subseteq \mathbb{R}^m$, we denote P_w as the *face exposed* by w and $\mathrm{val}_w(P)$ the *value* w takes on this face. Specifically:

$$P_w = \{x \in P : \langle w, x\rangle \leq \langle w, y\rangle \text{ for all } y \in P\} \quad \text{and} \quad \mathrm{val}_w(P) = \min_{x \in P} \langle w, x\rangle,$$

with $\langle (w_1, \ldots, w_m), (x_1, \ldots, x_m)\rangle := w_1x_1 + \cdots w_mx_m$. If $p = \sum_{\alpha \in \mathrm{Newt}(f)} c_\alpha x^\alpha$, we call

$$\mathrm{init}_w(p) = \sum_{\alpha \in (\mathrm{Newt}(p))_w} c_\alpha x^\alpha$$

the *initial polynomial* of p. For convenience, let $\mathrm{val}_w(p) = \mathrm{val}_w(\mathrm{Newt}(p))$. Given a polynomial system $F = \langle p_1, \ldots, p_m\rangle$ we define $\mathrm{init}_w(p_1, \ldots, p_m) := \langle \mathrm{init}_w(p_1), \ldots, \mathrm{init}_w(p_m)\rangle$.

Theorem 2.2.7. *[20, Bernstein's Other Theorem] If for all* $0 \neq w \in \mathbb{Z}^m$ *the initial systems* $\mathrm{init}_w(p_1, ..., p_m)$ *have no solutions in* $(\mathbb{C}^*)^m$*, then the number of* $\mathbb{C}^*$ *solutions to* $F(x) = 0$ *is* $\mathrm{MVol}\big(\mathrm{Newt}(p_1), \ldots, \mathrm{Newt}(p_m)\big)$.

For sparse polynomial systems the mixed volume can be much smaller than the Bezout bound. Huber and Sturmfels proposed the first polyhedral homotopy algorithm that achieves this bound by deforming the start system to a system with number of solutions equal to the mixed volume of the original system [24]. The main disadvantage to polyhedral homotopy methods is that the start systems may not be as easy to solve as in the total degree case. There is still the potential for wasted computation here as the mixed volume of a system might not be a tight upper bound on the number of solutions.

[2] Generic in this context means that the coefficients lie in a Zariski open set. In relation to the standard Euclidean topology, this means that if the coefficients of $p_1, \ldots, p_m$ are drawn from the same continuous distribution independently at random, then with probability one the BKK bound is exact.

2.2.2 Parameter homotopies

Many times we consider a family of polynomial systems that are parameterized by coefficients that are allowed to vary. Here, we often would like to solve this polynomial system for multiple choices of the parameters. In this case we consider a type of homotopy known as a *parameter homotopy* where the start system $G(x)$ is given as an instance in this parameterized family where all solutions are known. We describe this more formally below but give [17] as a more detailed reference.

Consider a system of parametric polynomial equations

$$F(x, \hat{b}) = \{f_1(x, \hat{b}), \ldots, f_m(x, \hat{b})\} = 0 \tag{2.2.3}$$

where $\hat{b} \in \mathbb{C}^N$ are the parameters and $x \in \mathbb{C}^m$ are the variables. Suppose you have have a solution set $S_{\hat{b}}$ to (2.2.3). We then construct a homotopy taking all solutions $S_{\hat{b}}$ to our target parameters $b \in \mathbb{R}^N$

$$H(x; t) = F(x, \frac{\gamma_1(1-t)\hat{b} + \gamma_2 tb}{t\gamma_2 + (1-t)\gamma_1})$$

where t runs from 0 to 1. Again, we choose random $\gamma_1, \gamma_2 \in \mathbb{C}$ to avoid singularities. This returns solutions S_b to $F(x, b)$. This is an efficient homotopy method in the sense that for every solution in S_b we track exactly one path from $S_{\hat{b}}$.

Homotopy continuation algorithms have been used in a variety of applications from game theory, kinematics, and computer vision to polynomial optimization and many others [25, 26, 27, 28] and have many off the shelf software choices [29, 30, 31]. A downside of these algorithms is that they don't exploit any additional structure often found in polynomial systems arising in applications such as symmetry or decomposability. For this, we consider a continuation based method known as *monodromy*.

2.2.3 Monodromy

We briefly outline the idea of monodromy below using the same notation as [32], but give [32, 33, 34], as more complete references. Let F_b be a parameterized polynomial system in m

variables and call the space of all such polynomial systems B. Assume the solution set of F_b is zero dimensional. Let $\mathcal{V}$ denote the *solution variety* of F_b, meaning

$$\mathcal{V} = \{(x, F_b) \in \mathbb{C}^m \times B : F_b(x) = 0\}.$$

Consider the projection

$$\begin{aligned} \pi : \mathcal{V} &\to B \\ (x, F_b) &\mapsto F_b \end{aligned}$$

and the fiber $\pi^{-1}(F_b) = \{x \in \mathbb{C}^m : F_b(x) = 0\}$. For almost all choices of parameters in B, $|\pi^{-1}(F_b)| = K$ is constant. Define D to be the *discriminant locus* of F_b, this is the set of measure zero in B where $|\pi^{-1}(F_b)| \neq K$. We define the *fundamental group* $\pi_1(B\backslash D)$ as a set of loops modulo homotopy equivalence that start and finish at a point $b \in B\backslash D$. Each loop permutes elements in $\pi^{-1}(F_b)$ and induces a group action called the *monodromy action*. Monodromy methods work by taking one solution $\hat{x}$ to the system of equations $F_{\hat{b}}$ and finding other elements of $\pi^{-1}(F_{\hat{b}})$ via the monodromy action.

The monodromy action is transitive if and only if the solution variety $\mathcal{V}$ is irreducible [35, Proposition 2.5]. In words, this tells us that we only have a hope of finding all solutions to our polynomial system if the solution variety $\mathcal{V}$ is irreducible. Therefore, if we want to use monodromy methods we first need to verify that the solution variety is irreducible. A final benefit of monodromy is that it allows us to solve for solutions up to symmetry.

One downside of monodromy methods is that unless the number of $\mathbb{C}^*$ solutions is known, there is no clear stopping criterion for this algorithm to terminate.

2.3 Solving the power flow equations

Applying homotopy continuation techniques to the power flow equations has been done with some success [2, 6]. One drawback is that no matter what start system used, as the size of the network increases the number of paths needed to track increases exponentially so these methods are only practical on small networks. Secondly, polyhedral methods are limited to small or medium

networks, due to the computational expense in finding a start system. Some work has recently been done to address this using toric deformations [36]. Finally, there are certain situations, namely lossless and zero power injections, where the power flow equations have a solution set that decomposes into two subvarieties and have symmetry that we would like to exploit.

To address these points, we propose using a combination of monodromy and parameter homotopy techniques. In order to do so, we first need a few preliminary results.

Lemma 2.3.1. The subvariety of the solution variety of (2.1.5) corresponding to the nontrivial solutions with zero active power injections form an irreducible variety.

Proof. By Theorem 6 of [11], the nontrivial component of (2.1.5) for tree networks is empty, so this statement is vacuously true. Consider the change of variables $x_i = \frac{2t_i}{1+t_i^2}$ and $y_i = \frac{1-t_i^2}{1+t_i^2}$. This gives a new system of equations for $k = 1, \ldots, n-1$,

$$0 = \sum_{m=0}^{n-1} b_{km}\Big(\frac{2t_k(1-t_m^2) - 2t_m(1-t_k^2)}{(1+t_k^2)(1+t_m^2)}\Big). \tag{2.3.1}$$

By Remark 2 of [32] is suffices to show that the image of the following map is dense:

$$\pi : \mathcal{V} \to \mathbb{C}^{n-1}$$
$$(F_b, t) \mapsto t$$

where F_b is the system of equations defined in (2.3.1) and $t = (t_1, \ldots, t_{n-1})$. For all $t_k \in \mathbb{C}\backslash\{\pm\mathbf{i}, 0, \pm 1\}$, $k = 1, \ldots, n-1$, this gives a linear system of $n-1$ equations in $|E|$ unknowns where the unknowns are the susceptances b_{km}. Since we do not consider trees, $|E| \geq n$. Let $b \in \mathbb{R}^{|E|}$ be the vector of susceptances. Then this linear system can be written as $Ab = 0$ where $A \in \mathbb{C}^{n-1\times|E|}$ is a weighted incidence matrix of G with the first row removed. This matrix has rank $n-1$ so long as none of the weights are zero, which occurs for all $t_k, t_m \notin \{\pm\mathbf{i}, 0, \pm 1\}, t_k \neq t_m$. Therefore, for generic $t \in \mathbb{C}^{n-1}$ we can find a nonzero solution b to $(2.3.1)$ giving that the map $(F_b, t) \mapsto t$ is dense in $\mathbb{C}^{n-1}$. □

Corollary 2.3.2. The solution variety corresponding to (2.1.5) with nonzero active power injections is irreducible.

Proof. As in Lemma 3.1 of [9] we consider the change of coordinates $x_i = \frac{2t_i}{1+t_i^2}$ and $y_i = \frac{1-t_i^2}{1+t_i^2}$. This transforms (2.1.5) into for $k = 1, \ldots, n-1$

$$P_k = \sum_{m=0}^{n-1} b_{km}\Big(\frac{2t_k(1-t_m^2) - 2t_m(1-t_k^2)}{(1+t_k^2)(1+t_m^2)}\Big). \tag{2.3.2}$$

As explained in the proof of Lemma 3.1 in [9], it suffices to show that for almost all $t = (t_1, \ldots, t_{n-1}) \in \mathbb{C}^{n-1}$ there exists a $P_1, \ldots, P_{n-1} \in \mathbb{R}$ and $b \in \mathbb{R}^{|E|}$ that is a solution to (2.3.2). This system of equations is linear in the susceptances b_{km} and active power injections P_k for $m, k = 1, \ldots, n-1$ so we can write it as $Ab = P$ where $P = (P_1, \ldots, P_{n-1})$ and $A \in \mathbb{C}^{n-1\times|E|}$ is a weighted incidence matrix of G. This matrix generically has rank $n-1$, meaning for almost all choices of $t \in \mathbb{C}^{n-1}$, $Ab = P$ has a solution. □

Lemma 2.3.3. In the case with zero active power injections, if $(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})$ is a solution to (2.1.5), so is $(x_1, \ldots, x_{n-1}, -y_1, \ldots, -y_{n-1})$

Proof. Substituting in $(x_1, \ldots, x_{n-1}, -y_1, \ldots, -y_{n-1})$ to (2.1.5) the result is immediate. □

Lemma 2.3.4. Suppose (2.1.5) has zero active power injections. Let $G = (V, E)$ be a bipartite graph with disjoint vertex sets $S, T \subset V$ that partition V where for all $e = v_m v_n \in E$, $v_m \in S$ and $v_n \in T$. Without loss of generality, say $v_1, \ldots, v_s \in S$ and $v_{s+1}, \ldots, v_{n-1} \in T$. If $(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})$ is a solution to (2.1.5) so is

1. $(x_1, \ldots, x_{n-1}, -y_1, \ldots, -y_{n-1})$
2. $(-x_1, \ldots, -x_s, x_{s+1}, \ldots, x_{n-1}, y_1, \ldots, y_s, -y_{s+1}, \ldots, -y_{n-1})$
3. $(-x_1, \ldots, -x_s, x_{s+1}, \ldots, x_{n-1}, -y_1, \ldots, -y_s, y_{s+1}, \ldots, y_{n-1})$

Proof. At a node $k \in S$ the power flow equations are

$$0 = \sum_{m=s+1}^{n-1} b_{km}(x_m y_k - x_k y_m). \tag{2.3.3}$$

At a node $l \in T$ the power flow equations are

$$0 = \sum_{m=1}^{s} b_{lm}(x_m y_l - x_l y_m). \tag{2.3.4}$$

Substituting in $(1) - (3)$ to the two expressions above, the result is clear. □

Corollary 2.3.5. Let $G = (V, E)$ be a bipartite graph with disjoint vertex sets $S, T \subset V$ that partition V where for all $e = v_m v_k \in E$, $v_m \in S$ and $v_k \in T$ or vice versa. Without loss of generality, say $v_0, \ldots, v_s \in S$ and $v_{s+1}, \ldots, v_{n-1} \in T$. Consider (2.1.5) with nonzero active power injections: $P_k \neq 0$, for $k = 1, \ldots, n$. Then if $(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})$ is a solution to (2.1.5) so is $(-x_1, \ldots, -x_s, x_{s+1}, \ldots, x_{n-1}, y_1, \ldots, y_s, -y_{s+1}, \ldots, -y_{n-1})$.

These lemmas tell us that in the case of zero active power injections or if G is bipartite, the power flow equations have symmetry in their solutions. We exploit this for significant computational speed up. In addition, since for a fixed graph G the number of $\mathbb{C}^*$ solutions is constant for a generic choice of susceptances, this monodromy step only needs to be done once. After this parameter homotopy methods can be used to track these solutions to solutions of desired susceptance values. Putting all of these results together, we have an algorithm for finding all complex, and therefore real, solutions to the power flow equations much efficiently.

We present timings in Table 2.1 that demonstrate the superiority of Algorithm 1 over standard homotopy techniques. Table 2.1 gives the average amount of time it takes to find all solutions to the power flow equations in a trial of 100. We use `HomotopyContinuation.jl` [29] for all methods and do all computations on a 2018 Macbook Pro with a 2.3 GHz Quad-Core Intel Core i5 processor. In all cases, we see that using parameter homotopy and only solving for the nontrivial solutions up to symmetry is much faster than polyhedral and total degree homotopy. For the cycle cases we also see that polyhedral homotopy outperforms total degree homotopy. In contrast, polyhedral is never better than total degree in the complete cases. This is because the number of paths tracked in polyhedral homotopy is slightly smaller than in the total degree case and the start system in the polyhedral case is more time consuming to compute.

In addition, using just the monodromy step of Algorithm 1 we are able to find all complex solutions when other methods can't. We consider the cyclic graph on 20 vertices. This system has $1,847,560$ complex solutions but ignoring the trivial solutions and up to symmetry it has $330,818$. If we tried to use total degree homotopy on this system, the Bezout bound is $274,877,906,944$ so we would have to track over 274 billion paths. In addition, polyhedral methods aren't practical as the solver could not find a start system. Using monodromy we found all $330,818$ complex

Algorithm 1 An algorithm to find all complex solutions to the power flow equations that exploits irreducibility and symmetry.

- **Input**: A graph $G = (V, E)$, a choice of susceptances $b \in \mathbb{R}^{|E|}$
- **Output**: All solutions to the power flow equations of the graph G with susceptances b
- **Preprocessing Step**:

  1. Find one solution to the power flow equations for one choice of susceptance values $\hat{b} \in \mathbb{C}^{|E|}$
     - For $k = 1, \ldots, n-1$ pick random $x_k \in \mathbb{C}$ and set $y_k = \sqrt{1 - x_k^2}$ so (x_k, y_k) satisfy $x_k^2 + y_k^2 = 1 \; \forall \; k$.
     - Substitute these choices of x_k, y_k into (2.1.5) and find one solution $\hat{b}$ to the corresponding underdetermined linear system.
  2. Use monodromy to find remaining nontrivial solutions to the power flow equations for susceptances $\hat{b}$ up to the equivalence $(x_k, y_k) \sim (x_k, -y_k) \forall k$. Call this solution set $S_{\hat{b}}$.

  Procedure:

  1. Use parameter homotopy to track $S_{\hat{b}}$ from susceptances $\hat{b} \in \mathbb{C}^{|E|}$ to desired susceptances $b \in \mathbb{R}^{|E|}$

solutions in $15,375$ seconds after tracking 792,934 loops. In this case, $71,212$ of these $330,818$ complex solutions were real. This example is the largest network to the authors' knowledge for which all solutions to the power flow equations have been found for a power system model.[3]

[3]The authors' note that in [37] all real solutions to a network on 60 vertices were found, but as noted by the authors in [37], the assumptions in that paper are not attainable by any realistic power systems model.

Table 2.1: Average time (seconds) to find all solutions to K_n and C_n using Algorithm 1.

K_n	4	5	6	7	8	9
Total Degree	0.03	0.23	0.97	9.71	46.86	279.38
Polyhedral	0.04	0.29	1.03	18.57	115.25	644.52
Parameter	0.003	0.03	0.14	0.62	4.85	29.79
C_n	4	5	6	7	8	9
Total Degree	0.03	0.14	0.70	4.75	23.10	136.73
Polyhedral	0.02	0.10	0.36	2.16	9.60	23.69
Parameter	0.001	0.01	0.01	0.08	0.13	0.67

2.4 Distributions of the number of real solutions

Using Algorithm 1 we are able to find all real solutions for many instances of the power flow equations allowing for a statistical analysis of the behavior of the real solutions.

A downside of homotopy methods is that it is possible that not all paths tracked from a start system will make it to a target system. Note that from a theoretical perspective, all paths will be tracked from the start system to the target system but numerical implementation using finite precision can cause some of the paths to fail. Some reasons for this is that an algorithm could incorrectly conclude a path is diverging to infinity or that two paths converge to the same solution. The rate of failures can be reduced using adaptive precision path tracking methods [38].

We experienced these phenomena running our simulations. If the parameter homotopy step lost solutions, we ran monodromy on the solution set to recover the remaining solutions. This was largely successful and in each topology studied, we found all solutions at least 98.6% of the time, ensuring accuracy of the computed distributions.

We would like to calculate empirical distributions of the number of real solutions to the power flow equations for different topologies by varying susceptance values. In order to do this we need to choose how to sample the susceptances. Since the susceptances are linear in the other variables, it is natural to vary the susceptances on the unit sphere. Sampling randomly from the unit sphere is

equivalent to assuming the susceptances are independently sampled as $\mathcal{N}(0,1)$ random variables [39].

By using Algorithm 1, we are able to empirically find distributions of the number of real solutions to the power flow equations much faster, allowing for a more accurate description of the distributions. Using statistical methods, we can be precise about what more accurate means.

Given a random variable X, its *cumulative distribution function* is defined as $F(x) = \mathbb{P}(X \leq x)$ for $x \in \mathbb{R}$. Given n independent and identically distributed random variables $X_1, \ldots, X_n$ with cumulative distribution function F, we define the *empirical distribution function as* $F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{X_i \leq x\}}$ where $\mathbf{1}$ is the indicator function. The cumulative distribution function $F(x)$ gives the probability that one random variable is less than x where $F_n(x)$ gives the probability that a fraction of random variables is less than x. The Dvoretzky–Kiefer–Wolfowitz inequality allows us to give confidence statements about the accuracy of empirical distributions based on the number of samples collected.

Lemma 2.4.1 (Dvoretzky–Kiefer–Wolfowitz Inequality [40])**.** With probability $1 - \alpha$,

$$F_n(x) - \epsilon \leq F(x) \leq F_n(x) + \epsilon$$

where $\epsilon = \sqrt{\frac{\ln \frac{2}{\alpha}}{2n}}$.

This provides a way to assess the accuracy of our empirical results with high probability. For all distributions we evaluate the number of real solutions of the power flow equations on at least 1.4 million samples, implying by Lemma 2.4.1 that with 99% probability, the true cumulative distribution function is within $\epsilon = 0.0005$ of what is listed.

2.4.1 Cycle networks

The distribution of the number of nontrivial real solutions for C_3 was completely solved and for C_4 was closely analyzed in [1] by using Mathematica to symbolically solve the entire system. Let $\mathcal{N}$ be the number of nontrivial real solutions to the given network. The authors proved that the

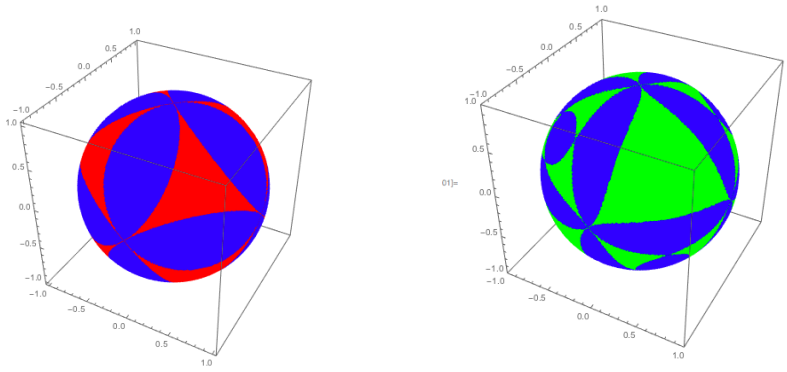


Figure 2.2: Solution Regions for C_3 (left) and C_4 with $b_{01} = 0.3$ (right)

distribution for C_3 is given by

$$\mathbb{P}(\mathcal{N} = 0) = 3 - \frac{4}{\sqrt{3}} \approx 0.6906$$
$$\mathbb{P}(\mathcal{N} = 2) = \frac{4}{\sqrt{3}} - 2 \approx 0.3904$$

and the distribution for C_4 is,

$$\mathbb{P}(\mathcal{N} = 0) \approx 0.6945, \qquad \mathbb{P}(\mathcal{N} = 4) \approx 0.3055.$$

Figure 2.2 shows the distribution for C_3 in the space of susceptances where blue regions are where there are no nontrivial real solutions and red regions are where there are two nontrivial real solutions. Figure 2.2 also shows an example for C_4 where b_{01} is fixed. In this figure, the green regions represent where there are four nontrivial real solutions.

We can also visualize solution regions for C_5 after fixing two of the susceptances. Examples of this are given in Figure 2.3. In these images we observe a lot of symmetry; this can be explained as the number of real solutions to cyclic networks is unchanged under any permutation of the edges, so also of the susceptances. The color scheme for solution regions of all pictures is given in Table 2.2.

Numerical results for $C_3 - C_{10}$ are shown graphically in Figure 2.4. We graph the distributions for C_n, C_{n+1} next to each other for $n \in \{3, 5, 7, 9\}$ since the support for C_{n+1} is that of C_n scaled by two. For all cycle graphs we notice a major left skew in the distribution. In addition, we notice

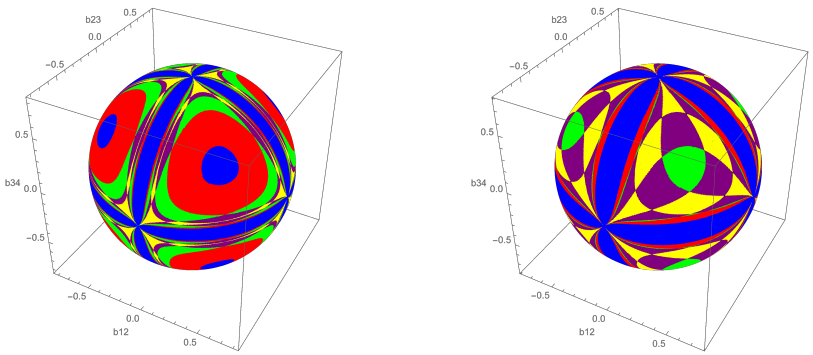


Figure 2.3: Solution regions for C_5 with $b_{01} = 0.5$, $b_{04} = 0.3$ (left) and $b_{01} = 0.6$, $b_{04} = 0.2$ (right)

Table 2.2: Colors of solution regions

# Nontrivial $\mathbb{R}$ Solutions	0	2	4	6	8	10	12	14
Color	Blue	Red	Green	Purple	Yellow	Black	Orange	Pink

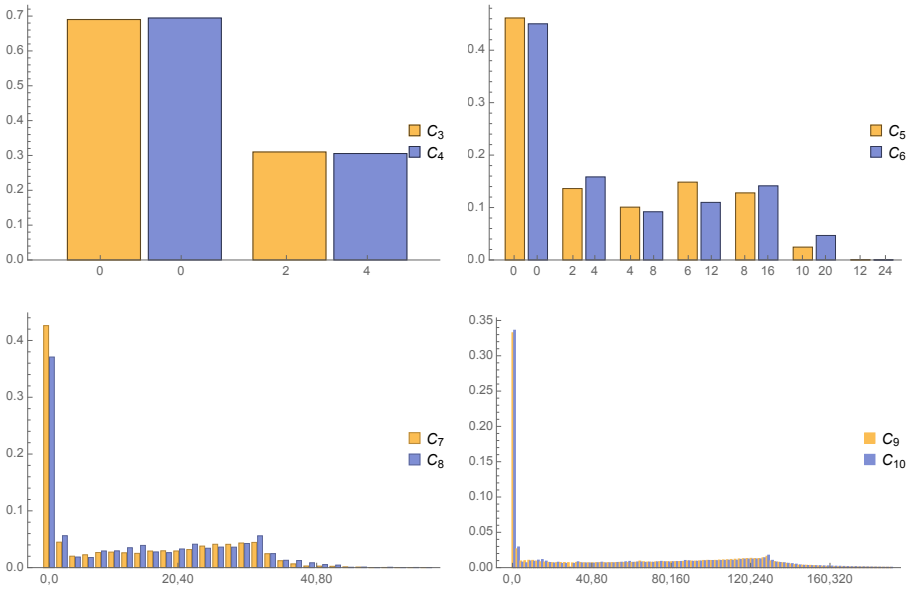


Figure 2.4: Distribution of number of nontrivial real solutions for cycle networks

that $C_5 - C_{10}$ are multimodal and C_n and C_{n+1} have similar numbers of modes, although they occur in different places.

2.4.2 Complete networks

We perform a similar analysis to Section 2.4.1 but this time on complete networks K_n where $n \in \{4, 5, 6, 7, 8\}$. The results of these simulations are shown in Figure 2.5.

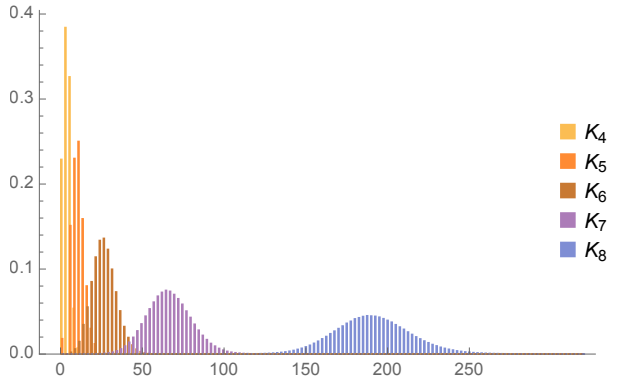


Figure 2.5: Distribution of the number of nontrivial real solutions for K_4, K_5, K_6, K_7, K_8.

In contrast to the cycle networks we see the distributions for the complete networks tending to more of a normal shape. While they are still left skewed compared to the range given by the complex bound, there isn't as large of a number of instances with zero nontrivial real solutions. We also see that as n increases the variance becomes much larger.

We also study the solution regions for K_4 in Figure 2.6. In each case there appear to be almost convex, quasi polygonal areas. In contrast to the cyclic cases we don't observe any symmetry. This is explained as the only automorphism of K_4 that fix edges e_{01}, e_{02} and e_{03} is the identity mapping.

2.4.3 Number of Real Solutions to Random Polynomials

Much of this work was motivated by the observation that the power flow equations generally admit few real solutions compared to the complex bounds. This has been well documented in existing power systems literature [2, 5]. While we agree that the number of real solutions tends to be low when compared with the total number of complex solutions, we observe that when compared with a random polynomial system, the power flow equations actually admit more real solutions than should be expected! In [41] it is shown that finding the number of real solutions to a system of polynomial equations is equivalent to finding the number of real roots of a univariate polynomial

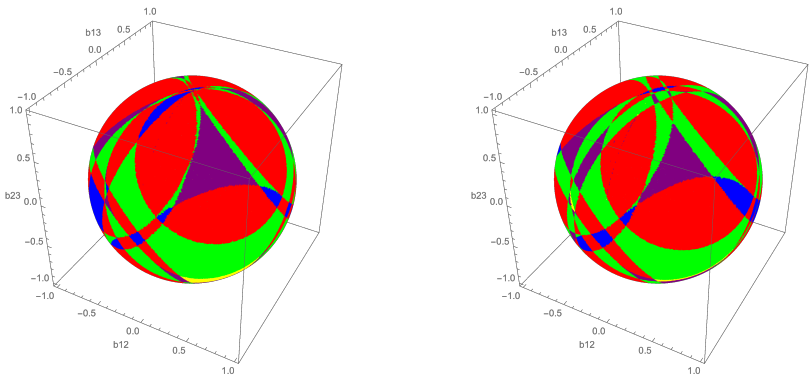


Figure 2.6: Solution Regions for K_4 with $b_{01} = 0.03$, $b_{02} = 0.15$, $b_{03} = 0.2$ (right) and $b_{01} = 0.1$, $b_{02} = 0.2, b_{03} = 0.3$ (left)

whose coefficients are polynomials in the coefficients of the original polynomial system. Specifically, the distribution of the number of nontrivial real solutions to the power flow equations is equivalent to that of the polynomial $q(x) = \sum_{i=0}^{N} c_i x^i$ where N is the number of nontrivial complex solutions to the power flow equations and each c_i is a polynomial in the susceptances. The variable x is defined as a random linear combination of original variables x_i, y_i, $i = 1, \ldots, n-1$.

Since we can reduce the distribution of the number of real solutions to the power flow equations to a single univariate polynomial, a natural question then arises.

Question 2.4.2. How does the distribution of the number of nontrivial real solutions to the power flow equations with N complex solutions compare to that of a corresponding random univariate polynomial of degree N?

We compare the distribution of the number of real solutions to the power flow equations with N nontrivial complex solutions to that of a random polynomial $q(x) = \sum_{i=0}^{N} c_i x^i$ where $c_i \sim \mathcal{N}(0, 1)$.

Remark 2.4.3. This is the set up Kac considered in [42]. Kac showed that the expected number of real roots of $q(x)$ scales logarithmically in N. Other work has considered the expected number of real roots under a different distribution on the coefficients. Using this distribution it was shown that the expected number of real roots is $\sqrt{N}$ [43]. Since it is natural to sample the susceptances uniformly at random from a unit sphere and therefore assume they are independently drawn from

a $\mathcal{N}(0,1)$ distribution, the more natural comparison in this case is when the coefficients of the $q(x)$ are also drawn iid $\mathcal{N}(0,1)$. In either case, we find the power flow equations admit more real solutions than these results would predict.

2.4.3.1 Cycle networks

We compare the distribution of the number of real solutions to cyclic $3-10$ node networks to that of corresponding random polynomials. Figure 2.7 plots the distributions of the number of nontrivial real solutions for cycle networks against that of real roots corresponding to random polynomials of appropriate degree. We see that while the cycle networks seem to give many more instances of zero nontrivial real solutions than random polynomials give of zero real roots, there also seems to be a much higher chance of getting instances of larger numbers of nontrivial real solutions with cycle networks than with that of a random polynomial. This phenomenon is reflected in Table 2.3 as we see the expected number of nontrivial real solutions is higher than that of random polynomials for $C_5, \ldots, C_{10}$. We suspect that this is true for all C_n, $n \geq 5$, and the gap between the two values will continue to increase.

Table 2.3: Expected Number of Nontrivial Real Solutions to cycle Networks

C_3	0.62	C_4	1.22	C_5	2.85	C_6	5.93
$q(x)$	1.30	$q(x)$	1.64	$q(x)$	2.35	$q(x)$	2.77
C_7	11.57	C_8	25.57	C_9	52.38	C_{10}	105.40
$q(x)$	2.96	$q(x)$	3.83	$q(x)$	4.40	$q(x)$	4.84

2.4.3.2 Complete networks

We do a similar analysis here as in the cyclic case. Here we compare the number of nontrivial real solutions to power flow equations of complete $4-8$ node networks to that of even polynomials of degree $12, 54, 220, 860$ and 3304, the generic number of nontrivial complex solutions for each network respectively. Distribution results are shown graphically in Figure 2.8 and expected values

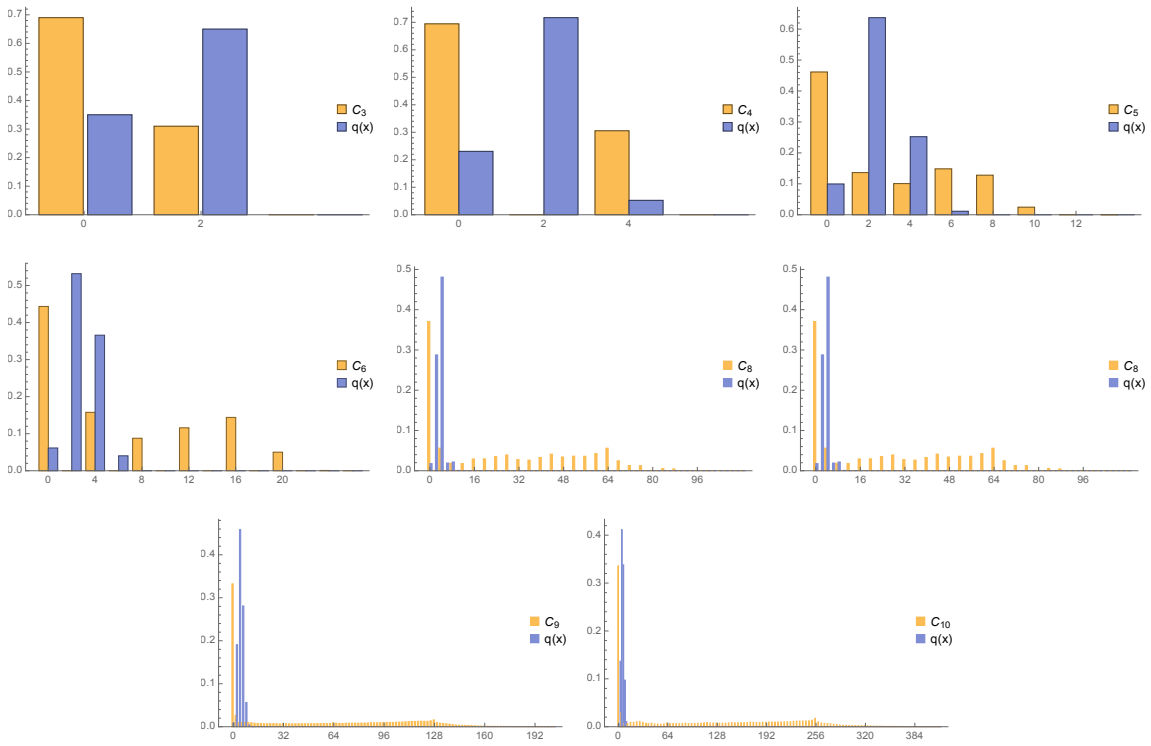


Figure 2.7: Distributions of the number of nontrivial real solutions to cycle networks and the number of real roots of a random polynomial

are given in Table 2.4. We see in Figure 2.8 that as the number of vertices grows, the distribution of the number of nontrivial real solutions to the power flow equations shifts further right than for random polynomials. This is reflected in the expected values as the expected number of nontrivial real solutions for $K_5 - K_8$ is much higher than that for a random polynomial. For K_8 we see that the expected number of nontrivial real solutions is over 28 times as high as that for a random polynomial of degree 3304.

In both cases, our computations show that the power flow equations have many more real valued solutions then random polynomial systems with comparable complexity. The main difference between the power flow equations and such a random system is that some of the coefficients in the power flow equations are equal. Specifically, the susceptance b_{ij} shows up four times in the power flow equations, in power flow equations i and j in front of monomials $x_i y_j$ and $x_j y_i$.

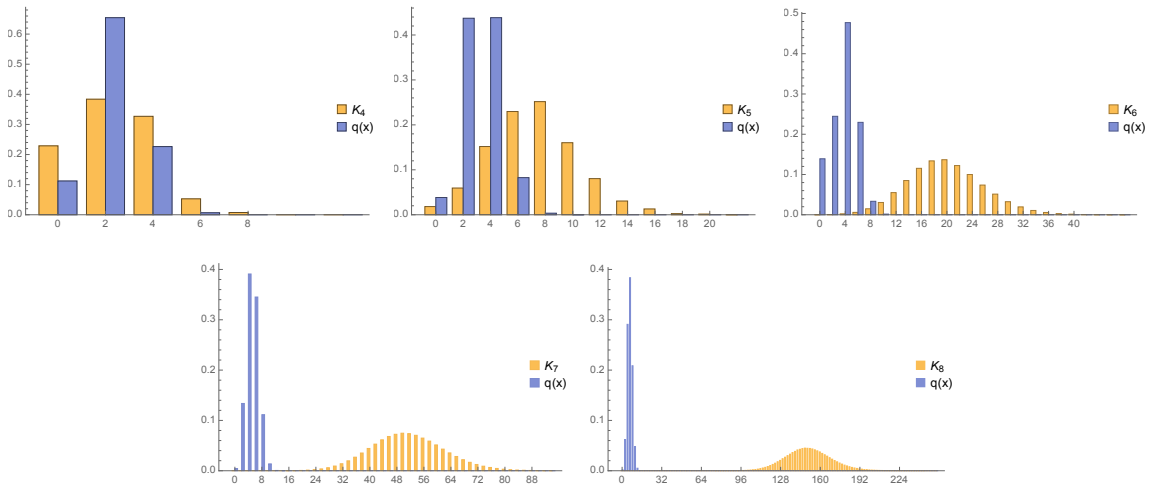


Figure 2.8: Distributions of the number of nontrivial real solutions to complete networks and the number of real roots of a random polynomial

Table 2.4: Expected Number of Nontrivial Real Solutions to Complete Networks

K_3	0.62	K_4	2.45	K_5	7.41	K_6	20.11	K_7	51.54	K_8	150.65
$q(x)$	1.30	$q(x)$	2.26	$q(x)$	3.18	$q(x)$	4.06	$q(x)$	4.93	$q(x)$	5.30

2.4.4 Solutions when all susceptances are equal

We now consider when the susceptances are assumed to be the same or approximately the same. Again we differentiate between both cyclic and complete networks and find that in both cases, when the susceptances are the same, we find many real solutions.

Theorem 2.4.4. *The cyclic graph on n vertices, C_n for $n \geq 1$, has susceptance values that achieve the generic maximum bound of $n\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$ real solutions.*

Proof. First consider the case of C_n where $4 \nmid n$. Set all susceptances equal to 1. The system of equations defined in (2.1.1) with $|V_k| = 1, g_{km} = 0$ becomes

$$\sin(\theta_k - \theta_{k-1}) = \sin(\theta_{k+1} - \theta_k) \tag{2.4.1}$$

for $k = 1, \ldots, n$. Consider the change of variables $u_k = \frac{\theta_k - \theta_{k-1}}{\pi}$ for $k = 1, \ldots, n$. This transforms (2.4.1) into

$$\sin(\pi u_k) = \sin(\pi u_{k+1}) \tag{2.4.2}$$

$$\sum_{k=1}^{n} u_k \equiv 0 \mod 2 \tag{2.4.3}$$

for $k = 1, \ldots, n$. This means that for all k, m, $u_k = u_m$ or $u_k = 1 - u_m$. This allows us to partition the set $S = \{u_1, \ldots, u_n\}$ into two sets: $S_1 = \{u_k \in S : u_k = 1 - u\}$ and $S_2 = \{u_k \in S : u_k = u\}$ for some $u \in \mathbb{R}$. Let $|S_1| = m$ and $|S_2| = n - m$.

By (2.4.3) we have $(n - 2m)u + m \equiv 0 \mod 2$. Suppose $n - m$ is odd. This gives $(n - 2m)u \equiv 1 \mod 2$. There are $n - 2m$ different u that satisfy this, namely $u = \frac{s}{n-2m}$ for $s \in \{1, 3, 5, \ldots, 2(n - 2m) - 1\}$. Now suppose $n - m$ is even. By (2.4.3) we have $(n - 2m)u \equiv 0 \mod 2$. There are $n - 2m$ solutions to this equation, namely $u = \frac{s}{n-2m}$ where $s \in \{0, 2, 4, \ldots, 2(n-2m-1)\}$. In either case, there are $\binom{n}{m}$ ways to construct S_1, giving $\binom{n}{m}(n-2m)$ such solutions u for each $m \leq k$. This gives $\sum_{m=0}^{k} \binom{n}{m}(n - 2m) = (k + 1)\binom{n}{k+1} = n\binom{n-1}{k}$ real solutions where the first equality is (5.18) of [44] and the second equality is (1.2) of [45].

Now consider C_n for $n = 4k$ for $k \in \mathbb{N}$. As per the proof of Lemma 2.4.6, the previous choice of susceptances produces infinitely many solutions. So instead, consider susceptances $b_{01} = -1$ and $b_{ij} = 1$ for all other edges ij. Using the same notation as above, we have that $u_1 = -u$ or $u_1 = 1+u$ and $u_k = u$ or $u_k = 1-u$ for all $2 \leq k \leq n$. For all $k \geq 2$, let $S_1 = \{u_k \in S \backslash u_1 : u_k = 1 - u\}$ and $S_2 = \{u_k \in S \backslash u_1 : u_k = u\}$. Note that for $u_1 = -u$, $|S_1| = m$, and $|S_2| = n - m - 1$ (2.4.3) gives $(2+2m-n)u \equiv m \mod 2$. Similarly, for $u_1 = 1+u$, $|S_1| = n-m-1$ and $|S_2| = m$ (2.4.3) also gives $(2 + 2m - n)u \equiv m \mod 2$ so the solutions to the two cases are redundant. Therefore, without loss of generality we suppose $u_1 = -u$, $|S_1| = m$ and $|S_2| = n-m-1$. When m is odd (2.4.3) gives $(2 + 2m - n)u \equiv 1 \mod 2$. This equation has $|2 + 2m - n|$ solutions $\mod 2$, namely $u = \frac{s}{2+2m-n}$ for $s \in \{1, 3, 5, \ldots, 2(2 + 2m - n) - 1\}$. When m is even we want to find all solutions to $(2 + 2m - n)u \equiv 0 \mod 2$. Again, this equation has $|2 + 2m - n|$ solutions $\mod 2$, namely $u = \frac{s}{2+2m-n}$ for $s \in \{0, 2, 4, \ldots, 2(2m - n + 1)\}$. For each $m \leq n - 1$ there are

$\binom{n-1}{m}$ ways to construct S_1. This gives a total of $\sum_{m=0}^{n-1}\binom{n-1}{m}|2+2m-n| = n\binom{n-1}{k-1}$ real solutions where the equality is proven below in Lemma 2.4.5. □

Lemma 2.4.5. $\sum_{m=0}^{2k-1}\binom{2k-1}{m}|2+2m-2k| = 2k\binom{2k-1}{k-1}$.

