MINIMUM-VOLUME ELLIPSOIDS

MOS-SIAM Series on Optimization

This series is published jointly by the Mathematical Optimization Society and the Society for Industrial and Applied Mathematics. It includes research monographs, books on applications, textbooks at all levels, and tutorials. Besides being of high scientific quality, books in the series must advance the understanding and practice of optimization. They must also be written clearly and at an appropriate level for the intended audience.

Series Volumes

Todd, Michael J., *Minimum-Volume Ellipsoids: Theory and Algorithms*

Bienstock, Daniel, *Electrical Transmission System Cascades and Vulnerability: An Operations Research Viewpoint*

Koch, Thorsten, Hiller, Benjamin, Pfetsch, Marc E., and Schewe, Lars, editors, *Evaluating Gas Network Capacities*

Corberán, Ángel, and Laporte, Gilbert, *Arc Routing: Problems, Methods, and Applications*

Toth, Paolo, and Vigo, Daniele, *Vehicle Routing: Problems, Methods, and Applications, Second Edition*

Beck, Amir, *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*

Attouch, Hedy, Buttazzo, Giuseppe, and Michaille, Gérard, *Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization, Second Edition*

Shapiro, Alexander, Dentcheva, Darinka, and Ruszczynski, Andrzej, *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*

Locatelli, Marco and Schoen, Fabio, *Global Optimization: Theory, Algorithms, and Applications*

De Loera, Jesús A., Hemmecke, Raymond, and Köppe, Matthias, *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*

Blekherman, Grigoriy, Parrilo, Pablo A., and Thomas, Rekha R., editors, *Semidefinite Optimization and Convex Algebraic Geometry*

Delfour, M. C., *Introduction to Optimization and Semidifferential Calculus*

Ulbrich, Michael, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*

Biegler, Lorenz T., *Nonlinear Programming: Concepts, Álgorithms, and Applications to Chemical Processes*

Shapiro, Alexander, Dentcheva, Darinka, and Ruszczynski, Andrzej, *Lectures on Stochastic Programming: Modeling and Theory*

Conn, Andrew R., Scheinberg, Katya, and Vicente, Luis N., *Introduction to Derivative-Free Optimization*

Ferris, Michael C., Mangasarian, Olvi L., and Wright, Stephen J., *Linear Programming with MATLAB*

Attouch, Hedy, Buttazzo, Giuseppe, and Michaille, Gérard, *Variational Analysis in Sobolev and BV Spaces: Applications to PDEs and Optimization*

Wallace, Stein W. and Ziemba, William T., editors, *Applications of Stochastic Programming*

Grötschel, Martin, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*

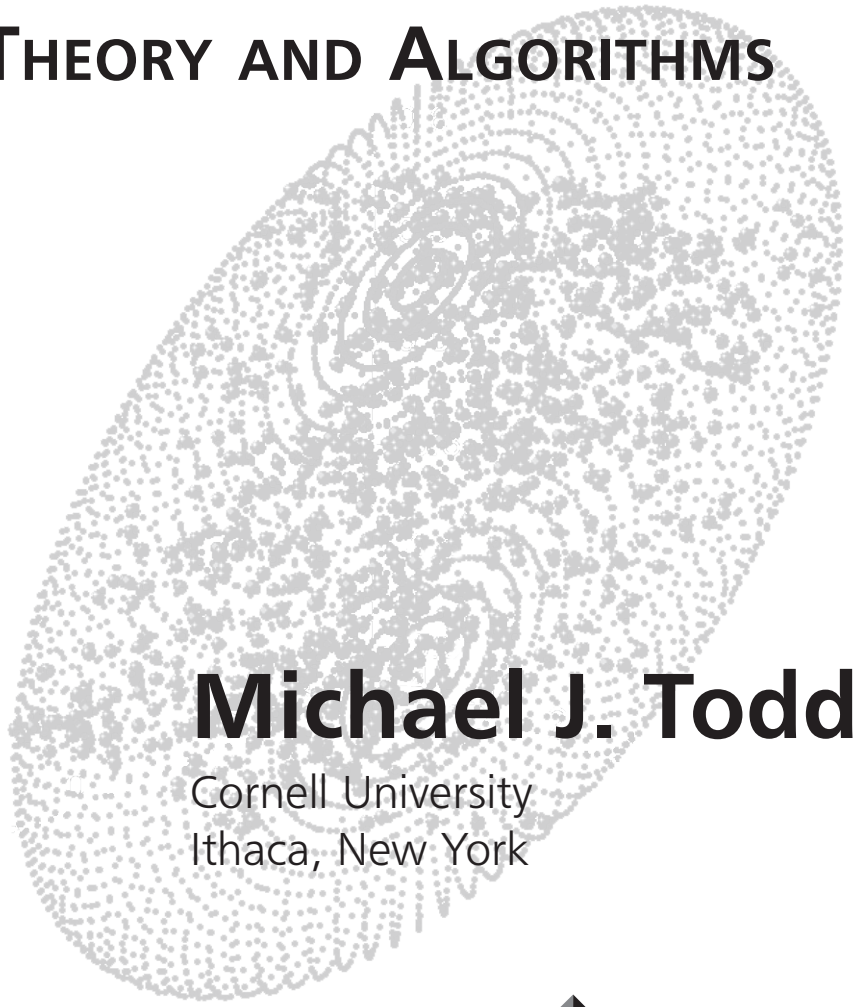
Renegar, James, *A Mathematical View of Interior-Point Methods in Convex Optimization*

Ben-Tal, Aharon and Nemirovski, Arkadi, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*

Conn, Andrew R., Gould, Nicholas I. M., and Toint, Phillippe L., *Trust-Region Methods*

MINIMUM-VOLUME ELLIPSOIDS

THEORY AND ALGORITHMS



Michael J. Todd

Cornell University
Ithaca, New York

Society for Industrial and Applied Mathematics
Philadelphia

Mathematical Optimization Society
Philadelphia

10 9 8 7 6 5 4 3 2 1

Publisher	David Marshall
Acquisitions Editor	Elizabeth Greenspan
Developmental Editor	Gina Rinelli Harris
Managing Editor	Kelly Thomas
Production Editor	Ann Manning Allen
Copy Editor	Nicola Howcroft
Production Manager	Donna Witzleben
Production Coordinator	Cally Shrader
Compositor	Techsetters, Inc.
Graphic Designer	Lois Sellers

Library of Congress Cataloging-in-Publication Data

Names: Todd, Michael J., 1947-
Title: Minimum-volume ellipsoids : theory and algorithms / Michael J. Todd, Cornell University, Ithaca, New York.
Other titles: Ellipsoids
Description: Philadelphia : Society for Industrial and Applied Mathematics : Mathematical Optimization Society, 2016. | Series: MOS-SIAM series on optimization ; 23 | Includes bibliographical references and index.
Identifiers: LCCN 2016012176 (print) | LCCN 2016016279 (ebook) | ISBN 9781611974379 | ISBN 9781611974386
Subjects: LCSH: Ellipse. | Cylinders.
Classification: LCC QA559 .T63 2016 (print) | LCC QA559 (ebook) | DDC 516/.156–dc23
LC record available at *https://lccn.loc.gov/2016012176*

This book is dedicated to the memory of
Leonid Khachiyan.

Contents

List of Figures

List of Algorithms

Preface

Optimization is concerned with choosing several variables to optimize (maximize or minimize) an objective function, usually subject to several constraints. In the last twenty-five years, there has been considerable interest in the case where the decision variables are the entries of a matrix, frequently required to be symmetric and positive semidefinite. If all remaining constraints and the objective function are linear, this leads to semidefinite programming. Such problems arise not only in standard matrix optimization problems, like minimizing the maximum singular value of a parametrized matrix (or the maximum eigenvalue in the symmetric case), but also in optimal control, in obtaining good approximate solutions to hard combinatorial problems, and in approximating optimal solutions of nonconvex optimization problems involving polynomials.

Traditional algorithms often use second-order approximations of the objective and constraint functions at each iteration to obtain an improved iterate. Such methods have attractive local convergence properties, usually at a quadratic or superlinear rate. However, as problems have grown in size—and particularly when the decision variables form a matrix—the construction, storage, and updating of a second-order approximation, as well as the linear algebra cost at each iteration of solving the corresponding optimization subproblem, become prohibitive. Hence there has been renewed interest in first-order methods, which scale well to such large problems.

This book studies particular matrix optimization problems, and first-order methods for solving them, in a very simple and geometrically appealing situation: finding a minimum-volume ellipsoid containing a set of points in Euclidean space. The matrix decision variables arise since an ellipsoid is defined by a symmetric positive definite matrix, and its volume is related to the determinant of that matrix. While this is a rather special problem, it provides a fundamental approach to data analysis of a large set of points in a high-dimensional space. It also arises in various problems in computational geometry and, rather surprisingly, in optimal design in statistics.

We will discuss formulations of this problem, duality results and optimality conditions, geometric properties of their optimal solutions, and in particular efficient first-order algorithms for their solution. We will see that the low iteration cost of these methods, and their analysis, rely on the beautiful properties of the "log determinant" function of a symmetric matrix, the formulae for updating the inverse and the determinant of such a matrix after a rank-one modification, and sensitivity analysis results on nonlinear optimization problems.

A closely associated problem asks for an ellipsoidal cylinder containing a set of points whose cross section in a certain coordinate subspace has minimum area. This problem also arises in computational geometry, and in a more general optimal design setting in statistics. We provide a theoretical and algorithmic analysis of this problem as well.

A final chapter addresses in a more abbreviated way a number of related problems: dealing with outliers, approximating by parallelotopes instead of ellipsoids, and ellipsoidal

approximation of a polyhedron given by linear inequalities rather than as a convex hull of points.

An intriguing aspect of this area is that algorithms were developed independently and simultaneously in two different disciplines and on either side of the Iron Curtain. Thus our basic algorithms were developed by Frank and Wolfe in the optimization community, and by Wynn and Atwood (in the West) and Fedorov (in the East) in the statistics community. Very similar methods were proposed in electrical engineering for system parameter identification. There is also a close relation to the ellipsoid algorithm in convex optimization. We try to add perspective to our discussion by including a "Notes and references" section at the end of each chapter. There are also mathematical connections to geometric functional analysis and to spectral sparsification in graph theory.

This book is aimed at graduate students in applied mathematics or operations research who are interested in matrix optimization problems or in first-order methods. I hope it will also be of interest to a variety of researchers in these and related fields. Whether the reader is concerned with these particular problems or not, we feel the techniques used and the connections made will prove instructive. For those who want to experiment with solving instances of these problems, we provide some computational results as well as MATLAB codes for the algorithms. They are listed in Appendix B and posted at `www.siam.org/books/mo23`.

The mathematical background required is quite modest: familiarity with linear algebra and real analysis suffices. We also include in Appendix A some basic material on positive (semi)definiteness, low-rank updates, matrix analysis, convexity, and optimality conditions and duality.

I would like to thank my collaborators on ellipsoid-related optimization over the years: Selin Damla Ahipaşaoğlu, Bruce Burrell, Leonid Khachiyan, Martin Larsson, Peng Sun, and Emre Alper Yıldırım. Special thanks go to Selin Damla Ahipaşaoğlu for her careful reading of the manuscript and for providing some of the figures, including the cover art.

This work was supported in part by NSF through grant DMS-0513337 and by ONR through grants N00014-02-1-0057 and N00014-08-1-0036.

Chapter 1
Introduction

This monograph is concerned with the problem of representing a (large) set of points in a (high-dimensional) Euclidean space by an ellipsoid, in some sense optimally. This might seem a very special problem, but we shall see that it provides a beautiful example of the interplay of ideas from optimization, convex analysis, geometry, and linear algebra. We will be concerned with the theory of this problem and its extensions, its applications, and in particular the development of efficient algorithms for its solution. The subject also provides a simple introduction to the study of matrix optimization problems where a matrix variable must be symmetric and positive semidefinite, which under the names of semidefinite programming (in the optimization community) and linear matrix inequalities (in control theory circles) has been a fast-growing and vibrant area over the last twenty years. On the other hand, it turns out that optimal ellipsoid containment problems have a long, if intermittent, history in convex geometry and optimization. Finally, the problems have numerous and wide-ranging applications in data analysis, computational geometry, and (through their duals) optimal experimental design in statistics.

In the next section, we discuss why we choose ellipsoids to make our approximations, while the following section shows how the resulting optimization problems can be formulated using the "logdet" function. Section 1.3 considers the closely related optimal design problem in statistics. We then give some applications of the ellipsoid approximation problem, after which Section 1.5 gives an outline of the rest of the book.

1.1 ▪ Why ellipsoids?

An ellipsoid is the affine image of a ball. We can picture it as the analog of an ellipse in two dimensions, or a (rugby or American) football or flying saucer in three. Here we shall argue that ellipsoids provide a good way to represent a more complicated convex set in a Euclidean space.

Suppose we are given a bounded polyhedral set $\mathbf{X} \subseteq \mathrm{R}^n$, described either as the convex hull of m points,

$$\mathbf{X} := \mathrm{conv}\{x_1, x_2, \ldots, x_m\}, \tag{1.1.1}$$

where each x_i lies in R^n, or as the set of solutions to m linear inequalities,

$$\mathbf{X} := \{x \in \mathrm{R}^n : Ax \leq b\}, \tag{1.1.2}$$

where $A \in \mathrm{R}^{m \times n}$ and $b \in \mathrm{R}^m$.

Each of these descriptions is reasonably compact, although it depends on the potentially huge parameter m as well as the merely large dimension n. In the first case, it is easy to optimize any linear function $c^T x$ cheaply over $\mathbf{X}$ just by computing the m inner products $c^T x_i$, but testing membership in $\mathbf{X}$ requires the solution of a linear programming problem. In the second case, membership only requires a matrix-vector product Ax and some comparisons, but optimizing a linear function over $\mathbf{X}$ again requires linear programming. We would like to find a set that represents $\mathbf{X}$ well in some sense, which allows easy tests for membership and linear optimization, and whose description is of a size depending only on n. We may be willing to invest a reasonable amount of computing time in order to obtain this representative set, but then we would like queries on the resulting set to be cheap.

A simple choice is just to take a random sample of $O(f(n))$ points from the x_i's in the first case, where f is a modestly growing function. This may work well statistically, but we have no guarantee that the sample will represent the whole set accurately. In the same vein, we could take the solution set for $O(f(n))$ linear inequalities, chosen at random from those representing $\mathbf{X}$ in the second set, with similar drawbacks. Instead, we will choose the set from a suitable family that has the minimum volume among all those that contain the set $\mathbf{X}$.

One family we could consider is that of (axis-aligned) boxes, sets of the form $\{x \in \mathrm{R}^n : l \leq x \leq u\}$, where l and u ($u \geq l$) are vectors in R^n. The minimum-volume enclosing box is relatively easy to compute (at the cost of $2n$ linear programming calculations in the second case), has a trivial description, and allows simple linear optimization and membership tests. In addition, boxes have the attractive property that they are robust with respect to the minimality criterion: the minimum-volume box is also the unique inclusion-minimal box. However, they do not fit certain data sets very well, and are very dependent on the coordinate representation of the vectors. To make this latter point more precise, we would like our representative sets to have the affine invariance property that, for any nonsingular affine transformation $\mathcal{A} : x \mapsto Mx + b$ for (nonsingular) $M \in \mathrm{R}^{n \times n}$ and $b \in \mathrm{R}^n$, if $\mathbf{Y}$ represents $\mathbf{X}$, then $\mathcal{A}(\mathbf{Y})$ represents $\mathcal{A}(\mathbf{X})$. Boxes unfortunately fail this requirement, as can easily be seen by considering a rotation of 45 degrees in the plane.

One way to circumvent this difficulty is to consider the set of all parallelotopes, which are affine images of boxes. Indeed, this family has some nice properties, and we will come back to it in the final chapter. However, parallelotopes do not fit convex sets quite as well as the bodies we will choose, and they certainly do not have smooth boundaries.

The choice we will make in this monograph is to consider the family of *ellipsoids*, that is, sets of the form

$$\mathcal{E}(H, \bar{x}) := \{x \in \mathrm{R}^n : (x - \bar{x})^T H (x - \bar{x}) \leq n\}, \tag{1.1.3}$$

where $\bar{x} \in \mathrm{R}^n$ is the *center* of the ellipsoid, and H is a symmetric positive definite matrix of order n, i.e., $H^T = H$ and $v^T H v > 0$ for all nonzero $v \in \mathrm{R}^n$. We say that H defines the *shape* (which we take to include the size) of the ellipsoid. We choose the right-hand side to be n to simplify some later analysis. Abusing notation slightly, we write $\mathcal{E}(H)$ to denote an ellipsoid centered at the origin with shape matrix H, so $\mathcal{E}(H)$ is an abbreviation of $\mathcal{E}(H, 0)$. As a simple example, if H is n times the identity matrix, this set is just the unit Euclidean ball centered at $\bar{x}$. Indeed, ellipsoids are just affine transformations of balls. To see this, let L be the Cholesky factor of H; that is, L is a lower triangular matrix with positive diagonal entries satisfying $H = LL^T$. (A review of linear algebra and some matrix analysis appears in Appendix A; in particular, we show in Section A.2 that every positive definite matrix H has a Cholesky factorization and a positive definite square root $H^{1/2}$

with $H^{1/2}H^{1/2} = H$.) Then x lies in $\mathcal{E}(H,\bar{x})$ iff $||L^T(x-\bar{x})|| \le \sqrt{n}$, so that

$$\mathcal{E}(H,\bar{x}) = \{x = \bar{x} + (\sqrt{n}L^{-T})z : z \in \mathrm{R}^n, ||z|| \le 1\}. \tag{1.1.4}$$

It is helpful to have some notation for symmetric matrices. We let $\mathcal{S}^k$ denote the space of real symmetric $k \times k$ matrices; $\mathcal{S}^k_+$ denotes the cone of positive semidefinite matrices in $\mathcal{S}^k$ (H with $v^T H v \ge 0$ for all v), while $\mathcal{S}^k_{++}$ denotes the cone of positive definite matrices in $\mathcal{S}^k$. If Y and Z lie in $\mathcal{S}^k$, we write $Y \succeq Z$ or $Z \preceq Y$ if $Y - Z$ is positive semidefinite, and $Y \succ Z$ or $Z \prec Y$ if $Y - Z$ is positive definite. In particular, $Y \succ 0$ ($Y \succeq 0$) denotes that Y is positive (semi)definite.

Let us check that ellipsoids satisfy our desired criteria. First, under the nonsingular affine transformation $\mathcal{A} : x \mapsto \hat{x} := Mx + b$ for some $M \in \mathrm{R}^{n\times n}$ and $b \in \mathrm{R}^n$, $\mathcal{E}(H,\bar{x})$ transforms to $\mathcal{E}(\hat{H},\hat{\bar{x}})$, where

$$\hat{H} := M^{-T} H M^{-1}, \quad \hat{\bar{x}} := \mathcal{A}(\bar{x}) = M\bar{x} + b.$$

Moreover, the volume of any (measurable) set is multiplied by $|\det M|$ under this transformation, which yields the affine invariance property. Ellipsoids can be compactly described by the n components of the center and the $n(n+1)/2$ entries in the lower triangle of H (or, perhaps more usefully, by the $n(n+1)/2$ nonzero entries of its Cholesky factor L). Membership can be trivially checked by computing $(x-\bar{x})^T H(x-\bar{x})$ or the norm of $L^T(x-\bar{x})$. Finally, we exhibit a closed-form solution to

$$\begin{array}{ll} \min_x & c^T x \\ & (x-\bar{x})^T H(x-\bar{x}) \quad \le \quad n, \end{array} \tag{1.1.5}$$

assuming $c \in \mathrm{R}^n$ is nonzero. For this, we use the Karush–Kuhn–Tucker optimality conditions, which are necessary (since the Slater condition holds) and sufficient for this convex problem. (A review of optimality conditions for nonlinear programming problems is given in Section A.6.) An optimal solution x must satisfy $c + 2\lambda H(x-\bar{x}) = 0$, where $\lambda \ge 0$ with equality unless the constraint holds with equality. Hence we see that $c^T x$ is minimized over $\mathcal{E}(H,\bar{x})$ at

$$\bar{x} - \sqrt{\frac{n}{c^T H^{-1} c}} H^{-1} c$$

and that the optimal value is

$$c^T \bar{x} - \sqrt{n\, c^T H^{-1} c}. \tag{1.1.6}$$

These are easy to compute if we have the Cholesky factorization of H, since $H^{-1}c = L^{-T}(L^{-1}c)$, and solving linear systems of the form $Lv = w$ or $L^T y = z$ is simple when L is triangular.

It is also worth mentioning that, whenever a random variable has a multivariate Gaussian distribution, the level sets of its probability distribution function will be ellipsoids, whose shape matrices are proportional to the inverse of the covariance matrix of the random variable. Further, as can easily be seen from (1.1.6), the set of ellipsoids containing the origin in their interiors is invariant under *polarity*: the polar $\mathbf{X}^\circ$ of a set $\mathbf{X}$ is defined as $\{z \in \mathrm{R}^n : z^T x \le 1 \text{ for all } x \in \mathbf{X}\}$.

We let

$$\mathcal{E}_*(\mathbf{X}) \text{ denote the minimum-volume ellipsoid containing } \mathbf{X}. \tag{1.1.7}$$

(We shall see later that this ellipsoid exists and is unique, at least in the case where $\mathbf{X}$ is the convex hull of a finite set of points.)

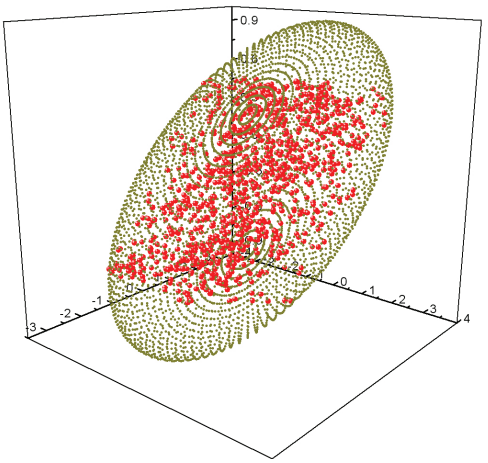


Figure 1.1. *Minimum-volume ellipsoid.*

We now present a remarkable result due to F. John [45] in 1948 that demonstrates the suitability of ellipsoids for fitting convex bodies. (A convex body in R^n is a convex compact set with nonempty interior.) To state the result, we need to define (with a slight abuse of notation) the *homothetic scaling* $\alpha\mathcal{E}$ of an ellipsoid $\mathcal{E}$ as the set scaled by α around its center $\bar{x}$:

$$\alpha\mathcal{E} := \{\bar{x} + \alpha z : \bar{x} + z \in \mathcal{E}\}.$$

Theorem 1.1. *Let* $\mathbf{X}$ *be a convex body in* R^n.

(a) *The homothetic scaling* $\frac{1}{n}\mathcal{E}_*(\mathbf{X})$ *is contained in* $\mathbf{X}$.

(b) *Further, if* $\mathbf{X}$ *is symmetric* ($-\mathbf{X} = \mathbf{X}$), *then* $\frac{1}{\sqrt{n}}\mathcal{E}_*(\mathbf{X})$ *is contained in* $\mathbf{X}$.

In the next chapter, we will give a proof of this result when $\mathbf{X}$ is the convex hull of a finite set of points.

John also implicitly showed that there is a finite subset of $\mathbf{X}$, of cardinality at most $n(n+3)/2$, such that the minimum-volume ellipsoid containing this subset of points is also the minimum-volume ellipsoid containing $\mathbf{X}$. This small subset, called a *core set*, in some sense represents $\mathbf{X}$ much better than a random subset of its points of comparable size.

An example showing a cloud of points in R^3 and the minimum-volume ellipsoid containing them is shown in Figure 1.1.

1.2 ▪ The minimum-volume enclosing ellipsoid problem

Our next task is to formulate the problem of finding the minimum-volume ellipsoid containing a set $\mathbf{X}$ that is given as in (1.1.1) as the convex hull of a finite set of points $x_1, x_2, \ldots, x_m$ in R^n. Sets of the form (1.1.2), given by linear inequalities, will not be considered again until the final chapter.

We saw in (1.1.4) that the ellipsoid $\mathcal{E}(H, \bar{x})$ could be written as an affine transformation of the unit ball, using the matrix $\sqrt{n}L^{-T}$, where L is the Cholesky factor of H. Thus its

volume is that of the unit ball times $|\det(\sqrt{n}L^{-T})|$; since $H = LL^T$, and so $\det H = (\det L)^2$, we have

$$\mathrm{vol}(\mathcal{E}(H,\bar{x})) = \frac{n^{n/2}\Omega_n}{\sqrt{\det H}}, \tag{1.2.1}$$

where Ω_n is the volume of a ball of radius 1 in $\mathbf{R}^n$. Hence to minimize the volume of an ellipsoid, we can equivalently minimize the negative of the logarithm of the determinant of its shape matrix. Recalling that we want H to be positive definite, we define the *logdet* function by the following.

Definition 1.2.

$$\mathrm{lndet}(H) := \begin{cases} \ln\det H & \text{if } H \text{ is positive definite,} \\ -\infty & \text{otherwise.} \end{cases}$$

Why do we introduce this seemingly superfluous logarithm? It turns out (see Section A.5) that $-\mathrm{lndet}$ is a *strictly convex* function on the space of symmetric matrices. Indeed, it suffices to show that the second directional derivative at any positive definite H in the direction of any symmetric nonzero E is positive, but Section A.5 shows that this is

$$\begin{aligned} (H^{-1}EH^{-1}) \bullet E &= \mathrm{Trace}(H^{-1}EH^{-1}E) \\ &= \mathrm{Trace}[(H^{-1/2}EH^{-1/2}H^{-1/2}EH^{-1/2})] \\ &= \|H^{-1/2}EH^{-1/2}\|_F^2 > 0. \end{aligned}$$

Here we have used the inner product $U \bullet V := \mathrm{Trace}(U^T V)$ of two similarly dimensioned matrices and the corresponding Frobenius norm $\|U\|_F := (U \bullet U)^{1/2}$, and the identity $\mathrm{Trace}(UV) = \mathrm{Trace}(VU)$ for any $m \times n$ U and $n \times m$ V (see Section A.1). The first directional derivative is also easy to evaluate: it is $-H^{-1} \bullet E$, and it is key to the efficiency of our algorithms that both the inverse (and the Cholesky factorization) and the value of the logdet function are simple to update if we make a rank-one update to H.

We therefore see that the problem of finding the minimum-volume ellipsoid containing $x_1, x_2, \ldots, x_m$ (and thus $\mathbf{X}$) can be formulated as

$$(P_1) \qquad \begin{array}{ll} \min_{H,\bar{x}} & -\mathrm{lndet}(H) \\ & (x_i - \bar{x})^T H (x_i - \bar{x}) \le n, \quad i = 1,2,\ldots,m. \end{array} \tag{1.2.2}$$

Note that here the variables are the symmetric $n \times n$ shape matrix H and the center n-vector $\bar{x}$, rather than the vector x as in problem (1.1.5) with very similar constraints. Indeed, here we are *designing* the ellipsoid, rather than optimizing over a fixed one. Also, we do not need to add the explicit constraint that H be positive definite, since this is implicit from the form of the objective function (and from our stipulation that the logdet function take the value negative infinity when its argument is not positive definite).

If $n = 1$, (P_1) is trivially solvable. We let $\hat{x}$ and $\check{x}$ denote the largest and smallest x_i, and then set $\bar{x} := (\hat{x} + \check{x})/2$ and $H := 4/(\hat{x} - \check{x})^2$. Hence we assume $n > 1$ in the following.

Finally, it is important to observe that, although the objective function of (P_1) is convex and its constraints are convex separately in H and in $\bar{x}$, they are not convex in H and $\bar{x}$ jointly because of the cross terms $-2x_i^T H\bar{x}$ and $\bar{x}^T H\bar{x}$. If we restrict ourselves to *centered* ellipsoids, with $\bar{x}$ fixed at the origin, we obtain the optimization problem

$$(P) \qquad \begin{array}{ll} \min_{H} & -\mathrm{lndet}(H) \\ & x_i^T H x_i \le n, \quad i = 1,2,\ldots,m. \end{array} \tag{1.2.3}$$

This problem is now convex, since its objective function is convex and its constraints are linear in the variable H. Moreover, even though it seems very special, we will see in the next chapter that the general minimum-volume enclosing ellipsoid problem can be reduced to the centered case by considering points in the next higher dimension, which is a very modest price to pay for convexity. We call (P) the minimum-volume enclosing ellipsoid (MVEE) problem.

Note that any such centered ellipsoid, along with x_i, also naturally contains $-x_i$, so we may consider the set $\mathbf{X}$ to be the convex hull of the centrally symmetric set $\{\pm x_1, \ldots, \pm x_m\}$. Of course, we do not need to double the number of constraints, but it is often helpful to think of x_i as representing the pair of points $\pm x_i$.

1.3 ▪ Optimal design in statistics

And now for something (apparently) completely different! Suppose we would like to study the relationship between some independent variable $t \in \mathrm{R}^p$ and a dependent variable $v \in \mathrm{R}$. We assume that v is related to t by a model involving some unknown parameters θ. In fact, we will assume that the model is *linear* in the parameters θ, but we allow nonlinearity in t. So our model is

$$v = x(t)^T \theta,$$

where $x : \mathrm{R}^p \to \mathrm{R}^n$ captures the nonlinear dependence on t: for example, if we wanted to build a cubic model of a scalar t, then $x(t) := (1; t; t^2; t^3) \in \mathrm{R}^4$ and $\theta \in \mathrm{R}^4$ would correspond to the unknown coefficients. (We use $(p; q; \ldots; r)$ to denote the column vector obtained by concatenating the scalars or vectors $p, q, \ldots, r$.) Of course, there is usually some error in the model and/or the measurement of v, and so we will assume that V is a random variable,

$$V = x(t)^T \theta + \epsilon,$$

and we will further suppose that ϵ is distributed as a normal random variable with mean 0 and variance σ^2.

To estimate θ, we might observe V at m different values of t, $t_1, t_2, \ldots, t_m$, corresponding to m different values of x, $x_1 := x(t_1), x_2 := x(t_2), \ldots, x_m := x(t_m)$, obtaining the vector (abusing notation slightly) $v \in \mathrm{R}^m$. In fact, we will henceforth mostly ignore the original variables t and concentrate on the linear relationship between x and V. We assume that the m different observations of V are independent.

Let us denote by X the $n \times m$ matrix whose columns are the x_i's. Then one estimator of θ with attractive statistical properties is the solution to the least-squares problem $\min_\theta ||X^T\theta - v||$, which is (assuming X has rank n, which is necessary and sufficient for the least-squares problem to have a unique solution)

$$\hat{\theta} := (XX^T)^{-1} X v.$$

Since v is a sample of the random variable $X^T\theta + \underline{\epsilon}$, where $\underline{\epsilon}$ is an m-dimensional $N(0, \sigma^2 I)$-distributed random variable, $\hat{\theta}$ is a sample from the random variable

$$\hat{\Theta} := (XX^T)^{-1} X (X^T\theta + \underline{\epsilon}) = \theta + (XX^T)^{-1} X \underline{\epsilon},$$

whose mean is exactly θ. Hence our estimator is unbiased. Further, its variance is $\mathbf{E}(\hat{\Theta} - \theta)(\hat{\Theta} - \theta)^T$, which is

$$(XX^T)^{-1} X \mathbf{E}(\underline{\epsilon}\underline{\epsilon}^T) X^T (XX^T)^{-1} = \sigma^2 (XX^T)^{-1}.$$

In an experiment, we might choose to make multiple (or no) observations at the point x_i. This choice amounts to *designing* the experiment. In a more general setting, we might also be able to choose the points x_i from a given design space $\mathcal{X}$, some compact subset of R^n (which itself corresponds to all $x(t)$'s for t lying in a design space $\mathcal{T} \subset \mathrm{R}^p$), but here we confine ourselves to a finite design space $\mathcal{X} = \{x_1, x_2, \ldots, x_m\}$ for simplicity. Note that $x_1, x_2, \ldots, x_m$ correspond to *all* possible choices for $x(t)$, and so it is helpful to think of m as potentially very large. Our choice is then just the number of observations to make at each point. If we make n_i (independent) observations at x_i for $i = 1, 2, \ldots, m$, we get an *experimental design of size* $N := \sum_i n_i$. (Throughout, the range of a summation over i is always 1 through m.) Let W denote the diagonal matrix $\mathrm{Diag}(n_1/N, \ldots, n_m/N)$. Then the corresponding estimator is $(XWX^T)^{-1}XW\bar{v}$, where $\bar{v}$ contains the sample averages of the n_i observations when $x = x_i$, $i = 1, 2, \ldots, m$, with variance

$$\frac{\sigma^2}{N}(XWX^T)^{-1}.$$

We would like to choose the n_i's, or equivalently W, to make this variance small in some sense. This is a hard nonlinear integer programming problem, so instead we allow the weighting matrix to correspond to *any* distribution on $\mathcal{X}$, not just a distribution using rational probabilities with denominator N, as above.

If we assign weight (probability) u_i to x_i, and choose to minimize the determinant of the variance (D-optimality), we arrive at the problem

$$(D) \qquad \begin{array}{ll} \max_{u \in \mathbf{R}^m} & \mathrm{lndet}(XUX^T) \\ & e^T u = 1, \\ & u \geq 0, \end{array} \tag{1.3.1}$$

where, here and below, U denotes $\mathrm{Diag}(u)$ and e denotes a vector of 1's of dimension m. So once again, optimization of the logdet function appears, here in the very different context of optimal design. We will see that the problems of finding a minimum-volume centered ellipsoid containing $\mathcal{X}$ and obtaining a D-optimal design are very closely related: in fact, they are dual problems.

Of course, D-optimality is just one way we can choose a design to somehow make the variance matrix small. Another natural possibility is to consider the resulting variance of an estimator for an observation taken at any design point: at x_i, it is proportional to $x_i^T(XUX^T)^{-1}x_i$. Hence we arrive at another criterion, called G-optimality, where we seek to minimize the largest such variance:

$$\min_{u \in \mathbf{R}^m} \{\max_i (x_i^T (XUX^T)^{-1} x_i) : e^T u = 1, u \geq 0\}.$$

Remarkably, it turns out that a vector u is D-optimal iff it is G-optimal, so that this optimality criterion has some robustness. Also, as in our discussion of ellipsoids, D- and G-optimality have some invariance properties. A vector u is optimal for X iff it is optimal for MX, where M is any nonsingular $n \times n$ matrix representing a nonsingular linear transformation of the design space. So we could, for instance, use any desired basis of the polynomials of degree n, and the resulting design would be the same. Other optimality criteria are discussed in the references on optimal design given in Section 2.5.

1.4 ▪ Applications

Here we briefly list some applications of minimum-volume containing ellipsoids from various fields. One large area is optimal design in statistics, which we have briefly

described in the previous section and which we will continue to discuss throughout the book. There are other applications in statistics. Silverman and Titterington [71] mention using minimum-volume ellipsoids as "peeling" devices in data analysis: in particular, the first few points peeled off can be regarded as outliers in the distribution generating the points. (While this is a useful technique, there is clearly a danger in this approach, since the minimum-volume ellipsoid itself is mostly determined by these outliers and hence may be corrupted by noise and error.) They also note that the minimum-volume ellipsoid can be used to estimate the mean and correlation structure of a population, especially in the case where data points in the interior might be obliterated. Finally, they mention the use of minimum-volume ellipsoids in pattern recognition in order to separate clouds of points, citing Rosen [66], although he used a different criterion. In a similar way, we could evaluate a proposed clustering of a set of points by, for example, summing the volumes of the minimum ellipsoids containing each cluster. The advantage of using ellipsoids rather than balls is that their affine invariance can better model the anisotropy of a given subpopulation. Glineur [35] also uses ellipsoids in pattern recognition and clustering, but with a different criterion of maximal separation. Silverman and Titterington [71] also mention that minimum-volume ellipsoids can be viewed as anisotropic versions of minimal covering spheres, which arise in facility location problems in operations research; see, e.g., Elzinga and Hearn [26].

Containing ellipsoids have been used in parameter identification and control theory to describe uncertainty sets for parameters or state vectors: see Schweppe [68] and Chernousko [20] and the references therein. Vicino and Zappa [82] use a minimum-volume containing parallelotope in a similar context. Hero, Zhang, and Rogers [42, 43] use either ellipsoids or parallelotopes for tomographic feature detection and classification in medical contexts.

Minimum-volume ellipsoids also arise in computational geometry and computer graphics [24]. For example, computational challenges arise relating to obstacle avoidance in robotics, and in avoiding intersection between moving objects, as in game design. Calculating bounding ellipsoids for all the objects of interest can give a simple sufficient condition for avoidance: if all bounding ellipsoids are disjoint, then so are the objects of interest, while if two bounding ellipsoids intersect, further analysis can consider the corresponding objects themselves.

Mathematical analysts are interested in approximating the norm in a Banach space by a Euclidean norm: finding the minimum-volume centered ellipsoid containing its unit ball and using Theorem 1.1 give a solution. (Some readers may think that this stretches the notion of "application.")

Finally, although our interest is in the optimization of ellipsoids, we should mention that minimum-volume ellipsoids arise within other optimization methods. In his integer-programming method, Lenstra [58] uses an initial rounding procedure that can be performed by computing a minimum-volume ellipsoid for a bounded polyhedron. And every step of the famous ellipsoid method for convex programming requires the minimum-volume ellipsoid containing the intersection of an ellipsoid and a half-space or slab.

Because we want to discuss this application in the future, let us define the situation more precisely. Suppose we want to find a point in the polyhedron $\mathbf{Z} := \{z \in \mathrm{R}^n : a_j^T z \geq c_j, j = 1,\ldots,m\}$, and we know somehow that $\mathbf{Z}$ is contained in the ellipsoid $\mathcal{E} := n^{-1/2}\mathcal{E}(H,\overline{z}) = \{z \in \mathrm{R}^n : (z-\overline{z})^T H(z-\overline{z}) \leq 1\}$ (for this application it is simpler and more traditional to use a right-hand side of 1 rather than n). We check whether the center $\overline{z}$ lies in $\mathbf{Z}$, and if not we find a constraint, say $a^T z \geq \gamma$, which is violated by $\overline{z}$.

Without loss of generality, we can assume that a is scaled so that $a^T H^{-1} a = 1$, and then $a^T(z - \overline{z})$, for $z \in \mathcal{E}$, lies between -1 and $+1$. Let $\alpha := \gamma - a^T \overline{z}$. Then all points in $\mathbf{Z}$ lie in the set $\{z \in \mathcal{E} : a^T(z - \overline{z}) \geq \alpha\}$, the intersection of an ellipsoid and a half-space. If some of the constraints defining $\mathbf{Z}$ are two-sided, we may be able to find some β so that all points in $\mathbf{Z}$ lie in the set

$$\mathcal{E}_{\alpha\beta} := \{z \in \mathcal{E} : \alpha \leq a^T(z - \overline{z}) \leq \beta\}, \tag{1.4.1}$$

the intersection of an ellipsoid and a slab, i.e., the set between two parallel hyperplanes. Indeed, the previous case is a special case where $\beta = 1$. In order to continue our method of finding a point in $\mathbf{Z}$, we need to find an ellipsoid containing $\mathcal{E}_{\alpha\beta}$, and to make good progress, we choose the minimum-volume such ellipsoid.

1.5 ▪ Outline of the book

The preceding sections have given some idea of the simplicity and flexibility of ellipsoids and the elegant properties of the logdet function. In the rest of this book you will learn more than you ever (thought you) wanted to learn about these geometric objects and this convex function, organized as follows.

In the next chapter we study the basic minimum-volume enclosing ellipsoid problem for the convex hull of a finite set of points. We first consider the centered version of this problem and derive its dual, showing the relationship to the D-optimal design problem of statistics. We then obtain optimality conditions for these two problems, and hence show how the noncentered ellipsoid problem can be reduced to the centered case. John's theorem on the goodness of fit of minimum-volume ellipsoids follows.

Chapter 3 considers algorithms for the centered minimum-volume enclosing ellipsoid problem. Although the functions involved are smooth, and we have closed-form expressions for their derivatives, it turns out that, for large-scale problems, first-order methods are often more efficient, and these are the focus of our development. We show how rank-one update formulae make each iteration of Frank–Wolfe-type algorithms very cheap to perform, and analyze, both from a complexity viewpoint and for its local convergence, a particular algorithm of this type that incorporates away steps.

In Chapter 4 we consider an extension of the ellipsoid problem in which we seek an ellipsoidal cylinder of minimum cross-sectional area containing a point set. This problem is motivated by optimal design in statistics, but also has an appealing geometric nature. We again consider the dual problem and optimality conditions, but these are more complicated than in the ellipsoid case.

Chapter 5 addresses a first-order method for the ellipsoidal cylinder problem. Again, rank-one update formulae are key to the efficient implementation of this method. As we shall see, here the situation is quite complicated, and we are only able to establish convergence results under rather strong conditions.

Finally, in Chapter 6 we consider related problems and methods. In particular, we discuss a method for dealing with problems of noisy points or outliers; we consider approximating a body via parallelotopes; and we address the problem of ellipsoid optimization when the body is given as the solution set to a system of linear inequalities.

1.6 ▪ Notes and references

The idea of considering the minimum-volume enclosing ellipsoid and proofs of its existence and uniqueness appeared first in unpublished work of C. Löwner. Proofs in

the literature can be found in Danzer, Laugwitz, and Lenz [22] and Zaguskin [90]. These authors also consider the related question of the maximum-volume ellipsoid inscribed in a convex body.

Theorem 1.1 appears in [45] as an application of John's necessary conditions for inequality-constrained nonlinear optimization. Amazingly, this early work also allows a set of constraints indexed by a compact set rather than just a finite set of constraints, and hence John's theorem applies when $\mathbf{X}$ is any compact body in $\mathbf{R}^n$. John's necessary conditions required no assumptions other than differentiability, but when an additional constraint qualification holds (and there are only a finite number of constraints), Kuhn and Tucker [55] obtained stronger conditions in 1951. Surprisingly, both John's (for a finite number of constraints) and Kuhn and Tucker's results were anticipated in the Master's thesis of Karush [47] in 1939, and the conditions are now known as the Karush–John and Karush–Kuhn–Tucker conditions.

Our interest throughout in approximating point sets or, more generally, compact sets by ellipsoids is perhaps a little simplistic, but it scales well with the dimension and the number of points. More precise information for when the points approximate a manifold embedded in $\mathbf{R}^n$ can be obtained by more sophisticated algorithms with higher complexity. Amenta et al. [5] give an algorithm for reconstructing surfaces in $\mathbf{R}^3$, and algorithms for higher dimensions are given in, for example, Niyogi, Smale, and Weinberger [59] and Chazal, Cohen-Steiner, and Lieutier [19].

We may have been a little too hasty in dismissing the general minimum-volume ellipsoid problem (1.2.2) as intractable because of its lack of convexity. Note that the constraints can be written as $\|H^{1/2}x_i - H^{1/2}\bar{x}\| \leq \sqrt{n}$ for all i. If we choose as new variables the positive definite matrix $B := H^{1/2}$ and the vector $\bar{y} := H^{1/2}\bar{x}$, we can state the problem in the form

$$(P_1') \qquad \begin{array}{ll} \min_{B,\bar{y}} & -2\ln\det(B) \\ & \|Bx_i - \bar{y}\| \leq \sqrt{n}, \quad i = 1,2,\ldots,m, \end{array} \tag{1.6.1}$$

which is convex. However, this is still clearly more complex than the centered problem with linear constraints and, as mentioned, we will reduce the general case to the centered one in the next chapter.

The topic of optimal design in statistics goes back to Wald [83] and Elfving [25]. Notable results, including the equivalence of D- and G-optimality, were obtained by Kiefer and Wolfowitz [52, 53], and later contributions were made by Fedorov, Wynn, Atwood, Sibson, Silvey, Titterington, and Pukelsheim, among others. See the books of Fedorov [27], Silvey [73], and Pukelsheim [63].

For an account of the use of John's theorem in the analysis of Banach spaces, see the book of Pisier [62].

The ellipsoid method was developed by Yudin and Nemirovskii [89] and Shor [69] as an implementable approximation to an optimal (in the oracle sense) method for convex programming. Khachiyan [48] later used it to prove the polynomial solvability of linear programming. Bland, Goldfarb, and Todd [14] give a survey of the ellipsoid method, and its consequences in combinatorial optimization are explored in depth in Grötschel, Lovász, and Schrijver [37].

Chapter 2
Minimum-Volume Ellipsoids

Now we begin our consideration of the minimum-volume ellipsoid optimization problem. In this chapter we discuss the problems from a theoretical point of view, while the next chapter considers algorithms.

In the first section we obtain the dual of the centered minimum-volume enclosing ellipsoid (MVEE) problem, which coincides with the D-optimal design problem addressed in Section 1.3. We prove existence and uniqueness theorems, and derive weak and strong duality. In Section 2.2 we obtain optimality conditions for the problem and its dual, and also define notions of approximate optimality.

In Section 2.3 we return to the general minimum-volume enclosing ellipsoid problem, where the center is not required to be the origin, and we show that this problem can be reduced to a centered problem in the next higher dimension. This justifies our concentration on the centered version of the problem throughout this monograph. We prove John's theorem on the quality of approximation by the minimum-volume ellipsoid in Section 2.4.

2.1 ▪ Duality, existence, and uniqueness

Recall that the MVEE problem of finding the minimum-volume centered ellipsoid enclosing a set $\mathbf{X}$ that is the convex hull of points $x_i, i = 1,\dots,m$, in R^n can be formulated as

$$\begin{array}{ll} \min_{H\in\mathcal{S}^n} & f(H) := -\ln\det(H) \\ (P) & x_i^T H x_i \le n, \quad i = 1,2,\dots,m. \end{array} \tag{2.1.1}$$

We assume throughout that the points x_i span all of R^n, since otherwise ellipsoids of arbitrarily small n-dimensional volume circumscribe $\mathbf{X}$. Equivalently, we assume that

$$X \text{ has full row rank,} \tag{2.1.2}$$

where we denote by X the matrix whose columns are the x_i's,

$$X := [x_1, x_2, \dots, x_m] \in \mathrm{R}^{n\times m}.$$

We say H is feasible for (P) if it satisfies the constraints and yields a finite objective value, so that H is positive definite.

If we apply a Lagrange multiplier u_i to each constraint, we arrive at the Lagrangian

$$L(H,u) := -\ln\det(H) + \sum_i u_i (x_i^T H x_i - n), \tag{2.1.3}$$

which is finite for any positive definite $H \in \mathcal{S}^n$ and $u \in \mathrm{R}^m$. If we again denote by e the vector of 1's in R^m and employ the very useful notation

$$U := \mathrm{Diag}(u) \in \mathcal{S}^m,$$

we can rewrite the Lagrangian as

$$L(H,u) = -\ln\det(H) + H \bullet XUX^T - ne^T u$$

(for details, see (2.1.6) below). It is immediate that, for any nonnegative $u \in \mathrm{R}^m$ and H feasible in (P),

$$-\ln\det(H) \geq L(H,u),$$

from which we deduce that, for any nonnegative u,

$$v(P) \geq \min_H L(H,u),$$

where $v(P)$ denotes the optimal value of (P). Moreover, since $L(H,u)$ is a strictly convex function of H with gradient

$$\nabla_H L(H,u) = -H^{-1} + XUX^T,$$

$L(H,u)$ is minimized by $\bar{H}$ iff XUX^T is positive definite and

$$\bar{H} = (XUX^T)^{-1}.$$

What if XUX^T is positive semidefinite but not positive definite? Then let $v \in \mathrm{R}^n$ satisfy $XUX^T v = 0$ and have unit norm, and consider $H(\lambda) := I + \lambda vv^T$. We have

$$\begin{aligned} L(H(\lambda,u) &= -\ln\det(I + \lambda vv^T) + (I + \lambda vv^T) \bullet XUX^T - ne^T u \\ &= -\ln(1+\lambda) + \mathrm{Trace}(XUX^T) - ne^T u, \end{aligned}$$

which tends to $-\infty$ as $\lambda \to \infty$. We conclude that

$$\begin{aligned} \min_H L(H,u) &= -\ln\det[(XUX^T)^{-1}] + (XUX^T)^{-1} \bullet XUX^T - ne^T u \\ &= \ln\det(XUX^T) + n - ne^T u. \end{aligned}$$

To obtain the best bound on $v(P)$, we should choose $u \in \mathrm{R}^m_+$ to maximize the right-hand side above. This leads to a first dual problem (we will refine it just below):

$$(D') \quad \max_{u \in \mathbf{R}^m} \{\ln\det(XUX^T) + n - ne^T u : u \geq 0\}. \tag{2.1.4}$$

In fact, we can restrict u to satisfy $e^T u = 1$, for the following reason. Any nonnegative $\hat{u}$ can be written as λu, where λ is nonnegative and u satisfies $e^T u = 1, u \geq 0$. Then

$$\begin{aligned} \ln\det(X\hat{U}X^T) + n - ne^T \hat{u} &= \ln\det[\lambda(XUX^T)] - n\lambda + n \\ &= n\ln\lambda + \ln\det(XUX^T) - n\lambda + n, \end{aligned}$$

and this is maximized by choosing $\lambda = 1$. To obtain the best bound on $v(P)$, we are thus led to consider the dual to the MVEE problem as

$$\begin{array}{lll} \max_{u \in \mathbf{R}^m} & g(u) := \ln\det(XUX^T) & \\ (D) & e^T u & = 1, \\ & u & \geq 0, \end{array} \tag{2.1.5}$$

which is exactly the D-optimal design problem from Section 1.3! We say that u is feasible for (D) if it satisfies the constraints and yields a finite objective value, so XUX^T must be positive definite.

Even though it is implicit in the argument above, let us prove weak duality: the short proof also highlights conditions to ensure equality in primal and dual objective values.

Proposition 2.1. *For any H and u feasible in (P) and (D), respectively,*

$$f(H) \geq g(u).$$

Proof. First note that

$$\begin{aligned} H \bullet XUX^T &= \mathrm{Tr}[HX(\textstyle\sum_i u_i e_i e_i^T X^T)] = \mathrm{Tr}(\textstyle\sum_i u_i x_i^T H x_i) \\ &= \textstyle\sum_i u_i x_i^T H x_i \qquad\qquad \leq n e^T u \quad = \quad n, \end{aligned} \tag{2.1.6}$$

using the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ from (A.1.1) in Appendix A.

Observe that the matrix $HXUX^T$ is similar to the positive definite matrix $H^{1/2}XUX^TH^{1/2}$, and hence has n positive eigenvalues λ_j, $j = 1,\ldots,n$. The inequality above shows that the sum of these is at most n.

Now the difference of the objective values is

$$\begin{aligned} f(H) - g(u) &= -\mathrm{lndet}(H) - \mathrm{lndet}(XUX^T) \\ &= -\mathrm{lndet}(HXUX^T) \\ &= -\ln(\textstyle\prod_j \lambda_j) \\ &= -n \ln(\textstyle\prod_j \lambda_j)^{1/n} \\ &\geq -n \ln(\textstyle\sum_j \lambda_j / n) \\ &\geq -n \ln(n/n) = 0, \end{aligned} \tag{2.1.7}$$

where the first inequality follows from the arithmetic-geometric mean inequality and the second follows from (2.1.6). ∎

The main result of this section shows that (P) always has a unique optimal solution, (D) always has an optimal solution, and there is no duality gap. For this we use optimality conditions for (P), but in honor of John's fundamental work on this problem, we use those of Karush and John rather than the more usual Karush–Kuhn–Tucker conditions.

Theorem 2.2. *Under assumption* (2.1.2), *(P) has a unique optimal solution H^*, (D) has an optimal solution u^* with XU^*X^T unique, and $f_* := f(H^*) = g(u^*) =: g^*$.*

Proof. We start with (P). We would like to use the celebrated Weierstrass theorem, that a continuous function on a compact set attains its minimum, but unfortunately the set of feasible H is not compact since it is not closed (H must be positive definite, not just positive semidefinite). We therefore employ a standard trick: since ϵI is feasible for sufficiently small positive ϵ, we can without loss of generality add the constraint that $-\mathrm{lndet}(H) \leq -\mathrm{lndet}(\epsilon I)$, which defines a closed set containing only positive definite matrices. Further, the objective function f is continuous on this set.

Next, we see that for each j, $\mu_j e_j$ is a convex combination of the points $\pm x_i$, $i = 1,\ldots,m$, for some positive μ_j, by (2.1.2). Since all the points $\pm x_i$ lie in $\mathcal{E}(H)$, so does $\mu_j e_j$, and hence the diagonal entry $h_{jj} \leq n\mu_j^{-2}$. Thus there is a uniform bound on the

trace, and hence on the spectral norm, of every feasible H, and this implies that, with the added constraint, the feasible region is compact and thus an optimal solution exists. Uniqueness follows since f is a strictly convex function: indeed, if H_1 and H_2 were two distinct optimal solutions, then $(H_1 + H_2)/2$ would be feasible with a lower objective value, which is a contradiction.

We could use a similar argument to show the existence of an optimal solution to (D). Here the feasible region is bounded, but not closed, since some points like e_1, where the objective function is negative infinity, are not feasible. We could again add an additional constraint, but this doesn't help us to establish strong duality, so we use a different approach.

Problem (P) is a nonlinear programming problem, even if it lies in the rather unfamiliar space of symmetric matrices. If we add an extra constraint using ϵI as above, and choose $\epsilon > 0$ suitably, all the functions will be differentiable in a neighborhood of the feasible region, and the optimal solution will not satisfy the new constraint with equality. We now use the Karush–John optimality conditions at the optimal solution H^*. These imply that there are nonnegative multipliers, not all zero: τ for the objective function and u_i for the ith constraint in (P), $i = 1, \ldots, m$, such that

$$\begin{aligned} -\tau(H^*)^{-1} + \textstyle\sum_i u_i x_i x_i^T &= 0, \\ u_i(x_i^T H^* x_i - n) &= 0, \quad i = 1, \ldots, m. \end{aligned} \tag{2.1.8}$$

Here we have used the fact that the gradient of $-\ln\det(H)$ is $-H^{-1}$ and that the constraint function $x_i^T H x_i - n$ can be written as $x_i x_i^T \bullet H - n$, so that its gradient is $x_i x_i^T$.

Let us first take the scalar product of the first line of (2.1.8) with H^*. We get

$$-n\tau + \sum_i u_i x_i x_i^T \bullet H^* = 0,$$

and using the fact that $x_i x_i^T \bullet H^* = x_i^T H^* x_i = n$ whenever $u_i > 0$ (from the second line of (2.1.8)), this gives $\sum_i u_i = \tau$. We deduce that τ cannot be zero, because then all the u_i's would also be zero, so without loss of generality we can assume $\tau = 1$ by scaling all the multipliers. Then (writing u for the m-vector of the scaled u_i's)

$$e^T u = 1, \quad u \geq 0,$$

and u is feasible for (D).

The first line of (2.1.8) now states that

$$(H^*)^{-1} = X U X^T,$$

with U as always denoting $\mathrm{Diag}(u)$, and hence

$$-\ln\det(H^*) = \ln\det[(H^*)^{-1}] = \ln\det(X U X^T).$$

Hence the primal feasible solution H^* and the dual feasible solution u have the same objective values, and thus weak duality implies that both (in particular u) are optimal and that there is no duality gap.

It remains to show that $X U^* X^T$ is unique for optimal solutions u^*. Indeed, if $\bar{u}$ and $\hat{u}$ were both optimal, and $X \bar{U} X^T$ and $X \hat{U} X^T$ were distinct, then $u := (\bar{u} + \hat{u})/2$ would be feasible, and its objective value $\ln\det[(X \bar{U} X^T + X \hat{U} X^T)/2]$ would be strictly greater than those for $\bar{u}$ and $\hat{u}$, which is a contradiction. □

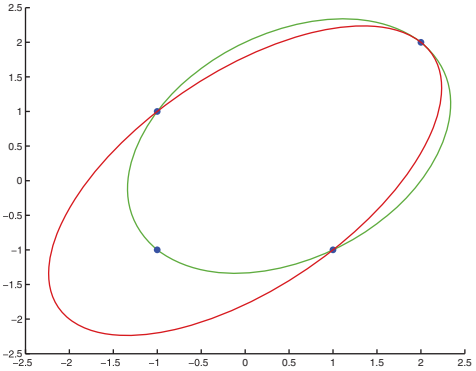


Figure 2.1. *Minimum-area ellipse.*

In fact, since the space of symmetric matrices of order n has dimension $n(n+1)/2$, it is sufficient to use just this many positive u_i's, $1 \le i \le m$, in (2.1.8). This implies that there is an optimal solution to (D) with only $n(n+1)/2$ positive components, and also that the minimum-volume centered ellipsoid containing the points x_i corresponding to these positive u_i's is also the minimum-volume centered ellipsoid containing them all; that is, we have a small core set. (The difference between the estimate here, $n(n+1)/2$, and that in Chapter 1, $n(n+3)/2$, is due to the fact that earlier we were concerned with not-necessarily-centered ellipsoids.)

