OPERATIONS RESEARCH
Vol. 52, No. 5, September–October 2004, pp. 690–706
ISSN 0030-364X | EISSN 1526-5463 | 04 | 5205 | 0690

informs®

DOI 10.1287/opre.1040.0115
© 2004 INFORMS

Computation of Minimum-Volume Covering Ellipsoids

Peng Sun
The Fuqua School of Business, Duke University, Box 90120, Durham, North Carolina 27708, peng.sun@duke.edu

Robert M. Freund
Sloan School of Management, Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, Massachusetts 02142, rfreund@mit.edu

We present a practical algorithm for computing the minimum-volume n-dimensional ellipsoid that must contain m given points $a_1, \dots, a_m \in \mathbb{R}^n$. This convex constrained problem arises in a variety of applied computational settings, particularly in data mining and robust statistics. Its structure makes it particularly amenable to solution by interior-point methods, and it has been the subject of much theoretical complexity analysis. Here we focus on computation. We present a combined interior-point and active-set method for solving this problem. Our computational results demonstrate that our method solves very large problem instances ($m = 30{,}000$ and $n = 30$) to a high degree of accuracy in under 30 seconds on a personal computer.

Subject classifications: programming, nonlinear: algorithms, large-scale systems; statistics: cluster analysis, data analysis; mathematics: convexity, matrices.
Area of review: Optimization.
History: Received August 2002; revision received June 2003; accepted July 2003.

1. Introduction

This paper is concerned with computing the minimum-volume ellipsoid in n-dimensional space $\mathbb{R}^n$ containing m given points $a_1, a_2, \dots, a_m \in \mathbb{R}^n$. This minimum-volume covering ellipsoid (MVCE) problem is useful in a variety of different application areas. In computational statistics, the minimum-volume ellipsoid covering k of m given points in $\mathbb{R}^n$ is well-known for its affine equivariance and positive breakdown properties as a multivariate location and scattering estimator (Croux et al. 2002). In the area of robust statistics and data mining, efficiently finding outliers is a challenge that has attracted much research interest (Knorr et al. 2001). Indeed, one can identify data outliers quickly if one can compute the minimum-volume ellipsoid quickly, because outliers are essentially points on the boundary of the minimum-volume covering ellipsoid.

Another emerging research area in data mining is that of finding linear-transformation-invariant (or scale-invariant) clustering methods that work for very large datasets. In a multidimensional setting, it is important that data-clustering results do not vary under changes in the relative scales of different dimensional coordinates. However, traditional distance-based clustering methods such as k-mean or k-median methods are not scale invariant, because typical distance measures (such as Euclidean distance) depend on the scale factor of each coordinate dimension (see, for example, §8.2.1 of Han and Kamber 2001). In contrast, clustering using minimum-volume ellipsoids, which use the minimum-volume covering ellipsoid to cover all points in each cluster and minimize the total volume of these covering ellipsoids, has the linear-transformation-invariance property. Such clustering schemes iteratively partition the datapoints into different clusters, and the minimum-volume covering ellipsoid of each cluster is computed and used to choose the next clustering partition. In these schemes, the fast computation of minimum-volume covering ellipsoids is critical to the computational success of the clustering method because of the large number of times that minimum-volume covering ellipsoids (for different subsets of points) must be computed.

The minimum-volume covering ellipsoid problem has been studied for over 50 years. As early as 1948, John (1948) discussed this problem in his work on optimality conditions. Barnes (1982) provides an algorithm for this problem based on matrix eigenvalue decomposition. Khachiyan and Todd (1993) first used interior-point methods in developing an algorithm and a complexity bound for the closely related maximum-volume inscribed ellipsoid problem (MVIE), together with a linear reduction from MVCE to MVIE; the complexity of their algorithm for finding an ϵ-optimal ellipsoid is

$$O\left(m^{3.5} \ln\left(\frac{mR}{\epsilon}\right) \ln\left(\frac{n \ln R}{\epsilon}\right)\right)$$

arithmetic operations. Here, R is defined such that the convex hull of the given points contains the unit ball centered at 0 and is contained in the concentric ball of a given radius R, and ϵ is a relative measure of nonoptimality.

Nesterov and Nemirovskii (1994) obtain a complexity bound of $O(m^{3.5}\ln((mR)/\epsilon))$ operations, and more recently Khachiyan (1996) has reduced this to $O(m^{3.5}\ln(m/\epsilon))$ operations. Zhang (1998) presents interior-point algorithms for MVIE, based on various equation system reduction schemes. In 2003, Zhang and Gao extended their earlier results and compare different practical algorithms for the maximum-volume inscribed ellipsoid problem. Vandenberghe et al. (1998) and Toh (1999) also consider the minimum-volume ellipsoid problem as a special case of the more general maximum determinant problem.

In contrast to the theoretical work on the MVCE problem, we herein develop a practical algorithm for the problem that is designed to solve very large instances ($m = 30{,}000$ and $n = 30$) such as those that arise in data-mining contexts. We present a combined interior-point and active-set method for solving this problem. Our computational results demonstrate that our method solves these very large problem instances to a high degree of accuracy in under 30 seconds on a personal computer.