Proof. We see that $\sum_{m=0}^{2k-1}\binom{2k-1}{m}|2+2m-2k|$ is equal to

$$\sum_{m=0}^{k-1}\binom{2k-1}{m}(2k-2m-2) \tag{2.4.4}$$

$$+\sum_{m=k-1}^{2k-1}\binom{2k-1}{m}(2+2m-2k) \tag{2.4.5}$$

Applying (5.18) of [44] to (2.4.4), (2.4.4) is equal to

$$k\binom{2k-1}{k} - \sum_{m=0}^{k-1}\binom{2k-1}{m} = k\binom{2k-1}{k-1} - \sum_{m=0}^{k-1}\binom{2k-1}{m} \tag{2.4.6}$$

Again, using (5.18) of [44] we get the identity

$$\sum_{m=0}^{2k-1}\binom{2k-1}{m}(2k-1-2m) = 2k\binom{2k-1}{2k} = 0 \tag{2.4.7}$$

This gives the identity for (2.4.5) as

$$\sum_{m=0}^{2k-1}\binom{2k-1}{m} - \sum_{m=0}^{k-1}\binom{2k-1}{m}(2+2m-2k) \tag{2.4.8}$$

Adding (2.4.6) and (2.4.8) we see

$$\sum_{m=0}^{2k-1}\binom{2k-1}{m}|2+2m-n| = k\binom{2k-1}{k-1} - \sum_{m=0}^{k-1}\binom{2k-1}{m} \tag{2.4.9}$$

$$+\sum_{m=0}^{2k-1}\binom{2k-1}{m} - \sum_{m=0}^{k-1}\binom{2k-1}{m}(2+2m-2k) \tag{2.4.10}$$

Applying (2.4.6) to the last term in (2.4.10) gives

$$\begin{aligned}\sum_{m=0}^{2k-1}\binom{2k-1}{m}|2+2m-n| &= 2k\binom{2k-1}{k-1} - 2\sum_{m=0}^{k-1}\binom{2k-1}{m} + \sum_{m=0}^{2k-1}\binom{2k-1}{m}\\ &= 2k\binom{2k-1}{k-1} - \sum_{m=0}^{k-1}\binom{2k-1}{m} + \sum_{m=k}^{2k-1}\binom{2k-1}{m}\\ &= 2k\binom{2k-1}{k-1}\end{aligned}$$

□

In the final case of Theorem 2.4.4 we switch from the susceptances being equal. The following explains why.

Lemma 2.4.6. There exist susceptance values for C_n, $4 \mid n$, where there are infinitely many real solutions.

Proof. Set all susceptances equal to 1. The system of equations as defined in $(2.1.1)$ with $|V_k| = 1, g_{km} = 0$ becomes $\sin(\theta_k - \theta_{k-1}) = \sin(\theta_{k+1} - \theta_k)$, for $k = 1, \ldots, n$, where the indices wrap around $\mod n$. Set $u_k = \frac{\theta_k - \theta_{k-1}}{\pi}$ for $k = 1 \ldots, n$. Under these coordinates we know that $\sum_{k=1}^{n} u_k \equiv 0 \mod 2$ and $u_k = u_l$ or $u_k = 1 - u_l$ for all k, l. Now partition the set $\{u_1, \ldots, u_n\}$ into two equal size sets S_1 and S_2. For all $u_k \in S_1$ set $u_k = u$ for some $u \in \mathbb{R}$. For all $u_k \in S_2$ set $u_k = 1 - u$. This implies that $\sum_{k=1}^{n} u_k = \frac{n}{2}u + \frac{n}{2}(1 - u) = \frac{n}{2} \equiv 0 \mod 2$ satisfying the first condition. Since we can choose any $u \in \mathbb{R}$, this implies that there are infinitely many real solutions. □

By [46] we know that generically there are finitely many solutions to the power flow equations implying that such susceptances where C_n admits infinitely many real solutions lie on an algebraic set of codimension at least two. We extend the results of Lemma 2.4.6 to complete networks.

Lemma 2.4.7. There are susceptance values for K_n with $n \geq 4$ even that admit infinitely many real solutions.

Proof. Set all susceptances $b_{km} = 1$. For n even it can be verified that there exists a family of solutions of the form

$$\{y_1 = 0, y_{k+1} = -y_k, x_1 = -1, x_{k+1} = -x_k\}$$

for all even $k \geq 2$. Taking $x_m, y_m \in (0, 1)$ where $x_m^2 + y_m^2 = 1$ for all odd $m \geq 1$ gives infinitely many real solutions.

For n odd it can be verified that there exists a family of solutions of the form $y_1 = 0, x_1 = -1$ and

$$\{y_2 = \frac{\sqrt{3}}{2}x_3 - \frac{1}{2}y_3, y_4 = -\frac{\sqrt{3}}{2}x_3 - \frac{1}{2}y_3, y_k = -y_{k-1},$$
$$x_2 = \frac{1}{2}x_3 + \frac{\sqrt{3}}{2}y_3, x_4 = \frac{1}{2}x_3 - \frac{\sqrt{3}}{2}y_3, x_k = -x_{k-1}\}$$

for k even and $k \geq 6$. Taking $x_m, y_m \in (0, 1)$ where $x_m^2 + y_m^2 = 1$ for odd $m \geq 3$ gives infinitely many real solutions. □

Theorem 2.4.4 shows that for cyclic networks on $n \not\equiv 0 \mod 4$ nodes when all susceptances are the same, there are the generic maximum number of real solutions. Since this instance lies off the discriminant locus, this implies that for small perturbations of the susceptances there will still be the maximum number of real solutions.

In contrast, the situations in Lemma 2.4.6 and Lemma 2.4.7 are not generic. The fact that there are infinitely many solutions here means that the Jacobian is singular at this point. This has practical importance as in a neighborhood of this point the Jacobian will be close to singular. This affects the accuracy and efficiency of numerical methods like Newton-Raphson algorithms, which are commonly used to find solutions to the power flow equations.

In addition, we find that when we perturb the susceptances in these cases to have finitely many solutions, we also find instances with many real solutions. This indicates that a good heuristic for finding instances with many real solutions is to consider when the susceptances are equal or almost equal.

2.4.5 Other families of solutions

Finally, we conclude this section by showing that under the present assumptions, tree networks only have trivial solutions.

Lemma 2.4.8. Tree networks admit only trivial solutions.

Proof. Let $T = (V, E)$ be a tree and suppose T has s vertices with degree equal to 1 and t vertices with degree greater than or equal to one. Since T is a tree, $s \geq 1$. Let $A_k = \{m :$

v_m is adjacent to $v_k\}$. This gives equations

$$\sum_{m \in A_k} b_{km} \sin(\theta_k - \theta_m) = 0 \text{ when } \deg(v_k) \geq 1 \tag{2.4.11}$$

$$b_{km} \sin(\theta_k - \theta_m) = 0 \text{ when } \deg(v_k) = 1. \tag{2.4.12}$$

If $|V| > 2$ for each vertex of degree 1, we know it must be adjacent to at least one vertex of degree greater than 1. This means we can rewrite $(2.4.11)$ as

$$\sum_{\substack{m \in A_k \\ \deg(v_m) \geq 2}} b_{km} \sin(\theta_k - \theta_m) = 0 \text{ when } \deg(v_k) \geq 1 \tag{2.4.13}$$

The equations defined in (2.4.13) are the same as those on tree $T = (V', E')$ where $V' = \{v \in V : \deg(v) > 1\}$ and $E' = \{e \in E : e \text{ is adjacent to } v_k, v_m \in V'\}$. Since T was a tree and T' is a subgraph of T, this means T' is also a tree. We can repeat this argument again on T' and so on until we are left with a system of equations where each equation only involves one term, $b_{km} \sin(\theta_k - \theta_m)$. The equation at v_1 simplifies to $b_{01} \sin(\theta_1) = 0$ so $\theta_1 = n\pi$ for some $n \in \mathbb{Z}$. This forces $y_1 = \sin(\theta_1) = 0$ and $x_1 = \cos(\theta_1) = \pm 1$. We also have that for all v_l adjacent to v_1 that $b_{1l} \sin(\theta_l - \theta_1) = 0$ so $\theta_1 - \theta_l = n\pi$ for some $n \in \mathbb{Z}$. This implies $\theta_l = n'\pi$ for $n' \in \mathbb{Z}$ giving that $y_l = 0$ and $x_l = \pm 1$. This argument repeats for all vertices adjacent to v_l and so on. Since T is connected, this covers all vertices $v \in V$. □

Corollary 2.4.9. Solution sets for tree networks are always zero dimensional.

Remark 2.4.10. Lemma 2.4.8 would follow from a result proven in [11] with the assumption that the variety is zero dimensional. The proof provided here does not rely on this assumption.

2.5 Bounding the number of complex solutions

Using monodromy methods it is theoretically possible to find all solutions to the power flow equations given just one solution but a necessary component to monodromy is a stopping criterion. One downside of monodromy is that a stopping criterion is not always immediate. If the number of $\mathbb{C}^*$ solutions is known, this gives a stopping criterion. Another stopping criterion, known as

a *trace test*, gives a way to verify the number of solutions to a polynomial system [47, 48, 33]. The downside of a trace test is that it relies on finding all solutions to an even larger polynomial system than the one under consideration. Moreover, there does not exist a certified numerical implementation of the trace test.

Recently, another approach that uses monodromy to statistically estimate the number of solutions was proposed [49]. Further, a common heuristic when the number of $\mathbb{C}^*$ solutions is not known, is to terminate the calculation after there have been 10 loops without finding any new solutions. At this point you can run a trace test to verify that there are no other solutions.

We denote the number of $\mathbb{C}^*$ solutions as K. For a fixed network, K is generically independent of the susceptance values. For complete graphs K_n and cycle graphs C_n on n vertices with real-valued power injections, [10, 11] prove that the number of C^* solutions is $\binom{2n-2}{n-1}$ and $n\binom{n-1}{\lfloor\frac{n-1}{2}\rfloor}$ respectively[4]. This provides an upper bound for K and can be used as a stopping criterion for the monodromy method.

Our first result in this section is to show that removing a leaf halves the number of solutions to the power flow equations. This is stated formally below.

Theorem 2.5.1. *Let $G = (V, E)$ be a graph on n vertices where the corresponding power flow equations have K complex solutions. Then the power flow equations of $G' = (V', E')$ where $V' = V\backslash\{v_k\}$ and $E' = E\backslash\{e_{km}\}$ have $\frac{K}{2}$ solutions if $e = v_k v_m$ where $\deg(v_k) = 1$.*

Proof. Suppose $e = \{v_k, v_m\}$ where $\deg(v_k) = 1$. The active power injection and voltage equation at node v_k is then

$$P_k = b_{km}(x_k y_m - x_m y_k), \qquad 1 = x_k^2 + y_k^2. \tag{2.5.1}$$

Observe that G' is no longer connected to node v_k and moreover, any solution to the power flow equations of G' where the active power injection at node v_m is node $P_k + P_m$, is a solution to the power flow equations of G so long as x_k is chosen so (2.5.1) is satisfied. There are two x_k, y_k that satisfy (2.5.1) for fixed x_m, y_m, giving the result. □

[4]We note that while these bounds were proven for networks with nonzero active power injections, they are still valid under our assumption of zero active power injections. This is because zeroing out the constant terms does not change the Jacobian of the system and therefore won't force the system onto the discriminant locus.

Corollary 2.5.2. Let $G = (V, E)$ be a graph on n vertices and $G' = (V', E')$ where $V' = V\backslash\{v_{n-1}\}$ and $E' = E\backslash\{e_{n-1,m}\}$ where $e = v_{n-1}v_m$ and $\deg(v_{n-1}) = 1$. If

$$(x^*, y^*) = (x_1^*, \ldots, x_{n-2}^*, y_1^*, \ldots, y_{n-2}^*)$$

is a solution to the power flow equations on graph G' with active power injections $P_1, \ldots, P_{m-1}, P_{n-1} + P_m, P_{m+1}, \ldots, P_{n-2}$ then $(x^*, y^*, x_{n-1}^*, y_{n-1}^*)$ is a solution to the power flow equations on graph G with active power injections $P_1, \ldots, P_{n-1}$ where

$$\Big\{x_{n-1}^* = \frac{P_{n-1}y_m^* - x_m^*\sqrt{b_{m,n-1}^2 - P_{n-1}^2}}{b_{m,n-1}}, \quad y_{n-1}^* = -\frac{P_{n-1}x_m^* + y_m^*\sqrt{b_{m,n-1}^2 - P_{n-1}^2}}{b_{m,n-1}}\Big\}$$
$$\Big\{x_{n-1}^* = \frac{P_{n-1}y_m^* + x_m^*\sqrt{b_{m,n-1}^2 - P_{n-1}^2}}{b_{m,n-1}}, \quad y_{n-1}^* = -\frac{P_{n-1}x_m^* - y_m^*\sqrt{b_{m,n-1}^2 - P_{n-1}^2}}{b_{m,n-1}}\Big\}.$$

Finding explicit closed forms for other graphs is an active area of research. Attaining bounds via Theorem 2.2.6 and mixed volume computations is one method for this. Unfortunately, using the form of power flow equations in (2.1.5), the mixed volume is a strict upper bound on the number of $\mathbb{C}^*$ solutions.

To remedy this, we observe that there is more than one way to make (2.1.1) algebraic. For instance, we can use the identity $\sin(\theta_k - \theta_m) = \frac{1}{2\mathbf{i}}(e^{\mathbf{i}\theta_k}e^{-\mathbf{i}\theta_m} - e^{-\mathbf{i}\theta_k}e^{\mathbf{i}\theta_m})$ then set $x_k = e^{-\mathbf{i}\theta_k}$ to write

$$P_k = \sum_{m=0}^{n-1} \frac{b_{km}}{2\mathbf{i}}\big(\frac{x_k}{x_m} - \frac{x_m}{x_k}\big) \tag{2.5.2}$$

where now $x_0 = 1$.

Let $p_k := P_k - \sum_{m=0}^{n-1} \frac{b_{km}}{2\mathbf{i}}\big(\frac{x_k}{x_m} - \frac{x_m}{x_k}\big)$. Using this formulation of the power flow equations, all complex solutions lie in the torus and we have the following conjecture.

Conjecture 2.5.3. *For generic susceptance values, b_{km}, the number of $\mathbb{C}^*$ solutions to* (2.5.2) *is equal to* $\mathrm{MVol}\big(\mathrm{Newt}(p_1), \ldots, \mathrm{Newt}(p_{n-1})\big)$.

Observe that in Conjecture 2.5.3 we make no assumptions on the genericity of the active power injections, P_k. This is because the origin is in the interior of $\mathrm{Newt}(p_k)$ for $k \in [n-1]$, so genericity of the active power injections is not necessary.

In [11] it is shown that Conjecture 2.5.3 is true for tree networks and cyclic networks under the assumption that $b_{km} \neq b_{km}$. As a corollary of Theorem 2.4.4 we can show that Conjecture 2.5.3 is true for cyclic graphs even when $b_{km} = b_{mk}$.

Corollary 2.5.4. Let C_n be the cyclic network on n nodes. For generic choice of susceptances and $n \geq 1$, the number of complex solutions to the power flow equations for C_n is $n\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$.

Proof. By Theorem 2.2.6 and [11, Theorem 15] we know that generically, there are at most $n\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$ solutions for cyclic networks. By Theorem 2.4.4 we know that bound is achievable by all real (and therefore complex) solutions. □

Chapter 3

Real intersection points of ellipsoids

Much of the work in Chapter 2 focused on studying the real solutions to the power flow equations only after finding all complex solutions. The work in this chapter wishes to circumvent having to find all complex solutions to the power flow equations by exploiting the ellipsoid structure of the power flow equations.

3.1 Tracing around the intersections of ellipsoids

As outlined in Section 2.4, typically the number of real solutions to the power flow equations is much less than the total number of complex solutions. Therefore, even though Algorithm 1 significantly decreased the amount of time it took to find all real solutions to the power flow equations, there is still a lot of wasted computation associated with finding all complex solutions then filtering out the real ones.

There have been methods proposed to only find the real-valued solutions to (2.1.5), but none exist that are able to provably find all of them [3, 4, 5]. The methods proposed in these papers rely on the fact that the power flow equations can be reformulated as high dimensional ellipsoids, so finding all real solutions to the power flow equations is equivalent to finding all real intersection points of high dimensional ellipsoids. This transforms (2.1.5) into the system

$$\begin{aligned} x^T A_1 x &= 0 \\ x^T A_2 x &= 0 \\ &\vdots \\ x^T A_{2n-2} x &= 0 \end{aligned} \tag{3.1.1}$$

where $A_i \in \text{Sym}_{2n-2}(\mathbb{R})$ is positive definite for $i = 1, \ldots, 2n-2$ and $x = (x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})$.

Previous methods [4, 5] used the fact that the intersection of $n-1$ ellipsoids in n dimensional space is a one dimensional curve, $\mathcal{H}$ where $\mathcal{H} = \cup_{i=1}^{k} \mathcal{S}_i$ and each $\mathcal{S}_i$ is homeomorphic to a circle. The authors in [5] argue that each solution to the power flow equations is connected by such a curve $\mathcal{S}_i$. This was later disproved in [3].

Instead of trying to trace around $\mathcal{H}$, the authors in [4] instead introduced a parameter, α_i, for each ellipsoid and defined $x^T A_i x = \alpha_i$. Varying each α_i, corresponds to shrinking and expanding each ellipsoid which then allows one to trace along the hypersurface defined by the intersection of $n-1$ ellipsoids.

This idea is illustrated in Figure 3.1 for two ellipsoids defined by the equations

$$2x^2 + y^2 + xy - 1 = 0, \qquad x^2 + 3y^2 + 2xy - 1 = 0.$$

Here we set $x^2 + 3y^2 + 2xy - 1 = \alpha_1$ and vary α_1 from 0 to 3 then back down to 0 to find a second real solution to these two ellipsoids from the first.

In [4] the authors explicitly construct the ellipsoids representing the power flow equations. It is clear that when $n = 2$ this method will find all real solutions but work proving this method will find all real solutions when $n > 2$ is missing in the literature. The main results of this section are to show that this method works when $n = 3$ and when all eigenvectors of $A_1, \ldots, A_n$ are the same. We first need a few preliminaries.

Lemma 3.1.1. For $A_i \in \mathbb{R}^{n \times n}$, $i \in [n]$, consider the system of equations

$$x^T A_1 x = 1, \ldots, x^T A_n x = 1, \tag{3.1.2}$$

$$x^T M A_1 M^T x = 1, \ldots, x^T M A_n M^T x = 1 \tag{3.1.3}$$

where $M \in \mathbb{R}^{n \times n}$ is an invertible matrix. Then, x^* is a solution to (3.1.2) if and only if $M^{-T} x^*$ is a solution to (3.1.3), where M^{-T} denotes the inverse of the transpose of M. In addition, the number of real solutions between (3.1.2) and (3.1.3) is preserved.

Proof. Substituting in $M^{-T} x^*$ to $x^T M A_i M^T x$ we have

$$x^{*T} M^{-1} M A_i M^T M^{-T} x^* = x^{*T} A_i x^* = 1.$$

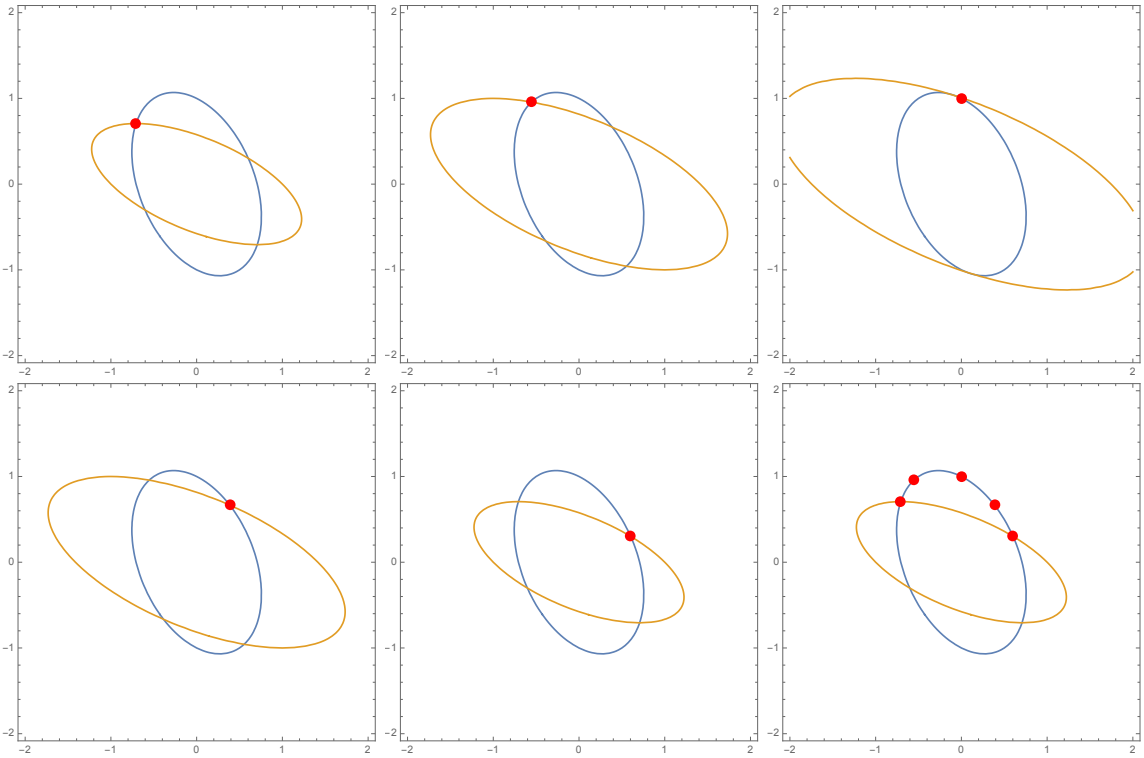


Figure 3.1: Expanding and shrinking the ellipse $x^2 + 3y^2 + 2xy - 1 = \alpha_1$ from $\alpha_1 = 0 \to 3 \to 0$ to find a second real solution to $2x^2 + y^2 + xy - 1 = 0, x^2 + 3y^2 + 2xy - 1 = 0$.

Both directions are then clear. For the second part, observe that since M^{-T} is a real matrix, if x^* is real-valued so is $M^{-T}x^*$ and vice versa. □

Lemma 3.1.1 tells us that we are allowed to simultaneously multiply our ellipses by a nonsingular matrix and its transpose and we still preserve all solutions to this system. We would like to chose such an M that simplifies our problem. The following theorem gives an indication of which such M to choose.

Theorem 3.1.2. *[50, Theorem 7.6.1] For symmetric, positive definite matrices $A, B \in \mathbb{R}^{n\times n}$, there exists a nonsingular matrix $M \in \mathbb{R}^{n\times n}$ such that $MAM^T = I$ and $MBM^T = D$ where D is a diagonal positive definite matrix.*

The proof of Theorem 3.1.2 given in [50] is constructive, meaning that for practical purposes we can easily compute such a matrix M using Cholesky and spectral decompositions.

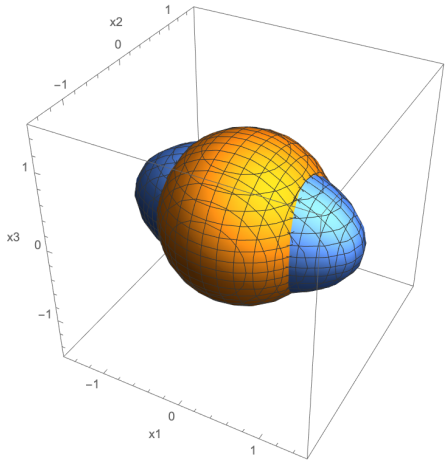


Figure 3.2: The intersection of ellipses defined by equations $x_1^2+x_2^2+x_3^2 = 1$, $\frac{1}{2}x_1^2+2x_2^2+\frac{3}{2}x_3^2 = 1$.

Theorem 3.1.3. *There exists a method to systematically shrink and expand three ellipsoids in* $\mathbb{R}^3$ *to find all real intersection points starting from one.*

Proof. Consider three ellipsoids, $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3 \subset \mathbb{R}^3$, where $\mathcal{E}_i = \{x \in \mathbb{R}^3 \ : \ x^T A_i x = 1\}$ and $A_i \succ 0$ for $i = 1, 2, 3$. By Lemma 3.1.1 and Theorem 3.1.2, we can scale each A_i by $A_i' = M A_i M^T$ such that $A_1' = I$ and $A_2' = D$ for diagonal positive definite D. Let $\mathcal{E}_1', \mathcal{E}_2'$ and $\mathcal{E}_3'$ be the corresponding ellipsoids of A_1', A_2' and A_3'. Our problem is now reduced to finding all real intersection points of $\mathcal{E}_1', \mathcal{E}_2'$ and $\mathcal{E}_3'$.

Observe that for ellipsoids $\mathcal{E}_1' = \{x \in \mathbb{R}^3 \ : \ x^T I x = 1\}$ and $\mathcal{E}_2' = \{x \in \mathbb{R}^3 \ : \ x^T D x = 1\}$, the corresponding defining equations can be simplified to

$$x_1^2 + x_2^2 + x_3^2 = 1$$
$$d_1 x_1^2 + d_2 x_2^2 + d_3 x_3^2 = 1$$

for positive d_1, d_2, d_3. If $d_i > 1$ for all $i \in [3]$ or if $d_i < 1$ for all $i \in [3]$ then $\mathcal{E}_1' \cap \mathcal{E}_2' = \emptyset$. In this case, there are no real solutions to $\mathcal{E}_1' \cap \mathcal{E}_2' \cap \mathcal{E}_3'$, and therefore no real solutions to $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$.

Now consider $\mathcal{E}_1' \cap \mathcal{E}_2' \neq \emptyset$. In this case, there are two, one-dimensional connected components of $\mathcal{E}_1' \cap \mathcal{E}_2'$, $\mathcal{S}_1$ and $\mathcal{S}_2$ where $\mathcal{S}_1 = -\mathcal{S}_2$. A picture of this is shown in Figure 3.2. Since all real intersection points of $\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3'$ must lie on $\mathcal{S}_1$ or $\mathcal{S}_2$, it is clear that by tracing around both curves one will find all real intersection points. What is not clear is that if one starts with a real intersection

point $x^* \in \mathcal{S}_1$, then there exists a path through the other one-dimensional curves defined by the intersection of $\mathcal{E}_1' \cap \mathcal{E}_3'$ or $\mathcal{E}_2' \cap \mathcal{E}_3'$. Fortunately, since $\mathcal{S}_1 = -\mathcal{S}_2$, by tracing around just one of these curves then negating all real solutions found, one will then find all real intersection points of $\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3'$.

□

Theorem 3.1.3 shows that for the intersection of three ellipsoids in $\mathbb{R}^3$, one can efficiently find all real intersection points without computing all complex ones. We conclude this section by showing that we can do the same thing if all eigenvectors of the ellipsoids are the same.

Theorem 3.1.4. *Consider the polynomial system*

$$x^T A_1 x = 1, \ \dots \ , x^T A_n x = 1$$

where $A_i \in Sym_n(\mathbb{R})$ is positive definite, $i \in [n]$ and $A_1, \dots, A_n$ have eigenvectors $\{v_1, \dots, v_n\}$. Then there exists an efficient way to find all real solutions to $x^T A_1 x = 1, \ \dots \ , x^T A_n x = 1$.

Proof. Let $V = [v_1 \ \cdots \ v_n] \in \mathbb{R}^{n \times n}$ be the matrix of eigenvectors of $A_1, \dots, A_n$. This means that $V A_i V^T = \Lambda_i$ for $i \in [n]$ where Λ_i is a diagonal matrix with the eigenvalues of A_i on the diagonal. By Lemma 3.1.1 we can scale each A_i by V and V^T giving an equivalent system of equations $x^T \Lambda_i x = 1$ for $i \in [n]$.

Observe that this system now only contains the monomials x_i^2, therefore finding all solutions to this system amounts to solving the linear system of equations

$$\begin{bmatrix} - \operatorname{Diag}(\Lambda_1) - \\ - \operatorname{Diag}(\Lambda_2) - \\ \vdots \\ - \operatorname{Diag}(\Lambda_n) - \end{bmatrix} \cdot \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_n^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.1.4}$$

Given $(x_1^2, \dots, x_n^2)$ satisfying (3.1.4), all 2^n solutions to $x^T \Lambda_1 x = 1, \dots, x^T \Lambda_n x = 1$ are then $(\pm\sqrt{x_1}, \dots, \pm\sqrt{x_n})$.

□

3.2 Approximating the convex hull of the real intersection points of ellipsoids

Aside from finding all real solutions to the intersection of ellipsoids, another problem of importance is to approximate the convex hull of these real intersection points. Specifically, given distinct ellipsoids $\mathcal{E}_1, \dots, \mathcal{E}_n \subseteq \mathbb{R}^n$ where $\mathcal{E}_i = \{x \in \mathbb{R}^n \ : \ x^T A_i x = 1\}$ for $A_i \succ 0$, we wish to approximate the polytope defined by the convex hull of the real intersection points of these ellipsoids:

$$\mathcal{C}(A_1, \dots, A_n) = \text{Conv}(\{x \in \mathbb{R}^n \ : \ x^T A_i x = 1, \ \forall \ i \in [n]\}).$$

Observe that $\mathcal{C}(A_1, \dots, A_n)$ is a polytope where each vertex is a real solution to $x^T A_i x = 1$ for $i \in [n]$. Approximating the convex hull of real varieties is an important problem in convex algebraic geometry due to the fact that the minimum of a convex function over a set is the same as the minimum over the convex hull of that set.

We are interested in minimizing a linear function over the intersection of ellipsoids. Specifically, we are interested in the polynomial optimization problem:

$$\min_{x \in \mathbb{R}^n} \ c^T x \quad \text{subject to} \quad x^T A_i x = 1, \ i \in [n], \tag{Opt-Ellipse}$$

where $c \in \mathbb{R}^n$ and $A_i \succ 0$ for $i \in [n]$. This problem is non-convex and NP hard to solve, therefore we are interested in computationally friendly approximations to (Opt-Ellipse). To do this, we first consider existing methods in solving quadratic polynomial optimization problems.

Quadratic programs (QP) are optimization problems of the form

$$\min_{x \in \mathbb{R}^n} \ x^T C x + 2c^T x \quad \text{subject to} \quad x^T A_i x + 2a_i^T x + \alpha_i \leq 0, \ i \in [m] \tag{QP}$$

where $C, A_i \in \text{Sym}_n(\mathbb{R})$ are symmetric $n \times n$ matrices, $c, a_i \in \mathbb{R}^n$ and $\alpha_i \in \mathbb{R}$ for $i \in [m]$. QPs have broad modelling power and have found applications in signal processing, combinatorial optimization, power systems engineering and more [51, 52, 53]. As mentioned above, in general these problems are NP hard to solve [54] but a convex relaxation defined by a *semidefinite program* gives an outer relaxation of (QP). This relaxation, called the *Shor relaxation* [55], lifts the

optimization variable $x \in \mathbb{R}^n$ to $\binom{n+1}{2}$-dimensional space by considering optimization variable $X \in \text{Sym}_{n+1}(\mathbb{R})$. It is defined as:

$$\min_{X \succeq 0} \langle \mathcal{C}, X \rangle \quad \text{subject to} \quad \langle \mathcal{A}_i, X \rangle \leq 0,\ i \in [m]$$
$$\langle \mathcal{A}_0, X \rangle = 0 \qquad \text{(QP-Relax)}$$

where $\langle \cdot, \cdot \rangle$ denotes the trace, $\mathcal{C} := \begin{bmatrix} 0 & c^T \\ c & C \end{bmatrix}$, $\mathcal{A}_i = \begin{bmatrix} \alpha_i & a_i^T \\ a_i & A_i \end{bmatrix}$ and $\mathcal{A}_0 = \begin{bmatrix} 1 & 0_{1\times n} \\ 0_{n\times 1} & 0_{n\times n} \end{bmatrix}$.

To solve the Shor relaxation, one must solve a semidefinite program which is a type of convex optimization problem that can be solved efficiently using interior point methods [56, Chapter 11]. If the optimal solution X^* to (QP-Relax) is rank 1 and unique, we say the relaxation is *exact*. If (QP-Relax) is not exact then its objective value still gives a lower bound on the objective value of (QP). An example of this convex relaxation is shown in Figure 3.3.

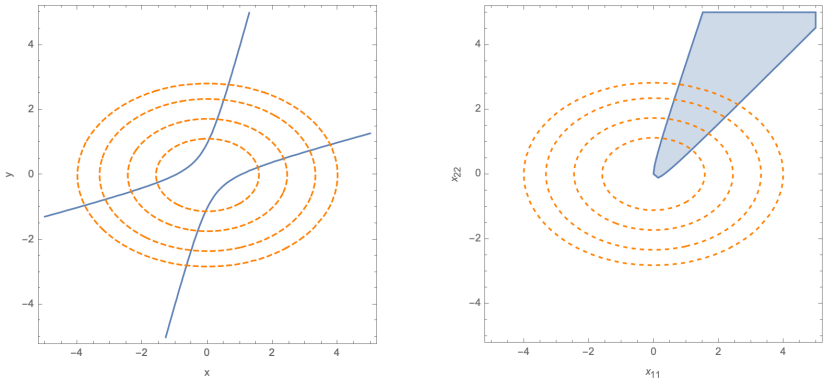


Figure 3.3: The feasible space and objective function contours to the optimization problem $\min_{x,y\in\mathbb{R}}\ x^2 + 2y^2$ subject to $3x^2 - 4xy + y^2 = 1$ (left) and the corresponding convex relaxation in the matrix variable space $X = \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix}$ projected onto the x_{11}, x_{22} coordinates (right).

A downside of the Shor relaxation is that if it is not exact, then, in general, the optimal solution attained gives no information about the optimal solution of the original problem. Specifically, since the Shor relaxation exists in a different variable space then the original problem, it is in general not even possible to get a feasible solution from it. In addition, while semidefinite programs have

polynomial time algorithms, these algorithms still can be too computationally expensive, both in terms of time and storage, for problems of medium to large size. For this reason, we consider a different relaxation of (Opt-Ellipse) that only requires linear programming.

The key observation that guides our relaxation is that ellipsoids are compact sets that can be easily bounded by the eigenvectors and values of their corresponding defining matrix. Specifically, for any ellipsoid $\mathcal{E} = \{x \in \mathbb{R}^n \ : \ x^T A x = 1\}$, let $0 < \lambda_1 \leq \cdots \leq \lambda_n$ be its eigenvalues with corresponding eigenvectors $v_1, \ldots, v_n \in \mathbb{R}^n$. Then $\mathcal{E} \subset \mathcal{P}$ where

$$\mathcal{P}(A) = \{x \in \mathbb{R}^n \ : \ \langle v_i, x\rangle \leq \frac{1}{\sqrt{\lambda_i}}, \ -\langle v_i, x\rangle \leq \frac{1}{\sqrt{\lambda_i}}, \ i \in [n]\}. \tag{3.2.1}$$

Example 3.2.1. We consider the ellipse $\mathcal{E} = \{x \in \mathbb{R}^n \ : \ x^T A x = 1\}$ where $A = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 2 \end{bmatrix}$. A has eigenpairs defined as:

$$\{v_1, \lambda_1\} = \Big\{ \begin{bmatrix} \frac{-1+\sqrt{2}}{\sqrt{4-2\sqrt{2}}} \\ \frac{1}{\sqrt{4-2\sqrt{2}}} \end{bmatrix}, \frac{1}{2}(3+\sqrt{2}) \Big\}$$

$$\{v_2, \lambda_2\} = \Big\{ \begin{bmatrix} \frac{-1-\sqrt{2}}{\sqrt{4-2\sqrt{2}}} \\ \frac{1}{\sqrt{4-2\sqrt{2}}} \end{bmatrix}, \frac{1}{2}(3-\sqrt{2}) \Big\}.$$

The ellipse $\mathcal{E}$ and its corresponding bounding polytope $\mathcal{P}(A)$ are shown in Figure 3.4.

The observation that each ellipse can be efficiently bounded by a polytope with at most $2n$ facets then suggests a natural outer linear relaxation for (Opt-Ellipse). Let $\mathcal{P}(A_1, \ldots, A_n) = \mathcal{P}(A_1) \cap \cdots \cap \mathcal{P}(A_n)$. We consider the following relaxation of (Opt-Ellipse):

$$\min \ c^T x \quad \text{subject to} \quad x \in \mathcal{P}(A_1, \ldots, A_n). \tag{Ellipse-Relax}$$

Observe that (Ellipse-Relax) is a linear program with $2n^2$ constraints, meaning that it can be solved in polynomial time. Figure 3.5 shows $\mathcal{P}(A_1, A_2)$ for the ellipses

$$\mathcal{E}_1 = \{x \in \mathbb{R}^2 \ : \ x^T \begin{bmatrix} 1 & 1/2 \\ 1/2 & 2 \end{bmatrix} x = 1\}, \quad \mathcal{E}_2 = \{x \in \mathbb{R}^2 \ : \ x^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x = 1\}.$$

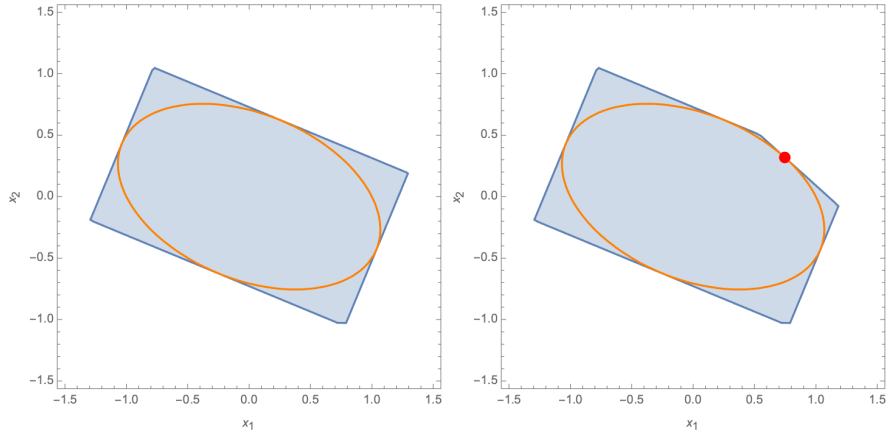


Figure 3.4: The ellipse $\mathcal{E}$ defined in Example 3.2.1 and the corresponding polytope $\mathcal{P}(A)$ (left). The ellipse $\mathcal{P}(A)$ with the constraint $\langle Ap, x - p\rangle \leq 0$ for $p = [\frac{3}{4}, \frac{1}{16}(-3 + \sqrt{65})]$ (right).

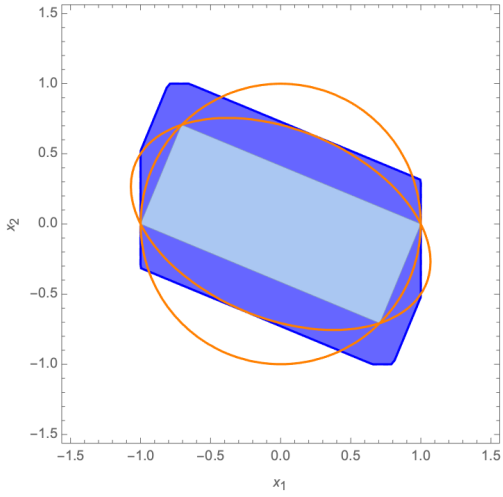


Figure 3.5: The ellipses $\mathcal{E}_1 = \{x \in \mathbb{R}^2 \ : x_1^2 + 2x_2^2 + x_1x_2 = 1\}$ and $\mathcal{E}_2 = \{x \in \mathbb{R}^2 \ : \ x_1^2 + x_2^2 = 1\}$ (orange), the convex hull of $\mathcal{E}_1 \cap \mathcal{E}_2$ (light blue) and the relaxation $\mathcal{P}(A_1, A_2)$ (dark blue).

As seen in Figure 3.5, $\mathcal{C}(A_1, \ldots, A_n) \subseteq \mathcal{P}(A_1, \ldots, A_n)$ so (Ellipse-Relax) gives an outer approximation of (Opt-Ellipse). Anytime one presents a relaxation of a difficult optimization problem, a natural question is when is it exact.

Theorem 3.2.2. *$\mathcal{P}(A_1, \ldots, A_n) = \mathcal{C}(A_1, \ldots, A_n)$ if A_i is rank one for all $i \in [n]$.*

Proof. Suppose $\text{rank}(A_i) = 1$ for $i \in [n]$. Then

$$x^T A_i x - 1 = x^T \lambda_i v_i v_i^T x = (\sqrt{\lambda_i} v_i^T x)^T \cdot (\sqrt{\lambda_i} v_i^T x) - 1 = (\langle \sqrt{\lambda_i} v_i, x \rangle - 1)(\langle \sqrt{\lambda_i} v_i, x \rangle + 1).$$

This shows that $\{x \in \mathbb{R}^n \ : \ x^T A_i x = 1\}$ is the union of two hyperplanes. Therefore,

$$\begin{aligned} \mathcal{C}(A_i) &= \text{Conv}(\{x \in \mathbb{R}^n \ : \ x^T A_i x = 1\}) \\ &= \{x \in \mathbb{R}^n \ : \ \langle \sqrt{\lambda_i} v_i, x \rangle - 1 \leq 0, \ \langle \sqrt{\lambda_i} v_i, x \rangle + 1 \geq 0\} \\ &= \mathcal{P}(A_i), \end{aligned}$$

giving that $\mathcal{C}(A_1, \ldots, A_n) = \mathcal{P}(A_1, \ldots, A_n)$. □

Since $\mathcal{C}(A_1, \ldots, A_n) \subseteq \mathcal{P}(A_1, \ldots, A_n)$ any optimal value to (Ellipse-Relax) will give a lower bound on the optimal solution to (Opt-Ellipse). Another benefit is that an optimal solution to (Ellipse-Relax) lives in the same variable space as a feasible solution to (Opt-Ellipse). With this in mind, and using Lemma 3.1.1 to assume without loss of generality that one of the ellipses is a unit sphere, we can bound how far an optimal solution to (Ellipse-Relax) is from (Opt-Ellipse).

Theorem 3.2.3. *Consider an optimal solution x^* to* (Opt-Ellipse) *where $\mathcal{E}_1$ is a unit sphere. Then an optimal solution to* (Ellipse-Relax)*, x^*_{relax}, satisfies $\|x^* - x^*_{relax}\|_2 \leq \sqrt{n + 1 + \sqrt{n}}$.*

Proof. Since $\mathcal{E}_1$ is a unit sphere, then any optimal solution to (Opt-Ellipse), x^*, must satisfy $\|x^*\|_2 = 1$. In this case, $\mathcal{P}(A_1)$ is defined as $\text{Conv}(\{-1, 1\})^n$. Since $\mathcal{P}(A_1, \ldots, A_n) \subseteq \mathcal{P}(A_1)$, we have that $\|x^*_{\text{relax}}\|_1 \leq 1$.

Now observe that the maximum distance between any two points in $\mathcal{P}(A_1)$ is $2\sqrt{n}$ which is attained for $x = (1, \ldots, 1)$ and $y = (-1, \ldots, -1)$. Since we have that $\|x^*\|_2 = 1$, we want the maximum distance from any point in $\mathcal{P}(A_1)$ to the unit sphere. This is given by projecting y onto the unit sphere. Therefore, the maximum distance between x^* and x^*_{relax} is less than or equal to the distance between $(1, \ldots, 1)$ and $-(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})$. The result then follows. □

As an immediate corollary of Theorem 3.2.3 we also have a bound on the optimal value of (Ellipse-Relax).