Example 2.3. Suppose we want the minimum-volume centered ellipse containing the four points that are the columns x_i, $i = 1,2,3,4$, of

$$X = \begin{bmatrix} -1 & -1 & 1 & 2 \\ 1 & -1 & -1 & 2 \end{bmatrix}$$

in R^2. If we set $u := (0;0;1/2;1/2)$, we find

$$XUX^T = \begin{bmatrix} 5/2 & 3/2 \\ 3/2 & 5/2 \end{bmatrix}, \quad H := (XUX^T)^{-1} = \begin{bmatrix} 5/8 & -3/8 \\ -3/8 & 5/8 \end{bmatrix}.$$

It is then easy to check that $x_i^T H x_i \le 2$ for all i, with equality for $i = 1,3,4$, and so H is feasible for the primal problem and u is for the dual. Moreover, their objective values are both $\ln 4$, and hence both are optimal. We also note that $\bar{u} := (1/2;0;0;1/2)$ is also optimal for the dual, but yields the same matrix XUX^T.

The points and the corresponding minimum-volume centered ellipse (in red) are shown in Figure 2.1. ∎

Example 2.4. Now let us consider the case where $m = n$, so that the number of points is equal to the dimension. Suppose we set $u = e/n$. Since the matrix X has full rank and is square, it is nonsingular, and so $H := (XUX^T)^{-1} = nX^{-T}X^{-1}$. We find

$$x_i^T H x_i = n x_i^T X^{-T} X^{-1} x_i = n e_i^T e_i = n,$$

so that H is feasible for the primal problem and u is for the dual. Their objective values are both $2\ln(|\det X|) - n\ln n$, so both are optimal. ∎

2.2 ▪ Optimality conditions

Now that we know that strong duality holds, we can state some necessary and sufficient optimality conditions. Indeed, one reason we gave an explicit proof of weak duality in Proposition 2.1 (weak duality holds automatically for the Lagrangian dual) is so that we could easily identify conditions for strong duality.

Proposition 2.5. *Necessary and sufficient conditions for H and u to be optimal in (P) and (D), respectively, are*

(a) $e^T u = 1$, $u \geq 0$, *and* $x_i^T H x_i \leq n$ *for* $i = 1, \ldots, m$;

(b) $H = (XUX^T)^{-1}$; *and*

(c) $x_i^T H x_i = n$ *if* $u_i > 0$, $i = 1, \ldots, m$.

Proof. Condition (a) (and the nonsingularity implied by (b)) says merely that H and u are feasible in their respective problems. Then Theorem 2.2 implies that necessary and sufficient conditions for optimality are $f(H) = g(u)$. From (2.1.7) this holds iff the positive eigenvalues of $HXUX^T$ are all equal (so their geometric mean equals their arithmetic mean) and that its trace is equal to n. In turn, this follows iff all eigenvalues are 1, so that $H = (XUX^T)^{-1}$. Moreover, from (2.1.6), the trace can only equal n if (c) above holds. □

Using (b) above, we can write these conditions solely with respect to the dual solution u.

Proposition 2.6. *A feasible solution u to (D) is optimal (and $H(u)$ is optimal in (P)) iff*

(i) $H(u) := (XUX^T)^{-1}$ *is feasible in* (P) *and*

(ii) $x_i^T H(u) x_i = n$ *if* $u_i > 0$, $i = 1, \ldots, m$.

Indeed, condition (i) above implies condition (ii): from (i),

$$n = H(u) \bullet XUX^T = \sum_i u_i x_i^T H(u) x_i \leq \sum_i n u_i = n,$$

since $H(u)$ is feasible in (P), so (ii) follows.

Henceforth, we will use $H(u)$ to denote the matrix above.

Definition 2.7. *If u is nonnegative and XUX^T is positive definite, then $H(u)$ denotes $(XUX^T)^{-1}$.*

From these conditions, we see again the optimality of u and $H(u)$ in Example 2.3, without the necessity of checking their objective values.

Let us use these results to show the equivalence of D-optimality and G-optimality in optimal design—see Section 1.3. Indeed, if u^* is D-optimal, it solves (D), and then we know from Proposition 2.5 that $H = (XU^*X^T)^{-1}$ is optimal in (P), so that

$$\max_i x_i^T (XU^*X^T)^{-1} x_i \leq n.$$

On the other hand, for any nonnegative u with $e^T u = 1$,

$$\sum_i u_i x_i^T (XUX^T)^{-1} x_i = (XUX^T) \bullet (XUX^T)^{-1} = \text{Trace}(I) = n,$$

so that $\max_i x_i^T(XUX^T)^{-1}x_i \geq n$. This shows that u^* minimizes this maximum, i.e., that it is G-optimal.

Conversely, if $\bar{u}$ is G-optimal, then it minimizes $\max_i x_i^T(XUX^T)^{-1}x_i$. Since this quantity is at most n for u^*, it is also at most n for $\bar{u}$, which implies that $\bar{u}$ satisfies the optimality condition (i) in Proposition 2.6, and hence is also D-optimal.

We will use these optimality conditions in our algorithms. Indeed, to terminate the methods with a guaranteed quality of solution, we state some approximate optimality conditions.

Definition 2.8. *A feasible u is said to be* ϵ-primal feasible *if* $H(u) := (XUX^T)^{-1}$ *satisfies*

$$x_i^T H(u) x_i \leq (1+\epsilon)n, \quad i = 1, \ldots, m.$$

If moreover, it satisfies

$$x_i^T H(u) x_i \geq (1-\epsilon)n \text{ if } u_i > 0, \quad i = 1, \ldots, m,$$

we say that u is ϵ-approximately optimal, *or that it satisfies the* ϵ-approximate optimality conditions.

We say a solution is *within δ of being optimal* in a problem if it is feasible and its objective value is within δ of the optimal value of the problem.

We can now prove

Proposition 2.9. *If u is ϵ-primal feasible (and a fortiori if it is ϵ-approximately optimal), then u and $(1+\epsilon)^{-1}H(u)$ are both within $n\ln(1+\epsilon)$, which is at most $n\epsilon$, of being optimal in their respective problems. Moreover, $\mathcal{E}((1+\epsilon)^{-1}H(u))$ contains all x_i and is within a factor of $(1+\epsilon)^{n/2}$ of the minimum-volume such ellipsoid.*

Proof. From the definition above, both these solutions are feasible. Moreover, the corresponding duality gap is

$$f((1+\epsilon)^{-1}H(u)) - g(u) = n\ln(1+\epsilon) + f(H(u)) - g(u) = n\ln(1+\epsilon) \leq n\epsilon,$$

and this proves the first result using weak duality. The second follows by the primal feasibility of $(1+\epsilon)^{-1}H(u)$ using (1.2.1). □

2.3 ▪ Relaxing the centered restriction

In Chapter 1 we claimed that the not-necessarily-centered minimum-volume ellipsoid problem could be reduced to the centered case. Here we show how this can be done.

Suppose we seek the minimum-volume ellipsoid containing a set

$$\mathbf{Y} := \text{conv}(\{y_1, \ldots, y_m\}),$$

where each $y_i \in \mathrm{R}^d$. We let Y denote the matrix whose columns are these points:

$$Y := [y_1, y_2, \ldots, y_m] \in \mathrm{R}^{d\times m}.$$

We now assume without loss of generality that the affine hull of the points y_i is R^d; that is, any point $y \in \mathrm{R}^d$ can be expressed as a linear combination $\sum_i \lambda_i y_i$, where the weights

λ_i sum to 1. Note that this is the same as saying that any point $(y;1)$ can be written as a linear combination of the m points $x_i := (y_i;1)$ in R^{d+1}, or equivalently, that the points x_i span R^n, where $n := d+1$. (We use notation similar to that of MATLAB, so x_i is a column vector with a 1 for its last component and the components of y_i for its first d components.) Note that, if matrix X is defined as before, we have

$$X = \begin{bmatrix} Y \\ e^T \end{bmatrix}. \tag{2.3.1}$$

Observe that the points x_i can be viewed as copies of the points y_i embedded in $\mathrm{R}^d \times \{1\} \subseteq \mathrm{R}^n$. The extra dimension allows the center of the ellipsoid in this hyperplane to be wherever it likes to minimize the volume. The most natural n-dimensional problem to solve now is to find a centered ellipsoidal *cylinder* in R^n containing all the points x_i and having the minimum-d-volume intersection with $\mathrm{R}^d \times \{1\} \subseteq \mathrm{R}^n$. This *minimum-area ellipsoidal cylinder* (MAEC) problem will be considered in Chapters 4 and 5, and we shall see that it is considerably harder than the MVEE problem. In our case, where all points x_i lie in the hyperplane at height 1, solving an MVEE problem suffices: we shall now show that the minimum-volume ellipsoid containing the y_i's can easily be obtained from the minimum-volume *centered* ellipsoid containing the x_i's, which is our desired reduction.

Theorem 2.10. *With the assumption and notation above, suppose u^* and $H^* = (X U^* X^T)^{-1}$ are optimal solutions to (D) and (P), respectively (defined using the points x_i, $i = 1,\ldots,m$). Then the unique minimum-volume ellipsoid containing the points y_i, $i = 1,\ldots,m$, is $\mathcal{E}(H^*_{YY}, \bar{y})$, where H^*_{YY} is the $d \times d$ leading submatrix of H^* and $\bar{y} := Yu^*$.*

Proof. First consider an arbitrary ellipsoid $\mathcal{E}(H_{YY}, \hat{y})$ containing the points y_i, so that

$$(y_i - \hat{y})^T H_{YY} (y_i - \hat{y}) \le d, \quad i = 1,\ldots,m.$$

Then we have also

$$\begin{pmatrix} y_i - \hat{y} \\ 1 \end{pmatrix}^T \begin{bmatrix} H_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} y_i - \hat{y} \\ 1 \end{pmatrix} \le n, \quad i = 1,\ldots,m,$$

or

$$\begin{pmatrix} y_i \\ 1 \end{pmatrix}^T \begin{bmatrix} I & 0 \\ -\hat{y}^T & 1 \end{bmatrix} \begin{bmatrix} H_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\hat{y} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix} \le n, \quad i = 1,\ldots,m.$$

This shows that all points x_i lie in $\mathcal{E}(H)$, where

$$\begin{aligned} H &:= \begin{bmatrix} I & 0 \\ -\hat{y}^T & 1 \end{bmatrix} \begin{bmatrix} H_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\hat{y} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} H_{YY} & -H_{YY}\hat{y} \\ -(H_{YY}\hat{y})^T & 1 + \hat{y}^T H_{YY} \hat{y} \end{bmatrix}. \end{aligned} \tag{2.3.2}$$

Note that $\det H = \det H_{YY}$, so that $-\mathrm{lndet}(H_{YY}) = -\mathrm{lndet}(H) \ge -\mathrm{lndet}(H^*)$.

Now suppose that H^* and u^* are optimal for (P) and (D), respectively. Then we know from Proposition 2.5 that

$$\begin{aligned} H^* &= (XU^*X^T)^{-1} \\ &= \left(\begin{bmatrix} Y \\ e^T \end{bmatrix} U^* \begin{bmatrix} Y \\ e^T \end{bmatrix}^T \right)^{-1} \\ &= \left(\begin{bmatrix} YU^*Y^T & \bar{y} \\ \bar{y}^T & 1 \end{bmatrix} \right)^{-1} \\ &= \left(\begin{bmatrix} I & \bar{y} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} YU^*Y^T - \bar{y}\bar{y}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ \bar{y}^T & 1 \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} I & 0 \\ -\bar{y}^T & 1 \end{bmatrix} \begin{bmatrix} (YU^*Y^T - \bar{y}\bar{y}^T)^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\bar{y} \\ 0 & 1 \end{bmatrix}, \end{aligned} \tag{2.3.3}$$

where $\bar{y} := Yu^*$. Let us set $H^*_{YY} := (YU^*Y^T - \bar{y}\bar{y}^T)^{-1}$; we note that this is the leading $d \times d$ principal submatrix of H^* and that $\det H^* = \det H^*_{YY}$ so that $-\ln\det(H^*_{YY}) = -\ln\det(H^*)$.

Now $\mathcal{E}(H^*)$ contains all the points x_i, so for all i,

$$\begin{pmatrix} y_i \\ 1 \end{pmatrix}^T \begin{bmatrix} I & 0 \\ -\bar{y}^T & 1 \end{bmatrix} \begin{bmatrix} H^*_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\bar{y} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix} \le n,$$

or

$$(y_i - \bar{y})^T H^*_{YY} (y_i - \bar{y}) \le d, \tag{2.3.4}$$

so that $\mathcal{E}(H^*_{YY}, \bar{y})$ contains $\mathbf{Y}$ and its volume is related to $-\ln\det(H^*_{YY}) \le -\ln\det(H_{YY})$. This proves the minimality of this ellipsoid, and uniqueness follows from the same arguments, using the fact that the minimum-volume ellipsoid containing the x_i's (and hence H above) is unique. ∎

Using Proposition 2.5 and the proof above, we obtain the following.

Corollary 2.11. *A necessary and sufficient condition for*

$$\{y \in \mathrm{R}^d : (y - \bar{y})^T H (y - \bar{y}) \le d\}$$

to be the minimum-volume ellipsoid containing $\mathbf{Y}$ *as above is the existence of* $u \in \mathrm{R}^m$ *satisfying*

(a) $H = (YUY^T - Yuu^TY^T)^{-1}, \quad \bar{y} = Yu;$

(b) $e^T u = 1$, $u \ge 0$, *and* $(y_i - \bar{y})^T H (y_i - \bar{y}) \le d, \quad i = 1, \ldots, m;$ *and*

(c) $(y_i - \bar{y})^T H (y_i - \bar{y}) = d$ *if* $u_i > 0, \quad i = 1, \ldots, m.$

Example 2.12. We return to Example 2.3, but now we seek the minimum-volume *not-necessarily-centered* ellipse containing the four points y_i that are the columns of

$$Y = \begin{bmatrix} -1 & -1 & 1 & 2 \\ 1 & -1 & -1 & 2 \end{bmatrix}$$

in R^2. (Note that we have changed notation to be consistent with that above.) We add a final component 1 to each point to get the columns x_i of the matrix

$$X = \begin{bmatrix} -1 & -1 & 1 & 2 \\ 1 & -1 & -1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

in R^3. Let us set $u := (9/32; 4/32; 9/32; 10/32)$, so that $\bar{y} = Yu = (1/2; 1/2)$ and

$$YUY^T = \begin{bmatrix} 31/16 & 13/16 \\ 13/16 & 31/16 \end{bmatrix};$$

these can be seen as submatrices of

$$XUX^T = \begin{bmatrix} 31/16 & 13/16 & 1/2 \\ 13/16 & 31/16 & 1/2 \\ 1/2 & 1/2 & 1 \end{bmatrix}.$$

Then we find that

$$H^*_{YY} := (YUY^T - Yuu^TY^T)^{-1} = \begin{bmatrix} 27/16 & 9/16 \\ 9/16 & 27/16 \end{bmatrix}^{-1} = \begin{bmatrix} 2/3 & -2/9 \\ -2/9 & 2/3 \end{bmatrix}.$$

H^*_{YY} can also be found as a submatrix of

$$H = (XUX^T)^{-1} = \begin{bmatrix} 2/3 & -2/9 & -2/9 \\ -2/9 & 2/3 & -2/9 \\ -2/9 & -2/9 & 11/9 \end{bmatrix}.$$

Then $(y_i - \bar{y})^T H^*_{YY}(y_i - \bar{y}) = 2$ for each i, so that all points y_i lie in the ellipse $\mathcal{E}(H^*_{YY}, \bar{y})$, which is the minimum-area ellipse containing them. Figure 2.1 also shows this ellipse (in green). ∎

2.4 ▪ Quality of fit of minimum-volume enclosing ellipsoids

We now have the machinery to prove John's fundamental theorem on the degree of fit of ellipsoids to convex bodies formed as convex hulls of finite sets of points.

Theorem 2.13. *Let $\mathcal{E}_*$ be the minimum-volume ellipsoid containing the convex body $\mathbf{X} = \mathrm{conv}\{x_1, x_2, \ldots, x_m\}$ in R^n.*

(a) *The homothetic scaling $\frac{1}{n}\mathcal{E}_*$ is contained in $\mathbf{X}$.*

(b) *Further, if $\mathbf{X}$ is symmetric ($-\mathbf{X} = \mathbf{X}$), then $\frac{1}{\sqrt{n}}\mathcal{E}_*$ is contained in $\mathbf{X}$.*

Proof. We begin with (b). Since $\mathbf{X}$ is symmetric, if $\mathcal{E}_*$ contains $\mathbf{X}$, then so does $-\mathcal{E}_*$. Now the minimum-volume enclosing ellipsoid is unique, which implies that $\mathcal{E}_*$ is centered, and hence is the minimum-volume centered ellipsoid $\mathcal{E}(H)$, where H and u are optimal solutions to (P) and (D), respectively. Moreover, by Proposition 2.5, $H = (XUX^T)^{-1}$. In this notation,

$$\bar{\mathcal{E}} := \frac{1}{\sqrt{n}}\mathcal{E}_* = \frac{1}{\sqrt{n}}\{x \in \mathrm{R}^n : x^T H x \le n\} = \{x \in \mathrm{R}^n : x^T H x \le 1\}.$$

How can we show that this set is contained in **X**? We use the *support function* of each. If D is a convex set in R^n, its support function is defined by $\delta^*_D(c) := \max\{c^T d : d \in D\}$; it is easy to show that $D_1 \subseteq D_2$ iff $\delta^*_{D_1} \le \delta^*_{D_2}$ pointwise. So let c be a nonzero vector in R^n. Using (1.1.6), we find that the maximum value of $c^T x$ over $\bar{\mathcal{E}}$ is $\sqrt{c^T H^{-1} c} = \sqrt{c^T (XUX^T) c}$. Now

$$c^T(XUX^T)c = \sum_i u_i (c^T x_i)^2 \le \max_i (c^T x_i)^2 = (\max_i c^T x_i)^2,$$

where the last step uses the fact that $\mathbf{X} = -\mathbf{X}$, so that $\min_j c^T x_j = -\max_i c^T x_i$. Hence the maximum of $c^T x$ over $\bar{\mathcal{E}}$ is at most the maximum of the $c^T x_i$'s, which is the maximum of $c^T x$ over **X**. This shows that $\bar{\mathcal{E}}$ is contained in **X** as desired.

Now we turn to the general case (a). It is helpful here to change the notation to fit better with that of the previous section concerning the not-necessarily-centered case. So suppose **Y** is the convex hull of the points y_i, $i = 1, \dots, m$, in R^d, and let $Y := [y_1, \dots, y_m]$. Then, according to (2.3.4), we can find $u \in \mathrm{R}^m$ with $e^T u = 1$, $u \ge 0$, such that

$$\mathcal{E}_* = \{y \in \mathrm{R}^d : (y - \bar{y})^T (YUY^T - Yuu^T Y)^{-1} (y - \bar{y}) \le d\}$$

with $\bar{y} = Yu$. Then

$$\bar{\mathcal{E}} := \frac{1}{d}\mathcal{E}_* = \left\{ y \in \mathrm{R}^d : (y - \bar{y})^T (YUY^T - Yuu^T Y)^{-1} (y - \bar{y}) \le \frac{1}{d} \right\}.$$

We want to show that this is contained in **Y**. Let us choose a nonzero $c \in \mathrm{R}^d$, so that the maximum of $c^T y$ over $\bar{\mathcal{E}}$ is, using again (1.1.6),

$$c^T \bar{y} + \frac{1}{\sqrt{d}} \sqrt{c^T (YUY^T - Yuu^T Y) c} = c^T \bar{y} + \frac{1}{\sqrt{d}} \sqrt{\sum_i u_i (c^T y_i - c^T \bar{y})^2}. \qquad (2.4.1)$$

We want to show that this is at most $\max_i c^T y_i$. Of course, we can choose any scale we want for c: let us assume without loss of generality that

$$c^T (YUY^T - Yuu^T Y) c = 1.$$

Then we want to show that $\max_i c^T y_i - c^T \bar{y}$ is at least $1/\sqrt{d}$. Let us consider a random variable Z which takes the value $z_i := c^T y_i - c^T \bar{y}$ with probability u_i. Since $\bar{y} = Yu$, it has mean 0. In view of (2.4.1), our assumption above shows that Z has variance 1. Also, our scaling of c and the fact that all y_i's lie in $\mathcal{E}_*$ implies by the generalized Cauchy–Schwarz inequality (see Section A.2) that all z_i's are at least $-\sqrt{d}$.

Now suppose that $\max_i z_i = \alpha$. Let us "center" Z to get the random variable

$$\hat{Z} := Z - \frac{\alpha - \sqrt{d}}{2}.$$

Then the values that $\hat{Z}$ takes are at most $(\alpha + \sqrt{d})/2$ in absolute value, and we therefore obtain

$$\left(\frac{\alpha + \sqrt{d}}{2}\right)^2 \ge E\hat{Z}^2 = \mathrm{var}(\hat{Z}) + \left(\frac{\alpha - \sqrt{d}}{2}\right)^2 = 1 + \left(\frac{\alpha - \sqrt{d}}{2}\right)^2,$$

since $\hat{Z}$ and Z have the same variance and $EZ = 0$. This yields $\alpha \ge 1/\sqrt{d}$, which is what we wanted to show. □

Note that the ratios are tight: for the nonsymmetric case, consider a regular n-simplex, and for a symmetric example, consider the n-cube (or the n-crosspolytope, the convex hull of all the plus and minus unit coordinate vectors).

We further remark that the proof shows that, for any u feasible for (D),

$$\{x \in \mathrm{R}^n : x^T H(u) x \leq 1\} \subseteq \mathrm{conv}\{\pm x_1, \ldots, \pm x_m\}.$$

The minimum-area ellipse for Example 2.3, as well as a copy shrunk by $\sqrt{2}$ and the convex hull of $\{\pm x_1, \ldots, \pm x_4\}$, are shown in Figure 2.2.

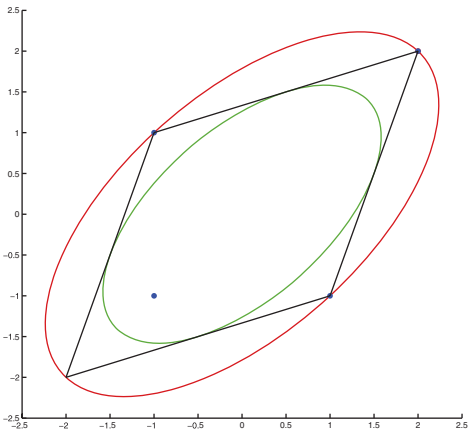


Figure 2.2. *Illustration of John's theorem.*

2.5 ▪ Notes and references

The possible duality relationship between (P) and (D) was raised by Silvey [74] in a discussion of two papers on optimal design, and answered affirmatively by Sibson [70] in the same discussion. (The comments by Kiefer in the same discussion are worth reading if only from a sociological point of view.) Necessary optimality conditions for the general minimum-volume ellipsoid problem were obtained by John [45], and necessary and sufficient conditions for the centered problem are implicitly present in Kiefer and Wolfowitz's proof [53] of the equivalence of D- and G-optimality. The notion of ϵ-primal feasibility was introduced by Khachiyan [49], while ϵ-approximate optimality was defined by Ahipaşaoğlu, Sun, and Todd [3]. The relation between minimum-volume ellipsoids not necessarily centered in R^d and those centered in R^{d+1} appears in Titterington [79] and later in Khachiyan and Todd [50]. Our proof is from Kumar and Yıldırım [56].

We confine ourselves to discrete point sets $\{x_1, \ldots, x_m\}$ (and their convex hulls $\mathbf{X}$) throughout, both for simplicity and because we are interested in algorithms, but problems (P) and (D) can be stated in a more general setting. If $\mathbf{X}$ is a compact subset of R^n, the problem of finding a minimum-volume centered ellipsoid containing $\mathbf{X}$ can be formulated as the semi-infinite programming problem

$$(P) \quad \min_{H \in \mathcal{S}^n} \{-\mathrm{lndet}(H) : x^T H x \leq n \text{ for all } x \in \mathbf{X}\}.$$

Its dual has as its variable a (probability) measure μ on $\mathbf{X}$, and can be written as

$$(D) \quad \max_{\mu} \left\{ \mathrm{lndet}\left(\int_{\mathbf{X}} x x^T d\mu \right) : \int_{\mathbf{X}} d\mu = 1, \mu \text{ nonnegative} \right\}.$$

This duality relationship is due to Sibson [70]; see also Gürtuna [39], where the duals for this problem and for the maximum-volume inscribed ellipsoid problem are derived using semi-infinite programming. In fact, John's optimality conditions imply that it suffices to consider discrete measures μ, which put positive measure on a finite subset of points of $\mathbf{X}$, but it is sometimes simpler to consider arbitrary (Borel) measures μ. Analogues of the results we have proved hold in this setting also. For example, a feasible μ is optimal in (D) iff $H := (\int xx^T d\mu)^{-1}$ satisfies $x^T Hx \le n$ for all $x \in \mathbf{X}$, and μ is supported on $\{x \in \mathbf{X} : x^T Hx = n\}$.

The duality we have exhibited in this chapter has been used to explicitly find optimal designs in several statistical settings. It can also be used to obtain explicit formulae for the new ellipsoids after an iteration of the ellipsoid method. The minimum-volume ellipsoid $\mathcal{E}(H, \hat{y})$ containing $\mathcal{E}_{\alpha\beta}$ in (1.4.1) has

$$\hat{y} = \int_{\mathcal{E}_{\alpha\beta}} y\, d\mu, \quad H = \left(\int_{\mathcal{E}_{\alpha\beta}} yy^T d\mu - \hat{y}\hat{y}^T \right)^{-1}$$

for a suitable probability measure μ on $\mathcal{E}_{\alpha\beta}$. This measure puts a certain measure uniformly on the "latitude"

$$\{y \in \partial\mathcal{E} : a^T(y - \bar{y}) = \alpha\}$$

and the rest uniformly on

$$\{y \in \partial\mathcal{E} : a^T(y - \bar{y}) = \beta\}$$

(if $\beta = 1$, this is a point mass on the point that maximizes $a^T y$ over $\mathcal{E}$). The particular weights are easily obtained by calculus from (D), and then the optimality of the resulting ellipsoid can be checked using duality. The algebraic details are omitted.

The technique above is more systematic and flexible than the ad hoc methods of obtaining the successive ellipsoids in König and Pallaschke [54] and Todd [80]. See also the related independent work of Fogel and Huang [29] in the system identification context.

Chapter 3

Algorithms for the MVEE Problem

Here we will consider methods for solving the MVEE problem

$$\begin{array}{ll} \min_{H\in\mathcal{S}^n} & f(H) := -\ln\det(H) \\ (P) & x_i^T H x_i \le n, \quad i=1,2,\dots,m, \end{array}$$

and its dual

$$\begin{array}{ll} \max_{u\in\mathbf{R}^m} & g(u) := \ln\det(XUX^T) \\ (D) & e^T u = 1, \\ & u \ge 0. \end{array}$$

Since these problems cannot usually be solved exactly, we will be interested in finding ϵ-primal feasible or ϵ-approximately optimal solutions; see Definition 2.8.

Problem (P) has linear constraints and a convex objective function; moreover, its objective function is smooth with simple formulae for its first two derivatives:

$$\nabla(-\ln\det)(H) = -H^{-1}, \quad D^2(-\ln\det)(H)[E_1,E_2] = (H^{-1}E_1H^{-1})\bullet E_2$$

(see Section A.4).

Similarly, (D) has as its feasible region the unit simplex in R^m, and the chain rule gives its first two derivatives using those of the lndet function. Indeed, the gradient of the logdet function is the inverse of its argument, and the derivative of XUX^T with respect to u_i is just $x_i x_i^T$. Hence the chain rule shows that $\partial_i \ln\det(XUX^T) = (XUX^T)^{-1}\bullet x_i x_i^T$, where ∂_i denotes the partial derivative with respect to u_i. Next, (A.4.4) in Appendix A gives the directional derivative of the inverse, from which $\partial_j\partial_i \ln\det(XUX^T) = [-(XUX^T)^{-1}x_j x_j^T (XUX^T)^{-1}]\bullet x_i x_i^T$. Thus,

$$\begin{aligned} \omega(u) := \nabla g(u) &= (x_i^T H(u) x_i)_{i=1}^m, \\ (\nabla^2 g(u))_{ij} &= -(x_i^T H(u) x_j)^2, \quad i,j=1,\dots,m, \end{aligned}$$

where, as in the previous chapter,

$$H(u) := (XUX^T)^{-1}.$$

These simple formulae suggest that it would be worthwhile to use second-order algorithms related to Newton's method to optimize (P) and (D). However, the necessity of computing inverses of (or at least factorizing) $n\times n$ matrices at each iteration makes

these quite costly in a large-scale setting. We will therefore concentrate on first-order methods for solving (D).

Note that even these seem to require us to invert $n \times n$ matrices: see the formula for $\omega(u)$ above. However, if we employ a coordinate-ascent-type algorithm, where only one component of u is altered at each iteration, then XUX^T changes by a rank-one modification and its inverse is easily updated.

Two comments need to be made. First, coordinate ascent is usually not viewed as a first-order method: the latter term is normally confined to algorithms of "steepest-ascent" or conjugate-gradient type. However, note that steepest ascent *with respect to the ℓ_1-norm* gives a direction that is plus or minus a unit coordinate vector, and hence is a variant of coordinate ascent. Second, coordinate ascent cannot stay on the unit simplex: we will deal with this by renormalizing to maintain the sum of the components equal to 1.

The first section below describes two coordinate-ascent algorithms for (D) and how they can be implemented efficiently. In Section 3.2 we describe how to initialize the methods. Then Section 3.3 discusses the global convergence and complexity of the algorithms, and Section 3.4 their local convergence. In Section 3.5 we describe an intriguing relationship between the algorithms we have described and the deepest two-sided cut ellipsoid algorithm applied to the polar polytope. Section 3.6 shows how the points x_i can be eliminated during the course of the algorithms to improve efficiency. A potential application to spectral sparsification of graphs is discussed in Section 3.7, and then Section 3.8 illustrates the algorithms with computational results.

3.1 ▪ Coordinate-ascent algorithms

Let us suppose that we have a current feasible point u for (D); recall that this means that u satisfies the constraints, so u is nonnegative and $e^T u = 1$, and that the objective function $g(u) = \text{lndet}(XUX^T)$ is finite, so XUX^T is positive definite. Suppose we also have at our disposal

$$\omega := \omega(u) = \nabla g(u) = (x_i^T H x_i)_{i=1}^m, \tag{3.1.1}$$

where $H := H(u) := (XUX^T)^{-1}$, and a scaled Cholesky factorization of XUX^T:

$$XUX^T = \phi^{-1} LL^T \tag{3.1.2}$$

with ϕ positive. We will continue to talk about scaled Cholesky factorizations of XUX^T in what follows, but it is important to note that it is more numerically stable to compute these via a QR factorization of $U^{1/2}X^T$, where $U^{1/2}$ is a diagonal matrix with the square roots of the components of u on its diagonal. If $U^{1/2}X^T = QR$, with Q an $m \times n$ matrix with orthonormal columns and R an upper triangular $n \times n$ matrix, it is easy to see that $XUX^T = R^T R$. We use this factorization initially and when it is deemed worthwhile to refactorize; at other iterations, the Cholesky factor R^T is updated to maintain the scaled Cholesky factorization.

Consider the following update of u:

$$u_+ := (1-\tau)u + \tau e_i \tag{3.1.3}$$

with τ not equal to 1 and e_i the ith unit coordinate vector, which is also a vertex of the feasible region of (D). Note that u_+ can also be viewed as the result of taking a coordinate-ascent step to

$$\hat{u} := u + \lambda e_i$$

with $\lambda := \tau/(1-\tau)$, followed by a scaling to keep the coordinate sum equal to 1: $u_+ = (1+\lambda)^{-1}\hat{u} = (1-\tau)\hat{u}$. We see that u_+ remains nonnegative and not equal to e_i as long as

$$-u_i \le \lambda < \infty, \tag{3.1.4}$$

which we henceforth assume.

While this seems a very limited choice of an updated u, it leads to the following crucial fact. In this case, XU_+X^T is a scaled rank-one update of XUX^T:

$$XU_+X^T = (1+\lambda)^{-1}(XUX^T + \lambda x_i x_i^T). \tag{3.1.5}$$

As seen in Section A.3, this provides simple update formulae for the determinant and the inverse of XU_+X^T. From Corollary A.10 and Theorem A.11, we have

Proposition 3.1.

$$\det(XU_+X^T) = (1+\lambda)^{-n}(1+\lambda\omega_i)\det(XUX^T)$$

and, if $1+\lambda\omega_i$ *is positive,* XU_+X^T *is positive definite with*

$$(XU_+X^T)^{-1} = (1+\lambda)\left(H - \frac{\lambda}{1+\lambda\omega_i} Hx_i x_i^T H\right).$$

From this we obtain

$$g(u_+) - g(u) = -n\ln(1+\lambda) + \ln(1+\lambda\omega_i) \tag{3.1.6}$$

and, if $\hat{x} := Hx_i$,

$$\omega_h(u_+) = (1+\lambda)\left(\omega_h - \frac{\lambda}{1+\lambda\omega_i}(\hat{x}^T x_h)^2\right) \tag{3.1.7}$$

for each h. In particular,

$$\omega_i(u_+) = (1+\lambda)\left(\omega_i - \frac{\lambda\omega_i^2}{1+\lambda\omega_i}\right) = \frac{(1+\lambda)\omega_i}{1+\lambda\omega_i}. \tag{3.1.8}$$

Note that $\hat{x}$ can easily be obtained from the scaled Cholesky factorization of XUX^T. Moreover, since

$$\phi(1+\lambda)XU_+X^T = LL^T + \phi\lambda x_i x_i^T$$

is a rank-one modification of a Cholesky factorization, we can obtain a scaled Cholesky factorization of XU_+X^T in $O(n^2)$ operations.

An iteration of each of our algorithms consists in replacing u by u_+ as above, with specific choices for i and λ. In both cases, λ is chosen to maximize $g(u_+)$, viewed as a function $\gamma(\lambda)$ of the chosen stepsize. From (3.1.6) we find

$$\gamma'(\lambda) = -\frac{n}{1+\lambda} + \frac{\omega_i}{1+\lambda\omega_i}.$$

From this, a little analysis and algebra yield that the optimal λ is the unique root of γ',

$$\lambda^* := \frac{\omega_i - n}{(n-1)\omega_i}, \tag{3.1.9}$$

yielding an increase in g of

$$(n-1)\ln\left(\frac{(n-1)\omega_i}{n(\omega_i-1)}\right)+\ln\left(\frac{\omega_i}{n}\right) \tag{3.1.10}$$

as long as the formula gives $\lambda^* \geq -u_i$. If this condition fails, $\gamma'(\lambda)$ is negative for all feasible λ and so the optimal λ is $-u_i$. The reader should check that, if $\lambda = \lambda^*$, then $\omega_i(u_+) = n$, and consider why this is to be expected.

How should we choose the index i? We will give several motivations. First, recall our initial statement of a dual problem for (P): in (2.1.4) we gave an optimization problem (D') over just the nonnegative orthant, where we wanted to maximize $g(u)+n-ne^T u$. Since there is no equality constraint, coordinate ascent is quite natural for this problem. The gradient of the objective function of (D') at u is $\omega - ne$, and steepest ascent with respect to the ℓ_1-norm would lead to a choice of i with $\omega_i - n$ maximal or j with $\omega_j - n$ minimal. Of course, we cannot decrease components that are already zero, so we should confine our choice for j to those indices with $u_j > 0$.

Next, let us consider the optimality conditions for (D). If we just consider condition (i) of Proposition 2.6, we see that the worst violation of the condition occurs for i with $\omega_i - n$ maximal. If we consider also condition (ii), we should also examine j with $u_j > 0$ and $\omega_j - n$ minimal. Here we should note that

$$u^T\omega = \sum_i u_i x_i^T H x_i = \left(\sum u_i x_i x_i^T\right) \bullet H = (XUX^T) \bullet (XUX^T)^{-1} = n, \tag{3.1.11}$$

so that, unless u is optimal, there is always an index i with $\omega_i > n$ and an index j with $u_j > 0$ and $\omega_j < n$. Note that, if u is ϵ-primal feasible but not ϵ'-primal feasible for any $\epsilon' < \epsilon$, index i above is the critical index with $x_i^T H x_i = (1+\epsilon)n$. Similarly, if u is ϵ-approximately optimal but not ϵ'-approximately optimal for any $\epsilon' < \epsilon$, the critical index is either i with $x_i^T H x_i = (1+\epsilon)n$ or j with $u_j > 0$ and $x_j^T H x_j = (1-\epsilon)n$.

For our third motivation, let us make a first-order linear (Taylor) approximation to the objective function g:

$$g(u+\Delta u) \approx \tilde{g}(u+\Delta u) := g(u) + \omega^T \Delta u.$$

We might then consider maximizing this linear function over the feasible region, the unit simplex. A linear function is maximized at a vertex, and clearly that vertex is the unit vector e_i where ω_i is maximal. Of course, the true function g is nonlinear, and so our linear approximation is only appropriate close to the current solution u. Hence we might wish to move along the line from u towards e_i, i.e., consider points of the form (3.1.3) for positive τ. This is the motivation for the classical Frank–Wolfe algorithm. It was independently proposed for the particular problem (P) by Fedorov and Wynn (the latter with a fixed rather than optimal stepsize). We therefore refer to it unambiguously as the FW Algorithm.

The current solution u can be viewed as a convex combination of all the e_j's with $u_j > 0$. Some of these vertices are better than others, and it therefore makes sense to reduce the weight on the worst vertex. According to $\tilde{g}$, this is the vector e_j with $u_j > 0$ and ω_j minimal. Hence we might wish to move along the line from e_j to u extended, or in other words, move *away from* e_j. This is again a point of the form (3.1.3), now with j replacing i and with negative τ. The resulting method (choosing either e_i with maximal

ω_i or e_j with minimal ω_j) is Wolfe's algorithm with away steps. It was independently proposed for the problem (D) by Atwood, and so we will refer to it as the WA Algorithm.

Let us state these algorithms more formally.

ALGORITHM 3.1.
(FW Algorithm)

Step 0. Choose u feasible for (D) and $\epsilon > 0$.
Compute $\omega = \omega(u)$ and a (scaled) Cholesky factorization of XUX^T.
Step 1. Given the current iterate u and its associated $\omega := \omega(u)$, compute $\epsilon_+ := \max_h(\omega_h - n)/n$, and let $h = i$ attain the maximum.
If $\epsilon_+ \leq \epsilon$, STOP: u is ϵ-primal feasible. Otherwise, go to Step 2.
Step 2. Compute λ^* from (3.1.9) and update $u \leftarrow (1+\lambda^*)^{-1}(u + \lambda^* e_i)$.
Step 3. Update ω and a scaled Cholesky factorization of XUX^T and go to Step 1.

ALGORITHM 3.2.
(WA Algorithm)

Step 0. Choose u feasible for (D) and $\epsilon > 0$.
Compute $\omega = \omega(u)$ and a (scaled) Cholesky factorization of XUX^T.
Step 1. Given the current iterate u and its associated $\omega := \omega(u)$, compute $\epsilon_+ := \max_h(\omega_h - n)/n$, with $h = i$ attaining the maximum, and $\epsilon_- := \max_h\{(n - \omega_h)/n : u_h > 0\}$, with $h = j$ attaining the maximum.
If $\max\{\epsilon_+, \epsilon_-\} \leq \epsilon$, STOP: u is ϵ-approximately optimal.
Otherwise, if $\epsilon_+ > \epsilon_-$, go to Step 2; else go to Step 3.
Step 2. Compute λ^* from (3.1.9) and update $u \leftarrow (1+\lambda^*)^{-1}(u + \lambda^* e_i)$. Go to Step 4.
Step 3. Set $\lambda = \max\{-u_j, \lambda^*\}$, where j replaces i in the definition of λ^* in (3.1.9). Update $u \leftarrow (1+\lambda)^{-1}(u + \lambda e_j)$. Go to Step 4.
Step 4. Update ω and a scaled Cholesky factorization of XUX^T and go to Step 1.

As stated, both algorithms only update ω and a scaled Cholesky factorization. However, it might be advisable to recompute these from time to time. Here is a suggested implementation. At each iteration, i (or possibly j in the WA Algorithm) is first chosen. Then it is necessary to compute $\hat{x} = Hx_i$ (or possibly $\hat{x} = Hx_j$; below we assume that Hx_i is needed, but obvious changes can be made in the latter case) in order to update ω: see (3.1.7). This is done by first calculating $z = L^{-1}x_i$. At this stage we can find a new estimate for ω_i: $\phi z^T z$. If the relative error of the old value compared to this exceeds, say, 10^{-8}, we can recompute XUX^T and its Cholesky factorization (actually the QR factorization of $U^{1/2}X^T$, as discussed above), and hence recompute ω. Otherwise, we proceed to compute $\hat{x} = \phi L^{-T} z$ and continue with the updated quantities. In addition, every, say, $\max\{n, 50000\}$ iterations, we can recompute XUX^T and compare ϕXUX^T to LL^T; again, if the relative error exceeds some threshold, we can recompute the Cholesky factorization and hence ω.

Each iteration requires $O(n^2)$ arithmetic operations to compute $\hat{x}$ and update the scaled Cholesky factorization of XUX^T. Selecting i or j and obtaining the optimal stepsize require $O(m)$ and $O(1)$ work. The dominant work at each iteration is updating ω, which requires $O(mn)$ arithmetic operations to calculate each $\hat{x}^T x_h$. Moreover, if we

only refactorize every $\Omega(n)$ iterations, the average work from refactorizing is also only $O(n^2)$. The cheapness of the iterations in our algorithms, together with their (relatively) attractive convergence properties, leads to their efficiency.

3.2 ▪ Initialization

We now discuss two ways to choose the initial vector u. The first choice is due to Khachiyan and is very straightforward: we choose $u = u_K := e/m$, putting equal weight on each point. Nevertheless, this choice provides a guaranteed quality of approximation. To state the next result, it is convenient to introduce notation to indicate how close to primal feasibility or optimality a given u is.

For feasible u, denote

$$\delta(u) := \delta_+(u) := \max_b(\omega_b(u) - n)/n, \tag{3.2.1}$$

$$\delta_-(u) := \min_b\{(\omega_b(u) - n)/n : u_b > 0\}, \tag{3.2.2}$$

and

$$\bar{\delta}(u) := \max\{\delta_+(u), -\delta_-(u)\}, \tag{3.2.3}$$

so that $\delta(u)$ is the smallest ϵ for which u is ϵ-primal feasible, and $\bar{\delta}(u)$ is the smallest ϵ such that u is ϵ-approximately optimal. As in Theorem 2.2, we use g^* to denote the optimal value of (D).

Proposition 3.2. *For $u_K = e/m$, we have $\delta(u_K) \le m-1$, $\bar{\delta}(u_K) \le m-1$, and $g^* - g(u_K) \le n \ln m$.*

Proof. We first note that u is feasible since it is positive and, by assumption, the columns of X span R^n. Next, from (3.1.11), $(1/m)e^T\omega = u^T\omega(u) = n$, so that each component of $\omega(u)$ is at most mn (and at least 0). This proves the bounds on $\delta(u)$ and $\bar{\delta}(u)$, and the bound on $g(u)$ follows from Proposition 2.9. □

The second initialization scheme is due to Kumar and Yıldırım. We will describe it for an arbitrary set of points x_i, $i = 1, \ldots, m$, and then outline the simplification that occurs when (as in our centered case), each x_i represents both x_i and $-x_i$.

The method first finds at most $2n$ points using an algorithm of Betke and Henk.

ALGORITHM 3.3.
(BH Algorithm)
Step 0. Choose an arbitrary nonzero $c_1 \in \mathrm{R}^n$ and set $j = 1$.
Step 1. Let $\overline{z}_j$ and $\underline{z}_j$ maximize and minimize $c_j^T x$ over the x_b's. Set $y_j := \overline{z}_j - \underline{z}_j$.
Step 2. If $j < n$, choose an arbitrary nonzero c_{j+1} orthogonal to $y_1, \ldots, y_j$, increase j by 1, and go to Step 1. Otherwise stop.

It is easy to see that this algorithm requires $O(n^2 m)$ arithmetic operations. Kumar and Yıldırım then choose $u = u_{KY}$ to put equal weight on each of the distinct points of $Z := \{\overline{z}_j, \underline{z}_j : j = 1, \ldots, n\}$ (note that there may be repetitions, so there are at most $2n$ such points).

Now suppose that each x_i represents the pair of points $\pm x_i$. Then if $\overline{z}_j$ is $\pm x_{b(j)}$, $\underline{z}_j$ can be chosen to be $\mp x_{b(j)}$ and $y_j = \pm 2x_{b(j)}$, so that $\pm x_{b(j)}$ will not be chosen in any

subsequent step. Thus exactly n pairs of points $\pm x_{b(j)}$ will be chosen, and without loss of generality we can put weight $1/n$ on each $x_{b(j)}$, since in XUX^T, any weight placed on $(-x_i)(-x_i)^T$ can be transferred to $x_i x_i^T$.

Let us return to the general case. To analyze the quality of this initialization, we need to consider two other polytopes:

$$\begin{aligned} \overline{C} &:= \{x \in \mathrm{R}^n : c_j^T \underline{z}_j \le c_j^T x \le c_j^T \overline{z}_j,\ j = 1,\dots,n\}, \\ \underline{C} &:= \mathrm{conv}(Z). \end{aligned} \tag{3.2.4}$$

It is clear that $\underline{C} \subseteq \mathbf{X} \subseteq \overline{C}$. Indeed, these two polytopes provide a guaranteed quality-of-fit approximation to $\mathbf{X}$ in terms of volume, by the following result.

Proposition 3.3. *We have*

$$\mathrm{vol}(\overline{C}) = |\det(y_1,\dots,y_n)|, \quad \mathrm{vol}(\underline{C}) \ge \frac{1}{n!}|\det(y_1,\dots,y_n)|,$$

and hence

$$\mathrm{vol}(\underline{C}) \ge \frac{1}{n!}\mathrm{vol}(\mathbf{X}). \tag{3.2.5}$$

Proof. The proof is by induction on n, the result being trivial for $n = 1$. So suppose it is true for dimensions less than n, and consider a case of dimension n. Since volumes are invariant under rotations, we may assume without loss of generality that y_1 is $||y_1||e_n$. We now let H be the hyperplane $\{x \in \mathrm{R}^n : e_n^T x = 0\}$, which is naturally identified with R^{n-1}, and consider the *Steiner symmetrizations* of $\mathbf{X}$, $\overline{C}$, and $\underline{C}$ with respect to H.

For any convex body $K \subseteq \mathrm{R}^n$, the Steiner symmetrization of K with respect to a hyperplane H with normal v is obtained as follows. For every $x \in H$ such that $\{x + \lambda v : \lambda \in \mathrm{R}\}$ intersects K, let μ (respectively, ν) be the maximum (respectively, minimum) value of λ that yields a point in K. Then replace the segment joining $x + \mu v$ and $x + \nu v$ with the equal-length segment joining $x + [(\mu - \nu)/2]v$ and $x - [(\mu - \nu)/2]v$, which is symmetric with respect to H. It can be shown that, if K is a polytope, then so is its Steiner symmetrization. Moreover, Steiner symmetrization preserves volume.

Let us now consider the result of Steiner symmetrization in our context. It is convenient to denote by x^p the projection of any point $x \in \mathrm{R}^n$ onto H, and similarly by K^p the projection of a set $K \subseteq \mathrm{R}^n$ onto H. Projections onto H are also clearly preserved under Steiner symmetrization. Since the c_j, $j > 1$, are all orthogonal to y_1, the Steiner symmetrization of $\overline{C}$ is

$$\{x \in \mathrm{R}^n : |e_n^T x| \le ||y_1||/2,\ c_j^T \underline{z}_j \le c_j^T x \le c_j^T \overline{z}_j,\ j = 2,\dots,n\}$$

with volume

$$\mathrm{vol}(\overline{C}) = ||y_1||\mathrm{vol}_{n-1}(\{x \in H : c_j^T \underline{z}_j \le c_j^T x \le c_j^T \overline{z}_j,\ j = 2,\dots,n\}). \tag{3.2.6}$$

(Here, vol_{n-1} denotes $(n-1)$-dimensional volume.)

Next, the Steiner symmetrization of $\underline{C}$ clearly contains all the points $\overline{z}_j^p$, $\underline{z}_j^p$, $j = 2,\dots,n$, as well as the points $\overline{z}_1^p \pm (y_1/2)e_n$, and so it contains the two pyramids with apices at the latter two points and bases equal to the convex hull of the former points. Hence it has volume

$$\mathrm{vol}(\underline{C}) \ge \frac{||y_1||}{n}\mathrm{vol}_{n-1}(\mathrm{conv}(\overline{z}_2^p, \underline{z}_2^p, \dots, \overline{z}_n^p, \underline{z}_n^p)). \tag{3.2.7}$$

Finally, consider the result of applying the BH Algorithm to the $(n-1)$-dimensional data $\{x_2^p,\dots,x_m^p\}$ in H. (Here and for the rest of the proof, final zero components of vectors should be suppressed.) It is clear that $c_2,\dots,c_n$ could be chosen exactly as before, and that the points $\overline{z}_j^p$ and $\underline{z}_j^p$, $j=2,\dots,n$, would then be chosen, with corresponding differences y_j^p, $j=2,\dots,n$. The induction hypothesis then gives

$$\mathrm{vol}_{n-1}(\{x\in H : c_j^T\underline{z}_j \le c_j^T x \le c_j^T\overline{z}_j, j=2,\dots,n\}) = |\det(y_2^p,\dots,y_n^p)|$$

and

$$\mathrm{vol}_{n-1}(\mathrm{conv}(\overline{z}_2^p,\underline{z}_2^p,\dots,\overline{z}_n^p,\underline{z}_n^p)) \ge \frac{1}{(n-1)!}|\det(y_2^p,\dots,y_n^p)|.$$

Since expansion down its first column shows

$$|\det(y_1,\dots,y_n)| = \|y_1\||\det(y_2^p,\dots,y_n^p)|,$$

using (3.2.6) and (3.2.7) completes the inductive step. □

Corollary 3.4. *For u_{KY} obtained as above, we have*

$$g^* - g(u_{KY}) \le 5n\ln n.$$

If each x_i represents $\pm x_i$, the bound improves to $4n\ln n$.

Proof. Let $\bar{g}$ denote the optimal solution to (D) when we consider only the points $\overline{z}_j$ and $\underline{z}_j$, $j=1,\dots,n$, instead of all the points x_i. Since there are at most $2n$ such points, we have $\bar{g} - g(u_{KY}) \le n\ln(2n) = n\ln n + n\ln 2$ by Proposition 3.2. By strong duality (Theorem 2.2), $\bar{g}$ is also the optimal value of the corresponding primal problem, and using (1.2.1), we find that

$$\bar{g} = 2\ln\mathrm{vol}(E_*(\underline{C})) - n\ln n - 2\ln\omega_n.$$

Similarly,

$$g^* = 2\ln\mathrm{vol}(E_*(\mathbf{X})) - n\ln n - 2\ln\omega_n.$$

It now suffices to relate these two volumes. But

$$\begin{aligned}\mathrm{vol}(E_*(\underline{C})) &\ge \mathrm{vol}(\underline{C})\\ &\ge \tfrac{1}{n!}\mathrm{vol}(\mathbf{X})\\ &\ge \tfrac{1}{n^n n!}\mathrm{vol}(E_*(\mathbf{X})),\end{aligned}$$

where the second inequality follows from (3.2.5) and the last from Theorem 1.1, since $(1/n)E_*(\mathbf{X})$ is contained in $\mathbf{X}$. Thus $g^* - \bar{g} \le 2\ln(n^n n!)$. A crude bound on this is $4n\ln n$, but using Stirling's formula gives a slightly lower estimate to cancel the $n\ln 2$ term in the bound on $\bar{g} - g(u_{KY})$, and hence yields the first inequality of the lemma. For the second, if each x_i represents $\pm x_i$, so that u_{KY} puts weight $1/n$ on just n points x_i, then Example 2.4 shows that u_{KY} is in fact optimal for the restricted problem, and so we save one $n\ln n$ in the bound. □

Although the bound in Corollary 3.4 is only better than that in Proposition 3.2 for $m > n^5$, in practice the Kumar–Yıldırım initialization turns out to be far more preferable.

3.3 ▪ Global convergence and complexity

We now turn to the iteration complexity of our two algorithms. Consider first the FW Algorithm. Suppose we are at an iterate u, with $\delta := \delta(u) = (\omega_i(u) - n)/n > 0$, and that the next iterate is denoted u_+. Note that, for this algorithm, the stepsize is always the optimal stepsize λ^*. We then have

Lemma 3.5.

$$g(u_+) - g(u) \geq \varkappa(\delta) := \ln(1+\delta) - \frac{\delta}{1+\delta}. \tag{3.3.1}$$

Proof. According to (3.1.10), we have

$$g(u_+) - g(u) = (n-1)\ln\left(\frac{(n-1)\omega_i}{n(\omega_i - 1)}\right) + \ln\left(\frac{\omega_i}{n}\right),$$

so that substituting $\omega_i = n(1+\delta)$ yields

$$\begin{aligned} g(u_+) - g(u) &= \ln(1+\delta) - (n-1)\ln\left(\tfrac{n(1+\delta)-1}{(n-1)(1+\delta)}\right) \\ &= \ln(1+\delta) - (n-1)\ln\left(1 + \tfrac{\delta}{(n-1)(1+\delta)}\right) \\ &\geq \ln(1+\delta) - \tfrac{\delta}{1+\delta}, \end{aligned}$$

where the inequality follows from $\ln(1+\beta) \leq \beta$ for all $\beta > -1$. □

We also require some estimates of the right-hand side $\varkappa(\delta)$ of (3.3.1); for later purposes, we also need to consider negative values of δ.

Lemma 3.6. *We have that*

(i) $\varkappa(\delta)$ *is increasing for* $\delta > 0$ *and decreasing for* $\delta < 0$*;*

(ii) *for* $\delta \geq 1$, $\varkappa(\delta) \geq \frac{1}{4}\ln(1+\delta)$*; and*

(iii) *for* $|\delta| \leq \frac{1}{2}$, $\varkappa(\delta) \geq \frac{2}{7}\delta^2$.

Proof. We find $\varkappa'(\delta) = \delta/(1+\delta)^2$, which is positive for positive δ and negative for negative δ. This proves (i).

Now let $\mu(\delta) := \varkappa(\delta)/\ln(1+\delta)$ for positive δ. Then a short computation yields

$$\mu'(\delta) = \frac{\delta - \ln(1+\delta)}{[(1+\delta)\ln(1+\delta)]^2},$$

which is positive for positive δ. Thus for $\delta \geq 1$,

$$\varkappa(\delta) \geq \frac{\varkappa(1)}{\ln(1+1)}\ln(1+\delta) \geq \frac{1}{4}\ln(1+\delta),$$

establishing (ii).

Finally, let $\nu(\delta) := \varkappa(\delta)/\delta^2$ for nonzero δ with absolute value at most a half, $\nu(0) := 1/2$. It is easily seen that ν is continuous at zero, and for nonzero δ, we find

$$\nu'(\delta) = \frac{-2\delta(1+\delta)^2\ln(1+\delta) + 2\delta^2 + 3\delta^3}{\delta^4(1+\delta)^2}.$$

From the power series for $\ln(1+\delta)$, we obtain

$$\delta\ln(1+\delta) \geq \delta^2 - \frac{\delta^3}{2} + \frac{\delta^4}{3} - \frac{\delta^5}{4},$$

and substituting this into the denominator above, we get

$$\nu'(\delta) \leq \frac{-2\delta^4/3 + \delta^5/2 + \delta^6 + \delta^7/2}{\delta^4(1+\delta)^2} < 0$$

for nonzero δ with absolute value at most a half. We conclude that for such δ,

$$\varkappa(\delta) \geq \frac{\varkappa(1/2)}{(1/2)^2}\delta^2 \geq \frac{2}{7}\delta^2. \qquad \square$$

We divide the analysis of the iteration complexity of the FW Algorithm into two steps.