This paper is organized as follows. In the rest of this section, we present notation. In §2, we review formulations of the minimum-volume covering ellipsoid problem and issues in solving the problem via available interior-point software. In §3, we present our algorithm for solving the MVCE. In §4, we review dual problem formulations and the conditional gradient method for solving the dual, and in §5, we develop active-set strategies. Computational results are presented in §6. Section 7 discusses some unsuccessful algorithmic approaches that we tried, and §8 contains concluding remarks. The appendix contains proofs, an expanded table of computational results, and results concerning the relationship between our interior-point algorithm and the theory of self-concordance of Nesterov and Nemirovskii (1994).

1.1. Notation

Let $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ denote the convex cone of $n \times n$ symmetric positive semidefinite matrices and symmetric positive definite matrices, respectively. We use $\succeq$ and $\succ$ to denote the partial ordering induced by the cones $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$, respectively. The vector of ones is denoted by $e := (1, 1, \ldots, 1)^T$, where the dimension is dictated by context. The capital letters U and T are used to denote the diagonal matrices whose diagonal entries correspond to the entries of the vectors u and t: $U := \operatorname{diag}(u)$ and $T := \operatorname{diag}(t)$. The Euclidean norm $\sqrt{y^Ty}$ is denoted by $\|y\|$. For a given symmetric positive definite matrix M, define the M-norm by $\|v\|_M := \sqrt{v^TMv}$.

2. Formulations and Solution via Available Interior-Point Software

Our concern is with covering m given points $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$ with an ellipsoid of minimum volume. Let A denote the $n \times m$ matrix whose columns are the vectors $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$:

$$A := [a_1|a_2|\cdots|a_m].$$

To avoid trivialities, we make the following assumption for the remainder of this paper, which guarantees that any ellipsoid containing $a_1, a_2, \ldots, a_m$ has positive volume:

Assumption 1. *The affine hull of $a_1, a_2, \ldots, a_m$ spans $\Re^n$. Equivalently,*

$$\operatorname{rank}\begin{bmatrix} A \\ e^T \end{bmatrix} = n + 1.$$

We point out that in most applications of the minimum-volume covering ellipsoid problem, particularly those in data mining, one cannot presume much in the way of special structure of the data $a_1, a_2, \ldots, a_m$. In particular, the matrix A may be fairly dense, and in all likelihood A^TA, as well as AA^T, will be completely dense.

For $c \in \mathbb{R}^n$ and $Q \in \mathbb{S}^n_{++}$, we define the ellipsoid

$$\mathbf{E}_{Q,c} := \{x \in R^n \mid (x - c)^TQ(x - c) \leqslant 1\};$$

here c is the center of the ellipsoid and Q determines its general shape. The volume of $\mathbf{E}_{Q,c}$ is given by the formula

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)}\frac{1}{\sqrt{\det Q}};$$

see Grötschel et al. (1998) for example. Here, $\Gamma(\cdot)$ is the standard gamma function of calculus.

Under Assumption 1, a natural formulation of the minimum-volume covering ellipsoid problem is

$$(\mathrm{MVCE}^1)\quad \min_{Q,c}\ \det Q^{-1/2} \quad \text{s.t. } (a_i - c)^TQ(a_i - c) \leqslant 1, \quad i = 1, \ldots, m, \quad Q \succ 0.$$

As written, MVCE^1 is not a convex program. By the change of variables

$$M = Q^{1/2} \quad \text{and} \quad z = Q^{1/2}c,$$

we restate the problem as

$$(\mathrm{MVCE}^2)\quad \min_{M,z}\ \psi(M,z) := -\ln\det M \quad \text{s.t. } (Ma_i - z)^T(Ma_i - z) \leqslant 1, \quad i = 1, \ldots, m, \quad M \succ 0, \tag{1}$$

which is now a convex program. If $(\bar{M}, \bar{z})$ is a solution of MVCE^2, we recover the solution of MVCE^1 by setting $(\bar{Q}, \bar{c}) = (\bar{M}^2, \bar{M}^{-1}\bar{z})$.

2.1. Solution via Available Interior-Point Software

MVCE^2 can be rewritten as a log-determinant maximization problem subject to linear equations and second-order cone constraints:

$$\begin{aligned}(\mathrm{MVCE}^3)\quad \min_{M,z,y,w}\quad & -\ln\det M\\ \text{s.t.}\quad & Ma_i - z - y_i = 0,\quad i=1,\ldots,m,\\ & w_i = 1,\quad i=1,\ldots,m,\\ & (y_i,w_i)\in C_2^n,\quad i=1,\ldots,m,\\ & M\succ 0,\end{aligned}$$

where C_2^n denotes the second-order cone $\{(y,w)\in\mathbb{R}^{n+1}\mid \|y\|\leqslant w\}$. The format of MVCE^3 is suitable for a solution using a slightly modified version of the software SDPT3 (see Toh et al. 1999, Tütüncü et al. 2003), where the software is modified to handle the parameterized family of barrier functions

$$B_\theta(M,y,w) := -\ln\det M - \theta\sum_{i=1}^m \ln\bigl(w_i^2 - (y_i)^T(y_i)\bigr) \tag{2}$$

for $\theta > 0$ (and the parameter θ is of course absent from the first term above). However, because the SDPT3 code does not allow for unrestricted variables, the linear equations defining the variables y_i, w_i, $i=1,\ldots,m$, cannot be eliminated, and as a result the Newton step at each iteration must unavoidably form and factorize an $m(n+1)\times m(n+1)$ Schur-complement matrix. Even for only reasonably large values n and m, say $n=10$ and $m=1{,}000$, the computational burden becomes prohibitive.