Corollary 3.2.4. Consider the optimal value c^* to (Opt-Ellipse) where $\mathcal{E}_1$ is a unit sphere. Then the optimal value to (Ellipse-Relax), c^*_{relax}, satisfies $|c^* - c^*_{\text{relax}}| \leq \|c\|_2 \cdot \sqrt{n+1+\sqrt{n}}$.

Proof. Observe that $c^* = c^T x^*$ and $c^*_{\text{relax}} = c^T x^*_{\text{relax}}$ and recall the Cauchy-Schwarz inequality which states for $x, y \in \mathbb{R}^n$, $|x^T y| \leq \|x\|_2 \cdot \|y\|_2$. Therefore,

$$\begin{aligned} |c^* - c^*_{\text{relax}}| &= |c^T(x^* - x^*_{\text{relax}})| \\ &\leq \|c\|_2 \cdot \|x^* - x^*_{\text{relax}}\|_2 \\ &\leq \|c\|_2 \cdot \sqrt{n+1+\sqrt{n}} \end{aligned}$$

□

While Theorem 3.2.3 and Corollary 3.2.4 gives an upper bound on the distance between optimal solutions and values of (Ellipse-Relax) and (Opt-Ellipse), for large n these bounds may still be large. For this reason, we would like to strengthen our approximation of $\mathcal{C}(A_1, \ldots, A_n)$.

Observe that for any $p \in \mathcal{E} = \{x \in \mathbb{R}^n \ : \ x^T A x = 1\}$, the hyperplane that is tangent to $\mathcal{E}$ at p adds a constraint to $\mathcal{P}(A)$ that tightens this approximation even further. This constraint can be written as $\langle Ap, x - p\rangle \leq 0$. A picture of this is shown in Figure 3.4

We see then that for any point $p \in \mathcal{E}_i$, intersecting $\mathcal{P}(A_1, \ldots, A_n)$ with the constraint $\langle Ap, x - p\rangle \leq 0$ can only tighten our approximation of $\mathcal{C}(A_1, \ldots, A_n)$. The following algorithm defines a way to systematically add these constraints.

Algorithm 2 A linear relaxation to the intersection of ellipsoids.

- **Input**: A set of ellipsoids $\mathcal{E}_1, \ldots, \mathcal{E}_n$ with defining matrices $A_1, \ldots, A_n$, a tolerance $\epsilon > 0$
- **Output**: A candidate solution to $\min \{c^T x \ : \ x \in \mathcal{C}(A_1, \ldots, A_n)\}$
- **Preprocessing Step**:
  1. Compute the polytope $\mathcal{P}(A_1, \ldots, A_n)$ using the inequalities defined in (3.2.1) and find $x_0 = \arg\min\{c^T x \ : \ x \in \mathcal{P}(A_1, \ldots, A_n)\}$
  2. Set $p_0 = c^T x_0$, $p^* = \infty$ and $\mathcal{P}^* = \mathcal{P}(A_1, \ldots, A_n)$
- **While**: $p^* - p_0 > \epsilon$
  1. Compute $x_0^{(i)} = x_0 / \sqrt{x_0 A_i^T x_0}$ for $i \in [n]$
  2. Set $\mathcal{P}^* = \mathcal{P}^* \cap \{\langle A_i x_0^{(i)}, x - x_0^{(i)} \rangle \leq 0, i \in [n]\}$
  3. Compute $x_0 = \arg\min\{c^T x \ : \ x \in \mathcal{P}^*\}$ and update $p^* = p_0$ and $p_0 = c^T x_0$
- **Return**: x_0 and p_0

We highlight Algorithm 2 on a small example.

Example 3.2.5. We consider the optimization problem

$$\min \ x_1 - 2x_2 \quad \text{subject to} \quad x_1, x_2 \in \mathcal{E}_1 \cap \mathcal{E}_2$$

where $\mathcal{E}_1 = \{x \in \mathbb{R}^2 \ : \ x_1^2 + x_1 x_2 + 2x_2^2 = 1\}$ and $\mathcal{E}_2 = \{x \in \mathbb{R}^2 \ : \ x_1^2 + x_2^2 = 1\}$. These ellipses and the corresponding polytope $\mathcal{P}(A_1, A_2)$ are shown in Figure 3.5.

We run Algorithm 2 with $\epsilon = 0.1$ and give the results on the optimal solution and objective value at each iteration in Table 3.1. We also show the geometric picture of what is happening in Algorithm 2 between iterations two and three in Figure 3.6. In this case, Algorithm 2 converges to the true optimal solution in four iterations.

Iteration Number	Optimal Solution	Objective Value
1	[-0.801, 1.00]	-2.801
2	[-0.707, 0.715]	-2.137
3	[-0.707, 0.707]	-2.121
4	[-0.707, 0.707]	-2.121

Table 3.1: Progression of Algorithm 2 from Example 3.2.5.

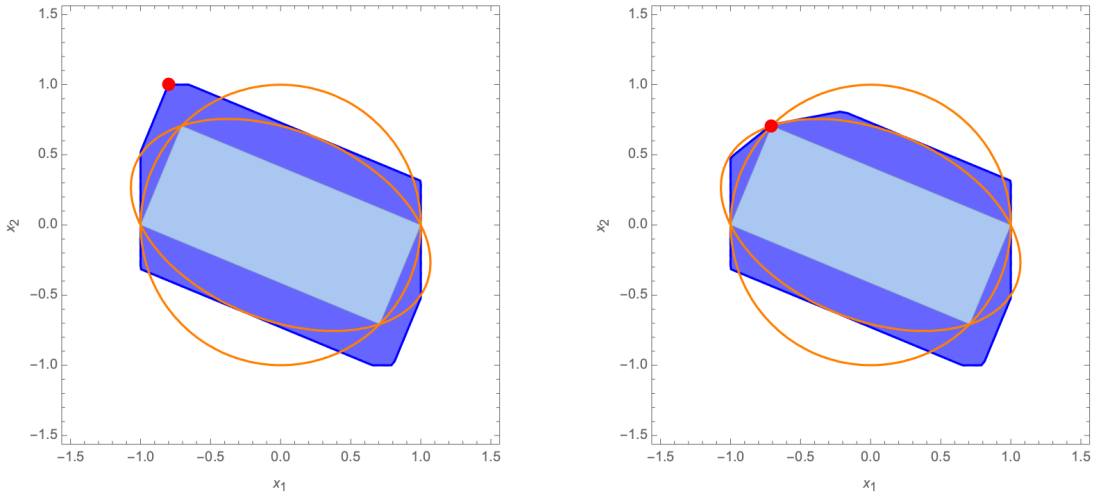


Figure 3.6: The feasible region of $\mathcal{P}^*$ (dark blue), $\mathcal{C}(A_1, A_2)$ (light blue) and the corresponding ellipses $\mathcal{E}_1, \mathcal{E}_2$ (orange) between iterations one (left) and two (right) running Algorithm 2 using the parameters in Example 3.2.5.

While in Example 3.2.5, Algorithm 2 converges to the true optimal solution of (Opt-Ellipse), in general this is not guaranteed to happen. Instead, we view Algorithm 2 as a way to quickly attain better lower bounds on the true optimal solution to (Opt-Ellipse) as well as find a potential solution relatively close to the true global optimizer.

We present the timings it took to run Algorithm 2 versus the standard Shor relaxation outlined in (QP-Relax) for (Opt-Ellipse) as the size of the problem increases. We see in Table 3.2 that Algorithm 2 is increasingly faster than (QP-Relax) as n increases.

n	10	25	50	100	150	200
Time Algorithm 2	0.001	0.02	0.34	9.21	58.63	214.00
Time (QP-Relax)	0.02	0.13	0.81	16.30	95.22	769.27

Table 3.2: The average time (seconds) it took to run Algorithm 2 with tolerance $\epsilon = 0.1 \cdot n$ versus (QP-Relax) as relaxations to (Opt-Ellipse) where each ellipse $\mathcal{E}_i \subset \mathbb{R}^n$, $i \in [n]$.

All computations were done on a 2018 Macbook Pro 2.3 GHz Quad-Core Intel Core i5 and the average is taken over 10 random problem instances. We solve the linear program in Algorithm 2 using `GLPK` [57] and we solve the semidefinite program needed for (QP-Relax) using `Mosek` [58].

The problem data is generated randomly as follows. Each entry of the objective value $c \in \mathbb{R}^n$ is chosen to be iid $\mathcal{N}(0,1)$. To generate each ellipsoid, we define a matrix B where each entry is chosen iid $\mathcal{N}(0,1)$. We then consider $\alpha \cdot B^T B$ where α is a scaling factor that ensures $x = (1, \ldots, 1) \in \mathbb{R}^n$ is a solution to $x^T \alpha B B^T x = 1$. This is done to ensure that the problem (Opt-Ellipse) is feasible.

Chapter 4

Data center geographical load shifting

The content of this chapter addresses a concrete problem. It is similar to Chapter 2 and Chapter 3 in that it considers an optimization problem on power systems but overall, it is less theoretical than the rest of this thesis. In this chapter we consider the linearized version of the power flow equations, so the complexity comes from the nested structure of the optimization model. This chapter focuses on modeling of geographic load shifting which is motivated by the following problem.

Increasing demand for computing has led to the development of large-scale, highly optimized data centers, which represent large loads in the electric power network. Many major computing and internet companies operate multiple data centers spread geographically across the world. Thus, these companies have a unique ability to shift computing load, and thus electric load, geographically. This chapter outlines a "bottom-up" load shifting model which uses data centers' geographic load flexibility to lower carbon emissions. This model utilizes information about the locational marginal carbon footprint of the electricity at individual nodes, but does not require direct collaboration with the system operator. We demonstrate how to calculate marginal carbon emissions, and assess the efficacy of our approach compared to other settings, including where the data centers bid their flexibility into a centralized market. We find that data center load shifting can achieve substantial reductions in carbon emissions even with modest load shifting.

4.1 Motivation

The recent technology revolution has led to an increase in demand for computing resources. Between 2010 and 2018 there was an estimated 550% increase globally in the number of data

center workloads and computing instances [59]. Technology companies like Amazon, Facebook, Google, Microsoft and Alibaba run networks of these highly optimized and efficient *hyper-scale* data centers dispersed geographically throughout the world [60, 61].

Hyper-scale data centers represent large loads on electric power networks. In an effort to mitigate their environmental impact, many of the companies that operate these data centers have made public pledges to reduce their carbon emissions through improved efficiency and by investing in renewable power generation [62, 63]. Google is working to become carbon free through the use of carbon-intelligent computing which shifts computing tasks to less carbon intensive hours or locations [64]. The concept of computing that adapts to the operation of the electric grid has been realized by start-ups [65], and plays an important part in the vision of zero carbon cloud computing [66, 67, 68].

Computing companies that operate networks of data centers have the ability to defer when computing tasks are processed or process them at different locations. This provides data centers with a unique tool to control and adapt their electricity use geographically. Previous research has examined the impact of integrating data centers and demand response [69, 70, 71, 72, 73, 74, 75], considered geographical load shifting to reduce electricity costs [76, 77, 78, 79] and studied optimal investment locations of data centers [80].

The electricity markets are operated by independent system operators (ISOs) to minimize generation cost without direct consideration of carbon emissions. Other work has investigated the potential benefit of cooperation between data centers and the ISO [78], and modelled data center flexibility through the use of virtual links in time and space [81]. Many of these works and others consider shifting of computing load to reduce the carbon emissions of data centers by increasing absorption of renewable energy [82, 83, 79, 84, 85]. An important aspect of these previous works is that they either assume collaboration between the ISO and the data centers (meaning that data center operators must give up control of their energy usage to the market clearing), or perform load shifting using very simplified metrics of carbon emissions associated with the electric grid (i.e., they neglect variations in CO_2 emissions across different locations which arise due to transmission

congestion). Here, we use additional data from the electric market clearing to compute the *locational marginal carbon emissions*, a metric that more accurately represents the carbon emissions associated with electricity usage at different locations in the grid.

In electricity markets the price of electricity is calculated based on locational marginal prices (LMPs), which reflect the increase in system cost for one additional unit of load. Previous work on reducing carbon emissions through load shifting has assumed that prices are directly tied to the fraction of non-renewable energy [86], or considered average carbon emissions for electricity in a region and/or renewable energy curtailment [87, 88, 85]. A benefit to these metrics is that several companies provide information about the average carbon intensity of electricity [89, 90] or total renewable energy curtailment [91], which makes the metrics easier to compute. However, these metrics fail to consider important aspects of electric grid operation, such as the impact of a marginal increase or decrease in load or transmission capacity. The use of marginal emissions as a tool to assess the impact of interventions such as energy efficiency or renewable energy has been discussed in [92, 93, 94, 95, 96]. The purpose of this chapter is to define a locational marginal carbon emission that is based off the underlying dynamics of the electric power system and to evaluate this metric against other commonly considered models.

4.2 Models for load shifting

Electricity markets are typically cleared using a DC optimal power flow (DC OPF) model, which minimizes generation cost subject to transmission and generation constraints [97, 98]. We distinguish between two different modes of interaction between the data centers and the ISO that operates the electricity market.

(1) **Data center-driven load shifting.** In this model, the data center loads are fixed values to the ISO, but can be adapted by the data centers themselves. Here the data centers solve an internal optimization problem (reflecting internal constraints on the load flexibility) to determine how to shift their electricity consumption to reduce carbon emissions.

(2) **ISO-driven load shifting.** In this model, the data center interacts with the market as a market participant with flexible demand, and provides a model of load flexibility to the ISO. The ISO then integrates the load flexibility model into the overall market clearing, and determines how to shift data center load to achieve the best solution for the whole system.

We outline both models below.

4.2.1 Data center-controlled load shifting

We first describe the data center driven load shifting model, which is adopted from [99]. This model operates in a three stages.

Step 1: The ISO solves a DC OPF. In the first step, the ISO clears the electricity market by solving the DC OPF. To formulate the DC OPF, we consider an electric power network with the set of nodes, loads, transmission lines and generators denoted $\mathcal{N}$, $\mathcal{D}$, $\mathcal{L}$ and $\mathcal{G}$ respectively. Let $\mathcal{G}_i \subset \mathcal{G}$ and $\mathcal{D}_i \subset \mathcal{D}$ be the subset of generators and loads connected to node i. Given this notation, the DC OPF is defined as:

$$\min_{\theta, P_g} \quad c^T P_g \tag{4.2.1a}$$

$$\text{s.t.} \quad \textstyle\sum_{\ell \in \mathcal{G}_i} P_{g,\ell} - \sum_{\ell \in \mathcal{D}_i} P_{d,\ell} = \sum_{j:(i,j)\in\mathcal{L}} \beta_{ij}(\theta_i - \theta_j), \qquad \forall i \in \mathcal{N} \tag{4.2.1b}$$

$$-P_{ij}^{\lim} \leq -\beta_{ij}(\theta_i - \theta_j) \leq P_{ij}^{\lim}, \qquad \forall (i,j) \in \mathcal{L} \tag{4.2.1c}$$

$$P_{g,i}^{\min} \leq P_{g,i} \leq P_{g,i}^{\max}, \qquad \forall i \in \mathcal{G} \tag{4.2.1d}$$

$$\theta_{\text{ref}} = 0. \tag{4.2.1e}$$

Here, the optimization variables are the voltage angles at each node, θ_i for $i \in \mathcal{N}$ as well as the generation dispatch $P_{g,\ell}$ for all $\ell \in \mathcal{G}$. The objective value (4.2.1a) seeks to minimize generation costs where $c \in \mathbb{R}^{|\mathcal{G}|}$ is a vector of generator costs and P_g is the vector of all generator variables $P_{g,\ell}$. The constraint (4.2.1b) ensures that nodal power balance constraints are met, where $\beta_{ij} \in \mathbb{R}$ is the susceptance value on line (i,j) and $P_{d,\ell}$ is the load demand at load $\ell \in \mathcal{D}_i$. The constraints (4.2.1d) and (4.2.1c) define transmission line and generator capacity constraints where $P_{ij}^{\lim}$ is the

transmission capacity, which we assume is the same in both directions, and $P_g^{\min}$ and $P_g^{\max}$ are the generator capacity constraints. Finally, (4.2.1e) fixes the voltage angle at the reference node to be zero.

Step 2: Data centers shift load. Independently of the ISO, data center operators shift their load to minimize carbon emissions. To estimate the impact of a load shift, the data centers utilize a shifting metric λ. There are multiple possible definitions of λ, which will be discussed in Section 4.2.2. We let $\mathcal{C}$ denote the set of all shiftable data center loads and consider optimization variables $\Delta P_{d,i}$ for all $i \in \mathcal{C}$ and s_{ij} for all $(i,j) \in \mathcal{C} \times \mathcal{C}$. The former represents the change in load at data center i and the latter represents the shift in load from data center i to j. The resulting optimization problem is given by:

$$\min_{\Delta P_d, s} \quad \sum_{i \in \mathcal{C}} \lambda_i \Delta P_{d,i} + \sum_{(i,j) \in \mathcal{C} \times \mathcal{C}} d_{ij} s_{ij} \tag{4.2.2a}$$

$$\text{s.t.} \quad \Delta P_{d,i} = \textstyle\sum_{j \in \mathcal{C}} s_{ji} - \sum_{k \in \mathcal{C}} s_{ik} \qquad \forall i \in C \tag{4.2.2b}$$

$$\sum_{i \in \mathcal{C}} \Delta P_{d,i} = 0 \tag{4.2.2c}$$

$$-\epsilon_i \cdot P_{d,i} \leq \Delta P_{d,i} \leq \epsilon_i \cdot P_{d,i} \qquad \forall i \in \mathcal{C} \tag{4.2.2d}$$

$$0 \leq s_{ij} \leq M_{ij} \qquad \forall ij \in \mathcal{C} \times \mathcal{C}. \tag{4.2.2e}$$

The objective value (4.2.2a) minimizes λ over the shift in data center load, ΔP_d, while considering a cost d_{ij} associated with shifting load from data center i to data center j. The constraint (4.2.2b) defines that the change in load at data center i is equal to the total load shifted in minus the total load shifted out, (4.2.2c) enforces the sum of all load shifts to be zero, (4.2.2d) limits the amount each data center can shift as a percentage, ϵ, of their original load and (4.2.2e) limits how much load data center i can send to data center j.

Step 3: ISO resolves DC OPF with new load pattern. Next, the ISO resolves the DC OPF (4.2.1) with new load profile, $P'_{d,i} = P_{d,i} + \Delta P^*_{d,i}$, where $\Delta P^*_{d,i}$ is the optimal solution to (4.2.2) for all $i \in \mathcal{N}$. We assume that the system is operated using this solution[1]

[1]Note that we assume that the ISO solves the OPF twice for each time step, once before and once after the load shift. In reality, OPF is only solved once for each time step, and the data center loads would likely use λ values

4.2.2 Shifting Metrics

For data centers to shift load as described in (4.2.2), the shifting metric λ needs to be specified. Below, we review four different metrics that have been proposed to guide load shifting. Note that all the metrics are defined as vectors with one entry for each node in the system.

The price of electricity λ_{LMP}. The most widely used metric for load shifting is the price of electricity, which is given by the locational marginal price (LMP) at each node in the power grid. We will refer to this metric as λ_{LMP}. The LMP represents the increase in overall system cost due to an incremental increase of 1 MW of load at the given node, and is calculated as the dual variable of the nodal balance constraints (4.2.1b) of the original DC OPF. LMPs are easy to access, as they are typically made available in real time, and have been proposed for data center load shifting in [78]. Furthermore, since renewable generators tend to be the cheaper generators, it is often assumed that shifting load to nodes with lower prices λ_{LMP} will contribute to reducing renewable energy curtailment.

Average carbon emissions λ_{average}. A common metric for the carbon content of electricity is average carbon emissions per MW of load across a region of the electric grid [89]. This metric, which we will denote by λ_{average}, has the same value for all nodes in a region $\mathcal{R}$. The kth entry of λ_{average}, corresponding to the kth node located in region $\mathcal{R}$, is defined as

$$\lambda_{\text{average}} = \frac{\sum_{i \in \mathcal{R}} g_i \cdot P_{g,i}}{\sum_{i \in \mathcal{R}} P_{g,i}}, \tag{4.2.3}$$

where g_i is the carbon intensity of generator i. The intuition behind this metric is to shift load to regions with lower average carbon footprint, and thus reducing the average carbon emissions associated with electricity consumption. This type of metric has been proposed for data center load shifting in [85]. A benefit of using λ_{average} is that these values are made publicly available by various companies [89, 90].

Excess low carbon power λ_{excess}. This metric considers shifting based on the amount of excess low carbon generation capacity available in a region, and accounts for renewable energy curtailment

calculated based on the OPF solution from the previous time step for shifting. However, if the OPF model is solved frequently enough, e.g. every 5 minutes, it is reasonable to assume that the load will remain largely constant between time periods and our model is a good estimate.

(solar PV and wind) as well as unused hydro, nuclear power and storage generation. The excess low carbon power is defined for a given region, where the value of the kth component of λ_{excess} is the same for all nodes k in a region $\mathcal{R}$. Let e_i be the excess capacity (MW) for each low carbon generator $P_{g,i}$ (with $e_i = 0$ for other generators). The kth component in λ_{excess} is given by

$$\lambda_{\text{excess}} = -\sum_{i \in \mathcal{R}} e_i. \tag{4.2.4}$$

This metric provides incentive to shift load to regions with high amounts of excess low carbon power (i.e., more negative values of λ_{excess}), which could allow for more utilization of low carbon generation. This metric has been proposed for inter-regional data center load shifting in [85].

Locational marginal carbon emissions λ_{CO_2}. We first derived this metric in [99] and propose this metric as the proper way to shift load geographically. This metric is defined as the change in carbon emission as a function of the change in load at a given node k in the network. Similar to the λ_{LMP}, it is derived by considering how the DC OPF solution would change given a change in the load at a specific location. However, instead of considering the change in the objective function (which measures overall system cost), the derivation of λ_{CO_2} uses sensitivity analysis of linear programs to identify the change in the carbon emissions.

Consider an optimal solution $x^* = [\theta^*, P_g^*] \in \mathbb{R}^n$ to the DC OPF (4.2.1). From linear optimization theory we know there exists at least one basic optimal solution with $Ax^* = b$, where $A \in \mathbb{R}^{n \times n}$ is a full rank matrix consisting of the coefficients for all the binding constraints of (4.2.1) at the optimal solution x^* and b is the right hand side. Specifically, the rows of A consist of the equality constraints (4.2.1b) and (4.2.1e) as well as a subset of the inequality constraints (4.2.1c), (4.2.1d) that are satisfied at equality for x^*.

A small change in load can be represented as a small change in the right hand side b, given by $\Delta b = \begin{bmatrix} \Delta P_d & 0 \end{bmatrix}^T$. Assuming that the change is sufficiently small to not alter the set of active constraints, we can compute the associated change in generation as $A\Delta x = \Delta b$ where $\Delta x = [\Delta\theta \ \Delta P_g]$, giving the linear relationship

$$\begin{bmatrix} \Delta\theta \\ \Delta P_g \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} \Delta P_d \\ 0 \end{bmatrix} \tag{4.2.5}$$

Denote the matrix consisting of the last $|\mathcal{G}|$ rows and first $|\mathcal{N}|$ columns of A^{-1} by B. This gives the linear relationship between load and generation changes, $\Delta P_g = B \cdot \Delta P_d$.

Let $g \in \mathbb{R}^{|\mathcal{G}|}$ be a cost vector that measures the carbon emissions of each generator per MW. The ith component of g, g_i, is the carbon intensity of generator i. Multiplying each side of $\Delta P_g = B \cdot \Delta P_d$ on the left by g gives us the following carbon sensitivity:

$$\Delta CO_2 = g \cdot \Delta P_g = g \cdot B \cdot \Delta P_d = \lambda_{\text{CO}_2} \Delta P_d. \tag{4.2.6}$$

where we define $\lambda_{\text{CO}_2} = g \cdot B$. Intuitively we think of the kth component of λ_{CO_2} as measuring how an increase of 1 MW of load at node k will affect the total carbon emissions of the system.

The benefit of λ_{CO_2} is that it captures the impact of load shifting on carbon emissions in a more direct way than λ_{LMP} and includes detailed information regarding the marginal impact of load shifting at specific data center locations compared with λ_{average} and λ_{excess}. However, since the full network information needed to calculate the λ_{CO_2} is not made publicly available, it is not easy to calculate these values in real time. In addition, since λ_{CO_2} represents a linear sensitivity, these values may only be accurate for small load shifts.

Remark 4.2.1. Marginal carbon emissions at a node depends on the objective function considered when calculating an operating point. In a situation where the ISO minimizes carbon emissions instead of generation costs by replacing the objective function (4.2.1a) with $g^T P_g$ where $g \in \mathbb{R}^{|\mathcal{G}|}$ is as defined above, the values of λ_{CO_2} could be obtained as the dual variables to the nodal power balance constraints in the same way that λ_{LMP} is obtained for the DC OPF where the ISO is minimizing cost.

4.2.3 ISO-controlled load shifting

A second model for load shifting considered in this chapter assumes that the data centers give their flexibility to the ISO, which uses this flexibility to achieve overall system objectives. We primarily consider this situation as a benchmark to the load shifting model defined in Section 4.2.1.

DC OPF with flexibility (DC-FLEX). In the DC-FLEX model, the ISO considers the following optimization problem which includes data center load shifting flexibility:

$$\min_{P_g,\theta,\Delta P_d,s} \quad c^T P_g + \textstyle\sum_{ij} d_{ij} s_{ij} \tag{4.2.7a}$$

$$\text{s.t.} \quad \textstyle\sum_{\ell\in\mathcal{G}_i} P_{g,\ell} - \sum_{\ell\in\mathcal{D}_i}(P_{d,\ell} + \Delta P_{d,\ell}) = \textstyle\sum_{j:(i,j)\in\mathcal{L}} \beta_{ij}(\theta_i - \theta_j), \quad \forall i \in \mathcal{N} \tag{4.2.7b}$$

$$\text{Constraints (4.2.1c), (4.2.1d), (4.2.1e)} \tag{4.2.7c}$$

$$\text{Constraints (4.2.2b), (4.2.2c), (4.2.2d), (4.2.2e)} \tag{4.2.7d}$$

The optimization variables $\theta_i, i \in \mathcal{N}$ and $P_{g,\ell}, \ell \in \mathcal{G}$ are as in DC OPF (4.2.1). The variables $\Delta P_{d,\ell}$ for $\ell \in \mathcal{C}$ and s_{ij} for $(i,j) \in \mathcal{C} \times \mathcal{C}$ are as defined in the data center driven shifting model (4.2.2). The objective value (4.2.7a) minimizes the cost of electricity production where $c \in \mathbb{R}^{|\mathcal{G}|}$ is a vector of generator costs with the additional cost d_{ij} of shifting load from data center i to data center j. Constraints (4.2.7b)-(4.2.7d) ensure the nodal power balance, line limits, generation capacity, reference node and data center load shifting flexibility constraints are met.

Cost of carbon emissions. In the formulation of DC OPF (4.2.1) and DC-FLEX (4.2.7), we assumed that the ISO is minimizing the overall cost of operating the electric system, without consideration of carbon emissions. This is representative of system operations today. However, as mentioned in Remark 4.2.1, it is also possible that the ISO may change their objective and include a cost on the carbon emissions. In this case, we consider objective function

$$[\alpha c^T + (1-\alpha) g^T] \cdot P_g \tag{4.2.8}$$

where α is a weighting factor that represents the emphasis on minimizing generation cost versus reducing carbon emissions.

Benchmarking. We use the DC-FLEX model as a benchmark for two reasons. First, many demand response schemes allow flexible loads to participate in markets by providing information about their flexibility to the ISO. Thus, the DC-FLEX model (4.2.7) represents a realistic model of potential future interactions between data centers and the ISO. Second, all of the load shifting metrics that are proposed to guide data center load shifting in Section 4.2.2 provide only partial information about the impact of a load shift. In comparison, DC-FLEX is able to optimize the load shift with exact knowledge of how the cost and/or carbon emissions will change as a result. Therefore, we can expect that these models will always find the most optimal load shift, i.e., the load shift that gives the lowest cost or carbon emissions. Theoretical results comparing these models were derived in [99].

4.3 Evaluation Metrics

Some evaluation metrics for data center efficiency and carbon emissions are provided in [100]. These metrics assess the flexibility and sustainability of data centers as well as the potential benefit to upgrading data center equipment. In contrast, we want to evaluate how shifting load geographically affects total system generation costs and carbon emissions.

Since the data centers represent a small subset of loads and only a relatively small percentage of the load at each data center is allowed to shift, the percentage decrease or increase is frequently only a small fraction of the total system emissions and cost. For this reason, we introduce two metrics that measure the change in generation cost and carbon emissions relative to the (small) amount of load that is shifted. The main purpose for introducing these two metrics is to normalize the change in carbon emissions and generation cost by the small amount of load being shifted.

Reduction per allowed MW: $\mu_{\%,\,\mathrm{CO_2}}$. This metric is defined as the change in carbon emission normalized by the maximum amount of load that can be shifted. The maximum amount of load that can be shifted (in MW), $\mathbb{L}$, can be calculated as $\mathbb{L} = \sum_{i \in \mathcal{C}} \epsilon_i \cdot P_{d,i}$. Let Δ be the total change in carbon emissions from the original DC OPF. We then define the relative reduction $\mu_{\%,\,\mathrm{CO_2}}$ as

$$\mu_{\%,\,\mathrm{CO_2}} = \frac{\Delta}{\mathbb{L}}.$$

The units of $\mu_{\%,\,\mathrm{CO}_2}$ are carbon tons per MW. This metric is a measure of how well a method is able to make use of the available flexibility.

Reduction per shifted MW: $\mu_{\mathbf{shift,\,CO_2}}$. This metric is similar to $\mu_{\%,\,\mathrm{CO}_2}$ but considers the change in carbon emissions when normalized by the actual load shift, as opposed to the maximum allowed load shift. Again, let Δ be the total change in carbon from the original DC OPF. Denote $\mathcal{S} = \sum_{i\in\mathcal{C}} |\Delta P_{d,i}|$ as the total amount of load shifted (in MW). We then define

$$\mu_{\mathrm{shift,\,CO}_2} = \frac{\Delta}{\mathcal{S}}. \tag{4.3.1}$$

The units of $\mu_{\mathrm{shift,\,CO}_2}$ are carbon tons per MW. This metric is a measure of the change in carbon emissions per MW shifted. Notice that if in each time step we shift the maximal amount of load as dictated by ϵ, then $\mu_{\mathrm{shift,\,CO}_2} = \mu_{\%,\,\mathrm{CO}_2}$.

The definitions of $\mu_{\%,\,\mathrm{CO_2}}$ and $\mu_{\mathrm{shift,\,CO_2}}$ can easily be adapted to assess cost reductions by defining Δ as the total change in cost relative to the original DC OPF. We denote the corresponding cost reduction metrics as $\mu_{\%,\,\$}$ and $\mu_{\mathrm{shift},\,\$}$.

Predicted vs actual impact of load shifting. For the data center-driven load shifting, all evaluation metrics mentioned above can be defined either for the predicted impact of the load shift obtained by considering the objective function of the data center problem (4.2.2) in Step 2 of our model, and for the actual load shift obtained after the ISO resolves the DC OPF (4.2.1) in Step 3. By evaluating the difference between the two, we can assess the accuracy of the data center-driven load shifting and check whether this model is overly optimistic when predicting the impact of a load shift.

4.4 Test Case

We next perform an extensive year long analysis of the different methods to guide data center load shifting, using the data center-driven shifting model and shifting metrics outlined in Section 4.2. Considering a full year of operations allows us to provide information based on a range of different operating conditions which gives a good overall idea of the behavior of each shifting metric.

For our analysis we use the IEEE RTS-GMLC system [101]. This system has 73 buses, 158 generators and 120 lines. The network has three regions, $\mathcal{R}_1 = \{v_1, \ldots, v_{24}\}$, $\mathcal{R}_2 = \{v_{25}, \ldots, v_{48}\}$ and $\mathcal{R}_3 = \{v_{49}, \ldots, v_{73}\}$, which are used as definitions of the regions when evaluating λ_{average} and λ_{excess}. Since the original system does not contain any loads that are designated as data centers, we assign data centers at buses $3, 7, 28$ and 70. We assume that each of the data centers consume a fixed power of 250 MW throughout the year. For all other loads and renewable generation, we use the hourly load and generation data provided with [101]. The system serves a total of $44,821,000$ MWh of load over the course of the whole year and the data centers account for $8,784,000$ MWh or 19.6% of the total energy consumption (though the relative share varies over time).

Adding these large data center loads to the network greatly increases the total system load and results in time steps where the original DC OPF is infeasible. To remedy this we set the minimum generation constraint, $P_g^{\min} = 0$ for all $g \in \mathcal{G}$, and increase the maximum generation constraints by 50% of the original value. We allow each data center to shift up to 20% of its load, or 50 MW, and put no limitations on how much load each data center can shift to one another. This means throughout the year at most $1,756,800$ MW of load can shift, or 3.92% of the total system load.

4.5 Comparison of Shifting Metrics

	CO_2 tons	Generation \$	Curtailment (MW)	$\mu_{\%,\,CO_2}$	$\mu_{\%,\,\$}$	$\mu_{\text{shift},\,CO_2}$	$\mu_{\text{shift},\,\$}$
DC OPF	13.96	322.69	4.01				
DC-FLEX	13.87 (-0.65%)	319.19 (-1.09%)	3.90 (-2.85%)	-0.05	-2.00	-0.06	-2.31
λ_{CO_2}	13.76 (-1.44%)	321.83 (-0.27%)	3.91 (-2.52%)	-0.11	-0.49	-0.14	-0.62
λ_{LMP}	13.87 (-0.68%)	319.26 (-1.07%)	3.90 (-2.83%)	-0.05	-1.96	-0.06	-1.99
λ_{average}	13.87 (-0.68%)	320.99 (-0.53%)	3.92 (-2.33%)	-0.06	-0.95	-0.09	-1.64
λ_{excess}	13.98 ($+0.14\%$)	323.47 ($+0.24\%$)	4.06 ($+1.16\%$)	$+0.01$	$+0.44$	$+0.03$	$+1.11$

Table 4.1: Carbon emissions, generation cost and curtailment (all $\times 10^6$) after shifting based on different shifting metrics.

	CO_2 tons	Generation \$	Curtailment (MW)	$\mu_{\%,\,CO_2}$	$\mu_{\%,\,\$}$	$\mu_{\text{shift},\,CO_2}$	$\mu_{\text{shift},\,\$}$
DC OPF	13.96	322.69	4.01				
DC-FLEX	13.87 (-0.65%)	319.19 (-1.09%)	3.90 (-2.85%)	-0.05	-2.00	-0.06	-2.31
λ_{CO_2}	13.62 (-2.45%)	319.60 (-0.96%)	3.81 (-4.94%)	-0.19	-1.77	-0.25	-2.22
λ_{LMP}	13.81 (-1.12%)	316.37 (-1.96%)	3.81 (-4.99%)	-0.09	-3.60	-0.09	-3.66
λ_{average}	13.77 (-1.42%)	318.76 (-1.22%)	3.83 (-4.45%)	-0.11	-2.22	-0.20	-3.83
λ_{excess}	13.93 (-0.27%)	321.96 (-0.23%)	4.00 (-0.17%)	-0.02	-0.42	-0.05	-1.05

Table 4.2: Predicted carbon emissions, generation cost and curtailment (all $\times 10^6$ after shifting based on the different shifting metrics.

In this section we compare the performance of the different shifting metrics and benchmark against ISO-controlled load shifting.

4.5.1 Carbon Emissions Reduction from ISO-controlled Load Flexibility

We first compare the impact of ISO-controlled load shifting on generation cost and carbon emission reduction by comparing results obtained from the DC OPF (4.2.1) and the DC OPF-FLEX (4.2.7). We repeatedly solve these two problems with cost function (4.2.8) and values of α ranging from 0 to 1. Figure 4.1 shows the generation cost and carbon emissions across all the different solutions for the cases with and without data center load flexibility. The low generation cost/high carbon emissions solution in the bottom right corner corresponds to the setting with $\alpha = 1$, where the ISO only considers cost minimization in their solution. The high generation cost/low carbon in the top left corner corresponds to the case with $\alpha = 0$, where the ISO minimizes carbon emissions. The intermediate values of alpha gives rise to the solutions along the blue and yellow lines.

We observe that as α is reduced, there is first a large decrease in carbon emissions with only a small increase in generation cost, but as we approach $\alpha = 0$, there is both a large decrease in carbon emissions and a large increase in generation cost. The point $\alpha = 0.1$ is where the gradient of each curve changes, indicating that if the ISO would even lightly weight minimizing carbon

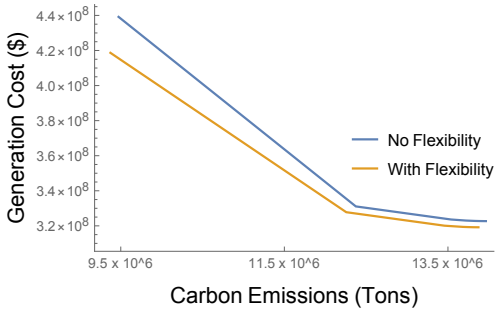


Figure 4.1: Trade off as ISO minimizes cost and carbon with and without data center flexibility.

emissions, substantial reductions could be achieved without dramatically increasing generation costs. This trend is similar for both cases (with/without flexibility), but the solution with flexibility consistently has lower generation cost for a solution with comparable carbon emissions. Similarly, the solution with flexibility can achieve lower carbon emissions at similar generation cost. This demonstrates the benefits of flexibility in general.

However, in our case, we are interested in understanding the impact of flexibility on carbon emission reductions for a specific choice of cost function. We can gain an initial understanding of this by comparing the difference in carbon emissions for the solutions without and with flexibility for $\alpha = 1$, i.e., when we only minimize cost. These two solutions correspond to the rightmost points on the blue and yellow lines. We observe that the solution with flexibility (yellow line) has lower carbon emissions than the solution with no flexibility (blue line). Specifically, the carbon emissions are reduced by 91 CO_2 tons (0.65%). This indicates that for a given cost function, the carbon emissions reduction achieved through data center load flexibility only amounts to a small reduction of the total system emissions, even in the best case scenario where the ISO is using all available system information to make the best possible use of the additional flexibility.

4.5.2 Comparison of Outcomes with Different Shifting Metrics

We next compare the performance of data center-controlled load shifting with different shifting metrics against each other and against the ISO-controlled benchmark. Using the data center-driven load shifting model and network parameter values as described in Section 4.4, we obtain solutions

for the whole year with each shifting metrics. We also include the results obtained from the initial DC OPF (without load flexibility, minimizing cost) and from the DC OPF-FLEX (with load flexibility, minimizing cost). The results in Table 4.1 represent the final results of the load shifting *after* the ISO has solved the DC OPF with the shifted load. The two first lines represent the benchmark results with the DC OPF and the DC OPF-FLEX, and the shifting metrics are ranked based on their total achieved carbon emission reduction. Observe that the DC OPF and DC OPF-FLEX models are both evaluated when $\alpha = 1$.

We notice that the λ_{CO_2} metric is by far the best in terms of reducing carbon emissions, both in terms of the total carbon emission reduction and the reduction per shifted MW of load $\mu_{\text{shift, CO}_2}$. On average, for every 1 MW shifted, we save 0.14 tons of carbon. The total carbon emission reduction is -1.44% which is twice as large as the reduction achieved with the DC OPF-FLEX. This is a particularly interesting result, because it indicates that data center-driven load shifting with respect to λ_{CO_2} achieves higher carbon reductions than if the data centers were to relinquish their flexibility to the ISO. This supports the idea that direct action by the data centers to minimize carbon emissions can make a substantial difference in carbon emission reductions. We note that the carbon emission reduction is achieved while the system cost is reduced by 0.27%.

Next, we observe that shifting with respect to λ_{LMP} results in carbon and cost savings that are comparable to the case where the ISO assumes control of the data center flexibility. The two cases decrease carbon emissions by -0.68% and -0.65% and give an overall decrease in generation cost of 1.07% and 1.09%, respectively. This similarity is explained by observing that when the data centers shift with respect to λ_{LMP} they have similar objectives and the same constraints as when the ISO assumes control of the data center flexibility. The main difference is that DC-FLEX gives a global solution while shifting with respect to λ_{LMP} relies on local sensitivity information.

Further, we observe that shifting with respect to λ_{average} gives a similar overall decrease in carbon as λ_{LMP}. However, when the reduction is considered per MW shifted as in μ_{shift}, we see that λ_{average} results in a greater carbon savings per MW shifted. This demonstrates that if a cost were to be associated to shifting load, λ_{average} would be more effective.

Finally, the shifting metric based on excess availability of low carbon power λ_{excess} actually increases the carbon emissions and generation costs of the network. This is a counter-intuitive since shifting load to a region of the network with the most excess low carbon resources seems like it should allow the data centers to make more use of low carbon generation sources. This is clearly not always the case as the actual location of the data center must be such that the data center can access the low carbon power. Line constraints within a region can prevent a data center from having access to these low carbon resources.

4.5.3 Comparison of Accuracy

For each of the data center-driven shifting metrics, we can assess the accuracy of the metric as the difference between the predicted carbon emission reduction (obtained from the data center internal optimization problem (4.2.2) in Step 2) and the actual change in carbon emissions (computed from based on the DC OPF in Step 3).

Table 4.2 and Table 4.1 show the predicted and actual change in carbon emissions, generation cost and curtailment, respectively. By comparing the results in the two tables, we see that the predicted carbon savings are better than the actual savings for all of the metrics. This implies that the shifting metrics are not able to provide an entirely accurate picture of the impact of load shifting. This is largely because the shifting model relies on a linear approximation of (4.2.1), which assumes the binding constraints remain the same before and after shifting. This need not be the case, which then makes the original linearization inaccurate.

In particular we notice that λ_{excess} predicts an overall decrease of $14,000$ tons of carbon or -0.05 tons per MW shifted, but actually produces a solution which leads to an increase in carbon emissions as seen in Table 4.1. Further we observe that λ_{average} predicts a better carbon savings then λ_{LMP}, both overall and per MW shifted. However, this contradicts the results in Section 4.5.2 which indicated that λ_{LMP} lead to a better overall savings.

4.6 A more realistic data center load shifting model

The above model has several drawbacks. First, it is unrealistic to assume that the ISO resolves the market clearing twice, once before and once after the shifting has happened. Second, since the model is linear, we tend to see large load shifts even with small differences in λ_{CO_2} between data center locations. Since λ_{CO_2} is a local sensitivity factor that is only accurate near the previous optimal solution, these large shifts lead to inaccurate results that sometimes increase carbon emissions. To address these issues, we introduce two improvements to the model: cumulative load shifting, regularization and more accurate data.

4.6.1 Cumulative load shifts

We refine the model defined above by considering *cumulative load shifts*. Instead of resolving the DC OPF in Step 3 of the above model, the load shift is applied to the market clearing in the next time step. Specifically, the algorithm runs as follows:

Step 1: At time t, the ISO solves the DC OPF (4.2.1) with data center load set to P_d^t.