Lemma 3.7. *The number of iterations for the FW Algorithm to reach an iterate u with $\delta(u) \leq 1$ is at most*

(i) $4n(\ln\ln m + 3/2)$ *if the initial iterate is u_K, and*

(ii) $4n(\ln\ln n + 7/2)$ *if the initial iterate is u_{KY}.*

Proof. Consider any iteration proceeding from the iterate u with $\delta := \delta(u) \geq 1$ to the next iterate u_+, and let $\gamma := g^* - g(u)$ and $\gamma_+ := g^* - g(u_+)$ denote the corresponding optimality gaps. From Lemmas 3.5 and 3.6 we have

$$\gamma - \gamma_+ = g(u_+) - g(u) \geq \ln(1+\delta) - \frac{\delta}{1+\delta} \geq \frac{1}{4}\ln(1+\delta) \geq \frac{1}{4n}\gamma, \tag{3.3.2}$$

where the last inequality follows from Proposition 2.9. We conclude that

$$\gamma_+ \leq \left(1 - \frac{1}{4n}\right)\gamma \leq \exp\left(-\frac{1}{4n}\right)\gamma.$$

Since the initial value of γ is at most $n\ln m$ if we use u_K from Proposition 3.2, we deduce that within

$$4n\ln\left(\frac{n\ln m}{2n/3}\right) \leq 4n\left(\ln\ln m + \frac{1}{2}\right)$$

iterations, γ is at most $2n/3$. Moreover, while δ is at least 1, γ decreases by at least $\ln(1+1) - 1/(1+1) \geq 1/6$, so since γ remains nonnegative, at most $(2n/3)/(1/6) = 4n$ further iterations are possible. This establishes the first part of the lemma. The proof is similar for the Kumar–Yıldırım initialization, using Corollary 3.4 to bound the initial γ. $\square$

Lemma 3.8. *The number of iterations for the FW Algorithm to reduce $\delta(u)$ from a value at most $\delta \in (0,1]$ to $\delta/2$ is at most $14n/\delta$.*

Proof. Suppose we have an iterate $\hat{u}$ with $\delta(\hat{u}) \leq \delta$. Then by Proposition 2.9, we have

$$g^* - g(\hat{u}) \leq n\delta. \tag{3.3.3}$$

By Lemma 3.5 and parts (i) and (iii) of Lemma 3.6, at every subsequent iterate u with $\delta(u) \geq \delta/2 \leq 1/2$ and following iterate u_+, we have

$$g(u_+) - g(u) \geq \varkappa\left(\frac{\delta}{2}\right) \geq \frac{2}{7}\left(\frac{\delta}{2}\right)^2 = \frac{1}{14}\delta^2.$$

Hence there can be at most $(n\delta)/(\delta^2/14) = 14n/\delta$ such iterations. □

Putting these two lemmas together, we obtain

Theorem 3.9. *The total number of iterations for the FW Algorithm to reach an ϵ-primal feasible u is at most*

$$4n(\ln\ln m + 3/2) + 28n/\epsilon \tag{3.3.4}$$

with the initial iterate u_K, and at most

$$4n(\ln\ln n + 7/2) + 28n/\epsilon \tag{3.3.5}$$

with the initial iterate u_{KY}.

Proof. The first term in each bound above gives the number of iterations to obtain an iterate u with $\delta(u) \leq 1$ in the two cases. Now let $k := \lceil \log_2(1/\epsilon) \rceil$, so that $2^{-k} \leq \epsilon$. The number of iterations to reduce $\delta(u)$ from at most 1 to at most 2^{-k} can be divided into the number to reduce it from at most 1 to at most $2^{-1}, \ldots$, plus the number to reduce it from at most 2^{-k+1} to at most 2^{-k}. From Lemma 3.8, the total for this second phase is at most

$$14n\left(\frac{1}{1} + \frac{1}{1/2} + \cdots + \frac{1}{2^{-k+1}}\right) < 14n\,2^k < 28n/\epsilon. \qquad \square \tag{3.3.6}$$

Let us now turn to the WA Algorithm. It is helpful to divide the iterations into four types. An *add* iteration is one where a component of u increases from zero to a positive level, while an *increase* iteration is one where an already positive component u_i is chosen to be increased. Analogously, a *drop* iteration is one where a component of u decreases to zero, while a *decrease* iteration is one where a positive component u_j is chosen to be decreased, but it decreases to a positive level. Note that, in all but drop iterations, an optimal stepsize is chosen so as to obtain a guaranteed increase in the objective function.

To analyze this increase, we use both

$$\delta_+(u) := \max_b (\omega_b(u) - n)/n \quad (= \delta(u))$$

and

$$\delta_-(u) := \min_b \{(\omega_b(u) - n)/n : u_b > 0\}$$

with $\bar{\delta}(u) = \max(\delta_+(u), -\delta_-(u))$.

Lemma 3.10. *In any add, increase, or decrease iteration at an iterate u with $\bar{\delta}(u) =: \delta \leq 1/2$,*

$$g(u_+) - g(u) \geq \frac{2}{7}\delta^2.$$

Proof. For an add or increase iteration, $\delta = \delta_+(u)$ and the result follows from Lemmas 3.5 and 3.6. Consider now a decrease iteration. Then $\delta = -\delta_-(u)$, and the proof of Lemma 3.5 can be repeated to show that

$$g(u_+) - g(u) \geq \varkappa(-\delta). \tag{3.3.7}$$

Then part (iii) of Lemma 3.6 shows that the right-hand side is at least $2\delta^2/7$, as desired. □

Consider the proof of Theorem 3.9. As long as $\bar{\delta}(u) > 1$, we have $\bar{\delta}(u) = \delta_+(u)$, and an add or increase iteration is taken. Hence the number of iterations for the WA Algorithm to attain $\bar{\delta}(u) \leq 1$ is exactly the same as for the FW Algorithm. Next consider the argument for the number of steps to reduce $\bar{\delta}$ from $\delta \leq 1$ to $\delta/2$. Using Lemma 3.10, (3.3.7), and parts (i) and (iii) of Lemma 3.6, we find that the number of add, increase, and decrease iterations in this phase is at most $14n/\delta$. Since drop iterations offer no guaranteed increase in g, we cannot nicely bound the number of drop iterations in such a phase. However, each drop iteration reduces to zero a component of u that was either positive initially (numbering at most m for the Khachiyan, or $2n$ for the Kumar–Yıldırım initialization), or was increased from zero in an add iteration. Hence we can bound the number of drop iterations by the sum of m or $2n$ and the number of add iterations. Putting these pieces together yields the following.

Theorem 3.11. *The total number of iterations for the WA Algorithm to reach an ϵ-approximately optimal u is at most*

$$4n(\ln\ln m + 3/2) + m + 56n/\epsilon \tag{3.3.8}$$

with the initial $u = u_K$, and at most

$$4n(\ln\ln n + 4) + 56n/\epsilon \tag{3.3.9}$$

with the initial $u = u_{KY}$.

Example 3.12. We consider Example 2.3 yet again. Recall that

$$X = \begin{bmatrix} -1 & -1 & 1 & 2 \\ 1 & -1 & -1 & 2 \end{bmatrix}$$

and each column x_i represents $\pm x_i$. If we use the Kumar–Yıldırım initialization, then any choice of c_1 and c_2 will lead to $Z = \{\pm x_1, \pm x_4\}$ (note that $x_3 = -x_1$). Then the initial u is $(1/2; 0; 0; 1/2)$ (or $(0; 0; 1/2; 1/2)$) and these are both optimal, so either the FW or the WA Algorithm terminates immediately. (See Example 2.3.)

Suppose instead that we use the Khachiyan initialization, so that the initial iterate is $u = (1/4; 1/4; 1/4; 1/4)$. Then

$$XUX^T = \begin{bmatrix} 7/4 & 3/4 \\ 3/4 & 7/4 \end{bmatrix}$$

with inverse

$$H(u) = \begin{bmatrix} 7/10 & -3/10 \\ -3/10 & 7/10 \end{bmatrix}.$$

We then calculate $\omega(u) = (2; 8/10; 2; 32/10)$, so that $\delta(u) = \delta_+(u) = (32/10 - 2)/2 = 3/5$ and $\delta_-(u) = (8/10 - 2)/2 = -3/5$.

The FW Algorithm chooses u_4 to increase. Then the stepsize is

$$\lambda^* = \frac{\delta_+(u)}{(n-1)(1+\delta_+(u))} = \frac{3}{8},$$

leading to $u_+ = (2/11; 2/11; 2/11; 5/11)$. Subsequent iterations are somewhat tedious to compute by hand. Suffice to say that the algorithm takes eight iterations to reduce $\delta(u)$ to below 0.1, and the successive iterates are

$$\begin{pmatrix} 5/14 \\ 1/7 \\ 1/7 \\ 5/14 \end{pmatrix}, \begin{pmatrix} 5/17 \\ 2/17 \\ 2/17 \\ 8/17 \end{pmatrix}, \begin{pmatrix} 2/5 \\ 1/10 \\ 1/10 \\ 2/5 \end{pmatrix}, \begin{pmatrix} 8/23 \\ 2/23 \\ 2/23 \\ 11/23 \end{pmatrix},$$

$$\begin{pmatrix} 11/26 \\ 1/13 \\ 1/13 \\ 11/26 \end{pmatrix}, \begin{pmatrix} 11/29 \\ 2/29 \\ 2/29 \\ 14/29 \end{pmatrix}, \begin{pmatrix} 7/16 \\ 1/16 \\ 1/16 \\ 7/16 \end{pmatrix}.$$

The corresponding expanding ellipses are shown in Figure 3.1, alternating between red ellipses going through the points $\pm(1;-1)$ and blue ellipses going through $(2;2)$. The slow convergence is clear.

Let us contrast this with the results of applying the WA Algorithm. In this case we choose to reduce u_2; since $\lambda^* = -3/2 < -1/4 = -u_2$, we set $\lambda = -u_2 = -1/4$ and move to $u_+ = (1/3; 0; 1/3; 1/3)$. Again we omit the details of the subsequent calculations, but at this new point we choose to increase u_4 and move immediately to the optimal solution $u = (1/4; 0; 1/4; 1/2)$. The corresponding three ellipses are shown in Figure 3.2, going from red to blue to green. Notice that the blue ellipse does not go through any of the points: since it is the result of a drop iteration, an optimal stepsize could not be taken. ∎

The reader is encouraged to experiment with other examples using the MATLAB code `minvol.m`, which is included in Appendix B and is also available at `www.siam.org/books/mo23`.

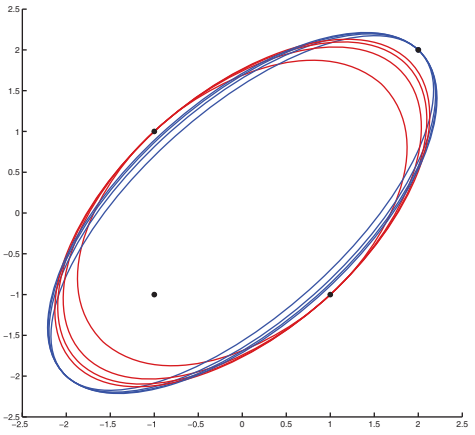


Figure 3.1. *Ellipses generated by the FW Algorithm.*

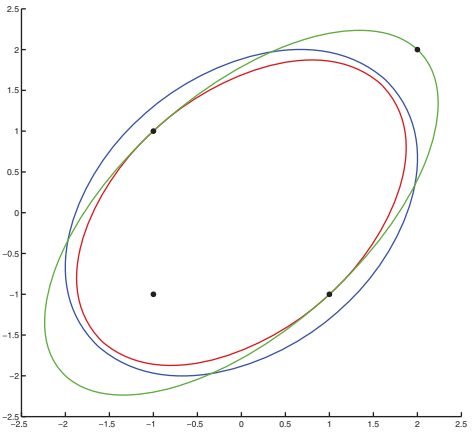


Figure 3.2. *Ellipses generated by the WA Algorithm.*

3.4 ▪ Local convergence

Although the global complexity bound for the FW Algorithm is better than that for the WA Algorithm, the dramatic difference in their practical performance in favor of the latter, as illustrated in the toy Example 3.12, is apparent in a wide variety of settings. Indeed, experimentation suggests that the number of iterations, rather than growing with ϵ^{-1}, seems to depend linearly on $\ln(\epsilon^{-1})$, i.e., linear convergence is exhibited. We prove that local linear convergence holds generally in this section.

The key is to consider the following perturbed version of the MVEE problem (P), where the right-hand side is perturbed by the vector $v \in \mathbf{R}^m$:

$$\begin{array}{ll} \min_{H \in \mathcal{S}^n} & -\ln\det(H) \\ (P(v)) & x_i^T H x_i \quad \leq n + v_i, \quad i = 1, 2, \ldots, m. \end{array} \tag{3.4.1}$$

Note that $(P(0))$ is our original problem (P). Suppose we are given a δ-approximately optimal u, and we set

$$v_i := v_i(u) := \begin{cases} \delta n & \text{if } u_i = 0, \\ x_i^T H(u) x_i - n & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, m$. Observe that each component of v is at most δn in absolute value, and that

$$u^T v = \sum_{i:u_i>0} u_i v_i = \sum_{i:u_i>0} u_i(\omega_i(u) - n) = u^T \omega(u) - n e^T u = n - n = 0, \tag{3.4.2}$$

using (3.1.11). In contrast, if u is δ-primal feasible but not δ-approximately optimal, then v as defined above has some components greater than δn in absolute value; if we redefine v as $\delta n e$, then its components are small but $u^T v = \delta n$.

What is the significance of choosing v in this way and the key equation (3.4.2)? First, observe that by its definition, and since u is assumed to be δ-approximately optimal, $H(u)$ is feasible in $(P(v))$. Indeed, we have

Proposition 3.13. *$H(u)$ is an optimal solution to $(P(v(u)))$.*

Proof. We have already observed that $H(u)$ is feasible. Now

$$-H(u)^{-1} + \sum_i u_i x_i x_i^T = 0$$

by the definition of $H(u)$, and if u_i is positive,

$$x_i^T H(u) x_i = n + \nu_i(u)$$

by the definition of $\nu_i(u)$. Hence the Karush–Kuhn–Tucker conditions hold at $H(u)$ with multipliers u, and since $(P(\nu(u)))$ is a convex problem, we conclude that $H(u)$ is optimal. □

Second, the sensitivity of the optimal value of a nonlinear programming problem to its right-hand sides is given by the negative of its multipliers. Hence we should expect the optimal value of (P) to differ from that of $(P(\nu))$ to first order by $(-u)^T(-\nu) = 0$. Thus we might hope that $g(u) = f(H(u))$ would differ from $g^* = f_*$ by much less than $O(||\nu||) = O(\delta)$, and this would dramatically improve the analysis of the previous section.

In fact, the optimal value function is convex, and so if we base our analysis on $(P(\nu))$ the inequality turns out to be going the wrong way. We therefore base it on $(P(0))$, which is just (P). So let H^* be an optimal solution to (P), and let u^* be any corresponding vector of multipliers. Then $(H^*)^{-1} = X U^* X^T$, and since f is convex,

$$\begin{aligned} g(u) = f(H(u)) &\geq f(H^*) + (-(H^*)^{-1}) \bullet (H(u) - H^*) \\ &= f(H^*) - \textstyle\sum_{i:u_i^*>0} u_i^* (x_i^T H(u) x_i - x_i^T H^* x_i) \\ &\geq g^* - \textstyle\sum_{i:u_i^*>0} u_i^* (n + \nu_i - n) \\ &= g^* - (u^*)^T \nu \\ &= g^* + (u - u^*)^T \nu \\ &\geq g^* - ||u - u^*|| ||\nu||, \end{aligned} \tag{3.4.3}$$

where we have used (3.4.2) and the fact that $x_i^T H^* x_i = n$ whenever u_i^* is positive. (Observe that these inequalities provide another proof of the weaker bound in Proposition 2.9. Indeed, $g(u) \geq g^* - \sum_{i:u_i^*>0} u_i^* (x_i^T H(u) x_i - n)$ holds for any feasible u, and if u is ϵ-primal feasible, then this is at least $g^* - \sum_{i:u_i^*>0} u_i^* n\epsilon = g^* - n\epsilon$.) We want to show that $g(u)$ is within $O(\delta^2)$ of g^*, and the inequality above offers strong evidence that this is true. Indeed, as we have seen, the perturbation ν is of order δ; so we only need to show that the corresponding change in the multipliers is also of order δ.

For this task we use a result of Robinson (Corollary 4.3 of [64]), which we state for our context as follows. We view the set of positive definite symmetric matrices $\mathcal{S}^n_{++}$ as an open subset of the set $\mathcal{S}^n$ of symmetric $n \times n$ matrices, and consider

$$(\hat{P}(\nu)) \qquad \begin{array}{ll} \min_{H \in \mathcal{S}^n} & \hat{f}(H, \nu) := -\text{lndet}(H) \\ & \hat{g}_i(H, \nu) := x_i^T H x_i - n - \nu_i \quad \leq \quad 0, \quad i = 1, \ldots, m, \end{array} \tag{3.4.4}$$

with $\hat{f}$ and $\hat{g}$ mapping $\mathcal{S}^n_{++} \times \mathrm{R}^m$ into R and R^m, respectively. Note that H^* is an optimal solution for $(\hat{P}(0))$. Then $\hat{f}$ and $\hat{g}$ are smooth, the constraints are regular (since they are convex and there is a Slater point, i.e., a feasible H satisfying the constraints strictly) at $\nu = 0$, and the constraint right-hand side, the nonpositive orthant in R^m, is polyhedral. Moreover, since the constraints are linear and the second derivative of the objective function at any feasible point is a positive definite operator, Robinson's second-order sufficient condition holds. Then Robinson's result yields the following.

Theorem 3.14. *There is a neighborhood N of the origin in R^m and a positive constant L such that, for any $\nu \in N$ and optimal solution H to $(\hat{P}(\nu))$ with corresponding multipliers u, there*

is a multiplier vector u^* for $(\hat{P}(0))$ with

$$||u - u^*|| \leq L||v||.$$

Now, if δ is sufficiently small, $v(u)$ will lie in this neighborhood N, and so combining this result with (3.4.3), we obtain the following.

Proposition 3.15. *There is a constant M such that, for every sufficiently small positive δ, any δ-approximately optimal u satisfies*

$$g^* - g(u) \leq M\delta^2. \tag{3.4.5}$$

We now repeat our analysis of the complexity of the WA Algorithm, but using (3.4.5) in place of (3.3.3). It will take a certain (data-dependent) constant P number of iterations to reduce $\bar{\delta}(u)$ to a level below 1 and below the level for which the proposition above applies. From then on, to reduce $\bar{\delta}(u)$ from δ to $\delta/2$ will require at most

$$\frac{M\delta^2}{\delta^2/14} = 14M$$

iterations (see the proof of Lemma 3.8). There are a logarithmic number of such phases to reduce $\bar{\delta}(u)$ from at most 1 to ϵ, but now we have a sum of constant terms instead of the sum of an increasing geometric series (see (3.3.6) in the proof of Theorem 3.9). Thus adding all these iterations gives a total of $28M \log_2(\epsilon^{-1})$, and we obtain

Theorem 3.16. *There are data-dependent constants P and Q such that the number of iterations of the WA Algorithm required to obtain an ϵ-approximately optimal u is at most*

$$P + Q \log_2(\epsilon^{-1}).$$

Besides showing local linear convergence, our analysis has shown that an ϵ-approximately optimal solution may be much closer to optimality in (D) than an ϵ-primal feasible solution: compare (3.4.5) to (3.3.3). Unfortunately, we don't know when we are very close to optimality, because the only way to certify this is by using duality, and the corresponding primal solution has an objective value $n \ln(1+\epsilon)$ higher.

3.5 ▪ Polarity and a striking relationship to the ellipsoid algorithm

In the previous two chapters we noted that every iteration of the ellipsoid algorithm requires us to obtain the minimum-volume ellipsoid containing part of a given ellipsoid, and hence involves a subproblem similar to (P). Now we investigate the relationship between the algorithms we have developed for (P) and the ellipsoid method and show the remarkable fact that the FW and WA Algorithms can be viewed as applying the deepest (or more accurately, least shallow) symmetric two-sided cut ellipsoid algorithm to the polar of $\mathbf{X} := \text{conv}\{\pm x_1, \ldots, \pm x_m\}$.

Given a closed convex set C in R^n containing the origin, its polar is defined to be

$$C^\circ := \{z \in \mathrm{R}^n : x^T z \leq 1 \text{ for all } x \in C\}. \tag{3.5.1}$$

It is easy to see that C° is also a closed convex set containing the origin; moreover, an easy application of the separating hyperplane theorem shows that polarity is an involution: $(C^\circ)^\circ = C$. For the set $\mathbf{X}$ above, clearly

$$\mathbf{Z} := \mathbf{X}^\circ = \{z \in \mathrm{R}^n : |x_i^T z| \le 1,\ i = 1, \ldots, m\}.$$

Moreover, an application of (1.1.6) shows that, if $\mathcal{F}$ is the ellipsoid

$$\mathcal{F} := \{x \in \mathrm{R}^n : x^T H x \le 1\}$$

for a positive definite matrix H, then its polar is the ellipsoid

$$\mathcal{F}^\circ = \{z \in \mathrm{R}^n : z^T H^{-1} z \le 1\}.$$

(When we deal with polarity, it is easier to use right-hand sides of 1 rather than n.) Finally, the definition immediately implies that

$$C \subseteq D \Rightarrow D^\circ \subseteq C^\circ.$$

Now suppose that we are applying the FW Algorithm to (D). At a particular iteration, we have a feasible u, and as noted at the end of Section 2.4, we have

$$\mathcal{F} := \{x \in \mathrm{R}^n : x^T H x \le 1\} \subseteq \mathbf{X}$$

for $H := H(u) = (XUX^T)^{-1}$. By the inclusion-reversing property of polarity,

$$\mathbf{Z} \subseteq \mathcal{F}^\circ =: \mathcal{E} = \{z \in \mathrm{R}^n : z^T H^{-1} z \le 1\}.$$

If the FW Algorithm does not terminate, it chooses a point x_i with $\omega_i := x_i^T H x_i \ge (1+\epsilon)n$, and of course $-x_i$ also satisfies this relation. The corresponding inequalities defining $\mathbf{Z}$ can be written as

$$-\beta (x_i^T H x_i)^{1/2} \le x_i^T z \le \beta (x_i^T H x_i)^{1/2} \tag{3.5.2}$$

with

$$\beta = \omega_i^{-1/2} \le [(1+\epsilon)n]^{-1/2}.$$

The set of points in $\mathcal{E}$ satisfying these inequalities is exactly $\mathcal{E}_{\alpha\beta}$ of (1.4.1) with $a := \omega_i^{-1/2} x_i$ and $\alpha := -\beta$.

We described the ellipsoid method in Section 1.4 as seeking a point satisfying a system of inequalities, but it can also be used to find a point "deep" inside a polyhedron. In this case, we might continue the algorithm even when a feasible point is obtained, by using inequalities that are "too close" to the current center. Of course, for a centrally symmetric polyhedron like $\mathbf{Z}$, clearly the origin is the deepest point, but we may still want an ellipsoid that fits the polyhedron well in the sense that no inequality is too close to the center, relative to its distance to the boundary of the ellipsoid. In our case, the inequalities (3.5.2) are too close to the center, because of the bound on β, and a smaller volume ellipsoid,

$$\mathcal{E}_+ := \{z \in \mathrm{R}^n : z^T H_+^{-1} z \le 1\},$$

can be found; indeed, according to the formulae in [80], the minimum-volume ellipsoid containing $\mathcal{E}_{\alpha\beta}$ has

$$H_+ = \delta \left(H - \sigma \frac{(Hx_i)(Hx_i)^T}{x_i^T H x_i} \right)$$

with

$$\delta := \frac{n(1-\beta^2)}{n-1}, \quad \sigma := \frac{1-n\beta^2}{1-\beta^2}.$$

Then the rank-one update formula in Corollary A.10 gives

$$\begin{aligned} H_+^{-1} &= \delta^{-1}\left(H^{-1} + \tfrac{\sigma}{(1-\sigma)x_i^T H x_i} x_i x_i^T\right) \\ &= \delta^{-1}\left(H^{-1} + \tfrac{\sigma\beta^2}{1-\sigma} x_i x_i^T\right) \\ &= \delta^{-1}(1+\mu)\left[\left(1 - \tfrac{\mu}{1+\mu}\right)H^{-1} + \tfrac{\mu}{1+\mu} x_i x_i^T\right], \end{aligned} \tag{3.5.3}$$

where

$$\mu := \frac{\sigma\beta^2}{1-\sigma}.$$

Now, substituting in the value for σ, we find $\mu = (1-n\beta^2)/(n-1)$, and since $\omega_i = \beta^{-2}$,

$$\mu = \frac{\omega_i - n}{(n-1)\omega_i} = \lambda^*,$$

which is the optimal stepsize in the FW Algorithm. Moreover, substituting in the value for δ we find $\delta^{-1}(1+\mu) = 1$, so that (3.5.3) gives

$$H_+^{-1} = (1+\lambda^*)^{-1}(H^{-1} + \lambda^* x_i x_i^T) = X U_+ X^T$$

with $u_+ = (1+\lambda^*)^{-1}(u + \lambda^* e_i)$. It follows that the updated ellipsoid in the deepest symmetric cut ellipsoid method is exactly the polar of the updated inscribed ellipsoid in the FW Algorithm! This establishes the desired correspondence. If the FW Algorithm terminates when u is ϵ-primal feasible, the corresponding ellipsoid algorithm terminates when all the inequalities defining $\mathbf{Y}$ have corresponding β values greater than $[(1+\epsilon)n]^{1/2}$.

The correspondence with the WA Algorithm is not so clear cut. A negative λ^* corresponds to a ω_i that is less than n. In the polar space, this yields a pair of hyperplanes that are far from the center. If the weight on these is positive, an improvement in the volume can be made by decreasing this weight. This is not a feature of the usual ellipsoid algorithm, although variants do consider such steps.

Given that "everyone knows" that the ellipsoid algorithm is very inefficient, this relationship would seem to imply that our coordinate-ascent methods would be equally inefficient. However, while the usual one-sided cut ellipsoid method is indeed slow, typically leading to a reduction in volume by a factor of the order of $1-1/n$, the same is not true of its two-sided variant. Indeed, the factor of volume reduction is equal to that of $(\det H)^{1/2}$, and Lemma 3.5 implies that this is a constant less than 1 while ϵ is bounded away from zero. This also follows from the formulae in [80].

3.6 ▪ Small core sets and eliminating points

Recall that a core set of $\mathbf{X}$ is a subset of $\{x_1, \ldots, x_m\}$ such that the minimum-volume ellipsoid containing this subset is also the minimum-volume ellipsoid containing $\mathbf{X}$. We showed just before Example 2.3 that (for centered ellipsoids) a core set of cardinality at most $n(n+1)/2$ exists.

Suppose that u is ϵ-primal feasible. Then $g(u)$ provides a lower bound on $-\ln\det(H^*)$ and $\mathcal{E}([1+\epsilon]^{-1}H(u))$ contains $\mathbf{X}$. Then if $S := \{x_i : u_i > 0\}$, $g(u)$ is also similarly related

to the minimum-volume ellipsoid containing S. The volume of the latter is thus within a factor $(1+\epsilon)^n$ of the minimum-volume ellipsoid containing $\mathbf{X}$. We call S an *ϵ-core set*.

Now consider the FW or the WA Algorithm, starting with $u = u_{KY}$. The initial u has at most $2n$ positive components, and every iteration (respectively, every add, increase, or decrease iteration) increases the number of positive components by at most 1. Hence, when the algorithm terminates with an ϵ-primal feasible u (respectively, an ϵ-approximately optimal u), this has at most $4n(\ln\ln n + 4) + 28n/\epsilon$ positive components.

Proposition 3.17. *When initialized with $u = u_{KY}$, both the FW and the WA Algorithms generate an ϵ-core set of cardinality at most $4n(\ln\ln n + 4) + 28n/\epsilon$.*

Although the bound is the same for both algorithms, note that the WA Algorithm has the potential for producing much smaller core sets, since each drop iteration decreases the number of positive components of u.

Next we turn to trying to decrease the complexity of each iteration. Recall that this is dominated by the work to update ω, which requires $O(mn)$ floating-point operations. This is large because m can be much larger than n. However, we know from the discussion above that a much smaller set of x_i's is important for the algorithm. We would therefore like to eliminate certain x_i's from consideration during the algorithms, when we are sure that they are not relevant. Of course, if a drop iteration makes u_i zero, or u_i is very small, we may suspect that x_i can be eliminated; however, we would like a guaranteed test, since if we eliminate a point x_i and later want to check that $x_i^T H x_i \le n$, we need to do $O(n^2)$ work, whereas just updating ω_i requires only $O(n)$ work at each iteration. We will call a point x_i *inessential* if it lies in the interior of the minimum-volume ellipsoid, and *essential* otherwise. Clearly inessential points can be eliminated.

Suppose we have a feasible u with $H := H(u)$. Also, let H^* and u^* denote optimal solutions to (P) and (D), respectively. Then any x_i with $x_i^T H^* x_i < n$ can be eliminated; unfortunately, we don't know H^*. However, if $\delta := \delta(u)$ is small, we may suspect that H^* is close to H, and thus perhaps we can eliminate points with $\omega_i(u) = x_i^T H x_i$ sufficiently less than n.

To measure the distance between these two matrices, we define

$$M := H^{1/2}(H^*)^{-1}H^{1/2},$$

which we hope is close to the identity. Indeed, let its eigenvalues be $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Then we have

$$\sum_j \lambda_j = \mathrm{Trace}(M) = H \bullet \left(\sum_i u_i^* x_i x_i^T\right) = \sum_i u_i^* x_i^T H x_i \le (1+\delta)n. \tag{3.6.1}$$

Similarly,

$$\sum_j \lambda_j^{-1} = \mathrm{Trace}(M^{-1}) = H^* \bullet \left(\sum_i u_i x_i x_i^T\right) = \sum_i u_i x_i^T H^* x_i \le n. \tag{3.6.2}$$

These two inequalities imply that the eigenvalues must all be close to 1 if δ is small, and hence H and H^* must be close.

To see how we will use this, suppose x_i is essential, so that $x_i^T H^* x_i = n$. Now let $z := M^{-1/2}H^{1/2}x_i$, which implies $z^T z = x_i^T H^* x_i = n$. Also,

$$\omega_i(u) = x_i^T H x_i = z^T M z \ge \lambda_1 z^T z = \lambda_1 n. \tag{3.6.3}$$

Thus we will be able to eliminate any x_b with $\omega_b(u) < \lambda_1 n$. Hence we would like a lower bound on λ_1. Consider the optimization problem

$$\min\{\lambda_1 : \lambda > 0,\ \lambda \text{ satisfies (3.6.1) and (3.6.2)}\}.$$

It is clear that the feasible region of this problem is compact, since each λ_j must lie between $1/n$ and $(1+\delta)n$, so that the problem has an optimal solution. Moreover, any optimal solution must satisfy both constraints at equality. If the second constraint is satisfied strictly, we may decrease λ_1 slightly and obtain a better feasible solution. If the first constraint is satisfied strictly, we may increase λ_2 slightly while keeping the solution feasible, and then the second constraint is satisfied strictly and we may proceed as above. Moreover, we must have $\lambda_2 = \cdots = \lambda_n$; otherwise replacing these components by their arithmetic mean keeps the first constraint satisfied and makes the second constraint strict. So the optimal solution is of the form $(\lambda; \mu; \cdots; \mu)$, where

$$\lambda + (n-1)\mu = (1+\delta)n, \quad \frac{1}{\lambda} + \frac{n-1}{\mu} = n.$$

Solving the first equation for μ in terms of λ and substituting the result in the second equation yields a quadratic equation with roots

$$\lambda = 1 + \frac{\delta n}{2} \pm \sqrt{\delta n - \delta + \frac{\delta^2 n^2}{4}},$$

and since we want $\lambda \le \mu$, the negative sign is the appropriate one. Combining this result with our earlier discussion, we obtain the following.

Proposition 3.18. *Given a feasible solution u with $\delta := \delta(u)$, any point x_i with*

$$\omega_i(u) < n\left(1 + \frac{\delta n}{2} - \sqrt{\delta n - \delta + \frac{\delta^2 n^2}{4}}\right)$$

is inessential.

3.7 ▪ A connection to spectral sparsification of graphs

In this section we show how the FW and WA Algorithms "almost" provide a constructive approach to the problem of spectral sparsification of graphs. This highlights the need to improve our knowledge of the convergence properties of these algorithms.

Suppose we are given a weighted graph $G = (V, E, w)$ with $w : E \to \mathrm{R}_{++}$. Without loss of generality we assume $V = \{1, 2, \ldots, n\}$. The graph Laplacian is the symmetric matrix $L(G)$ of order n with ij entry $-w_{ij}$ if $ij \in E$, 0 for $i \neq j$ otherwise, and ii entry $\sum_{j:ij\in E} w_{ij}$. We seek a (weighted) subgraph $\hat{G} = (V, \hat{E}, \hat{w})$ with $\hat{w} : \hat{E} \to \mathrm{R}_{++}$, with $|\hat{E}|$ "small," such that

$$(1-\delta)L(G) \preceq L(\hat{G}) \preceq (1+\delta)L(G)$$

for some $0 < \delta < 1$. We call this a spectral sparsification of the graph G. This notion was introduced by Spielman and Teng [76], who described its applications. Since

$$z^T L(G) z = \sum_{ij\in E} w_{ij}(z_i - z_j)^2,$$

such a subgraph guarantees that all cuts $\hat{w}(S,\bar{S})$ in $\hat{G}$ are close to the corresponding cuts $w(S,\bar{S})$ in G, by choosing z to be the indicator vector of $S \subseteq V$. (A cut $\hat{w}(S,\bar{S})$, where $\bar{S} = V \setminus S$, is defined to be the sum of all $\hat{w}_{ij}$ for $i \in S, j \in \bar{S}$.) A subgraph approximately preserving the size of cuts in this way was earlier considered by Benczur and Karger [12].

Without loss of generality, G is connected (else just consider separately its connected components), in which case $L(G)$ is positive semidefinite with rank $n-1$, its nullspace being the space spanned by the vector e of 1's. To reduce to the positive definite case, we can either project the Laplacians to the space orthogonal to e, or replace $L(G)$ by $L(G)+\lambda ee^T$ for $\lambda > 0$; we can choose $\lambda := \mathrm{Tr}(L(G))/(n(n-1))$ to make its eigenvalues of comparable size. Note that the matrix inequalities above hold if

$$(1-\delta)(L(G)+\lambda ee^T) \preceq L(\hat{G}) + \hat{\lambda} ee^T \preceq (1+\delta)(L(G)+\lambda ee^T)$$

for some positive λ and $\hat{\lambda}$.

Following Batson, Spielman, and Srivastava [11], we now reduce this to a linear algebra problem. Note that

$$L(G) = \sum_{ij \in E} w_{ij}(e_i - e_j)(e_i - e_j)^T$$

with e_i the ith unit vector. Hence

$$L(G) + \lambda ee^T = YWY^T,$$

where Y is a matrix with columns e and $e_i - e_j$, $ij \in E$, and W is a diagonal matrix with diagonal entries λ and the w_{ij}'s. Since $L(G)+\lambda ee^T$ is positive definite, it can be written as JJ^T with J a nonsingular $n \times n$ matrix, so that

$$I = (J^{-1}Y)W(J^{-1}Y)^T.$$

We now scale the columns of $J^{-1}Y$ to get X, with all the columns of X having norm $\sqrt{n}$. If we correspondingly scale the diagonal entries of W, we get

$$I = X\bar{U}X^T,$$

where we let $\bar{u} := \mathrm{diag}(\bar{U})$. It can now be seen that our goal will be achieved if we find a nonnegative u with a "small" number of positive components so that

$$(1-\delta)I \preceq XUX^T \preceq (1+\delta)I$$

with $U = \mathrm{Diag}(u)$. The nonzero entries of u pick out the edges of $\hat{E}$, and rescaling these entries gives the weights $\hat{w}$. We next relate this problem to the MVEE problem and its dual.

In order to find a suitable u, we restrict the arithmetic mean of the eigenvalues of XUX^T to 1 and then maximize their geometric mean. Note that for $u = \bar{u}$, the geometric mean is also 1 and hence maximum. Observe that the arithmetic mean is 1 iff the trace is n, which holds iff

$$\sum_i u_i = \mathrm{Tr}(UX^TX)/n = \mathrm{Tr}(XUX^T)/n = 1,$$

since the diagonal entries of X^TX are all n by our scaling. Also, maximizing the geometric mean amounts to maximizing the determinant, the product of the eigenvalues, or its

logarithm. Hence the optimization problem becomes exactly the dual (D) of the MVEE problem!

Just as we know an optimal solution $\bar{u}$ of (D), we know one of (P): indeed, $H = I$ is feasible by our scaling, and since it leads to no duality gap with $u = \bar{u}$, it is optimal. However, we are not after an optimal solution of (D): we want a near-optimal u with a small number of positive components. But this is exactly what our coordinate-ascent algorithms achieve, as we saw in the previous section.

By Theorems 3.9 and 3.11, we then take at most

$$O(n(\ln \ln n + \epsilon^{-1}))$$

iterations to obtain an ϵ-approximately optimal solution, and then u has the same order of positive components.

What value of ϵ is necessary to achieve our goal? This is where the argument unfortunately falls apart. The worst case for a matrix "near the identity" is to have one eigenvalue of $1-\delta$ and $n-1$ of $1+\delta/(n-1)$, which has an objective value of about $-\delta^2/2$ (or a similar objective from the opposite sign configuration). So we need to set ϵ to about $\delta^2/(2n)$, and this extra n factor means we need $O(n^2)$ iterations, and hence we obtain a dense graph!

Batson, Spielman, and Srivastava [11] give an algorithm that yields a subgraph with at most $O(n/\delta^2)$ edges. This method is also based on rank-one updates, but each iteration requires an order of magnitude more work than our method. One should always learn from one's failures: how could we obtain a comparable result from our approach? If we could obtain a global linear convergence rate, with a complexity of the order of $O(n \ln \ln n + n \ln(1/\epsilon))$ iterations, we would be able to improve this result in terms of the dependence on δ while worsening it in its dependence on n. There may also be room for improvement in our analysis of the accuracy required. We want all eigenvalues of XUX^T to be within δ of 1. Our algorithm ensures that all quantities $x_i^T(XUX^T)^{-1}x_i$ are at most $(1+\epsilon)n$. However, to relate these two, we seem to need to go through the intermediate step of bounding the lack of optimality in the log determinant objective function, and for this our tolerance needs to be very tight. While our approach fails, it still seems constructive to show the power of an optimization algorithm to potentially establish a theoretical result, even when it is applied to a problem with a known optimal solution.

3.8 ▪ Computational results

Our previous examples have been toy problems in two or three dimensions. Here we provide an indication of the power of the algorithms discussed in this chapter to solve large-scale instances.

Extensive computational testing has been carried out by Sun and Freund [77] on minimum-volume ellipsoid problems using various interior-point-type methods applied to (P) as well as the FW Algorithm. However, the latter was implemented without exploiting rank-one updates. They found that the fastest method was the dual reduced Newton (DRN) method, applied together with an active-set strategy for large m.

Sun and Freund generate test sets by combining clusters of points, where each cluster consists of independent points from a Gaussian distribution (each cluster has a different mean and covariance matrix). The problem sizes they consider range up to 30,000 points in dimension 30.

Ahipaşaoğlu, Sun, and Todd [3] implemented the FW and WA Algorithms and compared them to the DRN method on the same data sets. While the FW Algorithm was exceedingly slow, it was able to obtain moderately accurate solutions (ϵ-primal feasible solutions with ϵ of the order 10^{-2} or 10^{-3}). The WA Algorithm achieved much more accurate solutions, ϵ-approximately optimal with $\epsilon = 10^{-7}$ or even 10^{-10}. It was much faster than the DRN method without an active set strategy, but somewhat slower than the latter with an active set strategy. These comparisons were on moderately sized problems, up to 1,800 points in dimension 30. For larger problems, the DRN active method was considerably superior, by a factor of almost 7 for 30,000 points in dimension 30. The Kumar–Yıldırım initialization was important for these results. However, no elimination of points was used in these tests. Moreover, it was demonstrated that the DRN active method could not be applied to truly large-scale problems (of the order of 500,000 points in dimension 500) because of memory problems, while the WA Algorithm had no such difficulty.

In her doctoral thesis, Ahipaşaoğlu [2] used the strategy of Section 3.6 to eliminate points. A considerable speedup ensued. For ten random problems involving 30,000 points in dimension 50, a speedup of over 4 was observed, while with 500,000 points, the speedup was over 7.

Here we will just give results for one randomly generated problem using the variants of coordinate-ascent algorithms discussed in this chapter. It is not easy to generate appropriate test sets. One way of doing so is to take the union of various clusters, as above, which seems reasonable from an applications point of view. If the points are taken from a single distribution, difficulties arise. For example, if all coordinates are generated independently from a standard Gaussian distribution, then the law of large numbers implies that all points will lie very close to a spherical surface—clearly not a typical situation. The same holds if all coordinates are generated independently from any distribution with finite variance. If the points are generated from a general Gaussian distribution, they will similarly all lie close to the surface of an ellipsoid, also a very special situation. Instead, we generate them as follows. First, we use the Cauchy distribution which has heavy tails. (A centered Cauchy random variable can be generated as a/b, where a and b are independent standard Gaussian random variables.) If we just generated an $n \times m$ matrix of independent Cauchy random variables, each point would likely have a dominant component due to the heavy tails, and so there would be only $2n$ "interesting" points, each near a coordinate axis. Instead we generate m independent Cauchy random variables and an $n \times m$ matrix A of independent standard Gaussian random variables, and then set each point x_i to be the normalized ith column of A times the ith Cauchy sample. These points satisfy rotational symmetry, and their distances from the origin are Cauchy. (An implementation of this method in MATLAB is included in Appendix B as `rot_cauchy.m`.)

We generated 5,000 points in dimension 200 as the columns of a matrix X generated as above. Our basic method is the WA Algorithm with Kumar–Yıldırım initialization and with elimination of points (every $\max\{n, 100\}$ iterations) as in Section 3.6. To obtain an ϵ-approximately optimal solution with $\epsilon = 10^{-7}$ required 1,514 iterations and 1.2 seconds. (All runs were made on a MacBook Air with a 1.7 GHz Intel Core i7 with MATLAB 2014a.) Of these iterations, none were drop, 746 were decrease, 106 were add, and 662 were increase iterations. At the initial iteration, 859 of the 5,000 points were eliminated, and after 200 iterations, a further 2,752 were eliminated. The final solution had just 306 positive components of u.

Figures 3.3 and 3.4 show the progress of $\max \omega_i$ and $\min\{\omega_j : u_j > 0\}$ after the first 25 iterations, and the linear convergence of the error $\epsilon := \max(\max(\omega_i - n)/n, \max\{(n -$

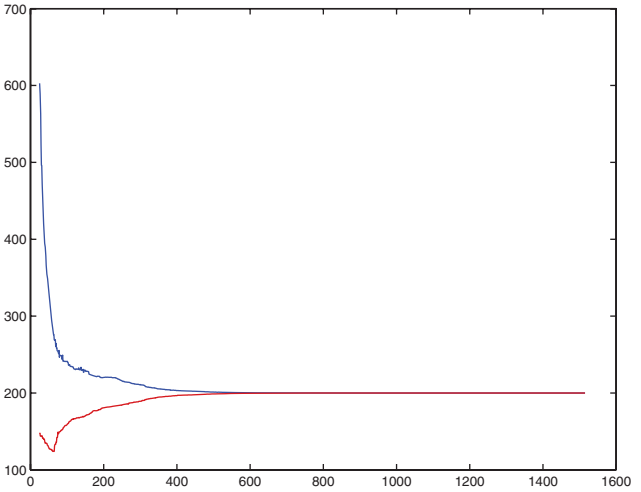


Figure 3.3. *Convergence of* $\max \omega_i$ *(blue) and* $\min\{\omega_j : u_j > 0\}$ *(red).*

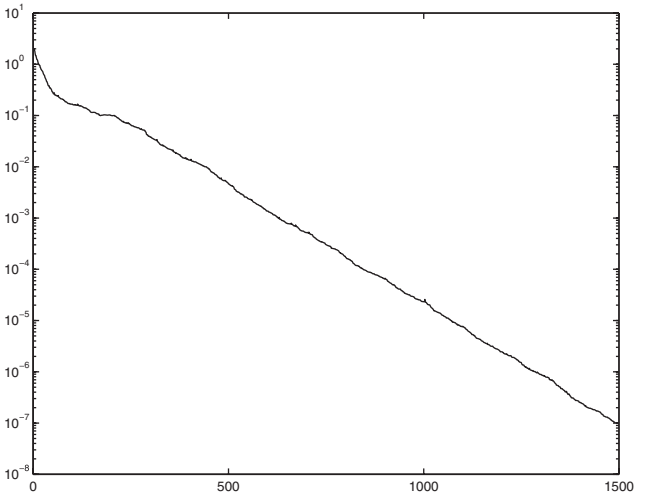


Figure 3.4. *Linear convergence of the error.*

$\omega_j)/n : u_j > 0\}$), plotted on a log scale. Note that the linear convergence takes hold almost from the first iteration, and that reasonably accurate solutions are available even after just 500 iterations.

Now let us make some variations on the algorithm and note the effects. First, if we ask for $\epsilon = 10^{-10}$, we need only 2,196 iterations and 1.4 seconds, a modest increase in effort. If we use the Khachiyan initialization, we need 6,451 iterations (4,694 drop, 652 decrease, no add, and 1,105 increase) and 5.8 seconds; no elimination of points takes place until iteration 5400, when 4,370 points are eliminated. If we disable the elimination of points in the basic algorithm, we need the same 1,514 iterations, but now the time required is 2.2 seconds, twice as much.

Lastly, we applied the FW Algorithm (no away steps) with the Khachiyan initialization. For this, we decreased the accuracy required. To obtain an ϵ-primal feasible solution

with $\epsilon = 10^{-2}$ required 19,494 iterations and 23.7 seconds. For $\epsilon = 10^{-3}$, these increased to 188,738 iterations and 229.2 seconds. Note the ten-fold increase for a ten-fold decrease in ϵ, corresponding to the complexity bound in Theorem 3.9, whereas the WA Algorithm needed only about a 20% increase to improve by three orders of magnitude, due to the linear convergence.

If we used the Kumar–Yıldırım initialization with the FW Algorithm, there was a noticeable improvement. For $\epsilon = 10^{-2}$ the method needed 2,353 iterations and 1.5 seconds, while $\epsilon = 10^{-3}$ required 35,153 iterations and 13.9 seconds. (However, for these accuracies, the WA Algorithm only needed 0.6 and 0.8 seconds, respectively.)

Let us stress again that, in this example, linear convergence of the WA Algorithm seems to occur from the very first iteration, as opposed to asymptotically. Indeed, we have observed this behavior over many examples. It would be very desirable to prove this rigorously, or more precisely prove a *global* convergence estimate of $O(p(m,n)\ln(1/\epsilon))$ steps to obtain an ϵ-approximately optimal solution for some polynomial p.

3.9 ▪ Notes and references

The FW Algorithm was proposed to solve the D-optimal design problem of statistics by Fedorov [27] with optimal stepsize as described here. Wynn [85] independently proposed a variant: if the initial u was chosen as $u_i = 1/\ell$, $i \in S$, $u_i = 0$ otherwise, for some subset S of $\{1,\ldots,m\}$ of cardinality ℓ, then the stepsize at the pth iteration was chosen to be $(\ell + p)^{-1}$, $p = 1,\ldots$, so that the resulting u was rational with all denominators equal to $\ell + p$; it then corresponded to a design with this number of design points. However, convergence was slow because of the deterministic stepsize. (As so often happened in this era, when an idea was ripe for discovery, it arose independently and simultaneously, in this case around 1970, on both sides of the Iron Curtain.)

Frank and Wolfe [30] proposed their method for quadratic programming problems in 1958, but also proposed an extension to arbitrary smooth convex linearly constrained problems. Both algorithms are now known as the Frank–Wolfe method; other names include the conditional gradient method. The idea of linearizing the objective function and solving the corresponding linear programming subproblem is natural, but since the resulting solution always lies at an extreme point, whereas the original problem may not share this property, convergence can be slow. Wolfe recognized this drawback and suggested the remedy of away steps in his 1970 paper [84], which also contained an analysis of the resulting convergence. Atwood [9] independently obtained the same method in the context of the D-optimal design problem three years later. Böhning [15] suggested an algorithm where at each iteration two components of u are chosen and a quantity is subtracted from one and added to the other. No rescaling is needed, but a rank-two update is required and it is not clear this is superior to an add/increase followed by a drop/decrease iteration, which is the same work. Barnes [10] proposed a coordinate-ascent method to solve the dual of the minimum-volume fixed-center ellipsoid problem as part of an approach to solving the general problem, motivated by Rosen's pattern recognition problem [66]. His nested approach did not recognize that the general case can be reduced to the centered one.

Khachiyan's initialization appears in his analysis [49] of the complexity of the FW Algorithm. Kumar and Yıldırım [56] developed their initialization in the context of searching for small core sets, but still using the FW Algorithm. Proposition 3.3 was proved by Betke and Henk [13] in connection with approximating the volume of convex bodies. The global complexity bound for the FW Algorithm is due to Khachiyan for his initialization and to Kumar and Yıldırım for theirs, while Todd and Yıldırım [81]

extended the analysis for the WA Algorithm. (Our analysis above of the number of iterations to decrease δ below 1 gives a slight improvement over the bounds in the latter two papers, replacing $\ln n$ (or $\ln m$) with $\ln \ln n$ (or $\ln \ln m$).) Local linear convergence for the WA Algorithm was established by Ahipaşaoğlu, Sun, and Todd [3], using the theory of perturbed nonlinear programming problems developed by Robinson [64]. Note that local linear convergence for the Frank–Wolfe method with Wolfe's away steps was already discussed in Wolfe [84] and then established in general by Guélat and Marcotte [38]. However, they assumed that the objective function, g in our notation, was continuously differentiable on the entire feasible region and strictly concave; neither of these conditions holds in our problem.

The relationship between these methods and the ellipsoid method was obtained by Todd and Yıldırım [81]. The formulae for the minimum-volume ellipsoid containing the intersection of an ellipsoid with a slab can be found in Todd [80], whereas the idea that the quadratic inequality defining each successive ellipsoid could be obtained by taking a linear combination of that defining the previous ellipsoid and that defining the slab was developed in Burrell and Todd [18]. That paper also describes situations in which steps like drop iterations could be useful, and while decrease or drop iterations were not explicitly suggested, I used them at that time in computational testing, in particular to remove the effects of initial very large bounds in the initial ellipsoid.

The bounds on the size of the ϵ-core sets generated by the FW and the WA Algorithms are due to Kumar and Yıldırım and Todd and Yıldırım, respectively (again, our analysis here gives a slight improvement). The test for eliminating points based on their ω-values is due to Harman and Pronzato [41].

There has been much recent interest in the Frank–Wolfe method and its variants due in part to their suitability for large-scale problems arising in statistical learning. See, for instance, Freund and Grigas [31], Jaggi [44], Lacoste-Julien and Jaggi [57], and Peña and Rodriguez [61].

Chapter 4
Minimum-Area Ellipsoidal Cylinders

We now turn to a generalization of the MVEE problem, where we seek an ellipsoidal cylinder containing a given set of points in R^n, whose intersection with the coordinate subspace corresponding to the first k components ($k \leq n$) has minimal area. (We call this k-dimensional measure area instead of volume to stress that it is lower-dimensional.) This is the minimum-area enclosing cylinder (MAEC) problem. Once again, there is a motivation in terms of an optimal design problem in statistics. In addition, the problem seems to have intrinsic interest. Finally, if we are trying to foresee possible collisions between objects moving in space, solving the MAEC problem for a collection of points in five-dimensional space can provide a guarantee of no collisions.

An illustration of the problem for the case $n = 3$, $k = 2$ is shown in Figure 4.1.

We formulate the problem in two ways in the next section. Section 4.2 derives the dual problem and shows weak and strong duality results. Optimality conditions are given in Section 4.3. In Section 4.4 we discuss D_k-optimal design, while Section 4.5 treats the collision problem.

4.1 ▪ Formulations of the MAEC problem

In Chapter 2 we assumed that the m points x_i which we wish to enclose spanned R^n, and that this was without loss of generality, since otherwise arbitrarily small ellipsoids circumscribed $\mathbf{X}$. This is not the case for ellipsoidal cylinders and cross-sectional areas, but the subspace S spanned by the x_i's must contain the subspace $\mathrm{R}^k \times \{0\}$; otherwise, ellipsoidal cylinders of arbitrarily small cross-sectional area can be found containing all the points by modifying the minimum-volume ellipsoid in S that contains all the points.

Now let us consider the case where S is a proper subspace of R^n. Then we can take a basis of S whose first k members are the first k coordinate vectors, with the remaining basis vectors chosen arbitrarily. We can then represent the points x_i in this basis, and we will obtain a new equivalent instance of the problem with a smaller n. Moreover, in this reformulation, the points span the full space. Following this discussion, we can assume again, without loss of generality, that

$$X \text{ has full row rank.} \tag{4.1.1}$$

Here X as before is the $n \times m$ matrix whose columns are the points x_i. However, it is convenient to partition this matrix into its first k rows, forming the matrix $Y \in \mathrm{R}^{k \times m}$,

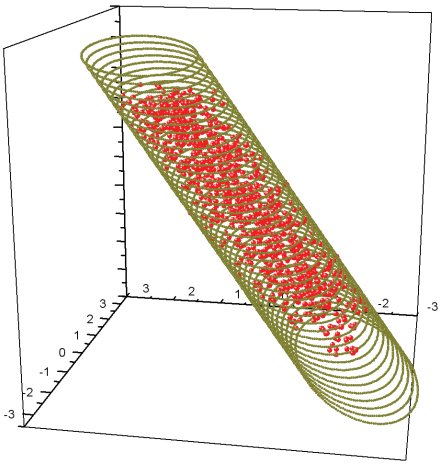


Figure 4.1. *Minimum-area ellipsoidal cylinder. Reprinted with permission from Elsevier* [4].

and the remaining rows, forming $Z \in \mathrm{R}^{\ell \times m}$, where $\ell := n - k$. Similarly, each x_i is partitioned into $(y_i; z_i)$, and any $x \in \mathrm{R}^n$ is partitioned into $(y; z)$.

Let $E \in \mathrm{R}^{k \times \ell}$ be given, and consider the columns of

$$\hat{E} := \begin{bmatrix} -E \\ I \end{bmatrix}$$

as the "axes" of a cylinder with its base lying in the subspace $\mathrm{R}^k \times \{0\}$. Any point $(y; z)$ can be obtained from the base point $(y + Ez; 0)$ by adding a linear combination of the columns of $\hat{E}$. If we restrict the base point to an ellipsoid in $\mathrm{R}^k \times \{0\}$, we obtain an ellipsoidal cylinder. We call E the "axis matrix."

Definition 4.1. *Let E be a matrix in $\mathrm{R}^{k \times \ell}$ and $H' \in \mathrm{R}^{k \times k}$ be positive definite. Then the set*

$$\mathcal{C}(E, H') := \{(y; z) \in \mathrm{R}^n : (y + Ez)^T H'(y + Ez) \leq k\}$$

is called the ellipsoidal cylinder *defined by the axis matrix E and the shape matrix H'.*

Note that if $k = n$ and thus E is an empty matrix, we obtain the ellipsoid $\mathcal{E}(H')$.

More properly, we might have called $\mathcal{C}(E, H')$ a *centered* ellipsoidal cylinder, but the restriction to the centered case is much more straightforward here than in the case of ellipsoids. Indeed, suppose we define a noncentered ellipsoidal cylinder as above, but with a new parameter $\bar{y} \in \mathrm{R}^k$, its center, and $y + Ez$ in the definition replaced by $y + Ez - \bar{y}$. Then it is easy to see that the minimum-area noncentered ellipsoidal cylinder in R^n containing the x_i's can be trivially obtained from the minimum-area centered ellipsoidal cylinder in R^{n+1} containing the points $(x_i; 1)$ with the same value of k. The negative of the last column of the axis matrix $E \in \mathrm{R}^{k \times (n+1-k)}$ gives the center of the general ellipsoidal cylinder in R^n. We henceforth only consider centered ellipsoidal cylinders.

As a special case of this reduction, we can solve a noncentered minimum-volume ellipsoid problem in R^n as a minimum-area centered ellipsoidal cylinder problem in R^{n+1}. However, ellipsoidal cylinder problems are harder to solve than comparably sized ellipsoid problems; so the more complicated reduction in Chapter 2 is preferable computationally.

With this definition of a (centered) ellipsoidal cylinder, we can formulate the MAEC problem as

$$(P') \qquad \begin{array}{ll} \min_{E\in\mathbf{R}^{k\times\ell},H'\in\mathcal{S}^k} & \bar{f}(E,H') := -\ln\det(H') \\ & (y_i+Ez_i)^T H'(y_i+Ez_i) \le k, \quad i=1,\dots,m. \end{array} \tag{4.1.2}$$

Unfortunately, this problem is nonconvex in its variables E and H', due to the cross-terms $2y_i^T H' E z_i$ and $z_i^T E^T H' E z_i$ in the constraints. We will see in the "Notes and references" section that the above problem reduces to a linear programming problem if $k=1$. We therefore assume whenever it is useful that $1<k<n$, although we sometimes illustrate ideas with simple examples where $k=1$.