Similar to Khachiyan (1996), one can construct the following dual problem of MVCE^2 (see §4 for a derivation and further exposition):

$$\begin{aligned}(\mathrm{RD})\quad \max_{\tilde u}\quad & \left(\frac{n}{2}\ln n\right) + \frac{1}{2}\ln\det\begin{pmatrix} A\tilde U A^T & A\tilde u\\ \tilde u^T A^T & e^T\tilde u\end{pmatrix}\\ \text{s.t.}\quad & e^T\tilde u = 1,\\ & \tilde u \geqslant 0.\end{aligned} \tag{3}$$

Note that RD is also a determinant maximization problem subject to linear inequality constraints and can also be solved using a modified version of SDPT3. In solving RD via SDPT3, the computational bottleneck at each iteration lies in forming and factorizing a Schur-complement matrix of size

$$\left(\frac{n^2+3n+4}{2}\right)\times\left(\frac{n^2+3n+4}{2}\right);$$

furthermore, computing each entry of the Schur-complement matrix involves m additions.

To solve large practical instances of the minimum-volume covering ellipsoid problem ($n\geqslant 20$, $m\geqslant 1{,}000$, for example), we develop our own specialized methodology, designed to take explicit advantage of the structure of the problem and the typical instance sizes that one might encounter in practice, particularly in the context of data mining. In §3, we present our basic algorithm, which we call the "dual reduced Newton" (DRN) algorithm; this algorithm is then applied and/or modified to work with active set strategies in §5.

3. Dual Reduced Newton Algorithm

In this section, we describe and derive our basic algorithm for the minimum-volume covering ellipsoid problem; we call this algorithm the "dual reduced Newton" algorithm for reasons that will soon be clear.

3.1. Newton Step

By adding a logarithmic barrier function to the problem formulation MVCE^2, we obtain the formulation

$$\begin{aligned}(\mathrm{MVCE}^2_\theta)\quad \min_{M,z,t}\quad & -\ln\det M - \theta\sum_{i=1}^m \ln t_i\\ \text{s.t.}\quad & (Ma_i - z)^T(Ma_i - z) + t_i = 1,\\ & \qquad\qquad i=1,\ldots,m,\\ & M\succ 0,\\ & t > 0.\end{aligned}$$

The parameterized solutions to this problem as θ varies in the interval $(0,\infty)$ define the central trajectory of the problem MVCE^2. Identifying dual multipliers u_i, $i=1,\ldots,m$, with the equality constraints in MVCE$^2_\theta$, the optimality conditions for (MVCE$^2_\theta$) can be written as

$$\sum_{i=1}^m u_i\bigl[(Ma_i - z)a_i^T + a_i(Ma_i - z)^T\bigr] - M^{-1} = 0, \tag{4}$$

$$\sum_{i=1}^m u_i(z - Ma_i) = 0, \tag{5}$$

$$(Ma_i - z)^T(Ma_i - z) + t_i = 1,\quad i=1,\ldots,m, \tag{6}$$

$$Ut = \theta e, \tag{7}$$

$$u, t \geqslant 0, \tag{8}$$

$$M \succ 0. \tag{9}$$

We could attempt to solve (4)–(9) for (M,z,t,u) directly by using Newton's method, which would necessitate forming and factorizing an

$$\left(\frac{n(n+3)}{2} + 2m\right)\times\left(\frac{n(n+3)}{2} + 2m\right)$$

matrix. However, as we now show, the variables M and z can be directly eliminated, and further analysis will result in only having to form and factorize a single $m\times m$ matrix.

To see how this is done, note that we can solve (5) for z and obtain

$$z = \frac{MAu}{e^T u}. \tag{10}$$

Substituting (10) into (4), we arrive at the following equation for the matrix M:

$$\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)M + M\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right) = M^{-1}. \tag{11}$$

The following proposition, whose proof is in the appendix, demonstrates an important property of the matrix arising in (11):

Proposition 2. *Under Assumption* 1, *if* $u > 0$, *then* $(AUA^T - Auu^TA^T/e^Tu) \succ 0$.