Step 2: Given information about λ_{CO_2} as described above, the data center operator computes a load shift ΔP_d^t according to (4.2.2). Then, the data center load for time $t+1$ is set to $P_d^{t+1} = P_d^t + \Delta P_d^t$, and the algorithm proceeds to Step 1 of the next time step.

While the cumulative load shifting model more accurately reflects the current market set up, it introduces an additional inaccuracy in our model. The locational marginal carbon emission value λ_{CO_2} at each data center is derived as a linearization from the operating point at time t, but the internal data center shifting optimization will only affect the market clearing at time $t+1$. The expectation is that since operating conditions remain similar between time steps, shifting with respect to λ_{CO_2} will still lead to a decrease in total system carbon emissions. We also note that cumulative load shifting can increase accuracy relative to the existing model, particularly the load shift allowed in each time step is small (i.e., only a small fraction ϵ can be shifted). In this case, changes in the data center load build up slowly over time. This is in contrast to our previous model, where the data center load was reset to the original value P_d in each time step.

4.6.2 Regularizing load shifts

To discourage large load shifts which can cause oscillations and increased emissions, we propose to use a regularization term (i.e., a quadratic penalty) that discourages large shifts. Specifically, this model replaces the objective value (4.2.2a) with

$$\sum_{i \in \mathcal{C}} \lambda_{\mathrm{CO}_2,i} \Delta P_{d,i} + \gamma \|\Delta P_{d,i}\|_2^2$$

where $\gamma \in \mathbb{R}$ is a regularization parameter. The goal in using this regularization term is to discourage large shifts that lead to an increase in carbon emissions as well as increase the accuracy of the data center driven shifting model. However, the regularization term can also be interpreted as a quadratic cost on load shifting. This ensures that while small shifts are cheap and frequent, we only shift a large amount of load when there will be a large reduction in carbon emissions.

Throughout the rest of this paper we refer to the model outlined in this section as ($\lambda_{\mathrm{CO}_2}-$shift).

4.6.3 More realistic data

We again perform an extensive year long analysis of the carbon reduction methods mentioned above using the RTS-GMLC system [101]. This time, for all other loads and renewable generation, we use the real time, i.e. 5 minute, load and generation data provided with [101]. We assume that cumulatively, the four data centers consume a fixed power of 1000 MW at each time step throughout the year, although the distribution of that power among the four data centers varies. We assume at time step 0, each data center starts with 250 MW of load. Over the course of the year, this system serves $526,220,000$ MW of load, and $105,408,000$ MW or roughly 20.03% of it is data center load.

As above, we change the generation limits by setting $P_g^{\min} = 0$ for all $g \in \mathcal{G}$, and increase the maximum generation limits by 50%. At each time step we allow each data center to shift up to 20% of its total capacity, i.e. 50 MW, and enforce that data center capacities remain between 0 and 300 MW. Further, we put no limitations on how much load each data center can shift to one another.

4.7 Benchmark model for optimal shifting

We introduce a new model to benchmark this more accurate data center driven shifting model. Since the shifts provided by λ_{CO_2} are calculated by a linear sensitivity, they can be inaccurate, even giving shifting profiles that increase carbon emissions.

The problem of identifying the optimal load shift data center operators should employ to minimize carbon emissions can be modelled as a bilevel linear program. The upper level problem identifies the optimal choice of load shift ΔP_d to minimize carbon, i.e.,

$$\begin{aligned} &\min_{\Delta P_d, s, P_g^*} \; g^T P_g^* \\ &\quad \text{s.t.} \quad P_g^* = \arg\min \text{(DC-shift)} \\ &\qquad (\Delta P_d, s) \in \mathcal{P} \end{aligned} \tag{Opt-shift}$$

Here, the last constraint represents the set of feasible load shifts from the data center perspective, i.e., $\mathcal{P}$ is the polytope of permissible load shifts defined by the constraints in (4.2.2). The first constraint states that the generation value P_g^* is the solution to the lower level optimization problem (DC-shift). This problem is a version of the standard DC OPF (4.2.1) where the nodal balance constraints include the change in demand. Formally we write it as

$$\begin{aligned} &\min_{P_g, \theta} \; c^T P_g \quad \text{subject to} \\ &\text{Constraints (4.2.1c), (4.2.1d), (4.2.1e)} \\ &\sum_{\ell \in \mathcal{G}_i} P_{g,\ell} - \sum_{\ell \in \mathcal{D}_i} (P_{d,\ell} + \Delta P_{d,\ell}) = \sum_{j:(i,j)\in\mathcal{L}} \beta_{ij}(\theta_i - \theta_j), \qquad \forall i \in \mathcal{N} \end{aligned} \tag{DC-shift}$$

As in (λ_{CO_2}-shift), we also consider cumulative load shifting in (Opt-shift). Specifically, at each data center $\ell \in \mathcal{C}$, at time step t we assume the load $P_{d,\ell}$ in (DC-shift) reflects the sum of new load $P_{d,\ell}$ from time t and the load shift $\Delta P_{d,\ell}$ from time $t-1$. Herein out when we refer to the model (Opt-shift) we assume it is employed with this cumulative load shifting.

4.8 Computational Results

We next analyze the efficacy in carbon reduction of (λ_{CO_2}-shift) versus (Opt-shift).

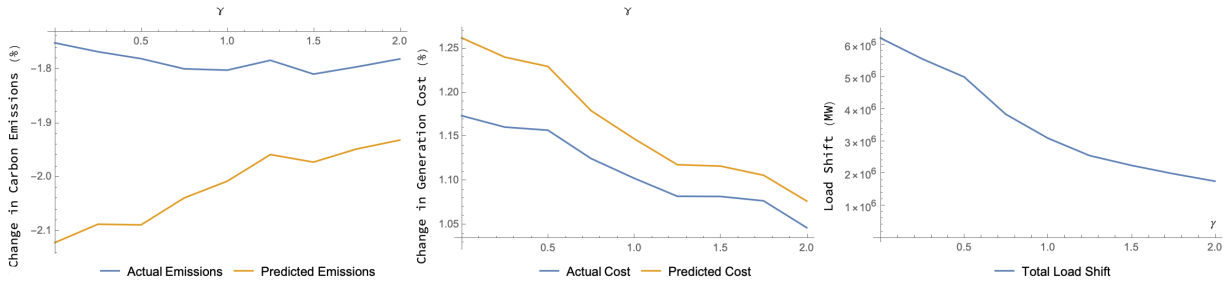


Figure 4.2: Change in carbon (left) emissions, generation cost (middle) and load shift (right) as the regularization parameter γ varies.

4.8.1 The effect of regularization

We first investigate the effect of the regularization parameter γ. The effect of various regularization parameters on generation cost, total system carbon emissions and total load shift is shown in Figure 4.2, where the orange and blue lines represent the predicted and actual values, respectively. Figure 4.2(a) show that the minimum total system carbon emissions occurs when the regularization parameter $\gamma = 1.5$ is used. In addition, we see in both Figure 4.2(a) and (b) that as the regularization parameter γ increases, the difference between the predicted carbon emissions and generation cost and the actual carbon emissions and generation costs decreases. This indicates that including a regularization term helps not only in the efficacy of the data center driven shifting model, but also in the accuracy.

The reason regularization is considered is to discourage load shifting in cases where it is not predicted to make large differences. Figure 4.2(c) shows how as the regularization parameter increases, the total load shifted throughout the year decreases. We see that when the regularization parameter is set at $\gamma = 1.5$, the total amount of load shifted is less than half of the amount of load shifted when $\gamma = 0$. Considering that the carbon emissions and generation cost when $\gamma = 1.5$ are lower than when $\gamma = 0$, this demonstrates that shifting less load, more strategically can lead to a larger reduction in carbon emissions and a smaller increase in generation costs.

	DC OPF	(Opt-shift)	$\lambda_{CO_2}-$shift: $\gamma = 0$	$\lambda_{CO_2}-$shift: $\gamma = 1.5$
Generation Cost	$3,802,706,000$	$4,981,076,000$	$3,847,332,000$	$3,843,847,000$
CO_2 Emissions	$164,402,000$	$110,444,000$	$161,522,000$	$161,427,000$
Total Shifts	0	$1,048,000$	$6,199,000$	$2,245,000$

Table 4.3: Summary of results from all models

4.8.2 Comparison with Opt-Shift and Original DC OPF solution

We next compare the solutions for (λ_{CO_2}-shift) with regularization parameters $\gamma = 0$ and $\gamma = 1.5$ with the original DC OPF solution and the solution obtained using our benchmark model (Opt-shift). These results are given in Table 4.3. We see that when considering (λ_{CO_2}-shift) with no regularization, carbon emissions relative to the original DC OPF decreases by around 2.8 million tons or 1.75%. This reduction is achieved while shifting around 6.2 million MW of load. Conversely, once the regularization term $\gamma = 1.5$ is added, we achieve an even greater reduction in carbon emissions, namely $2,975,000$ tons or 1.81% while only shifting around 2.25 million MW of load. In addition, when considering regularization, total system generation costs only increased by 1.08% while without regularization it increased by 1.17%.

In contrast to the above results, we see a dramatic carbon savings when using the benchmark (Opt-shift). In this case we save $53,958,000$ tons of carbon, i.e. 32.82%. This occurs while only shifting a little over 1 million MW. This dramatic savings occurs at a major increase to generation costs. Namely, (Opt-shift) results in an increase in $\$1,178,370,000$ to generation costs or 30.99% over the original DC OPF. This benchmark model suggests that dramatic reductions in carbon emissions are possible even with limited data center flexibility, but come at a large increase to generation costs.

4.8.3 Carbon Emissions vs Generation Costs

As seen above, minimizing carbon emissions can lead to an increase in generation cost. To better understand the trade-off between carbon emissions and cost, we consider the benchmark

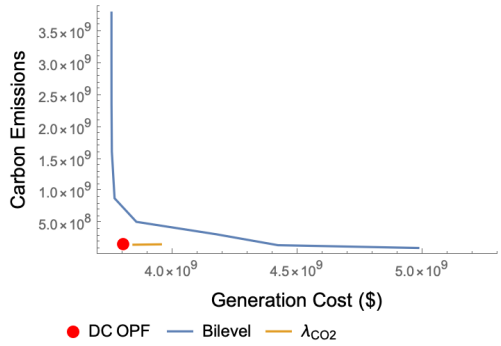


Figure 4.3: Trade off between carbon emissions and generation cost.

model (Opt-shift) with objective function

$$(\alpha c^T + (1-\alpha)g^T)P_g^*$$

and $(\lambda_{\text{CO}_2}-\text{shift})$ with objective function

$$(\alpha \text{LMP} + (1-\alpha)\lambda_{\text{CO}_2})\Delta P_d + 1.5 \cdot \|\Delta P_d\|_2^2$$

in place of (4.2.2a) where $\alpha \in [0,1]$ is a trade off parameter that allows us to weight the emphasis on minimizing carbon emissions versus generation costs and LMP is a vector of the locational marginal prices at each node. The trade off between minimizing carbon emissions and generation cost is shown graphically in Figure 4.3.

When considering $(\lambda_{\text{CO}_2}-\text{shift})$, shown in yellow, we see a small variation in the overall system carbon emissions and generation cost that remains close to the carbon emissions and generation cost of the original DC OPF. This is consistent with the results shown above, and is due to the fact that this model considers small shifts away from an operating point that minimizes generation costs. The benchmark model (Opt-shift) produces a much larger variation in operating points as we change the trade-off parameter α. As α increases, the model produces a large increase in carbon emissions for only a moderate cost savings. In addition, we see that for (Opt-shift) to achieve lower carbon emissions than the DC OPF and the $(\lambda_{\text{CO}_2}-\text{shift})$, a major increase to generation cost is needed. This demonstrates that even with limited geographic load shifting flexibility, a large reduction in carbon emissions is possible but it comes at the price of significantly higher generation costs.

Figure 4.3 also demonstrates an interesting phenomenon, namely the greedy nature of (Opt-shift). When only trying to minimize carbon emissions, (Opt-shift) is able to reduce total system carbon emissions by roughly 33% but this comes at a significant increase to total system generation cost. However, for the same generation cost, (Opt-shift) gives a solution with higher carbon emissions than the DC OPF or (λ_{CO_2}-shift). This demonstrates that the greedy nature of (Opt-shift) is not necessarily an optimal way to shift load over a long time span. Specifically, (Opt-shift) finds a load shift that gives the largest reduction in carbon emissions at that time step, with no consideration to how the load shift will affect the carbon emissions of the system at the next time step. Using forecasts of future load and generation information to aid in a long term load shifting strategy is left as future work.

4.8.4 Data Center Operating Load

Finally, we consider the impact of each model on the data center operating load. We consider (λ_{CO_2}-shift) with regularization parameter $\gamma = 1.5$ and (Opt-shift), and two different limits on the amount of load that can be shifted in each time step, $\epsilon = 0.01$ and $\epsilon = 0.2$.

In Figure 4.4 we see the operating conditions of each data center over the course of the first day when using (λ_{CO_2}-shift) when $\epsilon = 0.01$ (left) and $\epsilon = 0.2$ (right). In both cases we see similar overall trends in operating load. However, $\epsilon = 0.2$ leads to much quicker changes and also dramatic oscillations in the load at data centers 1 and 3 towards the end of the day. Similarly, in Figure 4.5 we see the operating conditions of each data center over the course of the first day using (Opt-shift) when $\epsilon = 0.01$ (left) and $\epsilon = 0.2$ (right). Again, we see similar trends data center load for both values of ϵ, but for (λ_{CO_2}-shift), $\epsilon = 0.2$ leads to more oscillations in data center operating load.

Interestingly, there are some differences between the (λ_{CO_2}-shift) and (Opt-shift). In both cases we see an initial pull for data center 4 to operate at maximum capacity while the other data centers operate at lower capacities. This implies that the λ_{CO_2} value for data center 4 is accurately dictating that it is the most carbon neutral data center. In contrast, we see that when shifting with respect to (λ_{CO_2}-shift), data center 2 is also operating at maximum capacity. This is in contrast to shifting

when using (Opt-shift). In this instance data center 2 initially drops to be the data center operating at the lowest load. This discrepancy highlights the inaccuracy when shifting with respect to λ_{CO_2}.

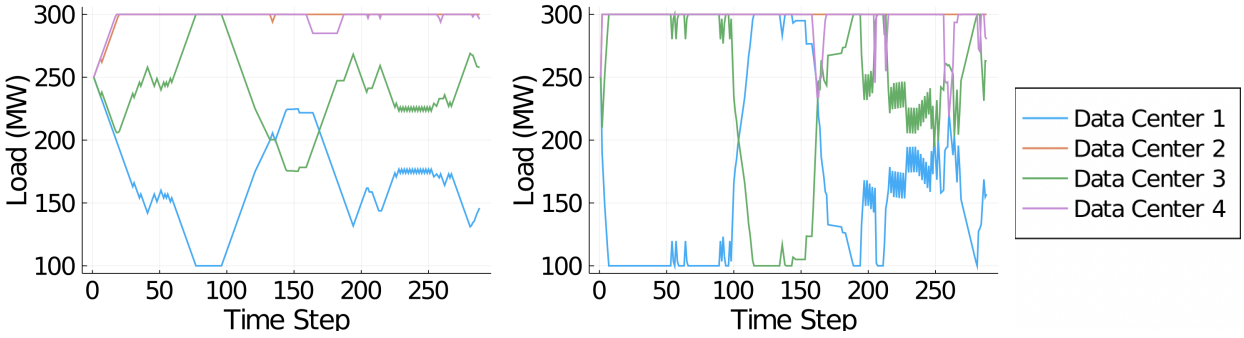


Figure 4.4: Load at each data center during the first 24 hours using λ_{CO_2}-shift with $\gamma = 1.5$ and $\epsilon = 0.01$ (left) and $\epsilon = 0.2$ (right).

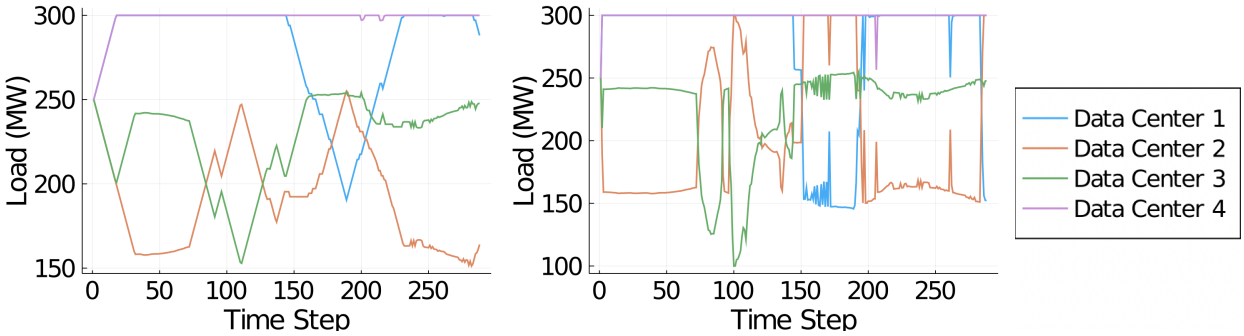


Figure 4.5: Load at each data center during the first 24 hours using (Opt-shift) with $\epsilon = 0.01$ (left) and $\epsilon = 0.2$ (right).

Finally, we investigate how using different ϵ values impacts the overall effect on carbon emissions. In Figure 4.6 we see the change in total system carbon emissions as ϵ varies for both (λ_{CO_2}-shift) with $\gamma = 1.5$ as well as (Opt-shift). In both cases we see only a very mild decrease in total carbon emissions as we allow ϵ to increase. Further, for (λ_{CO_2}-shift), we see that as ϵ increases, the accuracy of the model decreases and once $\epsilon > 0.1$, the carbon emissions starts to increase. This indicates that allowing small shifts not only is more desirable from an operational stand point to avoid rapid changes and oscillations in data center loading, but it leads to similar carbon savings as allowing larger shifts.

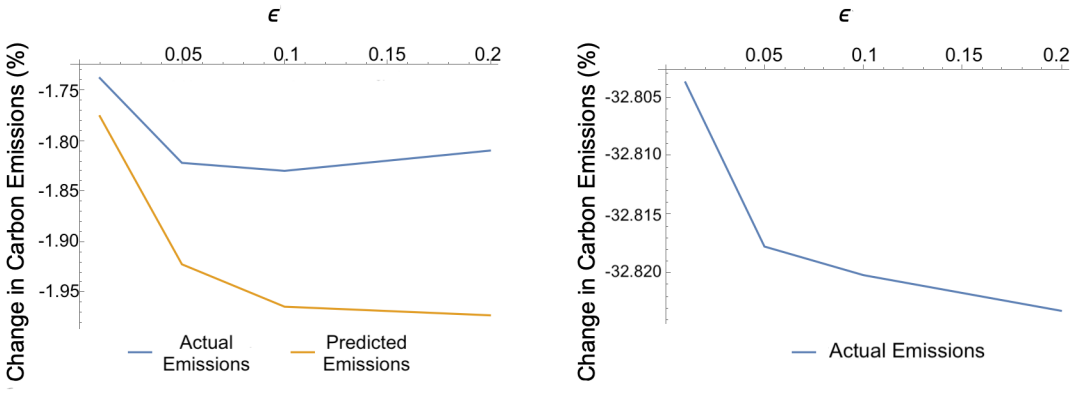


Figure 4.6: Predicted and actual change in carbon emissions using $\lambda_{CO_2}-$shift (left) and the change in carbon emissions using (Opt-shift) (right) for varying epsilon values.

Chapter 5

Method of moments for Gaussian mixture models

5.1 Problem set-up

A fundamental problem in statistics is to estimate the parameters of a density from samples. This problem is called *density estimation* and formally it asks, "Given n samples from an unknown distribution p, can we estimate p"? To have any hope of solving this problem we need to assume our density lives in a family of distributions. One family of densities known as Gaussian mixture models are a popular choice due to their broad expressive power.

Theorem 5.1.1. *[102, Chapter 3] A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components.*

Theorem 5.1.1 motivates our study of Gaussian mixture models. These are ubiquitous in the literature with applications in modeling geographic events [103], the spread of COVID-19 [104], the design of planar steel frame structures [105], speech recognition [106, 107, 108], image segmentation [109] and biometrics [110].

A *Gaussian random variable*, X, has a probability density function given by

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

where $\mu \in \mathbb{R}$ is the mean and $\sigma \in \mathbb{R}_{>0}$ is the standard deviation. In this case we write $X \sim \mathcal{N}(\mu,\sigma^2)$. A random variable X is the *mixture of k Gaussians* if its probability density function is the convex combination of k Gaussian densities. Here we write $X \sim \sum_{\ell=1}^{k} \lambda_\ell \mathcal{N}(\mu_\ell,\sigma_\ell^2)$ where

$\mu_\ell \in \mathbb{R}$, $\sigma_\ell \in \mathbb{R}_{>0}$ for all $\ell \in [k] = \{1, \ldots, k\}$ and $(\lambda_1, \ldots, \lambda_k) \in \Delta_{k-1} = \{\lambda \in \mathbb{R}^k_{>0} : \sum_{i=1}^k \lambda_i = 1\}$. Each λ_ℓ, $\ell \in [k]$, is the mixture weight of the ℓth component.

A *Gaussian k-mixture model* is a collection of mixtures of k Gaussian densities. Often one imposes constraints on the means, variances or weights to define such models. For example, one might assume that all variances are equal or that the mixture weights are all equal. The former is known as a homoscedastic mixture model, and the latter is called a uniform mixture model in the literature. In this chapter we consider four classes of Gaussian mixture models.

1. The λ-weighted model, where the mixture weights are fixed for $\lambda \in \Delta_{k-1}$
2. The λ-weighted homoscedastic model, which is the λ-weighted model under the additional assumption that the variances are equal
3. The λ-weighted known variance model, where the weights and variances are assumed known
4. The $k = 4$ model

We wish to do parameter recovery for these Gaussian mixture models, that is, we would like to solve the following problem.

Problem 5.1.2. Given samples, $y_1, \ldots, y_N$, distributed as the mixture of k Gaussian densities, recover the parameters $\mu_i, \sigma_i^2, \lambda_i$ for $i \in [k]$.

It is important to distinguish parameter recovery from density estimation. For density estimation, one wishes to estimate a density that is close to the true density, with no restriction on how close each individual component is. In this chapter we wish to do parameter recovery. Namely, we wish to recover accurate estimates of the mean, variance and mixing weight of each component. It is clear that density estimation follows trivially once all of the parameters are known. The important distinction between density estimation and parameter recovery is illustrated next.

Example 5.1.3. Consider a mixture of three univariate Gaussians with sample moments $m_1, \ldots, m_8$ given by:

$$(0.1661,\ 2.133,\ 1.3785,\ 12.8629,\ 16.0203,\ 125.6864,\ 239.2856,\ 1695.5639).$$

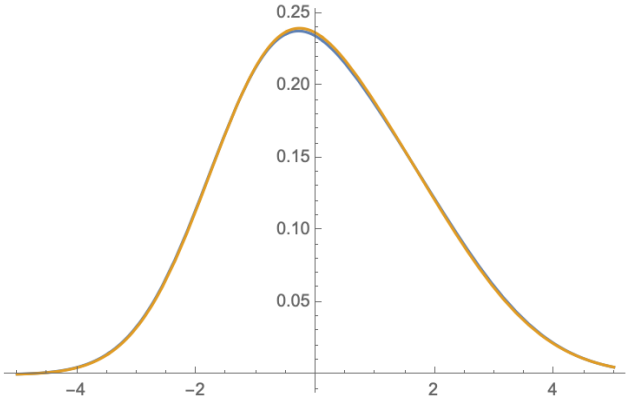


Figure 5.1: Two distinct Gaussian mixture densities with $k = 3$ components and the same first eight moments.

There are two Gaussian mixture densities with these eight moments. These densities are shown in Figure 5.1 where it is seen that they are almost indistinguishable. In contrast, the individual components and weights are noticeably different. The weights and individual components for each of the mixture models are shown in Figure 5.2.

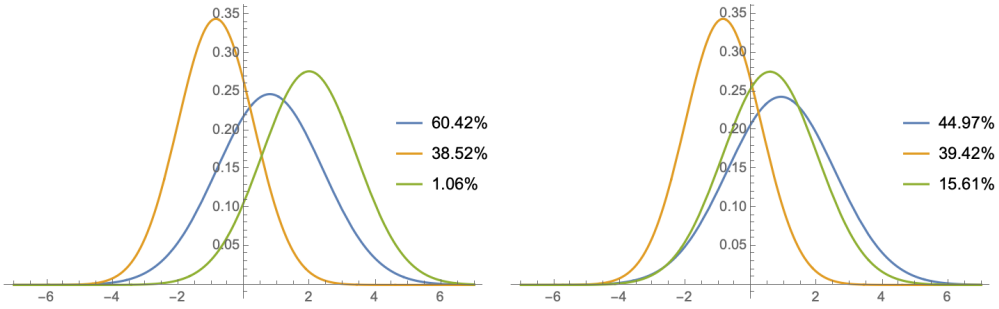


Figure 5.2: Individual components of two Gaussian mixture models with similar mixture densities.

Problem 5.1.2 is well-defined because Gaussian mixtures are identifiable [111]. Specifically, one can recover the mean, variance and weight of each component if given the full mixture density.

One idea to solve Problem 5.1.2 is to use *maximum likelihood estimation*. Maximum likelihood estimation aims to maximize the likelihood function by solving the following optimization

problem:

$$\text{argmax}_{\mu,\sigma^2,\lambda} \prod_{i=1}^{n} \sum_{i=1}^{k} \lambda_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\Big(- \frac{(y_i - \mu_i)^2}{2\sigma_i^2} \Big). \tag{5.1.1}$$

Unless $k = 1$, (5.1.1) is a nonconvex optimization problem and obtaining a global optimum is difficult or impossible. In general (5.1.1) is unbounded, so no global maximum exists. Iterative algorithms such as expectation maximization (EM) try to find the largest local maximum [112]. On top of being sensitive to the starting point, another downside of the EM algorithm is that it needs to access all data in each iteration, which is prohibitive for applications with large data sets. Even worse, there is no bound on the number of critical points of the likelihood function and in general these estimates are transcendental [113].

Recent work has analyzed the local behavior of (5.1.1) by considering maximum likelihood estimation for two Gaussians in $\mathbb{R}^n$ when $\lambda_1 = \lambda_2 = \frac{1}{2}$ and all of the covariance matrices are known and equal — this is a special case of the models we consider. It has been shown that in this regime the EM algorithm converges to the global optimum in 10 iterations [114]. Other work has studied the global landscape of the EM algorithm and the structure of local optima in this setting [115, 116]. Further work has considered inference for Gaussian mixture models with known mixing coefficients and identity covariance matrices [117] and clustering analysis of the mixture of two Gaussians where the covariance matrices are equal and unknown [118]. When these covariance matrices are further assumed to be spherical, [119] gives polynomial time approximation schemes for (5.1.1). Recently, techniques from numerical algebraic geometry have been used to identify the number of components in a Gaussian mixture model [120]. Further progress has been made on giving optimal sampling bounds needed to learn a Gaussian mixture model [121].

A recent revolution of robust statistics has led to the development of algorithms to learn the parameters for mixtures of multivariate Gaussians when some of the samples are corrupted. A Computationally efficient algorithm with dimension-independent error guarantees for agnostically learning several fundamental classes of high-dimensional distributions including a single Gaussian and mixtures of spherical Gaussians was given in [122]. Clustering algorithms and learning algorithms for Gaussians that are well separated were given in [123] and [124] respectively. For a

mixture of two Gaussians with equal mixture weights, [125] gives an algorithm to learn each mean and variance up to a distance that is poly(ϵ) from the true density. Recent work gives a polynomial-time algorithm for robustly learning the parameters of a high-dimensional mixture of a constant number of Gaussians under assumptions on the weights, the covariances, and the separation between the components [126]. A downside to this algorithm is that it runs in time polynomial in the sample size which can be prohibitive for large data sets.

Another idea for density estimation in this set-up is to use the *generalized method of moments*. The generalized method of moments was proposed in [127] and aims to minimize the difference between the fitted moments and the sample moments. For Gaussian mixture models, this again cannot be solved in a way guaranteeing global optimality due to the nonconvexity of the moment equations. Recently this method has been remedied for Gaussian mixture models in one dimension with the same variance parameter, where the authors provably and efficiently find the global optimum of the generalized method of moments [128]. It is important to note that in many of the cases above, assumptions are made on the values that each Gaussian component can take and also that these algorithms focus on density estimation, not parameter recovery. In other words, with high probability they return a density that is close to the true density with no guarantees on how close the estimated components are to the true ones.

In this chapter we are interested in identifying the means, variances and weights of each mixture *from its moments* up to a certain order. It has been shown that one dimensional Gaussian k mixture models can be uniquely recovered using the first $4k-2$ moments [129] and using the first $3k-1$ moments generically gives finitely many solutions [130]. Multivariate Gaussians are still identifiable [131] and there exists a finite number of moments that identify them [132].

In this chapter we propose using the method of moments to estimate the density arising from the mixture of k multivariate Gaussians. This methodology was first proposed and resolved for the mixture of two univariate Gaussians by Karl Pearson [133]. Pearson reduced solving this system of 6 polynomial equations in the 6 unknown density parameters, $\mu_i, \sigma_i^2, \lambda_i$, $i = 1, 2$, to understanding the roots of a degree nine polynomial with coefficients in the sample moments $\overline{m}_i$, $i \in [5]$.

Parameter recovery using the method of moments for Gaussian mixture models was revisited in 2010 in a series of papers [134, 135]. The case of a $k = 2$ mixture model in n dimensions was handled in [134] where a polynomial time algorithm was presented. This approach was generalized in [135] where an algorithm for a general k mixture model in n dimensional space was presented that scales polynomially in n and the number of samples required scales polynomially in the desired accuracy.

5.1.1 Method of moments

This chapter focuses on an approach for parameter recovery known as the *method of moments*. The method of moments for parameter estimation is based on the law of large numbers. This approach expresses the moments of a density as functions of its parameters.

Let $f : \mathbb{R} \to \mathbb{R}$ be the probability density function of a random variable X. For $i \geq 0$, the *$i-th$ moment of* X is

$$m_i = \mathbb{E}[X^i] = \int_{\mathbb{R}} x^i f(x) dx.$$

We consider a statistical model with n unknown parameters, $\theta = (\theta_1, \ldots, \theta_n)$, and consider the moments up to order n as functions of θ, $g_1(\theta), \ldots, g_n(\theta)$.

Assume $y_1, \ldots, y_N$ are independent samples from the same distribution. The rth *sample moment* is given by

$$\overline{m}_r = \frac{1}{N} \sum_{i=1}^{N} y_i^r.$$

The number of samples, N, needed to accurately estimate $\overline{m}_r$ is dependent on the distribution.

The method of moments works by using samples from the statistical model to calculate sample moments $\overline{m}_1, \ldots, \overline{m}_n$, and then solve the corresponding system $\overline{m}_i = g_i(\theta)$, $i = 1, \ldots, n$, for the parameters $\theta_1, \ldots, \theta_n$.

The moments of Gaussian distributions are polynomials in the variables μ, σ^2 and can be calculated recursively as $M_0(\mu, \sigma^2) = 1, M_1(\mu, \sigma^2) = \mu$ and

$$M_i(\mu, \sigma^2) = \mu M_{i-1} + (i-1)\sigma^2 M_{i-2}, \quad i \geq 2. \tag{5.1.2}$$

We calculate the ith moment of a mixture of k Gaussian densities as the convex combinations of $M_i(\mu_1, \sigma_1^2), \ldots, M_i(\mu_k, \sigma_k^2)$.

We are interested solving the system,

$$f_i^k(\mu, \sigma^2, \lambda) := \lambda_1 M_1(\sigma_1^2, \mu_1) + \cdots + \lambda_k M_k(\sigma_k^2, \mu_k) - \overline{m}_i = 0 \tag{5.1.3}$$

under assumptions on the parameters, μ, σ^2, λ where i varies over an appropriate index set.

As stated in the introduction, for a k mixture model with generic sample moments, the first $3k-1$ moments are needed to have a polynomial system with finitely many solutions [130]. This shows that Gaussian mixture models in one dimension are *algebraically identifiable* using moment equations

$$f_0^k = 0, \ \ldots, \ f_{3k-1}^k = 0.$$

In other words, for a generic choice of sample moments, the polynomial system (5.1.3) for $i = 0, \ldots, 3k-1$ has finitely many solutions.

5.1.2 Statistically meaningful solutions

We note that for any set of real-valued sample moments it is not guaranteed that the moment equations will give any statistically meaningful solutions. A *statistically meaningful solution* is a real valued solution with positive variances and mixing weights. In other words, it is a solution that corresponds to a true density. If the sample moments are inaccurate, it may happen that no solutions obtained from the method of moments is statistically meaningful. By the law of large numbers, as the number of samples goes to infinity the sample moments will converge to the true moments and the method of moments will return a consistent estimator [136, Theorem 9.6].

A property of parameterized polynomial systems is that the number of real solutions is constant in open Euclidean sets of the parameter space. Such open sets can be computed via cylindrical algebraic decomposition [137]. The constraints differing real solutions and statistically meaningful ones will further divide these cells. Therefore, in any of these open cells the number of statistically meaningful solutions will be constant. So long as the sample moments lie in a cell that has at least one statistically meaningful solution, the method of moments will return a true density.

Example 5.1.4. Consider the mixture of two Gaussians with equal mixing weights, $\lambda_1 = \frac{1}{2} = \lambda_2$. In this case there are four equations and four unknowns: $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2$. Restricting to $m_1 = 0, m_2 = 1$ there is one open cell that dictates whether or not there is a statistically meaningful solution. The region where there is only one statistically meaningful solution (up to symmetry) is shown in blue in Figure 5.3, where the horizontal and vertical axes correspond to m_3 and m_4, respectively.

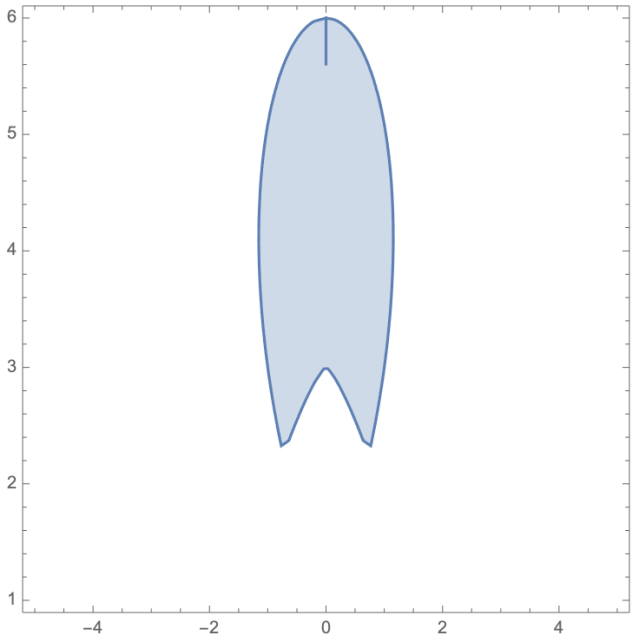


Figure 5.3: Blue region shows where there is one statistically meaningful solution for $k = 2$, $m_1 = 0, m_2 = 1$ and $\lambda_1 = \lambda_2 = \frac{1}{2}$.

5.1.3 Mixed volumes and the BKK bound

Recall Theorem 2.2.6 which states that the number of $\mathbb{C}^*$ solutions to a polynomial system is bounded above by the mixed volume of the Newton polytopes of each polynomial. We will use this theorem again for the results derived in this section.

Example 5.1.5. The mixed volume of $K_1, \ldots, K_n$ is easy to describe when K_i is a line segment from the origin to the vertex $v_i \in \mathbb{Z}^n$. The Minkowski sum $\lambda_1 K_1 + \cdots + \lambda_n K_n$ is a parallelpiped.

Hence, its volume is given by a determinant:

$$\text{Vol}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \left|\det\begin{bmatrix}\lambda_1 v_1 & \lambda_2 v_2 & \ldots & \lambda_n v_n\end{bmatrix}\right|.$$

So $\text{MVol}(K_1, \ldots, K_n)$ equals the absolute value of the determinant of the matrix with the vertices $v_1, \ldots, v_n$ as its columns.

Recall that an important property of mixed volumes is that they are monotonic. Namely, if $\hat{P}_1 \subseteq P_1$ then

$$\text{MVol}(\hat{P}_1, P_2, \ldots, P_n) \leq \text{MVol}(P_1, P_2, \ldots, P_n).$$

Therefore by Example 5.1.5, taking line segments $\text{Conv}(\{0, v_i\}) = Q_i \subseteq P_i$ for $i \in [n]$, is an easy way to get a lower bound on $\text{MVol}(P_1, \ldots, P_n)$. We would like conditions under which such a lower bound is tight.

Definition 5.1.6. [138, Definition 7.29] Let $P_1, \ldots, P_m$ be convex polytopes in $\mathbb{R}^n$. We say $P_1, \ldots, P_m$ are *dependent* if there is a non-empty subset $\mathcal{I} \subseteq [m]$ such that $\dim(\sum_{i \in \mathcal{I}} \mathcal{P}_i) < |\mathcal{I}|$. Otherwise we say $P_1, \ldots, P_m$ are *independent*.

This definition may be difficult to parse on a first read. But it is related to the usual definition of *linear independence*: if each P_i is a line through the origin, then the two ideas of dependent agree. Moreover, the collection of empty sets is independent.

Recall from Section 2.2.1 the definition of $\text{init}_w(f)$ for a polynomial f and $\text{init}_w(P)$ for a polytope $P \subseteq \mathbb{R}^n$.

Proposition 5.1.7. [138, Proposition 7.36] Let $P_i = \text{Conv}(\mathcal{A}_i)$ and $Q_i = \text{Conv}(\mathcal{B}_i) \subseteq P_i$ for $i \in [n]$. The following are equivalent:

1. $\text{MVol}(P_1, \ldots, P_n) = \text{MVol}(Q_1, \ldots, Q_n)$
2. One of the following holds:
   (a) $P_1, \ldots, P_n$ are dependent i.e. $\text{MVol}(P_1, \ldots, P_n) = 0$

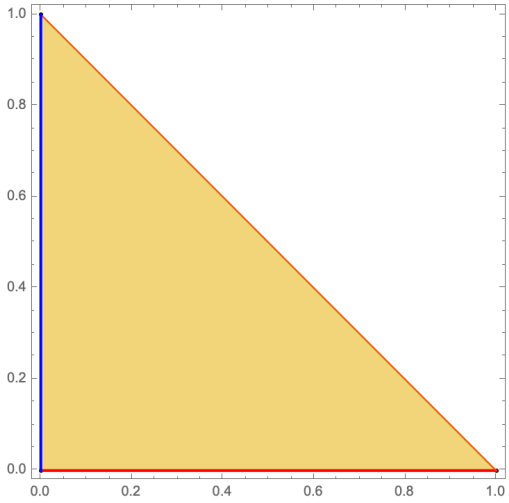


Figure 5.4: The triangle P_1 with the line segments Q_1 (red) and Q_2 (blue) from Example 5.1.8.

(b) For each $w \in \mathbb{R}^n \backslash \{0\}$, the collection of polytopes

$$\{\text{init}_w(Q_i) : Q_i \cap \text{init}_w(P_i) \neq \emptyset\}$$

is dependent.

Proposition 5.1.7 gives conditions under which is suffices to consider the (potentially much simpler) polytopes $Q_i \subseteq P_i$ to compute the mixed volume of $P_1, \ldots, P_n$.

Example 5.1.8. Consider the triangles $P_1 = P_2 = \text{Conv}(\{(0,0),(1,0),(0,1)\})$, and the line segments

$$Q_1 = \text{Conv}(\{(0,0),(1,0)\}) \subset P_1, \qquad Q_2 = \text{Conv}(\{(0,0),(0,1)\}) \subset P_2.$$

The regions P_1, Q_1 and Q_2 are shown in Figure 5.4.

Direct computation shows

$$\text{MVol}(P_1, P_2) = 1 = \text{MVol}(Q_1, Q_2).$$

We can also use Example 5.1.5 to prove $\text{MVol}(Q_1, Q_2) = 1$ and Proposition 5.1.7 to prove $\text{MVol}(P_1, P_2) = \text{MVol}(Q_1, Q_2)$ since for any nonzero $w \in \mathbb{Z}^2$, the collection of polytopes

$\{\text{init}_w(Q_i) : Q_i \cap \text{init}_w(P_i) \neq \emptyset\}$, $i = 1, 2$, contains a single point so the collection is dependent. This type of argument, where we use the dependence of polytopes and Proposition 5.1.7, is also used in the proofs of Proposition 5.2.3 and Proposition 5.2.7.

The following lemma will be of use later to apply Proposition 5.1.7 in the proof of our main results.

Lemma 5.1.9. Let $P_i \subseteq \mathbb{R}^n$ and $Q_i = \text{Conv}(\{0_n, v_i\}) \subseteq P_i$ for $i \in [n]$ be convex polytopes. Consider the set, W, of nonzero $w \in \mathbb{R}^n$ such that $Q_i \cap \text{init}_w(P_i) \neq \emptyset$ for all $i \in [n]$. I.e.,

$$W = \{w \in \mathbb{R}^n \backslash \{0_n\} : Q_i \cap \text{init}_w(P_i) \neq \emptyset, \ \forall i \in [n]\}.$$

If $\{v_1, \ldots, v_n\}$ are linearly independent, then the polytopes

$$\{\text{init}_w(Q_1), \ldots, \text{init}_w(Q_n)\}$$

are dependent for all $w \in W$.

Proof. Fix $w \in W$. By the definition of dependent, we need to show that

$$\dim\Big(\sum_{i=1}^{n} \text{init}_w(Q_i)\Big) < n.$$

Since $\text{init}_w(Q_i) \subseteq \mathbb{R}^n$, for all $i \in [n]$, $\sum_{i=1}^n \dim(\text{init}_w(Q_i)) \leq n$. Furthermore, since each Q_i is one dimensional, $\sum_{i=1}^n \dim(\text{init}_w(Q_i)) = n$ if and only if w minimizes all of Q_i for all $i \in [n]$. I.e., $\sum_{i=1}^n \dim(\text{init}_w(Q_i)) = n$ if and only if $\text{init}_w(Q_i) = Q_i$ for all $i \in [n]$.

Recalling $Q_i = \text{Conv}(\{0_n, v_i\})$, one sees $\text{init}_w(Q_i) = Q_i$ if and only if

$$0 = \langle w, 0_n \rangle = \langle w, v_i \rangle,$$

for all $i \in [n]$. Since $\{v_1, \ldots, v_n\}$ are linearly independent, the only w that satisfies this is $w = 0_n \notin W$. □

5.2 Density estimation in one dimension

We are now able to present our first set of results: efficiently finding all complex solutions stemming from the moment equations.