However, a simple reformulation restores convexity. We encode both E and H' in a positive semidefinite matrix $H\in\mathcal{S}^n$. This matrix is partitioned into its first k rows and columns and its last ℓ rows and columns:

$$H =: \begin{bmatrix} H_{YY} & H_{YZ} \\ H_{YZ}^T & H_{ZZ} \end{bmatrix}.$$

Our second formulation is then

$$(P) \qquad \begin{array}{ll} \min_{H\in\mathcal{S}^n} & f(H) := -\ln\det(H_{YY}) \\ & x_i^T H x_i \le k, \quad i=1,\dots,m, \\ & H \succeq 0. \end{array} \tag{4.1.3}$$

Note that we are using f for the objective function again, but it differs from that in the previous two chapters. Also, a finite objective function implies that H_{YY} is positive definite, but it does not imply (unless $k=n$) that the full matrix H is positive semidefinite, and so we need to add this requirement as an explicit constraint. (P) *is* a convex programming problem, with m linear inequality constraints and one positive semidefiniteness constraint on the variable H. We say that H is feasible in (P) if it satisfies the constraints and has finite objective value, i.e., H_{YY} is positive definite.

Lemma 4.2. *Problems* (P) *and* (P') *are equivalent.*

Proof. First assume that H is any feasible solution to (P). Then H_{YY} is positive definite, so that by Theorem A.11, H is positive semidefinite iff $H_{ZZ}\succeq H_{YZ}^T H_{YY}^{-1} H_{YZ}$. It follows that we may assume that $H_{ZZ}=H_{YZ}^T H_{YY}^{-1} H_{YZ}$ without loss of generality, since replacing H_{ZZ} by the right-hand side will maintain feasibility and keep the same objective value. If we define $E := H_{YY}^{-1} H_{YZ}$, we then have

$$H = \begin{bmatrix} H_{YY} & H_{YY}E \\ E^T H_{YY} & E^T H_{YY} E \end{bmatrix},$$

so that $x_i^T H x_i = (y_i+Ez_i)^T H_{YY}(y_i+Ez_i)$ for all i. It follows that $(E,H') := (E,H_{YY})$ is feasible in (P') with the same objective value.

Conversely, if (E,H') is feasible in (P'), it is easily seen that

$$H := \begin{bmatrix} H' & H'E \\ E^T H' & E^T H' E \end{bmatrix}$$

is feasible in (P) with the same objective value. ∎

4.2 ▪ Duality for the MAEC problem

Now that we have a convex formulation, we can derive the dual of (P). It is helpful to rewrite this problem as follows:

$$(P)\qquad \begin{array}{ll} \min_{J\in\mathcal{S}^k,H\in\mathcal{S}^n} & -\operatorname{lndet}(J) \\ J & - \quad H_{YY}=0, \\ & \quad x_i^T H x_i \le k, \quad i=1,\dots,m, \\ & \quad H\succeq 0. \end{array}$$

We associate a symmetric matrix multiplier $K\in\mathcal{S}^k$ with the first constraint and a nonnegative multiplier u_i with the ith constraint of the second set to obtain the Lagrangian

$$L(J,H,K,u) := -\operatorname{lndet}(J)+K\bullet J-K\bullet H_{YY}+H\bullet(XUX^T)-ke^Tu, \tag{4.2.1}$$

where U as before is $\operatorname{Diag}(u)$, defined for $J\in\mathcal{S}^k$, $H\in\mathcal{S}^n$, $K\in\mathcal{S}^k$, and $u\in\mathrm{R}^m$. For a given symmetric K and nonnegative u, we would like to minimize this with respect to positive definite J and positive semidefinite H's. Let us introduce the notation: for $K\in\mathcal{S}^k$,

$$\bar{K} := \begin{bmatrix} K & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{S}^n.$$

Then the Lagrangian can be written as the sum of two terms:

$$[-\operatorname{lndet}(J)+K\bullet J]+[H\bullet(XUX^T-\bar{K})-ke^Tu].$$

By the same argument as that above (2.1.4), we see that the infimum of the first term with respect to J is $-\infty$ unless K is positive definite, in which case the infimum is attained by $J=K^{-1}$ and equals $\operatorname{lndet}(K)+k$. The infimum of the second term with respect to H is $-\infty$ unless $XUX^T-\bar{K}$ is positive semidefinite, since otherwise we can choose H of the form λvv^T, where v is an eigenvector corresponding to a negative eigenvalue of $XUX^T-\bar{K}$, and let $\lambda\to\infty$. If this matrix is positive semidefinite, the infimum is attained by $H=0$ and equals $-ke^Tu$. Hence

$$\min_{J,H} L(J,H,K,u) = \operatorname{lndet}(K)+k-ke^Tu$$

as long as $XUX^T-\bar{K}$ is positive semidefinite, and $-\infty$ otherwise. Hence the Lagrangian dual $\max_{K,u\ge 0}\{\min_{J,H\succeq 0} H(J,H,K,u)\}$ can be written as

$$(\tilde{D})\quad \max_{K\in\mathcal{S}^k,u\in\mathbf{R}^m}\{\operatorname{lndet}(K)+k-ke^Tu : XUX^T-\bar{K}\succeq 0,\ u\ge 0\}.$$

Since K can be scaled proportionally with u, an argument identical to that below (2.1.4) shows that we can assume without loss of generality that $e^Tu=1$. We are thus led to the dual problem

$$(D)\qquad \begin{array}{ll} \max_{u\in\mathbf{R}^m,K\in\mathcal{S}^k} & g(u,K) := \operatorname{lndet}(K) \\ & XUX^T-\bar{K}\succeq 0, \\ & e^Tu = 1, \\ & u\ge 0. \end{array} \tag{4.2.2}$$

Note that we are again reusing the notation g for the objective function of the dual problem. We say (u,K) is feasible for (D) if it satisfies the constraints and has finite

objective value, i.e., K is positive definite. Clearly (D) is a convex programming problem, because the objective function (to be maximized) is concave, and the constraints are positive semidefiniteness, equality, or nonnegativity constraints on linear functions of the variables.

We can also express the dual problem solely in terms of u at the expense of a more complicated objective function. We need the following result.

Proposition 4.3. *Let $u \in \mathrm{R}^m$ be nonnegative. Then*

(a) *there exists $E \in \mathrm{R}^{k\times\ell}$ with*

$$EZUZ^T = -YUZ^T; \tag{4.2.3}$$

(b) *$EZU^{1/2}$ and*

$$K(u) := YUY^T - EZUZ^TE^T \tag{4.2.4}$$

are independent of which E satisfying (4.2.3) *is chosen; and*

(c) *$XUX^T - \bar{K}$ is positive semidefinite iff $K \preceq K(u)$.*

Proof. (a) We need to show that the row space of YUZ^T is contained in that of ZUZ^T. If not, there is a vector q that is orthogonal to the latter but not to the former, so that $ZUZ^Tq = 0$ while YUZ^Tq is nonzero. But then

$$\begin{aligned} 0 &\le \begin{pmatrix} p \\ q \end{pmatrix}^T \begin{bmatrix} YUY^T & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \\ &= p^TYUY^Tp + 2p^TYUZ^Tq + q^TZUZ^Tq, \end{aligned}$$

which is negative for p a sufficiently small negative multiple of YUZ^Tq.

(b) Suppose E and E' both satisfy (4.2.3). Then $(E-E')ZUZ^T = 0$, and hence $\|(E-E')ZU^{1/2}\|^2 = \mathrm{Trace}((E-E')ZUZ^T(E-E')^T) = 0$, so $EZU^{1/2}$ is uniquely defined. As for $K(u)$, we find

$$EZUZ^TE^T - E'ZUZ^T(E')^T = (E-E')ZUZ^TE' + EZUZ^T(E-E')^T = 0,$$

as desired.

(c) For any p and q and any E satisfying (4.2.3), we have

$$\begin{aligned} \begin{pmatrix} p \\ q \end{pmatrix}^T (XUX^T - \bar{K}) \begin{pmatrix} p \\ q \end{pmatrix} &= \begin{pmatrix} p \\ q \end{pmatrix}^T \begin{bmatrix} YUY^T - K & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \\ &= \begin{pmatrix} p \\ q \end{pmatrix}^T \begin{bmatrix} EZUZ^TE^T & -EZUZ^T \\ -ZUZ^TE^T & ZUZ^T \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \\ &\quad + \begin{pmatrix} p \\ q \end{pmatrix}^T \begin{bmatrix} K(u)-K & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} \\ &= (q - E^Tp)^TZUZ^T(q - E^Tp) + p^T(K(u)-K)p. \end{aligned}$$

Hence if $K \preceq K(u)$, the left-hand side is nonnegative, while if not, we can choose p and then $q = E^Tp$ so that the left-hand side is negative. ∎

If E satisfies (4.2.3), there are several ways to write $K(u)$:

$$\begin{aligned} K(u) &= YUY^T - EZUZ^TE^T = YUY^T + EZUY^T \\ &= YUY^T + YUZ^TE^T = (Y+EZ)U(Y+EZ)^T. \end{aligned} \tag{4.2.5}$$

The equation above immediately shows that $K(u)$ is positive semidefinite (since XUX^T is positive semidefinite, this also follows from (c) above), but we will be interested in cases where it is positive definite.

From the proposition, $K(u)$ is the maximal K (in the sense of the positive semidefiniteness order) with $XUX^T - \bar{K}$ positive semidefinite, but we would like to show that K can be chosen as $K(u)$ without loss of generality. For this we use

Proposition 4.4. *If K and K' are matrices in $\mathcal{S}^k$ with $K \preceq K'$, then* $\operatorname{lndet}(K) \le \operatorname{lndet}(K')$.

Proof. If K fails to be positive definite, $\operatorname{lndet}(K)$ is negative infinity and there is nothing to prove. Hence assume K, and a fortiori K', are positive definite. Since $-\operatorname{lndet}$ is convex, we have

$$\begin{aligned} -\operatorname{lndet}(K) &\ge -\operatorname{lndet}(K') + \nabla(-\operatorname{lndet})(K') \bullet (K - K') \\ &= -\operatorname{lndet}(K') + (K')^{-1} \bullet (K' - K) \ge -\operatorname{lndet}(K'), \end{aligned}$$

since the trace product of two positive semidefinite matrices is nonnegative. ∎

From the two propositions above, we see that (D) can be alternatively written as

$$(D') \quad \max\{\bar{g}(u) := \operatorname{lndet}(K(u)) : u \in \mathrm{R}^m,\ e^T u = 1,\ u \ge 0\}. \tag{4.2.6}$$

This is also a convex problem. We need to show that $\bar{g}$ is concave. Indeed, if u and w are nonnegative, with finite values of $\bar{g}$, and $0 \le \lambda \le 1$, then for $v := (1-\lambda)u + \lambda w$, we have $XUX^T - K(u) \succeq 0$ and $XWX^T - K(w) \succeq 0$, implying $XVX^T - [(1-\lambda)K(u) + \lambda K(w)] \succeq 0$. Hence by Proposition 4.3, $K(v) \succeq (1-\lambda)K(u) + \lambda K(w)$. Then Proposition 4.4 and the concavity of the logdet function yield the desired result.

Note that $K(u) = XUX^T$ if $k = n$, so that we recover the dual of the MVEE problem as expected.

We say u is dual feasible if it is feasible in (D') with a finite objective value, or equivalently, if there is some K with (u, K) feasible for (D).

We now provide a direct proof of weak duality, since it highlights the conditions for optimality.

Proposition 4.5. *If H and (u, K) are feasible in (P) and (D), respectively, then $f(H) \ge g(u, K)$.*

Proof. Once again we use the fact that the trace product of two positive semidefinite matrices is nonnegative. So we have

$$0 \le H \bullet (XUX^T - \bar{K}) = \sum_i u_i x_i^T H x_i - H \bullet \bar{K} \le k - H_{YY} \bullet K. \tag{4.2.7}$$

This shows that the positive definite matrix $H_{YY}^{1/2} K H_{YY}^{1/2}$ has positive eigenvalues λ_j that sum to $I \bullet H_{YY}^{1/2} K H_{YY}^{1/2} = H_{YY} \bullet K \le k$; and of course the similar matrix $H_{YY}K$ has the same eigenvalues.

Hence

$$\begin{aligned} f(H)-g(u,K) &= -\ln\det(H_{YY})-\ln\det(K) \\ &= -\ln\det(H_{YY}K) \\ &= -\ln(\textstyle\prod_j \lambda_j) \\ &= -k\ln(\textstyle\prod_j \lambda_j)^{1/k} \\ &\geq -k\ln(\textstyle\sum_j \lambda_j/k) \\ &\geq -k\ln(k/k) = 0, \end{aligned} \tag{4.2.8}$$

where the first inequality is the arithmetic-geometric mean inequality and the second follows from the bound on the sum of the eigenvalues from (4.2.7). □

Now we establish strong duality.

Theorem 4.6. *Under assumption* (4.1.1), (P) *and* (D) *have optimal solutions* H^* *and* (u^*,K^*) *and there is no duality gap:* $f(H^*)=g(u^*,K^*)$.

Proof. We first prove that (P) has an optimal solution. We proceed as in the MVEE case. We note that ϵI is feasible for sufficiently small positive ϵ, and so we can add the constraint that $-\ln\det(H_{YY}) \leq -k\ln\epsilon$. With the linear constraints on H and the positive semidefiniteness requirement, this defines a closed set on which the objective function is continuous.

Next, since we are assuming X has full rank, we can show exactly as in Theorem 2.2 that the feasible region is bounded. Hence an application of the Weierstrass theorem implies that (P) has an optimal solution, say H^*. Then the equivalence of (P) and (P') implies that the latter also has an optimal solution, say $(H^\dagger,E^*)$. We need to work with (P') rather than (P) when we discuss optimality conditions to avoid dealing with the complicated semidefiniteness constraint.

The Karush–John optimality conditions for (P') imply that there are nonnegative multipliers $\tau \in \mathrm{R}$ and $u \in \mathrm{R}$, not both zero, satisfying

$$-\tau(H^\dagger)^{-1} + \sum_i u_i(y_i+E^*z_i)(y_i+E^*z_i)^T = 0, \tag{4.2.9}$$

$$2\sum_i u_i H^\dagger y_i z_i^T + 2\sum_i u_i H^\dagger E^* z_i z_i^T = 0, \text{ and} \tag{4.2.10}$$

$$u_i((y_i+E^*z_i)^T H^\dagger (y_i+E^*z_i)-k) = 0, \quad i=1,\ldots,m. \tag{4.2.11}$$

The first equation sets the derivative of $\tau\bar{f}(E,H')+\sum_i u_i((y_i+Ez_i)^T H'(y_i+Ez_i)-k)$ with respect to H' at $(E^*,H^\dagger)$ to zero, and the third is complementary slackness. The second equation sets the derivative of the function above with respect to E at $(E^*,H^\dagger)$ to zero, noting that

$$(y_i+Ez_i)^T H'(y_i+Ez_i)-k = y^T H' y - k + 2E\bullet(H'y_i z_i^T) + E\bullet(H'Ez_i z_i^T).$$

Our first task is to show that τ is positive. Indeed, suppose that it is zero. Then (4.2.9) implies that

$$\left(\sum_i u_i(y_i+E^*z_i)(y_i+E^*z_i)^T\right)\bullet H^\dagger = 0,$$

and together with (4.2.11), we obtain $\sum_i u_i = 0$, so that all multipliers vanish, which is a contradiction. So τ is positive, and without loss of generality we may scale the multipliers

so that it is 1. In this case, (4.2.9) gives

$$\left(\sum_i u_i(y_i + E^* z_i)(y_i + E^* z_i)^T\right) \bullet H^\dagger = (H^\dagger)^{-1} \bullet H^\dagger = k.$$

Now, using (4.2.11), we find that $\sum_i u_i = 1$.

Next, (4.2.10) can be written as $2H^\dagger(YUZ^T + E^* ZUZ^T) = 0$, and since $H^\dagger$ is positive definite and hence nonsingular,

$$YUZ^T + E^* ZUZ^T = 0.$$

Then (4.2.9) can be written as

$$\begin{aligned}(H^\dagger)^{-1} &= YUY^T + YUZ^T(E^*)^T + E^* ZUY^T + E^* ZUZ^T(E^*)^T \\ &= YUY^T - E^* ZUZ^T(E^*)^T.\end{aligned} \tag{4.2.12}$$

Let $K := K(u) = YUY^T - E^* ZUZ^T(E^*)^T$. Then (u,K) is feasible in (D). Moreover, since $K = (H^\dagger)^{-1}$, we find $\bar{f}(H^\dagger, E^*) = g(u,K)$, so that by weak duality, both solutions are optimal. It follows that (D) has an optimal solution, and also that all optimal solutions have no duality gap. □

Our proof also establishes the following.

Corollary 4.7. *Any Karush–John point for (P') is an optimal solution.*

Since $-\ln\det$ is a strictly convex function ($\ln\det$ strictly concave), it follows that the H_{YY} part of an optimal solution H to (P) is unique, as is K in an optimal solution (u,K) to (D). However, the rest of H and u can be far from unique, as illustrated below.

Example 4.8. Suppose we seek the minimum-area cylinder with $k = 1$ containing the points x_i, $i = 1,2,3$, that are the columns of the matrix

$$X = \begin{bmatrix} 3 & 2 & 1 \\ 0 & 2 & 3 \end{bmatrix}.$$

Any positive semidefinite 2×2 matrix can be written in the form

$$H := H(\beta,\eta,\phi) := \begin{bmatrix} \beta^2 & \beta\eta \\ \beta\eta & \eta^2 + \phi \end{bmatrix};$$

the rather strange parametrization will become more intuitive below. Here without loss of generality $\beta \geq 0$ and we require $\phi \geq 0$. Then $x_i^T H x_i \leq 1$ for all i iff $9\beta^2 \leq 1$, $(2\beta + 2\eta)^2 + 4\phi \leq 1$, and $(\beta + 3\eta)^2 + 9\phi \leq 1$. Hence if $H(\beta,\eta,\phi)$ is feasible, so is $H(\beta,\eta,0)$, and in this case $x^T H x \leq 1$ defines a *strip*, that is the set of points between two parallel lines; more precisely,

$$x^T H(\beta,\eta,0)x \leq 1 \Leftrightarrow -1 \leq \beta y + \eta z \leq 1.$$

In (P) we wish to minimize $-\ln\det(H_{YY}) = -\ln\beta^2$, so we set $\beta = 1/3$, and for feasibility, $-4/9 \leq \eta \leq 1/6$. This gives the set of optimal ellipsoidal cylinders, the set of all strips defined by $-1 \leq y/3 + \eta z \leq 1$ for $-4/9 \leq \eta \leq 1/6$. Note that, in the equivalent problem (P'), $H' = 1/9$ and $E = 3\eta$ lies in $[-4/3, 1/2]$. All these ellipsoidal cylinders share the

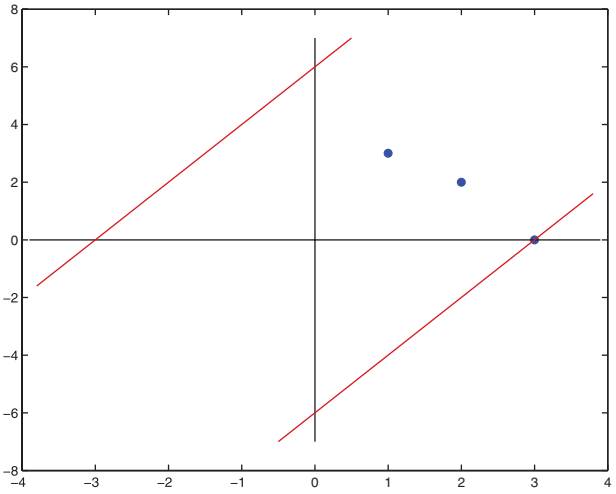


Figure 4.2. *Minimum-area (-length) ellipsoidal cylinder (strip).*

same base one-dimensional ellipsoid, the interval joining $[-3;0]$ and $[3;0]$. The points and one possible cylinder, corresponding to $\eta = -1/6$, are shown in Figure 4.2.

In the dual problem, if $u := (\lambda;\mu;\nu)$ and $K = \varkappa$, then

$$XUX^T - \bar{K} = \begin{bmatrix} 9\lambda+4\mu+\nu-\varkappa & 4\mu+3\nu \\ 4\mu+3\nu & 4\mu+9\nu \end{bmatrix},$$

so (u,K) is feasible if $\lambda,\mu,\nu \geq 0$, $\lambda+\mu+\nu = 1$, and $\varkappa \leq 9\lambda+16\mu\nu/(4\mu+9\nu)$ (or $\mu = \nu = 0$ and $\varkappa \leq 9\lambda$). Since we wish to maximize $\ln\det(K) = \ln\varkappa$, we set $\varkappa$ to 9, λ to 1, and μ and ν to 0. Note that *any* E satisfies $EZUZ^T = -YUZ^T$, since both YUZ^T and ZUZ^T are zero matrices. ∎

In the example above, there are several alternative optimal primal solutions but a unique optimal dual solution. We now modify the example so that the optimal primal solution is unique, but several optimal dual solutions and XUX^T matrices are possible.

Example 4.9. We change the second and third columns of X to get

$$X := \begin{bmatrix} 3 & 3 & 3 \\ 0 & 1 & -1 \end{bmatrix}.$$

Then $H = H(\beta,\eta,\phi)$ is feasible iff $9\beta^2 \leq 1$, $(3\beta+\eta)^2+\phi \leq 1$, and $(3\beta-\eta)^2+\phi \leq 1$. We wish to maximize $-\ln\beta^2$ again, so the only optimal solution has $\beta = 1/3$, and $\eta = \phi = 0$. This corresponds to the strip $-3 \leq y \leq 3$.

In the dual problem, if $u := (\lambda;\mu;\nu)$ and $K = \varkappa$, then

$$XUX^T - \bar{K} = \begin{bmatrix} 9-\varkappa & 3\mu-3\nu \\ 3\mu-3\nu & \mu+\nu \end{bmatrix},$$

so (u,K) is feasible if $\lambda,\mu,\nu \geq 0$, $\lambda+\mu+\nu = 1$, and $\varkappa \leq 9-9(\mu-\nu)^2/(\mu+\nu)$ if $\mu+\nu > 0$, $\varkappa \leq 9$ otherwise. Since we wish to maximize $\ln\varkappa$, we set $\varkappa$ to 9, and choose any nonnegative λ,μ,ν summing to 1 with $\mu = \nu$. Corresponding to the optimal solutions $u = (1;0;0)$ and $\hat{u} = (0;1/2;1/2)$, we have

$$XUX^T = \begin{bmatrix} 9 & 0 \\ 0 & 0 \end{bmatrix} \text{ and } X\hat{U}X^T = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}.$$

For the first choice, any E satisfies $EZUZ^T = -YUZ^T$, while for the second, only $E = 0$ satisfies $EZ\hat{U}Z^T = -Y\hat{U}Z^T$. ∎

4.3 ▪ Optimality conditions for the MAEC problem

Having established strong duality, we easily obtain necessary and sufficient optimality conditions.

Proposition 4.10. *Suppose H and (u,K) are feasible for (P) and (D), respectively. Then each is optimal iff the following conditions hold:*

(a) $H \bullet (XUX^T - \bar{K}) = 0$;

(b) $u_i > 0$ *only if* $x_i^T H x_i = k$; *and*

(c) $H_{YY} = K^{-1}$.

Proof. Indeed, condition (c) alone implies that the two objective values are equal, and hence that each solution is optimal. Conversely, if the solutions are optimal, by strong duality they must have the same objective values. Hence we must have equality throughout (4.2.8) and therefore throughout (4.2.7). The latter implies (a) and, since H is feasible, (b). It also shows that the sum of the eigenvalues of $H_{YY}K$ is exactly k. The former implies that all these eigenvalues are equal, and hence all are 1. This then gives (c). □

We will show that these two definitions are related, but first we want to make some remarks about the axis matrix E.

We have defined the axis matrix so far in two ways: corresponding to a primal solution H via $E = H_{YY}^{-1}H_{YZ}$ as in the proof of Lemma 4.2; and corresponding to a dual solution u as a solution to $EZUZ^T = -YUZ^T$ as in Proposition 4.3. We now show that these two methods are related for feasible solutions if H, u, and K, where $K = K(u)$, satisfy condition (a) above.

Indeed, if $H \bullet (XUX^T - \bar{K}) = 0$, then this equation remains true if H_{ZZ} is replaced by $E^T H_{YY} E$, where E satisfies the "primal" equation $E = H_{YY}^{-1}H_{YZ}$. Indeed, we can write

$$H = \begin{bmatrix} H_{YY} & H_{YY}E \\ E^T H_{YY} & E^T H_{YY} E \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & H_{ZZ} - E^T H_{YY} E \end{bmatrix},$$

and since all the matrices above are positive semidefinite and therefore have a nonnegative inner product with $XUX^T - \bar{K}$, our assumption implies that all these inner products must be zero. But then

$$\begin{aligned} 0 &= \left(\begin{bmatrix} I \\ E^T \end{bmatrix} H_{YY} [I \quad E] \right) \bullet \begin{bmatrix} YUY^T - K & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} \\ &= H_{YY} \bullet \left([I \quad E] \begin{bmatrix} YUY^T - K & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} \begin{bmatrix} I \\ E^T \end{bmatrix} \right). \end{aligned}$$

Since H_{YY} is positive definite and the second matrix above is positive semidefinite, the latter must in fact be zero. This then implies that $[I \quad E]J$ is zero, where J is the positive semidefinite square root of $XUX^T - \bar{K}$, and hence

$$[I \quad E] \begin{bmatrix} YUY^T - K & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} = 0.$$

We deduce that $EZUZ^T = -YUZ^T$, so E satisfies the "dual" equation. The reverse implication does not hold, as shown below.

Let us illustrate these results in our two examples. In Example 4.8, several possible H's are optimal; one, corresponding to the strip in Figure 4.2, is

$$H = \begin{bmatrix} 1/9 & -1/18 \\ -1/18 & 1/36 \end{bmatrix}.$$

The only optimal u is $(1;0;0)$ and the only optimal K is 9. Then $XUX^T - \bar{K}$ is the zero matrix. Thus optimality condition (a) holds trivially. Also, $x_i^T H x_i$ is 1, 1/9, and 1/36 for $i = 1,2,3$, respectively, confirming (b). Finally, H_{YY} is 1/9 and K is 9, so (c) holds. For this H, $E = H_{YY}^{-1} H_{YZ} = -1/2$. The full range of E's corresponding to all optimal H's turns out to be $[-4/3, 1/2]$ (in our previous notation, $E = \eta/\beta$ with $\beta = 1/3$ and $\eta \in [-4/9, 1/6]$). However, any E satisfies $EZUZ^T = -YUZ^T$, since $YUZ^T = ZUZ^T = 0$.

In Example 4.9, there is only one optimal H given by

$$H = \begin{bmatrix} 1/9 & 0 \\ 0 & 0 \end{bmatrix}.$$

Several optimal u's are possible, but only one K, which is 9. If $u = (1;0;0)$, then $XUX^T - \bar{K}$ is the zero matrix and condition (a) holds trivially. If $u = (0;1/2;1/2)$, then

$$XUX^T - \bar{K} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that (a) still holds. Next $x_i^T H x_i$ is 1 for $i = 1,2,3$, confirming (b) for either choice of u above, or for any convex combination, and (c) holds as above. Since there is a unique H, there is a unique $E = H_{YY}^{-1} H_{YZ}$, equal to 0. If $u = (0;1/2;1/2)$, only $E = 0$ satisfies $EZUZ^T = -YUZ^T$, but for $u = (1;0;0)$ any E works, since $YUZ^T = ZUZ^T = 0$ in this case.

If H and (u,K) are optimal in (P) and (D), respectively, we can assume as above that $H_{ZZ} = E^T H_{YY} E$ with $E = H_{YY}^{-1} H_{YZ}$, and then it follows from the optimality conditions above that $(y_i + Ez_i)^T K^{-1} (y_i + Ez_i) \le k$ for all i, with equality if u_i is positive. Moreover, E also satisfies $EZUZ^T = -YUZ^T$. It is hard to recognize optimality or near-optimality of (u,K) alone without choosing a specific corresponding E. Thus the definition below refers to a triple (u,K,E).

Definition 4.11. *We call a triple* (u,K,E) dual feasible *if* $u \ge 0$, $e^T u = 1$, $EZUZ^T = -YUZ^T$, *and* $K = YUY^T - EZUZ^T E^T$ *is positive definite. We call such a triple* ϵ-primal feasible *if* $(y_i + Ez_i)^T K^{-1} (y_i + Ez_i) \le (1+\epsilon)k$ *for all* i, *and* ϵ-approximately optimal *if moreover* $(y_i + Ez_i)^T K^{-1} (y_i + Ez_i) \ge (1-\epsilon)k$ *whenever* $u_i > 0$.

Note that u is dual feasible if there are some K, E such that (u,K,E) is dual feasible. Also, in parallel with (3.1.11), we see that, with $\omega_i(u,K,E) := (y_i + Ez_i)^T K^{-1} (y_i + Ez_i)$ for each i,

$$u^T \omega(u,K,E) = \sum_i u_i (y_i + Ez_i)^T K^{-1} (y_i + Ez_i) = (Y + EZ) U (Y + EZ)^T \bullet K^{-1} = k, \tag{4.3.1}$$

where the last equation follows from (4.2.5).

It is then easy to show the following.

Proposition 4.12. *Suppose (u,K,E) is ϵ-primal feasible (in particular, it could be ϵ-approximately optimal). Define $H = H(u,K,E)$ by setting $H_{YY} = K^{-1}$, $H_{YZ} = H_{YY}E$, and $H_{ZZ} = E^T H_{YY} E$. Then $(1+\epsilon)^{-1}H$ is feasible in (P), and both this solution and (u,K) are within $k\ln(1+\epsilon)$ of being optimal in their respective problems. Furthermore, the ellipsoidal cylinder $\mathcal{C}(E,(1+\epsilon)^{-1}H_{YY})$ contains all x_i and comes within a factor $(1+\epsilon)^{k/2}$ of having the smallest cross-sectional area among such cylinders.*

Proof. The first claim follows directly from the definition of H and the hypothesis, and the second is a consequence of the fact that $f((1+\epsilon)^{-1}H)$ and $g(u,K)$ differ by exactly $k\ln(1+\epsilon)$. The final claim follows by the relationship between the objective function of (P) and the area of the cross section of $\mathcal{C}(E,(1+\epsilon)^{-1}H_{YY})$. □

The same proof shows that, if (u,K) is feasible in (D) and E is any matrix in $\mathrm{R}^{k\times\ell}$ such that $(y_i + Ez_i)^T K^{-1}(y_i + Ez_i) \le k$, $i = 1,\dots,m$, then (u,K) is optimal in (D) (and $H(u,K,E)$ as above is optimal in (P)).

We conclude this section with yet another characterization of the axis matrix E. As a consequence, we see that the bound k above is the best possible.

Proposition 4.13. *Let $\bar{E}\,ZUZ^T = -YUZ^T$. Then $\bar{E}$ minimizes $K(E) := (Y+EZ)U(Y+EZ)^T$ in the sense that, for all $E \in \mathrm{R}^{k\times\ell}$, $K(E) \succeq K(\bar{E})$. Moreover, $\hat{E}$ minimizes $K(E)$ in this sense iff $\hat{E}\,ZUZ^T = -YUZ^T$.*

(We are abusing notation here by using $K(E)$, whereas before we defined $K(u)$. However, as we have seen in (4.2.5), if $\bar{E}\,ZUZ^T = -YUZ^T$, then $K(\bar{E}) = YUY^T - \bar{E}\,ZUZ^T\bar{E}^T = K(u)$.)

Proof. We have

$$\begin{aligned}
&K(E) - K(\bar{E}) \\
&= YUY^T + E\,ZUY^T + YUZ^TE^T + E\,ZUZ^TE^T \\
&\qquad - YUY^T - \bar{E}\,ZUY^T - YUZ^T\bar{E}^T - \bar{E}\,ZUZ^T\bar{E}^T \\
&= (E-\bar{E})ZUY^T + YUZ^T(E-\bar{E})^T + E\,ZUZ^TE^T - \bar{E}\,ZUZ^T\bar{E}^T \\
&= -(E-\bar{E})ZUZ^T\bar{E}^T - \bar{E}\,ZUZ^T(E-\bar{E})^T + E\,ZUZ^TE^T - \bar{E}\,ZUZ^T\bar{E}^T \\
&= (E-\bar{E})ZUZ^T(E-\bar{E})^T \quad \succeq \quad 0.
\end{aligned}$$

Moreover, if $\hat{E}$ also minimizes $K(E)$ in this sense, then $K(\hat{E}) \succeq K(\bar{E})$ as above and $K(\bar{E}) \succeq K(\hat{E})$, so their difference is zero. Hence $(\hat{E}-\bar{E})ZUZ^T(\hat{E}-\bar{E})^T = 0$, and then by considering the positive semidefinite square root of ZUZ^T, we find as in the argument below Proposition 4.10 that $(\hat{E}-\bar{E})ZUZ^T = 0$, which implies that $\hat{E}$ also satisfies $\hat{E}\,ZUZ^T = -YUZ^T$. □

As a corollary, we prove a result related to ϵ-primal feasibility.

Corollary 4.14. *For any (u,K) feasible in (D), and for any matrix E in $\mathrm{R}^{k\times\ell}$, we have $\max_i (y_i + Ez_i)^T K^{-1}(y_i + Ez_i) \ge k$.*

Proof. Indeed, since (u,K) is feasible only if $(u,K(u))$ is, and $K \preceq K(u)$ so that $K^{-1} \succeq K(u)^{-1}$ (see Corollary A.7), it suffices to assume that $K = K(u)$. In this case,

$$\max_i (y_i + Ez_i)^T K^{-1}(y_i + Ez_i) \geq \sum_i u_i (y_i + Ez_i)^T K^{-1}(y_i + Ez_i)$$
$$= K^{-1} \bullet (Y + EZ)U(Y + EZ)^T.$$

But K^{-1} is positive definite, so by the proposition, the right-hand side is at least $K^{-1} \bullet (Y + \bar{E}Z)U(Y + \bar{E}Z)^T = K^{-1} \bullet K = k$, where $\bar{E}$ satisfies $\bar{E}ZUZ^T = -YUZ^T$. ∎

4.4 ▪ D_k-optimal design in statistics

We now return to the D-optimal design problem in statistics as discussed in Section 1.3, but now we suppose we are only interested in the first k (out of n) parameters.

As before, we assume a random variable V depends on some independent variables $x(t)$ through some unknown parameters $\theta \in \mathrm{R}^n$, but now we are only interested in the first k of these parameters (the others are sometimes termed "nuisance" parameters). We accordingly divide θ into $\theta_Y \in \mathrm{R}^k$ and $\theta_Z \in \mathrm{R}^\ell$, and similarly $x(t)$ into $y(t) \in \mathrm{R}^k$ and $z(t) \in \mathrm{R}^\ell$, to get the model

$$V = x(t)^T\theta + \epsilon = y(t)^T\theta_Y + z(t)^T\theta_Z + \epsilon,$$

where, as before, ϵ is a normal random variable with mean 0 and variance σ^2.

To estimate θ_Y, we observe V at m different values of t, corresponding to $x_i = (y_i; z_i) := x(t_i)$, $i = 1, \ldots, m$, obtaining the vector $v \in \mathrm{R}^m$. We denote by X the $n \times m$ matrix whose columns are the x_i's, and similarly Y and Z. Again our estimator for θ is the (or a) solution of the least-squares problem $\min_\theta ||X^T\theta - v||$, which we prefer to write in the equivalent form

$$\min_\theta \frac{1}{2}||X^T\theta - v||^2 = \frac{1}{2}||Y^T\theta_Y + Z^T\theta_Z - v||^2.$$

Since the objective function is convex and differentiable, it is minimized exactly at those $\hat{\theta}$ satisfying the condition that the gradient vanishes, i.e., $XX^T\hat{\theta} = Xv$. We will not assume that X has rank n, and hence we cannot claim that this has a unique solution. However, it is important to realize that it always has at least one solution. Indeed, more generally we have

Proposition 4.15. *Suppose $W \in \mathrm{R}^{n\times m}$ and $u \in \mathrm{R}^m$ is nonnegative. Then, with $U := \mathrm{Diag}(u)$, $WUW^Tp = WUq$ has a solution for all $q \in \mathrm{R}^m$.*

Proof. We can split $U^{1/2}q$ into a part that lies in the nullspace of $WU^{1/2}$ and a part $(WU^{1/2})^T r$ that lies in the range of $(WU^{1/2})^T$. Then $p = r$ solves the system. ∎

Applying this with $W = X$ and u the vector of 1's shows that a solution θ exists.

Let us write the system in partitioned form:

$$\begin{bmatrix} YY^T & YZ^T \\ ZY^T & ZZ^T \end{bmatrix} \begin{pmatrix} \hat{\theta}_Y \\ \hat{\theta}_Z \end{pmatrix} = \begin{pmatrix} Yv \\ Zv \end{pmatrix}.$$

Now from Proposition 4.3 (with U the identity) there is a matrix E satisfying $E\,ZZ^T = -YZ^T$, and by premultiplying the second set of equations by E and adding to the first set, the system above is equivalent to

$$(YY^T - E\,ZZ^TE^T)\hat{\theta}_Y = (YY^T + E\,ZY^T)\hat{\theta}_Y = (Y+EZ)v, \quad ZZ^T\hat{\theta}_Z = Z(v - Y^T\hat{\theta}_Y).$$

We will assume that $K := YY^T - E\,ZZ^TE^T$ is nonsingular, so that $\hat{\theta}_Y$ is unique and given by $K^{-1}(Y+EZ)v$ (K and EZ do not depend on the choice of E by Proposition 4.3), and then there is always a corresponding $\hat{\theta}_Z$ by Proposition 4.15 applied to $W = Z$; $\hat{\theta}_Z$ may not be unique if Z does not have rank ℓ.

Since v is a sample from the random variable $Y^T\theta_Y + Z^T\theta_Z + \underline{\epsilon}$, where $\underline{\epsilon}$ is an m-dimension $N(0,\sigma^2 I)$-distributed random variable, $\hat{\theta}_Y$ is a sample from the random variable

$$\begin{aligned}\hat{\Theta}_Y &:= K^{-1}(Y+EZ)(Y^T\theta_Y + Z^T\theta_Z + \underline{\epsilon}) \\ &= K^{-1}(Y+EZ)((Y+EZ)^T\theta_Y + \underline{\epsilon}) \\ &= \theta_Y + K^{-1}(Y+EZ)\underline{\epsilon},\end{aligned}$$

where the second equation used $(Y+EZ)Z^T = 0$ and the third equation used $K = (Y+EZ)(Y+EZ)^T$ from (4.2.5), with U the identity. Hence our estimator is unbiased, and its variance is

$$K^{-1}(Y+EZ)\mathbf{E}(\underline{\epsilon\epsilon}^T)(Y+EZ)^TK^{-1} = \sigma^2K^{-1}.$$

As before, our interest is in designing an experiment, and so we will choose a distribution given by weights u on the points x_i, so that the variance from making N experiments will be

$$\frac{\sigma^2}{N}(YUY^T - E\,ZUZ^TE^T)^{-1},$$

assuming that we make an experiment at point x_i exactly Nu_i times. A D_k-optimal design is one where we choose the weights u to minimize the determinant of this variance matrix, ignoring the requirement that each Nu_i should be integer. We are thus led to the optimization problem

$$(D'') \qquad \begin{aligned} \max_u \quad & \ln\det(K(u)) \\ & e^Tu = 1, \\ & u \geq 0, \end{aligned}$$

which coincides exactly with the dual problem (D') to the MAEC problem.

It can be seen that the discussion in the previous section is in a sense related to the equivalence of D-optimality and G-optimality (see Sections 1.3 and 2.2), but is complicated by the need to choose the correct axis matrix in considering the criterion analogous to G-optimality: u minimizes $\max_i(y_i + Ez_i)^TK(u)^{-1}(y_i + Ez_i)$ and achieves the optimal value *k as long as* we choose the appropriate E.

4.5 ▪ Collision detection

Now we turn to a very different application of the MAEC problem. Suppose we have planned trajectories for a collection of objects in $\mathbf{R}^3$: these could be robotic arms and parts, or characters in an animation. We want to know if these trajectories will lead to

collisions; if so, we need to revise the trajectories to either avoid the collisions or take them into account.

For each object, suppose we have a set $(y_i; t_i)$, $i = 1, m$, of the space-time coordinates of key points of the object. We approximate the object by an ellipsoid moving at a uniform speed in a fixed direction without rotation. The latter gives rise to the set of space-time pairs defined by

$$(y - vt - \bar{y})^T H'(y - vt - \bar{y}) \leq 3,$$

which is a noncentral ellipsoidal cylinder. The minimum-volume ellipsoid with such a corresponding cylinder containing all the points (y_i, t_i) can be found as discussed in Section 4.1, by finding the minimum-area (central) ellipsoidal cylinder containing all $(y_i; t_i; 1)$ in R^5 with $k = 3$. If the key points contain all vertices of a polytopal object, then this ellipsoidal cylinder is a conservative model of the object's movement, in the sense that, if two such ellipsoidal cylinders do not intersect, then there is no collision between the objects, at least at the common times of observation.

How do we detect if two such cylinders intersect? We would like to determine if there is some $(y; t) \in \mathrm{R}^4$ satisfying

$$(y - v_j t - \bar{y}_j)^T H_j'(y - v_j t - \bar{y}_j) \leq 3$$

for $j = 1, 2$. First, we perform a Cholesky factorization of H_1' and scale it so that the first of the two inequalities above becomes

$$\|L(y - v_1 t - \bar{y}_1)\| \leq 1.$$

Next, we change the variables to $w := L(y - v_1 t - \bar{y}_1)$ and t, so that $y - v_2 t - \bar{y}_2 = L^{-1}w + (v_1 - v_2)t + (\bar{y}_1 - \bar{y}_2)$. Now we can write $(y - v_2 t - \bar{y}_2)^T H_2'(y - v_2 t - \bar{y}_2)$ as a convex quadratic function of w and t, and we want to know if its minimal value subject to $\|w\| \leq 1$ is at most 3. We can minimize this as a function of t (which will be a linear function of w) and substitute this in; the result will be a convex quadratic function of w alone, which we wish to minimize subject to the norm constraint on w. This is exactly the so-called trust-region subproblem of nonlinear programming, and there are efficient (both theoretically and practically: see, e.g., [21, 87]) methods for its solution.

If the objects are moving in a gravitational field, they may be better described by parabolic trajectories:

$$y = \bar{y} + vt + \frac{1}{2} g t^2,$$

where $g \in \mathrm{R}^3$ represents the gravitational field. We can then proceed exactly as above after replacing y and y_i throughout by $\hat{y} := y - (1/2)g t^2$ and $\hat{y}_i := y_i - (1/2)g t_i^2$, respectively.

4.6 ▪ Notes and references

The problem of finding the minimum-area ellipsoidal cylinder first arose in the context of D_k-optimal design. (We must comment briefly on our notation. Statisticians have consistently used k for the dimension of the full space, our n, and s for the number of parameters of interest, our k. The choice made here was dictated by the customary usage in optimization of n and m as the dimensions.) Again, Silvey [74] in 1972 raised the question of whether this problem was related to the optimal design problem, although Sibson's reply [70] only dealt with the minimum-volume ellipsoid case, $k = n$. A year later, Silvey and Titterington [72] gave precise formulations of these problems in the forms (P') and (D') and established duality properties. They termed the MAEC

problem the "thinnest cylinder" problem. Titterington [79] discusses both the centered and the noncentered MAEC problems, and shows the reduction of the noncentered MVEE problem, both to a higher-dimensional centered MVEE problem and to a higher-dimensional centered MAEC problem. He also shows that the noncentered MAEC problem can be reduced to a higher-dimensional centered MAEC problem.

As mentioned above, duality questions were raised by Silvey and dealt with formally by Silvey and Titterington. Optimality conditions have a more checkered history. Kiefer [51] gave some incomplete optimality conditions for D_k-optimality, and Karlin and Studden [46] obtained conditions that also allowed ZUZ^T to be singular. Atwood [8] noted an ambiguity in these conditions related to the matrix E and clarified them. A clear statement appears in Silvey and Titterington.

Further discussion of the D_k-optimal design problem appears in the books of Fedorov [27], Silvey [73], and Pukelsheim [63]. Instead of just the first k (s) parameters, statisticians are often interested in certain linear combinations of the parameters, so that $A^T\theta$ replaces θ_Y as the object of interest. By a change of variables, this can be reduced to the case treated here, for which the dual MAEC problem is more intuitive.

The application to collision detection is believed to be new.

Once again, our claim that (P') is nonconvex is perhaps too strong, since the constraints can be written in the form $\|(H')^{1/2}y_i + (H')^{1/2}Ez_i\| \leq \sqrt{k}$, $i = 1,\ldots,m$. When expressed in terms of the symmetric matrix $B := (H')^{1/2}$ and the rectangular matrix $F := (H')^{1/2}E$, these are convex constraints, and the objective function can be written as $\min -2\ln\det(B)$. However, the resulting constraints are much more complicated than our linear constraints in (P) if $k > 1$. On the other hand, for $k = 1$, this can be written as a linear programming problem in the scalar variable $\beta = B$ and the vector variable $f = F^T$: $\max 2\beta$, $-1 \leq y_i\beta + f^T z_i \leq 1$ for all i.

Chapter 5

Algorithms for the MAEC Problem

Here we develop and analyze first-order algorithms for the solution of the MAEC problem

$$(P)\qquad \begin{array}{cc} \min_{H\in\mathcal{S}^n} & f(H) := -\mathrm{lndet}(H_{YY}) \\ & x_i^T H x_i \le k, \quad i=1,\dots,m, \\ & H \succeq 0 \end{array}$$

and its dual

$$(D)\qquad \begin{array}{cc} \max_{u\in\mathbf{R}^m, K\in\mathcal{S}^k} & g(u,K) := \mathrm{lndet}(K) \\ & XUX^T - \bar{K} \succeq 0, \\ & e^T u = 1, \\ & u \ge 0 \end{array}$$

with its equivalent form

$$(D')\quad \max\{\bar{g}(u) := \mathrm{lndet}(K(u)) : u\in\mathrm{R}^m,\ e^T u = 1,\ u\ge 0\}.$$

Recall that, as long as ZUZ^T is nonsingular,

$$K(u) = YUY^T - YUZ^T(ZUZ^T)^{-1}ZUY^T.$$

The partial derivative of $K(u)$ with respect to u_i can be obtained by differentiating this equation using the product rule. Recalling that

$$E := -YUZ^T(ZUZ^T)^{-1}$$

when the right-hand side is defined, we find this partial derivative is

$$y_i y_i^T + Ez_i y_i^T + y_i z_i^T E^T + Ez_i z_i^T E^T = (y_i + Ez_i)(y_i + Ez_i)^T,$$

where the last term on the left-hand side uses the derivative of the inverse in (A.4.4). The chain rule then yields

$$\nabla\bar{g}(u) = \omega(u) := ((y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i))_{i=1}^m.$$

As in Chapter 3, we can use the derivative of the inverse matrix in (A.4.4) to obtain second derivatives:

$$(\nabla^2\bar{g}(u))_{ij} = ((y_i + Ez_i)^T K(u)^{-1}(y_j + Ez_j))^2, \quad i,j = 1,\dots,m.$$

However, it is important to realize that, when ZUZ^T is singular, $\bar{g}$ may not be differentiable, although it is always concave. Consider the following simple perturbation of Example 4.9.

Example 5.1. We seek the minimum-area cylinder with $k = 1$ containing the points x_i, $i = 1,2,3$, that are the columns of the matrix

$$X = \begin{bmatrix} 2 & 3 & 3 \\ 0 & 1 & -1 \end{bmatrix}.$$

An analysis similar to that used in Examples 4.8 and 4.9 shows that the unique optimal solution to (P) is

$$H := \begin{bmatrix} 1/9 & 0 \\ 0 & 0 \end{bmatrix}$$

with objective value $\ln 9$, and the unique optimal solution to (D) is $\bar{u} := (0; 1/2; 1/2)$ with the same objective. However, let us consider the point $u := (1; 0; 0)$. We will consider the directional derivatives of $\bar{g}$ in the directions $d_2 := (-1; 1; 0)$, $d_3 := (-1; 0; 1)$, and $d := d_2 + d_3$. For $\hat{u} := u + \epsilon d_2$ $(\epsilon \geq 0)$, we have

$$X\hat{U}X^T = \begin{bmatrix} 4+5\epsilon & 3\epsilon \\ 3\epsilon & \epsilon \end{bmatrix},$$

so that $K(\hat{u}) = 4 - 4\epsilon$ and the directional derivative of $\operatorname{lndet}(K(u))$ in direction d_2 is -1. For $\hat{u} := u + \epsilon d_3$ $(\epsilon \geq 0)$, the situation is exactly the same, except that the off-diagonal entries of $X\hat{U}X^T$ are negated, so that again $K(\hat{u}) = 4 - 4\epsilon$ and the directional derivative of $\operatorname{lndet}(K(u))$ in direction d_3 is -1. Finally, for $\hat{u} := u + \epsilon d$ $(\epsilon \geq 0)$, we have

$$X\hat{U}X^T = \begin{bmatrix} 4+10\epsilon & 0 \\ 0 & 2\epsilon \end{bmatrix}$$

so that $K(\hat{u}) = 4 + 10\epsilon$ and the directional derivative of $\operatorname{lndet}(K(u))$ in direction d is $5/2$. Since this is not the sum of -1 and -1, while d is the sum of d_2 and d_3, we conclude that $\bar{g}$ is not differentiable at this u. ∎

In the first section below we discuss the derivative properties of $\bar{g}$ in more detail. Section 5.2 considers coordinate-ascent algorithms assuming that ZUZ^T remains nonsingular at all iterations. Global and local convergence properties are discussed in Sections 5.3 and 5.4, while Section 5.5 treats the case that ZUZ^T may be singular at some iterations. Finally, in Section 5.6 we give some computational results.

5.1 ▪ Derivative properties of the dual objective function

First we note that by combining Propositions 4.4 and 4.13 we obtain

Proposition 5.2. *We can write $\bar{g}$ as the pointwise minimum of continuously differentiable concave functions:*

$$\bar{g}(u) = \min\{\operatorname{lndet}([Y + EZ]U[Y + EZ]^T) : E \in \mathrm{R}^{k\times\ell}\} \tag{5.1.1}$$

for all dual feasible u, and the minimum is attained by all E satisfying $E\,ZUZ^T = -YUZ^T$.

Indeed, it follows from Section A.7 that there is a compact set $\mathcal{F}$ of $k \times \ell$ matrices so that, for any $u \geq 0$ with $e^T u = 1$, there is some $E \in \mathcal{F}$ satisfying $EZUZ^T = -YUZ^T$. We conclude that in the proposition above, we can restrict E in (5.1.1) to $\mathcal{F}$, and then a standard result in convex analysis gives the subdifferential of $\bar{g}$.

Corollary 5.3. *For any dual feasible u, the subdifferential of $\bar{g}$ at u is*

$$\begin{aligned}\partial \bar{g}(u) &:= \{z \in \mathrm{R}^m : z^T(v-u) \geq \bar{g}(v) - \bar{g}(u) \text{ for all } v \in \mathrm{R}^m\} \\ &= \mathrm{conv}\{[(y_i + Ez_i)^T K(u)^{-1}(y + Ez_i)]_{i=1}^m : E \in \mathcal{F}, EZUZ^T = -YUZ^T\}.\end{aligned} \tag{5.1.2}$$

We next consider directional derivatives of $\bar{g}$. We assume we are at some u feasible for (D) and we want to move towards or away from some vertex e_i of the unit simplex. So we will be moving in direction $d_i := e_i - u$, or maybe its negative. (Note that we used this convention above.) We use ∂_i to denote the directional derivative in direction d_i.

Proposition 5.4.

(i) *Suppose z_i lies in the range of ZUZ^T, so that $ZUZ^T v_i = z_i$ for some v_i. Then, for any E with $EZUZ^T = -YUZ^T$, we have:*

(a) *for any ϵ sufficiently small in absolute value, $U(\epsilon) := \mathrm{Diag}(u + \epsilon d_i)$, and $\eta := \epsilon/(1 - \epsilon + \epsilon v_i^T z_i)$,*

$$[E - \eta(y_i + Ez_i)v_i^T]ZU(\epsilon)Z^T = -YU(\epsilon)Z^T; \tag{5.1.3}$$

(b) $\partial_i K(u) = (y_i + Ez_i)(y_i + Ez_i)^T - K(u)$; *and*

(c) $\partial_i \bar{g}(u) = \omega_i - k$, *where* $\omega_i := (y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i)$.

(ii) *Suppose z_i does not lie in the range of ZUZ^T. Then, for any E with $E[ZUZ^T + z_i z_i^T] = -[YUZ^T + y_i z_i^T]$, we have*

$$y_i + Ez_i = 0 \tag{5.1.4}$$

and:

(a) *for any nonnegative ϵ and $U(\epsilon)$ as above,*

$$EZU(\epsilon)Z^T = -YU(\epsilon)Z^T; \tag{5.1.5}$$

(b) $\partial_i K(u) = (y_i + Ez_i)(y_i + Ez_i)^T - K(u) = -K(u)$; *and*

(c) $\partial_i \bar{g}(u) = \omega_i - k = -k$, *where* $\omega_i := (y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i) = 0$.

(iii) *If $0 < u_i < 1$, then case* (i) *above holds and, for any E with $EZUZ^T = -YUZ^T$, we have:*

(a) *for any ϵ sufficiently small in absolute value, $U(\epsilon) := \mathrm{Diag}(u + \epsilon d_i)$, and $\eta := \epsilon/(1 - \epsilon + \epsilon v_i^T z_i)$,*

$$[E - \eta(y_i + Ez_i)v_i^T]ZU(\epsilon)Z^T = -YU(\epsilon)Z^T; \tag{5.1.6}$$

(b) *the directional derivative of $K(u)$ in direction $-d_i$ is $K(u) - (y_i + Ez_i)(y_i + Ez_i)^T$; and*

(c) *the directional derivative of* $\bar{g}(u)$ *in direction* $-d_i$ *is* $k-\omega_i$, *where* $\omega_i := (y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i)$.

Proof. For part (i), note first that η is defined for any ϵ sufficiently small in absolute value, and also that $\eta = \delta/(1+\delta v_i^T z_i)$ for $\delta := \epsilon/(1-\epsilon)$. We have

$$\begin{aligned}
&[E-\eta(y_i+Ez_i)v_i^T][ZUZ^T+\delta z_i z_i^T]\\
&=-YUZ^T+\delta E z_i z_i^T-\eta(y_i+Ez_i)(z_i+\delta v_i^T z_i z_i)^T\\
&=-YUZ^T+\delta E z_i z_i^T-\delta(y_i+Ez_i)z_i^T\\
&=-[YUZ^T+\delta y_i z_i^T],
\end{aligned}$$

and multiplying by $1-\epsilon$ proves (a). Hence, since $\eta = \delta + o(\epsilon)$,

$$\begin{aligned}
&K(u+\delta e_i)\\
&=YUY^T+\delta y_i y_i^T-[E-\eta(y_i+Ez_i)v_i^T][ZUZ^T+\delta z_i z_i^T][E-\eta(y_i+Ez_i)v_i^T]^T\\
&=K(u)+\delta(y_i y_i^T+(y_i+Ez_i)v_i^T ZUZ^T E^T-Ez_i z_i^T E+E\,ZUZ^T v_i(y_i+Ez_i)^T)+o(\epsilon)\\
&=K(u)+\delta(y_i y_i^T+(y_i+Ez_i)z_i^T E^T-Ez_i z_i^T E^T+Ez_i(y_i+Ez_i)^T)+o(\epsilon)\\
&=K(u)+\delta(y_i+Ez_i)(y_i+Ez_i)^T+o(\epsilon).
\end{aligned}$$

Multiplying by $1-\epsilon$ gives $K(u+\epsilon d_i) = (1-\epsilon)K(u)+\epsilon(y_i+Ez_i)(y_i+Ez_i)^T+o(\epsilon)$, and this proves (b). Finally, the chain rule gives

$$\partial_i \bar{g}(u) = (\nabla(\mathrm{lndet})(K(u)))\bullet\partial_i K(u) = K(u)^{-1}\bullet[(y_i+Ez_i)(y_i+Ez_i)^T-K(u)] = \omega_i - k,$$

giving (c).