The following remark presents a closed-form solution for the equation system (11); see Lemma 4 of Zhang and Gao (2003):

Remark 3. For a given $S \succ 0$, $X := S^{-1/2}$ is the unique positive definite solution of the equation system

$$\frac{1}{2}(X^TS + SX) = X^{-1}.$$

Utilizing Proposition 2 and Remark 3, the unique solution of (11) is easily derived:

$$M := M(u) := \left[2\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\right]^{-1/2}, \tag{12}$$

and substituting (12) into (10), we conclude:

Proposition 4. *Under Assumption* 1, *if* $u > 0$, *then the unique solution of* (4), (5), *and* (9) *in* M, z *is given by*

$$M := M(u) := \left[2\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\right]^{-1/2} \tag{13}$$

and

$$z := z(u) := \frac{\left[2\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\right]^{-1/2} Au}{e^Tu}. \tag{14}$$

Substituting (13) and (14) into the optimality conditions (4)–(9), we can eliminate the variables M and z explicitly from the optimality conditions, obtaining the following reduced optimality conditions involving only the variables (u, t):

$$\begin{aligned} &h(u) + t = e, \\ &Ut = \theta e, \\ &u, t \geqslant 0, \end{aligned} \tag{15}$$

where $h_i(u)$ is the following nonlinear function of u:

$$\begin{aligned} h_i(u) &:= (M(u)a_i - z(u))^T(M(u)a_i - z(u)) \\ &= \left(a_i - \frac{Au}{e^Tu}\right)^T \left[2\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\right]^{-1} \\ &\quad \cdot \left(a_i - \frac{Au}{e^Tu}\right), \end{aligned} \tag{16}$$

for $i = 1, \ldots, m$, where $M(u)$ and $z(u)$ are specified by (13) and (14).

For a given value of the barrier parameter θ, we will attempt to approximately solve (15) using Newton's method. Let $\nabla_u h(u)$ denote the Jacobian matrix of $h(u)$. The Newton direction $(\Delta u, \Delta t)$ for (15) at the point (u, t) is then the solution of the following system of linear equations in $(\Delta u, \Delta t)$:

$$\begin{aligned} &\nabla_u h(u)\Delta u + \Delta t = r_1 := e - t - h(u), \\ &T\Delta u + U\Delta t = r_2 := \theta e - Ut. \end{aligned} \tag{17}$$

This system will have the unique solution

$$\begin{aligned} &\Delta u = (\nabla_u h(u) - U^{-1}T)^{-1}(r_1 - U^{-1}r_2), \\ &\Delta t = U^{-1}r_2 - U^{-1}T\Delta u, \end{aligned} \tag{18}$$

provided we can show that the matrix $(\nabla_u h(u) - U^{-1}T)$ is nonsingular.

To implement the above methodology, we need to explicitly compute $\nabla_u h(u)$, and we also need to show that $(\nabla_u h(u) - U^{-1}T)$ is nonsingular. Towards this end, we define the following matrix function:

$$\Sigma(u) := \left(A - \frac{Aue^T}{e^Tu}\right)^T M^2(u) \left(A - \frac{Aue^T}{e^Tu}\right) \tag{19}$$

as a function of the dual variables u. Let $A \circ B$ denote the Hadamard product of the matrices A, B, namely $(A \circ B)_{ij} := A_{ij}B_{ij}$ for $i, j = 1, \ldots, m$. The following result conveys an explicit formula for $\nabla_u h(u)$ and also demonstrates other useful properties.

Proposition 5. *Under Assumption* 1,
(i) $\nabla_u h(u) = -2(\Sigma(u)/e^Tu + \Sigma(u) \circ \Sigma(u))$,
(ii) $\nabla_u h(u) \preceq 0$, *and*
(iii) $h(u) = \mathrm{diag}(\Sigma(u))$.

The proof of this proposition is presented in the appendix. From part (ii) of Proposition 5 and the fact that $U^{-1}T \succ 0$ whenever $u, t > 0$, we then have

Corollary 6. *Under Assumption* 1, *if* $u > 0$ *and* $t > 0$, *then* $(\nabla_u h(u) - U^{-1}T) \prec 0$, *and so is nonsingular.*

Now let us put all of this together. To compute the Newton direction $(\Delta u, \Delta t)$ for the reduced optimality conditions (15) at a given point (u, t), we compute according to the following procedure:

Procedure DRN-DIRECTION(u, t, θ): Given (u, t) satisfying $u, t > 0$ and given $\theta \geqslant 0$, *Step* 1. Form and factorize the matrix

$$M^{-2}(u)=\left[2\left(AUA^T-\frac{Auu^TA^T}{e^Tu}\right)\right].$$

Step 2. Form the matrix

$$\Sigma(u)=\left(A-\frac{Aue^T}{e^Tu}\right)^T M^2(u)\left(A-\frac{Aue^T}{e^Tu}\right).$$

Step 3. Form

$$\nabla_u h(u)=-2\left(\frac{\Sigma(u)}{e^Tu}+\Sigma(u)\circ\Sigma(u)\right)$$

and factorize $(\nabla_u h(u)-U^{-1}T)$.

Step 4. Solve (18) for $(\Delta u,\Delta t)$.