5.2.1 Mixed volume of λ-weighted models

First consider a λ-weighted model with k mixture components. We consider the moment system

$$f_1^k(\mu, \sigma^2, \overline{\lambda}) = 0, \ldots, f_{2k}^k(\mu, \sigma^2, \overline{\lambda}) = 0, \tag{5.2.1}$$

where $\overline{\lambda}$ are known mixing coefficients, and f_i^k, $i \in [2k]$ is as defined in (5.1.3). In this set-up the unknowns are μ_ℓ, σ_ℓ, $\ell \in [k]$.

First, we record the following fact about the moment functions M_k.

Lemma 5.2.1. The partial derivatives of M_k satisfy

$$\frac{\partial}{\partial \mu} M_k(\sigma^2, \mu) = k M_{k-1} \quad \text{and} \quad \frac{\partial}{\partial \sigma^2} M_k(\sigma^2, \mu) = \binom{k}{2} M_{k-2}.$$

Proof. We verify both by induction. For both identities, the base case $k = 1$ is immediate. Suppose $\frac{\partial}{\partial \mu} M_{k-1}(\sigma^2, \mu) = (k-1) M_{k-2}$. By the recursive relationship (5.1.2) we have

$$\begin{aligned} \frac{\partial}{\partial \mu} M_k(\mu, \sigma^2) &= M_{k-1} + \mu \frac{\partial}{\partial \mu} M_{k-1} + \sigma^2 (k-1) \frac{\partial}{\partial \mu} M_{k-2} \\ &= M_{k-1} + \mu (k-1) M_{k-2} + \sigma^2 (k-1)(k-2) M_{k-3} \\ &= M_{k-1} + (k-1) M_{k-1} = k M_{k-1}. \end{aligned}$$

Similarly, suppose that $\frac{\partial}{\partial \sigma^2} M_{k-1}(\sigma^2, \mu) = \binom{k-1}{2} M_{k-3}$. Using (5.1.2) again,

$$\begin{aligned} \frac{\partial}{\partial \sigma^2} M_k(\mu, \sigma^2) &= \mu \frac{\partial}{\partial \sigma^2} M_{k-1} + (k-1) M_{k-2} + \sigma^2 (k-1) \frac{\partial}{\partial \sigma^2} M_{k-2} \\ &= \mu \binom{k-1}{2} M_{k-3} + (k-1) M_{k-2} + \sigma^2 (k-1) \binom{k-2}{2} M_{k-4} \\ &= (k-1) M_{k-2} + \binom{k-1}{2} \left(\mu M_{k-3} + \sigma^2 (k-3) M_{k-4} \right) \\ &= (k-1) M_{k-2} + \binom{k-1}{2} M_{k-2} = \binom{k}{2} M_{k-2}. \end{aligned}$$

□

Now we prove our first algebraic identifiability result.

Proposition 5.2.2. For nonzero $\overline{\lambda}_\ell$, $\ell \in [k]$, and generic $\overline{m}_i$, $i \in [2k]$, the number of solutions to (5.2.1) is finite.

Proof. By [139, Ch. 1, Section 5] it is enough to show the Jacobian of (5.2.1) is full rank at a generic point. The Jacobian of (5.2.1) is a $2k \times 2k$ matrix with rows indexed by equations and columns indexed by the variables $\mu_1, \sigma_1, \ldots, \mu_k, \sigma_k$:

$$\begin{aligned} J_k &= \tilde{J}_k \cdot \tilde{D}_k \\ &= \begin{bmatrix} \frac{\partial M_1}{\partial \mu_1}(\sigma_1,\mu_1) & \frac{\partial M_1}{\partial \sigma_1}(\sigma_1,\mu_1) & \cdots & \frac{\partial M_1}{\partial \mu_k}(\sigma_k,\mu_k) & \frac{\partial M_1}{\partial \sigma_k}(\sigma_k,\mu_k) \\ \frac{\partial M_2}{\partial \mu_1}(\sigma_1,\mu_1) & \frac{\partial M_2}{\partial \sigma_1}(\sigma_1,\mu_1) & \cdots & \frac{\partial M_2}{\partial \mu_k}(\sigma_k,\mu_k) & \frac{\partial M_2}{\partial \sigma_k}(\sigma_k,\mu_k) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial M_{2k-1}}{\partial \mu_1}(\sigma_1,\mu_1) & \frac{\partial M_{2k-1}}{\partial \sigma_1}(\sigma_1,\mu_1) & \cdots & \frac{\partial M_{2k-1}}{\partial \mu_k}(\sigma_k,\mu_k) & \frac{\partial M_{2k-1}}{\partial \sigma_k}(\sigma_k,\mu_k) \\ \frac{\partial M_{2k}}{\partial \mu_1}(\sigma_1,\mu_1) & \frac{\partial M_{2k}}{\partial \sigma_1}(\sigma_1,\mu_1) & \cdots & \frac{\partial M_{2k}}{\partial \mu_k}(\sigma_k,\mu_k) & \frac{\partial M_{2k}}{\partial \sigma_k}(\sigma_k,\mu_k) \end{bmatrix} \cdot \tilde{D}_k, \end{aligned}$$

where $\tilde{D}_k$ is the diagonal matrix given by $(\lambda_1, \lambda_1, \lambda_2, \lambda_2, \ldots, \lambda_k, \lambda_k)$. Note that for nonzero λ_ℓ, $\ell \in [k]$, J_k is full rank if and only if $\tilde{J}_k$ is full rank. We now show that $\tilde{J}_k$ is full rank by induction on k. When $k = 1$,

$$\tilde{J}_1 = \begin{bmatrix} 1 & 0 \\ 2\mu_1 & 1 \end{bmatrix}$$

is rank 2 for any μ_1.

Note that μ_k, σ_k^2 only appear in the last two columns of $\tilde{J}_k$. Further, by Lemma 5.2.1, the nonzero entries of each row of $\tilde{J}_k$ have higher degree than the previous row.

Doing Laplace's cofactor expansion along the last two columns of $\tilde{J}_k$, we get

$$\det(\tilde{J}_k) = \det(\tilde{J}_{k-1}) \cdot \mu_k^{2k-2}\sigma_k^{2k-2} + \text{lower order terms.}$$

By induction, $\det(\tilde{J}_{k-1})$ is nonzero at generic $(\mu_1, \sigma_1^2, \ldots, \mu_{k-1}, \sigma_{k-1}^2)$. This shows that $\det(\tilde{J}_k)$ is a nonzero bivariate polynomial in μ_k, σ_k with leading coefficient $\det(\tilde{J}_{k-1})$. At generic μ_k, σ_k, $\det(\tilde{J}_k)$ does not vanish so we conclude $\tilde{J}_k$, and hence J_k, is invertible at a generic point $(\mu_1, \sigma_1^2, \ldots, \mu_k, \sigma_k^2)$.

□

Lemma 5.2.2 shows that for generic sample moments, the moment equations yield finitely many solutions. We now use Theorem 2.2.6 and Propsition 5.1.7 to give an upper bound on the number of complex solutions to (5.2.1). Recall that if N is odd, the double factorial is defined as

$$N!! = 1 \cdot 3 \cdot 5 \cdots N.$$

Theorem 5.2.3 (Mixing coefficients known)**.** *Consider a Gaussian k-mixture model with known and nonzero mixing coefficients $\overline{\lambda}_\ell$ and generic sample moments $\overline{m}_i$, $\ell \in [k]$, $i \in [2k]$. The moment system* (5.2.1) *has at most $(2k-1)!!k!$ complex solutions.*

Proof. Consider the moment equations $f_1^k, \ldots, f_{2k}^k$ as defined in (5.2.1) with variable ordering $(\mu_1, \sigma_1^2, \ldots, \mu_k, \sigma_k^2)$. Denote $P_i = \mathrm{Newt}(f_i^k)$.

Let $Q_\ell \subset P_\ell$ be the line segment defined as:

$$Q_{2\ell-1} = \mathrm{Conv}(\{0_{2k}, (2\ell-1) \cdot e_{2\ell-1}\}), \quad \ell \in [k]$$
$$Q_{2\ell} = \mathrm{Conv}(\{0_{2k}, \ell \cdot e_{2\ell}\}), \quad \ell \in [k],$$

where $0_{2k} \in \mathbb{R}^{2k}$ is the vector of all zeros and $e_\ell \in \mathbb{R}^{2k}$ is the ℓth standard basis vector. By Example 5.1.5 we have

$$\mathrm{MVol}(Q_1, \ldots, Q_{2k}) = (1 \cdot 3 \cdot 5 \cdots (2k-1)) \cdot (1 \cdot 2 \cdot 3 \cdots k) = (2k-1)!!k!.$$

We want to use the equivalence of (1) and (2) in Proposition 5.1.7 to show

$$\mathrm{MVol}(P_1, \ldots, P_{2k}) = (2k-1)!!k!.$$

Theorem 2.2.6 then gives that the number of complex solutions to (5.2.1) is bounded above by $(2k-1)!!k!$.

For a nonzero vector $w \in \mathbb{R}^{2k}$, let

$$\mathcal{I}_w = \{i \in [2k] : Q_i \cap \mathrm{init}_w(P_i) \neq \emptyset\}.$$

We will show for each $w \in \mathbb{R}^n \backslash \{0\}$, the collection of polytopes

$$\{\mathrm{init}_w(Q_i) : i \in \mathcal{I}_w\}$$

is dependent.

By Lemma 5.2.4, (which we postpone to after this proof) $\mathcal{I}_w$ is nonempty. Since each Q_i is a one dimensional line segment, it suffices to show that for any nonzero $w \in \mathbb{R}^{2k}$, w minimizes some Q_i at a unique point for $i \in \mathcal{I}_w$. This follows from the definition of dependent since each Q_i is one dimensional, so if $\operatorname{init}_w(Q_i)$ is a single point for some $i \in \mathcal{I}_w$, then $\sum_{i \in \mathcal{I}_w} \operatorname{init}_w(Q_i) < |\mathcal{I}_w|$.

We look at two cases.

- First, consider when $2\ell \notin \mathcal{I}_w$ for all $\ell \in [k]$. Since the origin is in Q_i, we have $0_{2k} \notin \operatorname{init}_w(P_{2\ell})$ for all $\ell \in [k]$. Since $P_{2\ell} \subset \mathbb{R}^{2k}_{\geq 0}$, this means $\operatorname{val}_w(P_{2\ell}) < 0$ for all $\ell \in [k]$. Hence, $w_i < 0$ for some $i \in [2k]$.

  Let i be the index of the smallest element of w. If i is odd, then

  $$P_i = \operatorname{Conv}\big(\{0_{2k}, \frac{i-1}{2}e_2, \ldots, \frac{i-1}{2}e_{2k},\ ie_1, , \ldots, ie_{2k-1}\}\big)$$

  is in the nonnegative orthant. So

  $$\operatorname{val}_w(P_i) = \min\{0, \frac{i-1}{2}w_2, \ldots, \frac{i-1}{2}w_k,\ iw_1, , \ldots iw_{2k-1}\} = iw_i.$$

  So $\operatorname{init}_w(Q_i) = \{ie_i\}$ for $i \in I_w$ and we are done.

  Now consider when i is even. Recall,

  $$P_{2\ell} = \operatorname{Conv}\big(\{0_{2k}, \ell e_2, \ell e_4, \ldots,\ \ell e_{2k}, 2\ell e_1, 2\ell e_3, \ldots, 2\ell e_{2k-1}\}\big) = \ell \cdot P_2,$$

  and so $\operatorname{init}_w(P_{2\ell}) = \ell \cdot \operatorname{init}_w(P_2)$ for all $\ell \in [k]$. Therefore, $2\ell \notin \mathcal{I}_w$ for all $\ell \in [k]$ implies that P_2 cannot be minimized at e_j for any even j. This implies that if i is even, $0 > w_i > 2w_j$ for some odd j (otherwise P_i would be minimized at $\frac{i}{2}e_i$). Let j be the index of the smallest odd element of w. In this case, P_j would be minimized at je_j so $j \in \mathcal{I}_w$. Hence $\operatorname{init}_w(Q_j)$ is $\{je_j\}$.

- Second, suppose $2\ell \in \mathcal{I}_w$ for some $\ell \in [k]$. If $\operatorname{init}_w(P_{2\ell}) \cap Q_{2\ell}$ is a point, then we are done. Otherwise, we may assume $P_{2\ell}$ is minimized by w at a face containing the line segment $Q_{2\ell} = \operatorname{Conv}(\{0_{2k}, \ell e_{2\ell}\})$.

This means

$$0 = w^T 0_{2k} \leq w^T a \quad \forall a \in P_{2\ell}.$$

So $w_i \geq 0$ for all $i \in [2k]$ because the vertices of $P_{2\ell}$ are scaled standard basis vectors. With the fact each P_i is in the nonnegative orthant, this implies

$$0 = \mathrm{val}_w(P_i) \text{ for all } i \in [2k],$$

so $0_{2k} \in \mathrm{init}_w(P_i)$ for all $i \in [2k]$.

This shows that $\mathcal{I}_w = [2k]$ so by Lemma 5.1.9, we conclude that the collection of polytopes $\{\mathrm{init}_w(Q_1), \ldots, \mathrm{init}_w(Q_{2k})\}$ is dependent.

□

Lemma 5.2.4. The index set $\mathcal{I}_w$ as defined in the proof of Theorem 5.2.3 is nonempty for any nonzero $w \in \mathbb{R}^{2k}$.

Proof. Recall that

$$P_2 = \mathrm{Conv}\big(\{0_{2k}, e_2, \ldots, e_{2k},\ 2e_1, \ldots, 2e_{2k-1}\}\big).$$

We consider three cases, depending on $\mathrm{init}_w(P_2)$.

- If either 0_{2k} or e_2 is in $\mathrm{init}_w(P_2)$, then $Q_2 \cap \mathrm{init}_w(P_2) \neq \emptyset$ and hence $2 \in \mathcal{I}_w$.
- Now suppose $e_j \in \mathrm{init}_w(P_2)$ for some even $j > 2$. This means $2w_j \leq w_i$ for all odd i and $w_j \leq w_m$ for all even m. Consider

  $$P_j = \mathrm{Conv}\big(\{0_{2k}, je_1, \frac{j}{2}e_2, \ldots, je_{2k-1}, \frac{j}{2}e_{2k}\}\big).$$

  Then $\frac{j}{2}e_j \in \mathrm{init}_w(P_j)$. Since $Q_j = \mathrm{Conv}(\{0_{2k}, \frac{j}{2}e_j\})$, we have $j \in \mathcal{I}_w$.
- Now suppose $e_i \in \mathrm{init}_w(P_2)$ for some odd $i \geq 1$. This means $w_i \leq w_j$ for all odd j and $2w_i \leq w_m$ for all even m. Consider

  $$P_i = \mathrm{Conv}\big(\{0_{2k}, \frac{i-1}{2}e_2, \ldots, \frac{i-1}{2}e_{2k},\ ie_1, \ldots, ie_{2k-1}\}\big).$$

  Then $ie_i \in \mathrm{init}_w(P_i)$. Since $Q_i = \mathrm{Conv}\big(\{0_{2k}, ie_i\}\big)$, we have $i \in \mathcal{I}_w$.

□

Remark 5.2.5. An instance of the previous theorem is when the mixing weights are all equal. In this case, there are $(2k-1)!!$ solutions up to the standard label-swapping symmetry. Monodromy techniques outlined in Section 2.2.3 exploit this symmetry for tremendous computational speed up. This technique was recently used for this problem with success [33, Section 4.1].

5.2.2 Mixed volume of λ-weighted homoscedastic models

We consider the $\lambda-$weighted homoscedastic model. In this setting the means are unknown and the variances are unknown but all equal. In this case, a k mixture model has $k+1$ unknowns. We address the high dimensional version of this problem in Section 5.3 which is also considered in recent work, for example [118].

We consider the moment system

$$f_1^k(\mu, \sigma^2, \overline{\lambda}) = 0, \ \ldots, \ f_{k+1}^k(\mu, \sigma^2, \overline{\lambda}) = 0, \tag{5.2.2}$$

where $\overline{\lambda}$ are known mixing coefficients, and f_i^k, $i \in [k+1]$ is as defined in (5.1.3). In this set-up the unknowns are μ_ℓ, $\ell \in [k]$ and σ^2.

Again, the first step is to prove that this model is algebraically identifiable.

Proposition 5.2.6. For nonzero $\overline{\lambda}_\ell, \overline{m}_i$, $\ell \in [k], i \in [k+1]$ the number of solutions to (5.2.2) is finite.

Proof. This argument is analogous to the one given in Lemma 5.2.2. Again, we consider the Jacobian of (5.2.2) with rows indexed by equations and columns indexed by the variables $\sigma^2, \mu_1, \ldots, \mu_k$. It suffices to show that for generic, $\sigma^2, \mu_1, \ldots, \mu_k$ the Jacobian, J_k, is full rank. We proceed by induction on k. When $k = 1$,

$$J_1 = \tilde{J}_1 \cdot \tilde{D}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 2\mu_1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix}$$

is full rank for $\lambda_1 \neq 0$. Now consider the $(k+1)\times(k+1)$ matrix J_k for any k. By cofactor expansion along the last column, $\det(\tilde{J}_k) = \det(\tilde{J}_{k-1})\mu_k^{k+1}$ where $\tilde{J}_{k-1}$ is the upper left $k \times k$ block of $\tilde{J}_k$.

By induction $\det(\tilde{J}_{k-1})$ is nonzero at generic $(\sigma^2, \mu_1, \ldots, \mu_{k-1})$ so $\det(\tilde{J}_k)$ is a nonzero univariate polynomial in μ_k, and by Lemma 5.2.1 it has leading coefficient $\det(\tilde{J}_{k-1})$. Therefore, $\det(\tilde{J}_k)$ is not the zero polynomial and for generic μ_k it does not vanish. This shows that $\det(\tilde{J}_k) \neq 0$ at a generic point $(\sigma^2, \mu_1, \ldots, \mu_k)$, hence the Jacobian is full rank, giving that the variety defined by (5.2.2) is zero dimensional, i.e., has finitely many solutions. □

We bound the number of solutions to (5.2.2) for a generic λ-weighted homoscedastic mixture model using mixed volumes.

Theorem 5.2.7 (Mixing coefficients known, variances equal)**.** *Consider a Gaussian k-mixture model with known and nonzero mixing coefficients λ_ℓ, $\ell \in [k]$, generic sample moments $\overline{m}_i$, $i \in [k+1]$ and equal variances. The moment system of equations* (5.2.2) *has at most $\frac{(k+1)!}{2}$ complex solutions.*

Proof. Let $P_i = \mathrm{Newt}(f_i^k)$ where f_i^k is as defined in (5.2.2) with variable ordering $(\mu_1, \ldots, \mu_k, \sigma^2)$ for $i \in [k+1]$.

Define $Q_i \subset P_i$ as follows:

$$\begin{aligned} Q_1 &= \mathrm{Conv}(\{0_{k+1}, e_1\}) \\ Q_2 &= \mathrm{Conv}(\{0_{k+1}, e_{k+1}\}) \\ Q_i &= \mathrm{Conv}(\{0_{k+1}, i \cdot e_{i-1}\}), \qquad 3 \leq i \leq k+1. \end{aligned} \tag{5.2.3}$$

where $0_{k+1} \in \mathbb{R}^{k+1}$ is the vector of all zeros and $e_i \in \mathbb{R}^{k+1}$ is the ith standard basis vector. By Example 5.1.5,

$$\mathrm{MVol}(Q_1, \ldots, Q_{k+1}) = 1 \cdot 3 \cdot 4 \cdots \cdot (k+1) = \frac{(k+1)!}{2}.$$

As in the proof of Theorem 5.2.3, we want to show the equality of mixed volumes $\mathrm{MVol}(P_1, \ldots, P_{k+1}) = \mathrm{MVol}(Q_1, \ldots, Q_{k+1})$ by using the equivalence of (1) and (2) in Proposition 5.1.7.

Let $\mathcal{I}_w$ be the set of indices such that Q_i has a vertex in $\mathrm{init}_w(P_i)$. Specifically,

$$\mathcal{I}_w = \{i \in [k+1] : Q_i \cap \mathrm{init}_w(P_i) \neq \emptyset\}.$$

By Lemma 5.2.8, $\mathcal{I}_w$ is nonempty. Now we want to show that for any $w \in \mathbb{Z}^{k+1}\backslash\{0\}$ the nonempty set of polytopes

$$\{\text{init}_w(Q_i) : i \in \mathcal{I}_w\}$$

is dependent. Since Q_i is a one dimensional line segment, it suffices to show that for any w, there exists an $i \in \mathcal{I}_w$ such that Q_i is minimized at a single vertex. We consider 2 cases.

- Suppose $2 \in \mathcal{I}_w$. If $\text{init}_w(P_2)$ is a single point, then we are done. Otherwise, assume P_2 is minimized at $Q_2 = \text{Conv}(\{0_{k+1}, e_{k+1}\})$. This means $w_{k+1} = 0$ and $w_j \geq 0$ for all $j \in [k]$, giving $\text{val}_w(P_i) = 0$ so $0 \in \text{init}_w(P_i)$ for all $i \in [k]$. This shows that $\mathcal{I}_w = [k+1]$. By Lemma 5.1.9, the collection of polytopes $\{\text{init}_w(Q_1), \ldots, \text{init}_w(Q_{k+1})\}$ is dependent.
- Now suppose $2 \notin \mathcal{I}_w$. If $1 \in \mathcal{I}_w$ and Q_1 is minimized at a single vertex then $\{\text{init}_w(Q_i) : i \in \mathcal{I}_w\}$ is dependent, so we are done. If Q_1 is not minimized at a single vertex, then $w_1 = 0$ and $w_j \geq 0$ for all $2 \leq j \leq k$. Since $2 \notin \mathcal{I}_w$, this gives that $w_{k+1} > 2w_j$ for all $j \in [k]$. Therefore, $\text{val}_w(P_j) = 0$ for all $j \in [k+1]$ which shows that $0 \in \text{init}_w(P_2)$. This contradicts $2 \notin \mathcal{I}_w$. On the other hand, if $i \in \mathcal{I}_w$ where $i \geq 3$, either Q_i is minimized at a single vertex or $w_{i-1} = 0$ and $w_j \geq 0$ for all $j \in [k+1]\backslash(i-1)$. In the latter case, this shows $\text{val}_w(P_i) = 0$ for all $i \in [k+1]$, contradicting $2 \notin \mathcal{I}_w$.

□

Lemma 5.2.8. The index set $\mathcal{I}_w$ as defined in the proof of Theorem 5.2.7 is nonempty for any nonzero $w \in \mathbb{R}^{k+1}$.

Proof. If w is in the nonnegative orthant, then w minimizes P_i at the origin for all $i \in [k+1]$. In this case $\mathcal{I}_w = [k+1] \neq \emptyset$. Now let i be the index of the smallest element of w. If $i = k+1$ then w minimizes P_2 and Q_2 at $\{e_{k+1}\}$. If $i < k+1$ then w minimizes P_{i+1} and Q_{i+1} at $\{(i+1)e_i\}$, which shows $\mathcal{I}_w$ is nonempty. □

5.2.3 Finding all solutions using homotopy continuation

To do parameter recovery for any of the set-ups described in Section 5.2.1 and Section 5.2.2, it is not enough to know the number of complex solutions to the moment equations, we need a way to find all of them. We propose using homotopy continuation methods as outlined in Section 2.2.

Recall both total degree and polyhedral homotopies outlined in Section 2.2.1. For the λ-weighted model, by Theorem 5.2.3 we have that the number of solutions to the corresponding moment system is at most $(2k-1)!!k!$ while the Bezout bound is $(2k)!$. The ratio

$$\lim_{k\to\infty} \frac{(2k-1)!!k!}{(2k)!} = 0$$

meaning for large enough k, polyhedral methods are arbitrarily better than total degree methods. Similarly, for the λ-weighted homoscedastic model Theorem 5.2.7 gives that there are at most $\frac{(k+1)!}{2}$ complex solutions to the moment equations. In this case, the Bezout bound is $(k+1)!$, so for any k, polyhedral homotopies will be twice as good as total degree homotopies.

Recall that the main drawback to polyhedral methods is that there is often a computational bottleneck associated with computing the start system. Our related approach circumvents this bottleneck and relies on the following lemma.

The collection of Newton polytopes of a polynomial system $F = (f_1, \ldots, f_n)$ is denoted by $\mathrm{Newt}(F) = (\mathrm{Newt}(f_1), \ldots, \mathrm{Newt}(f_n))$.

Lemma 5.2.9. Suppose $G(x) = 0$ is a general sparse binomial system such that $\mathrm{MVol}(\mathrm{Newt}(G)) = \mathrm{MVol}(\mathrm{Newt}(F))$ and $\mathrm{Newt}(G) \subseteq \mathrm{Newt}(F)$ element-wise. If the origin is in each Newton polytope of G, then the three assertions:

1. the solutions to $H(x, 0) = G(x) = 0$ are trivial to find,
2. there are no singularities along the path $t \in [0, 1)$, and
3. all isolated solutions of $H(x, 1) = F(x) = 0$ can be reached,

hold for the homotopy $H(x; t) = \gamma(1-t)G(x) + tF(x)$.

Proof. By Theorem 2.2.6 the number of solutions for $G(x) = 0$ equals the generic number of solutions for a polynomial system with Newton polytopes $\mathrm{Newt}(F)$. Since $\mathrm{Newt}(G) \subseteq \mathrm{Newt}(F)$ and γ is generic, we have $\mathrm{Newt}(F) = \mathrm{Newt}(\gamma(1-t)G + tF)$ for $t \in (0, 1]$. So the mixed volume, and therefore the number of solutions, of $(1-t)G + tF$ agrees with the mixed volume of $\mathrm{Newt}(F)$. □

The fact that the total degree homotopy works is a special case of the previous lemma applied to polynomials with full monomial support. Combining Lemma 5.2.9 with Theorem 5.2.3 and Theorem 5.2.7 we get the following corollary.

Corollary 5.2.10. The binomial system induced by the polytopes Q_i in the proofs of Theorem 5.2.3 and Theorem 5.2.7 constructs an optimal homotopy continuation start system for the corresponding moment system.

Proof. In this proof we construct the homotopy; give its start points; and show that it is optimal. We only show the details for the case of Theorem 5.2.3 because the other statement's proof is analogous.

Consider the binomial system

$$g_{2\ell-1} = a_\ell \mu_\ell^{2\ell-1} + b_\ell, \quad \ell \in [k]$$
$$g_{2\ell} = c_\ell (\sigma_\ell^2)^\ell + d_\ell, \quad \ell \in [k]$$

where $a_\ell, b_\ell, c_\ell, d_\ell \in \mathbb{C}^*$ are generic for $\ell \in [k]$.

Since each $g_{2\ell-1}$ and $g_{2\ell}$ is a univariate polynomial in distinct variables, multiplying the degrees we know that there are $(2k-1)!!k!$ solutions. This number agrees with the mixed volume of the respective moment system by Theorem 5.2.3. Moreover, the solutions are the start points of the homotopy

$$H(\mu, \sigma^2; t) := \begin{cases} (1-t)\gamma g_1 + t f_1^k = 0 \\ \vdots \\ (1-t)\gamma g_{2k} + t f_{2k}^k = 0. \end{cases} \tag{5.2.4}$$

where f_i^k are defined as in (5.2.1). By Lemma 5.2.9 the result follows. □

Theorem 5.2.10 bypasses the computational bottleneck associated with polyhedral homotopy methods. Therefore, the proofs of each theorem give the number of complex solutions to the corresponding variety *and* provide an efficient way to find all of them.

Example 5.2.11. Consider (5.2.1) when $k = 2$ and $\overline{\lambda} = (1/2, 1/2)$. Here we have

$$\begin{aligned}
f_1^2(\mu, \sigma^2, \overline{\lambda}) &= \frac{1}{2}\mu_1 + \frac{1}{2}\mu_2 - \overline{m}_1 \\
f_2^2(\mu, \sigma^2, \overline{\lambda}) &= \frac{1}{2}(\mu_1^2 + \sigma_1^2) + \frac{1}{2}(\mu_2^2 + \sigma_2^2) - \overline{m}_2 \\
f_3^2(\mu, \sigma^2, \overline{\lambda}) &= \frac{1}{2}(\mu_1^3 + 3\mu_1\sigma_1^2) + \frac{1}{2}(\mu_2^3 + 3\mu_2\sigma_2^2) - \overline{m}_3 \\
f_4^2(\mu, \sigma^2, \overline{\lambda}) &= \frac{1}{2}(\mu_1^4 + 6\mu_1^2\sigma_1^2 + 3\sigma_1^4) + \frac{1}{2}(\mu_2^4 + 6\mu_2^2\sigma_2^2 + 3\sigma_2^4) - \overline{m}_4
\end{aligned}$$

In this case we consider the start system:

$$\begin{aligned}
g_1 &= \mu_1 - 10, & g_2 &= \sigma_1^2 - 12, \\
g_3 &= \mu_2^3 - 27, & g_4 &= \sigma_2^4 - 4.
\end{aligned}$$

This gives six start solutions of the form $(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$:

$$\begin{array}{lll}
(10, 12, \eta \cdot 3, 2), & (10, 12, \eta^2 \cdot 3, 2), & (10, 12, 3, 2) \\
(10, 12, \eta \cdot 3, -2), & (10, 12, \eta^2 \cdot 3, -2), & (10, 12, 3, -2)
\end{array}$$

where η is a primitive third root of unity. We chose integers as the coefficients for ease of exposition. In practice, random complex numbers with norm close to one are used as the coefficients.

5.2.4 Means unknown

A final case of interest is the λ-weighted, known variance model. This is where only the means are unknown. This set-up was considered in high dimensions in [115, 114, 116].

We consider the moment system

$$f_1^k(\mu, \overline{\sigma}_1^2, \overline{\lambda}) = \cdots = f_k^k(\mu, \overline{\sigma}_k^2, \overline{\lambda}) = 0 \tag{5.2.5}$$

where $\overline{\lambda}$ are known mixing coefficients, $\overline{\sigma_\ell}^2$ is a known variance, and f_ℓ^k, $\ell \in [k]$ is as defined in (5.1.3). In this set-up, the unknowns are μ_ℓ, $\ell \in [k]$.

Theorem 5.2.12 (Means unknown)**.** *A λ-weighted, known variance Gaussian k-mixture model is algebraically identifiable using moment system* (5.2.5)*. Moreover, for generic $\overline{\lambda}_\ell, \overline{\sigma}_\ell^2, \overline{m}_\ell$, $\ell \in [k]$, the moment system* (5.2.5) *has $k!$ solutions.*

Proof. First we observe that (5.2.5) generically has finitely many solutions by the same arguments as in the proof of Lemma 5.2.2 and Proposition 5.2.6. This proves the first part of the theorem.

It follows from the Bezout bound that there are at most $k!$ complex solutions to (5.2.5). We now show that this bound is generically achieved with equality by giving parameter values where there are $k!$ solutions.

Consider $\lambda_\ell = \frac{1}{k}, \sigma_\ell^2 = 1$ and $\overline{m}_\ell = \sum_{\ell=1}^k \frac{1}{k} M_\ell(1, \ell)$ for $\ell \in [k]$. It is clear that this has a solution of the form $\mu = (1, 2, \ldots, k)$. Further, by the same induction argument involving the Jacobian of (5.2.5) referenced above, there are finitely many solutions for this set of parameters. We observe that in this case our solution set has the typical label-swapping symmetry. This shows that any action by the symmetric group on k letters, S_k, on any solution is also a solution. Therefore, there are $k!$ solutions to (5.2.5) in this case namely $\{\sigma \cdot (1, 2, \ldots, k) : \sigma \in S_k\}$. □

Corollary 5.2.13 (Equal mixture weights and variances)**.** A generic Gaussian $k-$mixture model with uniform mixing coefficients and known and equal variances is identifiable up to label-swapping symmetry using moments $m_1, \ldots, m_k$.

As a consequence of Theorem 5.2.12, when only the means are unknown the Bezout bound is equal to the BKK bound. In this case using polyhedral homotopy gives no advantage to total degree.

As discussed in Corollary 5.2.13, when the mixture weights and variances are known and equal the standard label-swapping symmetry observed with mixture models gives only one solution up to symmetry. Tracking a single path from a total degree start system, this one solution is easy to find.

Example 5.2.14. When $k = 2$, $\lambda_1 = \lambda_2 = \frac{1}{2}$ and $\sigma_1^2 = \sigma_2^2 = \sigma^2$ is a known parameter, we can symbolically solve the corresponding moment system and see that up to symmetry

$$\mu_1 = \overline{m}_1 - \sqrt{-\overline{m}_1^2 + \overline{m}_2 - \sigma^2}, \qquad \mu_2 = \overline{m}_1 + \sqrt{-\overline{m}_1^2 + \overline{m}_2 - \sigma^2}.$$

This shows that so long as $-\overline{m}_1^2 + \overline{m}_2 - \sigma^2 > 0$ we are guaranteed to get something statistically meaningful. A picture of that region in $\mathbb{R}^3$ is shown in Figure 5.5.

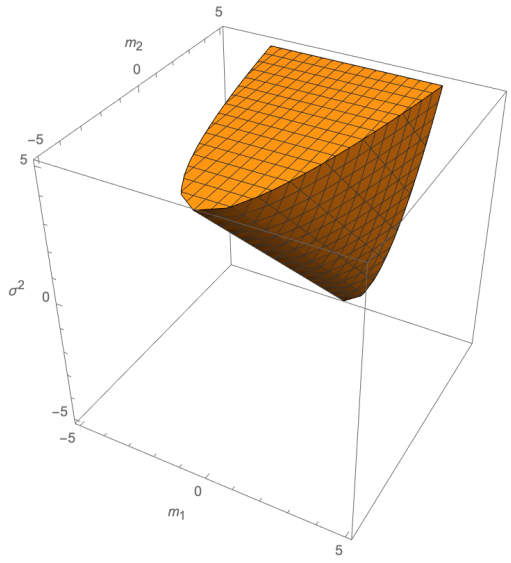


Figure 5.5: Region in the space of parameters $\overline{m}_1, \overline{m}_2, \sigma^2$ where there are statistically meaningful solutions for $k = 2$ mixture model with unknown means and $\lambda_1 = \lambda_2 = \frac{1}{2}$.

5.2.5 Mixture models with $k = 4$ components

The final result of this chapter is for a general Gaussian mixture model with $k = 4$ components. For $k = 4$, the conjectured structure of the solutions to the system of twelve variables and twelve equations according to [140] consists of 264600 complex solutions arranged in 11025 equivalence classes of size $4! = 24$. We disprove this conjecture using a numerical tool called a *trace test*. We omit details of trace tests but refer the reader to [33] for more details.

Theorem 5.2.15. *The number of solutions for a $k = 4$ mixture model is $248400 = 10350 \cdot 4!$ for generic moments $(m_0, m_1, \ldots, m_{11})$.*

Proof. Let $\bar{F} : \mathbb{C} \times \mathbb{C}^{13} \to \mathbb{C}^{13}$ be the parametric system in the unknowns $(v, \boldsymbol{\mu}, \boldsymbol{\sigma^2})$ given by

$$\bar{F}(t; v, \boldsymbol{\mu}, \boldsymbol{\sigma^2}) = \begin{cases} m_\ell - (\lambda_1 M_\ell(\mu_1, \sigma_1) + \cdots + \lambda_4 M_\ell(\mu_4, \sigma_4) & \ell = 0, \ldots 11 \\ \ell_1(v) \cdot \ell_2(g(\boldsymbol{\mu}, \boldsymbol{\sigma})) + t \end{cases}$$

with $L(v) = (m_0, m_1, \ldots, m_{11})$ where $L : \mathbb{C} \to \mathbb{C}^{12}$ is a general affine linear function and $\ell_1, \ell_2 : \mathbb{C} \to \mathbb{C}$ are general affine linear functions.

For $t = 0$, we find 31815 solutions and verify this is a complete set of solutions up to symmetry using Algorithm 2 in [33] with $\epsilon < 10^{-12}$. Of the 31815 solutions, 10350 solutions satisfy $\ell_1(v) = 0$. Since ℓ_1 is a general affine linear function, all of these solutions have the same v-coordinate, say $v^\star$. The 10350 are solutions for the moment system chosen as $L(v^\star)$. □

5.3 Density estimation for high dimensional Gaussian mixture models

Section 5.2 gives upper bounds on the number of solutions to the moment equations for Gaussian mixture models where some parameters of the model are assumed known. Using homotopy continuation algorithms we can efficiently perform density estimation in these cases. We now use our results to do density estimation on Gaussian mixture models in high dimensions.

5.3.1 High-dimensional Gaussian mixture models

A random variable $X \in \mathbb{R}^n$ is distributed as a *multivariate Gaussian* with mean $\mu \in \mathbb{R}^n$ and symmetric positive definite covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$, if it has density

$$f_X(x_1, \ldots, x_n | \mu, \Sigma) = ((2\pi)^n \det(\Sigma))^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right).$$

We denote $X \sim \mathcal{N}(\mu, \Sigma)$.

A random variable is distributed as the *mixture of k multivariate Gaussians* if it is a convex combination of k multivariate Gaussian densities. It has probability density function

$$f_X(x_1, \ldots, x_n | \lambda_\ell, \mu_\ell, \Sigma_\ell)_{\ell=1,\ldots,k} = \sum_{\ell=1}^{k} \lambda_\ell f_{X_\ell}(x_1, \ldots, x_n | \mu_\ell, \Sigma_\ell)$$

where $(\lambda_1, \ldots, \lambda_k) \in \Delta_{k-1}$, $\mu_\ell \in \mathbb{R}^n$, and $\Sigma_\ell \in \mathbb{R}^{n \times n}$ is symmetric and positive definite for $\ell \in [k]$. Here we write $X \sim \sum_{\ell=1}^k \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)$.

Let $f_X : \mathbb{R}^n \to \mathbb{R}$ be the probability density function of a random vector $X = (X_1, \ldots, X_n)$. The $(i_1, \ldots, i_n)-$*th moment* of X is

$$m_{i_1,\ldots,i_n} = \mathbb{E}[X_1^{i_1} \cdots X_n^{i_n}] = \int_{\mathbb{R}^n} x_1^{i_1} \cdots x_n^{i_n} f_X(x_1, \ldots, x_n) dx_1 \cdots dx_n$$

where $i_s \geq 0$ for all $s \in [n]$. The non-negative integer $i_1 + \ldots + i_n = d$ is called the *order* of $m_{i_1,\ldots,i_n}$.

We can get the explicit polynomials $m_{i_1,\ldots,i_n}$ of a Gaussian mixture using the moment generating function. This gives the identity:

$$\sum_{i_1,\ldots,i_n \geq 0} \frac{m_{i_1,\ldots,i_n}}{i_1! \cdots i_n!} t_1^{i_1} \cdots t_n^{i_n} = \sum_{\ell=1}^k \lambda_\ell \exp(t_1 \mu_{\ell 1} + \ldots t_n \mu_{\ell n}) \cdot \exp(\frac{1}{2} \sum_{i,j=1}^n \sigma_{\ell ij} t_i t_j). \tag{5.3.1}$$

Using Taylor's formula, we can expand the left side of (5.3.1) and equate coefficients of each side to get $m_{i_1,\ldots,i_n}$. Note that $m_{0,\ldots,0} = 1$.

Example 5.3.1. Consider a Gaussian mixture model with $k = 2$ components in $\mathbb{R}^2$. Here we have $X \sim \lambda_1 \mathcal{N}(\mu_1, \Sigma_1) + \lambda_2 \mathcal{N}(\mu_2, \Sigma_2)$ where

$$\mu_1 = \begin{pmatrix} \mu_{11} \\ \mu_{12} \end{pmatrix}, \qquad \Sigma_1 = \begin{pmatrix} \sigma_{111} & \sigma_{112} \\ \sigma_{112} & \sigma_{122} \end{pmatrix},$$
$$\mu_2 = \begin{pmatrix} \mu_{21} \\ \mu_{21} \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} \sigma_{211} & \sigma_{212} \\ \sigma_{212} & \sigma_{222} \end{pmatrix}.$$

Observe that our convention is to use the first index to identify the component of the mixture. The moments up to order three are

$$
\begin{aligned}
m_{00} &= \lambda_1 + \lambda_2 \\
m_{10} &= \lambda_1\mu_{11} + \lambda_2\mu_{21} \\
m_{01} &= \lambda_1\mu_{12} + \lambda_2\mu_{22} \\
m_{20} &= \lambda_1(\mu_{11}^2 + \sigma_{111}) + \lambda_2(\mu_{21}^2 + \sigma_{211}) \\
m_{11} &= \lambda_1(\mu_{11}\mu_{12} + \sigma_{112}) + \lambda_2(\mu_{21}\mu_{22} + \sigma_{212}) \\
m_{02} &= \lambda_1(\mu_{12}^2 + \sigma_{122}) + \lambda_2(\mu_{22}^2 + \sigma_{222}) \\
m_{30} &= \lambda_1(\mu_{11}^3 + 3\mu_{11}\sigma_{111}) + \lambda_2(\mu_{21}^3 + 3\mu_{21}\sigma_{211}) \\
m_{21} &= \lambda_1(\mu_{11}^2\mu_{12} + 2\mu_{11}\sigma_{112} + \mu_{12}\sigma_{111}) + \lambda_2(\mu_{21}^2\mu_{22} + 2\mu_{21}\sigma_{212} + \mu_{22}\sigma_{211}) \\
m_{12} &= \lambda_1(\mu_{11}\mu_{12}^2 + \mu_{11}\sigma_{122} + 2\mu_{12}\sigma_{112}) + \lambda_2(\mu_{21}\mu_{22}^2 + \mu_{21}\sigma_{222} + 2\mu_{22}\sigma_{212}) \\
m_{03} &= \lambda_1(\mu_{12}^3 + 3\mu_{12}\sigma_{122}) + \lambda_2(\mu_{22}^3 + 3\mu_{22}\sigma_{222}).
\end{aligned}
\tag{5.3.2}
$$

The $m_{0,0,\ldots,0,i_s,0,\ldots 0}-$th moment is the same as the i_s-th order moment for the univariate Gaussian mixture model $\sum_{\ell=1}^{k} \lambda_\ell \mathcal{N}(\mu_{\ell s}, \sigma_{\ell ss})$. This observation is key to our proposed density estimation algorithm in Section 5.3.2 and it follows from the property that marginal distributions of a Gaussian are Gaussian themselves.

5.3.2 Dimension reduction and recovery algorithm

We propose an algorithm for density estimation of multivariate Gaussian densities using the method of moments. The main idea is that if we use higher order moment equations, density estimation for multivariate Gaussian mixture models reduces to multiple instances of density estimation for univariate Gaussian mixture models. The algorithm is described in Algorithm 3.

Remark 5.3.2. There are many possible choices for the input $\overline{\mathfrak{m}}_2$ and such a choice exists by Isserlis' Theorem [141]. To avoid cumbersome and unnecessary notation for the reader, we do not explicitly list all such options. Our personal preference is to only use second and third order moments.

Example 5.3.3. Suppose $X \sim \lambda_1\mathcal{N}(\mu_1, \Sigma_1) + \lambda_2\mathcal{N}(\mu_2, \Sigma_2)$ as in Example 5.3.1.