For part (ii), since z_i does not lie in the range of ZUZ^T, there is some vector v_i with $ZUZ^T v_i = 0$ but $z_i^T v_i$ nonzero. Without loss of generality we suppose $z_i^T v_i = 1$, and then $[ZUZ^T + z_i z_i^T]v_i = 0 + (z_i^T v_i)z_i = z_i$. Define E (maybe not uniquely) to satisfy

$$E[ZUZ^T + z_i z_i^T] = -[YUZ^T + y_i z_i^T]. \tag{5.1.7}$$

Such an E exists by Proposition 4.3. Then $Ez_i = E[ZUZ^T + z_i z_i^T]v_i = -[YUZ^T + y_i z_i^T]v_i = -YUZ^T v_i - y_i$. But $v_i^T ZUZ^T v_i = 0$ implies $U^{1/2}Z^T v_i = 0$, and hence $YUZ^T v_i = 0$. Thus $y_i + Ez_i = 0$, as desired. This equality and (5.1.7) give (a). So, with δ as above,

$$\begin{aligned}
K(u+\delta e_i) &= YUY^T + \delta y_i y_i^T - E[ZUZ^T + \delta z_i z_i^T]E^T\\
&= K(u) + \delta(y_i y_i^T - Ez_i z_i^T E)\\
&= K(u) = K(u) + \delta(y_i + Ez_i)(y_i + Ez_i)^T,
\end{aligned}$$

using $y_i = -Ez_i$. Multiplying by $1-\epsilon$ gives $K(u+\epsilon d_i) = (1-\epsilon)K(u) + \epsilon(y_i+Ez_i)(y_i + Ez_i)^T$, from which (b) follows, and then (c) is immediate as in the proof of part (i).

Finally, for part (iii) we note that Proposition 4.15 implies that whenever u_i is positive, z_i lies in the range of ZUZ^T. Then the analysis for part (i) holds, and the results hold by noting that the directional derivatives in the direction $-d_i$ are the derivatives of the appropriate expressions with respect to $-\epsilon$. ∎

Example 5.5. Let us revisit Example 5.1. Recall that $k = 1$,

$$X = \begin{bmatrix} 2 & 3 & 3 \\ 0 & 1 & -1 \end{bmatrix},$$

and, for $u := (1;0;0)$,

$$XUX^T = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}.$$

For $i = 2$, case (ii) above holds, and we find that $ZUZ^T + z_i z_i^T = 1$ and $YUZ^T + y_i z_i^T = 3$, so $E = -3$ and indeed $y_i + Ez_i = 0$. For any positive ϵ and $U(\epsilon) := \mathrm{Diag}(u + \epsilon d_i)$, we have

$$XU(\epsilon)X^T = \begin{bmatrix} 4+5\epsilon & 3\epsilon \\ 3\epsilon & \epsilon \end{bmatrix},$$

so that $EZU(\epsilon)Z^T = -YU(\epsilon)Z^T$ and, from our previous calculation, $\partial_i \bar{g}(u) = -1 = -k$, verifying the conclusions of the proposition.

Now let us add a new column $(3;0)$ to X (as in Example 4.9) in the fourth position and consider the case $i = 4$. Then, even though ZUZ^T is singular, z_i lies in the range of ZUZ^T, and so case (i) occurs. We find that any E satisfies $EZUZ^T = -YUZ^T$, and since now

$$XU(\epsilon)X^T = \begin{bmatrix} 4+5\epsilon & 0 \\ 0 & 0 \end{bmatrix},$$

we also have that any E satisfies $EZU(\epsilon)Z^T = -YU(\epsilon)Z^T$. So $K(u(\epsilon)) = 4 + 5\epsilon$, from which $\partial_i \bar{g}(u) = 5/4$. This agrees with the conclusion of part (i), since $(y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i) = 3(1/4)3 = 9/4$. ∎

5.2 ▪ Coordinate-ascent algorithms

Suppose we have a current feasible point u for (D); recall that this means that u satisfies the constraints, so that u is nonnegative and $e^T u = 1$, and that the objective function $\bar{g}(u) = \mathrm{lndet}(K(u))$ is finite, so that $K(u)$ is positive definite. We will assume in this and the next two sections that ZUZ^T is positive definite, so that $K(u) = YUY^T - YUZ^T(ZUZ^T)^{-1}ZUY^T$.

Suppose we also have at our disposal

$$\omega := \omega(u) = \nabla \bar{g}(u) = (y_i + Ez_i)^T K(u)^{-1}(y_i + Ez_i))_{i=1}^m, \tag{5.2.1}$$

$E = -YUZ^T(ZUZ^T)^{-1}$, and scaled Cholesky factorizations of XUX^T, ZUZ^T, and $K(u)$:

$$XUX^T = \phi^{-1}LL^T, \qquad ZUZ^T = \phi^{-1}L_Z L_Z^T, \qquad K := K(u) = \phi^{-1}L_K L_K^T, \tag{5.2.2}$$

with ϕ positive. (We will see below that the latter two factorizations come for free from the first if we make a simple modification.)

As in Chapter 3, we consider the following update of u:

$$u_+ := (1-\tau)u + \tau e_i. \tag{5.2.3}$$

Note again that u_+ can also be viewed as the result of taking a coordinate-ascent step to

$$\hat{u} := u + \lambda e_i$$

with $\lambda := \tau/(1-\tau)$, followed by a scaling to keep the coordinate sum equal to one: $u_+ := u_+(\lambda) := (1+\lambda)^{-1}\hat{u} = (1-\tau)\hat{u}$. Note that u_+ remains nonnegative as long as

$$-u_i \le \lambda < \infty, \tag{5.2.4}$$

which we henceforth assume. Note that $\lambda = \infty$ leads to $u_+ = e_i$, but this is infeasible if $k > 1$, since then $K(u_+)$ is singular. As we mentioned earlier, the case $k = 1$ reduces to a linear programming problem, so we do not consider it further, except for examples.

Let us examine the resulting changes in K, $\bar{g}$, and E. We have $ZU_+Z^T = (1+\lambda)^{-1}(ZUZ^T + \lambda z_i z_i^T)$, and hence

$$(ZU_+Z^T)^{-1} = (1+\lambda)((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_i z_i^T(ZUZ^T)^{-1}),$$

with $\mu := \lambda/(1+\lambda\zeta_i)$, where

$$\zeta_i := \zeta_i(u) := z_i^T(ZUZ^T)^{-1}z_i, \qquad \xi_i := \xi_i(u) := x_i^T(XUX^T)^{-1}x_i,$$

and we similarly abbreviate $\omega_i(u)$ to ω_i. Correspondingly,

$$(XU_+X^T)^{-1} = (1+\lambda)((XUX^T)^{-1} - \nu(XUX^T)^{-1}x_i x_i^T(XUX^T)^{-1}),$$

with $\nu := \lambda/(1+\lambda\xi_i)$.

Since $YU_+Z^T = (1+\lambda)^{-1}(YUZ^T + \lambda y_i z_i^T)$, we find

$$\begin{aligned} E_+ &= -(YUZ^T + \lambda y_i z_i^T)((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_i z_i^T(ZUZ^T)^{-1}) \\ &= E - \lambda y_i z_i^T(ZUZ^T)^{-1} - \mu E z_i z_i^T(ZUZ^T)^{-1} + \lambda\mu\zeta_i y_i z_i^T(ZUZ^T)^{-1} \\ &= E - \mu(y_i + Ez_i)((ZUZ^T)^{-1}z_i)^T. \end{aligned}$$

Then we obtain

$$\begin{aligned} K_+ &= YU_+Y^T + E_+ZU_+Y^T \\ &= (1+\lambda)^{-1}(YUY^T + \lambda y_i y_i^T + (E - \mu(y_i + Ez_i)z_i^T(ZUZ^T)^{-1})(ZUY^T + \lambda z_i y_i^T)) \\ &= (1+\lambda)^{-1}(K + \lambda y_i y_i^T + \lambda E z_i y_i^T + \mu(y_i + Ez_i)(Ez_i)^T - \lambda\mu\zeta_i(y_i + Ez_i)y_i^T) \\ &= (1+\lambda)^{-1}(K + \lambda(y_i + Ez_i)y_i^T + \mu(y_i + Ez_i)(Ez_i)^T - (\lambda - \mu)(y_i + Ez_i)y_i^T) \\ &= (1+\lambda)^{-1}(K + \mu(y_i + Ez_i)(y_i + Ez_i)^T). \end{aligned} \tag{5.2.5}$$

From this it is easy to derive an update of ω, but an indirect method provides insight and gives further information. The proposition below relates ω to ξ and ζ, and these can be updated using the formulae from Chapter 3.

Proposition 5.6. *With E and K as above, we have*

$$XUX^T = \begin{bmatrix} YUY^T & YUZ^T \\ ZUY^T & ZUZ^T \end{bmatrix} = \begin{bmatrix} I & -E \\ 0 & I \end{bmatrix}\begin{bmatrix} K & 0 \\ 0 & ZUZ^T \end{bmatrix}\begin{bmatrix} I & 0 \\ -E^T & I \end{bmatrix} \tag{5.2.6}$$

and

$$\det(XUX^T) = \det(K)\det(ZUZ^T). \tag{5.2.7}$$

Further, if K and ZUZ^T are positive definite, then

$$\omega_i = \xi_i - \zeta_i. \tag{5.2.8}$$

Proof. The first equation follows from multiplying out the right-hand side, and then the determinant formula is immediate by taking the determinant on both sides. If K and ZUZ^T are positive definite, then all matrices in (5.2.6) are nonsingular, and taking the inverses we obtain

$$(XUX^T)^{-1} = \begin{bmatrix} I & 0 \\ E^T & I \end{bmatrix} \begin{bmatrix} K^{-1} & 0 \\ 0 & (ZUZ^T)^{-1} \end{bmatrix} \begin{bmatrix} I & E \\ 0 & I \end{bmatrix}.$$

Now pre- and postmultiplying by x_i gives

$$\begin{aligned} \xi_i = x_i^T (XUX^T)^{-1} x_i &= (y_i^T + z_i^T E^T \; z_i) \begin{bmatrix} K^{-1} & 0 \\ 0 & (ZUZ^T)^{-1} \end{bmatrix} \begin{pmatrix} y_i + E z_i \\ z_i \end{pmatrix} \\ &= \omega_i + \zeta_i, \end{aligned}$$

as desired. □

Notice that (5.2.8) provides an alternative proof of (4.3.1) when it applies, since (3.1.11) shows $u^T \xi = n$ and $u^T \zeta = \ell = n - k$ follows similarly.

From (3.1.8) we obtain an update of ξ_i, and similar reasoning leads to an update of ζ_i:

$$\xi_i(u_+) = \frac{(1+\lambda)\xi_i}{1+\lambda\xi_i}, \qquad \zeta_i(u_+) = \frac{(1+\lambda)\zeta_i}{1+\lambda\zeta_i}. \tag{5.2.9}$$

Then (5.2.8) gives

$$\omega_i(u_+) = \frac{(1+\lambda)\omega_i}{(1+\lambda\xi_i)(1+\lambda\zeta_i)}. \tag{5.2.10}$$

How should we choose the index i and the stepsize λ? As for the minimum-volume ellipsoid problem, we choose i according to the directional derivative of the objective in the direction $d = e_i - u$. Using Proposition 5.4, we choose either i with ω_i maximum, or i with u_i positive and ω_i minimum. From (4.3.1), the largest ω_i is at least k and the smallest corresponding to a positive u_i is at most k. For the stepsize, note that

$$\det K(u_+(\lambda)) = (1+\lambda)^{-k}(1+\mu\omega_i)\det K$$

from (5.2.5), so that the objective function in terms of λ is

$$\begin{aligned} \bar{\gamma}(\lambda) &:= \operatorname{lndet}(K(u_+(\lambda))) \\ &= -k\ln(1+\lambda) + \ln\left(1 + \tfrac{\lambda\omega_i}{1+\lambda\zeta_i}\right) + \operatorname{lndet}(K) \\ &= -k\ln(1+\lambda) + \ln(1+\lambda\xi_i) - \ln(1+\lambda\zeta_i) + \operatorname{lndet}(K). \end{aligned} \tag{5.2.11}$$

Its derivative is

$$\begin{aligned} \bar{\gamma}'(\lambda) &= -\frac{k}{1+\lambda} + \frac{\xi_i}{1+\lambda\xi_i} - \frac{\zeta_i}{1+\lambda\zeta_i} \\ &= \frac{-k}{(1+\lambda)(1+\lambda\xi_i)(1+\lambda\zeta_i)}(a\lambda^2 - 2b\lambda + c), \end{aligned}$$

where $a := \xi_i\zeta_i \geq 0$, $b := -\zeta_i - \frac{\omega_i}{2} + \frac{\omega_i}{2k} \leq 0$, and $c := 1 - \frac{\omega_i}{k}$. Note that the term multiplying the quadratic is negative for positive λ and for λ greater than $-1/\xi_i$. From (5.2.11), we see that as long as ω_i is positive, so that $\zeta_i < \xi_i$, $\bar{\gamma}(\lambda)$ approaches $-\infty$ as λ

tends to $-1/\xi_i$ from above, so if its derivative is negative at 0, there will be a maximizer between 0 and $-1/\xi_i$. On the other hand, if ω_i is zero, then $\bar{\gamma}'(\lambda) = -k/(1+\lambda)$, so $\bar{\gamma}$ is maximized over feasible λ at $-u_i$. For now, assume the latter case does not arise, so that we are concerned with the roots of the quadratic.

Suppose first $c < 0$, so that $\omega_i > k$ and we wish to increase λ. Then if a is positive, $ac < 0$, so the quadratic has one root of each sign, and we want to increase λ to

$$\lambda^* = \frac{c}{b - \sqrt{b^2 - ac}} \tag{5.2.12}$$

(we have chosen this form to avoid cancellation in the expression $(b + \sqrt{b^2 - ac})/a$). If a is zero, we have a linear equation, with root $c/(2b)$, which is again given by (5.2.12).

Now assume $c > 0$, so that $\omega_i < k$ and we want to decrease λ. If $-b \leq \sqrt{ac}$, then the quadratic has either no roots or a repeated root, so the quadratic is nonpositive for all negative λ, and we use

$$\lambda^* = -u_i. \tag{5.2.13}$$

On the other hand, if $-b > \sqrt{ac}$, then there are two negative roots, so we set λ to the larger of these, if feasible, or to

$$\lambda^* = \max\left\{-u_i, \frac{c}{b - \sqrt{b^2 - ac}}\right\}. \tag{5.2.14}$$

Finally note that, if $\omega_i = 0$, then $c > 0$ and $-b = \zeta_i = \sqrt{\zeta_i^2} = \sqrt{ac}$, so that λ^* is correctly set by (5.2.13) in this case also.

We remark that exactly the same quadratic arises if we set $\omega_i(u_+)$ to k, from (5.2.10). Again, this is not surprising in view of the formula for the directional derivative in the direction d_i.

We are now ready to state our algorithms formally. As with the minimum-volume problem, we distinguish a Frank–Wolfe/Fedorov–Wynn algorithm that only considers positive λ, and a Wolfe–Atwood algorithm that also allows negative λ. Their names are appended by "C" to indicate that these are for the ellipsoidal cylinder case.

ALGORITHM 5.1.
(FWC Algorithm)

Step 0. Choose u feasible for (D) and $\epsilon > 0$. Compute $\omega = \omega(u)$ and (scaled) Cholesky factorizations of XUX^T, ZUZ^T, and $K(u)$.
Step 1. Given the current iterate u and its associated $\omega := \omega(u)$, compute $\epsilon_+ := \max_b(\omega_b - k)/k$, and let $b = i$ attain the maximum.
If $\epsilon_+ \leq \epsilon$, STOP: u (with $K(u)$ and $E(u) = -(YUZ^T)(ZUZ^T)^{-1}$) is ϵ-primal feasible. Otherwise, go to Step 2.
Step 2. Compute λ^* from (5.2.12) and update $u \leftarrow (1+\lambda^*)^{-1}(u + \lambda^* e_i)$.
Step 3. Update ω and scaled Cholesky factorizations of XUX^T, ZUZ^T, and $K(u)$, and go to Step 1.

ALGORITHM 5.2.
(WAC Algorithm)

Step 0. Choose u feasible for (D) and $\epsilon > 0$. Compute $\omega = \omega(u)$ and (scaled) Cholesky factorizations of XUX^T, ZUZ^T, and $K(u)$.

Step 1. Given the current iterate u and its associated $\omega := \omega(u)$, compute $\epsilon_+ := \max_h(\omega_h - k)/k$, with $h = i$ attaining the maximum, and $\epsilon_- := \max_h\{(k-\omega_h)/k : u_h > 0\}$, with $h = j$ attaining the maximum.
If $\max\{\epsilon_+, \epsilon_-\} \le \epsilon$, STOP: u (with $K(u)$ and $E(u) = -(YUZ^T)(ZUZ^T)^{-1}$) is ϵ-approximately optimal.
Otherwise, if $\epsilon_+ > \epsilon_-$, go to Step 2; else go to Step 3.
Step 2. Compute λ^* from (5.2.12) and update $u \leftarrow (1+\lambda^*)^{-1}(u + \lambda^* e_i)$. Go to Step 4.
Step 3. Compute λ^* from (5.2.13) or (5.2.14) with j replacing i and update $u \leftarrow (1+\lambda^*)^{-1}(u + \lambda^* e_j)$. Go to Step 4.
Step 4. Update ω and scaled Cholesky factorizations of XUX^T, ZUZ^T, and $K(u)$ and go to Step 1.

At the beginning of this section, we mentioned that after a slight modification, a scaled factorization of XUX^T automatically yields similar factorizations of ZUZ^T and $K(u)$. Indeed, suppose we switch the first k rows and columns of XUX^T with its last ℓ rows and columns. By rearranging (5.2.6), we obtain

$$XUX^T = \begin{bmatrix} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T \end{bmatrix} = \begin{bmatrix} I & 0 \\ -E & I \end{bmatrix} \begin{bmatrix} ZUZ^T & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} I & -E^T \\ 0 & I \end{bmatrix},$$

and hence

$$\begin{bmatrix} ZUZ^T & 0 \\ 0 & K \end{bmatrix} = \begin{bmatrix} I & 0 \\ E & I \end{bmatrix} XUX^T \begin{bmatrix} I & E^T \\ 0 & I \end{bmatrix}.$$

Suppose we have a scaled Cholesky factorization of XUX^T, which we write as

$$XUX^T = \phi^{-1} LL^T = \phi^{-1} \begin{bmatrix} L_Z & 0 \\ L_{KZ} & L_K \end{bmatrix} \begin{bmatrix} L_Z & 0 \\ L_{KZ} & L_K \end{bmatrix}^T;$$

then from the equation above we obtain

$$\begin{bmatrix} ZUZ^T & 0 \\ 0 & K \end{bmatrix} = \phi^{-1} \begin{bmatrix} L_Z & 0 \\ EL_Z + L_{KZ} & L_K \end{bmatrix} \begin{bmatrix} L_Z & 0 \\ EL_Z + L_{KZ} & L_K \end{bmatrix}^T.$$

From this we deduce that $\phi^{-1} L_Z L_Z^T$ and $\phi^{-1} L_K L_K^T$ are scaled Cholesky factorizations of ZUZ^T and K (and that $L_{KZ} = -EL_Z$). Again, we note that, initially and at times of refactorization, the Cholesky factorization of XUX^T is obtained by performing a QR factorization of $U^{1/2}X^T$.

5.3 ▪ Global convergence

In this section we prove global convergence of the FWC Algorithm and the WAC Algorithm (with slight modifications). We also discuss complexity bounds.

Let ϵ_p denote ϵ_+ for the FWC Algorithm and $\max\{\epsilon_+, \epsilon_-\}$ for the WAC Algorithm at the pth iteration. We will assume $\epsilon_p > \epsilon > 0$ for all p and seek a contradiction. In fact, for technical reasons, we decrease ϵ if necessary to 1, so that

$$\sqrt{1+\epsilon} \ge 1 + \frac{\epsilon}{3}.$$

We will show that $\ln\det(K)$ increases at each iteration by an amount related to ϵ and the stepsize λ. However, just as we decreased ϵ if necessary, we find it convenient to

analyze a possibly smaller (in absolute value) stepsize than the actual stepsize λ_p chosen at the pth iteration. To simplify the exposition, let us assume for now that we are considering the FWC Algorithm, so that the stepsize is positive.

Let us choose $0 < \lambda' < 1/k$ so that, for $0 \le \lambda \le \lambda'$,

$$\ln[(1-k\lambda)(1+k\lambda(1-\lambda)(1+\epsilon/3))] \ge k\epsilon\lambda/4 \tag{5.3.1}$$

(which is possible since the derivative of the left-hand side with respect to λ is $k\epsilon/3$ at 0). We set $\lambda'_p := \min\{\lambda_p, \lambda'\}$. Then, because we do not move past the maximum of $\bar{\gamma}$,

$$\Delta\bar{g}_p := \bar{\gamma}(\lambda_p) - \bar{\gamma}(0) \ge \bar{\gamma}(\lambda'_p) - \bar{\gamma}(0). \tag{5.3.2}$$

Moreover, as we remarked before the statement of the algorithms, ω_i moves monotonically towards k as we move towards λ_p, so

$$k \le \omega_i(u(\lambda'_p)) = \frac{(1+\lambda'_p)\omega_i}{(1+\lambda'_p\xi_i)(1+\lambda'_p\zeta_i)} \le \frac{(1+\lambda'_p)\omega_i}{(1+\lambda'_p\zeta_i)^2}, \tag{5.3.3}$$

whence

$$1+\lambda'_p\zeta_i \le \sqrt{(1+\lambda'_p)\frac{\omega_i}{k}} = \sqrt{(1+\lambda'_p)(1+\epsilon_p)} \le (1+\lambda'_p)\sqrt{1+\epsilon_p}. \tag{5.3.4}$$

Combining this with (5.2.11) and (5.3.2), we find

$$\begin{aligned}
\Delta\bar{g}_p &\ge -k\ln(1+\lambda'_p) + \ln\left(1+\tfrac{\lambda'_p\omega_i}{1+\lambda'_p\zeta_i}\right)\\
&= \ln\left((1+\lambda'_p)^{-k}\left[1+\tfrac{k\lambda'_p(1+\epsilon_p)}{1+\lambda'_p\zeta_i}\right]\right)\\
&\ge \ln\left((1-k\lambda'_p)\left[1+\tfrac{k\lambda'_p(1+\epsilon_p)}{(1+\lambda'_p)\sqrt{1+\epsilon_p}}\right]\right)\\
&\ge \ln\left((1-k\lambda'_p)\left[1+k\lambda'_p(1-\lambda'_p)\sqrt{1+\epsilon_p}\right]\right)\\
&\ge \ln\left((1-k\lambda'_p)\left[1+k\lambda'_p(1-\lambda'_p)\sqrt{1+\epsilon}\right]\right)\\
&\ge \ln[(1-k\lambda'_p)(1+k\lambda'_p(1-\lambda'_p)(1+\epsilon/3)]\\
&\ge k\epsilon\lambda'_p/4,
\end{aligned} \tag{5.3.5}$$

where the last inequality uses (5.3.1).

Since $\bar{g}$ is bounded above (e.g., by the value of any feasible solution to (P)), we deduce that $\sum_p \lambda'_p$ converges. This implies that the λ'_p's themselves converge to zero, so that for all large p they equal the λ_p's, whence $\sum_p \lambda_p$ also converges. From this, $\prod_p(1+\lambda_p)$ converges, since its logarithm is $\sum_i \ln(1+\lambda_p)$, which is bounded by $\sum_p \lambda_p$.

Now let ξ_i^p and $\xi_{\max}^p$ denote the values of ξ_i and of the largest ξ_j at iteration p, and let h be the maximizing index for iteration $p+1$. Then, using (3.1.7), we see that

$$\xi_{\max}^{p+1} = \xi_h^{p+1} \le (1+\lambda_p)\xi_h^p \le (1+\lambda_p)\xi_{\max}^p,$$

so that from the previous paragraph, $\xi_{\max}^p$ is bounded above for all p, say by Ξ.

Since λ_p is the optimal stepsize, we have

$$k = \omega_i(u(\lambda_p)) = \frac{(1+\lambda_p)\omega_i}{(1+\lambda_p\xi_i)(1+\lambda_p\zeta_i)} \ge \frac{(1+\lambda_p)\omega_i}{(1+\lambda_p\xi_i)^2},$$

so that

$$1+\lambda_p\xi_i \geq \sqrt{(1+\lambda_p)\frac{\omega_i}{k}} \geq \sqrt{(1+\lambda_p)(1+\epsilon)} \geq 1+\frac{\epsilon}{3}. \tag{5.3.6}$$

But this yields $\Xi\lambda_p \geq \epsilon/3$, which implies that the λ_p's are bounded away from zero, a contradiction. We have proved the following.

Theorem 5.7. *For any positive ϵ, the FWC Algorithm terminates in a finite number of iterations with an ϵ-primal feasible triple (u,K,E).*

Now we consider the WAC Algorithm, which allows decrease and drop steps. For this, we add a further requirement on λ': for $0 \leq \lambda \leq \lambda'$,

$$\ln\left((1+k\lambda)\left[1-\frac{k\lambda}{\sqrt{1-\lambda}}\left(1-\frac{\epsilon}{2}\right)\right]\right) \geq \frac{k\epsilon\lambda}{4}. \tag{5.3.7}$$

(Again, this is possible since the derivative of the left-hand side with respect to λ is $k\epsilon/2$ at 0.) Increase and add steps in the algorithm are analyzed exactly as before. Suppose we now consider a decrease step at the pth iteration, so that the stepsize λ_p is negative. We let $\lambda'_p := \max\{\lambda_p, -\lambda'\}$. Again, we do not move past the maximum of $\bar{\gamma}$, so that

$$\Delta\bar{g}_p := \bar{\gamma}(\lambda_p) - \bar{\gamma}(0) \geq \bar{\gamma}(\lambda'_p) - \bar{\gamma}(0). \tag{5.3.8}$$

As before, ω_j moves monotonically (now upwards) towards k as we move towards λ_p, so

$$k \geq \omega_j(u(\lambda'_p)) = \frac{(1+\lambda'_p)\omega_j}{(1+\lambda'_p\xi_j)(1+\lambda'_p\zeta_j)} \geq \frac{(1+\lambda'_p)\omega_j}{(1+\lambda'_p\zeta_j)^2}, \tag{5.3.9}$$

whence

$$1+\lambda'_p\zeta_j \geq \sqrt{(1+\lambda'_p)\frac{\omega_j}{k}} = \sqrt{(1+\lambda'_p)(1-\epsilon_p)}. \tag{5.3.10}$$

Combining this with (5.2.11) and (5.3.8), we find, in parallel with (5.3.5),

$$\begin{aligned}
\Delta\bar{g}_p &\geq -k\ln(1+\lambda'_p) + \ln\left(1+\tfrac{\lambda'_p\omega_j}{1+\lambda'_p\zeta_j}\right) \\
&= \ln\left((1+\lambda'_p)^{-k}\left[1+\tfrac{k\lambda'_p(1-\epsilon_p)}{1+\lambda'_p\zeta_j}\right]\right) \\
&\geq \ln\left((1-k\lambda'_p)\left[1+\tfrac{k\lambda'_p(1-\epsilon_p)}{\sqrt{(1+\lambda'_p)(1-\epsilon_p)}}\right]\right) \\
&= \ln\left((1-k\lambda'_p)\left[1+\tfrac{k\lambda'_p}{\sqrt{1+\lambda'_p}}\sqrt{1-\epsilon_p}\right]\right) \\
&\geq \ln\left((1-k\lambda'_p)\left[1+\tfrac{k\lambda'_p}{\sqrt{1+\lambda'_p}}\sqrt{1-\epsilon}\right]\right) \\
&\geq \ln\left((1-k\lambda'_p)\left[1+\tfrac{k\lambda'_p}{\sqrt{1+\lambda'_p}}\left(1-\tfrac{\epsilon}{2}\right)\right]\right) \\
&\geq k\epsilon|\lambda'_p|/4,
\end{aligned} \tag{5.3.11}$$

where the last inequality uses (5.3.7), since $0 \leq |\lambda'_p| = -\lambda'_p \leq \lambda'$. We cannot guarantee a particular increase at drop steps, since the decrease of u_j is truncated at such steps, but certainly $\Delta\bar{g}_p \geq 0$ for a drop iteration.

We conclude that, since the increase in $\bar{g}$ is bounded, $\sum_{\mathcal{P}} |\lambda'_p|$ converges, where $\mathcal{P}$ indexes the add, increase, and decrease iterations. As before, this implies that $\sum_{\mathcal{P}} |\lambda_p|$ converges, and hence for any $\gamma \geq 1$, so does $\sum_{\mathcal{P}} \gamma|\lambda_p|$. Thus, as above, we deduce that $\prod_{\mathcal{P}}(1+\gamma|\lambda_p|)$ converges. We want to bound all $\xi^p_{\max}$'s to obtain a contradiction. For add and increase iterations, (3.1.7) shows as above that $\xi^p_{\max}$ increases at most by a factor $1+\lambda_p$, hence at most by a factor $1+\gamma\lambda_p$. For a decrease or drop iteration, (3.1.7) shows (using the Cauchy–Schwarz inequality) that $\xi^p_{\max}$ increases at most by a factor

$$(1+\lambda_p)\left(\xi^p_h - \frac{\lambda_p}{1+\lambda_p\xi^p_j}\xi^p_j\xi^p_h\right)\Big/\xi^p_h = \frac{1+\lambda_p}{1+\lambda_p\xi^p_j}.$$

Thus if for a decrease iteration $(1+\lambda_p)/(1+\lambda_p\xi^p_j)$ is at most $1+\gamma|\lambda_p|$, the increase in $\xi_{\max}$ is covered by the infinite product. We maintain a factor ρ by which the multiplicative increase in $\xi_{\max}$ might exceed the product of the $1+\gamma|\lambda_p|$)'s for add, increase, and decrease iterations so far. To keep ρ small, we also decrease it as much as possible at add, increase, and decrease iterations. If ρ ever exceeds γ, we reject the decrease or drop iteration and instead perform an increase or add step, terminating if necessary. Hence we have the following

Modification to the WAC Algorithm
Initialize $\rho := 1$. At each decrease iteration p, multiply ρ by $(1+\lambda_p)/[(1+\lambda_p\xi^p_j)(1+\gamma|\lambda_p|)]$. At each drop iteration p, multiply ρ by $(1+\lambda_p)/(1+\lambda_p\xi^p_j)$. At each add or increase iteration, multiply ρ by $(1+\lambda_p)/(1+\gamma\lambda_p)$. If the resulting ρ at any iteration exceeds γ, reject the decrease or drop step. If $\epsilon_+ \leq \epsilon$, stop; otherwise perform an increase or add step.

We choose γ large (in our computational experiments, 1,000) to discourage the intrusion of this modification. In any case, the algorithm thus modified generates iterates with all $\xi^p_{\max}$ bounded by the infinite product times γ, say Ξ. As above, this implies that every add or increase iteration has $\lambda_p \geq \epsilon/(3\Xi)$. Suppose the pth iteration is a decrease iteration. Then, because we have an optimal stepsize, we find

$$k = \omega_j(u(\lambda_p)) = \frac{(1+\lambda_p)\omega_j}{(1+\lambda_p\xi_j)(1+\lambda_p\zeta_j)} \leq \frac{(1+\lambda_p)\omega_j}{(1+\lambda_p\xi_j)^2},$$

so that

$$1+\lambda_p\xi_j \leq \sqrt{(1+\lambda_p)\frac{\omega_j}{k}} \leq \sqrt{(1+\lambda_p)(1-\epsilon)} \leq 1-\frac{\epsilon}{2}. \tag{5.3.12}$$

This gives $1-|\lambda_p|\Xi \leq 1-\epsilon/2$, so that $|\lambda_p| \geq \epsilon/(2\Xi)$. Hence all absolute values of stepsizes for add, increase, and decrease iterations are bounded below, contradicting the conclusion above that the sum converges.

Note that, if the modification above is ever invoked to forbid decrease and drop steps, we might terminate because ϵ_+ drops below ϵ; otherwise, both ϵ_+ and ϵ_- must drop below ϵ. We have therefore proved the following.

Theorem 5.8. *For any positive ϵ, the WAC Algorithm modified as above terminates in a finite number of iterations with an ϵ-primal feasible triple (u,K,E). If the modification is never invoked to forbid decrease or drop steps, then it terminates in a finite number of iterations with an ϵ-approximately optimal triple (u,K,E).*

The results above prove global convergence, but do not provide a complexity bound on the number of iterations or arithmetic operations required to achieve a certain accuracy. Indeed, it is hard to obtain such a bound, because there is no easy way to bound the λ_p's, and hence the increase in $\xi^p_{\max}$'s, for the initial iterations where the stepsizes exceed λ'. If we further assume that $\xi^p_{\max}$ is bounded for all p, say by Ξ, then it is possible to prove a bound of the form $C_1 + C_2/\epsilon$ to obtain an ϵ-primal feasible or ϵ-approximately optimal solution, using methods like those for the MVEE case. Here C_1 and C_2 are constants depending on m, n, k, and Ξ.

5.4 ▪ Local convergence

As with the MVEE problem, our aim here is to establish local linear convergence of the algorithm with away steps, but now we need to make strong assumptions to ensure this. Once again the analysis relies on a perturbed problem. We consider

$$(P'(x)) \qquad \begin{array}{ll} \min_{E\in\mathbf{R}^{k\times\ell},H'\in\mathcal{S}^k} & -\ln\det(H') \\ & (y_i+Ez_i)^T H'(y_i+Ez_i)\le k+x_i, \quad i=1,\dots,m. \end{array} \tag{5.4.1}$$

Like (P'), this problem is nonconvex in (E,H'), but is equivalent by exactly the same argument as before to the convex problem

$$(P(x)) \qquad \begin{array}{ll} \min_{H\in\mathcal{S}^n} & -\ln\det(H_{YY}) \\ & x_i^T H x_i \le k+x_i, \quad i=1,\dots,m, \\ & H\succeq 0. \end{array} \tag{5.4.2}$$

We need to work with $(P'(x))$ because it is in the form of a standard nonlinear programming problem and so standard perturbation results apply, whereas $(P(x))$ has an extra positive semidefiniteness constraint.

Suppose we have a triple (u,K,E) that is δ-approximately optimal. We then set

$$x_i := x_i^u := \begin{cases} \delta k & \text{if } u_i = 0, \\ (y_i+Ez_i)^T K^{-1}(y_i+Ez_i) - k & \text{otherwise,} \end{cases}$$

for $i=1,\dots,m$. Note that each component of x is at most δk in absolute value, and that

$$u^T x^u = \sum_{i:u_i>0} u_i x_i^u = \sum_{i:u_i>0} u_i(\omega_i(u)-k) = u^T\omega(u) - ke^T u = k-k = 0, \tag{5.4.3}$$

using (4.3.1). We observe that it is necessary to have a δ-approximately optimal solution for this equation to hold; a δ-primal feasible u will not suffice.

We now define $H'(u) := K^{-1}$ and

$$H(u) = \begin{bmatrix} H'(u) & H'(u)E \\ E^T H'(u) & E^T H'(u) E \end{bmatrix}.$$

Then by definition, $(E,H'(u))$ is feasible in $(P'(x^u))$ and $H(u)$ is feasible in $(P(x^u))$. Indeed, we have

Proposition 5.9. *$(E, H'(u))$ is optimal in $(P'(x^u))$ with Karush–Kuhn–Tucker multipliers u and $H(u)$ is optimal in $(P(x^u))$.*

Proof. We have $H'(u)^{-1} = K = \sum_i u_i(y_i + Ez_i)(y_i + Ez_i)^T$ and $E\ ZUZ^T = -YUZ^T$. Hence the first two Karush–John conditions for $(P'(x^u))$, (4.2.9) and (4.2.10), hold at $(E, H'(u))$ with $\tau = 1$. Moreover, the third condition, (4.2.11), complementarity, holds by the definitions of x^u and $H'(u)$. Then optimality holds as in the unperturbed case (see Corollary 4.7). □

We next bound how far (u, K, E) is from optimality, i.e., how far $\bar{g}(u)$ is from the optimal value $\bar{g}^*$ of (D). For this we again need the relationship between $(P'(x))$ and $(P(x))$, but we also need to make a strong assumption on (P') in order to apply perturbation results. We suppose that the strong second-order sufficient conditions hold at an optimal solution (E^*, H'^*) to (P') with associated Karush–Kuhn–Tucker multipliers u^*. These comprise the second-order sufficient conditions, linear independence of the active constraint gradients, and strict complementarity—see, e.g., [28, 60]. Under this assumption, there are neighborhoods N_1 of 0 and N_2 of (E^*, H'^*, u^*), and a continuously differentiable function from N_1 to N_2, giving the unique Karush–Kuhn–Tucker triple $(E(x), H'(x), u(x))$ of $(P'(x))$ in N_2 (with $(E(0), H'(0), u(0)) = (E^*, H'^*, u^*)$). Moreover, the associated objective function $\phi(x)$ of this solution (called the value function) is also continuously differentiable, and its derivative at 0 is $\phi'(0) = -u^*$ (Theorem 6 of [28]).

Proposition 5.10. *Under the strong second-order sufficient conditions, there is a constant M so that, for every sufficiently small positive δ, every δ-approximately optimal (u, K, E) satisfies*

$$\bar{g}^* - \bar{g}(u) \le M\delta^2. \tag{5.4.4}$$

Proof. Since $(P(x))$ is equivalent to $(P'(x))$, ϕ is also the value function of the former. Moreover, because $(P(x))$ is convex, so is the function ϕ, and since it is continuously differentiable in a neighborhood of 0, $-u^*$ is also a subgradient of ϕ at 0, from which

$$\begin{aligned}\bar{g}(u) = \phi(x^u) &\ge \phi(0) + (-u^*)^T x^u \\ &= \bar{g}^* - (u^* - u)^T x^u \\ &\ge \bar{g}^* - \|u - u^*\| \|x^u\|.\end{aligned}$$

By definition, $\|x^u\| \le \sqrt{m}n\delta$.

Now let us shrink the neighborhood N_1 if necessary so that, for $x \in N_1$, linear independence of the active constraint gradients and strict complementarity hold at $(E(x), H'(x), u(x))$. Suppose δ is sufficiently small that x^u lies in N_1. Then (E, H') must be $(E(x^u), H'(x^u))$. Indeed, if not, there would be two different optimal solutions of $(P'(x^u))$ and hence two different solutions H and $H(x^u)$ of $P(x^u))$. But since the latter problem is convex, any convex combination of these is also an optimal solution, so there are distinct optimal solutions arbitrarily close to $H(x^u)$. These give distinct optimal solutions to $(P'(x^u))$ arbitrarily close to $(E(x^u), H'(x^u))$. Because of strict complementarity and linear independence of active constraint gradients, the associated multipliers must be arbitrarily close to $u(x^u)$. But this contradicts the uniqueness of Karush–Kuhn–Tucker triples in the neighborhood N_2. It now follows again from linear independence and strict complementarity that $u = u(x^u)$.

Since $u(x)$ is continuously differentiable in N_1, we can find a constant L so that

$$||u-u^*|| = ||u(x^u)-u(0)|| \le L||x^u||,$$

and hence (5.4.4) holds with $M := Lmn^2$. □

For the MVEE problem, inequality (5.4.4) sufficed to obtain linear convergence, because we had a strong bound on the increase of g at each iteration. Here we need to make a further assumption: that $\xi_{\max}^p$, the largest component of ξ_i at the pth iteration, is uniformly bounded, say by Ξ. In this case, if we choose ρ sufficiently large, the modification to the WAC algorithm will never be invoked, and so it will converge to a δ-approximately optimal solution for any positive δ.

We need (5.3.1) and (5.3.7) to hold, not just for a particular ϵ, but for all sufficiently small ϵ, say those at most 1. Let us also assume that

$$\lambda \le \frac{\epsilon}{36k} < \frac{1}{k}. \tag{5.4.5}$$

Then (5.3.1) holds if

$$1-k\lambda+k\lambda\left(1+\frac{\epsilon}{3}\right)-k\lambda^2\left(1+\frac{\epsilon}{3}\right)-k^2\lambda^2\left(1+\frac{\epsilon}{3}\right)+k^2\lambda^3\left(1+\frac{\epsilon}{3}\right) \ge \exp\left(\frac{k\epsilon\lambda}{4}\right).$$

The left-hand side is at least $1+k\epsilon\lambda/3-2k^2\lambda^2(1+\epsilon/3) \ge 1+k\epsilon\lambda/3-(8/3)k^2\lambda^2$. The right-hand side is $1+k\epsilon\lambda/4+k^2\lambda^2(\epsilon^2/32+k\lambda\epsilon^3/192+\cdots) \le 1+k\epsilon\lambda/4+k^2\lambda^2/3$. Hence the left-hand side is at least the right-hand side as long as $k\epsilon\lambda/12 \ge 3k^2\lambda^2$, but this follows from (5.4.5).

Now we consider (5.3.7), which holds if

$$(1+k\lambda)\left[1-\frac{k\lambda}{\sqrt{1-\lambda}}\left(1-\frac{\epsilon}{2}\right)\right] \ge \exp\left(\frac{k\epsilon\lambda}{4}\right).$$

For $\lambda < 1/2$, $1/\sqrt{1-\lambda} \le 1+\lambda$, so the left-hand side is at least $1+k\epsilon\lambda/2-(1-\epsilon/2)(k^2\lambda^2+k\lambda^2+k^2\lambda^3) \ge 1+k\epsilon\lambda/2-3k^2\lambda^2$, while as above the right-hand side is at most $1+k\epsilon\lambda/4+k^2\lambda^2/3$, so (5.4.5) also implies that (5.3.7) holds.

We can now show linear convergence under the strong assumptions we have made. We choose $\epsilon' \le 1$ so that every δ-approximately optimal (u,K,E) satisfies (5.4.4) for $\delta \le \epsilon'$. It takes a finite number P of iterations for the WAC algorithm to find an ϵ'-approximately optimal solution. We next bound the number of iterations required to go from an ϵ'-approximately optimal solution to an $\epsilon'/2$-approximately optimal solution, and from an $\epsilon'/2$-approximately optimal solution to an $\epsilon'/4$-approximately optimal solution, and so on.

Suppose we have a δ-approximately optimal solution as our current iterate for $\delta \le \epsilon'$. From now until we obtain a $\delta/2$-approximately optimal solution, we will have $\epsilon_p > \delta/2$. We can choose $\lambda' := \delta/(72k)$ to ensure that (5.3.1) and (5.3.7) will hold for all $0 \le \lambda \le \lambda'$ for $\epsilon = \delta/2$. Then for every add or increase iteration, the analysis in (5.3.5) with ϵ replaced by $\delta/2$ yields

$$\Delta \bar{g}_p \ge k\delta\lambda'_p/8.$$

Moreover, $\lambda' = \delta/(72k)$, and from (5.3.6), $\lambda_p \ge \delta/(6\Xi)$. Thus λ'_p is at least some constant multiple of δ, and hence the increase in $\bar{g}$ is at least some constant multiple of δ^2.

Similarly, for a decrease iteration, the analysis in (5.3.11) yields

$$\Delta \bar{g}_p \geq k\delta|\lambda'_p|/8,$$

and from (5.3.12), $|\lambda_p| \geq \delta/(4\Xi)$. Thus again we have that the decrease in $\bar{g}$ is at least some constant multiple of δ^2. Finally, the number of drop iterations in this phase of the algorithm is bounded by m plus the number of add iterations, and we can conclude from Proposition 5.10 that at most a constant number of iterations are required to go from a δ- to a $\delta/2$-approximately optimal solution.

Now the number of such phases needed to achieve an ϵ-approximately optimal solution is $\log_2(\epsilon^{-1})$, so we have proved the following.

Theorem 5.11. *Under the assumptions*

(a) *the strong second-order sufficient conditions hold at an optimal solution of* (P')*;*

(b) *the quantities* $\xi^p_{\max}$ *are uniformly bounded for all* p*; and*

(c) *the modification of the algorithm is never invoked to forbid decrease or drop steps, the (modified) WAC Algorithm obtains an* ϵ*-approximately optimal solution in at most*

$$P + Q\log_2(\epsilon^{-1})$$

iterations for some data-dependent constants P *and* Q*.*

5.5 ▪ Rank deficiency

In the last three sections, we have assumed throughout that ZUZ^T remains nonsingular, and our algorithms and analysis have used that fact. Here we will discuss the case where ZUZ^T becomes singular. It turns out that this can only happen at drop iterations, and at these a serendipitous condition holds: while many direction matrices E are possible after the iteration, the previous E is always a possibility. This theoretically allows the iterations to continue, but as we shall see, the algorithm can fail.

Example 5.12. Let us consider Example 5.1 again. Recall that $k = 1$,

$$X = \begin{bmatrix} 2 & 3 & 3 \\ 0 & 1 & -1 \end{bmatrix}.$$

Suppose we are at $u := (0.9; 0.1; 0)$, close to the point $(1; 0; 0)$ of nondifferentiability. Then

$$XUX^T = \begin{bmatrix} 4.5 & 0.3 \\ 0.3 & 0.1 \end{bmatrix}.$$

Then $E = -3$ and $K = 4.5 - 3(0.3) = 3.6$, and so

$$\begin{aligned} \omega_1 &= 2(3.6)^{-1}2 = 10/9 > k = 1; \\ \omega_2 &= (3 + (-3)1)(3.6)^{-1}(3 + (-3)1) = 0 < k = 1, \\ \omega_3 &= (3 + (-3)(-1))(3.6)^{-1}(3 + (-3)(-1)) = 10 > k = 1. \end{aligned}$$

Suppose that, instead of increasing u_3 (since $\omega_3 > k$), we choose to decrease u_2 (since $\omega_2 < k$). A simple computation shows that the quadratic determining the stepsize has a repeated root, so we set $\lambda^* = -u_2$ and move to $u = (1; 0; 0)$. At this point,

$$XUX^T = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix},$$

so that $ZUZ^T = 0$ is singular. However, $E = -3$ still satisfies $E\ ZUZ^T = -YUZ^T$, so we will use this. Then $K = 4$, and so

$$\omega_1 = 2(4^{-1})2 = 1 = k;$$
$$\omega_2 = (3+(-3)1)4^{-1}(3+(-3)1) = 0 < k,$$
$$\omega_3 = (3+(-3)(-1))4^{-1}(3+(-3)(-1)) = 9 > k.$$

The algorithm then prescribes that we should increase u_3 and move in the direction $(-1;0;1)$. But recall from Example 5.5 that this is a direction of decrease for $\bar{g}$ and so is the other direction $(-1;1;0)$ that results in a rank-one update. We could decide to take a short step in the direction $(-1;0;1)$ to move away from the point of nondifferentiability, say to $(0.9;0;0.1)$. But then the same problem occurs if we decide to decrease u_3; the computations are almost identical, with the off-diagonal entries of XUX^T, and E, changing sign. ∎

In this example, $k = 1$ for simplicity. We have constructed similar examples for $k = 2$. So far, to obtain cycling, our examples require us to take a nonstandard choice of component, but it seems quite possible that examples can be found where the standard rules of the WAC algorithm lead to cycling.

Let us examine further the case of a drop in the rank of ZUZ^T. The following two results are useful.

Proposition 5.13. *Let $Z = [z_1, z_2, \ldots, z_m] \in \mathrm{R}^{r\times m}$ and let $u \in \mathrm{R}^m$ be nonnegative. Then* $\mathrm{range}(ZUZ^T) = \mathrm{span}\{z_i : u_i > 0\}$ *and so*

$$\mathrm{rank}(ZUZ^T) = \dim \mathrm{span}(\{z_i : u_i > 0\}).$$

Proof. Indeed, it is clear that the range of ZUZ^T is included in the span of those z_i's with u_i positive. But every such z_i lies in the range of ZUZ^T by Proposition 4.15. □

It follows that a drop in the rank of ZUZ^T cannot occur in a decrease iteration. It could occur in an add or an increase iteration if the corresponding τ were 1, so the next u is a unit vector. But then XUX^T and hence K have rank at most one, which is only possible for $k = 1$. Since this is a special case that can be dealt with by linear programming techniques, we ignore it, so that a loss of rank can only occur at drop iterations.

So let the current iterate $\hat{u}$ be such that $K(\hat{u})$ is nonsingular, so that $\hat{u}$ has at least two positive components. Suppose $Z\hat{U}Z^T$ is nonsingular, and let $u := \frac{1}{1-\hat{u}_j}(\hat{u} - \hat{u}_j e_j)$, so that x_j is dropped, and suppose ZUZ^T is singular. Then $0 < \hat{u}_j < 1$. There are many solutions to $YUZ^T = -E\ ZUZ^T$, but one is particularly easy to find.

Proposition 5.14. *Let $\hat{u}$ and u be as above, and suppose $Y\hat{U}Z^T = -\hat{E}\ Z\hat{U}Z^T$. Then we have*

(a) $y_j + \hat{E}z_j = 0$;
(b) $YUZ^T = -\hat{E}\ ZUZ^T$;
(c) $K(u) = \frac{1}{1-\hat{u}_j}K(\hat{u})$; *and*
(d) $\hat{u}_j\zeta_j(\hat{u}) = 1$.

Proof. First note that $\hat{u} = (1-\hat{u}_j)u + \hat{u}_j e_j$, so that $Z\hat{U}Z^T = (1-\hat{u}_j)ZUZ^T + \hat{u}_j z_j z_j^T$. Since ZUZ^T is singular, there is a nonzero w with $ZUZ^T w = 0$, and then

$$0 \neq Z\hat{U}Z^T w = (1-\hat{u}_j)ZUZ^T w + \hat{u}_j z_j z_j^T w = \hat{u}_j (z_j^T w) z_j,$$

so that $z_j^T w$ is nonzero. Also, $w^T ZUZ^T w = 0$, whence $U^{1/2}Z^T w = 0$, and so $YUZ^T w = 0$.

Now we have

$$\begin{aligned}
\hat{u}_j(y_j + \hat{E}z_j)z_j^T w &= \hat{u}_j y_j z_j^T w + \hat{u}_j \hat{E} z_j z_j^T w \\
&= (1-\hat{u}_j)YUZ^T w + \hat{u}_j y_j z_j^T w \\
&\quad + (1-\hat{u}_j)\hat{E}\,ZUZ^T w + \hat{u}_j \hat{E} z_j z_j^T w \\
&= Y\hat{U}Z^T w + \hat{E}\,Z\hat{U}Z^T w = (Y\hat{U}Z^T + \hat{E}\,Z\hat{U}Z^T)w = 0.
\end{aligned}$$

Since $\hat{u}_j$ and $z_j^T w$ are nonzero, part (a) follows.

Next,

$$\begin{aligned}
0 &= Y\hat{U}Z^T + \hat{E}\,Z\hat{U}Z^T \\
&= (1-\hat{u}_j)(YUZ^T + \hat{E}\,ZUZ^T) + \hat{u}_j(y_j z_j^T + \hat{E} z_j z_j^T) \\
&= (1-\hat{u}_j)(YUZ^T + \hat{E}\,ZUZ^T) + \hat{u}_j(y_j + \hat{E}z_j)z_j^T = (1-\hat{u}_j)(YUZ^T + \hat{E}\,ZUZ^T)
\end{aligned}$$

by part (a), and so part (b) is proved.

We can now use the same axis matrix for u as for $\hat{u}$, so that

$$\begin{aligned}
K(u) &= YUY^T + \hat{E}\,ZUY^T \\
&= \frac{1}{1-\hat{u}_j}(Y\hat{U}Y^T + \hat{E}\,Z\hat{U}Y^T - \hat{u}_j(y_j y_j^T + \hat{E} z_j y_j^T)) \\
&= \frac{1}{1-\hat{u}_j}(K(\hat{u}) - \hat{u}_j(y_j + \hat{E}z_j)y_j^T) = \frac{1}{1-\hat{u}_j}K(\hat{u}),
\end{aligned}$$

proving (c).

Finally, we find

$$\begin{aligned}
0 = \det(ZUZ^T) &= (1-\hat{u}_j)^{-r}\det(Z\hat{U}Z^T - \hat{u}_j z_j z_j^T) \\
&= (1-\hat{u}_j)^{-r}\det(Z\hat{U}Z^T)(1-\hat{u}_j\zeta_j(\hat{u}))
\end{aligned}$$

by the rank-one update formula, which shows part (d). Note that in fact, $\hat{u}_j\zeta_j = 1$ is a sufficient condition for dropping x_j to lead to rank deficiency, as well as a necessary condition. □

It is possible that many indices j lead to rank deficiency, and then an extension of the above result shows that all such j can be dropped simultaneously, with analogous properties.

These propositions give us very useful information about rank deficiency. Part (a) might seem to imply that losing rank is exceptional, but in fact this occurs whenever z_j does not lie in the span of the remaining z_i's corresponding to positive $\hat{u}_j$'s and so may

take place with a sparse $\hat{u}$. If it does occur, part (a) shows that ω_j is zero, as observed in our example, and (b) then indicates that the same E can be used after the drop. Parts (b) and (c) show that all quantities ω_i will be scaled up by $1/(1-\hat{u}_j)$ after the drop. Hence we can detect beforehand whether dropping x_j will lead to an ϵ-approximately optimal solution for an acceptably small ϵ.

If not, it may be unwise to proceed with the drop, as it could lead to cycling. In fact, such a step is proscribed by our modification. Note that, since $\xi_j = \zeta_j$, (5.2.9) shows that ξ_j approaches infinity as λ approaches $-\hat{u}_j$ from above by (d). So unless we are going to terminate after the drop, we should choose a different component to adjust.

5.6 ▪ Computational results

Here we will give some results of computational experiments performed with the algorithms of this chapter. We have conducted a number of experiments, but here we confine ourselves to the same set of 5,000 points in 200-dimensional space as in Section 3.8. Our basic method is the WAC algorithm with Kumar–Yıldırım initialization, which doesn't exploit knowledge of the dimension k of y-space, but seems to perform adequately.

If we choose tolerance $\epsilon = 10^{-7}$ and solve all problems with k increasing from 20 to 200 in increments of 20, we see a number of iterations between 1,464 and 2,870 and times between 2.2 and 6.7 seconds, generally decreasing with k. In all cases, linear convergence is apparent, taking hold almost from the first iteration, and the modification to ensure global convergence was never invoked. If we decrease the tolerance to 10^{-10}, the number of iterations increases to between 2,117 and 4,209, and the time to between 3.5 and 9.7 seconds, with no slowing of the linear convergence rate.

For $k = 200$, the algorithm coincides with the WA algorithm for the MVEE problem, so for $\epsilon = 10^{-7}$ we see the same 1,514 iterations, but now taking 2.2 seconds. This is essentially the same as the algorithm for the MVEE problem when the elimination of points is disabled. For the MAEC problem with $k < n$, we have no effective technique for identifying points that can be eliminated; if we try to extend the argument of Harman and Pronzato [41] in Section 3.6, we find that we can still obtain bounds on the eigenvalues of $M := (H_{YY})^{1/2}(H^*_{YY})^{-1}(H_{YY})^{1/2}$, but these do not enable us to bound $\omega_i(u)$ for essential points x_i because of the difference between E and E^*.

Let us focus on the case with $k = 100$. Then with $\epsilon = 10^{-7}$, the algorithm took 1,691 iterations and 3.4 seconds, while for $\epsilon = 10^{-10}$, these increased to 2,374 and 5.0. For the tighter tolerance, there were no drop, 1,066 decrease, 330 add, and 1,178 increase iterations—this shows the effectiveness of the Kumar–Yıldırım scheme in choosing a good set of initial points. Figures 5.1 and 5.2 show the convergence for the looser tolerance, as in Section 3.8. The first depicts $\max \omega_i$ and $\min\{\omega_j : u_j > 0\}$ after the first 25 iterations, while the second shows the linear convergence of the error $\epsilon := \max(\max(\omega_i - k)/k, \max\{(k-\omega_j)/k : u_k > 0\})$, plotted on a log scale.