The computational burden of each of the four steps in Procedure DRN-DIRECTION is dominated by the need to factorize the matrices in Steps (1) and (2) above. The matrix $(AUA^T-Auu^TA^T/e^Tu)$ in Step (1) is $n\times n$; it requires mn^2 operations to form and n^3 steps to factorize, while the matrix $(\nabla_u h(u)-U^{-1}T)$ in Step (4) is $m\times m$; it requires nm^2 steps to form and m^3 steps to factorize.

The direction $(\Delta u,\Delta t)$ given in Procedure DRN-DIRECTION is the solution to the linearization of the reduced optimality conditions (15), which were derived from the original optimality conditions (4)–(9) of MVCE_θ^2. We call $(\Delta u,\Delta t)$ the DRN direction for "dual reduced Newton." The reason for this is that we started with the optimality conditions (4)–(9), and reduced them by eliminating the primal variables M, z before linearizing the resulting equation system in defining the Newton step.

Note that MVCE_θ^2 is itself an optimization problem of a self-concordant barrier function; see Nesterov and Nemirovskii (1994). Because the theory of self-concordant functions is essentially a theory for Newton-based algorithms, it is natural to ask whether or not the Newton direction $(\Delta u,\Delta t)$ given in Procedure DRN-DIRECTION is the Newton direction for minimizing some self-concordant function. In the appendix, we give a negative answer to this question. However, we show that the Newton direction $(\Delta u,\Delta t)$ given in Procedure DRN-DIRECTION is the Newton direction for the minimization of a function that is "almost" self-concordant.

Note from (10) that $c=M^{-1}z=Au/e^Tu$, which states that the center of the optimal ellipsoid is a convex weighting of the points $a_1,\dots,a_m$, with the weights being the normalized dual variables u_i/e^Tu, $i=1,\dots,m$. It is also easy to see that when $\theta=0$, the complementarity condition $u_it_i=\theta=0$ has a nice geometric interpretation: A point has positive weight u_i only if it lies on the boundary of the optimal ellipsoid. These observations are well-known. Another property is that if one considers the points $a_1,\dots,a_m$ to be a random sample of m i.i.d. random vectors, then with $u:=e/m$ we have that

$$M^{-2}(u)=\frac{2}{m}\left(A-\frac{Aee^T}{m}\right)\left(A-\frac{Aee^T}{m}\right)^T$$

is proportional to the sample covariance matrix.

3.2. Algorithm DRN

Based on the Newton step procedure outlined in §3.1, we construct the following basic interior-point algorithm for solving the MVCE_θ^2 formulation of the minimum-volume covering ellipsoid problem. We name this algorithm "DRN" for dual reduced Newton algorithm.

Algorithm DRN

Step 0. Initialization. Set $r\leftarrow 0.99$. Choose initial values of (u^0,t^0) satisfying $u^0,t^0>0$. Set $(u,t)\leftarrow(u^0,t^0)$.

Step 1. Check Stopping Criteria. $\text{OBJ}:=-\ln\det[M(u)]$. If $\|e-h(u)-t\|\leqslant\epsilon_1$ and $(u^Tt)/\text{OBJ}\leqslant\epsilon_2$, STOP. Return u, $Q:=[M(u)]^2$, $c:=[M(u)]^{-1}z(u)$ and OBJ.

Step 2. Compute Direction. Set $\theta\leftarrow(u^Tt)/10m$. Compute $(\Delta u,\Delta t)$ using Procedure DRN-DIRECTION(u,t,θ).

Step 3. Step-Size Computation and Step. Compute $\bar\beta\leftarrow\max\{\beta\mid(u,t)+\beta(\Delta u,\Delta t)\geqslant 0\}$ and $\tilde\beta\leftarrow\min\{r\bar\beta,1\}$. Set $(u,t)\leftarrow(u,t)+\tilde\beta(\Delta u,\Delta t)$. Go to *Step* 1.

The algorithm is initialized in Step 0. Strategies for choosing (u^0,t^0) are discussed immediately below in §3.3. The quantity $r<1$ is used to keep the iterate values of (u,t) strictly positive; see Step 3. The stopping criteria are checked in Step 1; the tolerances are ϵ_1 for feasibility and ϵ_2 for optimality. We discuss the stopping criteria in a bit more detail below in §3.4. In Step 2 the barrier parameter θ is updated and the DRN direction is computed; similar to standard interior-point methods for conic optimization, we use a rather aggressive shrinking factor of 10 when updating θ at each iteration. In Step 3 we compute the step size using a standard min-ratio test augmented by a fraction $r\in(0,1.0)$ that keeps the new iterate values of (u,t) strictly positive. We found that setting $r=0.99$ tended to work best.