Algorithm 3 Density Estimation for Mixtures of Multivariate Gaussians

Input: The set of sample moments:

$$\overline{\mathfrak{m}}_1 := \{\overline{m}_{e_1}, \overline{m}_{2e_1}, \ldots, \overline{m}_{(3k)e_1}, \overline{m}_{e_i}, \ldots, \overline{m}_{(2k+1)e_i} : 2 \leq i \leq n\}$$
$$\overline{\mathfrak{m}}_2 := \{\overline{m}_{a_1}, \ldots, \overline{m}_{a_N} : a_j \in \mathbb{N}^n\}$$

that are the moments to the multivariate Gaussian mixture model:

$$\lambda_1 \mathcal{N}(\mu_1, \Sigma_1) + \cdots + \lambda_k \mathcal{N}(\mu_k, \Sigma_k)$$

where $N = \frac{k}{2}(n^2 - n)$ and $\overline{\mathfrak{m}}_2$ is any set of sample moments with polynomials where the off-diagonal entries of Σ_ℓ are linear for $\ell \in [k]$.

Output: Parameters $\lambda_\ell \in \mathbb{R}$, $\mu_\ell \in \mathbb{R}^n$, $\Sigma_\ell \succ 0$ for $\ell \in [k]$ such that $\overline{\mathfrak{m}}_1, \overline{\mathfrak{m}}_2$ are the moments of distribution $\sum_{\ell=1}^k \lambda_\ell \mathcal{N}(\mu_\ell, \Sigma_\ell)$.

1. Solve the general univariate case using sample moments $\{\overline{m}_{e_1}, \overline{m}_{2e_1}, \ldots, \overline{m}_{(3k-1)e_1}\}$ to get parameters λ_ℓ, $\mu_{\ell,1}$ and $\sigma_{\ell,1,1}$.
2. Select the statistically meaningful solution with sample moment $\overline{m}_{3ke_1}$.
3. Using the mixing coefficients λ_ℓ and sample moments $\{\overline{m}_{e_i}, \ldots, \overline{m}_{2ke_i}\}$, solve (5.2.1) $n-1$ times to obtain $\mu_{\ell i}$ and $\sigma_{\ell ii}$ for $\ell \in [k]$, $2 \leq i \leq n$
4. Select the statistically meaningful solution with sample moment $\overline{m}_{(2k+1)e_i}$ for $2 \leq i \leq n$.
5. Using $\overline{\mathfrak{m}}_2$, solve the remaining system of N linear equations in N unknowns
6. Return $\lambda_\ell, \mu_\ell, \Sigma_\ell$, $\ell \in [k]$.

The first steps of Algorithm 3 use the sample moments $\overline{\mathfrak{m}}_1$. Here we use the input

$$[\overline{m}_{10}, \overline{m}_{20}, \overline{m}_{30}, \overline{m}_{40}, \overline{m}_{50}, \overline{m}_{60}] = [-0.25,\ 2.75,\ -1.0,\ 22.75,\ -6.5,\ 322.75]$$
$$[\overline{m}_{01}, \overline{m}_{02}, \overline{m}_{03}, \overline{m}_{04}, \overline{m}_{05}] = [2.5,\ 16.125,\ 74.5,\ 490.5625,\ 2921.25].$$

In Step 1 we solve the general univariate case to obtain $\lambda_\ell, \mu_{\ell 1}, \sigma^2_{\ell 11}$ for $\ell = 1, 2$.

We find up to symmetry that there are two statistically meaningful solutions:

$$(\lambda_1, \lambda_2, \mu_{11}, \mu_{21}, \sigma^2_{111}, \sigma^2_{211}) = (0.25,\ 0.75,\ 0, -1,\ 3,\ 1)$$
$$(\lambda_1, \lambda_2, \mu_{11}, \mu_{21}, \sigma^2_{111}, \sigma^2_{211}) = (0.967,\ 0.033,\ -0.378,\ 3.493,\ 2.272,\ 0.396).$$

The first solution has $m_{60} = 322.75$ and the second has $m_{60} = 294.686$ so in Step 2 we select the first solution.

Using $\lambda_1 = 0.25$, $\lambda_2 = 0.75$ we solve

$$\begin{aligned}
2.5 &= 0.25 \cdot \mu_{12} + 0.75 \cdot \mu_{22} \\
16.125 &= 0.25 \cdot (\mu^2_{12} + \sigma^2_{122}) + 0.75 \cdot (\mu^2_{22} + \sigma^2_{222}) \\
74.5 &= 0.25 \cdot (\mu^3_{12} + 3\mu_{12}\sigma^2_{122}) + 0.75 \cdot (\mu^3_{22} + 3\mu_{22}\sigma^2_{222}) \\
490.5625 &= 0.25 \cdot (\mu^4_{12} + 6\mu^2_{12}\sigma^2_{122} + 3\sigma^4_{122}) + 0.75 \cdot (\mu^4_{22} + 6\mu^2_{22}\sigma^2_{222} + 3\sigma^4_{222}),
\end{aligned}$$

and find there is one statistically meaningful solution:

$$(\mu_{12}, \mu_{22}, \sigma^2_{122}, \sigma^2_{222}) = (-2,\ 4,\ 2,\ 3.5\ ).$$

To recover the off-diagonal entries of Σ_1, Σ_2, we use sample moments

$$\overline{\mathfrak{m}}_2 = [\overline{m}_{11}, \overline{m}_{21}] = [0.8125,\ 7.75]$$

then solve the linear system

$$\begin{aligned}
0.8125 &= 0.25 \cdot (2 + \sigma_{112}) + 0.75 \cdot \sigma_{212} \\
7.75 &= 0.25 \cdot (-4 - 2 \cdot \sigma_{112}) + 9
\end{aligned}$$

to find $(\sigma_{112}, \sigma_{212}) = (0.5,\ 0.25)$. We estimate that our samples came from density

$$0.25 \cdot \mathcal{N}\Big(\begin{bmatrix} -1 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix} \Big) + 0.75 \cdot \mathcal{N}\Big(\begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 3 & 0.25 \\ 0.25 & 3.5 \end{bmatrix} \Big).$$

The clear benefit to Algorithm 3 is that it avoids the curse of dimensionality. We observe that Step 1 is the most computationally prohibitive step since density estimation for univariate Gaussian mixture models with many components is difficult.

Currently, the only known way to solve Step 1 is by finding all complex solutions then selecting the statistically meaningful one closest to the next sample moment. It has been shown, in [133, 140, 33] respectively, that for a mixture of $k = 2, 3$ and 4 Gaussian densities there are $9 \cdot 2!$, $225 \cdot 3!$ and $10350 \cdot 4!$ complex solutions. Since the number of complex solutions is known in each of these cases, state of the art polynomial system solvers can exploit the label swapping symmetry to find all solutions quickly.

Also, unlike varying n, which just changes the dimension of the problem, changing k gives a new class of problems — for example, a one-mixture is qualitatively different from a two-mixture. Our algorithm hints that every k deserves its own independent analysis.

Remark 5.3.4. Using the homotopy continuation methods discussed in Section 5.2.3, we can efficiently solve Step 3 of Algorithm 3. Using these continuation methods for fixed k, the number of homotopy paths we need to track is $\mathcal{O}(n)$. One observation for implementation is that Steps 3 and 4 can be performed in parallel.

Now we establish preliminary results for the proof of correctness of Algorithm 3.

Lemma 5.3.5. A generic one dimensional Gaussian k mixture model for $k \leq 4$ is uniquely identifiable up to symmetry using moments $m_0, \ldots, m_{3k}$.

Proof. When $k = 1$ the result is trivial. Lazard proved the result when $k = 2$ [142]. For $k = 3, 4$, we observe this statement is equivalent to saying the polynomial system

$$m_0 = 1, m_i = \overline{m}_i, i \in [3k]$$

where $\overline{m}_{3k}$ is considered as a variable has solutions where the $\overline{m}_{3k}$ coordinate has multiplicity $k!$. Since multiplicity is generic behavior, we find all solutions to this polynomial system when $k = 3, 4$ for generic $\overline{m}_1, \ldots, \overline{m}_{3k-1} \in \mathbb{C}$ and verify the result numerically using `HomotopyContinuation.jl` [29] and the certificate described in [143]. □

We conjecture that Lemma 5.3.5 is true for all k. It has been shown that every Gaussian mixture model is uniquely identifiable by the first $4k-2$ moments [129] and that if all means are equal, this bound is tight [144]. A generic Gaussian mixture model does not have equal means, which distinguishes the conjectured generic bound of $3k$ moments from the proven upper bound of $4k-2$.

We consider a lemma similar to Lemma 5.3.5.

Lemma 5.3.6. A generic Gaussian k mixture model for $k \leq 4$ with known mixing coefficients is uniquely identifiable up to symmetry using moments $m_1, \ldots, m_{2k+1}$.

Proof. When $k = 1$ the result is trivial. When $k = 2, 3, 4$ we provably find all solutions using Theorem 5.2.3 then compute the multiplicity of the coordinate $\overline{m}_{2k+1}$ as in Lemma 5.3.5. □

Theorem 5.3.7. *For a generic Gaussian k mixture model with $k \leq 4$ in $\mathbb{R}^n$,*

$$\lambda_1 \mathcal{N}(\mu_1, \Sigma_1) + \cdots + \lambda_k \mathcal{N}(\mu_k, \Sigma_k)$$

with sets of moments $\overline{\mathfrak{m}}_1$ and $\overline{\mathfrak{m}}_2$, Algorithm 3 is guaranteed to recover the parameters $\lambda_\ell, \mu_\ell, \Sigma_\ell, \ell \in [k]$.

Proof. The proof of correctness follows from Lemma 5.3.5 and Lemma 5.3.6. □

The philosophy behind Algorithm 3 extends to any k. The main bottleneck is that state of the art numerical methods to solve Step 1 by finding all complex solutions are exhausted for $k \geq 5$. The benefit of finding all complex solutions is that we are guaranteed to find all of the statistically meaningful ones. While the number of complex solutions to the moment equations seems to grow exponentially with k, our computational results (discussed in Section 5.4) show that typically there are few statistically meaningful ones. This motivates future work to efficiently find only the statistically meaningful solutions.

Remark 5.3.8. In Step 2 and Step 4, strict identifiability results are needed to guarantee correctness. This is already settled if one identifies the desired statistically meaningful solution from sample moments $\overline{m}_{3k}, \ldots, \overline{m}_{4k-2}$ instead of just $\overline{m}_{3k}$ as done in Algorithm 3.

Remark 5.3.9. Although Algorithm 3 uses higher order moments, this order doesn't exceed what we would need for the univariate case. In addition, it has been shown that there exist algebraic dependencies among lower order moment equations complicating the choice of which moment system to consider [130].

5.3.3 Uniform mixtures with equal variances

We consider the special case of estimating the mixture of k Gaussians in $\mathbb{R}^n$ where all of the means $\mu_i \in \mathbb{R}^n$ are unknown, each $\lambda_i = \frac{1}{k}$ and each covariance matrix $\Sigma_i \in \mathbb{R}^{n\times n}$ is equal and known. This is the in [115, 114, 117, 116].

The fundamentals of Algorithm 3 can be applied with even greater success. Recall from Corollary 5.2.13 that in one dimension, there is generically a unique solution up to symmetry to (5.2.5) that can be found efficiently. We use this fact in Step 1 of Algorithm 4. In Step 2 there are other choices for sample moments that will give a square linear system. Some of these choices involve sample moments of lower order.

Observe that Step 1 of Algorithm 4 requires *tracking a single homotopy path*. This is in contrast to Step 1 of Algorithm 3 in which one needs to track many homotopy paths to obtain all complex solutions. Further, Algorithm 4 requires solving a $k \times k$ linear system $n-1$ times (Step 2). This is again in contrast to Algorithm 3 where one needs to solve a nonlinear polynomial system that tracks $(2k-1)!!k!$ paths $n-1$ times (Step 3). In both cases, we see that we need to solve n polynomial systems, where n is the dimension of the Gaussian mixture model.

If we consider the tracking of a single homotopy path as unit cost, we consider the number of paths tracked as the complexity of the algorithm (as is customary in numerical algebraic geometry). With this choice, Algorithm 4 tracks n homotopy paths, while Algorithm 3 tracks $(2k-1)!!k!(n-1)+N_k$ paths where N_k is the number of homotopy paths needed to complete Step 1 in Algorithm 3. This highlights how Algorithm 4 is highly efficient and the method of moments up can be effective for large n and k.

Algorithm 4 Density Estimation for Uniform Mixtures of Multivariate Gaussians with Equal Covariances

Input: The set of sample moments:

$$\overline{\mathfrak{m}}_1 := \{\overline{m}_{e_1}, \ldots, \overline{m}_{ke_1}\}$$
$$\overline{\mathfrak{m}}_i := \{\overline{m}_{e_i}, \overline{m}_{e_1+e_i}, \overline{m}_{2e_1+e_i}, \ldots, \overline{m}_{(k-1)e_1+e_i}\}, \quad 2 \leq i \leq n$$

that are the moments to multivariate Gaussian mixture model:

$$\frac{1}{k}\mathcal{N}(\mu_1, \overline{\Sigma}) + \cdots + \frac{1}{k}\mathcal{N}(\mu_k, \overline{\Sigma}).$$

Output: Parameters $\mu_\ell \in \mathbb{R}^n$, such that $\overline{\mathfrak{m}}_i$, $i \in [n]$, are the moments of distribution $\sum_{\ell=1}^k \frac{1}{k}\mathcal{N}(\mu_\ell, \overline{\Sigma})$

1. Using mixing coefficients $\lambda_\ell = \frac{1}{k}$ for $\ell \in [k]$ and sample moments $\overline{\mathfrak{m}}_1$ solve (5.2.5) to obtain $\mu_{\ell 1} \in \mathbb{R}$.

2. Using sample moments $\overline{\mathfrak{m}}_i$ solve the $k \times k$ linear system in $\mu_{i1}, \ldots, \mu_{ik}$ for $2 \leq i \leq n$.

5.4 Computational results

We perform numerical experiments by running Algorithm 3 on randomly generated Gaussian mixture models with diagonal covariance matrices. We use `HomotopyContinuation.jl` to do all of the polynomial system solving [29]. The average running time and error for $k = 2$ are given in Table 5.1 and for $k = 3$ in Table 5.2. Overall, we see that the error incurred from doing homotopy continuation is negligible. In addition, we see that we don't suffer from any other numerical errors associated with homotopy continuation, such as path jumping.

In addition we run simulations on the number of statistically meaningful solutions for a k mixture model in one dimension. We find that if we choose random, real valued sample moments $\overline{m}_i$, $i \in [3k-1]$ then there are generically no statistically meaningful solutions. Instead we generate a set of sample moments by computing the moments of a generic Gaussian mixture model, then

n	10	100	1,000	10,000	100,000
Time (s)	0.17	0.71	6.17	62.05	650.96
Error	7.8×10^{-15}	4.1×10^{-13}	5.7×10^{-13}	2.9×10^{-11}	1.8×10^{-9}
Normalized Error	1.9×10^{-16}	1.0×10^{-15}	1.4×10^{-16}	7.3×10^{-16}	4.5×10^{-15}

Table 5.1: Average running time and numerical error running Algorithm 3 on a mixture of 2 Gaussians in $\mathbb{R}^n$. The error is $\epsilon = \|v - \hat{v}\|_2$ where $v \in \mathbb{R}^{4n+2}$ is a vector of the true parameters and $\hat{v}$ is a vector of the estimates. The normalized error is $\epsilon/(4n+2)$.

n	10	100	1,000	10,000	100,000
Time (s)	4.71	10.87	73.74	845.55	8291.84
Error	3.6×10^{-13}	4.6×10^{-12}	1.3×10^{-10}	4.6×10^{-10}	9.6×10^{-9}
Normalized Error	1.1×10^{-14}	1.5×10^{-14}	4.2×10^{-14}	1.5×10^{-14}	3.2×10^{-14}

Table 5.2: Average running time and numerical error running Algorithm 3 on a mixture of 3 Gaussians in $\mathbb{R}^n$. The error is $\epsilon = \|v - \hat{v}\|_2$ where $v \in \mathbb{R}^{6n+3}$ is a vector of the true parameters and $\hat{v}$ is a vector of the estimates. The normalized error is $\epsilon/(6n+3)$.

finding all other statistically meaningful solutions. We generate a generic Gaussian k-mixture model as follows: Let $X_i, Y_i, Z_i \sim \mathcal{N}(0,1)$ for $i \in [k]$ be independent random variables. Then $\lambda_i = \frac{|X_i|}{\sum_{i=1}^{k} |X_i|}$, $\mu_i = Y_i$ and $\sigma_i^2 = |Z_i|$ for $i \in [k]$.

The results of these simulations is given in Table 5.3. We run each simulation $10,000$ times and the results are given up to the label swapping symmetry.

k	2	3	4
Average number of $\mathbb{R}$ solutions	2.93	4.34	13.17
Maximum number of $\mathbb{R}$ solutions	5	17	55
Average number of statistically meaningful solutions	1.51	1.77	2.81
Maximum number of statistically meaningful solutions	2	5	15

Table 5.3: Average and maximum number of real and statistically meaningful solutions (up to label-swapping symmetry) for generic Gaussian k-mixture models.

Chapter 6

The maximum likelihood degree of sparse polynomial sytems

6.1 Introduction

Maximum likelihood estimation is a statistical method of density estimation that seeks to maximize the probability that a given set of samples comes from a distribution. Given independent and identically distributed (iid) samples $s^{(1)}, \ldots, s^{(N)}$ we can form a *data vector* $u \in \triangle_{n-1} := \{p \in \mathbb{R}^n_{>0} : \sum_{i=1}^n p_i = 1\}$ which counts the fraction of times each event happened in the sample set $s^{(1)}, \ldots, s^{(N)}$.

Given u, the *log likelihood function* for a discrete random variable is given by

$$\log(p_1^{u_1} \cdots p_n^{u_n}) = u_1 \log(p_1) + \ldots + u_n \log(p_n).$$

Maximum likelihood estimation aims to select the set of points $p \in \triangle_{n-1}$ that maximizes the likelihood that u came from that distribution. In many instances, we assume that our density p lives in a *statistical model* $\mathcal{M} \subseteq \triangle_{n-1}$. In this set-up, maximum likelihood estimation amounts to solving the (often) nonconvex optimization problem

$$\max_p \; u_1 \log(p_1) + \ldots + u_n \log(p_n) \quad \text{subject to} \quad p \in \mathcal{M}.$$

This is the primary problem under consideration. While nonconvex optimization is often much more challenging than its convex counterpart, methods exist to tackle this problem. We consider the set-up where $\mathcal{M}$ is defined by a set of polynomial equations and use tools from algebraic geometry to study the critical points of this optimization problem. This problem has been studied from several points of view. An algebraic geometry approach and definition of maximum likelihood (ML) degree were made in [145, 146]. The results in [147] show that the ML degree of a smooth

variety equals a signed Euler characteristic, and in the case of a hypersurface, that the ML degree equals a signed volume of a Newton polytope. For the singular case, formulas for the ML Degree are given by the Euler obstruction function [148]. ML degrees also make an appearance in toric geometry [149, 150] and are studied for other statistical models [151, 152, 153].

Specifically, we consider when $\mathcal{M}$ is given by the variety of a system of *sparse polynomial equations*. Sparse polynomials have been studied in several contexts [154, 24]. A good introduction to this material is [155, Chapter 3]. Following similar conventions as those in [156], we specify a family of sparse polynomials by its monomial support using the following notation. For each $\alpha = (\alpha_1, ..., \alpha_n) \in \mathbb{N}^n$, the monomial $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ with exponent α is the map $x^\alpha : \mathbb{C}^n \to \mathbb{C}$. A sparse polynomial is a linear combination of monomials. Let $\mathcal{A}_\bullet = (\mathcal{A}_1, \ldots, \mathcal{A}_k)$ denote a k-tuple of nonempty finite subsets of $\mathbb{N}^n$. A general sparse polynomial system of equations with support $\mathcal{A}_\bullet$ is given by

$$\sum_{\alpha \in \mathcal{A}_1} c_{1,\alpha} x^\alpha = \ldots = \sum_{\alpha \in \mathcal{A}_k} c_{k,\alpha} x^\alpha = 0,$$

where the coefficients $\{c_{i,\alpha}\}_{\alpha \in \mathcal{A}_i, i \in [k]}$ are general.

Related work has considered a similar optimization problem

$$\min_{x \in \mathbb{R}^n} \; g(x) \quad \text{subject to} \quad x \in \mathcal{X} \tag{6.1.1}$$

where $\mathcal{X}$ is a real algebraic variety and g is a specified objective function. A particular choice of g that is of interest is when $g = \|x - u\|_2^2$ for a point $u \in \mathbb{R}^n$. This is called the *Euclidean distance function* and the number of critical points to this optimization problem for general u is called the *ED degree of* $\mathcal{X}$. The study of ED degrees began with [157] and initial bounds on the ED degree of a variety were given in [158]. Other work has found the ED degree for real algebraic groups [159], Fermat hypersurfaces [160], orthogonally invariant matrices [161], smooth complex projective varieties [162], the multiview variety [163], and when $\mathcal{X}$ is a hypersurface [164]. Further work has considered instances of this problem when the data u are not general [165] as well as when the semidefinite relaxation is tight [166].

A final connection is when the objective function in (6.1.1) is a polynomial. In this case, the number of critical points is called the *algebraic degree of the optimization problem.* In [167], the

algebraic degree of (6.1.1) is considered when $\mathcal{X} = V(f_1, \ldots, f_k)$ with f_i and g are all generic polynomials of some degree. By [167, Proposition 2.1] the number of solutions $(x^*, \lambda_1^*, \ldots, \lambda_k^*) \in \mathbb{C}^{n+k}$ to the the *Karush-Kuhn-Tucker (KKT)-system*

$$\nabla g(x^*) + \sum_{i=1}^{k} \lambda_i^* \nabla f_i(x^*) = 0$$
$$f_1(x^*) = \cdots = f_k(x^*) = 0$$

is the algebraic degree. Moreover, a formula for this degree is given in [167, Theorem 2.2] in terms of the degrees of g and $f_1, \ldots, f_k$. Other formulas for many classes of convex polynomial optimization problems are given in [168] and [169]. Related topics and background on algebraic optimization problems and the corresponding convex geometry can be found in [170].

6.2 The ML degree of sparse systems

Let $F : \mathbb{R}^n \to \mathbb{R}^k$ be a sparse polynomial system with general coefficients. Let $u \in \mathbb{R}^n$ be a general point. Here u is the data and $F = \langle f_1, \ldots, f_k \rangle$ is the model. We want to solve the maximum likelihood optimization problem:

$$\sup_{x \in \mathbb{R}^n_{>0}} \sum_{i=1}^{n} u_i \log(x_i) \qquad \text{subject to} \qquad x \in \mathcal{V}(F). \tag{MLE}$$

One approach to solving (MLE) is to find all critical points which can be done using Lagrange multipliers. The *Lagrangian function* for (MLE) is defined as

$$\Lambda(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) := \sum_{i=1}^{n} u_i \log(x_i) - \sum_{i=1}^{k} \lambda_i f_i. \tag{6.2.1}$$

To find all critical points of (MLE) we solve the square polynomial system $\mathcal{L} : \mathbb{R}^{n+k} \to \mathbb{R}^{n+k}$ obtained by taking the partial derivatives of Λ. The partial derivatives are

$$\frac{\partial}{\partial x_i}\Lambda = \frac{u_i}{x_i} - \frac{\partial}{\partial x_i}\Big(\sum_{j=1}^{k} \lambda_j f_j\Big), \qquad i \in [n] \tag{6.2.2}$$
$$\frac{\partial}{\partial \lambda_j}\Lambda = -f_j, \qquad j \in [k]. \tag{6.2.3}$$

Multiplying $\frac{\partial}{\partial x_i}\Lambda$ by x_i clears the denominators to get the polynomials

$$\ell_i := x_i \cdot \frac{\partial}{\partial x_i}\Lambda = u_i - x_i \sum_{j=1}^{k} \lambda_j \frac{\partial}{\partial x_i}(f_j), \quad i \in [n]. \tag{6.2.4}$$

Using the notation in (6.2.3)-(6.2.4), the *ML system* of F is

$$\mathcal{L}(F) = \langle \ell_1, \ldots, \ell_n, f_1, \ldots, f_k \rangle. \tag{6.2.5}$$

In the literature, the ML system is also known as the Lagrange likelihood equations. We use the former terminology for brevity.

The critical points to (MLE) are given by the real solutions to $\mathcal{L}(F)$ with positive x-coordinates. It is often more convenient to work over algebraically closed fields because the number of complex solutions to $\mathcal{L}(F)$ is constant over a dense Zariski open subset of the parameter space. This number is called the *ML degree* of F.

Remark 6.2.1 (Sum-to-one-constraint)**.** Typically $f_1 \in F$ will be $x_1 + \ldots + x_n - 1$. Although this polynomial does not have general coefficients, we can rescale the variables so the traditional MLE situation falls into our set-up.

The following proposition shows that the ML degree of a sparse polynomial system is well defined.

Proposition 6.2.2. For a general sparse polynomial system $F = \langle f_1, \ldots, f_k \rangle$ and for generic data u, the corresponding ML system has finitely many solutions in $\mathbb{C}^n \times \mathbb{C}^k$. Moreover, all solutions to the ML system are in $(\mathbb{C}^*)^n \times (\mathbb{C}^*)^k$.

Proof. This proof uses genericity in two different ways. First we use genericity of the coefficients of $f_1, \ldots, f_k$. By Bertini's Theorem [139, Ch. III,§10.9.2]., the variety of $\langle f_1, \ldots, f_k \rangle$ saturated by the coordinate hyperplanes is either empty or codimension k. Denote this variety by X. Moreover, by Bertini's Theorem, if $k < n$, then the variety X is irreducible.

The polynomials $\ell_1, \ldots, \ell_n$ give a map $X \times \mathbb{C}^k_\lambda \to \mathbb{C}^n_u$. The source of this map is n-dimensional and irreducible and therefore the image is at most n-dimensional.

Now we use genericity of the data. If the image is n dimensional, then a fiber over a generic point is zero dimensional. This means the ML system for generic data has finitely many solutions. On the other hand, if the image is lower dimensional, then the fiber over a generic point is empty. In such a case the ML degree is zero.

Since X is defined by saturating by the coordinate hyperplanes, we must show that there are still only finitely many solutions to the ML System in $(\mathbb{C}^n \setminus (\mathbb{C}^*)^n) \times \mathbb{C}^k$. By the data being generic, we may assume the u_i coordinate is nonzero. For $i = 1, \ldots, n$, having $u_i \neq 0$ and $\ell_i = 0$ implies that the x_i coordinate of the solution is not zero. Therefore all solutions to the ML system are in $(\mathbb{C}^*)^n \times \mathbb{C}^k$.

We have shown the first statement and part of the second statement. It remains to show that there are no solutions with $\lambda_i = 0$ for $i \in [k]$. If we assume $\lambda_k^* = 0$ by way of contradiction, then $(x_1^*, \ldots, x_n^*, \lambda_1^*, \ldots, \lambda_{k-1}^*)$ is a solution to the ML system of $f_1, \ldots, f_{k-1}$. By the argument above, this new ML system has finitely many solutions. By the genericity of f_k, none of these solutions will satisfy $f_k(x^*) = 0$. □

We remark that the arguments used in the first half of the proof are analogous to the ones presented in [146, Proposition 3].

6.2.1 Newton polytopes of likelihood equations and the algebraic torus

We want to use existing results on sparse polynomial systems from algebraic geometry. To do this, recall the definition of Newton polytopes and initial systems. For simplicity, we denote $\|(f)$ to be the set of vertices of $\mathrm{Newt}(f)$.

The next lemma describes the Newton polytopes of the ML system (6.2.5). We use the notation $x_1 \cdots x_n \mid f$ when there exists a polynomial g such that $x_1 \cdots x_n \cdot g = f$.

Lemma 6.2.3. Consider a sparse polynomial system $F = \langle f_1, \ldots, f_k \rangle$. If $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$, then for every $i \in [n]$ and $j \in [k]$, $\mathrm{Newt}(f_j) = \mathrm{Newt}(x_i \frac{\partial}{\partial x_i} f_j)$. Moreover, for every $i \in [n]$ the Newton polytope of ℓ_i is equal to

$$\mathrm{Newt}(\ell_i) = \mathrm{Conv}(\{0_{n+k}\} \cup \mathrm{Vert}(\lambda_1 f_1) \cup \cdots \cup \mathrm{Vert}(\lambda_k f_k)).$$

Proof. The proof of the first statement follows from the fact that a Newton polytope is determined by its vertices, and that

$$\frac{\partial}{\partial x_i}\left(x_1^{\alpha_1}\cdots x_n^{\alpha_n}\right) = \begin{cases} 0 & \alpha_i = 0 \\ \alpha_i \frac{x_1^{\alpha_1}\cdots x_n^{\alpha_n}}{x_i} & \text{otherwise.} \end{cases}$$

The proof of the second statement follows from the the definition of the likelihood equations and Newton polytopes.

□

The following example provides an intuitive description of Lemma 6.2.3.

Example 6.2.4. Let $F = f = \langle 2x^4 + 3y^3 - 5\rangle$, and consider variable ordering (x, y, λ). Then the Newton polytopes given by $\mathcal{L}(f)$ are

$$\begin{aligned} \text{Newt}(f) &= \text{Conv}(\{(4,0,0),(0,3,0),(0,0,0)\}), \\ \text{Newt}(\ell_1) &= \text{Conv}(\{(0,0,0),(4,0,1)\}), \text{ and} \\ \text{Newt}(\ell_2) &= \text{Conv}(\{(0,0,0),(0,3,1)\}). \end{aligned}$$

These are different from the Newton polytopes coming from $\mathcal{L}(\hat{f})$, where $\hat{f} = \langle xyf\rangle$:

$$\begin{aligned} \text{Newt}(\hat{f}) &= \text{Conv}(\{(5,1,0),(1,4,0),(1,1,0)\}), \text{ and} \\ \text{Newt}(\hat{\ell_1}) = \text{Newt}(\hat{\ell_2}) &= \text{Conv}(\{(5,1,1),(1,4,1),(1,1,1),(0,0,0)\}). \end{aligned}$$

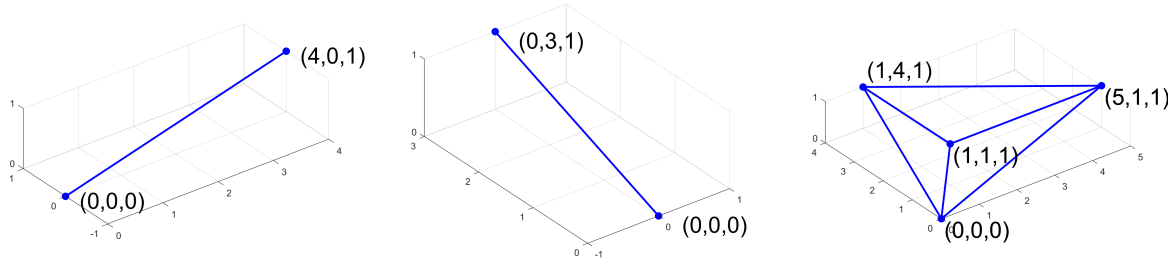


Figure 6.1: $\text{Newt}(\ell_1), \text{Newt}(\ell_2)$ and $\text{Newt}(\hat{\ell_1}) = \text{Newt}(\hat{\ell_2})$ from Example 6.2.4

The following proposition shows that the assumption $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$ in Lemma 6.2.3 is not an issue.

Proposition 6.2.5. Let $F = (f_1, \ldots, f_k)$ and $\hat{F} = (\hat{f}_1, \ldots, \hat{f}_k)$ where $f_j \in \mathbb{C}[x_1, \ldots, x_n]$ and $\hat{f}_j = x_1 \cdots x_n \cdot f_j$ for $j \in [k]$. The ML degree of F equals the ML degree of $\hat{F}$.

Proof. Recall the definition of ML system in (6.2.5), and let

$$\mathcal{L}(F) = \langle \ell_1, \ldots, \ell_n, f_1, \ldots, f_k \rangle \text{ and } \mathcal{L}(\hat{F}) = \langle \hat{\ell}_1, \ldots, \hat{\ell}_n, \hat{f}_1, \ldots, \hat{f}_k \rangle.$$

By Proposition 6.2.2 it suffices to show that there is a bijection between $\mathcal{V}(\mathcal{L}(F)) \cap (\mathbb{C}^*)^{n+k}$ and $\mathcal{V}(\mathcal{L}(\hat{F})) \cap (\mathbb{C}^*)^{n+k}$. We claim such a bijection is given by

$$\begin{aligned} &\phi : \mathcal{V}(\mathcal{L}(F)) \cap (\mathbb{C}^*)^{n+k} \to \mathcal{V}(\mathcal{L}(\hat{F})) \cap (\mathbb{C}^*)^{n+k} \\ &(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) \mapsto (x_1, \ldots, x_n, \frac{\lambda_1}{x_1 \cdots x_n}, \ldots, \frac{\lambda_k}{x_1 \cdots x_n}) \end{aligned}$$

We need to show that ϕ is well defined. Since we assume $(x, \lambda) \in (\mathbb{C}^*)^{n+k}$, $\frac{\lambda_i}{x_1 \cdots x_n}$ is well defined. Now observe that if $f_j(x_1, \ldots, x_n) = 0$ then $\hat{f}_j = x_1 \cdots x_n \cdot f_j(x_1, \ldots, x_n) = 0$ so we only need to show $\hat{\ell}_i$ vanishes on the image of ϕ. By definition,

$$\begin{aligned} \hat{\ell}_i &= u_i - x_i \cdot \sum_{j=1}^{k} \lambda_j \frac{\partial}{\partial x_i}(x_1 \cdots x_n f_j) \\ &= u_i - x_i \cdot \sum_{j=1}^{k} \lambda_j (x_1 \cdots x_{i-1} x_{i+1} \cdots x_n f_j + x_1 \cdots x_n \frac{\partial}{\partial x_i}(f_j)). \end{aligned}$$

Since $f_j(x_1, \ldots, x_n) = 0$ the first term in the summand vanishes. Substituting $\lambda_j \mapsto \frac{\lambda_j}{x_1 \cdots x_n}$, the result is then clear.

Consider

$$\begin{aligned} &\phi^{-1} : \mathcal{V}(\mathcal{L}(F)) \cap (\mathbb{C}^*)^{n+k} \to \mathcal{V}(\mathcal{L}(\hat{F})) \cap (\mathbb{C}^*)^{n+k} \\ &(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) \mapsto (x_1, \ldots, x_n, x_1 \cdots x_n \lambda_1, \ldots, x_1 \cdots x_n \lambda_k) \end{aligned}$$

It is clear that the map $\phi \circ \phi^{-1} = \phi^{-1} \circ \phi$ is the identity, and that

$$\phi^{-1}(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_k) \in \mathcal{V}(\mathcal{L}(F)) \cap (\mathbb{C}^*)^{n+k}.$$

□

6.2.2 Initial systems of the likelihood equations

We now consider the geometry of the Newton polytopes of the likelihood equations and how it relates to the number of $\mathbb{C}^*$ solutions to these equations.

By Proposition 6.2.2 we know that for a general sparse polynomial system F and data vector u, there are finitely many complex solutions to the likelihood equations and that all such complex solutions live in the torus. Therefore, we would like to use the BKK bound (Theorem 2.2.6) to identify the ML degree of F. To do this we need some preliminary results.

By Lemma 6.2.3, if $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$ then

$$\mathrm{Newt}(\ell_j) = \mathrm{Newt}(\ell_i)$$

for $i, j \in [n]$. Call this polytope P. Given some nonzero weight vector $w \in \mathbb{Z}^{n+k}$, we would like to determine which face of P is exposed by w, based on which faces of $\mathrm{Newt}(f_1), \ldots, \mathrm{Newt}(f_k)$ are exposed by w.

Lemma 6.2.6. Let $F = (f_1, \ldots, f_k)$ denote a general sparse polynomial system. Let $\tilde{e}_j \in \mathbb{R}^{n+k}$ be the vector with $(n+j)$-th entry equal to 1 and all other entries equal to 0. Suppose w is a nonzero weight vector in $\mathbb{Z}^{n+k}$.

If $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$, then up to reordering the $f_1, \ldots, f_k$, w exposes P on one of the following faces:

1. the origin
2. $\mathrm{Conv}(\tilde{e}_1 + \mathrm{Newt}_w(f_1), \ldots, \tilde{e}_t + \mathrm{Newt}_w(f_t))$ for some $t \in [k]$
3. $\mathrm{Conv}(0, \tilde{e}_1 + \mathrm{Newt}_w(f_1), \ldots, \tilde{e}_t + \mathrm{Newt}_w(f_t))$ for some $t \in [k]$.

Proof. Fix a nonzero weight vector $w = (a, b) \in \mathbb{R}^n \times \mathbb{R}^k$ and suppose $(v, 0_k) \in \mathrm{Newt}(f_1) \setminus \mathrm{Newt}_w(f_1)$. From the description of P in Lemma 6.2.3 we have that

$$\mathrm{val}_w(P) \in \{0, b_1 + \mathrm{val}_w(f_1), \ldots, b_k + \mathrm{val}_w(f_k)\}.$$

If $b_j + \mathrm{val}_w(f_j) > 0$ for all $j \in [k]$, then P is exposed at the origin, so we are in Case 1. If $b_1 + \mathrm{val}_w(f_1) = \cdots = b_t + \mathrm{val}_w(f_t) = \gamma < 0$ for some $t \in [k]$, where $b_j + \mathrm{val}_w(f_j) > \gamma$ for all $t+1 \leq j \leq k$, then we are in Case 2. If above $\gamma = 0$, then we are in Case 3. $\square$

We illustrate Lemma 6.2.6 with the following example.

Example 6.2.7. Recall the ML system $\mathcal{L}(\hat{f})$ from Example 6.2.4 where $\hat{f} = \langle xy(2x^4 + 3y^3 - 5)\rangle$ and P from Figure 6.1. Consider the three weight vectors

$$w_1 = (-3, 14, 3), \quad w_2 = (-3, -4, 3), \quad w_3 = (-3, 12, 3).$$

The respective exposed faces of P for these weight vectors are

$$P_{w_1} = \{(0,0,0)\}, \quad P_{w_2} = \text{Conv}(\{(5,1,1),(1,4,1)\}), \quad P_{w_3} = \text{Conv}(\{(0,0,0),(5,1,1)\}),$$

and are shown in red in Figure 6.2. Each P_{w_i} corresponds to one of the three cases in Lemma 6.2.6. Namely, P_{w_1} is the origin; P_{w_2} is in Case 2; and P_{w_3} is in Case 3.

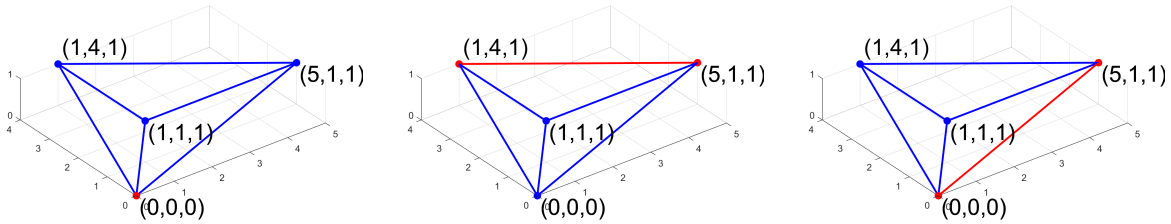


Figure 6.2: P_{w_1}, P_{w_2} and P_{w_3} from Example 6.2.7.

We now need to show that for each of the three cases outlined in Lemma 6.2.6, there are no $\mathbb{C}^*$ solutions to the corresponding initial system.

Lemma 6.2.8. Let $F = (f_1, \ldots, f_k)$ denote a general sparse polynomial system. If $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$, then there are no $\mathbb{C}^*$ solutions to $\text{init}_w(\mathcal{L}(F))$ when P_w is as in Case 1.

Proof. Recall that in Case 1 in Lemma 6.2.6, P_w is the origin. In this case we have $\text{init}_w(\ell_i) = u_i = 0$. Since generally $u_i \neq 0$, this initial system has no solutions. □

Lemma 6.2.9. Let $F = (f_1, \ldots, f_k)$ denote a general sparse polynomial system. If $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$, then there are no $\mathbb{C}^*$ solutions to $\text{init}_w(\mathcal{L}(F))$ when P_w is as in Case 2.

Proof. Recall that in Case 2 in Lemma 6.2.6, P_w is $\mathrm{Conv}(\tilde{e}_1 + \mathrm{Newt}_w(f_1), \ldots, \tilde{e}_t + \mathrm{Newt}_w(f_t))$ for some $1 \leq t \leq k$.

Let $\tilde{f}_j = \mathrm{init}_w(f_j)$ for $j \in [k]$. We consider the following as a subsystem of $\mathrm{init}_w(\mathcal{L}(F))$:

$$\tilde{f}_1 = \ldots = \tilde{f}_t = 0$$
$$x_1\Big(\sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_1}(\tilde{f}_j)\Big) = \ldots = x_n\Big(\sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_n}(\tilde{f}_j)\Big) = 0$$

Since we only consider $\mathbb{C}^*$ solutions, this reduces to

$$\tilde{f}_1 = \ldots = \tilde{f}_t = 0$$
$$\sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_1}(\tilde{f}_j) = \ldots = \sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_n}(\tilde{f}_j) = 0.$$

By Bertini's Theorem [139, Ch. III,§10.9.2], the variety cut out by $\tilde{f}_1 = 0, \ldots, \tilde{f}_t = 0$ has codimension t in $(\mathbb{C}^*)^n$ and $\mathcal{V}(\tilde{f}_1, \ldots, \tilde{f}_t)$ has no singular solutions in the torus. So this initial system has no $\mathbb{C}^*$ solutions.

□

Before we consider the final case of Lemma 6.2.6, we need a preliminary lemma.

Lemma 6.2.10. Let $F = (f_1, \ldots, f_k)$ denote a general sparse polynomial system where $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$. Furthermore, let $w = (a, b) \in \mathbb{Z}^n \times \mathbb{Z}^k$ be a nonzero weight vector. If $a = 0$ then there are no $\mathbb{C}^*$ solutions to $\mathrm{init}_w(\mathcal{L}(F))$.

Proof. Under the assumption $a = 0$,

$$\mathrm{Newt}_w(f_j) = \mathrm{Newt}(f_j) \text{ and } \mathrm{val}_w(f_j) = 0 \text{ for all } j \in [k].$$

Recall from the proof of Lemma 6.2.6,

$$\mathrm{val}_w(P) \in \{0, b_1 + \mathrm{val}_w(f_1), \ldots, b_k + \mathrm{val}_w(f_k)\}.$$

Since $\mathrm{val}_w(f_j) = 0$, this gives that $\mathrm{val}_w(P) \in \{0, b_1, \ldots, b_k\}$. If any $b_j < 0$ or all $b_j > 0$ for $j \in [k]$, then by Lemmas 6.2.9 and 6.2.8 there are no $\mathbb{C}^*$ solutions to the initial system.