Now, as in Section 3.8, we apply some variations to the algorithms and note their effects. If we use the Khachiyan initialization, we need 6,850 iterations (4,670 drop, 750 decrease, no add, and 1,430 increase) and 12.3 seconds to achieve $\epsilon = 10^{-7}$. If we use the FWC algorithm with no away steps, we need 28,935 iterations and 50.9 seconds to reach the modest tolerance of 10^{-2}, and 281,107 iterations and 493 seconds to reach $\epsilon = 10^{-3}$ with the Khachiyan initialization. These figures improve to 11,672 and 19.9 (for $\epsilon = 10^{-2}$) and 110,523 and 193 (for $\epsilon = 10^{-3}$) if the Kumar–Yıldırım initialization scheme is used. We observe again the 10-fold increase to obtain a reduction in ϵ by a factor of 10. But

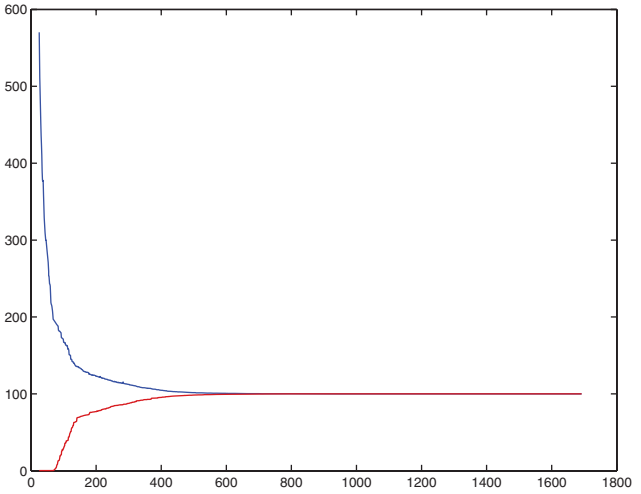


Figure 5.1. *Convergence of* $\max \omega_i$ *(blue) and* $\min\{\omega_j : u_j > 0\}$ *(red).*

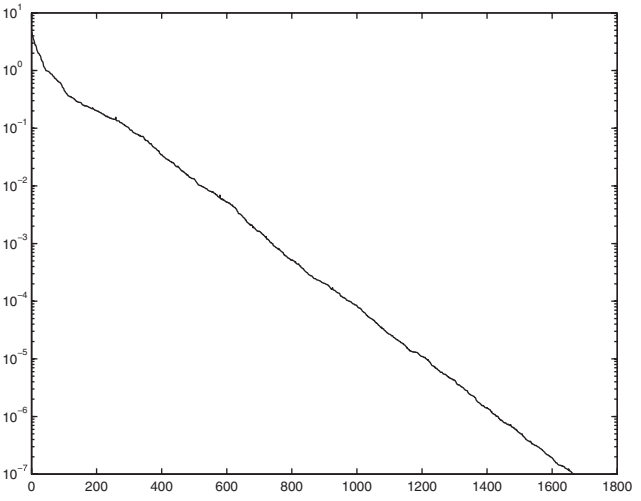


Figure 5.2. *Linear convergence of the error.*

note that the WAC algorithm with the latter initialization only needs 546 iterations and 1.2 seconds, or 763 iterations and 1.5 seconds for these relaxed tolerances.

We conclude that the WAC algorithm is highly effective for the MAEC problem, as is the WA algorithm for the MVEE problem, in spite of its weaker theoretical convergence properties.

5.7 ▪ Notes and references

The algorithms in this chapter, like those of Chapter 3, are based on the Frank–Wolfe method and its variant with away steps. In the case of the MAEC problem, as related to D_k-optimal design in statistics, they were independently developed by Wynn [86] with a fixed stepsize and Fedorov [27] with an optimal stepsize. Atwood [9] pointed out that

Fedorov's convergence analysis failed to treat the case where ZUZ^T is singular at the limiting u, and provided a rigorous global convergence proof, which we have adapted here.

Atwood also introduces away steps in his paper and claims that his convergence proof carries over, without taking into account that the increase of $\xi^p_{\max}$ is not controlled in this case. Our analysis corrects this by a suitable modification of the WAC algorithm.

Our analysis of local convergence is based on that in Ahipaşaoğlu [2] and Ahipaşaoğlu and Todd [4], only slightly simplified. These references also provide a global complexity bound under the strong assumption that $\xi^p_{\max}$ is uniformly bounded. Further, they treat the case of rank deficiency, developing an algorithm whose iterates u may fail to have XUX^T nonsingular while maintaining information allowing the algorithm to proceed, but the example of cycling here is new.

Chapter 6
Related Problems and Algorithms

In this final chapter we discuss three problems related to our main study of minimum-volume containing ellipsoids or minimum-area containing ellipsoidal cylinders.

First, we consider approaches to finding good approximating ellipsoids when some of the points may be contaminated with noise. Two possibilities arise: in one, we seek an ellipsoid that contains a certain fraction of the points, but this is a hard combinatorial problem. Hence we address an alternative formulation due to Gotoh and Takeda [36], which roughly requires that the average of the ellipsoidal distances, among those in the highest $(1-\beta)$ fraction, be no greater than n. This is related to the notion of conditional value-at-risk in finance, as the first approach is related to the traditional concept of value-at-risk. We provide a dual problem and duality and optimality theorems, and discuss the possibility of efficient first-order methods. We also show that the general problem reduces to the centered problem as in the usual minimum-volume ellipsoid problem.

Second, we discuss minimum-volume enclosing parallelotopes, as we hinted at in the opening chapter. Here we do not give algorithms to obtain optimal solutions, but provide some bounds from simple constructions.

Third, we consider a polar problem: given a bounded polyhedron defined by inequalities, we wish to find a maximum-volume inscribed ellipsoid. Although the centered problem reduces to the MVEE problem, the general version seems considerably harder.

6.1 ▪ Conditional minimum-volume ellipsoids

Suppose we are given m points $x_1,\dots,x_m$ in R^n satisfying assumption (2.1.2), as in Chapter 2. Associated with a positive definite $H \in \mathcal{S}^n$, we have the m quantities $p_i := x_i^T H x_i$, $i = 1,\dots,m$. When we consider the MVEE problem, all these p_i's must be at most n. But suppose some of the points are subject to noise. Then it might be reasonable to allow some outlying points to be ignored and to consider the β-minimum-volume ellipsoid (β-MVE) problem

$$\begin{array}{lll} \min_{H\in\mathcal{S}^n, J\subset\{1,\dots,m\}} & -\ln\det(H) & \\ & x_i^T H x_i & \le n \quad \text{for all } i \in J, \\ & |J| & \ge \beta m, \end{array}$$

where $\beta \in (0,1)$ determines the fraction $1-\beta$ of points that can lie outside the ellipsoid $\mathcal{E}(H)$. Clearly, for $\beta > 1 - 1/m$, this problem is equivalent to the MVEE problem. Note that, for the β-MVE problem, we need to strengthen our assumption above and suppose that every collection of at least βm points spans R^n. This problem has been

considered by many statisticians; see, for example, Rousseeuw and Leroy [67]. However, the combinatorial nature of its constraints causes algorithmic difficulties, and the available algorithms either use heuristics or branch-and-bound; see, e.g., Agullo [1].

Instead, we will study here the β-conditional minimum-volume ellipsoid (β-CMVE) problem, defined as

$$(P) \qquad \begin{array}{lll} \min_{H\in\mathcal{S}^n,\alpha\in\mathbf{R},z\in\mathbf{R}^m_+} & f(H) & := -\ln\det(H) \\ & x_i^T H x_i & \le \alpha + z_i, \quad i = 1,2,\ldots,m, \\ & \alpha + \frac{1}{(1-\beta)m} e^T z & \le n \end{array} \tag{6.1.1}$$

for $\beta \in [0,1)$. This problem was introduced by Gotoh and Takeda [36]. The constraints above were modeled on those used by Rockafellar and Uryasev [65] in their discussion of conditional value-at-risk.

To explain these constraints, note first that in any optimal solution, the last constraint must hold with equality; otherwise, α can be strictly increased, and hence H can be multiplied by a factor strictly greater than 1, while maintaining feasibility. But this would improve the objective. Similarly, for any fixed values of the $p_i := x_i^T H x_i$'s in an optimal solution, α and z must be chosen subject to $\alpha e + z \ge p$ to minimize $\alpha + e^T z/[(1-\beta)m]$. Hence each z_i is the nonnegative part of $p_i - \alpha$.

Suppose for now that β is positive, and first that βm is an integer, say k. For simplicity, assume the p_i's are in nondecreasing order. If $\alpha < p_k$, we can increase α by a small amount, say ϵ, and decrease each z_j, $j \ge k$, by ϵ, and maintain feasibility. In so doing, $\alpha + e^T z/[(1-\beta)m]$ increases by $(1-(m-k+1)/(m-k))\epsilon < 0$. Similarly, if $\alpha > p_k$, we can decrease α by some $\epsilon > 0$ while increasing each z_j for $p_j > \alpha$ by ϵ. In so doing, $\alpha + e^T z/[(1-\beta)m]$ increases by at most $(-1+(m-k)/(m-k))\epsilon = 0$. Hence, without loss of generality, $\alpha = p_k$ in an optimal solution. Then z_i is zero for $i \le k$ and equal to $p_i - \alpha$ for $i > k$, so that

$$\alpha + \frac{1}{(1-\beta)m} e^T z = \frac{\sum_{j>k}(\alpha + z_k)}{m-k} = \frac{\sum_{j>k} p_k}{m-k},$$

the average of the values of the p_i's larger than the β-quantile.

If βm is not an integer, we let k denote the ceiling of βm. Similar reasoning to that above shows that again α must be p_k in an optimal solution. We can then write

$$\alpha + \frac{1}{(1-\beta)m} e^T z = \frac{(k-\beta m)\alpha + \sum_{j>k}(\alpha + z_k)}{(1-\beta)m} = \frac{(k-\beta m)\alpha + \sum_{j>k} p_k}{(1-\beta)m},$$

which we can view as an average of the $m-k$ largest p_j's together with a fraction $k-\beta m$ of the $(m-k+1)$st largest. This is a natural way to generalize the average of the p_i's larger than the β-quantile when this does not fall on an integer. If $1 - 1/m \le \beta < 1$, we obtain a bound on the largest p_i; in other words, all points must lie in $\mathcal{E}(H)$ and the problem reduces to the MVEE problem. For smaller values of β, this constraint on the average of the largest values of $x_i^T H x_i$ clearly relaxes this requirement and hence accounts for outliers.

If $\beta = 0$, a similar argument to that above shows that, without loss of generality, α can be taken as the smallest $x_i^T H x_i$, and then $\alpha + \sum z_i/m$ is just the average of these values, which can be no greater than n. As shown by Gotoh and Takeda, the problem is then

easily solved via its optimality conditions, and results in

$$H = \left(\frac{\sum_i x_i x_i^T}{m} \right)^{-1},$$

the inverse of the sample variance of the points.

In the next subsection, we develop the dual of problem (P), then we discuss the generality of just considering the centered version of the problem, and finally we consider algorithms. The development parallels that of the simpler MVEE problem, so we will be brief, but the details are easy to fill in.

6.1.1 ▪ Duality

Let us write γ for $[(1-\beta)m]^{-1}$. We say (H, α, z) is feasible in (P) if it satisfies the constraints and the objective is finite, so that XUX^T is positive definite. We apply Lagrange multipliers $u \in \mathbf{R}^m_+$ and $\lambda \in \mathbf{R}_+$ to the constraints in β-CMVE to obtain

$$\begin{aligned} L(H, \alpha, z, u, \lambda) &:= -\mathrm{lndet}(H) + H \bullet XUX^T - \alpha e^T u - u^T z + \lambda(\alpha + \gamma e^T z - n) \\ &= [-\mathrm{lndet}(H) + H \bullet XUX^T] + [\alpha(\lambda - e^T u)] + [z^T(\lambda\gamma e - u)] - \lambda n. \end{aligned}$$

Minimizing the first term with respect to $H \in \mathcal{S}^n$ gives $\mathrm{lndet}(XUX^T) + n$ (using $H = (XUX^T)^{-1}$) if XUX^T is positive definite, and $-\infty$ otherwise; hence we obtain $\mathrm{lndet}(XUX^T) + n$.

Minimizing the second term with respect to $\alpha \in R$ gives 0 (for $\alpha = 0$, say) if $e^T u = \lambda$, and $-\infty$ otherwise, and similarly minimizing the third term over nonnegative z gives 0 (for $z = 0$) if $u \le \lambda\gamma e$, and $-\infty$ otherwise.

Thus the Lagrangian dual $\max_{u \in \mathbf{R}^m_+, \lambda \in \mathbf{R}_+} \min_{H \in \mathcal{S}^n, \alpha \in \mathbf{R}, z \in \mathbf{R}^m_+} L(H, \alpha, z, u, \lambda)$ reduces to

$$\max_{0 \le u \le \gamma(e^T u)e} \{\mathrm{lndet}(XUX^T) + n - ne^T u\}.$$

Just as in Chapter 2, if we replace u by $\hat{u} := \mu u$, where $\mu \ge 0$ and $e^T u = 1$, we find that the optimal μ is 1, and so we reach the dual problem

$$(D) \qquad \begin{array}{ll} \max & g(u) := \mathrm{lndet}(XUX^T), \\ & e^T u = 1, \\ & u \le \gamma e, \\ & u \ge 0. \end{array}$$

Note that this is exactly the dual of the MVEE problem with the addition of an upper bound of γ on each component of u. If $\beta \ge 1 - 1/m$, γ is at least 1 and these upper bounds are superfluous, while if $\beta = 0$, $\gamma = 1/m$ and the only feasible solution is $u = e/m$. We say u is feasible in (D) if it satisfies the constraints and yields a finite objective value, so that XUX^T is positive definite. We remark that this problem might also arise in optimal statistical design, if it is desired to restrict the proportion of observations made at any individual data point x_i.

We now give a short proof of weak duality to highlight the conditions for optimality.

Proposition 6.1. *If (H, α, z) and u are feasible in (P) and (D), respectively, then*

$$f(H) \ge g(u).$$

Proof. The argument is very close to that for Proposition 2.1. First,

$$\begin{aligned} H \bullet XUX^T &= \textstyle\sum_i u_i x_i^T H x_i \le \sum_i u_i(\alpha + z_i) \\ &= \alpha e^T u + u^T z \le \alpha + \gamma e^T z \le n. \end{aligned}$$

The rest of the proof proceeds exactly as in Chapter 2, using the eigenvalues of $HXUX^T$. □

Note that strong duality holds iff $H = (XUX^T)^{-1}$, u_i positive implies $x_i^T H x_i = \alpha + z_i$, $u_i < \gamma$ implies $z_i = 0$, and $\alpha + \gamma e^T z = n$.

Our next task is to show that strong duality does indeed hold.

Theorem 6.2. *Under assumption* (2.1.2), (P) *has an optimal solution* (H^*, α^*, z^*) *with* H^* *unique,* (D) *has an optimal solution* u^* *with* XU^*X^T *unique, and* $f_* := f(H^*, \alpha^*, z^*) = g(u^*) =: g^*$.

Proof. Consider (P). We have shown above that, if it has an optimal solution, it has one with α nonnegative. Hence we can add the redundant constraint $\alpha \ge -1$. With z nonnegative and the last constraint, this implies that the set of feasible (α, z)'s is compact. Also, $\alpha + z_i$ is bounded above, say by K, for all feasible solutions. Thus all the points x_i lie in the ellipsoid $\mathcal{E}((n/K)H)$. Next, as in Chapter 2, we see that we can add the redundant constraint $-\ln\det(H) \le -\ln\det(\epsilon I)$ for some positive ϵ so that $H = \epsilon I$ is feasible for some α, z. Then the objective function is continuous on this modified feasible region.

By (2.1.2), for each j, μe_j is a convex combination of the points $\pm x_i$ for some positive μ. Hence we obtain a uniform bound on the spectral norm of all feasible H exactly as in the proof of Theorem 2.2. Now we can again apply the Weierstrass theorem to conclude that there is an optimal solution for (P). Moreover, the strict convexity of $-\ln\det$ implies that H^* must be unique.

The objective and constraints are differentiable in the neighborhood of the optimal solution, so we can use the Karush–John optimality conditions at (H^*, α^*, z^*). These imply that there are nonnegative multipliers, not all zero, τ for the objective function, u_i for the ith constraint in (P) for $i = 1, \ldots, m$, and λ for the last constraint, with

$$\begin{aligned} -\tau(H^*)^{-1} + \textstyle\sum_i u_i x_i x_i^T &= 0, \\ \lambda - e^T u &= 0, \\ \lambda\gamma e - u &\ge 0, \\ u_i(x_i^T H^* x_i - \alpha^* - z_i^*) &= 0, \quad i = 1, \ldots, m, \\ \lambda(\alpha^* + \gamma e^T z^* - n) &= 0, \\ z_i^*(\lambda\gamma - u_i) &= 0, \quad i = 1, \ldots . m. \end{aligned} \tag{6.1.2}$$

Note that we have ignored the redundant constraints we added, since they are not tight and the associated multipliers are zero. Also, we have eliminated the multipliers for the nonnegativity constraints on z, instead using the inequalities and the last equations in (6.1.2).

Taking the trace product of the first equation with H^* gives $-\tau n + \sum_i u_i H^* \bullet x_i x_i^T = 0$, so we obtain, using the remaining conditions,

$$\begin{aligned} \tau n &= \sum_i u_i x_i^T H^* x_i \\ &= \alpha^* e^T u + u^T z^* \end{aligned}$$

$$\begin{aligned} &= \alpha^* \lambda + u^T z^* \\ &= \lambda(\alpha^* + \gamma e^T z^*) \\ &= \lambda n. \end{aligned}$$

First, if $\tau = 0$, (u, λ) must be nonzero, so some u_i must be positive, and so λ must be positive, contradicting the equation above. Hence τ must be positive, and by scaling, we can assume that it is 1. Then the equation above gives $\lambda = 1$, and then we see that u is feasible in (D). Moreover, the conditions above imply that equality holds in the weak duality inequality we derived above, so that u is optimal in (D). Since the feasible region of (D) is convex and lndet is concave, XUX^T must be unique. □

6.1.2 ▪ Relaxing the centered restriction

Above we have assumed that we are seeking a centered ellipsoid related to the points x_i in R^n. However, we may be more interested, as were Gotoh and Takeda, in finding a possibly noncentered ellipsoid. Suppose we have m points y_i in R^d whose affine hull is R^d. As in Section 2.3, we define $x_i := (y_i; 1) \in \mathrm{R}^n$ for $i = 1, \ldots, m$, where $n := d + 1$. Then the x_i's span R^n. If we form matrices Y and X from these vectors, we have

$$X = \begin{bmatrix} Y \\ e^T \end{bmatrix}.$$

We would like to choose H_{YY} and $\bar{y}$ so that "$\mathcal{E}(H_{YY}, \bar{y})$ captures the average of the farthest $(1-\beta)m$ points y_i" or more precisely, to minimize $-\mathrm{lndet}(H_{YY})$ subject to $(y_i - \bar{y})^T H_{YY}(y_i - \bar{y}) \le \alpha + z_i$ for all i and $\alpha + \gamma e^T z \le d$ for some α and nonnegative z. The following result shows how these quantities can be obtained by solving the corresponding centered problem for the x_i's.

Theorem 6.3. *With the assumption and notation above, suppose that u^* and (H^*, α^*, z^*) are optimal solutions to (D) and (P), respectively (defined using the points x_i, $i = 1, \ldots, m$), so that $H^* = (XU^*X^T)^{-1}$. Then the unique solution to the noncentered problem above for the y_i's is H^*_{YY} equal to the leading $d \times d$ submatrix of H^* and $\bar{y} = Yu^*$.*

Proof. First consider an arbitrary feasible solution $(H_{YY}, \bar{y})$ to the noncentered problem. Then, for some α and nonnegative z, we have

$$\begin{gathered} (y_i - \bar{y})^T H_{YY}(y_i - \bar{y}) \le \alpha + z_i, \quad i = 1, \ldots, m, \\ \alpha + \gamma e^T z \le d, \end{gathered}$$

or

$$\begin{gathered} \begin{pmatrix} y_i \\ 1 \end{pmatrix}^T \begin{bmatrix} I & 0 \\ -\hat{y}^T & 1 \end{bmatrix} \begin{bmatrix} H_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\hat{y} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix} \le (\alpha + 1) + z_i, \quad i = 1, \ldots, m, \\ (\alpha + 1) + \gamma e^T z \le n. \end{gathered}$$

This shows $(H, \alpha + 1, z)$ is feasible for (P), where

$$\begin{aligned} H &:= \begin{bmatrix} I & 0 \\ -\hat{y}^T & 1 \end{bmatrix} \begin{bmatrix} H_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\hat{y} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} H_{YY} & -H_{YY}\hat{y} \\ -(H_{YY}\hat{y})^T & 1 + \hat{y}^T H_{YY} \hat{y} \end{bmatrix}. \end{aligned}$$

Note that $\det H = \det H_{YY}$, so that $-\mathrm{lndet}(H_{YY}) = -\mathrm{lndet}(H) \ge -\mathrm{lndet}(H^*)$.

Now suppose that (H^*, α^*, z^*) and u^* are optimal for (P) and (D), respectively. Then we know from strong duality that

$$\begin{aligned}
H^* &= (XU^*X^T)^{-1} \\
&= \left(\begin{bmatrix} Y \\ e^T \end{bmatrix} U^* \begin{bmatrix} Y \\ e^T \end{bmatrix}^T \right)^{-1} \\
&= \left(\begin{bmatrix} YU^*Y^T & \bar{y} \\ \bar{y}^T & 1 \end{bmatrix} \right)^{-1} \\
&= \left(\begin{bmatrix} I & \bar{y} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} YU^*Y^T - \bar{y}\bar{y}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ \bar{y}^T & 1 \end{bmatrix} \right)^{-1} \\
&= \begin{bmatrix} I & 0 \\ -\bar{y}^T & 1 \end{bmatrix} \begin{bmatrix} (YU^*Y^T - \bar{y}\bar{y}^T)^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\bar{y} \\ 0 & 1 \end{bmatrix},
\end{aligned}$$

where $\bar{y} := Yu^*$. Let us set $H^*_{YY} := (YU^*Y^T - \bar{y}\bar{y}^T)^{-1}$; we note that this is the leading $d \times d$ principal submatrix of H^* and that $\det H^* = \det H^*_{YY}$ so that $-\text{lndet}(H^*_{YY}) = -\text{lndet}(H^*)$.

Now (H^*, α^*, z^*) is feasible in (P), so

$$\begin{aligned}
\begin{pmatrix} y_i \\ 1 \end{pmatrix}^T \begin{bmatrix} I & 0 \\ -\bar{y}^T & 1 \end{bmatrix} \begin{bmatrix} H^*_{YY} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\bar{y} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix} &\le \alpha^* + z_i^*, \quad i = 1, \ldots, m, \\
\alpha^* + \gamma e^T z^* &\le n,
\end{aligned}$$

or

$$\begin{aligned}
(y_i - \bar{y})^T H^*_{YY} (y_i - \bar{y}) &\le (\alpha^* - 1) + z_i^*, \quad i = 1, \ldots, m, \\
(\alpha^* - 1) + \gamma e^T z^* &\le d,
\end{aligned}$$

so that $(H^*_{YY}, \bar{y})$, with $\alpha^* - 1$ and z^*, satisfies the constraints above. Hence these values are optimal, and uniqueness follows from the same arguments, using the fact that H^*, and hence the last column of its inverse, are unique. □

6.1.3 ▪ Algorithms

Gotoh and Takeda [36] developed an interior-point method for the β-CMVE problem, actually a more general noncentered version where $-\text{lndet}(Q)$ is minimized and $||Qx_i - q||^2$ replaces $x_i^T H x_i$ in the constraints. Their algorithm is based on the dual reduced Newton method of Sun and Freund [77]. They provide encouraging computational results for moderately sized problems. Using our reduction to the centered case should provide a more efficient version of their method.

However, since (D) is a simple modification of the dual of the MVEE problem, it is tempting to devise a coordinate-ascent-like algorithm, as in Section 3.1. There we proposed increasing or decreasing a single component of u and then rescaling the result. Here this leads to problems: our current iterate u will have some components zero, say for $j \in J$; some between 0 and γ, say for indices $k \in K$; and some equal to γ, say for indices $\ell \in L$. We would like to be able to increase a component indexed by $j \in J$, increase or decrease a component indexed by $k \in K$, or decrease a component indexed by $\ell \in L$. After any of these changes, we need to rescale to maintain $e^T u = 1$; but if we scale all the

indices, those in L will no longer be at the upper bound γ, while if we scale just those in K, this will not lead to a simple rescaling of the matrix XUX^T.

Let us instead view such a method as a Frank–Wolfe algorithm, or the variant with away steps. Then at each step we would maximize or minimize a first-order Taylor approximation to the objective over the feasible region, leading to an extreme point. These have the form of a vector with at most one component between 0 and γ, with the rest 0 or γ, and it is easy to find the appropriate extreme point after sorting the components of the gradient vector. However, moving either towards or away from such an extreme point leads to a high-rank correction to XUX^T (which cannot be decomposed into a low-rank correction followed by a scaling), and hence we cannot find an optimal stepsize in closed form or cheaply update the objective and its gradient.

Hence a very simple change to the dual problem makes a drastic change to the efficiency of such a first-order method, and we can therefore only recommend a variant of the interior-point method for solving the β-CMVE problem. We will not provide further details here.

A little work shows that (P) above can also be written as

$$\begin{array}{ll} \min_H & -\ln\det(H) \\ & \max_{u\in Q}(H \bullet XUX^T) \le n, \end{array}$$

where $Q := \{u \in \mathrm{R}^m : e^T u = 1, 0 \le u \le \gamma e\}$. This follows a standard way of viewing conditional value-at-risk as a coherent risk measure (see Artzner et al. [7]). Gotoh and Takeda (private communication) note that the problem above gives a generalization of the minimum-volume enclosing ellipsoid problem for any set Q of probability distributions, with dual problem

$$\max\{\ln\det(XUX^T) : u \in Q\}.$$

However, Frank–Wolfe-type methods will be inefficient for any such problem unless Q is the set of all probability distributions.

6.2 ▪ Approximating by parallelotopes

We next turn to approximating the convex hull of m points x_i in R^n by a small enclosing parallelotope, the affine image of a hypercube or box. Equivalently, we wish to find a nonsingular affine transformation, $x \to Ax + a$, so that the image of each x_i lies in the unit hypercube, and so that the inverse image of this hypercube is small. If we measure size as with ellipsoids by the volume, we are led to the problem

$$(P) \qquad \begin{array}{ll} \min_{A\in\mathbf{R}^{n\times n},\det A>0,a\in\mathbf{R}^n} & -\ln\det A, \\ & \|Ax_i + a\|_\infty \le 1, \quad i = 1,\ldots,m. \end{array}$$

This is because the inverse transformation, $z \to A^{-1}(z - a)$, multiplies volumes by $|\det A^{-1}|$, and we can assume the determinant is positive without loss of generality since the signs of the first row of A and of a can be switched without affecting feasibility. We can also consider the centered variant, where we wish to enclose the points in the linear image of a hypercube centered at the origin—then we merely set a to zero in the above problem. At first sight, this problem seems very similar to the MVEE problem and we might hope that similar duality results, optimality conditions, and efficient algorithms could be developed. However, there is a fundamental difference: on the set of possibly *nonsymmetric* matrices with positive determinant, the function $A \to -\ln\det A$ is not convex. This is easily seen by considering

$$A_1 := \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad A_2 := \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \text{ and } \frac{1}{2}A_1 + \frac{1}{2}A_2,$$

which have determinants 2, 2, and 1, respectively. Thus we give up our hope of obtaining optimal parallelotopes, and instead develop efficient algorithms to obtain good approximations. We measure the quality of approximation by the ratio of circumscribed and inscribed parallelotopes, as we did with ellipsoids. However, our bounds are far from optimal: see the "Notes and references" section.

Let us start with the centered case, where each x_i represents the pair of points $\pm x_i$. We let X be the matrix with columns x_i, and $\mathbf{X}$ be the convex hull of $\pm x_i$, $i = 1, \ldots, m$. We can then apply the BH Algorithm (see Section 3.2) to find n points z_j among the x_i's so that the convex hull of $\pm z_j$, $j = 1, \ldots, n$, has volume at least $1/n!$ times that of the convex hull of $\mathbf{X}$. Note that the former set is a crosspolytope, and is the union of 2^n simplices with vertices the origin and one of z_j and $-z_j$ for each j; each of these simplices has the same volume, $|\det Z|/n!$, where Z is the matrix with columns z_j.

Let us choose a tolerance $\epsilon \in (0, 1]$. We now generate a sequence of nonsingular $n \times n$ submatrices of X, Z_k, $k = 0, 1, \ldots, K$, with $Z_0 := Z$, as follows. Given Z_{k-1}, we look for the largest entry in absolute value of $Z_{k-1}^{-1}X$. If this is no larger than $1+\epsilon$ in absolute value, we terminate. Otherwise, if it occurs in row i and column j, we replace the ith column of Z_{k-1} by x_j to obtain Z_k. We note that the absolute value of the determinant of Z_k is at least $1 + \epsilon$ times that of Z_{k-1}. Analogously, the simplex with vertices the origin and the columns of Z_k, and the corresponding crosspolytope, have volumes that have increased by at least this factor. Since the initial volume is at least $1/n!$ times that of $\mathbf{X}$, the algorithm terminates within $\ln n!/\ln(1+\epsilon) \leq n \ln n/(\epsilon \ln 2)$ steps, each requiring $O(nm)$ arithmetic operations.

Geometrically, at each iteration the columns of $Z_{k-1}^{-1}X$ are the points x_i after the linear transformation represented by Z_{k-1}^{-1}. The current simplex is transformed into the convex hull of the origin and the canonical basis. The condition for continuing the algorithm is that there is a transformed point with a component, say the ith, greater than $1 + \epsilon$ in absolute value, and it is clear that then replacing the ith unit vector with this point will increase the volume of the simplex by at least this factor. If there is no such point, then the transformed set $\mathbf{X}$ lies in the hypercube of side $2(1 + \epsilon)$ centered at the origin. Moreover, the standard crosspolytope, the convex hull of plus or minus the canonical basis vectors, lies within the transformed set $\mathbf{X}$, and hence so does the inscribed hypercube of side $2/n$. We have thus found two concentric hypercubes, of sides $2(1+\epsilon)$ and $2/n$, which circumscribe and inscribe the transformed set $\mathbf{X}$, and transforming these back yields the desired parallelotopes. We have proved

Proposition 6.4. *The algorithm described above for a symmetric $\mathbf{X} \subseteq \mathrm{R}^n$ finds a symmetric parallelotope P such that*

$$\frac{1}{(1+\epsilon)n} P \subseteq \mathbf{X} \subseteq P$$

within $O\left(\frac{n \ln n}{\epsilon}\right)$ iterations, each requiring $O(nm)$ arithmetic operations.

Notice that the estimate above dominates the operations required by the BH Algorithm.

The method above was inspired by an algorithm of Applegate and Kannan [6]. They instead use a method of Lenstra [58] to find a simplex inscribed in $\mathbf{X}$ with vertices drawn

from the columns of X with volume at least half that of the largest such simplex, using the ellipsoid algorithm, with no subsequent iterations. We use a simpler initialization, and then take iterations like those of Eaves [23] to find the largest determinant submatrix, with a relaxed criterion so that we can bound the number of steps.

We next consider the asymmetric case. If we apply the BH Algorithm, we obtain a polytope with at most $2n$ points, but since there may be repetition among the $\overline{z}_j$'s and the $\underline{z}_j$'s, it is not easy to extract a large simplex. Instead, we start by lifting the polytope to a higher dimension as in Section 2.3. Let us therefore change notation, as we did there, to consider a set of m points y_i in R^d and their convex hull $\mathbf{Y}$. We seek concentric parallelotopes that inscribe and circumscribe $\mathbf{Y}$ as above. We again define Y as the $d \times m$ matrix whose columns are the y_i's, and then set $x_i := (y_i; 1) \in \mathrm{R}^n$, $i = 1, \ldots, m$, where $n := d + 1$. We define X and $\mathbf{X}$ as above, so that

$$X = \begin{bmatrix} Y \\ e^T \end{bmatrix}.$$

We now apply the BH Algorithm and the pivoting iterations above to find a submatrix Z $(= Z_K)$ of X so that all entries of $Z^{-1}X$ are at most $1 + \epsilon$ in absolute value. Let the columns of Z be $(w_i; 1)$, $i = 0, \ldots, d$, where the w_i's are columns of Y, and set $W := [w_1 - w_0, \ldots, w_d - w_0]$. Note that the absolute value of the determinant of W is the same as that of Z, and our condition on Z implies that replacing any column by one from X cannot increase this quantity by a factor of more than $1+\epsilon$. Hence replacing any column of W by some $y_i - w_0$ cannot increase the absolute value of its determinant by more than this factor.

Consider the affine transformation $y \to W^{-1}(y - w_0)$ of R^d. This takes the simplex with vertices w_i, $i = 0, \ldots, m$, into the standard simplex with vertices the origin and the canonical basis vectors. Moreover, the condition on W implies that all y_i's, after transformation, lie in the hypercube of side $2(1 + \epsilon)$ centered at the origin. Finally, the hypercube of side $1/d$ centered at the vector with all components $1/(2d)$ is inscribed in the simplex, and hence in $\mathbf{Y}$. Hence the cube of side $2(1+\epsilon)+1/d$ centered at the same point contains $\mathbf{Y}$. Transforming these two hypercubes back gives the two desired concentric parallelotopes inscribing and circumscribing $\mathbf{Y}$. We have proved the following.

Proposition 6.5. *The algorithm described above for a general $\mathbf{Y} \subseteq \mathrm{R}^d$ finds a parallelotope P with center p such that*

$$\frac{1}{2(1+\epsilon)d + 1} P \subseteq \mathbf{Y} \subseteq P$$

within $O\left(\frac{d \ln d}{\epsilon}\right)$ iterations, each requiring $O(dm)$ arithmetic operations.

(Here we use λP to denote the homothetic scaling of P about its center by λ.) Note that the scaling factors here, about n and $2d$ for the symmetric and asymmetric case, are much larger than those for the ellipsoid case, $\sqrt{n}$ and d, respectively.

6.3 ▪ Maximum-volume ellipsoids inscribed in a polyhedron

In this final section, we consider ellipsoids that approximate polyhedral sets defined by linear inequalities, of the form

$$\mathbf{Z} := \{z \in \mathrm{R}^n : y_i^T z \le b_i, \, i = 1, \ldots, m\}.$$

The reason for our somewhat strange notation will become apparent shortly.

Whether we are seeking a minimum-volume circumscribed ellipsoid or a maximum-volume inscribed ellipsoid, we can assume that $\mathbf{Z}$ has a nonempty interior and is bounded. We will assume that we know a point in the interior, and then by translating this point to the origin we can assume all right-hand sides are positive. By scaling, we can assume all these are in fact 1. Thus we have

$$\mathbf{Z} = \{z \in \mathrm{R}^n : y_i^T z \le 1,\ i = 1, \ldots, m\}. \tag{6.3.1}$$

Note that then the polar of $\mathbf{Z}$ is

$$\mathbf{Z}^\circ = \mathbf{Y} := \mathrm{conv}\{y_1, \ldots, y_m\}.$$

Further, since we assume that $\mathbf{Z}$ is bounded, the origin is in the interior of $\mathbf{Z}^\circ$, so it is a convex combination of the y_i's. We also assume without loss of generality that all the y_i's are nonzero, so this is a nontrivial convex combination.

We may seek the minimum-volume ellipsoid containing $\mathbf{Z}$, or the maximum-volume ellipsoid contained in $\mathbf{Z}$. In each case, we may ask for a centered ellipsoid when $\mathbf{Z}$ is symmetric, so that the constraints come in pairs, $\pm y_i^T z \le 1$, or a not-necessarily-centered ellipsoid in the general situation.

Let us consider first the minimum-volume ellipsoid containing $\mathbf{Z}$. Khachiyan and Todd [50] conjecture that this problem is NP-hard, but we may still be able to obtain a reasonable approximation. Indeed, the ellipsoid method (see Sections 1.4 and 3.5) is designed to cut the volume of a circumscribing ellipsoid by a suitable factor at every iteration. We discuss first the noncentered case. Suppose

$$\mathbf{Z} \subseteq \mathcal{E} := \{z \in \mathrm{R}^n : (z - \overline{z})^T H (z - \overline{z}) \le 1\}.$$

If for some i, $y_i^T \overline{z} \ge 1 - (y_i^T H^{-1} y_i)^{1/2}/[(1+\epsilon)n]$, then for all $z \in \mathbf{Z}$,

$$-y_i^T (z - \overline{z}) \ge -1 + y_i^T \overline{z} \ge \alpha (y_i^T H^{-1} y_i)^{1/2}$$

for $\alpha \ge -1/[(1+\epsilon)n]$, and then from, e.g., [80], we can replace $\mathcal{E}$ by another circumscribing ellipsoid whose volume is reduced by at least the factor $\exp(-\epsilon^2/[2(1+\epsilon)^2(n+1)])$. On the other hand, if we cannot find such an i, then

$$\left\{ z \in \mathrm{R}^n : (z - \overline{z})^T H (z - \overline{z}) \le \frac{1}{(1+\epsilon)^2 n^2} \right\} \subseteq \mathbf{Z},$$

and we have found two ellipsoids, respectively inscribed in and circumscribing $\mathbf{Z}$, which are homothetic with ratio $(1+\epsilon)n$.

Let us assume that $\mathbf{Z}$ lies in a ball of radius R. Clearly a ball of radius $1/\max_i \|y_i\|$ is contained in $\mathbf{Z}$. Hence the ellipsoid algorithm will terminate with the desired ellipsoids in at most $2n(n+1)\ln(R \max_i \|y_i\|)(1+\epsilon)^2/\epsilon^2$ iterations.

The centered case is more favorable, as we suggested in Section 3.5. Here we assume each y_i represents $\pm y_i$, so that

$$\mathbf{Z} = \{z \in \mathrm{R}^n : |y_i^T z| \le 1\}.$$

We use centered ellipsoids, so at any iteration we have

$$\mathbf{Z} \subseteq \mathcal{E} := \{z \in \mathrm{R}^n : z^T H z \le 1\}.$$

If for some i, $(y_i^T H^{-1} y_i)^{1/2}/[(1+\epsilon)n^{1/2}] \geq 1$, then for all $z \in \mathbf{Z}$,

$$|y_i^T z| \leq \beta (y_i^T H^{-1} y_i)^{1/2}$$

for $\beta \leq 1/[(1+\epsilon)n^{1/2}]$, and then from, e.g., [80], we can replace $\mathcal{E}$ by another circumscribing ellipsoid whose volume is reduced by at least the factor $\exp(-\epsilon^2/[2(1+\epsilon)^2])$. On the other hand, if we cannot find such an i, then

$$\left\{ z \in \mathrm{R}^n : z^T H z \leq \frac{1}{(1+\epsilon)^2 n} \right\} \subseteq \mathbf{Z},$$

and we have found two ellipsoids, respectively inscribed in and circumscribing $\mathbf{Z}$, which are homothetic with ratio $(1+\epsilon)n^{1/2}$. Now, if $\mathbf{Z}$ lies in a ball of radius R, the algorithm will terminate within $2n \ln(R \max_i \|y_i\|)(1+\epsilon)^2/\epsilon^2$ iterations. We have saved a factor of $n+1$ in the complexity.

We next turn to the construction of maximum-volume inscribed ellipsoids. One motivation for studying this is that such ellipsoids are required at each step in the method of inscribed ellipsoids of Tarasov, Khachiyan, and Erlikh [78], an optimal (in the oracle model) algorithm for convex optimization. Let us consider first the centered case. Then

$$\mathcal{E} \subseteq \mathbf{Z} := \{ z \in \mathrm{R}^n : |y_i^T z| \leq 1 \}$$

iff

$$\mathcal{E}^\circ \supseteq \mathbf{Z}^\circ =: \mathbf{Y} = \mathrm{conv}\{ \pm y_i : i = 1, \ldots, m \}.$$

Moreover, if $\mathcal{E} =: \{ z \in \mathrm{R}^n : z^T H^{-1} z \leq 1 \}$, then $\mathcal{E}^\circ = \{ y \in \mathrm{R}^n : y^T H y \leq 1 \}$, and hence, up to a constant factor, the volume of $\mathcal{E}$ is the reciprocal of that of $\mathcal{E}^\circ$. Thus finding the maximum-volume centered ellipsoid inscribed in $\mathbf{Z}$ is equivalent to finding the minimum-volume centered ellipsoid containing $\mathbf{Y}$, that is, to the MVEE problem, which we know how to solve efficiently.

6.3.1 ▪ Formulations of the general case

Now we consider the noncentered case. Unfortunately, this does not easily reduce to the centered case as with containing ellipsoids. Similarly, although a polarity result like that above still holds, when $\mathcal{E}$ is not centered, the relationship between the volumes of $\mathcal{E}$ and $\mathcal{E}^o$ fails to hold. Thus we are faced with a more complicated problem.

Let us consider $\mathbf{Z}$ as in (6.3.1). This contains the ellipsoid

$$\mathcal{E} := \{ z \in \mathrm{R}^n : (z-v)^T H^{-1} (z-v) \leq 1 \} \tag{6.3.2}$$

iff the support function of the latter in each direction y_i is at most 1, or

$$y_i^T v + \sqrt{y_i^T H y_i} \leq 1, \quad i = 1, \ldots, m.$$

Subject to these constraints, we wish to maximize the volume of $\mathcal{E}$, which amounts to minimizing $-\ln\det(H)$. Unfortunately, the constraints above are not convex in H.

One way to fix this is by replacing the variable H by B^2, for $B \in \mathcal{S}^n$, to get

$$\begin{array}{ll} \min_{v \in \mathrm{R}^n, B \in \mathcal{S}^n} & -2 \ln\det(B) \\ & y_i^T v + \|B y_i\| \leq 1, \quad i = 1, \ldots, m. \end{array} \tag{6.3.3}$$

Alternatively, we can square the constraints above to make them linear in H, obtaining

$$\begin{array}{lrcl} \min_{v\in\mathbf{R}^n, H\in\mathcal{S}^n} & -\mathrm{lndet}(H) & & \\ & y_i^T H y_i - (1-y_i^T v)^2 & \le & 0, \quad i=1,\dots,m, \\ & y_i^T v & \le & 1, \quad i=1,\dots,m. \end{array} \tag{6.3.4}$$

(The last constraints are necessary to avoid the case that some $y_i^T v$ is greater than 1 but the square of the negative number $1-y_i^T v$ is at least $x_i^T H x_i$.) These constraints are now convex (linear) in H, but are nonconvex in v. We therefore consider replacing the concave function $-(1-y_i^T v)^2$ by a convex approximation.

6.3.2 ▪ Successive paraboloid approximations

First, let us approximate this quadratic by its linear approximation around some $\bar{v}\in\operatorname{int} Z$, our current approximation to the center of the maximum-volume inscribed ellipsoid:

$$-(1-y_i^T v)^2 \approx -(1-y_i^T\bar{v})^2 + 2(1-y_i^T\bar{v})y_i^T(v-\bar{v}) = -(1-y_i^T\bar{v})(1-y_i^T(2v-\bar{v})).$$

If we write w for $2v-\bar{v}$, the constraints become

$$y_i^T H y_i \le (1-y_i^T\bar{v})(1-y_i^T w), \quad i=1,\dots,m.$$

These constraints automatically imply that $1-y_i^T w$ is nonnegative, since H is positive definite because of the objective and $y_i^T\bar{v}<1$, and hence with $v=(\bar{v}+w)/2$, $1-y_i^T v>0$. Thus the extra constraints can now be eliminated.

We have therefore replaced the convex problem (6.3.3), or the related nonconvex problem (6.3.4), with an approximating problem:

$$\begin{array}{lrcl} \min_{w\in\mathbf{R}^n, H\in\mathcal{S}^n} & -\mathrm{lndet}(H) & & \\ & y_i^T H y_i & \le (1-y_i^T\bar{v})(1-y_i^T w), & i=1,\dots,m, \end{array} \tag{6.3.5}$$

which is convex and in fact has linear constraints. Khachiyan and Todd [50] show that the original problem can be solved to within an arbitrary accuracy by solving a relatively small number of problems of this form, at each step replacing the approximate center $\bar{v}$ by $(\bar{v}+w)/2$; indeed, in the application to the method of inscribed ellipsoids, at most 12 such subproblems need to be solved if $n\le 10^6$. The authors propose using an interior-point method to solve each subproblem, exploiting its structure and in particular the linearity of its constraints. Bearing in mind the similarity of (6.3.5) to the MVEE problem, we might ask if a simple first-order method based on its dual might be employed.

Let us write $\alpha_i := 1-y_i^T\bar{v}>0$ for each i, and $A := \operatorname{Diag}(\alpha)$. Then, applying a nonnegative multiplier u_i to each constraint, we obtain the Lagrangian

$$L(H,w,u) := -\mathrm{lndet}(H) + \sum_i u_i(y_i^T H y_i + \alpha_i y_i^T w - \alpha_i),$$

leading to the dual problem

$$\max_{u\ge 0}(\min_{H,w}[-\mathrm{lndet}(H) + H\bullet YUY^T + w^T YAu - \alpha^T u]).$$

As in Section 2.1, the first two terms lead to $\mathrm{lndet}(YUY^T)+n$. However, since $w\in\mathbf{R}^n$ is unrestricted, we must add the constraint $YAu=0$, leading to the problem

$$\max\{\mathrm{lndet}(YUY^T)+n-\alpha^T u : YAu=0, u\ge 0\}.$$

As before, if we replace u with $\hat{u} =: \lambda u$, where $\alpha^T u = 1$, we find the optimal value of λ is n, giving as the dual of (6.3.5)

$$\begin{aligned} \max_{u \in \mathbf{R}^m} \quad & \operatorname{lndet}(YUY^T) \\ & YAu = 0, \\ & \alpha^T u = n, \\ & u \geq 0. \end{aligned} \tag{6.3.6}$$

This is similar to the dual of the MVEE problem, except for the presence of the factors α_i, which can easily be absorbed by scaling u and the columns of Y, and the unpleasant appearance of the linear constraints $YAu = 0$. Unfortunately, the latter precludes the use of a simple first-order method that will take advantage of low-rank updates to YUY^T, so it seems unlikely that interior-point methods can be improved on.

Before we move on to the second technique for approximating the nonconvex problem, let us describe a geometric interpretation [50] of (6.3.5). Once again we lift the problem to a higher dimension, but in a different way. Let us embed $\mathbf{Z}$ in the hyperplane $\Pi := \{(z;\zeta) \in \mathrm{R}^{n+1} : \zeta = 1\}$, and then consider the cone on this set with vertex at $(\bar{v};0)$, defined to be

$$\mathbf{K} = \{(z;\zeta) \in \mathrm{R}^{n+1} : y_i^T z \leq y_i^T \bar{v} + \zeta(1 - y_i^T \bar{v}),\ i = 1,\dots,m,\ \zeta \geq 0\}.$$

We next inscribe in this cone a paraboloid tangent to Π at the point $(w;1)$, of the form

$$\mathbf{P} := \left\{(z;\zeta) \in \mathrm{R}^{n+1} : \zeta \geq 1 + \frac{1}{4}(z-w)^T H^{-1}(z-w)\right\}.$$

Note that

$$\begin{aligned} &\max\{(y_i; y_i^T \bar{v} - 1)^T (z;\zeta) : (z;\zeta) \in \mathbf{P}\} \\ &\qquad = \max_z \left\{ y_i^T z + (y_i^T \bar{v} - 1)\left(1 + \tfrac{1}{4}(z-w)^T H^{-1}(z-w)\right)\right\} \end{aligned}$$

is attained when $z = w + \frac{2}{1 - y_i^T \bar{v}} H y_i$, and then it gives the value

$$y_i^T \bar{v} - 1 + y_i^T w + \frac{1}{1 - y_i^T \bar{v}} y_i^T H y_i.$$

This is at most $y_i^T \bar{v}$ exactly when H and w satisfy the constraints of (6.3.5). Thus these constraints provide the conditions for $\mathbf{P}$ to lie inside $\mathbf{K}$.

We measure the size of such a paraboloid by the volume of its intersection with the hyperplane with $\zeta = 5/4$, which is

$$\left\{\left(z; \frac{5}{4}\right) : (z-w)^T H^{-1}(z-w) \leq 1\right\},$$

with volume related to $\operatorname{lndet}(H)$. Thus maximizing this size, subject to the constraints that $\mathbf{P} \subseteq \mathbf{K}$, is exactly our problem (6.3.5).

Since a linear approximation overestimates a concave function, any feasible solution (H, w) to (6.3.5) gives a feasible solution $(H, v := (\bar{v} + w)/2)$ to (6.3.4), and hence an inscribed ellipsoid $\mathcal{E}$. But we can do even better. Let us set $v = (\bar{v} + w)/2$ and $t := (\bar{v} - w)/2$. Then

$$\begin{aligned} (1 - y_i^T \bar{v})(1 - y_i^T w) &= (1 - y_i^T (v + t))(1 - y_i^T (v - t)) \\ &= (1 - y_i^T v)^2 - (y_i^T t)^2, \end{aligned}$$

and so any feasible solution (H, w) to (6.3.5) gives v satisfying $y_i^T v < 1$ for all i and

$$y_i^T(H + t\,t^T)y_i - (1 - y_i^T v)^2 \le 0,\ i = 1,\ldots,m,$$

and so $(H + t\,t^T, v)$ is feasible in (6.3.4), giving a slightly larger inscribed ellipsoid. Of course, as the iterations progress and w becomes closer to $\bar{v}$, t becomes smaller and so the difference between these two ellipsoids grows smaller. It is clear that **K** contains not only **P**, but also the cone with vertex $(\bar{v}; 0)$ on the set **P**; thus **Z** contains the intersection of this cone with Π. Khachiyan and Todd [50] show that this intersection is exactly the slightly larger ellipsoid above, but we will not give details here.

6.3.3 ▪ Successive polar ellipsoid approximations

We now turn to the second approximation of the concave function $-(1 - y_i^T v)^2$ in our problem (6.3.4), now by a strictly convex function. Again, we assume we have some $\bar{v} \in \operatorname{int} \mathbf{Z}$ as an estimate of the center. We start with a linear approximation of the concave function, but then add a quadratic term to get

$$-(1 - y_i^T v)^2 \approx -(1 - y_i^T \bar{v})^2 + 2(1 - y_i^T \bar{v})y_i^T(v - \bar{v}) + (1 - y_i^T \bar{v})^2(v - \bar{v})^T H^{-1}(v - \bar{v}). \tag{6.3.7}$$

This is a worse approximation than the linear approximation alone, but, as we shall see, it results in a more tractable approximating problem. Indeed, if we divide the ith constraint by $(1 - y_i^T \bar{v})^2 > 0$ and write

$$\hat{y}_i := \frac{y_i}{1 - y_i^T \bar{v}}, \quad i = 1,\ldots,m, \tag{6.3.8}$$

the constraints become

$$\hat{y}_i^T H \hat{y}_i - 1 + 2\hat{y}_i^T(v - \bar{v}) + (v - \bar{v})^T H^{-1}(v - \bar{v}) \le 0, \quad i = 1,\ldots,m. \tag{6.3.9}$$

These imply that $2\hat{y}_i^T(v - \bar{v}) \le 1$, or $2y_i^T(v - \bar{v}) \le 1 - y_i^T \bar{v}$, so that $1 - y_i^T v \ge (1 - y_i^T \bar{v})/2 > 0$, and the last constraints are again redundant. Now the constraints above can be written as

$$(\hat{y}_i + H^{-1}(v - \bar{v}))^T H(\hat{y}_i + H^{-1}(v - \bar{v})) \le 1, \quad i = 1,\ldots,m,$$

so that the problem (6.3.4) is approximated by (setting $\bar{y} := -H^{-1}(v - \bar{v})$)

$$\begin{array}{ll} \min_{H \in \mathcal{S}^n, \bar{y} \in \mathbf{R}^n} & -\ln\det(H) \\ & (\hat{y}_i - \bar{y})^T H(\hat{y}_i - \bar{y}) \quad \le \quad 1, \quad i = 1,\ldots,m, \end{array} \tag{6.3.10}$$

a not-necessarily-centered minimum-volume enclosing ellipsoid problem, which we know how to solve efficiently. (Note the right-hand side of 1 rather than n above—appropriate changes need to be made to the algorithms.)

Having solved the problem above, we can use its optimal solution $(H, \bar{y})$ to construct a feasible solution $(H, v := \bar{v} - H\bar{y})$ to (6.3.4), since our approximating function in (6.3.7) was an overestimate. But we can do even better. Having obtained H and $\bar{y}$, we see that they satisfy the approximate constraints in (6.3.9), so that

$$y_i^T H y_i - (1 - y_i^T \bar{v})^2 + 2(1 - y_i^T \bar{v})y_i^T(-H\bar{y}) + (1 - y_i^T \bar{v})^2 \bar{y}^T H \bar{y} \le 0, \quad i = 1,\ldots,m.$$

Collecting terms by their degree in y_i, we obtain

$$y_i^T(H - \bar{v}\bar{v}^T + 2\bar{v}\bar{y}^T H + \bar{y}^T H \bar{y}\bar{v}\bar{v}^T)y_i + 2y_i^T(\bar{v} - H\bar{y} - \bar{y}^T H \bar{y}\bar{v}) - (1 - \bar{y}^T H \bar{y}) \le 0$$

for $i = 1,\dots,m$. Now recall that 0 is a convex combination of the y_i's, and hence, by scaling the weights of the combination, of the $\hat{y}_i$'s. Since at least two of these weights are positive, and the function $g(w) := (w-\bar{y})^T H(w-\bar{y})$ is strictly convex, we see that $\bar{y}^T H\bar{y} < 1$. Dividing the constraints above by $1-\bar{y}^T H\bar{y}$, completing the square, and simplifying, we find

$$y_i^T\left(\frac{1}{1-\bar{y}^T H\bar{y}}H + \frac{1}{(1-\bar{y}^T H\bar{y})^2}H\bar{y}\bar{y}^T H\right)y_i - \left(1-y_i^T\left[\bar{v}-\frac{1}{1-\bar{y}^T H\bar{y}}H\bar{y}\right]\right)^2 \le 0$$

for $i = 1,\dots,m$. Hence

$$H_+ := \frac{1}{1-\bar{y}^T H\bar{y}}H + \frac{1}{(1-\bar{y}^T H\bar{y})^2}H\bar{y}\bar{y}^T H, \qquad v_+ = \bar{v} - \frac{1}{1-\bar{y}^T H\bar{y}}H\bar{y} \tag{6.3.11}$$

satisfy the first set of constraints in (6.3.4). Moreover, (6.3.9) implies that $-2\hat{y}_i^T H\bar{y} + \bar{y}^T H\bar{y} \le 1$, so that $-2y_i^T H\bar{y} \le (1-\bar{y}^T H\bar{y})(1-y_i^T\bar{v})$, and then simple algebra yields $y_i^T v_+ \le 1$. Hence (H_+, v_+) is feasible in the second set of constraints of (6.3.4) as well and thus provides a new inscribed ellipsoid for $\mathbf{Z}$. It is easy to check that $\det H_+ = (1-\bar{y}^T H\bar{y})^{-n}\det H(1+\bar{y}^T H\bar{y}/(1-\bar{y}^T H\bar{y})) = (1-\bar{y}^T H\bar{y})^{-n-1}\det H$, so that we have an improvement over the simple update to (H, v).

This complicated algebraic development has an elegant geometric interpretation. Indeed, let us translate the set $\mathbf{Z}$ to center it at $\bar{v}$, to get

$$\begin{aligned}\hat{\mathbf{Z}} &:= \{z-\bar{v} : z \in \mathbf{Z}\}\\ &= \{\hat{z} : y_i^T\hat{z} \le 1 - y_i^T\bar{v},\ i = 1,\dots,m\}\\ &= \{\hat{z} : \hat{y}_i^T\hat{z} \le 1,\ i = 1,\dots,m\}.\end{aligned}$$

Then we see that $\hat{\mathbf{Z}}^\circ$ is the convex hull of the $\hat{y}_i$'s. Recall that in the centered case, the search for a maximum-volume inscribed ellipsoid in $\mathbf{Z}$ was equivalent to finding the minimum-volume ellipsoid enclosing the convex hull of the y_i's. Now we think that $\bar{v}$ is a good approximation to the center of the maximum-volume ellipsoid inscribed in $\mathbf{Z}$, but since we are no longer confined to centered ellipsoids, we can find the minimum-volume not-necessarily-centered ellipsoid, say $\mathcal{E}(nH, \bar{y})$, enclosing $\hat{\mathbf{Z}}^\circ$. (We use nH here since, when we are dealing with polars, a right-hand side of 1 is more convenient.) Then its polar will be contained in $\hat{\mathbf{Z}}$.