3.3. Initialization Strategies

One way to start Algorithm DRN is to choose any pair (u^0,t^0) that satisfies $u^0,t^0>0$, for example $(u^0,t^0)=(\alpha e,\alpha e)$ for some suitable positive scalar α. However, we found it preferable to choose (u^0,t^0) in a way that guarantees the initial primal feasibility of $M(u^0)$, $z(u^0)$. We start by setting $u^0=(n/2m)e$ (where the factor $n/2m$ was chosen empirically). We then compute $M(u^0)$, $z(u^0)$ via (13) and (14) and test for strict primal feasibility by checking if $h(u^0)\leqslant(0.95)e$, and if so we set $t^0=e-h(u^0)>0$, thus ensuring positivity of (u^0,t^0) as well as initial feasibility of the equations $h(u)+t=e$ at $(u,t)=(u^0,t^0)$. If $h(u^0)\nleqslant(0.95)e$, observe that because $h(\alpha u)=h(u)/\alpha$ from (16), we can rescale u^0 to ensure strict feasibility of the algorithm as follows: compute

$$\alpha=\frac{\max\{h_1(u^0),\dots,h_m(u^0)\}}{0.95},\qquad u^0\leftarrow\alpha u^0,\qquad t^0\leftarrow e-h(u^0).\tag{20}$$

This initialization strategy then guarantees strict positivity of (u^0, t^0) as well as initial feasibility of the equations $h(u) + t = e$ at $(u, t) = (u^0, t^0)$.

3.4. Stopping Criteria

The following result is the basis of the stopping criteria of Algorithm DRN, where recall from (1) that $\psi(M, z) = -\ln\det M$:

Proposition 7. *Under Assumption* 1, *suppose that* $u > 0$. *If* $h(u) \leqslant e$, *then* $(M, z) := (M(u), z(u))$ *is feasible for* MVCE^2 *and* $\psi(M, z) - u^T t$ *is a lower bound on the optimal objective function value of* MVCE^2.

Proposition 7 states that the optimality gap of a feasible solution $(M, z) := (M(u), z(u))$ of MVCE^2 is at most $u^T t$, where $t = e - h(u) \geqslant 0$. The stopping criteria of Algorithm DRN, specified in Step 1, is to check that primal feasibility is satisfied to a prespecified tolerance ϵ_1, and then to check if the relative optimality gap is not greater than ϵ_2, where ϵ_2 is the prespecified tolerance. In our computational tests, we set $\epsilon_1 = \epsilon_2 = 10^{-7}$. However, in practical applications in data mining where optimal objective function values are desirable but not critical, we might expect the optimality tolerance to be on the order of $\epsilon_2 = 10^{-4}$, for example.

4. Dual Formulations and the Conditional Gradient Method for the Dual

4.1. Dual Formulations

Using standard Lagrangean duality constructs, one can construct the following dual problem of MVCE^2:

$$\begin{aligned}\max_{u}\quad & \phi(u) := \left(\frac{n}{2}\ln 2 + \frac{n}{2}\right) + \frac{1}{2}\ln\det\left[AUA^T - \frac{Auu^TA^T}{e^Tu}\right] - e^Tu \qquad (21)\\ \text{s.t.}\quad & u \geqslant 0,\end{aligned}$$

and $\phi(u) \leqslant \psi(M, z)$ for all u and (M, z) feasible for (1) and (21), respectively.

Using the fact that

$$\det\left[AUA^T - \frac{Auu^TA^T}{e^Tu}\right] = \det\begin{pmatrix} AUA^T & Au \\ u^TA^T & e^Tu \end{pmatrix} \cdot \frac{1}{e^Tu},$$

we can rewrite (21) as

$$\begin{aligned}\max_{u}\quad & \left(\frac{n}{2}\ln 2 + \frac{n}{2}\right) + \frac{1}{2}\ln\det\begin{pmatrix} AUA^T & Au \\ u^TA^T & e^Tu \end{pmatrix} - \frac{1}{2}\ln(e^Tu) - e^Tu \qquad (22)\\ \text{s.t.}\quad & u \geqslant 0.\end{aligned}$$

If we then rewrite u as $u = \tilde{\lambda}\tilde{u}$, where $\tilde{\lambda} = e^Tu$, and $\tilde{u} \in \mathbb{R}^m_+$, we can further rewrite (22) as follows:

$$\begin{aligned}\max_{\tilde{\lambda},\,\tilde{u}}\quad & \left(\frac{n}{2}\ln 2 + \frac{n}{2}\right) + \frac{n+1}{2}\ln\tilde{\lambda} + \frac{1}{2}\ln\det\begin{pmatrix} A\tilde{U}A^T & A\tilde{u} \\ \tilde{u}^TA^T & e^T\tilde{u} \end{pmatrix} - \frac{1}{2}\ln\tilde{\lambda} - \tilde{\lambda} \qquad (23)\\ \text{s.t.}\quad & e^T\tilde{u} = 1,\\ & \tilde{\lambda} \geqslant 0, \quad \tilde{u} \geqslant 0.\end{aligned}$$

Gathering terms in the objective function of (23) and optimizing with respect to $\tilde{\lambda}$ yields $\tilde{\lambda} = n/2$, which when substituted yields the refined dual problem RD of (3). We refer to RD as a refinement of (23) because (23) has been optimized with respect to the scalar variable $\tilde{\lambda}$. The dual problem RD is well-known in the study of minimum-volume covering ellipsoids; see Khachiyan (1996), for example. Furthermore, the D-optimal experimental design problem can be formulated as an instance of RD (see, for example, Vandenberghe et al. 1998); and hence the computational methods developed herein are applicable to large-scale experimental design problems.