It remains to consider when

$$b_1 = \ldots = b_t = 0 \text{ and } b_{t+1}, \ldots, b_k > 0.$$

Note that $t < k$, because otherwise b would be the all zeros vector, which is not allowed. Observe that $\mathrm{init}_w(\ell_1, \ldots, \ell_n, f_1, \ldots, f_t)$ is equal to the ML system of $(f_1, \ldots, f_t)$. By Proposition 6.2.2 there are finitely many solutions to this ML system. Since $f_{t+1}, \ldots, f_k$ are general, they won't intersect the variety of this Lagrange system. □

Lemma 6.2.11. Let $F = (f_1, \ldots, f_k)$ denote a general sparse polynomial system. If $x_1 \cdots x_n \mid f_j$ for all $j \in [k]$, then there are no $\mathbb{C}^*$ solutions to $\mathrm{init}_w(\mathcal{L}(F))$ when P_w is as in Case 3.

Proof. Recall from Case 3 in Lemma 6.2.6, P_w is $\mathrm{Conv}(0, \tilde{e}_1 + \mathrm{Newt}_w(f_1), \ldots, \tilde{e}_t + \mathrm{Newt}_w(f_t))$ for some $1 \leq t \leq k$. Let $\tilde{f}_j = \mathrm{init}_w(f_j)$ for $j \in [k]$.

We consider the subsystem of $\mathrm{init}_w(\mathcal{L}(F))$ given by:

$$\begin{aligned} \tilde{f}_1 = \ldots = \tilde{f}_t &= 0 \\ x_1 \sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_1}(\tilde{f}_j) &= u_1 \\ &\vdots \\ x_n \sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_n}(\tilde{f}_j) &= u_n. \end{aligned}$$

Multiplying $\tilde{f}_j$ by λ_j, this becomes

$$\begin{aligned} \lambda_1\tilde{f}_1 = \ldots = \lambda_t\tilde{f}_t &= 0 \\ x_1 \sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_1}(\tilde{f}_j) &= u_1 \\ &\vdots \\ x_n \sum_{j=1}^{t} \lambda_j \frac{\partial}{\partial x_n}(\tilde{f}_j) &= u_n. \end{aligned}$$

Observe that $\lambda_j \tilde{f}_j$ has the same monomial support as $x_i \lambda_j \frac{\partial}{\partial x_i}(\tilde{f}_j)$ for all $i \in [n]$. Therefore if we write $\lambda_j \tilde{f}_j = \sum_{i=1}^{M_j} c_{i,j} x^{\alpha_{i,j}}$ we can write $\mathrm{init}_w(\ell_1, \ldots, \ell_n, f_1, \ldots, f_t)$ as the linear system $AX = U$:

$$\begin{bmatrix} | & & | & & | & & | \\ \alpha_{1,1} & \cdots & \alpha_{1,M_1} & \cdots & \alpha_{t,1} & \cdots & \alpha_{t,M_t} \\ | & & | & & | & & | \\ - & \mathbf{1}_{M_1} & - & & 0 & \cdots & 0 \\ & \vdots & & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & - & \mathbf{1}_{M_t} & - \end{bmatrix} \cdot \begin{bmatrix} c_{1,1}x^{\alpha_{1,1}} \\ \vdots \\ c_{1,M_1}x^{\alpha,M_1} \\ \vdots \\ c_{t,1}x^{\alpha_t,1} \\ \vdots \\ c_{t,M_t}x^{\alpha_{t,M_t}} \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Note that $A \in \mathbb{N}^{(n+t)\times(M_1+\ldots+M_t)}$, $X \in \mathbb{R}^{M_1+\ldots+M_t}$, $U \in \mathbb{N}^{n+t}$, and $\mathbf{1}_{M_i}$ is a row vector of size M_i of all ones.

For $M_1, \ldots, M_t$ large enough, a dimension count of A suggests its rows are linearly independent. However, it turns out that no matter the size of $M_1, \ldots, M_t$, the matrix A always has a nontrivial left kernel vector:

$$(a_1, \ldots, a_n, -\mathrm{val}_w(f_1), \ldots, -\mathrm{val}_w(f_t)),$$

where $w = (a, b) \in \mathbb{Z}^n \times \mathbb{Z}^k$. This follows from $\langle a, \alpha_{j,i} \rangle = \mathrm{val}_w(f_j)$ for $j \in [t]$.

By Lemma 6.2.10, we know that some of the a_i are nonzero, which contradicts the generality of u, as it would imply that $\langle a, u \rangle = 0$ with $a \neq 0$.

□

6.3 Main result and consequences

Theorem 6.3.1 (Main Result)**.** *For general sparse polynomials $F = (f_1, \ldots, f_k)$, denote its ML system* (6.2.5) *by $\mathcal{L}(F)$. The ML degree of F equals the mixed volume of $\mathcal{L}(F)$.*

Proof. Consider the system $\hat{F} = (\hat{f}_1, \ldots, \hat{f}_k)$ where $\hat{f}_j = x_1 \cdots x_n \cdot f_j$ for $j \in [k]$. It follows from the BKK bound (Theorem 2.2.6), Proposition 6.2.2 and Lemmas 6.2.6, 6.2.8, 6.2.9 and 6.2.11 that the ML degree of $\hat{F}$ equals the mixed volume of $\mathcal{L}(\hat{F})$.

Now we show that the ML degree of F equals the mixed volume of $\mathcal{L}(F)$. First observe that $\mathrm{Newt}(f_j) + (\mathbf{1}_n, 0_k) = \mathrm{Newt}(\hat{f}_j)$ for $j \in [k]$. Since the mixed volume is translation invariant, this gives

$$\mathrm{MVol}(\ell_1, \ldots, \ell_n, f_1, \ldots, f_k) = \mathrm{MVol}(\ell_1, \ldots, \ell_n, \hat{f}_1, \ldots, \hat{f}_k)$$

For $\phi \in SL_{n+k}$ given by

$$\begin{bmatrix} I_n & \mathbf{1}_{n\times k} \\ 0_{k\times n} & I_k \end{bmatrix},$$

we have

$$\begin{aligned} \mathrm{MVol}(\ell_1, \ldots, \ell_n, \hat{f}_1, \ldots, \hat{f}_k) &= \mathrm{MVol}(\phi \cdot \ell_1, \ldots, \phi \cdot \ell_n, \phi \cdot \hat{f}_1, \ldots, \phi \cdot \hat{f}_k) \\ &= \mathrm{MVol}(\phi \cdot \ell_1, \ldots, \phi \cdot \ell_n, \hat{f}_1, \ldots, \hat{f}_k) \end{aligned}$$

Since $\phi(\mathrm{Newt}(\ell_i)) \subseteq \mathrm{Newt}(\hat{\ell}_i)$, by monotonicity of mixed volume we get

$$\mathrm{MVol}(\phi \cdot \ell_1, \ldots, \phi \cdot \ell_n, \hat{f}_1, \ldots, \hat{f}_k) \leq \mathrm{MVol}(\hat{\ell}_1, \ldots, \hat{\ell}_n, \hat{f}_1, \ldots, \hat{f}_k).$$

Thus far we have shown the inequality $\mathrm{MVol}(\mathcal{L}(F)) \leq \mathrm{MVol}(\mathcal{L}(\hat{F}))$. We claim the following list of equalities also holds:

$$\mathrm{MVol}(\mathcal{L}(\hat{F})) = \text{ML Degree of } \hat{F} = \text{ML Degree of } F \leq \mathrm{MVol}(\mathcal{L}(F)). \tag{6.3.1}$$

From the argument given above we also have that the mixed volume of $\mathcal{L}(\hat{F})$ is equal to the ML degree of $\hat{F}$. By Proposition 6.2.5 we have that the ML degree of $\hat{F}$ equals the ML degree of F. The first part of the BKK bound (Theorem 2.2.6) tells us that the ML degree of F is upper bounded by the mixed volume of $\mathcal{L}(F)$. The inequalitiy $\mathrm{MVol}(\mathcal{L}(F)) \leq \mathrm{MVol}(\mathcal{L}(\hat{F}))$ paired with (6.3.1) shows that the mixed volume $\mathcal{L}(F)$ equals the ML degree of F. □

Remark 6.3.2. Theorem 6.3.1 shows that an optimal homotopy method to find all critical points for maximum likelihood estimation is given by a standard polyhedral homotopy (as outlined in Section 2.2.1) from its ML system.

A corollary of our results is that the ML degree of a general sparse polynomial system F depends only on the Newton polytopes.

Corollary 6.3.3. Consider two general sparse polynomial systems: $F = (f_1, \ldots, f_k)$ and $G = (g_1, \ldots, g_k)$, where $\mathrm{Newt}(f_j) = \mathrm{Newt}(g_j)$ for $j \in [k]$. The ML degree of F equals the ML degree of G.

Proof. Suppose F and G have the same Newton polytopes. Consider $x_1 \cdots x_n F = \hat{F}$ and $x_1 \cdots x_n G = \hat{G}$. The ML systems of $\hat{F}$ and $\hat{G}$ have the same Newton polytopes, so by Theorem 6.3.1 the ML degree of $\hat{F}$ equals the ML degree of $\hat{G}$. Proposition 6.2.5 then gives that the ML degree of F equals the ML degree of $\hat{F}$ and likewise for G and $\hat{G}$, giving the result. □

This is a surprising corollary because the Newton polytopes of F do *not* determine the Newton polytopes of the respective ML system.

Example 6.3.4. Consider the ML systems $\mathcal{L}(f)$ and $\mathcal{L}(\hat{f})$ from Example 6.2.4 and Example 6.2.7. Both of these systems have a mixed volume of 12 even though the Newton polytopes of the corresponding Lagrangian likelihood equations are quite different.

Corollary 6.3.3 suggests a way to design homotopy algorithms to do maximum likelihood estimation in the case when the statistical model, $\mathcal{M}$, is algebraic. By considering only the vertices of the Newton polytopes of the polynomials defining $\mathcal{M}$ the likelihood equations can dramatically simplify, leading to optimal start systems that circumvent the bottleneck associated with traditional polyhedral methods. This is illustrated on an example below.

Example 6.3.5. Consider when $\mathcal{M} = \{x \in \mathbb{R}^n \ : \ f(x) = 0\}$ where f is a generic quadratic polynomial. The ML degree of f is the same as the ML degree of g where $g(x) = a_0 + a_1 x_1^2 + \ldots + a_n x_n^2$ for generic a_i, $i \in [n]$. The ML system of g is

$$0 = \ell_i = u_i - 2\lambda a_i x_i^2, \ i \in [n]$$
$$0 = g = a_0 + a_1 x_1^2 + \ldots + a_n x_n^2.$$

We can explicitly solve this system and see

$$\lambda = -\frac{\frac{1}{2}(u_1 + \ldots + u_n)}{a_0}$$
$$x_i^2 = \frac{a_0 u_i}{-a_i(u_1 + \ldots + u_n)}.$$

This shows that the ML degree of g (and therefore f) is 2^n. Moreover, the binomial system $\mathcal{B} = (\ell_1, \ldots, \ell_n, a_0 + a_1 x_1^2)$ gives an optimal polyhedral homotopy start system to find all critical points of the MLE problem for f. Observe that in this case, the Bezout bound of the ML system of f is $2 \cdot 3^n$, so for large enough n, this polyhedral start system will be arbitrarily better.

Chapter 7

Exact semidefinite relaxations to binary programs

We conclude this thesis by studying the Shor relaxation of classes of quadratic polynomial optimization problems. We first study this relaxation when the feasible region is the image of a linear transformation and when it is acted upon by a group. We then consider when the feasible region consists of finitely many points and we show that the set of polynomial optimization problems where this relaxation is tight consists of the union of finitely many spectrahedra. We then focus on quadratic binary programs. We give explicit descriptions of the regions where the Shor relaxation is exact in a variety of situations.

7.1 Problem set up

We first recall from Section 3.2 the Shor relaxations of quadrtaic programs. Let $\mathrm{Sym}_n(\mathbb{R})$ be the set of all $n \times n$ real symmetric matrices. We consider a quadratic program

$$\min_{x \in \mathbb{R}^n} \; g(x) \quad \text{subject to} \quad f_i(x) = 0, \; i \in [m]. \tag{QP}$$

where $g(x) = x^T C x + 2d^T x$ and $f_i(x) = x^T A_i x + 2a_i^T x + \alpha_i$ for $C, A_i \in \mathrm{Sym}_n(\mathbb{R})$ and $d, a_i \in \mathbb{R}^n$ for $i \in [m]$. Denote $\mathcal{V}(F)$ to be the complex variety of $F = (f_1, \ldots, f_m)$.

The optimization problem (QP) has Lagrangian

$$\mathcal{L}(\lambda, x) = g(x) - \sum_{i=1}^{m} \lambda_i f_i(x).$$

The Hessian of $\mathcal{L}$ with respect to x is

$$H(\lambda) = 2 \cdot (C - \sum_{i=1}^{m} \lambda_i A_i).$$

Definition 7.1.1. [171, Def. 3.2] The *SDP-exact region* of (QP), $\mathcal{R}_F$, is the set $(C, d) \in (\mathrm{Sym}_n(\mathbb{R}), \mathbb{R}^n)$ such that the Shor relaxation of (QP) is exact. Specifically,

$$\mathcal{R}_F = \{(C,d) \ : \ H(\lambda) \succ 0, \ d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda)x = 0 \ \text{ for some } x \in \mathcal{V}_{\mathbb{R}}(F), \lambda \in \mathbb{R}^m\}.$$

We wish to examine $\mathcal{R}_F$ for various classes of quadratic programs.

7.2 When is the SDP exact region is well-defined?

The goal of this section is to establish conditions under which $\langle F \rangle = \langle G \rangle$ implies $\mathcal{R}_F = \mathcal{R}_G$ where $\deg(f) = \deg(g) = 2$ for all $f \in F$ and $g \in G$. Unfortunately, as demonstrated in further examples, two ideals being the same does not imply that their corresponding SDP exact region is the same. This section makes progress towards understanding when the SDP exact region of an ideal is well defined. We begin with a proposition which classifies when $\mathcal{R}_F$ is empty.

Proposition 7.2.1. $\mathcal{R}_F = \emptyset$ if and only if $\mathcal{V}_{\mathbb{R}}(F) = \emptyset$.

Proof. $(\Rightarrow)$: We prove this by contrapositive. Assume $\mathcal{V}_{\mathbb{R}}(F) \neq \emptyset$, we want to show that then $\mathcal{R}_F \neq \emptyset$. Let $x \in \mathcal{V}_{\mathbb{R}}(F)$ and fix $\lambda \in \mathbb{R}^m$ and $C \in \mathrm{Sym}_n(\mathbb{R})$ such that $H(\lambda) \succ 0$. Observe that such a C exists since taking $C = \alpha \cdot I$ where α is greater than the smallest eigenvalue of $\sum_{i=1}^m \lambda_i A_i$ works. Then choose $d \in \mathbb{R}^n$ such that $d - \sum_{i=1}^m \lambda_i a_i + H(\lambda)x = 0$.

$(\Leftarrow)$: Suppose $\mathcal{V}_{\mathbb{R}}(F) = \emptyset$. Then trivially by Definition 7.1.1, $\mathcal{R}_F = \emptyset$. □

The condition for $\mathcal{R}_F$ to be empty as outlined in Proposition 7.2.1 is a very natural one. We would now like to extend our analysis of when $\mathcal{R}_F$ is well defined to the case when $\mathcal{V}_{\mathbb{R}}(F)$ is nonempty. The main result of this section is now presented.

Theorem 7.2.2. *The SDP exact region of $F = \langle f_1, \ldots, f_m \rangle$ is the same as $L \cdot F$ where $L \in \mathbb{R}^{N \times m}$ is a rank m matrix.*

Proof. Let $\mathcal{R}_F$ be the SDP exact region of $F = \langle f_1, \ldots, f_m \rangle$ where $f_i = x^T A_i x + a_i^T x + \alpha_i$ and $\mathcal{R}_{LF}$ be the SDP exact region of $L \cdot F = \langle \hat{f}_1, \ldots, \hat{f}_N \rangle$ where ℓ_{ij} is the (i,j)th entry of L and

$$\hat{f}_i = \sum_{j=1}^{m} \ell_{ij} f_j = \sum_{j=1}^{m} \ell_{ij}(x^T A_j x + a_j^T x + \alpha_j), \quad i \in [N].$$

Observe that the variety cut out by F and $L \cdot F$ are the same. Specifically, if $\mathcal{V}_{\mathbb{R}}(F) = \emptyset$ then $\mathcal{V}_{\mathbb{R}}(LF) = \emptyset$ so by Proposition 7.2.1 both SDP exact regions will be empty. Therefore, we now assume $\mathcal{V}_{\mathbb{R}}(F) \neq \emptyset$.

First we will show $\mathcal{R}_F \subseteq \mathcal{R}_{LF}$. Suppose $(C, d) \in \mathcal{R}_F$ corresponding to point $x \in \mathcal{V}_{\mathbb{R}}(F)$ and $\lambda \in \mathbb{R}^m$. We claim $(C, d) \in \mathcal{R}_{LF}$ corresponding to point $x \in \mathcal{V}(F)$ and $\hat{\lambda} \in \mathbb{R}^m$ where $L^T \hat{\lambda} = \lambda$. Observe that since L^T is a rank m, $N \times m$ matrix, such a $\hat{\lambda}$ exists. Let $\hat{H}$ be the Hessian of LF. Then

$$\begin{aligned}
\hat{H}(\hat{\lambda}) &= \sum_{i=1}^{N} \hat{\lambda}_i \hat{A}_i \\
&= \sum_{i=1}^{N} \hat{\lambda}_i \sum_{j=1}^{m} \ell_{ij} A_j \\
&= \sum_{i=1}^{m} A_i \sum_{j=1}^{N} \hat{\lambda}_j \ell_{ji} \\
&= \sum_{i=1}^{m} \lambda_i A_i = H(\lambda) \succ 0
\end{aligned}$$

where the last line is by definition of $\hat{\lambda}$. Now consider the equality constraint:

$$\begin{aligned}
d - \sum_{i=1}^{N} \hat{\lambda}_i \hat{a}_i + \hat{H}(\hat{\lambda}) x &= d - \sum_{i=1}^{N} \hat{\lambda}_i \sum_{j=1}^{m} \ell_{ij} a_j + H(\lambda) x \\
&= d - \sum_{i=1}^{m} a_i \sum_{j=1}^{N} \lambda_j \ell_{ji} + H(\lambda) x \\
&= d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda) x = 0.
\end{aligned}$$

The same argument holds for showing $\mathcal{R}_{LF} \subseteq \mathcal{R}_F$ by considering $\lambda = L^T \hat{\lambda}$. $\square$

Theorem 7.2.2 tells us that for an ideal $\mathcal{I} = \langle F \rangle = \langle f_1, \ldots, f_m \rangle$, $\mathcal{R}_F$ is well-defined if for any other ideal $\mathcal{J}$ such that $\mathcal{J} = \mathcal{I}$, there exists a set of generators of $\mathcal{J} = \langle G \rangle = \langle g_1, \ldots, g_N \rangle$ such that $G = L \cdot F$ for some $L \in \mathbb{R}^{N \times m}$ with rank m. We give a sufficient condition under which ideals defined by quadratic generators satisfy this assumption.

Theorem 7.2.3. *Consider $F = \langle f_1, \ldots, f_m \rangle$ where $\deg(f_i) = 2$, $f_i \in \mathbb{R}[x_1, \ldots, x_n]$ and $m \leq n$. Write $f_i = g_i + h_i$ where g_i is the homogeneous degree two part of f_i and h_i is the degree zero and one part. If the variety $\mathcal{V}(g_1, \ldots, g_m)$ has codimension m, then $\mathcal{R}_F$ is well-defined.*

Proof. Consider some ideal $Q = \langle q_1, \ldots, q_N \rangle$ where $Q = F$ and $\deg(q_j) = 2$ for all $j \in [N]$. Since $Q = F$, for all $q \in Q$ we can write

$$q = a_1 f_1 + \ldots + a_m f_m$$

for some $a_i \in \mathbb{R}[x_1, \ldots, x_n]$. By Theorem 7.2.2, it suffices to show that $\deg(a_i) = 0$. We prove this by induction on $d = \max_{i \in [m]} \deg(a_i)$.

If $d = 0$ we are done, so assume $d > 0$. Let $a_i = a_i^{(d)} + a_i'$ where $a_i^{(d)}$ is homogeneous of degree d and $\deg(a_i') < d$. The degree $d + 2$ part of q is then given by

$$q^{(d+2)} = a_1^{(d)} g_1 + \ldots + a_m^{(d)} g_m.$$

Since $\deg(q) = 2$ and $d > 0$, we have that $q^{(d+2)} = 0$. Therefore, we can write

$$\begin{pmatrix} a_1^{(d)} \\ \vdots \\ a_m^{(d)} \end{pmatrix} = M \cdot \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}$$

where M is a matrix of homogeneous forms of degree at most $d - 2$. Substituting, we write

$$\begin{pmatrix} a_1 - a_1' \\ \vdots \\ a_m - a_m' \end{pmatrix} = M \cdot \begin{pmatrix} f_1 - h_1 \\ \vdots \\ f_m - h_m \end{pmatrix}.$$

Putting this together, we write

$$\begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = M \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} - M \cdot \begin{pmatrix} h_1 \\ \vdots \\ h_m \end{pmatrix} + \begin{pmatrix} a_1' \\ \vdots \\ a_m' \end{pmatrix} = M \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} + \begin{pmatrix} a_1'' \\ \vdots \\ a_m'' \end{pmatrix}$$

where $\deg(a_i'') \leq d - 2 + 1 = d - 1$. We then see that

$$q = a_1'' f_1 + \ldots + a_m'' f_m$$

so we can use the induction hypothesis on a_i'' to conclude that

$$q = \alpha_1 f_1 + \ldots + \alpha_m f_m$$

where $\alpha_i \in \mathbb{R}$. By Theorem 7.2.2 we are done. □

Corollary 7.2.4. Consider $F = \langle f_1, \ldots, f_m \rangle$ where $\deg(f_i) = 2$, each f_i has full monomial support and the coefficients of each f_i lie in some Zariski open set for all $i \in [m]$. Then $\mathcal{R}_F$ is well-defined.

The requirement that each f_i has degree 2 is a critical assumption. This is demonstrated in the next proposition and example.

Proposition 7.2.5. Consider $F = \langle f_1, \ldots, f_m \rangle$ and $\hat{F} = \langle f_1, \ldots, f_m, g \rangle$ where $g \in \langle F \rangle$ and $\deg(g) = 2$. Then $\mathcal{R}_F \subseteq \mathcal{R}_{\hat{F}}$.

Proof. Consider $(C, d) \in \mathcal{R}_F$ corresponding to $\lambda \in \mathbb{R}^m$ and $x \in \mathcal{V}_{\mathbb{R}}(F)$. Then $(C, d) \in \mathcal{R}_{\hat{F}}$ corresponding to $\hat{\lambda} = (\lambda, 0) \in \mathbb{R}^{m+1}$ and $x \in \mathcal{V}_{\mathbb{R}}(\hat{F}) = \mathcal{V}_{\mathbb{R}}(F)$. □

The following example illustrates why $\mathcal{R}_F \neq \mathcal{R}_{\hat{F}}$ in Proposition 7.2.5 even though $\mathcal{V}(F) = \mathcal{V}(\hat{F})$.

Example 7.2.6. Consider $\mathcal{R}_F$ and $\mathcal{R}_{\hat{F}}$ where $F = \langle x - y \rangle$ and $\hat{F} = \langle x - y, x^2 - xy + x - y \rangle$. First observe that the Hessian of F, $H_F = C$. Therefore any $(C, d) \in \mathcal{R}_F$ must have $C \succ 0$. We claim there exists $C \not\succ 0$ in $\mathcal{R}_{\hat{F}}$. Consider $C = \begin{bmatrix} -\frac{1}{2} & -1 \\ -1 & 5 \end{bmatrix}$ and $d = [2 \;\; 2]^T$. Then taking $(\lambda_1, \lambda_2) = (-\frac{31}{5}, -1)$ and $x = -\frac{4}{5}$, one can check that $(C, d) \in \mathcal{R}_{\hat{F}}$.

While this example is counterintuitive, we explain it as follows. The objective value considered, C is non-convex so we think this SDP should be unbounded and therefore strong duality will not hold. In this case, restricting to the linear space $x = y$ then transforms the original objective value $-\frac{1}{2}x^2 + 5y^2 - 2xy$ into the objective value $\frac{5}{2}x^2$ which is convex. In other words, the objective value is convex when restricted to the feasible space. In the original formulation, our Hessian did not consider this.

7.3 Geometric descriptions of $\mathcal{R}_F$

We now consider situations where we can nicely characterize $\mathcal{R}_F$. We first consider when $\mathcal{V}_{\mathbb{R}}(F)$ has symmetry.

7.3.1 Group actions by subgroups of $GL_n(\mathbb{R})$

We now consider the SDP exact region of a variety $\mathcal{V}_{\mathbb{R}}(F)$ under the multiplication of an element of $GL_n(\mathbb{R})$ where an element $M \in GL_n(\mathbb{R})$ acts on a point $x \in \mathcal{V}_{\mathbb{R}}(F)$ by $x \mapsto M \cdot x$.

Theorem 7.3.1. *Suppose $(C, d) \in \mathcal{R}_F$ where $\mathcal{R}_F$ is the SDP exact region of* (QP)*. Consider*

$$\min_{x \in \mathbb{R}^n} \; g(x) \quad \textit{subject to} \quad x \in M \cdot \mathcal{V}_{\mathbb{R}}(F) \tag{M-QP}$$

where $M \in GL_n(\mathbb{R})$. Then $(M^{-T}CM^{-1}, M^{-T}d) \in \mathcal{R}_{M \cdot F}$, the SDP exact region of (M-QP)*.*

Proof. Consider $M \in GL_n(\mathbb{R})$ and suppose (M, b) acts on $x \in \mathcal{V}_{\mathbb{R}}(F)$ of (QP) by $x \mapsto Mx$. Using the change of coordinates induced by M, (QP) becomes

$$\min_{x \in \mathbb{R}^n} \; g(M^{-1}x) \quad \text{subject to} \quad f_i(M^{-1}x) = 0. \tag{7.3.1}$$

In matrix notation, $g(M^{-1}x)$ and $f_i(M^{-1}x)$ for $i \in [m]$ become

$$g(M^{-1}x) = x^T M^{-T} C M^{-1} x + 2d^T M^{-1} x$$
$$f_i(M^{-1}x) = x^T M^{-T} A_i M^{-1} x + 2a_i^T M^{-1} x + \alpha_i.$$

The Hessian of the Lagrangian of (7.3.1) problem is given as

$$\hat{H}(\lambda) = M^{-T} H(\lambda) M^{-1}$$

where $H(\lambda)$ is the Hessian of the Lagrangian associated to (QP). It is clear that if $H(\lambda) \succ 0$ for some $\lambda \in \mathbb{R}^m$ then $\hat{H}(\lambda) \succ 0$.

Writing out the definition of $\mathcal{R}_F$ for (7.3.1) we have

$$M^{-T}d - \sum_{i=1}^{m} \lambda_i M^{-T}a_i + M^{-T}H(\lambda)M\hat{x} = 0 \qquad \text{for } \hat{x} \in \mathcal{R}_{MF}$$

Since $\hat{x} = Mx$ for some $x \in \mathcal{V}_{\mathbb{R}}(F)$ we have

$$M^{-T}(d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda)x) = 0$$

It is then clear that if $(C, d) \in \mathcal{R}_F$, then $(M^{-T}CM^{-1}, M^{-T}d) \in \mathcal{R}_{MF}$. □

Corollary 7.3.2. Consider a subgroup $\mathcal{G} \subseteq GL_n(\mathbb{R})$ that acts on $\mathcal{V}_{\mathbb{R}}(F)$ as above and partitions $\mathcal{V}_{\mathbb{R}}(F)$ into orbits $\mathcal{O}_i$, $i \in \mathcal{I}$. Suppose $x \in \mathcal{O}_i$ for some i. Then the spectrahedral shadows, S_x and S_{Mx}, defined as the SDP exact regions corresponding to x and Mx are in bijection via the action induced by $M \in \mathcal{G}$ described above.

7.3.2 $\mathcal{R}_F$ as a finite union

In [171] the authors remark that $\mathcal{R}_F$ is nicely parameterized as the union of spectrahedral shadows where there is one spectrahedral shadow for every point $x \in \mathcal{V}_{\mathbb{R}}(F)$. We would like to refine this type of characterization where we only consider finite unions. We first begin for the general case of a complete intersections.

Theorem 7.3.3. *Let $F = \langle f_1, \ldots, f_m \rangle$ be a complete intersection such that $\mathcal{I}(F)$ is radical where $f_i = x^T A_i x + 2a_i^T x + \alpha_i$. The SDP exact region of*

$$\min_{x \in \mathbb{R}^n} \; x^T C x + 2d^T x \quad \textit{subject to} \quad x \in \mathcal{V}_{\mathbb{R}}(F)$$

is a finite union of basic semi-algebraic sets. Moreover, the number of such sets is bounded above by 2^{n+m}.

Proof. By Definition 7.1.1 we have

$$\mathcal{R}_F = \{(C,d) \ : \ H(\lambda) \succ 0, \ d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda)x = 0 \ \text{ for some } x \in \mathcal{V}_{\mathbb{R}}(F), \lambda \in \mathbb{R}^m\}.$$

In order for $(C,d) \in \mathcal{R}_F$ two conditions must be satisfied. The first is that $d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda)x = 0$ for some $x \in \mathcal{V}_{\mathbb{R}}(F)$. This can be rephrased as understanding the real variety of G where G is

$$\begin{aligned} d - \sum_{i=1}^{m} \lambda_i a_i + H(\lambda)x = 0 \\ f_1 = 0 \\ \vdots \\ f_m = 0 \end{aligned}$$

Observe that this is a system of $n+m$ equations in the $n+m$ unknowns, x_i, λ_j for $i \in [n]$ and $j \in [m]$. Since F is a complete intersection, this system of equations has finitely many complex, and therefore real, solutions. Projecting the real solutions of G onto the the λ coordinates gives finitely many $H(\lambda)$. Observe that $H(\lambda)$ has entries that are linear in C, A_i, $i \in [m]$ and roots of polynomials with coefficients in C, d, A_i, a_i, $i \in [m]$. Therefore, $H(\lambda)$ does not necessarily contain entries that are just linear in C, d so the set (C,d) such that $H(\lambda) \succ 0$ is not necessarily a spectrahedron, but the condition $H(\lambda) \succ 0$ does give a set (C,d) where the SDP exact region is exact.

By the Bezout bound, the number of complex, and therefore real, solutions to $\mathcal{V}(G)$ is bounded above by 2^{n+m}. The result then follows. □

We remark that in all computations, $\mathcal{V}(G)$ as defined in the proof of Theorem 7.3.3 has had fewer than 2^{n+m} complex solutions. Therefore, we conjecture that this upper bound is strict. We show the explicit description of the semialgebraic sets referenced in Theorem 7.3.3 below.

Example 7.3.4. Consider the problem

$$\min_{x_1,x_2 \in \mathbb{R}} \ x_2^2 + c_{12}x_1x_2 + 2d_1x_1 \quad \text{subject to} \quad x_1^2 = 1 \tag{7.3.2}$$

The SDP exact region of (7.3.2) is

$$\{c_{12}, d_1 \ : \ H(\lambda) \succ 0, d_1 - \lambda x_1 + \frac{1}{2}c_{12}x_2 = 0, \frac{1}{2}c_{12}x_1 + x_2 = 0, x_1^2 = 1\}.$$

We can solve the equations explicitly to get two solutions for λ:

$$\frac{-c_{12}^2 \pm 4d_1}{4}$$

Substituting this for λ in $H(\lambda)$ we have the SDP exact region of (7.3.2) is (c_{12}, d_1) such that one of the following holds:

$$H(\lambda_1) = \begin{bmatrix} \frac{c_{12}^2}{4} + d_1 & \frac{c_{12}}{2} \\ \frac{c_{12}}{2} & 1 \end{bmatrix} \succ 0$$

$$H(\lambda_2) = \begin{bmatrix} \frac{c_{12}^2}{4} - d_1 & \frac{c_{12}}{2} \\ \frac{c_{12}}{2} & 1 \end{bmatrix} \succ 0$$

In this case, $\mathcal{R}_F = \{(c_{12}, d_1) \in \mathbb{R}^2 \ : \ d_1 \neq 0\}$. The upper plane corresponds to points in $H(\lambda_1)$ and the lower to points in $H(\lambda_2)$.

We now wish to refine the results from Theorem 7.3.3 and give conditions under which $\mathcal{R}_F$ consists of finitely many spectrahedra, not just semialgebraic sets. The following theorem gives a necessary condition for this.

Theorem 7.3.5. *Let $F = (f_1, \ldots, f_n)$ be a quadratic ideal such that $\mathcal{I}(F)$ is radical where $f_i = x^T A_i x + 2a_i^T x + \alpha_i$ and $\mathcal{V}_{\mathbb{C}}(F)$ is finite. The SDP exact region of*

$$\min_{x \in \mathbb{R}^n} x^T C x + 2d^T x \quad \textit{subject to} \quad x \in \mathcal{V}_{\mathbb{R}}(F)$$

is a finite union of spectrahedra where the number of spectrahedra is $|\mathcal{V}_{\mathbb{R}}(F)|$.

Proof. The system $d - \sum_{i=1}^n \lambda_i a_i + H(\lambda)x = 0$ is linear in λ. Specifically, this system can be written as

$$[a_1 + A_1 x, \ldots, a_n + A_n x]\lambda = d + Cx. \tag{7.3.3}$$

Observe that $J_x = [a_1 + A_1x, \ldots, a_nA_nx]$ is the Jacobian of F at x. Since $\dim(\mathcal{V}_{\mathbb{C}}(F)) = 0$, this implies that J_x is full rank for all $x \in \mathcal{V}_{\mathbb{C}}(F)$. This means (7.3.3) has a unique solution, $\lambda_1(x), \ldots, \lambda_n(x)$, that can be written in terms of x. Observe that $\lambda = J_x^{-1}(d+Cx)$. This means that $\lambda_i(x)$ is a rational function in x, d, C but by properties of matrix multiplication, the denominator of λ_i does not contain any d, C coordinates. Moreover, the numerator is linear in d and C. This means we can write $H(\lambda) \succ 0$ as $H(x) \succ 0$. For each $x \in \mathcal{V}(F)$ this gives a spectrahedron. i.e. $S_x = \{(C, d) : H(x) \succ 0\}$. Therefore,

$$\mathcal{R}_F = \bigcup_{x \in \mathcal{V}_{\mathbb{R}}(F)} S_x.$$

□

7.4 Quadratic binary programs

Finally, we conclude this thesis by considering an application of the results derived above to an important class of problems prevelant in combinatorial optimization: quadratic binary programs. Specifically, we consider

$$\min_{x \in \mathbb{R}^n} x^TCx + 2d^Tx \quad \text{subject to} \quad f_i(x) := x_i^2 - 1 = 0,\ i \in [n]. \tag{QBP}$$

We would like to study the SDP exact region of $\mathcal{R}_F$. The constraint $H(\lambda) \succ 0$ and $d + H(\lambda)x = 0$ in this case reduces to finding which (C, d) give

$$H(x) = \begin{bmatrix} -\frac{1}{x_1}(d_1 + \sum_{i=2}^n c_{1i}x_i) & c_{12} & \cdots & c_{1n} \\ c_{12} & -\frac{1}{x_2}(d_2 + c_{12}x_1 + \sum_{i=3}^n c_{2i}x_i) & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{1n} & & & -\frac{1}{x_n}(d_n + \sum_{i=1}^{n-1} c_{in}x_i) \end{bmatrix} \succ 0 \tag{7.4.1}$$

for some $x_i \in \{-1, +1\}$.

Proposition 7.4.1. Let $D \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $d_{ii} \in \{-1, +1\}$ for all $i \in [n]$. Then $\mathcal{PD}_n = D \cdot \mathcal{PD}_n \cdot D$.

Proof. Since the eigenvalues of a matrix are invariant under conjugation, $\mathcal{PD}_n$ is invariant under conjugation by full rank matrices. In this case, $D = D^{-1}$ is full rank. The result then follows. $\square$

Theorem 7.4.2. *The SDP-exact region of* (QBP) *consists of* 2^n *disjoint spectrahedra which are all linearly equivalent to*

$$\mathcal{S} = \{(C,d) \ : \ \begin{bmatrix} -(d_1 + \sum_{i=2}^n c_{1i}) & c_{12} & \cdots & c_{1n} \\ c_{12} & -(d_2 + c_{12} + \sum_{i=3}^n c_{2i}) & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{1n} & & & -(d_n + \sum_{i=1}^{n-1} c_{in}) \end{bmatrix} \succ 0\}.$$

Specifically, $\mathcal{R}_F = \{g \cdot \mathcal{S} \ : \ g \in G\}$ *where* G *is the matrix group*

$$G = \{\mathrm{Diag}(x_1x_2, x_1x_3, \ldots, x_1x_n, x_2x_3, \ldots, x_2x_n, \ldots x_{n-1}x_n, x_1, \ldots, x_n) : x_i \in \{-1,1\},\ i \in [n]\}.$$

Moreover, G *is isomorphic to* $\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$.

Proof. First observe by Theorem 7.3.5, $\mathcal{R}_F$ consists of the union of 2^n disjoint spectrahedra, each one given by (7.4.1) for each $x \in \{-1,+1\}^n$. Next we see that $\mathcal{S}$ corresponds to $H(x)$ for the point $x = (1,\ldots,1) \in \{-1,+1\}^n$. Now fix a point $y \in \{-1,+1\}^n$ and let $M_y \in G$ be the corresponding diagonal matrix and $\mathcal{S}_y = \{(C,d) : H(y) \succ 0\}$ be the corresponding spectrahedron defined by (7.4.1). We claim that $\mathcal{S} \cdot M_y = \mathcal{S}_y$.

By direct computation, we first see that

$$\mathcal{S} \cdot M_y = \{(C,d) \ : \ \begin{bmatrix} -\frac{1}{y_1}(d_1 + \sum_{i=2}^n c_{1i}y_i) & y_1y_2c_{12} & \cdots & y_1y_nc_{1n} \\ y_1y_2c_{12} & -\frac{1}{y_2}(d_2 + c_{12}y_1 + \sum_{i=3}^n c_{2i}y_i) & \cdots & y_1y_nc_{2n} \\ \vdots & & \ddots & \vdots \\ y_1y_nc_{1n} & & & -\frac{1}{y_n}(d_n + \sum_{i=1}^{n-1} c_{in}y_i) \end{bmatrix} \succ 0\}.$$

Call $\mathcal{S} \cdot M_y = \mathcal{S}'_y$. It remains to show that

$$\mathcal{S}'_y = \{(C,d) \ : \ \begin{bmatrix} -\frac{1}{y_1}(d_1 + \sum_{i=2}^n c_{1i}y_i) & c_{12} & \cdots & c_{1n} \\ c_{12} & -\frac{1}{y_2}(d_2 + c_{12}y_1 + \sum_{i=3}^n c_{2i}y_i) & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{1n} & & & -\frac{1}{y_n}(d_n + \sum_{i=1}^{n-1} c_{in}y_i) \end{bmatrix} \succ 0\}$$

$$= \mathcal{S}_y.$$

Observe that if $\mathcal{S}'_y$ is considered in $\mathbb{R}^{\frac{n(n+1)}{2}}$ with coordinate ordering

$$(c_{12}, \ldots, c_{1n}, c_{23}, \ldots, c_{2n}, \ldots, c_{n-1,n}, d_1, \ldots, d_n)$$

then taking

$$D = \mathrm{Diag}(y_1y_2, \ldots, y_1y_n, y_2y_3, \ldots, y_2y_n, \ldots y_{n-1}y_n, y_1, \ldots, y_n)$$

we have that $D \cdot \mathcal{S}'_y \cdot D = \mathcal{S}_y$. By Proposition 7.4.1, $\mathcal{S}'_y = \mathcal{S}_y$.

The fact that $G \cong \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$ is immediate via the isomorphism:

$$\phi : G \to \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$$

$$g_x \mapsto x$$

□

Example 7.4.3. Consider when $n = 2$. Here we have

$$H(x) = \begin{bmatrix} -\frac{1}{x_1}(d_1 + c_{12}x_2) & c_{12} \\ c_{12} & -\frac{1}{x_2}(d_2 + c_{12}x_1) \end{bmatrix}.$$

In this case $(x_1, x_2) \in \{(1,1), (-1,1), (1,-1), (-1,-1)\}$. This gives

$$H(1,1) = \begin{bmatrix} -c_{12} - d_1 & c_{12} \\ c_{12} & -c_{12} - d_2 \end{bmatrix} \qquad H(-1,1) = \begin{bmatrix} c_{12} + d_1 & c_{12} \\ c_{12} & c_{12} - d_2 \end{bmatrix}$$

$$H(-1,-1) = \begin{bmatrix} -c_{12} + d_1 & c_{12} \\ c_{12} & -c_{12} + d_2 \end{bmatrix} \qquad H(1,-1) = \begin{bmatrix} c_{12} - d_1 & c_{12} \\ c_{12} & c_{12} + d_2 \end{bmatrix}.$$

Call the corresponding spectrahedron $S_{1,1}, S_{-1,1}, S_{-1,-1}, S_{1,-1}$. Note that each spectrahedron lives in $(c_{12}, d_1, d_2) = \mathbb{R}^3$. We claim every spectrahedron is linearly equivalent to $S_{1,1}$. Observe

$$S_{-1,-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot S_{1,1}$$

$$S_{-1,1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot S_{1,1}$$

$$S_{1,-1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot S_{1,1}.$$

We see that the set of matrices transforming $S_{1,1}$ to the other spectrahedron form a group. Note here that $\begin{bmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{bmatrix}$ and $\begin{bmatrix} c_{11} & -c_{12} \\ -c_{12} & c_{22} \end{bmatrix}$ define the same spectrahedron. In this case $\mathcal{R}_F = \{x \cdot S_{1,1} : x \in \mathcal{G}\}$ where

$$\mathcal{G} = \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \right\} \cong \mathbb{Z}_2 \times \mathbb{Z}_2.$$

A picture of $\mathcal{R}_F$ in this case is shown in Figure 7.1.