Now $\hat{z}$ lies in $\mathcal{E}^\circ(nH, \bar{y})$ iff $\hat{z}^T\bar{y} + \sqrt{\hat{z}^T H^{-1}\hat{z}} \le 1$, which holds when $\hat{z}^T H^{-1}\hat{z} \le 1 - 2\hat{z}^T\bar{y} + \hat{z}^T\bar{y}\bar{y}^T\hat{z}$ (and $\hat{z}^T\bar{y} \le 1$), or $\hat{z}^T(H^{-1} - \bar{y}\bar{y}^T)\hat{z} + 2\hat{z}^T\bar{y} \le 1$ (and $\hat{z}^T\bar{y} \le 1$), or

$$\left(\hat{z} + \tfrac{H\bar{y}}{1-\bar{y}^T H\bar{y}}\right)^T(H^{-1} - \bar{y}\bar{y}^T)\left(\hat{z} + \tfrac{H\bar{y}}{1-\bar{y}^T H\bar{y}}\right) \;\le\; 1 + \tfrac{\bar{y}^T H\bar{y}}{1-\bar{y}^T H\bar{y}} = \tfrac{1}{1-\bar{y}^T H\bar{y}}$$

(and $\hat{z}^T\bar{y} \le 1$). But this quadratic inequality defines $\mathcal{E}(n(1-\bar{y}^T H\bar{y})(H^{-1}-\bar{y}\bar{y}^T), -H\bar{y}/(1-\bar{y}^T H\bar{y}))$. Using the rank-one formula and translating this back by $\bar{v}$ gives $\mathcal{E}(nH_+^{-1}, v_+)$, as we obtained algebraically above. Also, any $\hat{z}$ satisfying the quadratic inequality has

$$\begin{aligned}\hat{z}^T\bar{y} &\le \left(-\frac{H\bar{y}}{1-\bar{y}^T H\bar{y}}\right)^T\bar{y} + \sqrt{\bar{y}^T[(1-\bar{y}^T H\bar{y})(H^{-1}-\bar{y}\bar{y}^T)]^{-1}\bar{y}}\\ &= -\frac{\bar{y}^T H\bar{y}}{1-\bar{y}^T H\bar{y}} + \sqrt{\bar{y}^T H_+\bar{y}}\\ &= \frac{\sqrt{\bar{y}^T H\bar{y}} - \bar{y}^T H\bar{y}}{1-\bar{y}^T H\bar{y}} \le 1,\end{aligned}$$

since as we saw above, $\bar{y}^T H \bar{y} < 1$. Hence the subsidiary condition in parentheses above is automatically satisfied.

The algorithm based on this second approximation is now clear. Given an approximation $\bar{v}$ to the center of the maximum-volume ellipsoid inscribed in $\mathbf{Z}$, define the $\hat{y}_i$'s by (6.3.8) and (approximately) solve (6.3.10). Then replace $\bar{v}$ with v_+ in (6.3.11) and repeat; the new approximating inscribed ellipsoid is $\mathcal{E}(nH_+^{-1}, v_+)$. This algorithm was suggested in the last section of Khachiyan and Todd [50]. (Note that there is a typo: b_{k+1} (our v_+) should be b_k (our $\bar{v}$) plus the center of the polar of E_k.) When we go from one outer iteration to the next, we can use as a warm start the final u vector from the previous iteration, suitably rescaled so that $\hat{Y}\hat{U}\hat{Y}^T$ remains the same.

The only problem with this algorithm is that there seems to be no good criterion for termination. We could stop when the improvement in $-\ln\det(H_+)$ is small, but this does not guarantee that we are close to an optimal solution. A more principled rule would use a lower bound on the optimal value, and such bounds usually arise from duality. We therefore work with the convex formulation (6.3.3).

Section A.8 shows that a dual problem to this can be written as

$$\begin{aligned} \max_{u\in\mathbf{R}^m,\xi\in\mathbf{R}^m} \quad & \ln\det(YUY^T) \\ & Y\xi = 0, \\ & e^T\xi = n, \\ & \xi_i \geq u_i \|(YUY^T)^{-1/2} y_i\|, \quad i = 1,\ldots,m, \\ & u \geq 0. \end{aligned} \tag{6.3.12}$$

Note the similarity to the dual (6.3.6) of the Khachiyan–Todd approximating problem; now the complicating constraints are $Y\xi = 0$ and the linking of the ξ and u variables. We also observe that this problem—and (6.3.3)—are invariant under a translation of $\mathbf{Z}$ by $\bar{v}$, or equivalently a replacement of the y_i's by the $\hat{y}_i$'s. Indeed, this transformation leads to new variables ($\hat{v} := v - \bar{v}$, $\hat{B} := B$ in the primal and $\hat{u} := ((1 - y_i^T\bar{v})^2 u_i)$, $\hat{\xi} := ((1 - y_i^T\bar{v})\xi_i)$ in the dual) that preserve feasibility and the objective function.

In fact, the derivation of the dual above in Section A.8 (due to Martin Larsson, now at ETH Zurich) obtains a problem where the linking constraints are equalities rather than inequalities. To show that inequalities can also be used, we show that weak duality holds. Indeed, suppose that B, v are feasible in (6.3.3) and u, ξ in (6.3.12). Then, using the primal and dual constraints and the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned} n = e^T\xi &\geq \sum_i (y_i^T v + \|By_i\|)\xi_i \\ &= \sum_i \xi_i \|By_i\| \\ &\geq \sum_i u_i \|(YUY^T)^{-1/2} y_i\| \|By_i\| \\ &\geq \sum_i u_i y_i^T (YUY^T)^{-1/2} By_i \\ &= B \bullet \left(\sum_i y_i u_i y_i^T (YUY^T)^{-1/2} \right) \\ &= B \bullet (YUY^T)^{1/2} = B^{1/2}(YUY^T)^{1/2} B^{1/2} \bullet I. \end{aligned}$$

Next, the concavity of lndet shows that $\text{lndet}(I+M) \le M \bullet I$ for any $M \in \mathcal{S}^n$. Hence

$$\begin{aligned} B^{1/2}(YUY^T)^{1/2}B^{1/2} \bullet I - n &= (B^{1/2}(YUY^T)^{1/2}B^{1/2} - I) \bullet I \\ &\ge \text{lndet}(B^{1/2}(YUY^T)^{1/2}B^{1/2}). \end{aligned}$$

Combining these inequalities yields

$$0 \ge \text{lndet}(B^{1/2}(YUY^T)^{1/2}B^{1/2}) = \text{lndet}(B) + \text{lndet}(YUY^T)^{1/2},$$

showing that $-2\,\text{lndet}(B) \ge \text{lndet}(YUY^T)$, as desired. (We also see from the string of inequalities above that equality can only be achieved if $\xi_i = u_i \|(YUY^T)^{-1/2} y_i\|$ for all i.) There is no duality gap for these problems because a Slater-type condition holds; see, e.g., Theorem 2.165 in Bonnans and Shapiro [16].

The inequality form is helpful in obtaining bounds. Suppose we have (approximately) solved (6.3.10), obtaining dual variables $\hat{u}$. These correspond to variables $u := ((1 - y_i^T \bar{v})^{-2} \hat{u}_i)$, which we may try to use as part of a feasible solution to (6.3.12). We will also have a scaled Cholesky factorization of $YUY^T = \hat{Y}\hat{U}\hat{Y}^T$, which we can obtain from one for $\hat{X}\hat{U}\hat{X}^T$, with

$$\hat{X} := \begin{bmatrix} \hat{Y} \\ e^T \end{bmatrix},$$

with $e \in \mathrm{R}^m$ a vector of 1's as in Section 2.3. From this we can obtain suitable parts of $\hat{H} := (\hat{X}\hat{U}\hat{X}^T)^{-1}$ as in (2.3.2) and (2.3.3). Note that we should set H to be $1/n$ times the leading $n \times n$ submatrix of $\hat{H}$ to reflect the scaling used here, and we therefore multiply $\hat{u}$ by n to correspond. From this we can find v_+ and H_+, or at least its log determinant, which is all we need until termination.

Since we have solved the noncentered MVEE problem inexactly, we need to scale v_+ and H_+ (or its positive definite square root B_+) to ensure they are feasible up to roundoff error, and then we have a feasible objective value for the primal.

For the dual, we take our rescaled $\hat{u}$ (so $e^T\hat{u} = n$), and multiply each component by the appropriate $\|(\hat{Y}\hat{U}\hat{Y}^T)^{-1/2}\hat{y}_i\|$ to get ξ'. This will not satisfy the equality constraints, so we perform an oblique projection, replacing ξ' by

$$\xi := \xi' - \hat{U}\hat{Y}^T(\hat{Y}\hat{U}\hat{Y}^T)^{-1}\hat{Y}\xi'.$$

Note that only nonzero components of ξ' are adjusted. If this vector is nonnegative, we scale it to make it at least $\hat{u}$, and then scale both it and $\hat{u}$ so that $e^T\xi = n$. This then gives us a feasible solution to (the scaled version of) the dual problem (6.3.12).

Why should we hope that this will provide a reasonable lower bound? We expect that as the iterations proceed, v will converge to its optimal value, so that $v_+ - \bar{v}$ and thus $\bar{y}$ will approach zero. Hence H will be close to $(\hat{Y}\hat{U}\hat{Y}^T)^{-1}$ and $-\text{lndet}(H)$ close to $\text{lndet}(\hat{Y}\hat{U}\hat{Y}^T)$. Also, since $\bar{y}$ is small, each $\hat{y}_i(\hat{Y}\hat{U}\hat{Y}^T)^{-1}\hat{y}_i$ for positive $\hat{u}_i$ will be close to 1, so ξ' will be close to $\hat{u}$. Finally, $\hat{Y}\hat{u} = n\bar{y}$ is small, so we can hope that $\hat{Y}\xi'$ will also be small and thus ξ will be close to ξ'. Since the components of $\hat{u}$ sum to n, those of ξ will sum to something close to n and little scaling will be needed.

In practice, our very preliminary computational experience indicates that this method works reasonably well, although it is much slower than solving the MVEE problem. We generated the columns of the data Y from the rotated Cauchy distribution as in Section 3.8, and then translated the origin to make it less central. With $n = 100$ and $m =$

5,000, and asking for the duality gap to be at most 10^{-2}, the method required 1,254 outer iterations and a total of 395,375 inner MVEE iterations, ranging from 827 for the first outer iteration to between 6 and 11 for the last eight, showing the value of the warm start procedure. The time required was 79 seconds. The first lower bound was obtained at the 251st outer iteration, and a good primal solution (within 2×10^{-2} of the best lower bound) was generated at the 431st outer iteration, although it was not known to be good at that time. Convergence of the primal and dual objective values is shown in Figure 6.1 (starting at the 251st iteration), where the slow convergence of the lower bound is apparent.

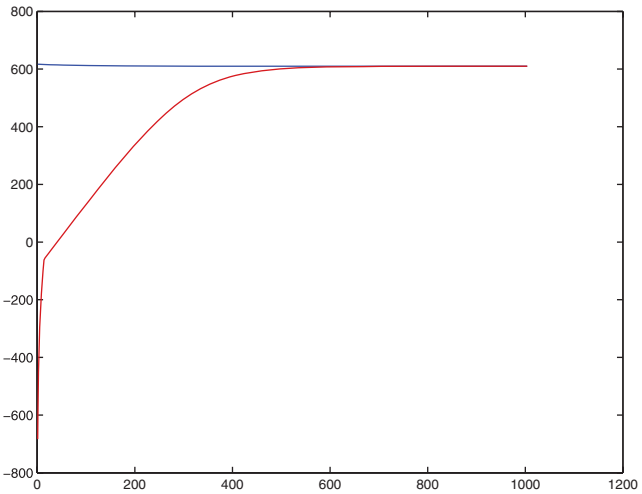


Figure 6.1. *Convergence of feasible primal (blue) and dual (red) objective values.*

For a larger problem with $n = 200$ and $m = 5{,}000$, we needed 2,490 outer iterations and 517 seconds. The total number of inner MVEE iterations was 1,383,704, ranging from 1,310 at the first outer iteration to between 4 and 11 for the last eight. The first good primal objective value was obtained at iteration 444, while the first lower bound was only generated at the 1,139th outer iteration. With a much smaller problem, for $n = 50$ and $m = 200$, only 542 outer and 109,012 inner iterations and 11 seconds were required. For this problem we also decreased the duality gap tolerance to 10^{-3}, at a cost of 638 outer and 153,014 inner iterations and 15 seconds. For all these problems, a good primal feasible solution was obtained early on, but proving it close to optimal by generating a good lower bound took much longer. Thus we have a reasonable algorithm for problems of moderate size and accuracy, but much more work needs to be done to come anywhere close to the efficiency of our algorithms for the MVEE and MAEC problems.

In addition, let us stress that we have no proof of convergence for this method, in contrast to the method using approximating paraboloids.

6.4 ▪ Notes and references

As noted, the approach of Gotoh and Takeda [36] in Section 6.1 is related to the concept of conditional value-at-risk used by Rockafellar and Uryasev [65] in a financial setting. Suppose we have a model of the possible loss to a financial institution as a certain random variable. We could then try to limit the risk by placing an upper bound on a certain quantile of this random variable—this is called the value-at-risk, and is widely used by regulators. However, it has a major drawback: if the threshold is exceeded, there is no

control on the possible loss. As an alternative, the conditional value-at-risk is the expected value of the loss conditional on its exceeding a certain quantile, and hence imposing an upper bound on this does indeed control the amount of catastrophic loss.

It is not clear whether the corresponding notion makes sense as a criterion for choosing an approximating ellipsoid in the case of outliers. If it is thought that the outliers are totally meaningless and chaotic points, then it makes more sense to just require that the ellipsoid contain a certain fraction of the points, as in the β-MVE problem. On the other hand, if the outliers are caused by some gross inaccuracies but are based on real points of a distribution, then limiting the average values of the p_i's exceeding a certain quantile is a reasonable criterion. In either case, the tractability of the β-conditional minimum-volume problem makes it a very attractive model.

We indicated in Section 6.2 that our bounds for approximating parallelotopes were far from tight. Indeed, finding tight bounds in the symmetric case is strongly related to the so-called Banach–Mazur distance to the cube, and is of considerable interest in geometric functional analysis; see, for example, the survey paper of Giannopoulos and Milman [33], especially Section 2 and item 4 on page 766. Bourgain and Szarek [17] provide a lower bound of a constant times $\sqrt{n}\ln n$ for the ratio of circumscribing and inscribed parallelotopes, and suspect that this is tight. Further, Giannopoulos [32] and Youssef [88] have proved an upper bound of a constant times $n^{5/6}$. (Our algorithm only achieves a ratio close to n.) Youssef's argument uses a result of Spielman and Srivastava [75], which is related to the technique of Batson, Spielman, and Srivastava [11] that we discussed in Section 3.7 in connection to spectral sparsification of graphs. Finally, Bourgain and Szarek remark that in the noncentral case, a simplex shows that a lower bound of n for the ratio of circumscribing and inscribed parallelotopes holds, while we attain an upper bound of about $2n$.

Section 6.3 is based on work by Khachiyan and Todd [50], which was motivated by the method of inscribed ellipsoids for convex optimization due to Tarasov, Khachiyan, and Erlikh [78]. This is an alternative to the usual ellipsoid method, generating a sequence of circumscribing polytopes instead of ellipsoids. At each iteration it finds an approximate maximum-volume inscribed ellipsoid so that an oracle can be called at its center. The volumes of the circumscribing polytopes shrink on average much faster than those of the circumscribing ellipsoids in the ellipsoid method, so that fewer oracle calls need to be made; the price paid is that the subproblems are much harder. The algorithms using a sequence of interior-point algorithm calls or of minimum-volume ellipsoid calls are described in [50], but the derivations via algebraic approximations of the nonconvex constraints are new. The possibility of stopping the second algorithm using feasible solutions to the dual is also new. Gürtuna [39] obtains a dual very similar to ours in the more general case of a convex body, giving a semi-infinite programming problem. Our formulation seems easier to use in the case of a polyhedron, and can exploit the solution of the minimum-volume ellipsoid subproblem directly. I derived this dual using second-order cone duality, but this approach gives first a dual involving a nonsymmetric matrix, which can then be manipulated into the form here. The simpler derivation given in Section A.8 is due to Martin Larsson when he was a graduate student at Cornell University; he is now in the Department of Mathematics at ETH Zurich.

It seems fitting to end this book as we began it, with a tribute to Leonid Khachiyan, whose great insights into the power of geometric reasoning in optimization problems, and the importance of geometric optimization problems, were an inspiration to me at several times in my career.

Appendix A
Background Material

A.1 ▪ Notation, inner products, and norms

We use $\mathbf{R}^{m\times n}$ and $\mathcal{S}^n$ to denote the spaces of real $m \times n$ and real symmetric $n \times n$ matrices, respectively. $\mathbf{R}^n$ denotes the space of n-dimensional real vectors, always taken to be columns unless converted to a row vector using a transpose symbol, and $\|\cdot\|$ denotes the Euclidean norm on vectors. Notation similar to that of MATLAB avoids the proliferation of transposes when listing the components of a vector; thus $(x_1;x_2;\ldots;x_n)$ denotes a column vector with the x_j's as components, while $(x_1,x_2,\ldots,x_n)$ denotes the corresponding row vector. We also use $(x_1;x_2;\ldots;x_n)$ to denote the columnwise concatenation of vectors $x_1,x_2,\ldots,x_n$ and similarly $(U_1;U_2;\ldots;U_n)$ for the columnwise concatenation of matrices with the same numbers of columns. We write e_j for the jth unit vector $(0;\ldots;0;1;0;\ldots,0)\in\mathbf{R}^n$, where the 1 is in the jth position, and write e for a vector of 1's, usually in $\mathbf{R}^m$. The ℓ_1 norm of the vector x is $\sum_j |x_j|$. Hence the ℓ_1-unit ball, the set of all vectors with ℓ_1-norm at most 1, is the convex hull of all plus or minus unit coordinate vectors.

We use the natural inner product on matrix spaces $\mathbf{R}^{m\times n}$ and $\mathcal{S}^n$: $U \bullet V$ denotes $\mathrm{Trace}(U^T V)$. The corresponding norm is called the *Frobenius* norm and is denoted $\|U\|_F := (U \bullet U)^{1/2}$. Note that this inner product and norm can be viewed as the usual vector inner product and norm applied to the vectors (of length mn or n^2) obtained by concatenating the columns (or rows) of the matrices. In particular, if u_j is the jth column of a matrix U with n columns, $\|U\|_F = \|(\|u_1\|;\|u_2\|;\ldots;\|u_n\|)\|$.

It is clear that the inner product is symmetric since $U^T V$ and $V^T U$, being transposes of each other, have the same trace. It is important to note also that products can be rearranged in the trace. Suppose U and V are $m \times n$ and $n \times m$, respectively. Then

$$\mathrm{Trace}(UV) = \sum_i \sum_j u_{ij} v_{ji} = \mathrm{Trace}(VU); \tag{A.1.1}$$

we use this frequently. A particular case is

$$H \bullet xx^T = \mathrm{Trace}(Hxx^T) = \mathrm{Trace}(x^T Hx) = x^T Hx$$

for $H \in \mathcal{S}^n$, $x \in \mathbf{R}^n$.

Another matrix norm is the *spectral* or *operator* norm: $\|U\|_2 := \max\{\|Ux\| : \|x\| = 1\}$. It is trivial to see that $\|UV\|_2 \le \|U\|_2\|V\|_2$: the spectral norm is *submultiplicative* and,

in fact, so is the Frobenius norm. For any x of norm 1, if u_i^T is the ith row of $U \in \mathrm{R}^{m\times n}$, then

$$\begin{aligned} ||Ux|| &= ||(u_1^T x; u_2^T x; \ldots; u_m^T x)|| \\ &\le ||(||u_1||; ||u_2||; \ldots; ||u_m||)|| \\ &= ||U||_F, \end{aligned}$$

which shows that $||U||_2 \le ||U||_F$. Similarly, if v_j is the jth column of $V \in \mathrm{R}^{n\times k}$, then

$$\begin{aligned} ||UV||_F &= ||(||Uv_1||; ||Uv_2||; \ldots; ||Uv_k||)|| \\ &\le ||(||U||_2||v_1||; ||U||_2||v_2||; \ldots; ||U||_2||v_k||)|| \\ &= ||U||_2||(||v_1||; ||v_2||; \ldots : ||v_k||)|| \\ &= ||U||_2||V||_F \le ||U||_F||V||_F, \end{aligned}$$

so that the Frobenius norm is also submultiplicative.

We use $\mathrm{Diag}(v) \in \mathcal{S}^n$ to denote the diagonal matrix whose diagonal entries are the components of $v \in \mathrm{R}^n$, and $\mathrm{diag}(U)$ to denote the vector whose components are the diagonal entries of $U \in \mathrm{R}^{n\times n}$.

Consider $U \in \mathrm{R}^{n\times n}$. Then λ is an eigenvalue, and x is an associated eigenvector, if x is nonzero and

$$Ux = \lambda x.$$

It follows that λ is an eigenvalue of U if it is a root of the characteristic equation $\det(\lambda I - U) = 0$. The left-hand side is a polynomial of degree n, and it follows that U has n eigenvalues, counting multiplicity, *if we allow eigenvalues, and their associated eigenvectors, to be complex-valued* (the complex field, as opposed to the reals, is algebraically closed). However, there is a special case, relevant to this monograph, in which n real eigenvalues and a set of n corresponding orthogonal real eigenvectors exist.

Theorem A.1. *If $H \in \mathcal{S}^n$, then there are an orthogonal matrix Q and a diagonal matrix Λ in $\mathrm{R}^{n\times n}$ satisfying*

$$H = Q\Lambda Q^T.$$

If we write $Q = [q_1, q_2, \ldots, q_n]$ and $\Lambda = \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, then it follows that

$$Hq_j = HQe_j = Q\Lambda Q^T Qe_j = Q\Lambda e_j = Q(\lambda_j e_j) = \lambda_j q_j,$$

so that the λ_j's are the eigenvalues, with associated unit eigenvectors q_j's, of H. Accordingly, $Q\Lambda Q^T$ is called the *eigenvalue decomposition* of H. We conventionally list the eigenvalues in nonincreasing order of their absolute values, and in this case write $\Lambda(H) := \Lambda$ and $\lambda(H) := (\lambda_1; \lambda_2; \ldots; \lambda_n)$. It is easy to see that $||H||_F = ||\Lambda(H)||_F = ||\lambda(H)||_2$ and $||H||_2 = ||\Lambda(H)||_2 = |\lambda_1|$.

A.2 ▪ Positive (semi)definiteness

Definition A.2. *A matrix $H \in \mathcal{S}^n$ is called* positive semidefinite *(denoted $H \in \mathcal{S}_+^n$ or $H \succeq 0$) if*

$$x^T Hx \ge 0 \text{ for all } x \in \mathrm{R}^n,$$

and positive definite *(denoted $H \in \mathcal{S}^n_{++}$ or $H \succ 0$) if*

$$x^T H x > 0 \text{ for all nonzero } x \in \mathrm{R}^n.$$

For $A, B \in \mathcal{S}^n$, we write $A \succ B$ or $B \prec A$ ($A \succeq B$ or $B \preceq A$) if $A - B$ is positive (semi)definite.

From the definition, it is clear that it is easy to demonstrate that H is *not* positive semidefinite or positive definite, but it doesn't seem easy to show that it is. The theorem below gives a number of ways to accomplish this. Perhaps the best way to certify positive definiteness is by using the following notion.

Definition A.3. *Given $H \in \mathcal{S}^n$, $H = LL^T$ is a* Cholesky factorization *and L is a* Cholesky factor *of H if L is a lower triangular $n \times n$ matrix with positive diagonal entries.*

Here is the characterization result.

Theorem A.4. *The following are equivalent for $H \in \mathcal{S}^n$:*

(a) *H is positive semidefinite (definite);*

(b) *$x^T H x \geq 0$ for all $x \in \mathrm{R}^n$ ($x^T H x > 0$ for all nonzero $x \in \mathrm{R}^n$);*

(c) *$\lambda(H) \geq 0$ ($\lambda(H) > 0$);*

(d) *$H = JJ^T$ for some $n \times r$ matrix J ($H = JJ^T$ for some nonsingular $n \times n$ matrix J; indeed, J can be taken to be lower triangular with positive diagonal entries).*

Proof. (a) $\Leftrightarrow$ (b) by definition. Suppose $H = Q\Lambda(H)Q^T$. Then for any x, $x^T H x = y^T \Lambda(H) y = \sum_j \lambda_j(H) y_j^2$ for $y = Q^T x$. Noting that x is nonzero iff y is, we see that (b) $\Leftrightarrow$ (c).

Finally, if $H = JJ^T$, then $x^T H x = x^T JJ^T x = \|J^T x\|^2 \geq 0$, and if J is nonsingular and x is nonzero, then $J^T x$ is also nonzero so that its norm is positive. This shows that (d) $\Rightarrow$ (b). Conversely, if H is positive semidefinite (definite), we can write $H = Q\Lambda(H)Q^T$ with $\lambda := \lambda(H) \geq 0$ ($\lambda := \lambda(H) > 0$). Then we can choose $J := Q\Lambda(H)^{1/2}Q^T$, where $\Lambda(H)^{1/2} := \mathrm{Diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_n})$, and it is easy to see that $H = JJ^T$. Moreover, J is nonsingular if H is positive definite, since its inverse can be explicitly obtained as $Q\Lambda(H)^{-1/2}Q^T$, where $\Lambda(H)^{-1/2} := \mathrm{Diag}(1/\sqrt{\lambda_1}, 1/\sqrt{\lambda_2}, \ldots, 1/\sqrt{\lambda_n})$.

It remains to prove that J can be chosen to be a Cholesky factor for a positive definite H. For this we use induction on n, the result being trivial for $n = 1$, since then H is a 1×1 positive definite matrix, whose only entry must be positive, and therefore has a positive square root. (The result is also vacuously true for $n = 0$, but some readers may be unhappy discussing vacuous matrices and vectors.) Suppose the result is true for positive definite matrices of order at most $n-1$, and consider an $n \times n$ positive definite matrix H, which we write in partitioned form as

$$H =: \begin{bmatrix} \gamma & h^T \\ h & \bar{H} \end{bmatrix}.$$

By considering vectors of the form $x = (\xi; 0)$, we see that γ is positive. (Similarly, $\bar{H}$ is positive definite, but we will see that in fact a stronger statement is true, and necessary to

the proof.) Suppose we can write $H = LL^T$ for a lower triangular matrix L with positive diagonal entries. Then, writing L in partitioned form as

$$L =: \begin{bmatrix} \lambda & 0^T \\ l & \bar{L} \end{bmatrix}, \tag{A.2.1}$$

we find

$$\lambda^2 = \gamma, \quad \lambda l = h, \quad \text{and } \bar{L}\bar{L}^T + ll^T = \bar{H}.$$

Conversely, if these equations hold, with λ positive and $\bar{L}$ a lower triangular matrix of order $n-1$ with positive diagonal entries, then L as in (A.2.1) is a Cholesky factor of H. The first two equations can be satisfied by setting $\lambda := \sqrt{\gamma}$ and $l := (1/\lambda)h = (1/\sqrt{\gamma})h$. Then the last equation is satisfied as long as $\bar{L}\bar{L}^T = \bar{H} - ll^T = \bar{H} - (1/\gamma)hh^T$. So we can use the inductive hypothesis to complete the proof as long as we can show that the latter matrix is itself positive definite. But using the definitions of λ and h, we obtain the factorization

$$\begin{bmatrix} \gamma & h^T \\ h & \bar{H} \end{bmatrix} = \begin{bmatrix} \lambda & 0^T \\ l & I \end{bmatrix} \begin{bmatrix} 1 & 0^T \\ 0 & \bar{H} - ll^T \end{bmatrix} \begin{bmatrix} \lambda & l^T \\ 0 & I \end{bmatrix}.$$

Since the matrix on the left-hand side is positive definite, so is the matrix in the middle on the right-hand side, and hence by considering vectors of the form $(0; \bar{y})$, we get that $\bar{H} - ll^T$ is positive definite, and we are done. ∎

We can derive a few consequences.

Corollary A.5. *The Cholesky factorization of a positive definite matrix is unique.*

Proof. Indeed, the proof above shows that the first column of the Cholesky factor is uniquely defined. Proceeding inductively shows that the whole matrix is unique. ∎

Corollary A.6. *Every positive semidefinite matrix H has a positive semidefinite square root $H^{1/2}$ satisfying $H^{1/2}H^{1/2} = H$. Every positive definite matrix has a positive definite inverse.*

Proof. If $H = Q\Lambda Q^T$ is positive semidefinite, we know that the diagonal entries of Λ are nonnegative, and so have nonnegative square roots, which we can arrange into the diagonal matrix $\Lambda^{1/2}$. Then $H^{1/2} := Q\Lambda^{1/2}Q^T$ establishes the first part. If H is positive definite, the diagonal entries of Λ are positive, and so have positive reciprocals, forming the diagonal matrix Λ^{-1}. Then $H^{-1} = Q\Lambda^{-1}Q^T$ has positive eigenvalues and so is positive definite. ∎

In fact, the positive semidefinite square root is unique, but we will not prove this here. Note the progression of extensions of the square root function from scalars to diagonal matrices to symmetric matrices; this technique can also be used for other functions, such as the inverse (which then coincides with the regular inverse) on nonzero scalars and nonsingular diagonal matrices or symmetric matrices, the exponential, and the logarithm on positive scalars and positive definite diagonal or symmetric matrices.

If H is positive definite, so is its positive semidefinite square root, and so it has an inverse, denoted $H^{-1/2}$. It is easy to see that this is also the positive semidefinite square root of H^{-1}.

Positive semidefinite square roots are very useful in proofs. For example, we have the following.

Corollary A.7. *If $A, B \in \mathcal{S}^n$ are positive definite, then $A \succeq B$ iff $B^{-1} \succeq A^{-1}$.*

Proof. $A \succeq B$ is equivalent to $I \succeq A^{-1/2}BA^{-1/2}$. But this just means that all eigenvalues of the positive definite matrix $A^{-1/2}BA^{-1/2}$ are positive and at most 1, which holds iff all the eigenvalues of its inverse, $A^{1/2}B^{-1}A^{1/2}$, are at least 1, or $A^{1/2}B^{-1}A^{1/2} \succeq I$. But this is equivalent to $B^{-1} \succeq A^{-1}$. □

For the next corollary, a *principal rearrangement* of a matrix H is obtained by reordering its rows and columns correspondingly: it is of the form PHP^T, where P is a permutation matrix, that is a 0–1 matrix with exactly one 1 in every row and in every column. A *principal submatrix* of a matrix is obtained by choosing a subset of its rows and the correspondingly indexed subset of columns. A *leading* principal submatrix is one corresponding to a subset $\{1, 2, \ldots, k\}$ for $1 \le k \le n$: it occupies the top left-hand corner of H. It is immediate that a principal rearrangement or a principal submatrix of a positive (semi)definite matrix is positive (semi)definite. A (leading) principal minor is the determinant of a (leading) principal submatrix.

Corollary A.8. *Every principal minor of a positive definite matrix is positive. Conversely, if every leading principal minor of a symmetric matrix is positive, it is positive definite.*

Proof. Since every principal minor is a leading principal minor of a principal rearrangement, it is enough to show that H is positive definite iff every leading principal minor of H is positive. For the "only if" part, suppose $H = LL^T$ is the Cholesky factorization of a positive definite matrix H, and let H_{11} and L_{11} be the leading $k \times k$ principal submatrices of H and L. Then it is easy to see that $H_{11} = L_{11}L_{11}^T$, so that its determinant is the square of that of L_{11}, and is hence positive. For the "if" part, suppose all leading principal minors of H are positive, and suppose we have completed the Cholesky factorization of H through the kth column, so that

$$H =: \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \bar{H}_{22} \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & I \end{bmatrix},$$

where all partitions are into the first k rows and columns and the last $n-k$, and where all diagonal entries of L_{11} are positive. Then the leading $k \times k$ principal minor of H, $\det H_{11}$, is equal to the determinant of the leading $k \times k$ principal minor of the right-hand side, which is $(\det L_{11})^2$ and positive. Now let us consider the leading $(k+1) \times (k+1)$ principal minors. For the left-hand side, this is a leading principal minor of H, and is hence positive. For the right-hand side, because of the triangularity of the first and last matrices, it is the product of the leading $(k+1) \times (k+1)$ principal minors of the three matrices on the right-hand side. For the first and third matrices, this is $\det L_{11}$ again, while for the middle matrix, it is exactly the top left-hand entry of $\bar{H}_{22}$. We can conclude that this entry is positive, and this allows us to continue the Cholesky factorization one more step, as in the proof of the existence of the Cholesky factorization. Continuing in this way, we see that positivity of the leading principal minors implies the existence of a Cholesky factorization, and hence positive definiteness. □

It turns out that positive semidefiniteness implies that all principal minors are nonnegative and conversely, but nonnegativity of just the leading principal minors is not sufficient, as shown by the matrix $\mathrm{Diag}(0, -1)$.

Computing the (leading) principal minors of a matrix is a reasonable way to test the positive definitiveness or semidefiniteness of a very small matrix, but it becomes less efficient (especially if all 2^n principal minors need to be calculated!) as n grows. Computing, or trying to compute, the Cholesky factorization is a more efficient technique, taking $O(n^3)$ floating-point operations, and is also numerically stable. Moreover, it is not hard to see that, if the factorization breaks down at any stage, it is straightforward to find a nonzero vector x with $x^T H x \leq 0$. The Cholesky factorization has other benefits also. We saw above that positive definite matrices H are nonsingular: the Cholesky factorization allows us to cheaply (in $O(n^2)$ arithmetic operations) compute the solution to systems $Hx = b$, since we can solve the lower triangular system $Ly = b$ easily, obtaining the components of y in the order $y_1, y_2, \ldots, y_n$, and then similarly solve the upper triangular system $L^T x = y$, finding the components of x in the order $x_n, x_{n-1}, \ldots, x_1$. We now have $Hx = LL^T x = Ly = b$, as desired.

The well-known Cauchy–Schwarz inequality states that, for two vectors x and y in R^n,

$$|y^T x| \leq ||x||||y||.$$

The existence of positive semidefinite square roots allows a very useful generalization of the Cauchy–Schwarz inequality. Often it is useful to think of a copy of R^n called the dual space, and a scalar product defined on two vectors, one from the dual space and one from the original space. For example, the original space might contain the arguments of a real-valued function, and then the derivative of this function would be regarded as a vector in the dual space. In this case, if we define a norm in the original space using a positive definite matrix $H \in \mathcal{S}^n$ by

$$||x||_H := \sqrt{x^T H x},$$

it is natural to define the norm in the dual space

$$||y||_H^* := \sqrt{y^T H^{-1} y}.$$

Note that, if $H^{1/2}$ is the positive definite square root of H and $H^{-1/2}$ is its inverse (or alternatively the positive definite square root of H^{-1}), then $||x||_H = ||H^{1/2}x||$ and $||y||_H^* = ||H^{-1/2}y||$, where the norms on the right are Euclidean. In this case, the generalized Cauchy–Schwarz inequality states that

$$|y^T x| \leq ||x||_H ||y||_H^*.$$

The proof of this is very straightforward from expanding $||H^{1/2}x + \lambda H^{-1/2}y||^2$ as a quadratic in λ and writing the condition for this quadratic to be nonnegative everywhere. It is then simple to see that the norm in the dual space is in fact the norm dual to the norm in the original space:

$$||y||_H^* = \max\{y^T x : ||x||_H \leq 1\}.$$

Similar dual norms of symmetric matrices can be defined: for any nonsingular $n \times n$ matrix M we can define $||X||_M := ||MXM^T||_F$ and $||Y||_M^* := ||M^{-T} Y M^{-1}||_F$ for X in $\mathcal{S}^n$ and Y in its dual space.

It is clear that the set of positive (semi)definite matrices forms a cone in $\mathcal{S}^n$: if U lies in either set, so does λU for any positive λ. In fact, the cone $\mathcal{S}_+^n$ is self-dual, i.e., it equals its dual cone

$$(\mathcal{S}_+^n)^* := \{U \in \mathcal{S}^n : U \bullet V \geq 0 \text{ for all } V \in \mathcal{S}_+^n\}.$$

To show that it is contained in its dual cone, let U and V be positive semidefinite. Then

$$U \bullet V = \mathrm{Trace}(U^{1/2} U^{1/2} V^{1/2} V^{1/2}) = \mathrm{Trace}(V^{1/2} U^{1/2} U^{1/2} V^{1/2}) = ||U^{1/2} V^{1/2}||_F^2 \geq 0,$$

where we used (A.1.1). For the reverse inclusion, note that Theorem A.4 implies that xx^T is positive semidefinite for any $x \in \mathrm{R}^n$. Thus any U in the dual cone has $x^T U x = \mathrm{Trace}(x^T U x) = \mathrm{Trace}(U x x^T) = U \bullet xx^T \geq 0$ for any x, so is positive semidefinite. Finally, note that if U and V are positive semidefinite and $U \bullet V$ is zero, then the equation above shows that $U^{1/2}V^{1/2}$ is zero. In particular, if U is positive definite, so that U and $U^{1/2}$ are nonsingular, then $V^{1/2}$ and hence V must be zero. Similarly, if V is positive definite, U must be zero. Also, $U^{1/2}V^{1/2} = 0$ implies $UV = 0$.

A.3 ▪ Schur complements and low-rank updates

Suppose we partition the matrix $M \in \mathrm{R}^{(n+k)\times(n+k)}$ as

$$M =: \begin{bmatrix} A & B \\ C^T & D \end{bmatrix}, \tag{A.3.1}$$

where A is $n \times n$, B and C are $n \times k$, and D is $k \times k$. If A is nonsingular, row operations can then eliminate the submatrix C^T, which amounts to factorizing M as

$$M = \begin{bmatrix} I & 0 \\ C^T A^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ 0 & D - C^T A^{-1} B \end{bmatrix}.$$

We call $D - C^T A^{-1} B$ the *Schur complement* of A in M. Note that $\bar{H} - (1/\gamma) h h^T$ in the proof of Theorem A.4 is the Schur complement of γ in H. Similarly, if D is nonsingular, we obtain

$$M = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - BD^{-1}C^T & 0 \\ C^T & D \end{bmatrix}.$$

(Note that $A - BD^{-1}C^T$ is the Schur complement of D in M.) We can use these expressions to relate the nonsingularity of M, $A - BD^{-1}C^T$, and $D - C^T A^{-1} B$ and obtain formulae for the inverses of the latter matrices. Usually we think of n as large and k as much smaller, so that $A - BD^{-1}C^T$ is a low-rank modification of A, and (A.3.2) below gives a formula for its inverse in terms of that of the original matrix A and the much smaller matrix $D - C^T A^{-1} B$.

Theorem A.9. *Suppose the matrix M is partitioned as in* (A.3.1). *If A is nonsingular,* $\det M = \det A \det(D - C^T A^{-1} B)$ *and* (a) *and* (b) *below are equivalent. If D is nonsingular,* $\det M = \det D \det(A - BD^{-1} C^T)$ *and* (a) *and* (c) *are equivalent.*

(a) *M is nonsingular.*

(b) *$D - C^T A^{-1} B$ is nonsingular.*

(c) *$A - BD^{-1} C^T$ is nonsingular.*

Finally, if both A and D are nonsingular, then (b) *and* (c) *are equivalent, and if these hold,*

$$(A - BD^{-1}C^T)^{-1} = A^{-1} + A^{-1}B(D - C^T A^{-1} B)^{-1} C^T A^{-1}, \tag{A.3.2}$$

and similarly $(D - C^T A^{-1} B)^{-1} = D^{-1} + D^{-1} C^T (A - BD^{-1} C^T)^{-1} BD^{-1}$.

Proof. All the statements follow immediately from the two equations above the theorem except the last one. For this, we take the inverses of these two equations to get

$$\begin{bmatrix} A-BD^{-1}C^T & 0 \\ C^T & D \end{bmatrix}^{-1} \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} A & B \\ 0 & D-C^TA^{-1}B \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ C^TA^{-1} & I \end{bmatrix}^{-1}.$$

This gives

$$\begin{bmatrix} (A-BD^{-1}C^T)^{-1} & 0 \\ -D^{-1}C^T(A-BD^{-1}C^T)^{-1} & D^{-1} \end{bmatrix} = \begin{bmatrix} A^{-1} & -A^{-1}B(D-C^TA^{-1}B)^{-1} \\ 0 & (D-C^TA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^TA^{-1} & I \end{bmatrix} \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix},$$

and evaluating the top left-hand corner of both sides gives (A.3.2). The other equation is established similarly. ∎

Applying the result with $B = u$, $C = v$, and $D = -1$, we obtain the rank-one update formulae.

Corollary A.10. *Suppose $A \in \mathrm{R}^{n\times n}$ is nonsingular and $u, v \in \mathrm{R}^n$. Then* $\det(A + uv^T) = (1 + v^TA^{-1}u)\det A$. *Moreover, $A + uv^T$ is nonsingular iff $1 + v^TA^{-1}u$ is nonzero, in which case*

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{1 + v^TA^{-1}u} A^{-1}uv^TA^{-1}.$$

Low-rank update formulae are frequently attributed to Sherman, Morrison, and Woodbury, but in fact they were discovered earlier and have been rediscovered many times since; see the survey article of Hager [40].

We now establish related results for symmetric matrices. Suppose we partition the symmetric matrix $M \in \mathrm{R}^{(n+k)\times(n+k)}$ as

$$M =: \begin{bmatrix} H & V \\ V^T & D \end{bmatrix}, \tag{A.3.3}$$

where H is $n \times n$ and symmetric, V is $n \times k$, and D is $k \times k$ and symmetric. Then we can perform both row and column operations to zero out the off-diagonal blocks in two ways. If D is positive definite,

$$M = \begin{bmatrix} I & VD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} H-VD^{-1}V^T & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}V^T & I \end{bmatrix},$$

while if H is positive definite,

$$M = \begin{bmatrix} I & 0 \\ V^TH^{-1} & I \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & D-V^TH^{-1}V \end{bmatrix} \begin{bmatrix} I & H^{-1}V \\ 0 & I \end{bmatrix}.$$

Since the matrices on the left and right in the products are nonsingular and transposes of one another, we can conclude the positive (semi)definiteness of larger matrices from that of smaller ones.

Theorem A.11. *Suppose M is given by* (A.3.3). *Then we have the following.*

(a) *If D is positive definite, M is positive (semi)definite exactly when $H - VD^{-1}V^T$ is.*

(b) *If H is positive definite, M is positive (semi)definite exactly when $D - V^T H^{-1} V$ is.*

(c) *If both D and H are positive definite, $H - VD^{-1}V^T$ is positive (semi)definite exactly when $D - V^T H^{-1} V$ is.*

We note that a Cholesky factorization of H can be updated to one of $H + \delta u u^T$ (with $1 + \delta u^T H^{-1} u > 0$ if δ is negative) in $O(n^2)$ arithmetic operations; see Gill et al. [34].

A.4 ▪ Matrix analysis

Here we discuss the differentiability of some useful functions of matrices. First, we note that occasionally we deal with matrix functions of scalars. A function $F(\xi) = (f_{ij}(\xi))$ of a scalar ξ is continuously differentiable if each entry f_{ij} is, and then its derivative is $F'(\xi) = (f'_{ij}(\xi))$, and similarly for twice continuous differentiability.

Now we turn to real-valued functions of matrices. A function of an $n \times n$ matrix is continuously differentiable if it is so when regarded as a function of its n^2 entries, or in the case of a function of symmetric matrices, of the $n(n+1)/2$ entries in its lower triangle, and similarly for twice continuous differentiability.

The directional derivative of a function of a matrix A in the direction of the matrix E will then be linear in the matrix E, and hence can be written as $M \bullet E$ for some matrix M, and we call M the *gradient* of f at A and denote it $\nabla f(A)$:

$$\frac{d}{d\alpha} f(A + \alpha E)|_{\alpha=0} =: \nabla f(A) \bullet E.$$

If the function's argument is a symmetric matrix, then the direction matrix E will be symmetric, and since any linear function of a symmetric matrix E can be written as $M \bullet E$ for a symmetric M, we take the gradient to also be a symmetric matrix.

Our prime example is the *logdet* function, defined on symmetric matrices by

$$\operatorname{lndet}(H) = \begin{cases} \ln\det(H) & \text{if } H \text{ is positive definite,} \\ -\infty & \text{otherwise.} \end{cases}$$

This is clearly infinitely differentiable anywhere it is finite. We will see that its derivatives are very closely related to those of the one-dimensional function $\ln \xi$, whose first and second derivatives are ξ^{-1} and $-\xi^{-2}$ for positive ξ. To find its gradient we proceed as follows:

$$\det(H + \alpha E) = \det H \det(I + \alpha H^{-1} E).$$

Now the last determinant is a polynomial of degree n in α, and its lowest degree terms are $1 + \alpha \operatorname{Trace}(H^{-1}E)$. Hence

$$\begin{aligned} \frac{d}{d\alpha} \operatorname{lndet}(H + \alpha E)|_{\alpha=0} &= \frac{1}{\det H} \frac{d}{d\alpha} \det(H + \alpha E)|_{\alpha=0} \\ &= \frac{1}{\det H} \det H \operatorname{Trace}(H^{-1}E) = H^{-1} \bullet E, \end{aligned}$$

from which we obtain

$$\nabla \operatorname{lndet}(H) = H^{-1} \text{ for positive definite } H. \tag{A.4.1}$$

We can also compute the second derivative of lndet: this is the linear function $D^2\text{lndet}(H)[\cdot,\cdot]$ of two symmetric direction matrices that is symmetric in its arguments, $D^2\text{lndet}(H)[E_1,E_2] = D^2\text{lndet}(H)[E_2,E_1]$, and satisfies

$$D^2\text{lndet}(H)[E,E] = \frac{d^2}{d\alpha^2}\text{lndet}(H+\alpha E)|_{\alpha=0}.$$

From our derivations above, the right-hand side is just

$$\frac{d}{d\alpha}(H+\alpha E)^{-1}|_{\alpha=0} \bullet E.$$

To evaluate this, we need the following remarkable lemma.

Lemma A.12. *If $F \in \mathrm{R}^{n\times n}$ satisfies $||F||_2 < 1$, then $I-F$ is nonsingular with*

$$(I-F)^{-1} = I + F + F^2 + \ldots.$$

Proof. First we need to show that the right-hand side is well defined. Let $\phi := ||F||_2 < 1$. Then every entry of F^k is bounded in absolute value by $||F^k||_2 \le ||F||_2^k = \phi^k$, so that every entry on the right-hand side is a series whose terms are bounded in magnitude by those of a geometric series. Hence all these series converge as desired. The result now follows by taking the limit as $k \to \infty$ of the identity

$$(I-F)(I+F+F^2+\cdots+F^k) = I - F^{k+1}. \qquad \square$$

The lemma implies that, if H is nonsingular and α sufficiently small,

$$\begin{aligned}(H+\alpha E)^{-1} &= H^{-1}(I+\alpha E H^{-1})^{-1} \\ &= H^{-1} - \alpha H^{-1}EH^{-1} + \alpha^2 H^{-1}EH^{-1}EH^{-1} - \ldots.\end{aligned} \tag{A.4.2}$$

Putting these facts together, we obtain

$$D^2\text{lndet}(H)[E_1,E_2] = -(H^{-1}E_1H^{-1}) \bullet E_2. \tag{A.4.3}$$

The identity (A.1.1) shows that this is symmetric in its two arguments. Similar analyses can be used to compute any derivative of lndet.

Equation (A.4.2) above allows us to obtain the useful directional derivative of the inverse function:

$$\frac{d}{d\alpha}(H+\alpha E)^{-1}|_{\alpha=0} = -H^{-1}EH^{-1}. \tag{A.4.4}$$

A.5 ▪ Convexity

A set C in R^n is called *convex* if whenever x and y lie in C, so do all convex combinations of the form $(1-\lambda)x + \lambda y$ for $0 \le \lambda \le 1$. The same definition can be used to define convex subsets of $\mathrm{R}^{m\times n}$ or $\mathcal{S}^n$. A function $f : \mathrm{R}^n \to \mathrm{R}$ is called *convex* if, for any arguments x and y in R^n, and any $0 \le \lambda \le 1$,

$$f((1-\lambda)x + \lambda y) \le (1-\lambda)f(x) + \lambda f(y).$$

This definition can be extended to functions defined only on a convex subset C of R^n by restricting x and y to C. It is often convenient to extend such a function to all of R^n by defining it to be $+\infty$ outside C. In this case, f is convex iff

$$\{(x;\xi)\in \mathrm{R}^{n+1} : \xi \geq f(x)\}$$

is a convex set. Again, similar definitions can be made for functions defined on $\mathrm{R}^{m\times n}$ or $\mathcal{S}^n$.

Suppose f is defined and differentiable on an open convex subset C of R^n. Then f is convex iff, for every $x, y \in C$,

$$f(y)\geq f(x)+\nabla f(x)^T(y-x).$$

If f is instead defined and differentiable on an open convex subset of $\mathrm{R}^{m\times n}$ or $\mathcal{S}^n$, the above inequality is replaced by

$$f(Y)\geq f(X)+\nabla f(X)\bullet(Y-X),$$

where $\nabla f(X)$ is the matrix gradient as defined in the previous section.

Finally, suppose f is defined and twice differentiable on an open convex subset of R^n. Then f is convex iff, for every $x \in C$,

$$\nabla^2 f(x) \text{ is positive semidefinite.}$$

If f is defined and twice differentiable on an open convex subset of a matrix space, the condition above becomes

$$D^2 f(X)[E,E]\geq 0 \text{ for all } E,$$

which can be taken as a definition of the positive definiteness of the operator $D^2 f(X)$. This shows that $-\ln\det$ is a convex function on symmetric matrices (see (A.4.3)), since $(H^{-1}EH^{-1})\bullet E = \|H^{-1/2}EH^{-1/2}\|_F^2 \geq 0$.

All of these results can be obtained easily by reducing properties of the function f defined on a high-dimensional space to those of the univariate function ϕ defined by $\phi(\lambda) := f(x+\lambda(y-x))$ or $\phi(\lambda) := f(X+\lambda E)$, etc. Indeed, convexity of f holds iff each such ϕ is convex, and the two conditions above just state that the corresponding ϕ has its second derivative nonnegative at 0.

A.6 ▪ Optimality conditions and duality

Consider the nonlinear programming problem

$$(P)\qquad \begin{array}{ll} \min_x & f(x) \\ & g_i(x)\leq 0,\ i=1,\dots,m, \\ & h_j(x)=0,\ j=1,\dots,p. \end{array}$$

Generally, we assume all functions are finite (real-valued) and differentiable throughout R^n, although it is also useful to allow functions that are finite and differentiable on an open subset of R^n and conventionally defined to be $+\infty$ elsewhere.

We say x is *feasible* for (P) if all functions are finite at x and all constraints are satisfied there. A feasible solution $\bar{x}$ is a *global (local) minimizer* for (P) if (there is a positive ϵ such that), for all feasible x (with $\|x-\bar{x}\|\leq\epsilon$), $f(x)\geq f(\bar{x})$. The following optimality conditions are due to Karush [47] and John [45].

Theorem A.13. *If $\bar{x}$ is a local minimizer for (P), there are multipliers $\tau \geq 0$, $u \in \mathrm{R}^m_+$, and $v \in \mathrm{R}^p$, not all zero, such that*

$$\tau \nabla f(\bar{x}) + \sum_i u_i \nabla g_i(\bar{x}) + \sum_j v_j \nabla h_j(\bar{x}) = 0,$$

$$u_i g_i(\bar{x}) = 0, \quad i = 1, \ldots, m.$$

The second set of equations above are called *complementary slackness* conditions: a multiplier u_i can only be positive if $g_i(\bar{x}) = 0$, i.e., the corresponding inequality constraint is tight at $\bar{x}$.

We can restrict τ to be positive and hence, without loss of generality, equal to 1 under additional regularity conditions. For example, the Slater condition suffices: f and all g_i's are convex, all h_j's are affine, and there is a feasible point $\hat{x}$ with $g(\hat{x}) < 0$ so that all inequality constraints hold strictly. In this case, we obtain the Karush–Kuhn–Tucker conditions [47, 55]:

$$\nabla f(\bar{x}) + \sum_i \bar{u}_i \nabla g_i(\bar{x}) + \sum_j \bar{v}_j \nabla h_j(\bar{x}) = 0,$$

$$\bar{u}_i g_i(\bar{x}) = 0, \quad i = 1, \ldots, m.$$

These conditions are also sufficient for optimality in the convex case: we say that a nonlinear programming problem is convex if f is convex (or concave if we are maximizing), all g_i's are convex, and all h_j's are affine.

Note that the first condition above can be written as

$$\nabla_x L(\bar{x}, \bar{u}, \bar{v}) = 0,$$

where $L : \mathrm{R}^n \times \mathrm{R}^m \times \mathrm{R}^p \to \mathrm{R}$ is the Lagrangian function

$$L(x, u, v) := f(x) + u^T g(x) + v^T h(x).$$

It is clear that $\max_{u \geq 0, v} L(x, u, v)$ equals $f(x)$ if x is feasible and $+\infty$ otherwise, so that we can rewrite our problem as

$$(P) \quad \min_x \{ \max_{u \geq 0, v} L(x, u, v) \}.$$

On the other hand, if $u \geq 0$, for every feasible x, $f(x) \geq L(x, u, v)$, so that $\min_x L(x, u, v)$ provides a lower bound on the optimal value of (P). Hence the Lagrangian dual of (P),

$$(D) \quad \max_{u \geq 0, v} \{ \min_x L(x, u, v) \},$$

has optimal value at most that of (P). In important cases, such as convex problems where an optimal solution $\bar{x}$ exists at which the Karush–Kuhn–Tucker conditions hold, both problems have optimal solutions and their objective values are equal: we say strong duality holds. The difference between the primal and dual optimal objective values is called the duality gap; hence if strong duality holds, there is no duality gap.

For simplicity above, we have taken x to lie in R^n and the constraints to be real valued. However, we can choose matrix spaces in each case. For example, we can replace x with a matrix variable X above, and then the gradients like $\nabla f(\bar{X})$ become the matrix gradients as in the previous sections. Similarly, if the equality constraints become, say, $H(X) = 0$,

where H takes matrices into matrices, the multiplier v becomes a matrix V in the same space as the range of H, $\sum_j v_j \nabla h_j(x)$ becomes $V \bullet \nabla H(X)$, and the last term in the Lagrangian function becomes $V \bullet H(X)$.

Finally, if we have a minimization problem over matrices $X \in \mathcal{S}^n$, and an additional constraint that X must be positive semidefinite, we form the Lagrangian in exactly the same way, but our minimizations in both (P) and (D) over X become minimizations over $X \in \mathcal{S}^n_+$.

A.7 ▪ Compactness of the set of direction matrices

Let $Y \in \mathrm{R}^{k\times m}$ and $Z \in \mathrm{R}^{\ell\times m}$, and let U be a nonzero positive semidefinite diagonal $m \times m$ matrix. In this section we will show that there is a solution $E \in \mathrm{R}^{k\times\ell}$ to

$$E(ZUZ^T) = -YUZ^T, \tag{A.7.1}$$

and that furthermore, there is a compact set $\mathcal{F}$ in $\mathrm{R}^{k\times\ell}$ (depending on Y and Z) such that (A.7.1) has a solution in $\mathcal{F}$ for all such U. (We apologize for the notation in this section. We used letters in (A.7.1) to conform to those in Chapters 4 and 5, but use $\mathcal{F}$ for the compact set since $\mathcal{E}$ is reserved for ellipsoids. Also, we use f_j for a column of E^T because e_j is used for a unit vector, and $-g_j$ for a column of Y^T because y_i is reserved for a column of Y.)

We write (A.7.1) in transposed form, and then consider each column, to get

$$(ZUZ^T)f_j = ZUg_j,$$

where f_j is the jth column of E^T and $-g_j$ is the jth column of Y^T. Note that this holds iff f_j solves the least-squares problem

$$\min ||U^{1/2}Z^T f - U^{1/2}g_j||. \tag{A.7.2}$$

If we can show that f can be restricted to a compact set $\mathcal{F}_j$ in R^ℓ, then existence follows since we are minimizing a continuous function over a compact set, and moreover we can take

$$\mathcal{F} := \{E \in \mathrm{R}^{k\times\ell} : \text{the } j\text{th column of } E^T \text{ lies in } \mathcal{F}_j \text{ for each } j\}.$$

Proposition A.14. *There is a compact set $\mathcal{F}_j$, depending on Y and Z, so that the solution to* (A.7.2) *can be restricted to $\mathcal{F}_j$.*

Proof. Consider the hyperplane arrangement given by the hyperplanes $z_i^T f = \gamma_{ij}, i = 1,\dots,m$, where z_i is the ith column of Z and γ_{ij} is the ith component of g_j. If the span S of the z_i's is not all of R^ℓ, then f can be restricted to S, since any component in its orthogonal complement doesn't change the objective in (A.7.2). Thus we assume $S = \mathrm{R}^\ell$ without loss of generality. Then the arrangement will have vertices, intersection points of ℓ hyperplanes with linearly independent normals z_i. Let $\mathcal{F}_j$ be the convex hull of all such vertices. If f does not lie in this set, it must lie in the relative interior of one of the unbounded polyhedral regions cut out by the hyperplane arrangement, so there is a nonzero direction d with $z_i^T(f + \lambda d) - \gamma_{ij}$ having the same sign for all positive λ. This implies that each $z_i^T(f - \lambda d) - \gamma_{ij}$ is nonincreasing in absolute value as we increase λ, so

we will reach a new point f' lying in a polyhedral region of smaller dimension, with all components of $Z^T f' - g_j$ no larger in absolute value than those of $Z^T f - g_j$. Continuing in this way, we will eventually reach a bounded polyhedral region, so that our point lies in $\mathcal{F}_j$, and has no larger objective value than f. □

A.8 ▪ Derivation of a dual to the maximum-volume inscribed ellipsoid problem

Recall the convex formulation of the maximum-volume inscribed ellipsoid problem in Subsection 6.3.1:

$$\begin{array}{ll} \min_{v\in\mathbf{R}^n, B\in\mathcal{S}^n} & -2\ln\det(B) \\ & y_i^T v + \|By_i\| \le 1, \quad i=1,\dots,m. \end{array} \tag{A.8.3}$$

Using Lagrange multipliers $2\xi \in \mathrm{R}^m$, we obtain the Lagrangian function

$$L(v,B,\xi) := -2\ln\det(B) + 2v^T Y\xi + 2\sum_i \xi_i \|By_i\| - 2e^T\xi.$$

Note that, if $Y\xi \neq 0$, we can drive this to $-\infty$ by an appropriate choice of v. Also, we have assumed that the y_i's span R^n, but if the y_i's corresponding to positive ξ_i's do not span R^n, there will be a vector z orthogonal to all such y_i's, and then fixing v and choosing $B = I + \mu zz^T$, with μ approaching $+\infty$, drives the Lagrangian to $-\infty$.