Taking the Lagrangean dual of RD and further refining the resulting minimization problem yields the following problem:

$$\begin{aligned}\text{(PL)}\quad \min_{Y}\quad & \left(\frac{n}{2}\ln n\right) - \left(\frac{n+1}{2}\ln(n+1)\right) - \frac{1}{2}\ln\det Y\\ \text{s.t.}\quad & \begin{pmatrix} a_i \\ 1 \end{pmatrix}^T [Y] \begin{pmatrix} a_i \\ 1 \end{pmatrix} \leqslant 1, \quad i = 1, \ldots, m, \qquad (24)\\ & Y \in \mathbb{S}^{n+1}_{++}.\end{aligned}$$

Problem (PL) seeks to find the minimum-volume ellipsoid in $\mathbb{R}^{n+1}$ centered at the origin that contains the lifted points $(a_i, 1)^T$ for $i = 1, \ldots, m$, where each point a_i has now been lifted into $\mathbb{R}^{n+1}$ onto the hyperplane $H := \{(x, x_{n+1}) \mid x_{n+1} = 1\}$. Khachiyan and Todd (1993), Khachiyan (1996), and Nesterov and Nemirovskii (1994) propose algorithms for solving minimum-volume covering ellipsoids based on this lifting. The minimum-volume ellipsoid of the original problem is recovered as the intersection of the hyperplane H and the minimum-volume covering ellipsoid centered at the origin containing the lifted points $(a_i, 1)^T$, $i = 1, \ldots, m$.

4.2. The Conditional Gradient Method for Solving RD

Interior-point methods and other second-order methods exhibit computational superiority in a number of settings in convex optimization. However, the work per iteration is necessarily large, which suggests that one might use a first-order method such as the conditional gradient method

(Bertsekas 1999) in the early solution stages. Khachiyan (1996) analyzed the theoretical complexity of a first-order method for the solution of the minimum-volume covering ellipsoid problem via formulation RD. Upon careful examination, the algorithm in Khachiyan (1996) can be interpreted as a version of the conditional gradient method applied to this problem. Here, we restate this algorithm in our notation and interpretation. Let $S^{(m-1)}$ denote the standard simplex in $\mathbb{R}^m$, namely $S^{(m-1)} := \{u \in \mathbb{R}^m \mid u \geqslant 0, e^T u = 1\}$, and let

$$V(u) := \begin{bmatrix} AUA^T & Au \\ u^T A^T & e^T u \end{bmatrix}.$$

Then, problem RD can be cast as $\max_{u\in S^{(m-1)}} \ln\det V(u)$. It is straightforward to derive the partial derivatives of the objective function of RD:

$$g_i(u) := \frac{\partial \ln\det V(u)}{\partial u_i} = (a_i^T \;\; 1)V(u)^{-1}\begin{pmatrix} a_i \\ 1 \end{pmatrix}, \quad i = 1, \ldots, m.$$

Let $\bar{u} \in S^{(m-1)}$ be the current iterate value. At each iteration of the conditional gradient method, we compute the gradient $g(\bar{u}) := (g_1(\bar{u}), \ldots, g_m(\bar{u}))$ of the objective function of RD and solve the subproblem $\max_{u\in S^{(m-1)}} g(\bar{u})^T u$, whose optimal solution is given by the jth unit vector $e_j \in \mathbb{R}^m$, where $j = \arg\max_i g_i(\bar{u})$. The method then requires the solution of the line-search problem

$$\max_{\alpha\in[0,1]} f_{\bar{u},j}(\alpha) := \ln\det V((1-\alpha)\bar{u} + \alpha e_j) = \ln\det\left((1-\alpha)V(\bar{u}) + \alpha\begin{pmatrix} a_j \\ 1 \end{pmatrix}(a_j^T \;\; 1)\right).$$

Khachiyan (1996) cleverly observed that this line-search problem has a closed-form solution, namely

$$\alpha = \frac{g_j(\bar{u}) - (n+1)}{(n+1)(g_j(\bar{u}) - 1)}$$

(see Khachiyan 1996 for details). This leads to the following algorithm:

Algorithm Conditional Gradient

Step 0. Initialization. Choose an initial value of u^0 satisfying $u^0 \geqslant 0$, $e^T u^0 = 1$. Set $u \leftarrow u^0$.

Step 1. Solve Subproblem. Compute

$$g_i(u) = (a_i^T \;\; 1)V(u)^{-1}\begin{pmatrix} a_i \\ 1 \end{pmatrix}, \quad i = 1, \ldots, m.$$

Set $j \leftarrow \arg\max_i g_i(u)$.

Step 2. Step-Size Computation and Step.

$$\alpha \leftarrow \frac{g_j(u) - (n+1)}{(n+1)(g_j(u) - 1)}.$$

$u \leftarrow (1-\alpha)u + \alpha e_j$. Go to Step 1.