7.4.1 An algorithm to test if (C, d) give an SDP exact solution

Recall from (7.4.1), determining if a quadratic binary program has exact SDP relaxation it is enough to say if

$$H(x) = \begin{bmatrix} -\frac{1}{x_1}(d_1 + \sum_{i=2}^n c_{1i}x_i) & c_{12} & \cdots & c_{1n} \\ c_{12} & -\frac{1}{x_2}(d_2 + c_{12}x_1 + \sum_{i=3}^n c_{2i}x_i) & \cdots & c_{2n} \\ \vdots & & \ddots & \vdots \\ c_{1n} & & & -\frac{1}{x_n}(d_n + \sum_{i=1}^{n-1} c_{in}x_i) \end{bmatrix} \succ 0 \tag{7.4.2}$$

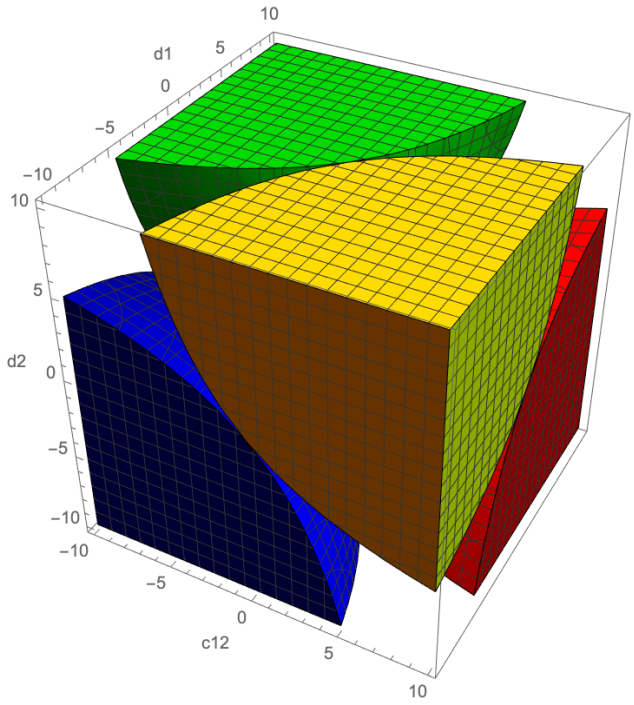


Figure 7.1: $\mathcal{R}_F$ for the problem $\min_{x\in\{-1,1\}^2} \; x^T C x + 2d^T x$.

for some $x_i \in \{-1, +1\}$. While this problem is NP hard [172], we still would like to use some of the results above to derive an algorithm to test if for a given $(C, d) \in \mathrm{Sym}_n(\mathbb{R}) \times \mathbb{R}^n$, the SDP relaxation of (QBP) is exact.

A necessary condition for $H(x) \succ 0$ for some $x \in \{-1, 1\}^n$ is for the diagonal entries of $H(x)$ to be positive. Observe that these entries are linear, so for each $x \in \{-1, 1\}^n$ we have a corresponding polyhedral cone, P_x. Observe that checking if $x \in P_x$ amounts to checking if n expressions are positive. For a given $(C, d) \in \mathrm{Sym}_n(\mathbb{R})\times\mathbb{R}^n$ we would like to find an $x \in \{-1, 1\}^n$ such that $(C, d) \in P_x$. The following algorithm outlines a way to do this.

Algorithm 5 Algorithm to find P_x such that $(C, d) \in P_x$

Input: Objective function $(C, d) \in \mathrm{Sym}_n(\mathbb{R}) \times \mathbb{R}^n$

Output: $x \in \{-1, 1\}^n$ such that $(C, d) \in P_x$

1. Fix $x \in \{-1, 1\}^n$. See if $(C, d) \in P_x$, if so then terminate.

2. If $(C, d) \notin P_x$ then let i^* be the first index of $v \in \mathbb{R}^n$ such that $v_{i^*} < 0$. Let $\hat{x} \in \{-1, 1\}^n$ be the vector such that
$$\hat{x}_j = \begin{cases} x_j & \text{for } j \neq i^* \\ -x_j & \text{for } j = i^* \end{cases}$$
Check if $(C, d) \in P_{\hat{x}}$.

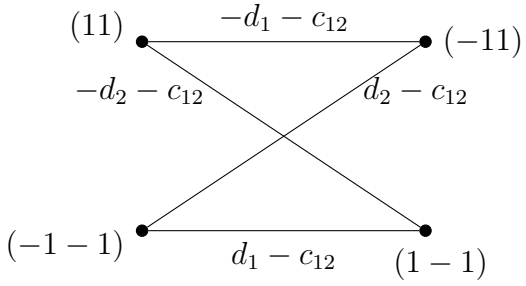
3. If $(C, d) \in P_{\hat{x}}$ then terminate. Else, repeat Step (2) until we find an $x' \in \{-1, 1\}^n$ such that $(C, d) \in P_{x'}$.

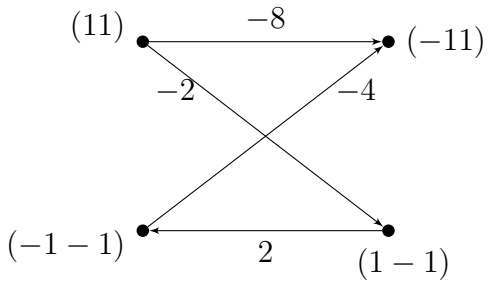
Every time Step 2 is repeated we jump from $x \in \{-1, 1\}^n$ to $x' \in \{-1, 1\}^n$ where x' only differs from x in the ith coordinate. We can consider the complexity of this algorithm as the number of times we repeat Step 2.

We write the sequence Algorithm 5 takes as $\{e_{i_1}, \ldots, e_{i_k}\}$ where e_{i_j} reflects the fact that in Step 2 we negate the i_jth entry. Each time we repeat Step 2, another inequality is induced on the entries of (C, d).

To follow the path that Algorithm 5 takes, we consider the bipartite graph, G, consisting of nodes V_E, V_O. Each $x \in \{-1, 1\}^n$ that differs from $(1, \ldots, 1)$ in an even number of places represents a node in V_E and each $x \in \{-1, 1\}^n$ that differs from $(1, \ldots, 1)$ in an odd number of places represents a node in V_O. There is an edge between $x \in V_E$ and $y \in V_O$ if x and y differ in exactly one place.

Example 7.4.4. Here is a picture of G for $n = 2$. Suppose we have $d_1 = 5, d_2 = -1, c_{12} = 3$. Then we would get





Each edge has a linear expression in the entries of C and d associated to it, given by one of the inequalities of P_x. Namely, consider an edge $e = xy$ where $x \in V_E$ and $y \in V_O$. Suppose that x and y differ in the ith place. Then, e has the ith inequality of P_x associated to it. If this inequality is positive we direct e to be pointing to the left i.e. $e = yx$. If it is negative we direct e to be pointing to the right i.e. $e = xy$.

For a fixed (C, d), G then consists of a directed bipartite graph. For this algorithm to terminate, G needs to be acyclic.

Lemma 7.4.5. Consider a sequence $e_{i_1}, \ldots, e_{i_k}$ starting from $x = (1, \ldots, 1)$. Let s_i be the number of times e_i appears in this sequence. Suppose e_j appears next in the sequence, then this induces the inequality

$$(-1)^{s_j} d_j + (-1)^{s_1+s_j} c_{1j} + \ldots + (-1)^{s_{j-1}+s_j} c_{j-1,j} + (-1)^{s_{j+1}+s_j} c_{j,j+1} + \ldots + (-1)^{s_n+s_j} c_{jn} < 0. \tag{7.4.3}$$

Proof. This is by definition of P_x. □

Theorem 7.4.6. *Algorithm 5 terminates. Moreover, for all (C, d) there exists at least one $x \in \{-1, 1\}^n$ such that $(C, d) \in P_x$.*

Proof. It suffices to show that G is acylic. For the sake of contradiction, suppose there is a cycle of length $2k$, $\mathcal{C} = \{e_{i_1}, \ldots, e_{i_{2k}}\}$. For each vertex in the cycle we get an inequality of the form (7.4.3). We claim the sum of these inequalities is 0.

Let $S = \{i_1, \ldots, i_{2k}\}$ be the set of moves used to form $\mathcal{C}$. Since $\mathcal{C}$ is a cycle, every $i_j \in S$ appears exactly twice. This tells us that d_j appears in exactly two inequalities: when e_j appears

first and when e_j appears second. By Lemma 7.4.5, when it first appears it is negative and when it appears second it is positive. This shows that summing these inequalities results in all d_i being canceled out.

Now we claim that c_{ij} appears in exactly 0, 2 or 4 inequalities. If $i \notin S$ and $j \notin S$ then c_{ij} does not appear in a single inequality. If $i \in S$ and $j \notin S$ then c_{ij} appears in exactly two inequalities, when e_i appears each time. By Lemma 7.4.5, the first time e_i appears c_{ij} is negative and the second time it appears it is positive. If $i \in S$ and $j \in S$ then c_{ij} appears in four inequalities. By Lemma 7.4.5 the first time e_i or e_j appears c_{ij} is negative, the second time it is positive, the third time it is negative and the fourth time it is positive. In either case, this shows that summing all inequalities results in all c_{ij} being canceled out. Therefore summing all inequalities gives $0 < 0$ which is an obvious contradiction. Therefore, no cycle can exist.

Since our graph is a directed acyclic graph, this means there exists a topological sorting. This means there exists a vertex with all arrows pointing in. The vertex x with all arrows pointing in then satisfies $(C, d) \in P_x$.

Since there are no cycles in our graph, as we follow a path from $(1, \ldots, 1)$ we will not visit any vertex twice. Since there are finitely many vertices, we keep going to distinct vertices and there is guaranteed to exist a vertex such that $(C, d) \in P_x$, this means eventually we will end up at one such vertex.

Since our graph is a directed acyclic graph, this means there exists a topological sorting. This means there exists a vertex with all arrows pointing in. The vertex x with all arrows pointing in then satisfies $(C, d) \in P_x$. □

We present empirical results on how long it takes to run Algorithm 5 in Table 7.1. We calculate each (C, d) randomly by sampling each element c_{ij}, d_i, $i, j \in [n]$, to be iid $\mathcal{N}(0, 1)$. Overall, we see on average that Algorithm 5 terminates in less than n iterations and it always terminates in less than $2n$ iterations.

Conjecture 7.4.7. *Algorithm 5 terminates in at most $2n$ iterations.*

n	10	20	30	40	50	60	70	80	90	100
Maximum	16	29	44	63	87	99	115	126	146	171
Minimum	1	3	7	8	15	24	31	37	47	58
Mean	6.21	13.39	21.81	31.82	42.19	53.46	66.06	78.34	91.30	104.81
Median	6	13	21	31.5	42	53	65	77	90	105
Standard Deviation	2.49	4.42	6.39	8.25	9.99	11.94	14.09	15.54	17.54	19.22

Table 7.1: Statistics on the number of iterations it took for Algorithm 5 to terminate using random $(C, d) \in \mathrm{Sym}_n(\mathbb{R}) \times \mathbb{R}^n$ in a trial of $1,000$.

Now, once we run Algorithm 5 and return $x \in \{-1, 1\}^n$, we have a prospective spectrahedron that (C, d) may live in. This spectrahedron is namely $\{(C, d) : H(x) \succ 0\}$ where $H(x)$ is as defined in (7.4.2). The natural next step is to check if (C, d) lies in this spectrahedron. If it does then we can conclude that not only is the Shor relaxation of this quadratic binary program exact, but it has optimal solution x. Therefore, Algorithm 5 gives a natural way to solve (QBP) for some objective functions.

The complexity of Algorithm 6 comes from the complexity of Algorithm 5 as well as the number of times we need to repeat Step 3. Since we suspect the complexity of Algorithm 5 is linear in n, we suspect the main source of complexity will come from the latter issue. We formalize this with the following two conjectures obtained via empirical analysis.

Conjecture 7.4.8. *For fixed $x \in \{-1, 1\}^n$, P_x intersects $P_{x'}$ nontrivially for $2^n - n$ different $x' \in \{-1, 1\}^n$. Namely, it intersects all $P_{x'}$ for all x' that differ from x in more than one place.*

Conjecture 7.4.9. *For some fixed $(C, d) \in P_x$, $x \in \{-1, 1\}^n$, $(C, d) \in P_{x'}$ for at most n other $x' \in \{-1, 1\}^n$.*

These two conjectures give us conflicting messages of hope. On the one hand, Conjecture 7.4.8 tells us that even if we know $(C, d) \in P_x$, there are still exponentially many other x' that we would need to consider. On the other hand, Conjecture 7.4.9 tells us that for any (C, d) there are at most $n + 1$ different P_x that if could live in which then implies there are at most $n + 1$ spectrahedra we

Algorithm 6 Algorithm to find $x \in \{-1, 1\}^n$ such that $H(x) \succ 0$

Input: Objective function $(C, d) \in \mathrm{Sym}_n(\mathbb{R}) \times \mathbb{R}^n$

Output: $x \in \{-1, 1\}^n$ such that $H(x) \succ 0$ or a message that the SDP relaxation of (QBP) with objective function (C, d) is not exact

1. Initialize the set $S = \{-1, 1\}^n$
2. Run Algorithm 5 to find $x \in \{-1, 1\}^n$ such that $(C, d) \in P_x$
3. Check if $H(x) \succ 0$. If it is, return x and a message that says that x is the global optimum to (QBP). Else, remove all points $x \in \{-1, 1\}^n$ that Algorithm 5 considered from the set S
4. Repeat Step 2 initializing at a new random point $x' \in S$, terminating if Algorithm 5 ever considers a point $x \notin S$
5. If $S = \emptyset$, return the message that the SDP relaxation of (QBP) is not exact

need to consider. Since the problem of determining if the SDP relaxation is exact for this class of problems is NP hard, we suspect that even if we can prove a positive answer to Conjecture 7.4.9, determining for which x, $(C, d) \in P_x$ will still be quite challenging.

LIST OF REFERENCES

[1] Alisha Zachariah, Zachary Charles, Nigel Boston, and Bernard Lesieutre. Distributions of the number of solutions to the network power flow equations. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2018. 4, 5, 16

[2] Dhagash Mehta, Hung D. Nguyen, and Konstantin Turitsyn. Numerical polynomial homotopy continuation method to locate all the power flow solutions. *IEEE Transactions on Power Systems*, 10(12):2972–2980, 2016. 5, 10, 19

[3] Daniel K. Molzahn, Bernard C. Lesieutre, and Heng Chen. Counterexample to a continuation-based algorithm for finding all power flow solutions. *IEEE Transactions on Power Systems*, 28(1):564–565, 2013. 5, 32, 33

[4] Bernard Lesieutre and Dan Wu. An efficient method to locate all the load flow solutions - revisited. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 381–388, 2015. 5, 32, 33

[5] Weimin Ma and James S. Thorp. An efficient algorithm to locate all the load flow solutions. *IEEE Transactions on Power Systems*, 8(3):1077–1083, 1993. 5, 19, 32, 33

[6] Fathi M. Salam, Lionel Ni, Shixiong Guo, and Xian-He Sun. Parallel processing for the load flow of power systems: the approach and applications. In *Proceedings of the 28th IEEE Conference on Decision and Control,*, pages 2173–2178, 1989. 5, 10

[7] Carlos J. Tavora and Otto J. M. Smith. Stability analysis of power systems. *IEEE Transactions on Power Apparatus and Systems*, 3(3):1138–1144, 1972. 5

[8] Daniel Wu. Algebraic set preserving mappings for electric power grid models and its applications. *UW-Madison Electrical and Computer Engineering PhD Thesis*, 2017. 5

[9] Julia Lindberg, Nigel Boston, and Bernard C. Lesieutre. Exploiting symmetry in the power flow equations using monodromy. *ACM Commun. Comput. Algebra*, 54(3):100–104, 2020. 5, 12

[10] John Baillieul and Christopher I. Byrnes. Geometric critical point analysis of lossless power system models. *IEEE Transactions on Circuits and Systems*, 29(11):724–737, 1982. 5, 29

[11] Tianran Chen, Robert Davis, and Dhagash Mehta. Counting equilibria of the Kuramoto model using birationally invariant intersection index. *SIAM J. Appl. Algebra Geom.*, 2(4):489–507, 2018. 5, 11, 28, 29, 31

[12] Tianran Chen and Dhagash Mehta. On the network topology dependent solution count of the algebraic load flow equations. *IEEE Transactions on Power Systems*, 33(2):1451–1460, 2018. 5

[13] Daniel K. Molzahn, Dhagash Mehta, and Matthew Niemerg. Toward topologically based upper bounds on the number of power flow solutions. In *2016 American Control Conference (ACC)*, pages 5927–5932, 2016. 5

[14] Bernard Lesieutre, Julia Lindberg, Alisha Zachariah, and Nigel Boston. On the distribution of real-valued solutions to the power flow equations. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 165–170, 2019. 5

[15] Julia Lindberg, Alisha Zachariah, Nigel Boston, and Bernard Lesieutre. The geometry of real solutions to the power flow equations. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 596–603, 2018. 5

[16] Bernd Sturmfels. *Solving systems of polynomial equations*, volume 97 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 2002. 5

[17] Andrew J. Sommese and Charles W. Wampler. *The numerical solution of systems of polynomials*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005. Arising in engineering and science. 5, 6, 9

[18] Tien Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 399–436. Cambridge Univ. Press, Cambridge, 1997. 5

[19] John C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, Ltd., Chichester, 2003. 6

[20] David N. Bernstein. The number of roots of a system of equations. *Funkcional. Anal. i Priložen.*, 9(3):1–4, 1975. 6, 8

[21] Anatoli G. Kouchnirenko. Polyèdres de Newton et nombres de Milnor. *Invent. Math.*, 32(1):1–31, 1976. 6, 8

[22] Askold G. Khovanskii. Newton polyhedra, and the genus of complete intersections. *Funktsional. Anal. i Prilozhen.*, 12(1):51–61, 1978. 6, 8

[23] Ioannis Emiris. An efficient computation of mixed volume. Technical Report UCB/CSD-93-734, EECS Department, University of California, Berkeley, Apr 1993. 7

[24] Birkett Huber and Bernd Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comp.*, 64(212):1541–1555, 1995. 8, 107

[25] Ruchira S. Datta. Finding all Nash equilibria of a finite game using polynomial algebra. *Econom. Theory*, 42(1):55–96, 2010. 9

[26] Charles W. Wampler, Alexander P. Morgan, and Andrew J. Sommese. Complete solution of the nine-point path synthesis problem for four-bar linkages. *Journal of Mechanical Design*, 114(1):153–159, 03 1992. 9

[27] Joseph David Kileel. *Algebraic Geometry for Computer Vision*. ProQuest LLC, Ann Arbor, MI, 2017. Thesis (Ph.D.)–University of California, Berkeley. 9

[28] Jiawang Nie and Kristian Ranestad. Algebraic degree of polynomial optimization. *SIAM J. Optim.*, 20(1):485–502, 2009. 9

[29] Paul Breiding and Sascha Timme. Homotopycontinuation.jl: A package for homotopy continuation in Julia. In *Mathematical Software – ICMS 2018*, pages 458–465. Springer International Publishing, 2018. 9, 13, 100, 103

[30] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013. 9

[31] Anton Leykin and Jan Verschelde. Interfacing with the numerical homotopy algorithms in PHCpack. In *Mathematical software—ICMS 2006*, volume 4151 of *Lecture Notes in Comput. Sci.*, pages 354–360. Springer, Berlin, 2006. 9

[32] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA J. Numer. Anal.*, 39(3):1421–1446, 2019. 9, 11

[33] Carlos Améndola, Julia Lindberg, and Jose Israel Rodriguez. Solving parameterized polynomial systems with decomposable projections. *arXiv:1612.08807*, 2021. 9, 29, 87, 94, 95, 100

[34] Abraham Martín del Campo and Jose Israel Rodriguez. Critical points via monodromy and local methods. *J. Symbolic Comput.*, 79(part 3):559–574, 2017. 9

[35] Jonathan D. Hauenstein, Jose Israel Rodriguez, and Frank Sottile. Numerical computation of Galois groups. *Foundations of Computational Mathematics*, Jun 2017. 10

[36] Tianran Chen and Robert Davis. A toric deformation method for solving Kuramoto equations. *arXiv preprint arXiv:1810.05690*, 2018. 11

[37] Owen Coss, Jonathan D. Hauenstein, Hoon Hong, and Daniel K. Molzahn. Locating and counting equilibria of the Kuramoto model with rank-one coupling. *SIAM J. Appl. Algebra Geom.*, 2(1):45–71, 2018. 14

[38] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler, II. Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.*, 46(2):722–746, 2008. 15

[39] Mervin E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2(4):19–20, apr 1959. 16

[40] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Ann. Math. Statist.*, 27:642–669, 1956. 16

[41] Fabrice Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Appl. Algebra Engrg. Comm. Comput.*, 9(5):433–461, 1999. 19

[42] Mark Kac. On the average number of real roots of a random algebraic equation. *Bull. Amer. Math. Soc.*, 49:314–320, 1943. 20

[43] Michael Shub and Stephen Smale. Complexity of Bezout's theorem. II. Volumes and probabilities. In *Computational algebraic geometry (Nice, 1992)*, volume 109 of *Progr. Math.*, pages 267–285. Birkhäuser Boston, Boston, MA, 1993. 20

[44] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics*. Addison-Wesley Publishing Company, Reading, MA, second edition, 1994. A foundation for computer science. 24, 25

[45] Henry W. Gould. Table for fundamentals of series: Part I: Basic properties of series and products, 2011. 24

[46] Enrique Mallada, Randy A Freeman, and Ao Kevin Tang. Distributed synchronization of heterogeneous oscillators on networks with arbitrary topology. *IEEE Transactions on Control of Network Systems*, 3(1):12–23, 2015. 26

[47] Anton Leykin, Jose Israel Rodriguez, and Frank Sottile. Trace test. *Arnold Math. J.*, 4(1):113–125, 2018. 29

[48] Jonathan D. Hauenstein and Jose Israel Rodriguez. Multiprojective witness sets and a trace test. *Adv. Geom.*, 20(3):297–318, 2020. 29

[49] Jonathan D. Hauenstein and Samantha N. Sherman. Using monodromy to statistically estimate the number of solutions. In William Holderbaum and J. M. Selig, editors, *2nd IMA Conference on Mathematics of Robotics*, pages 37–46, Cham, 2022. Springer International Publishing. 29

[50] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, second edition, 2013. 34

[51] Peng Hui Tan and L.K. Rasmussen. The application of semidefinite programming for detection in CDMA. *IEEE Journal on Selected Areas in Communications*, 19(8):1442–1449, 2001. 37

[52] Svatopluk Poljak, Franz Rendl, and Henry Wolkowicz. A recipe for semidefinite relaxation for $(0, 1)$-quadratic programming. *J. Global Optim.*, 7(1):51–73, 1995. 37

[53] Daniel K. Molzahn and Ian A. Hiskens. A survey of relaxations and approximations of the power flow equations. *Foundations and Trends in Electric Energy Systems*, 4(1-2):1–221, 2019. 37

[54] Stephen A. Vavasis. Quadratic programming is in NP. *Inform. Process. Lett.*, 36(2):73–77, 1990. 37

[55] Naum Z. Shor. Quadratic optimization problems. *Izv. Akad. Nauk SSSR Tekhn. Kibernet.*, (1):128–139, 222, 1987. 37

[56] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004. 38

[57] Andrew Makhorin. GLPK (GNU linear programming kit).
Available at `http://www.gnu.org/software/glpk/glpk.html`. 45

[58] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. 45

[59] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020. 47

[60] Pierre Delforge and Josh Whitney. Data center efficiency assessment-scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers. *Natural Resources Defense Council*, 2014. 47

[61] Yevgeniy Sverdlik. Analysts: There are now more than 500 hyperscale data centers in the world, October 2017. `https://bit.ly/3jx8lsz`. 47

[62] Google. Google environmental report 2019. Technical report, Google, 2019. `https://services.google.com/fh/files/misc/google_2019-environmental-report.pdf`. 47

[63] Justine Calma. Amazon boosts climate commitments and greenhouse gas emissions, June 2020. https://bit.ly/2Sr4Vfc. 47

[64] Ana Radovanovic. Our data centers now work harder when the sun shines and wind blows, April 2020. Google Official Blog, https://bit.ly/3cXPAMu. 47

[65] Lancium. With 5th patent, lancium powers ahead in fast-ramping data center innovation. https://lancium.com/lancium-smart-response-patent-press-release/. 47

[66] Fan Yang and Andrew A Chien. Zccloud: Exploring wasted green power for high-performance computing. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1051–1060. IEEE, 2016. 47

[67] Andrew A Chien. Zero-carbon compute project. 2019. https://bit.ly/34m68dh. 47

[68] Wei Deng, Fangming Liu, Hai Jin, Bo Li, and Dan Li. Harnessing renewable energy in cloud datacenters: opportunities and challenges. *IEEE Network*, 28(1):48–55, 2014. 47

[69] Adam Wierman, Zhenhua Liu, Iris Liu, and Hamed Mohsenian-Rad. Opportunities and challenges for data center demand response. In *Int. Green Comp. Conf.*, pages 1–10, 2014. 47

[70] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. Data center demand response: avoiding the coincident peak via workload shifting and local generation. In *ACM Int. Conf. on Measurement and Modeling of Computer Systems*, 2013. 47

[71] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. Pricing data center demand response. In *ACM Int. Conf. on Measurement and Modeling of Computer Systems*, 2014. 47

[72] Zhi Zhou, Fangming Liu, and Zongpeng Li. Bilateral electricity trade between smart grids and green datacenters: Pricing models and performance evaluation. *IEEE Journal on Selected Areas in Communications*, 34(12):3993–4007, 2016. 47

[73] Zhi Zhou, Fangming Liu, Shutong Chen, and Zongpeng Li. A truthful and efficient incentive mechanism for demand response in green datacenters. *IEEE Transactions on Parallel and Distributed Systems*, 31(1):1–15, 2020. 47

[74] Zhi Zhou, Fangming Liu, Zongpeng Li, and Hai Jin. When smart grid meets geo-distributed cloud: An auction approach to datacenter demand response. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2650–2658, 2015. 47

[75] Shutong Chen, Lei Jiao, Lin Wang, and Fangming Liu. An online market mechanism for edge emergency demand response via cloudlet control. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 2566–2574, 2019. 47

[76] Lei Rao, Xue Liu, Le Xie, and Wenyu Liu. Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment. In *2010 Proc. IEEE INFOCOM*, pages 1–9, 2010. 47

[77] Jie Li, Zhen Bao, and Zuyi Li. Modeling demand response capability by internet data centers processing batch computing jobs. *IEEE Trans. Smart Grid*, 6(2):737–747, 2014. 47

[78] Lei Rao, Xue Liu, M. D. Ilic, and Jie Liu. Distributed coordination of internet data centers under multiregional electricity markets. *Proceedings of the IEEE*, 100(1):269–282, 2011. 47, 51

[79] Hui Dou, Yong Qi, Wei Wei, and Houbing Song. Carbon-aware electricity cost minimization for sustainable data centers. *IEEE Transactions on Sustainable Computing*, 2(2):211–223, 2017. 47

[80] Yasmine Abdennadher, Julia Lindberg, Bernard C. Lesieutre, and Line Roald. Carbon efficient placement of data center locations. 2022. 47

[81] Weiqi Zhang, Line A. Roald, Andrew A. Chien, John R. Birge, and Victor M. Zavala. Flexibility from networks of data centers: A market clearing formulation with virtual links. *21st Power Systems Computation Conference*, 2020. 47

[82] Íngo Goiri, William Katsak, Kien Le, Thu D Nguyen, and Ricardo Bianchini. Parasol and greenswitch: Managing datacenters powered by renewable energy. *ACM SIGPLAN Notices*, 48(4):51–64, 2013. 47

[83] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pages 175–186, 2012. 47

[84] Kibaek Kim, Fan Yang, Victor M. Zavala, and Andrew A. Chien. Data centers as dispatchable loads to harness stranded power. *IEEE Trans. Sustain. Energy*, 8(1):208–218, 2016. 47

[85] Jiajia Zheng, Andrew A Chien, and Sangwon Suh. Mitigating curtailment and carbon emissions through load migration between data centers. *Joule*, 2020. 47, 48, 51, 52

[86] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. Greening geographical load balancing. *SIGMETRICS Perform. Eval. Rev.*, 39(1):193–204, jun 2011. 48

[87] Andrew A Chien, Richard Wolski, and Fan Yang. The zero-carbon cloud: High-value, dispatchable demand for renewable power generators. *The Electricity Journal*, 28(8):110–118, 2015. 48

[88] Fan Yang and Andrew A Chien. Large-scale and extreme-scale computing with stranded green power: Opportunities and costs. *IEEE Transactions on Parallel and Distributed Systems*, 29(5):1103–1116, 2017. 48

[89] Tomorrow. Data driven climate action. Technical report, `https://www.electricitymap.org`, 2020. 48, 51

[90] CAISO. Today's outlook: Emissions, 2021. 48, 51

[91] CAISO. Managing oversupply, 2021. 48

[92] Kyle Siler-Evans, Ines Lima Azevedo, and M Granger Morgan. Marginal emissions factors for the us electricity system. *Environmental science & technology*, 46(9):4742–4748, 2012. 48

[93] Duncan S Callaway, Meredith Fowlie, and Gavin McCormick. Location, location, location: The variable value of renewable energy and demand-side efficiency resources. *Journal of the Association of Environmental and Resource Economists*, 5(1):39–75, 2018. 48

[94] Pablo Ruiz and Aleksandr Rudkevich. Analysis of marginal carbon intensities in constrained power networks. pages 1 – 9, 02 2010. 48

[95] Aleksandr Rudkevich, Pablo Ruiz, and Rebecca C. Carroll. Locational carbon footprint and renewable portfolio policies: A theory and its implications for the eastern interconnection of the us. pages 1 – 12, 02 2011. 48

[96] Richard Tabors, Aleksandr Rudkevich, Hua He, Ninad Kumthekar, Xindi Li, Robert Bland, Garret Quist, and Joshua LaPenna. Methodology for calculation of the marginal emission rates from a complex cogeneration facility compared with that of the co-located ny iso bus. *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021. 48

[97] Richard D Christie, Bruce F Wollenberg, and Ivar Wangensteen. Transmission management in the deregulated environment. *Proceedings of the IEEE*, 88(2):170–195, 2000. 48

[98] Eugene Litvinov. Design and operation of the locational marginal prices-based electricity markets. *Generation, Transmission and Distribution, IET*, 4:315 – 323, 03 2010. 48

[99] Julia Lindberg, Line Roald, and Bernard Lesieutre. The environmental potential of hyperscale data centers: Using locational marginal CO2 emissions to guide geographical load shifting. *Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)*. 49, 52, 55

[100] Alexis I. Aravanis, Artemis Voulkidis, Jaume Salom, Jacinta Townley, Vasiliki Georgiadou, Ariel Oleksiak, Milagros Rey Porto, Fabrice Roudet, and Theodore Zahariadis. Metrics for assessing flexibility and sustainability of next generation data centers. In *2015 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2015. 55

[101] Clayton Barrows, Eugene Preston, Andrea Staid, Gord Stephen, Jean-Paul Watson, Aaron Bloom, Ali Ehlen, Jussi Ikaheimo, Jennie Jorgenson, Dheepak Krishnamurthy, Jessica Lau, Brendan McBennett, and Matthew O'Connell. The ieee reliability test system: A proposed 2019 update. *IEEE Transactions on Power Systems*, 35(1):119–127, 2020. 57, 63

[102] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. 71

[103] Shuo Zhou, Howard Bondell, Antoinette Tordesillas, Benjamin I. P. Rubinstein, and James Bailey. Early identification of an impending rockslide location via a spatially-aided Gaussian mixture model. *Ann. Appl. Stat.*, 14(2):977–992, 2020. 71

[104] Amit Singhal, Pushpendra Singh, Brejesh Lall, and Shiv Dutt Joshi. Modeling and prediction of COVID-19 pandemic using Gaussian mixture model. *Chaos Solitons Fractals*, 138:110023, 8, 2020. 71

[105] Bach Do and Makoto Ohsaki. Gaussian mixture model for robust design optimization of planar steel frames. *Struct. Multidiscip. Optim.*, 63(1):137–160, 2021. 71

[106] Douglas Alan Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification*. PhD thesis, Georgia Institute of Technology, 1992. 71

[107] Yaxin Zhang, Mike Alder, and Roberto Togneri. Using Gaussian mixture modeling in speech recognition. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages I–613. IEEE, 1994. 71

[108] Fei Sha and Lawrence K. Saul. Large margin Gaussian mixture modeling for phonetic classification and recognition. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages 265–268, 2006. 71

[109] Konstantinos Blekas, Aristidis Likas, Nikolaos Galatsanos, and Isaac Lagaris. A spatially constrained mixture model for image segmentation. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 16:494–8, 04 2005. 71

[110] Dan Hosseinzadeh and Sridhar Krishnan. Gaussian mixture modeling of keystroke patterns for biometric applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38:816 – 826, 12 2008. 71

[111] Henry Teicher. Identifiability of finite mixtures. *The Annals of Mathematical statistics*, pages 1265–1269, 1963. 73

[112] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977. 74

[113] Carlos Améndola, Mathias Drton, and Bernd Sturmfels. Maximum likelihood estimates for Gaussian mixtures are transcendental. In *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 579–590. Springer, 2015. 74

[114] Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of EM suffice for mixtures of two Gaussians. In *Conference on Learning Theory*, pages 704–710, 2017. 74, 92, 102

[115] Yudong Chen and Xumei Xi. Likelihood landscape and local minima structures of Gaussian mixture models. *arXiv:2009.13040*, 2020. 74, 92, 102

[116] Ji Xu, Daniel J. Hsu, and Arian Maleki. Global analysis of Expectation Maximization for mixtures of two Gaussians. In *Advances in Neural Information Processing Systems*, volume 29, pages 2676–2684, 2016. 74, 92, 102

[117] Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *Ann. Statist.*, 46(6A):2747–2774, 2018. 74, 102

[118] T. Tony Cai, Jing Ma, and Linjun Zhang. CHIME: clustering of high-dimensional Gaussian mixtures with EM algorithm and its optimality. *Ann. Statist.*, 47(3):1234–1267, 2019. 74, 87

[119] Arora Sanjeev and Ravi Kannan. Learning mixtures of arbitrary Gaussians. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 247–257, 2001. 74

[120] Sara Shirinkam, Adel Alaeddini, and Elizabeth Gross. Identifying the number of components in Gaussian mixture models using numerical algebraic geometry. *J. Algebra Appl.*, 19(11):2050204, 21, 2020. 74

[121] Hassan Ashtiani, Shai Ben-David, Nicholas J. A. Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. Near-optimal sample complexity bounds for robust learning of Gaussian mixtures via compression schemes. *J. ACM*, 67(6):Art. 32, 42, 2020. 74

[122] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. 74

[123] Ainesh Bakshi, Ilias Diakonikolas, Samuel B. Hopkins, Daniel Kane, Sushrut Karmalkar, and Pravesh K. Kothari. Outlier-robust clustering of Gaussians and other non-spherical mixtures. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science—FOCS 2020*, pages 149–159. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020. 74

[124] Ilias Diakonikolas, Samuel B Hopkins, Daniel Kane, and Sushrut Karmalkar. Robustly learning any clusterable mixture of Gaussians. *arXiv:2005.06417*, 2020. 74

[125] Daniel M. Kane. Robust learning of mixtures of Gaussians. In *SODA 2021: ACM-SIAM Symposium on Discrete Algorithms (SODA21)*, pages 1246–1258, 2021. 75

[126] Allen Liu and Ankur Moitra. Settling the robust learnability of mixtures of Gaussians. *arXiv:2011.03622*, 2020. 75

[127] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica*, 50(4):1029–1054, 1982. 75

[128] Yihong Wu and Pengkun Yang. Optimal estimation of Gaussian mixtures via denoised method of moments. *Ann. Statist.*, 48(4):1981–2007, 2020. 75

[129] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Disentangling Gaussians. *Communications of The ACM*, 55(2):113–120, 2012. 75, 101

[130] Carlos Améndola, Kristian Ranestad, and Bernd Sturmfels. Algebraic identifiability of Gaussian mixtures. *Int. Math. Res. Not. IMRN*, (21):6556–6580, 2018. 75, 77, 102

[131] Sidney J Yakowitz and John D Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, pages 209–214, 1968. 75

[132] Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. *SIAM J. Comput.*, 44(4):889–911, 2015. 75

[133] Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society A*, 185:71–110, 1894. 75, 100

[134] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two Gaussians. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 553–562. ACM, New York, 2010. 76

[135] Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*, pages 93–102. IEEE Computer Soc., Los Alamitos, CA, 2010. 76

[136] Larry Wasserman. *All of statistics*. Springer Texts in Statistics. Springer-Verlag, New York, 2004. A concise course in statistical inference. 77

[137] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, pages 134–183. Lecture Notes in Comput. Sci., Vol. 33. 1975. 77

[138] Pinaki Mondal. How many zeroes? counting the number of solutions of systems of polynomials via geometry at infinity. *arXiv:1806.05346*, 2020. 79

[139] Robin Hartshorne. *Algebraic geometry*. Springer-Verlag, New York-Heidelberg, 1977. Graduate Texts in Mathematics, No. 52. 83, 109, 115

[140] Carlos Améndola, Jean-Charles Faugère, and Bernd Sturmfels. Moment varieties of Gaussian mixtures. *J. Algebr. Stat.*, 7(1):14–28, 2016. 94, 100

[141] Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139, 1918. 97

[142] Daniel Lazard. Injectivity of real rational mappings: the case of a mixture of two Gaussian laws. *Math. Comput. Simulation*, 67(1-2):67–84, 2004. 100

[143] Paul Breiding, Kemal Rose, and Sascha Timme. Certifying zeros of polynomial systems using interval arithmetic. *arXiv:2011.05000*, 2020. 100

[144] Kavish Gandhi and Yonah Borns-Weil. Moment-based learning of mixture distributions. *https://math.mit.edu/research/undergraduate/spur/documents/2016Gandhi-Borns-Weil.pdf*, 2016. 101

[145] Fabrizio Catanese, Serkan Hoşten, Amit Khetan, and Bernd Sturmfels. The maximum likelihood degree. *Amer. J. Math.*, 128(3):671–697, 2006. 106

[146] Serkan Hoşten, Amit Khetan, and Bernd Sturmfels. Solving the likelihood equations. *Found. Comput. Math.*, 5(4):389–407, 2005. 106, 110

[147] June Huh. The maximum likelihood degree of a very affine variety. *Compos. Math.*, 149(8):1245–1266, 2013. 106

[148] Jose Israel Rodriguez and Botong Wang. The maximum likelihood degree of mixtures of independence models. *SIAM J. Appl. Algebra Geom.*, 1(1):484–506, 2017. 107

[149] Carlos Améndola, Nathan Bliss, Isaac Burke, Courtney R. Gibbons, Martin Helmer, Serkan Hoşten, Evan D. Nash, Jose Israel Rodriguez, and Daniel Smolkin. The maximum likelihood degree of toric varieties. *J. Symbolic Comput.*, 92:222–242, 2019. 107

[150] Patrick Clarke and David A. Cox. Moment maps, strict linear precision, and maximum likelihood degree one. *Adv. Math.*, 370:107233, 51, 2020. 107

[151] Elizabeth Gross, Mathias Drton, and Sonja Petrović. Maximum likelihood degree of variance component models. *Electron. J. Stat.*, 6:993–1016, 2012. 107

[152] Mateusz Michał ek, Leonid Monin, and Jarosław A. Wiśniewski. Maximum likelihood degree, complete quadrics, and $\mathbb{C}^*$-action. *SIAM J. Appl. Algebra Geom.*, 5(1):60–85, 2021. 107

[153] Bernd Sturmfels, Sascha Timme, and Piotr Zwiernik. Estimating linear covariance models with numerical nonlinear algebra. *Algebr. Stat.*, 11(1):31–52, 2020. 107

[154] Jan Verschelde, Pierre Verlinden, and Ronald Cools. Homotopies exploiting Newton polytopes for solving sparse polynomial systems. *SIAM J. Numer. Anal.*, 31(3):915–930, 1994. 107

[155] Bernd Sturmfels. *Solving systems of polynomial equations*, volume 97 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 2002. 107

[156] Taylor Brysiewicz, Jose Israel Rodriguez, Frank Sottile, and Thomas Yahl. Solving decomposable sparse systems. *Numer. Algorithms*, 88(1):453–474, 2021. 107

[157] Jan Draisma, Emil Horobeţ, Giorgio Ottaviani, Bernd Sturmfels, and Rekha Thomas. The Euclidean distance degree. In *SNC 2014—Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, pages 9–16. ACM, New York, 2014. 107

[158] Jan Draisma, Emil Horobeţ, Giorgio Ottaviani, Bernd Sturmfels, and Rekha R. Thomas. The Euclidean distance degree of an algebraic variety. *Found. Comput. Math.*, 16(1):99–149, 2016. 107

[159] Jasmijn A. Baaijens and Jan Draisma. Euclidean distance degrees of real algebraic groups. *Linear Algebra Appl.*, 467:174–187, 2015. 107

[160] Hwangrae Lee. The Euclidean distance degree of Fermat hypersurfaces. *J. Symbolic Comput.*, 80(part 2):502–510, 2017. 107

[161] Dmitriy Drusvyatskiy, Hon-Leung Lee, Giorgio Ottaviani, and Rekha R. Thomas. The Euclidean distance degree of orthogonally invariant matrix varieties. *Israel J. Math.*, 221(1):291–316, 2017. 107

[162] Paolo Aluffi and Corey Harris. The Euclidean distance degree of smooth complex projective varieties. *Algebra Number Theory*, 12(8):2005–2032, 2018. 107

[163] Laurentiu G. Maxim, Jose I. Rodriguez, and Botong Wang. Euclidean distance degree of the multiview variety. *SIAM J. Appl. Algebra Geom.*, 4(1):28–48, 2020. 107

[164] Paul Breiding, Frank Sottile, and James Woodcock. Euclidean distance degree and mixed volume. *arXiv preprint arXiv:2012.06350*, 2020. 107

[165] Laurentiu G. Maxim, Jose Israel Rodriguez, and Botong Wang. Defect of Euclidean distance degree. *Adv. in Appl. Math.*, 121:102101, 22, 2020. 107

[166] Diego Cifuentes, Corey Harris, and Bernd Sturmfels. The geometry of SDP-exactness in quadratic optimization. *Math. Program.*, 182(1-2, Ser. A):399–428, 2020. 107

[167] Jiawang Nie and Kristian Ranestad. Algebraic degree of polynomial optimization. *SIAM J. Optim.*, 20(1):485–502, 2009. 107, 108

[168] Hans-Christian Graf von Bothmer and Kristian Ranestad. A general formula for the algebraic degree in semidefinite programming. *Bull. Lond. Math. Soc.*, 41(2):193–197, 2009. 108

[169] Jiawang Nie, Kristian Ranestad, and Bernd Sturmfels. The algebraic degree of semidefinite programming. *Math. Program.*, 122(2, Ser. A):379–405, 2010. 108

[170] Grigoriy Blekherman, Pablo A. Parrilo, and Rekha R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*, volume 13 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2013. 108

[171] Diego Cifuentes, Corey Harris, and Bernd Sturmfels. The geometry of SDP-exactness in quadratic optimization. *Math. Program.*, 182(1-2, Ser. A):399–428, 2020. 122, 127

[172] Monique Laurent and Svatopluk Poljak. On a positive semidefinite relaxation of the cut polytope. volume 223/224, pages 439–461. 1995. Special issue honoring Miroslav Fiedler and Vlastimil Pták. 134