Thus we assume that $Y\xi = 0$ and the y_i's corresponding to positive ξ_i's span R^n, and consider

$$\begin{array}{l} \phi(\xi) := \min_{v\in\mathbf{R}^n, B\in\mathcal{S}^n}(-2\ln\det(B) + 2v^T Y\xi + 2\sum_i \xi_i\|By_i\| - 2e^T\xi) \\ \qquad = \min_{B\in\mathcal{S}^n} \psi(B,\xi), \end{array} \tag{A.8.4}$$

where

$$\psi(B,\xi) := -2\ln\det(B) + 2\sum_i \xi_i\|By_i\| - 2e^T\xi. \tag{A.8.5}$$

Proposition A.15. *Under the above conditions on ξ, the minimum of $\psi(\cdot,\xi)$ over $\mathcal{S}^n$ is attained by a unique positive definite B.*

Proof. Let $\lambda_{\max}(B)$ be the largest eigenvalue of B and let P_B denote orthogonal projection onto the corresponding eigenspace. Then

$$\psi(B,\xi) \ge -2n\ln\lambda_{\max}(B) + \min\{\xi_i : \xi_i > 0\}\lambda_{\max}(B) \sum_{i:\xi_i>0} \|P_B y_i\| - 2e^T\xi.$$

We claim there is some $\epsilon > 0$ such that $\sum_{i:\xi_i>0}\|P_B y_i\| \ge \epsilon$ for all $B \in \mathcal{S}^n$. Indeed, if not, there is a sequence $\{B_k\} \subset \mathcal{S}^n$ such that $\sum_{i:\xi_i>0}\|P_{B_k} y_i\| \to 0$. Let w_k be any unit vector in the range of P_{B_k}, and by taking limits if necessary, assume $w_k \to w \neq 0$. Then, since $w_k^T y_i \to 0$ for all i with $\xi_i > 0$, we find that $w^T y_i = 0$ for all such y_i's. But this contradicts the fact that these y_i's span R^n.

We conclude that

$$\psi(B,\xi) \ge -2n\ln\lambda_{\max}(B) + \epsilon\min\{\xi_i : \xi_i > 0\}\lambda_{\max}(B) - 2e^T\xi,$$

and since the right-hand side tends to ∞ with $\lambda_{\max}(B)$, we can confine our search for a minimizing B to those with $\lambda_{\max}(B) \le \Lambda$ for some $\Lambda < \infty$.

Also, if $\lambda_{\min}(B)$ denotes the smallest eigenvalue of such B, then

$$\begin{aligned}\psi(B,\xi) &\geq -2(n-1)\ln\lambda_{\max}(B) - 2\ln\lambda_{\min}(B) - 2e^T\xi \\ &\geq -2(n-1)\ln\Lambda - 2\ln\lambda_{\min}(B) - 2e^T\xi,\end{aligned}$$

and the right-hand side tends to ∞ as $\lambda_{\min}(B)$ approaches 0. Hence we can without loss of generality restrict B to the compact set $\{B \in \mathcal{S}^n : \lambda \leq \lambda_{\min}(B) \leq \lambda_{\max}(B) \leq \Lambda\}$ for some $0 < \lambda < \Lambda < \infty$, and since the objective function is continuous on this set, it attains its minimum. Moreover, the minimum is finite (since $B = I$ gives a finite value), so any minimizer must be positive definite, and it must be unique because the objective is strictly convex. □

Since all the y_i's are nonzero, the function $\psi(\cdot,\xi)$ is differentiable at all positive definite B, and with $\|By_i\| = (y_i^T B^2 y_i)^{1/2}$, we find

$$\nabla_B \psi(B,\xi) = -2B^{-1} + \sum_i \frac{\xi_i}{\|By_i\|}(By_i y_i^T + y_i y_i^T B).$$

Let us set

$$u_i := \frac{\xi_i}{\|By_i\|}, \quad i = 1,\ldots,m.$$

Then at a minimizer of $\psi(\cdot,\xi)$, we have

$$2B^{-1} = BYUY^T + YUY^T B.$$

Postmultiplying by B, we see that $YUY^T B^2$ is symmetric, so YUY^T commutes with B^2, and hence with B. Thus B is a minimizer iff

$$B^{-2} = YUY^T \text{ or } B = (YUY^T)^{-1/2},$$

in which case

$$\begin{aligned}\phi(\xi) = \psi(B,\xi) &= \operatorname{lndet}(YUY^T) + 2\sum_i u_i y_i^T (YUY^T)^{-1} y_i - 2e^T\xi \\ &= \operatorname{lndet}(YUY^T) + 2(YUY^T) \bullet (YUY^T)^{-1} - 2e^T\xi \\ &= \operatorname{lndet}(YUY^T) + 2n - 2e^T\xi.\end{aligned}$$

Now suppose that $\hat{\xi} = \lambda\xi$, where $\lambda \geq 0$ and $e^T\xi = n$, and that $\hat{u}$ is defined as above from $\hat{\xi}$. Then if $u := \lambda^2\hat{u}$, we find $u_i = \xi_i / \|(YUY^T)^{-1/2} y_i\|$, so that

$$\phi(\hat{\xi}) = 2n\ln\lambda + \operatorname{lndet}(YUY^T) + 2n - 2n\lambda.$$

In the dual problem, we are maximizing ϕ, so we can assume that λ maximizes the right-hand side above, i.e., $\lambda = 1$. Equivalently, we can assume that $e^T\xi = n$, and so we add this constraint. We can ignore the requirement that the y_i's corresponding to positive ξ_i's (or equivalently positive u_i's) span $\mathbf{R}^n$, because if not, YUY^T is singular and its log determinant is $-\infty$. We thus obtain the dual problem

$$\begin{aligned}\max_{u\in\mathbf{R}^m,\,\xi\in\mathbf{R}^m} \quad & \operatorname{lndet}(YUY^T) \\ & Y\xi = 0, \\ & e^T\xi = n, \\ & \xi_i = u_i \|(YUY^T)^{-1/2} y_i\|, \quad i = 1,\ldots,m, \\ & u \geq 0,\end{aligned}$$

and we have observed in Section 6.3 that in fact the equation linking the ξ and u variables can be relaxed to a greater-than-or-equal-to inequality.

Appendix B

MATLAB Codes

Here is a MATLAB file for the centered minimum-volume enclosing ellipsoid problem. For $n = 2$, it draws the successive ellipses, using David Long's code `ellipse.m` available from MATLAB Central.

```
 function [u,R,factor,improv,mxv,mnv,flagstep,lamhist,var,time,iter,act] =...
     minvol(X,tol,KKY,maxit,print,u)

% Finds the minimum-volume ellipsoid containing the columns of X using the
% Fedorov-Wynn-Frank-Wolfe method, with Wolfe-Atwood away steps if KKY = 0.
% The algorithm also uses the method of Harman and Pronzato to
% eliminate points that are found to be inessential.
%
% The algorithm returns an ellipsoid providing a (1+tol)n-rounding of
% the convex hull of the columns of X in n-space. Set tol to eps/n to get
% a (1+eps)-approximation of the minimum-volume ellipsoid.
%
%%%%%%%%%%%%%%%%%%  INPUT PARAMETERS  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% X is the input data set.
%
% tol is the tolerance (measure of duality gap), set to 10^-7 by default;
%
% KKY is:
%     0 (default) for the Wolfe-Atwood method using Wolfe's away steps
%     (sometimes decreasing the weights) with the Kumar-Yildirim start;
%     1 if using the Fedorov-Wynn-Frank-Wolfe algorithm
%     (just increasing the weights) with the Khachiyan initialization;
%     2 for the Wolfe-Atwood method with the Khachiyan initialization;
%
% maxit is the maximum number of iterations (default 100,000);
%
% print is the desired level of printing (default 1); and
%
% u is the initial value for the weights (default as above).
%
%%%%%%%%%%%%%%%%%%%  OUTPUT PARAMETERS   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
%    u determines the optimal weights on the m columns of X (U = Diag(u));
%
%    R is a scaled (upper triangular) Cholesky factor of
%       M := XUX': R^T*R = factor * XUX';
%
%    improv(i) gives the objective improvement at iteration i;
%
%    mxv(i) gives the maximum variance (x_i^T M^{-1} x_i) at iteration i;
%
%    mnv(i) gives the minimum variance for those i with u_i positive
%       at iteration i;
%
%    flagstep(i) identifies the type of step taken at iteration i: 1(drop),
%       2(decrease), 3(add), and 4(increase);
%
%    lamhist(i) holds the step length lam at iteration i;
%
%    var gives the variances of all the points at completion
%       (points that have been eliminated are assigned the value -1);
%
%    iter is the total number of iterations taken;
%
%    act is the index set of active columns of X, those that have not
%       been eliminated; and
%
%    time is the total cputime spent in order to obtain the optimal solution.
%
%    Calls initwt, updateR, updatevar, and ellipse if n = 2 to draw
%       the ellipse at each iteration.

%%%%%%%%%%%%%%%%  INITIALIZE INPUT PARAMETERS IF NOT DEFINED  %%%%%%%%%%

 if (nargin < 1), error('Please input X'); end
 [n,m] = size(X);
 if (nargin < 2), tol = 1e-07; end
 if (nargin < 3), KKY = 0; end;
 if (nargin < 4), maxit = 100000; end;
 if (nargin < 5), print = 1; end;
 if print,
    fprintf('\n Dimension = %5.0f, Number of points = %5.0f',n,m)
    fprintf(', Tolerance = %5.1e \n',tol);
 end;
 if (nargin < 6),
    if (KKY >= 1),
       u = (1/m) * ones(m,1);
       fprintf('\n Using Khachiyan initialization \n');
    else
        u = initwt(X,print);
    end;
 end;

%%%%%%%%%%%%%%%%  INITIALIZE NECESSARY PARAMETERS  %%%%%%%%%%%%%%%%%%%%

 st = cputime;
```

```
 iter = 1;
 n100 = max([n,100]);
 n50000 = max([n,50000]);
 tol2 = 1e-08;
 mxv = zeros(1,maxit);    % pre-allocate memory for output vectors
 mnv = zeros(1,maxit);
 flagstep = zeros(1,maxit);
 lamhist = zeros(1,maxit);
 mvarerrhist = zeros(1,maxit); improv = zeros(1,maxit);

%%%%%%%%%%%%%%%%  INITIALIZE CHOLESKY FACTOR  %%%%%%%%%%%%%%%%%%%%%%%%%

 upos = find(u > 0);
 lupos = length(upos);
 A = spdiags(sqrt(u(upos)),0,lupos,lupos)*X(:,upos)';  % A'A = M := XUX'
 [Q,R] = qr(A,0);
 factor = 1;                                    % M = factor^-1 * R' * R

% Draw the current ellipse if n = 2.

 if (n == 2),
    pause on;
    clf;
    radii = .02*ones(1,m);
    ellipse(radii,radii,zeros(1,m),X(1,:),X(2,:),'k',100);
    hold on;
    C = 'r';
    M = X(:,upos)*spdiags(u(upos),0,lupos,lupos)*X(:,upos)';
    [V,D] = eig(M);
    phi = atan(V(2,1)/V(1,1));
    aa = sqrt(2*D(1,1)); bb = sqrt(2*D(2,2));
    ellipse(aa,bb,phi,0,0,C,100);
    pause;
 end;

%%%%%%%%%%%%%%%%%%%%%%  INITIALIZE VARIANCES %%%%%%%%%%%%%%%%%%%%%%%%%%%

 RX = R' \ X;      % RX = R^{-T} X
 var = sum(RX .* RX,1); % var(i) = x_i^T M^{-1} x_i

% maxvar is the maximum variance.

 [maxvar,maxj] = max(var);

%%%%%%%%%%%%%%%%%% TRY ELIMINATING POINTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% act lists the mm non-eliminated columns of X,
% and XX is the corresponding submatrix.

 act = 1:1:m;
 XX = X;
 mm = m; oldmm = m;

% Use the Harman-Pronzato test to see if columns of X can be eliminated.
```

```
 ept = maxvar - n;
 tresh = n * (1 + ept/2 - (ept*(4+ept-4/n))^.5/2);
 e = find(var > tresh | u' > tol2);
 act = act(e);
 XX = XX(:,e);
 mm = length(e);

% If only n columns remain, recompute u and R.

 if mm == n,
     u = (1/n)*ones(n,1);
     upos = find(u > tol2);
     A = spdiags(sqrt(u),0,mm,mm) * XX';
     [Q,R] = qr(A,0);
     factor = 1;
     RX = R' \ XX;
     var = sum(RX .* RX,1);
 else
     var = var(e);
     u = u(e)/sum(u(e));
     upos = find(u > tol2);
 end;
 if print,
          fprintf('\n At iteration %6.0f', iter-1);
          fprintf(', number of active points %5.0f \n',length(act));
 end;
 oldmm = mm;

%%%%%%%%%%%%%%%%%%  FIND "FURTHEST" AND "CLOSEST" POINTS %%%%%%%%%%%%%%%

 [maxvar,maxj] = max(var);
 [minvar,ind] = min(var(upos)); minj = upos(ind); mnvup = minvar;

% minj has smallest variance among points with positive weight.

 mxv(iter) = maxvar; mnv(iter) = minvar;
 if KKY==1, fprintf('\n Using KKY'); mnvup = n; end;

%%%%%%%%%%%%%%%%%%%%%%%%%  START ITERATIONS  %%%%%%%%%%%%%%%%%%%%%%%%%

 while ((maxvar > (1+tol)*n) || (mnvup < (1-tol)*n)) && (iter < maxit),

%%%%%%%%%%%%%   SELECT THE COMPONENT TO INCREASE OR DECREASE  %%%%%%%%%%

    if maxvar + mnvup > 2*n,
       j = maxj;
       mvar = maxvar;
    else
       j = minj;
       mvar = mnvup;
    end;

%    Compute Mxj = M^{-1} x_j and recompute var(j).
```

```
    flag_recompute = 0;
    xj = XX(:,j);
    Rxj = R' \ xj;
    Mxj = factor * (R \ Rxj);
    mvarn = factor * (Rxj' * Rxj);
    mvarerror = abs(mvarn - mvar)/max([1,mvar]);
    mvarerrhist(iter) = mvarerror;
    if (mvarerror > tol2),
       flag_recompute = 1;
    end;
    mvar = mvarn;

%%%%%    COMPUTE STEPSIZE LAM (MAY BE NEGATIVE), EPSILON, AND  %%%%%%%%%%
%%%%%    IMPROVEMENT IN LOGDET                                 %%%%%%%%%%

    lam = (mvar - n) / ((n-1) * mvar);
    ep = (mvar/n - 1);
    uj = u(j);
    lam = max(lam,-uj);
    lamhist(iter) = lam;                        % record the stepsize taken
    if lam < -u(j) + tol2, flagstep(iter) = 1;       % drop step
       elseif lam < 0, flagstep(iter) = 2;           % decrease step
       elseif u(j) < tol2, flagstep(iter) = 3;       % add step
       else flagstep(iter) = 4;                      % increase step
    end

%    Update u and make sure it stays nonnegative.

    imp = log(1 + lam*mvar) - n * log(1 + lam);
    uold = u;
    u(j) = max(uj + lam,0); u = (1/(1 + lam)) * u;
    upos = find(u > tol2);
    if (print) && (iter > 1) && (iter-1 == floor((iter-1)/n100) * n100),

%       Print statistics.

       fprintf('\n At iteration %6.0f, maxvar %9.5f',iter-1,maxvar)
       fprintf(', minvar %9.5f',minvar)
    end;

%%%%%%%%    UPDATE (OR RECOMPUTE) CHOLESKY FACTOR AND VAR   %%%%%%%%%%%%%

    if (iter > 1) && ((iter-1 == floor((iter-1)/n50000) * n50000) ...
                    || (flag_recompute && print)),
       upos = find(uold > 0);
       lupos = length(upos);
       M = XX(:,upos) * spdiags(uold(upos),0,lupos,lupos) * XX(:,upos)';
       normdiff = norm(factor*M - R'*R) / (factor*norm(M));
       if (normdiff > tol2),
          flag_recompute = 1;
       end;
       if (flag_recompute && print)
            fprintf('\n Relative error in mvar = %8.1e', mvarerror);
            fprintf(' and in XUX'' = %8.1e; reinverting \n', normdiff);
```

```
            end;
        end;

        if flag_recompute,
           upos = find(u > 0);
           lupos = length(upos);
           A = spdiags(sqrt(u(upos)),0,lupos,lupos) * XX(:,upos)';
           [Q,R] = qr(A,0);
           factor = 1;
           RX = R' \ XX;
           var = sum(RX .* RX,1);
        else

            % Update factorizations.

           [R,factor,down_err] = updateR(R,factor,xj,lam);
           if down_err, fprintf('\n Error in downdating Cholesky'); break; end;
           mult = lam / (1 + lam*mvar);
           var = updatevar(var,lam,mult,Mxj,XX);
        end;

%    Update maxvar.

        [maxvar,maxj] = max(var);

%    Use the Harman-Pronzato test to see if
%    further columns can be eliminated.

        if (iter > 1) && (iter-1 == floor((iter-1)/n100) * n100),
           ept = maxvar - n;
           tresh = n * (1 + ept/2 - (ept*(4+ept-4/n))^.5/2);
           e = find(var > tresh | u' > tol2);
           if length(e) < mm,
              act = act(e);
              XX = XX(:,e);
              mm = length(e);
              if mm == n
                  u = (1/n)*ones(n,1);
                  uold = u;
                  upos = find(u > tol2);
                  A = spdiags(sqrt(u),0,mm,mm) * XX';
                  [Q,R] = qr(A,0);
                  factor = 1;
                  RX = R' \ XX;
                  var = sum(RX .* RX,1);
                  [maxvar,maxj] = max(var);
              else
                  var = var(e);
                  u = u(e)/sum(u(e));
                  uold = uold(e)/sum(uold(e));
                  upos = find(u > tol2);
                  [maxvar,maxj] = max(var);
              end;
              if (print == 2) || (print && (mm < oldmm / 2)),
```

```
                fprintf('\n \n At iteration %6.0f',iter - 1);
                fprintf(', number of active points %5.0f \n',length(act));
            end;
            oldmm = mm;
         end;
      end;

%    Update minvar, iteration statistics.

      upos = find(u > 0);
      [minvar,ind] = min(var(upos)); minj = upos(ind); mnvup = minvar;
      iter = iter+1;
      improv(iter) = imp;
      mxv(iter) = maxvar;
      mnv(iter) = minvar;
      if KKY == 1, mnvup = n; end;

%    Draw the current ellipse if n = 2.

      if (n == 2),
          if (C == 'r'), C = 'b';
             elseif (C == 'b'), C = 'g';
             elseif (C == 'g'), C = 'r';
          end;
          M = XX*spdiags(u,0,mm,mm)*XX';
          [V,D] = eig(M);
          phi = atan(V(2,1)/V(1,1));
          aa = sqrt(2*D(1,1)); bb = sqrt(2*D(2,2));
          ellipse(aa,bb,phi,0,0,C,100);
          pause;
      end;
   end;

%%%%%%%%%%%%%%% CALCULATE AND PRINT SOME OF THE OUTPUT VARIABLES %%%%%%%%%

% Put back eliminated entries.

   mxv = mxv(1:iter); mnv = mnv(1:iter);
   flagstep = flagstep(1:iter); improv = improv(1:iter);
   lamhist = lamhist(1:iter);
   uu = zeros(m,1); uu(act) = u; u = uu;
   varr = -ones(m,1); varr(act) = var; var = varr;
   iter = iter - 1;

   if print,
      for i=1:4, cases(i) = length(find(flagstep==i)); end
      fu = find(u > 1e-12);
      fprintf('\n \n maxvar - n = %4.3e', max(var) - n)
      fprintf(', n - minvar = %4.3e \n', n - min(var(fu)));
      fprintf('\n Drop, decrease, add, increase cases: %6.0f', cases(1));
      fprintf('%6.0f %6.0f %6.0f \n',cases(2),cases(3),cases(4)),
      fprintf('\n Number of positive weights = %7.0f \n', length(fu));
      fprintf('\n Number of iterations       = %7.0f \n', iter);
```

```
   fprintf('\n Time taken                = %7.2f \n \n', cputime - st);
end;

return;
```

Here is the code for generating an initial solution u as in Kumar and Yıldırım [56]:

```
function u = initwt(X,print)

%  obtains the initial weights u using the Kumar-Yildirim algorithm,
%  taking into account that X represents [X,-X].

if (nargin < 2), print = 0; end;
if print, st = cputime; end;
[n,m] = size(X);
u = zeros(m,1);
Q = eye(n);
d = Q(:,1);

% Q is an orthogonal matrix whose first j columns span the same space
% as the first j points chosen X(:,ind) (= (X(:,ind) - (-X(:,ind)))/2).

for j = 1:n,

%     compute the maximizer of | d'*x | over the columns of X.

    dX = abs(d'*X);
    [maxdX,ind] = max(dX);
    u(ind) = 1;
    if j == n, break, end;

%     update Q.

    y = X(:,ind);
    z = Q'*y;
    if j > 1, z(1:j-1) = zeros(j-1,1); end;
    zeta = norm(z); zj = z(j); if zj < 0, zeta = - zeta; end;
    zj = zj + zeta; z(j) = zj;
    Q = Q - (Q * z) * ((1/(zeta*zj)) * z');
    d = Q(:,j+1);
end;
u = u / n;
if print,
   fprintf('\n Initialization time = %5.2f \n',cputime - st);
end;
return;
```

Here are codes to update the "variances" $\omega_i(u)$ and the scaled Cholesky factorization of XUX^T after a rank-one update:

```
function var = updatevar(var,lam,mult,Mxj,XX);

% Updates the variances.
```

```
 tmp = Mxj' * XX;
 var = (1 + lam) * (var - mult* (tmp.^2));
 return;

 function [R,factor,down_err] = updateR(R,factor,xj,lam);

% Updates the Cholesky factor R.

 p = 0;
 xx = sqrt(abs(lam)*factor) * xj;
 if (lam > 0), R = cholupdate(R,xx,'+');
    else, [R,p] = cholupdate(R,xx,'-');
 end;
 factor = factor * (1 + lam);
 if (p>0), down_err = 1; else down_err = 0; end;
 return;
```

Below is the code to generate a matrix X from the rotationally invariant Cauchy distribution described in Section 3.8:

```
 function X=rot_cauchy(n,m,rnd,a);

% Generates a rotated Cauchy-distributed n x m matrix with scale a.

 if nargin < 3, rng('default'); else, rng(rnd); end;
 if nargin < 4, a = 1; end;
 b=randn(m,1);
 c=randn(m,1);
 d=a*(b./c);
 X = randn(n,m);
 d = d ./ (sqrt(sum(X.^2,1)))';
 X = X*spdiags(d,0,m,m);
 return;
```

And finally, here is the code for the minimum-area ellipsoidal cylinder problem (note that the data is to be stored as $X = [Z; Y]$, not $[Y; Z]$):

```
 function [u,R,factor,improv,mxv,mnv,flagstep,lamhist,var,time,iter]...
     =  minvolk(X,k,tol,KKY,maxit,print,u)

% Finds the ellipsoidal cylinder with minimal k-dimensional cross-sectional
% area containing the columns of X:=[Z;Y] with Z in R^(r*m)
% and Y in R^(k*m) using the Fedorov-Wynn-Frank-Wolfe method,
% with Wolfe-Atwood away steps if KKY = 0.
%
% The algorithm returns an ellipsoidal cylinder providing a
% (1+tol)n-rounding of the convex hull of the columns of X in n-space.
%
%%%%%%%%%%%%%%%%%%%  INPUT PARAMETERS  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% X:=[Z;Y] where Z in R^(r*m) and Y in R^(k*m) is the input data set;
%
% k is the dimension of the space where the minimum-area cross-section
%   is desired;
```

```
%
% tol is the tolerance (measure of duality gap), set to 10^-6 by default;
%
% KKY is:
%     0 (default) for the Wolfe-Atwood method using Wolfe's away steps
%     (sometimes decreasing the weights) with the Kumar-Yildirim start;
%     1 if using the Fedorov-Wynn-Frank-Wolfe algorithm
%     (just increasing the weights) with the Khachiyan initialization;
%     2 for the Wolfe-Atwood method with the Khachiyan initialization;
%
% maxit is the maximum number of iterations, default value 100000;
%
% print is the desired level of printing (default 1); and
%
% u is the initial value for the weights (default as above).
%
%%%%%%%%%%%%%%%%%%%%  OUTPUT PARAMETERS   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% u is the dual solution vector;
%
% R and factor are such that factor^-1/2 * R is the (upper
%     triangular) Cholesky factor of the optimal
%     M := XUX^T, and the trailing k x k submatrix Rbar
%     of R and factor are such that factor^-1/2 * Rbar is the (upper
%     triangular) Cholesky factor of the optimal
%     K(u) := YUY^T-YUZ^T(ZUZ^T)^{-1}ZUY^T;
%
% improv(i) holds the improvement in obj. function value at iteration i;
%
% mxv(i) holds the maximum variance over all points at iteration i;
%
% mnv(i) holds the minimum variance over all points with positive
%     weight at iteration i;
%
% flagstep(i) identifies the type of step taken at iteration i: 1(drop),
%     2(decrease), 3(add), and 4(increase);
%
% lamhist(i) holds the step length lam at iteration i;
%
% var gives the variances of all points w.r.t. the optimal u
%     (var_i(u)=(y_i+Ez_i)^T K(u)^{-1} (y_i+Ez_i));
%
% iterno is the total number of iterations taken; and
%
% time is the total cputime spent in order to obtain the optimal solution.
%
% Calls initwt, updatevar, and updateR.

%%%%%%%%%%%%%%%%  INITIALIZE INPUT PARAMETERS IF NOT DEFINED  %%%%%%%%%%

 if (nargin < 2), error('Please input X and k'); end
 [n,m] = size(X);
 if (nargin < 3), tol = 1e-06; end;
 if (nargin < 4), KKY = 0; end;
```

```
if (nargin < 5), maxit = 100000; end;
if (nargin < 6), print = 1; end;
if print,
   fprintf('\n Dimension = %5.0f, Number of points = %5.0f',n,m)
   fprintf(', Tolerance = %5.1e \n',tol);
   fprintf('\n Dimension of y-space = %5.0f \n',k);
end;
if (nargin < 7),
   if (KKY >= 1),
      u = (1/m) * ones(m,1);
      fprintf('\n Using Khachiyan initialization \n');
   else
       u = initwt(X,print);
   end;
end;


%%%%%%%%%%%%%%%%  INITIALIZE NECESSARY PARAMETERS  %%%%%%%%%%%%%%%%%%%%%%

st = cputime;
r = n - k;
iter = 1;
n100 = max([n,100]);
n50000 = max([n,50000]);
ximult = 1;
gamma = 1000;
tol2 = 10^-8;
mxv = zeros(1,maxit);    % pre-allocate memory for output vectors
mnv = zeros(1,maxit);
flagstep = zeros(1,maxit);
lamhist = zeros(1,maxit);
improv = zeros(1,maxit);

%%%%%%%%%%%%%%%%  PARTITION X INTO BLOCKS Y AND Z  %%%%%%%%%%%%%%%%%%%%%%

Z = X(1:r,:);

%Y = X(r+1:n,:);     % Y is not used, but is helpful for explaining some
                     % computed quantities.

%%%%%%%%%%%%%%%%  INITIALIZE FACTORIZATION  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

upos = find(u > 0);
lupos = length(upos);

% M = X(:,upos)*spdiags(u(upos),0,lupos,lupos)*X(:,upos)';
                                    % M   = XUX' (X * diag(u) * X')
% MZZ = M(1:r,1:r);                 % MZZ = ZUZ' (Z * diag(u) * Z')
                                    % M and MZZ are not used, but are
                                    % helpful in explaining some
                                    % computed quantities.

A = spdiags(sqrt(u(upos)),0,lupos,lupos)*X(:,upos)'; % A'A = M.
factor = 1;
```

```
 [Q,R]  = qr(A,0);                    % M    = factor^-1 * R' * R
 RZ     = R(1:r,1:r);                 % MZZ  = factor^-1 * RZ'* RZ

%%%%%%%%%%%%%%%%%%%%%%%%  INITIALIZE VARIANCES %%%%%%%%%%%%%%%%%%%%%%%%%%

 RX   = R' \ X;                   % RX = R^-T * X;
 RZZ  = RZ' \ Z;                  % RZZ = RZ^-T * Z

 zeta = sum(RZZ .* RZZ,1);        % zeta(i) = z_i' * (ZUZ')^-1 * z_i
 xi   = sum(RX .* RX,1);          % xi(i) = x_i' * (XUX^T)^-1 * x_i
 var  = xi - zeta;                % var(i) = (y_i+Ez_i)'K(u)^{-1}(y_i+Ez_i)

%%%%%%%%%%%%%%%%%%%%%%%%  FIND "FURTHEST" AND "CLOSEST" POINTS %%%%%%%%%%%%

 upos = find(u > 0);
 [maxvar,maxj] = max(var);   % maxj has greatest variance among all points
 [minvar,ind] = min(var(upos)); minj = upos(ind); mnvup = minvar;
 if (maxvar > k*(m-1)),
    fprintf('\n Initialization worse than Khachiyan''s');
    fprintf(', maxvar = %5.0f \n',maxvar);
 end;

% minj has smallest variance among points with positive weight

 mxv(iter) = maxvar; mnv(iter) = minvar;

 if (KKY == 1), fprintf('\n Using KKY \n'); mnvup = k; end;

%%%%%%%%%%%%%%%%%%%%%%%%%%  START ITERATIONS  %%%%%%%%%%%%%%%%%%%%%%%%%%%

 while ((maxvar > (1+tol) * k) || (mnvup < (1-tol) * k)) && (iter < maxit),

%%%%%%%%%%%%%%   SELECT THE COMPONENT TO INCREASE OR DECREASE  %%%%%%%%%%

    if (maxvar + mnvup < 2*k),
         j = minj;
         mvar = mnvup;
    else
         j = maxj;
         mvar = maxvar;
    end
    uj = u(j);

    % Recompute var(j) and zeta(j).

    flag_recompute = 0;
    xj = X(:,j); zj = Z(:,j);
    RZzj = RZ' \ zj;                    % RZzj  = RZ^-T * z(j)
    MZZzj = factor * (RZ \ RZzj);       % MZZzj = (ZUZ')^{-1} * z(j)
    Rxj = R' \ xj;
    Mxj = factor * (R \ Rxj);     % Mxj = (XUX')^{-1} * xj
    zetaj = factor * (RZzj' * RZzj);    % zeta(j) recomputed
    xij = factor * (Rxj' * Rxj);        % xi(j) recomputed
    mvarn = xij - zetaj;                % var(j) recomputed
```

```
    mvarerror = abs(mvarn - mvar)/max([1,mvar]);
    mvarerrhist(iter) = mvarerror;
    if (mvarerror > tol2),
       flag_recompute = 1;
    end;
    mvar = mvarn;

%%%%%    COMPUTE STEPSIZE LAM (MAY BE NEGATIVE) %%%%%%%%%%

    % The derivative of the improvement in the objective w.r.t lambda is a
    % negative quantity times a * lambda^2 - 2 b * lambda +c with
    % a, b, and c as below.

    % We need to find the roots of this quadratic; different
    % cases are investigated in the following if clauses such as no
    % real roots, single root, double real roots.

    b = - mvar / 2 + mvar / (2*k) - zetaj;
    a = zetaj * (mvar + zetaj);
    c = 1 - mvar / k;

    % Identify various cases that can arise when solving the quadratic
    % equation a*lambda^2 - 2b*lambda + c =0.

    if (abs(a) < 1e-15),
        if (abs(b) < 1e-15),
            if (c < 0),
                lam = 1e10;
            else
                lam = -uj;
            end
        else
            lam = c / (2*b);
        end
    else
        if b*b > a*c,
            lam = c / (b - sqrt(b*b - a*c));
        else
            if c < 0, lam = 1e10; else lam = -uj; end
        end
    end
    lam = max(lam,-uj);

%   If the step would make ZUZ' close to singular, take a slightly
%   shorter step.

    if (lam < -.9999*uj) & (1 + lam*zetaj < .0001),
        fprintf('\n Truncating step making ZUZ'' singular \n');
        lam = -.9999*uj;
    end

%   If the increase in xi is too great compared to lam, reject the decrease
%   or drop step, and perform an increase or add step
    if (lam < 0) && (lam > -uj),
```

```
   ximultold = ximult;
   ximult = (1+lam)*ximult / ((1+lam*xij)*(1-gamma*lam));
end;
if (lam == -uj),
   ximultold = ximult;
   ximult = (1+lam)*ximult / (1+lam*xij);
end;
if (lam > 0), ximult = (1+lam)*ximult / (1+gamma*lam); end;

if ximult > gamma,

   fprintf('\n Rejecting decrease or drop \n');

   % Reject drop or decrease step and do add or increase step

   ximult = ximultold;
   j = maxj;
   mvar = maxvar;
   uj = u(j);

   % Recompute var(j) and zeta(j).

   xj = X(:,j); zj = Z(:,j);
   RZzj = RZ' \ zj;                    % RZzj  = RZ^-T * z(j)
   MZZzj = factor * (RZ \ RZzj);       % MZZzj = (ZUZ')^{-1} * z(j)
   Rxj = R' \ xj;
   Mxj = factor * (R \ Rxj);
   zetaj = factor * (RZzj' * RZzj);    % zeta(j) recomputed
   xij = factor * (Rxj' * Rxj);        % xi(j) recomputed
   mvarn = xij - zetaj;                 % var(j) recomputed
   mvarerror = abs(mvarn - mvar)/max([1,mvar]);
   mvarerrhist(iter) = mvarerror;
   if (mvarerror > 1e-08),
      if print, mvarerror, end;
      flag_recompute = 1;
   end;
   mvar = mvarn;

   % The derivative of the improvement in the objective w.r.t lambda is a
   % negative quantity times a * lambda^2 - 2 b * lambda +c with
   % a, b, and c as below.

   % We need to find the roots of this quadratic; different
   % cases are investigated in the following if clauses such as no
   % real roots, single root, double real roots.

   b = - mvar / 2 + mvar / (2*k) - zetaj;
   a = zetaj * (mvar + zetaj);
   c = 1 - mvar / k;

   % Identify various cases that can arise when solving the quadratic
   % equation a*lambda^2 - 2b*lambda + c =0.

   if (abs(a) < 1e-15),
```

```
            if (abs(b) < 1e-15),
                if (c < 0),
                    lam = 1e10;
                else
                    lam = -uj;
                end
            else
                lam = c / (2*b);
            end
        else
            if b*b > a*c,
                lam = c / (b - sqrt(b*b - a*c));
            else
                if c < 0, lam = 1e10; else lam = -uj; end
            end
        end
        lam = max(lam,-uj);
        if (lam > 0), ximult = (1+lam)*ximult / (1+gamma*lam); end;
    end
    if (j == maxj) && (mvar < (1 + tol)*k),
        fprintf('\n Terminating with epsilon-primal feasible u \n');
        break;
    end;
    lamhist(iter)=lam;    % record the stepsize taken

    if lam < -uj+tol2, flagstep(iter) = 1;            % drop steps
    elseif lam < 0, flagstep(iter) = 2;               % decrease steps
    elseif uj < tol2, flagstep(iter) = 3;             % add steps
    else flagstep(iter) = 4;                          % increase steps
    end

    % Update u and make sure it stays nonnegative,

    imp = - k*log(1 + lam) + log(1 + lam*mvar/(1 + lam*zetaj));
    improv(iter) = imp;

    uold = u;
    u(j) = max(uj + lam,0); u = (1/(1 + lam)) * u;
    if print && (iter > 1) && (iter-1 == floor((iter-1)/n100) * n100),

%       Print statistics.

       fprintf('\n At iteration %6.0f, maxvar %9.5f',iter-1,maxvar)
       fprintf(', minvar %9.5f',minvar)
    end;

%%%%%%%%    UPDATE (OR RECOMPUTE) CHOLESKY FACTOR AND VAR   %%%%%%%%%%%%%%

    if (iter > 1) && ((iter-1 == floor((iter-1)/n50000) * n50000) ...
                      || (flag_recompute && print)),
        upos = find(uold > 0);
        lupos = length(upos);
        if (k > 0.5*n),
            M = X(:,upos) * spdiags(uold(upos),0,lupos,lupos) * X(:,upos)';
```

```
            normdiff = norm(factor*M - R'*R) / (factor*norm(M));
        else
            MZZ = Z(:,upos) * spdiags(uold(upos),0,lupos,lupos) * Z(:,upos)';
            normdiff = norm(factor*MZZ - RZ'*RZ) / (factor*norm(MZZ));
        end;
        if (normdiff > tol2),
            flag_recompute = 1;
        end;
        if (flag_recompute && print)
            fprintf('\n Relative error in mvar = %8.1e', mvarerror);
            fprintf(' and in XUX'' = %8.1e; reinverting \n', normdiff);
       end;
    end;

    if flag_recompute == 1,
        upos = find(u > 0);
        lupos = length(upos);
        % M = X(:,upos) * spdiags(u(upos),0,lupos,lupos) * X(:,upos)';
                                              % M   = XUX' (X * diag(u) * X')
        A = spdiags(sqrt(u(upos)),0,lupos,lupos) * X(:,upos)';
        factor = 1;
        [Q,R]  = qr(A,0);                % M    = factor * R' * R
        RZ     = R(1:r,1:r);             % MZZ  = factor * RZ'* RZ
        RX     = R' \ X;
        RZZ    = RZ' \ Z;
        zeta   = sum(RZZ .* RZZ,1);
        xi     = sum(RX .* RX,1);
        var    = xi - zeta;
    else

        % Update factorizations.

        [R,factor,down_err]  = updateR(R,factor,xj,lam);
        if down_err, fprintf('\n Error in downdating Cholesky \n');break;end;
        RZ   = R(1:r,1:r);               % MZZ  = factor * RZ'* RZ
        mu = lam / (1 + lam*zetaj);
        zeta = updatevar(zeta,lam,mu,MZZzj,Z);
        nu = lam / (1 + lam*xij);
        xi = updatevar(xi,lam,nu,Mxj,X);
        var = xi - zeta;
    end

    %%%%% FIND "FURTHEST" AND "CLOSEST" POINTS USING UPDATED VAR %%%%%%%%%%

    upos = find(u > tol2);
    [maxvar,maxj] = max(var);
    [minvar,ind] = min(var(upos)); minj = upos(ind); mnvup = minvar;
    iter = iter + 1;
    mxv(iter) = maxvar;
    mnv(iter) = minvar;
    if (KKY == 1), mnvup = k; end;

end
```

```
%%%%%%%%%%%%%%%% CALCULATE AND PRINT SOME OF THE OUTPUT VARIABLES %%%%%%%%%

 mxv = mxv(1:iter); mnv = mnv(1:iter);
 flagstep = flagstep(1:iter); improv=improv(1:iter);
 lamhist = lamhist(1:iter);
 iter = iter - 1;

 if print,
    fu = find(u > 1e-12);           % indices of points with positive weight
    for i=1:4, cases(i) = length(find(flagstep==i)); end
    fprintf('\n \n maxvar - k = %4.3e', max(var) - k)
    fprintf(', k - minvar = %4.3e \n', k - min(var(fu)));
    fprintf('\n Drop, decrease, add, increase cases: %6.0f', cases(1));
    fprintf('%6.0f %6.0f %6.0f \n',cases(2),cases(3),cases(4)),
    fprintf('\n Number of positive weights = %7.0f \n', length(fu));
    fprintf('\n Number of iterations       = %7.0f \n', iter);
    fprintf('\n Time taken                 = %7.2f \n \n', cputime - st);
 end;

 return
```

Bibliography

[1] J. Agullo. Exact iterative computation of the multivariate minimum volume ellipsoid estimator with a branch and bound algorithm. In A. Prat, editor, *Proceedings in Computational Statistics*, pages 175–180. Physica-Verlag, Heidelberg, 1996. (Cited on p. 90)

[2] S. D. Ahipaşaoğlu. *Solving ellipsoidal inclusion and optimal experimental design problems: Theory and algorithms*. Doctoral thesis, Cornell University, Ithaca, NY, 2009. (Cited on pp. 47, 87)

[3] S. D. Ahipaşaoğlu, P. Sun, and M. J. Todd. Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimization Methods and Software*, 23:5–19, 2008. (Cited on pp. 22, 47, 50)

[4] S. D. Ahipaşaoğlu and M. J. Todd. A modified Frank–Wolfe algorithm for computing minimum-area enclosing cylinders: Theory and algorithms. *Computational Geometry*, 46:494–519, 2013. (Cited on pp. ix, 52, 87)

[5] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *International Journal of Computational Geometry and Applications*, 12:125–141, 2002. (Cited on p. 10)

[6] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 156–163, 1990. (Cited on p. 96)

[7] P. Artzner, F. Delbaen, J. M. Eber, and D. C. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. (Cited on p. 95)

[8] C. L. Atwood. Optimal and efficient designs of experiments. *The Annals of Mathematical Statistics*, 40:1570–1602, 1969. (Cited on p. 66)

[9] C. L. Atwood. Sequences converging to D-optimal designs of experiments. *The Annals of Statistics*, 1:342–352, 1973. (Cited on pp. 49, 86)

[10] E. R. Barnes. An algorithm for separating patterns by ellipsoids. *IBM Journal of Research and Development*, 26:759–764, 1982. (Cited on p. 49)

[11] J. Batson, D. A. Spielman, and N. Srivastava. Twice-Ramanujan sparsifiers. *SIAM Review*, 56:315–334, 2014. (Cited on pp. 45, 46, 107)

[12] A. A. Benczur and D. R. Karger. Approximating $s-t$ cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 47–55, 1996. (Cited on p. 45)

[13] U. Betke and M. Henk. Approximating the volume of convex bodies. *Discrete Computational Geometry*, 10:15–21, 1993. (Cited on p. 49)

[14] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: A survey. *Operations Research*, 29:1039–1091, 1981. (Cited on p. 10)

[15] D. Böhning. A vertex-exchange-method in D-optimal design theory. *Metrika*, 33:337–347, 1986. (Cited on p. 49)

[16] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer-Verlag, New York, 2000. (Cited on p. 105)

[17] J. Bourgain and S. J. Szarek. The Banach–Mazur distance to the cube and the Dvoretsky–Rogers factorization. *Israel Journal of Mathematics*, 62:169–180, 1988. (Cited on p. 107)

[18] B. P. Burrell and M. J. Todd. The ellipsoid method generates dual variables. *Mathematics of Operations Research*, 10:688–700, 1985. (Cited on p. 50)

[19] F. Chazal, D. Cohen-Steiner, and A. Lieutier. A sampling theory for compacts in Euclidean space. In *Proceedings of the 22nd Annual ACM Symposium on Computational Geometry*, pages 319–326, 2006. (Cited on p. 10)

[20] F. L. Chernousko. Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis*, 63:872–879, 2005. (Cited on p. 8)

[21] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, Philadelphia, 2000. (Cited on p. 65)

[22] L. Danzer, D. Laugwitz, and H. Lenz. Über das Löwnersche ellipsoid und sein analogon unter den einem eikörper einbeschreibenen ellipsoiden. *Archiv der Mathematik*, 8:214–219, 1957. (Cited on p. 10)

[23] B. C. Eaves. Pivoting to normalize a basic matrix. *Mathematical Programming*, 62:553–556, 1993. (Cited on p. 97)

[24] D. Eberly. *3D Game Engine Design*. Morgan Kaufmann, San Francisco, CA, 2001. (Cited on p. 8)

[25] G. Elfving. Optimum allocation in linear regression theory. *Annals of Mathematical Statistics*, 23:255–262, 1952. (Cited on p. 10)

[26] D. J. Elzinga and D. W. Hearn. The minimum covering sphere problem. *Management Science*, 19:96–104, 1972. (Cited on p. 8)

[27] V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972. (Cited on pp. 10, 49, 66, 86)

[28] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York, 1968. (Cited on p. 80)

[29] E. Fogel and Y. F. Huang. On the value of information in system identification—bounded noisy case. *Automatica*, 18:229–238, 1982. (Cited on p. 23)

[30] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956. (Cited on p. 49)

[31] R. M. Freund and P. Grigas. New analysis and results for the Frank–Wolfe method. *Mathematical Programming*, 155:199–230, 2016. (Cited on p. 50)

[32] A. A. Giannopoulos. A note on the Banach–Mazur distance to the cube. *Operator Theory: Advances and Applications*, 77:67–73, 1995. (Cited on p. 107)

[33] A. A. Giannopoulos and V. D. Milman. Euclidean structure in finite dimensional normed spaces. In W. B. Johnson and J. Lindenstrauss, editors, *Handbook of the Geometry of Banach Spaces*, volume 1, chapter 17, pages 707–779. Elsevier Science, Amsterdam, 2001. (Cited on p. 107)

[34] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28:505–535, 1974. (Cited on p. 117)

[35] F. Glineur. *Pattern separation via ellipsoids and conic programming.* Master's thesis, Faculté Polytechnique de Mons, Belgium, 1998. (Cited on p. 8)

[36] J.-Y. Gotoh and A. Takeda. Conditional minimum volume ellipsoid with application to multiclass discrimination. *Computational Optimization and Applications*, 41:27–51, 2008. (Cited on pp. 89, 90, 94, 106)

[37] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988. (Cited on p. 10)

[38] J. Guélat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35:110–119, 1986. (Cited on p. 50)

[39] F. Gürtuna. Duality of ellipsoidal approximations via semi-infinite programming. *SIAM Journal on Optimization*, 20:1421–1438, 2009. (Cited on pp. 23, 107)

[40] W. W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31:221–239, 1989. (Cited on p. 116)

[41] R. Harman and L. Pronzato. Improvements on removing nonoptimal support points in *D*-optimum design algorithms. *Statistics and Probability Letters*, 77:90–94, 2007. (Cited on pp. 50, 85)

[42] A. Hero, Y. Zhang, and W. Rogers. Consistency set estimation for PET image reconstruction. In *Proceedings of the 1993 International Conference on Information Science and Systems*, pages 605–610. Johns Hopkins, Baltimore, MD, 1993. (Cited on p. 8)

[43] A. Hero, Y. Zhang, and W. Rogers. Tomographic feature detection and classification using parallelotope bounded error estimation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 284–289. IEEE, 1997. (Cited on p. 8)

[44] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 427–435. JMLR.org, 2013. (Cited on p. 50)

[45] F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, presented to R. Courant on his 60th birthday, January 8, 1948*, pages 187–204. Interscience, New York, 1948. (Cited on pp. 4, 10, 22, 119)

[46] S. Karlin and W. J. Studden. Optimal experimental designs. *The Annals of Mathematical Statistics*, 37:783–815, 1966. (Cited on p. 66)

[47] W. Karush. *Minima of functions of several variables with inequalities as side conditions.* Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, 1939. (Cited on pp. 10, 119, 120)

[48] L. G. Khachiyan. A polynomial algorithm for linear programming. English translation: *Soviet Mathematics Doklady*, 20:191–194, 1979. Russian original: *Doklady Akademiia Nauk SSSR*, 244:1093–1096. (Cited on p. 10)

[49] L. G. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21:307–320, 1996. (Cited on pp. 22, 49)

[50] L. G. Khachiyan and M. J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159, 1993. (Cited on pp. 22, 98, 100, 101, 102, 104, 107)

[51] J. Kiefer. Optimum designs in regression problems, II. *Annals of Mathematical Statistics*, 32:298–325, 1961. (Cited on p. 66)

[52] J. Kiefer and J. Wolfowitz. Optimum designs in regression problems. *Annals of Mathematical Statistics*, 30:271–294, 1959. (Cited on p. 10)

[53] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960. (Cited on pp. 10, 22)

[54] H. König and D. Pallaschke. On Khachiyan's algorithm and minimal ellipsoids. *Numerische Mathematik*, 36:211–223, 1981. (Cited on p. 23)

[55] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Oakland, CA, 1951. (Cited on pp. 10, 120)

[56] P. Kumar and E. A. Yıldırım. Minimum-volume enclosing ellipsoids and core sets. *J. Optimization Theory and Applications*, 126(1):1–21, 2005. (Cited on pp. 22, 49, 132)

[57] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In C. Cortes et al., editors, *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 496–504. Curran Associates, Inc., New York, 2015. (Cited on p. 50)

[58] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983. (Cited on pp. 8, 96)

[59] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with confidence from random samples. *Discrete and Computational Geometry*, 39:419–441, 2006. (Cited on p. 10)

[60] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006. (Cited on p. 80)

[61] J. Peña and D. Rodriguez. Polytope conditioning and linear convergence of the Frank–Wolfe algorithm. Technical report. Tepper School of Business, Carnegie-Mellon University, Pittsburgh, PA, 2015. (Cited on p. 50)

[62] G. Pisier. *The Volume of Convex Bodies and Banach Space Geometry*. Cambridge University Press, Cambridge, UK, 1989. (Cited on p. 10)

[63] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, New York, 1993. (Cited on pp. 10, 66)

[64] S. M. Robinson. Generalized equations and their solutions, part II: Applications to nonlinear programming. *Mathematical Programming Study*, 19:200–221, 1982. (Cited on pp. 39, 50)

[65] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2:21–41, 2000. (Cited on pp. 90, 106)

[66] J. B. Rosen. Pattern recognition by convex programming. *Journal of Mathematical Analysis and Applications*, 10:123–134, 1965. (Cited on pp. 8, 49)

[67] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987. (Cited on p. 90)

[68] F. C. Schweppe. Recursive state estimation: unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13:22–28, 1968. (Cited on p. 8)

[69] N. Z. Shor. Cut-off method with space extension in convex programming problems. English translation: *Cybernetics* 13(1), 94–96, 1977. Russian original: *Kibernetika*, 13(1):94–95. (Cited on p. 10)

[70] R. Sibson. Discussion on the papers by Wynn and Laycock. *Journal of the Royal Statistical Society*, 34:181–183, 1972. (Cited on pp. 22, 23, 65)

[71] B. W. Silverman and D. M. Titterington. Minimum covering ellipses. *SIAM Journal on Scientific and Statistical Computing*, 1:401–409, 1980. (Cited on p. 8)

[72] S. Silvey and D. Titterington. A geometric approach to optimal design theory. *Biometrika*, 62:21–32, 1973. (Cited on p. 65)

[73] S. D. Silvey. *Optimal Design: An Introduction to the Theory for Parameter Estimation*. Chapman and Hall, New York, 1980. (Cited on pp. 10, 66)

[74] S. D. Silvey. Discussion on the papers by Wynn and Laycock. *Journal of the Royal Statistical Society*, 34:174–175, 1972. (Cited on pp. 22, 65)

[75] D. A. Spielman and N. Srivastava. An elementary proof of the restricted invertibility problem. *Israel Journal of Mathematics*, 190:83–91, 2012. (Cited on p. 107)

[76] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40:981–1025, 2011. (Cited on p. 44)

[77] P. Sun and R. M. Freund. Computation of minimum volume covering ellipsoids. *Operations Research*, 52:690–706, 2004. (Cited on pp. 46, 94)

[78] S. P. Tarasov, L. G. Khachiyan, and I. I. Erlikh. The method of inscribed ellipsoids. English translation: *Soviet Mathematics Doklady*, 37:226–230, 1988. Russian original: *Doklady Akademiia Nauk SSSR*, 298:1081–1085. (Cited on pp. 99, 107)

[79] D. M. Titterington. Optimal design: some geometrical aspects of D-optimality. *Biometrika*, 62(2):313–320, 1975. (Cited on pp. 22, 66)

[80] M. J. Todd. On minimum volume ellipsoids containing part of a given ellipsoid. *Mathematics of Operations Research*, 7:253–261, 1982. (Cited on pp. 23, 41, 42, 50, 98, 99)

[81] M. J. Todd and E. A. Yıldırım. On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids. *Discrete and Applied Mathematics*, 155(13):1731–1744, 2007. (Cited on pp. 49, 50)

[82] A. Vicino and G. Zappa. Sequential approximation of feasible parameter sets for identification with set membership uncertainty. *IEEE Transactions on Automatic Control*, 41:774–785, 1996. (Cited on p. 8)

[83] A. Wald. On the power function of the analysis of variance test. *Annals of Mathematical Statistics*, 13:434–439, 1942. (Cited on p. 10)

[84] P. Wolfe. Convergence theory in nonlinear programming. In J. Abadie, editor, *Integer and Nonlinear Programming*, pages 1–36. North-Holland, Amsterdam, 1970. (Cited on pp. 49, 50)

[85] H. P. Wynn. The sequential generation of D-optimum experimental design. *Annals of Mathematical Statistics*, 41:1655–1664, 1970. (Cited on p. 49)

[86] H. P. Wynn. Results in the theory and construction of D-optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34:133–147, 1972. (Cited on p. 86)

[87] Y. Ye. On the complexity of approximating a KKT point of quadratic programming. *Mathematical Programming*, 80:195–211, 1998. (Cited on p. 65)

[88] P. Youssef. Restricted isometry and the Banach-Mazur distance to the cube. *Mathematika*, 60:201–218, 2014. (Cited on p. 107)

[89] D. B. Yudin and A. S. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. English translation: *Matekon* 13: 3–25, 1976. Russian original: *Ékonomika i Matematicheskie Metody*, 12:357–369. (Cited on p. 10)

[90] V. L. Zaguskin. Circumscribed and inscribed ellipsoids of extremal volume (in Russian). *Uspehi Matematicheskikh Nauk*, 13:89–93, 1958. (Cited on p. 10)

Index

This book, the first on these topics, addresses the problem of finding an ellipsoid to represent a large set of points in high-dimensional space, which has applications in computational geometry, data representations, and optimal design in statistics. The book covers the formulation of this and related problems, theoretical properties of their optimal solutions, and algorithms for their solution. Due to the high dimensionality of these problems, first-order methods that require minimal computational work at each iteration are attractive. While algorithms of this kind have been discovered and rediscovered over the past fifty years, their computational complexities and convergence rates have only recently been investigated. The optimization problems in the book have the entries of a symmetric matrix as their variables, so the author's treatment also gives an introduction to recent work in matrix optimization.

The author

- provides historical perspective on the problems studied by optimizers, statisticians, and geometric functional analysts;
- demonstrates the huge computational savings possible by exploiting simple updates for the determinant and the inverse after a rank-one update, and highlights the difficulties in algorithms when related problems are studied that do not allow simple updates at each iteration; and
- gives rigorous analyses of the proposed algorithms, MATLAB codes, and computational results.

This book will be of interest to graduate students and researchers in operations research, theoretical statistics, data mining, complexity theory, computational geometry, and computational science.

Michael J. Todd is Leon C. Welch Professor Emeritus of the School of Operations Research and Information Engineering, Cornell University. He received a Guggenheim Fellowship, 1980–1981; a Sloan Research Fellowship, 1981–1985; the George B. Dantzig Prize, 1988; and the John von Neumann Theory Prize, 2003. He is an INFORMS Fellow and a SIAM Fellow. He has served on the editorial boards of *Mathematics of Operations Research*, *Operations Research*, and *SIAM Journal on Optimization*. He was Managing Editor of *Foundations of Computational Mathematics* and served on the boards of *Acta Numerica* and *Foundations and Trends in Optimization*. He is the author of one and co-editor of five previous books.

For more information about MOS and SIAM books, journals, conferences, memberships, or activities, contact:

Society for Industrial
and Applied Mathematics
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA
+1-215-382-9800 • Fax +1-215-386-7999
siam@siam.org • *www.siam.org*

Mathematical Optimization Society
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA
+1-215-382-9800 x319
Fax +1-215-386-7999
service@mathopt.org • *www.mathopt.org*

MO23

ISBN 978-1-611974-37-9
90000
9781611974379