The computational effort at each iteration of the conditional gradient method is dominated by the gradient computation, which is $m(n+1)^2$ operations to form and factorize $V(u)$ and another $m(n+1)^2$ operations to then compute $g(u)$.

5. Active-Set Strategies

It is easy to see from the optimality conditions (5)–(9) at $\theta = 0$ that the minimum-volume covering ellipsoid is determined only by points a_i located on its boundary. The following well-known result of John (1948) states that the number of such boundary points is not too large:

REMARK 8. The minimum-volume covering ellipsoid is determined by a subset of at most $(n^2 + 3n)/2$ points (John 1948).

This motivates the design of active-set strategies for solving MVCE^2, wherein we try to make an intelligent guess of active points a_i at each iteration, and we discard presumably inactive points from time to time. Let $\mathcal{M}$ denote the set of points that must be covered, namely $\mathcal{M} := \{a_1, \ldots, a_m\}$, and consider the following active-set method:

Generic Active-Set Method

Step 0. Initialization. Define some initial active set of points $\mathcal{S}_0 := \{a_{i_1}, a_{i_2}, \ldots, a_{i_l}\}$ for which $\mathcal{S}_0$ satisfies Assumption 1. Define an initial starting point u_0^0. $k \leftarrow 0$.

Step 1. Solve MVCE^2 for the Set of Points $\mathcal{S}_k$. Use Algorithm DRN with starting point u_0^k. Let $(\bar{u}^k, Q_k, c_k)$ be the output returned.

Step 2. Check Feasibility. If $\|a_j - c_k\|_{Q_k} \leqslant 1 + \epsilon_1$ for $i \in \mathcal{M}\backslash\mathcal{S}_k$, stop. Return $(u, Q, c) := (\bar{u}_k, Q_k, c_k)$. Otherwise go to Step 3.

Step 3. Update Active Set. Update the active set to $\mathcal{S}_{k+1}$. Determine a new starting point u_0^{k+1}. $k \leftarrow k+1$. Go to Step 1.

To implement the above framework, we need to address the following specific questions: how to determine the initial active set $\mathcal{S}_0$ and the initial starting point u_0^0, how to update the active set from iteration to iteration, and how to choose the starting point u_0^k for all subsequent iterations.

5.1. Initial Active Set

One naïve approach is to randomly choose $l \geqslant n+1$ points that satisfy Assumption 1. Not surprisingly, this method is inefficient in practice. Also, linear programming could be used to test and permanently eliminate all points a_i that lie in the convex hull of $\mathcal{M}\backslash\{a_i\}$. This also is inefficient.

We concentrated on developing heuristics for determining which points a_i are "far away" from the "center" of the data. We developed two main active-set initialization schemes that we call Sample Covariance Initialization (SCI) and Conditional Gradient Initialization (CGI), both of which we now describe.

5.1.1. Sample Covariance Initialization (SCI). Following (12), the matrix $M^{-2}(e)$ is proportional to the sample covariance matrix

$$\frac{1}{m-1}\left[AA^T - \frac{Aee^TA^T}{m}\right]$$

of the datapoints $a_1, \ldots, a_m$. The inverse of the sample covariance matrix can serve as a reasonable initial guess of the shape matrix of the covering ellipsoid and its induced norm $\|\cdot\|_{Q(e)}$, where $Q(e) := M^2(e)$ can serve as a natural initial distance metric to determine which points are far from the sample mean $\bar{a} := (Ae)/m$. Following this idea, we define the initial active set to contain m_0 points whose distances from the sample mean $d_i := \|a_i - \bar{a}\|_{Q(e)}$ are the largest. To determine the cardinality of the initial set m^0, we need to trade off small size (for faster computation) against quality of information (which improves for larger m_0). We found that $m_0 := \min\{n^{1.5}, m\}$ worked well in practice. The computational burden of the SCI scheme is $O(mn^2)$ operations.

5.1.2. Conditional Gradient Initialization (CGI). The strategy in the CGI scheme is to run the conditional gradient algorithm for a small number of iterations starting at the barycenter $u = e/m$ of $S^{(m-1)}$. At each iteration, we record the point a_j whose index j gave rise to the maximum partial derivative $g_j(u)$ at that iteration; see Step 1 of the algorithm in §4.2. We accumulate these points to form the initial active set $\mathscr{S}_0$. Although this method tended to produce initial active sets that were superior to those produced by the SCI scheme, the computational effort of this method is much greater than for SCI. Each conditional gradient step needs $O(mn^2)$ operations (which is the same as the entire SCI scheme), and running the method for l steps, then, is $O(lmn^2)$ operations. To satisfy Assumption 1, we need to have at least $n+1$ affinely independent points in the initial active set to have a full-dimensional ellipsoid, and so we must set $l \geqslant n+1$. Because of this, we chose $l = n+1$ as the number of conditional gradient steps to run.

We compare the computational performance of SCI versus CGI in §6.

5.2. Determining u_0^k

We first discuss the issue of determining u_0^0. If the initial active set $\mathscr{S}_0$ is chosen via SCI, we set $(u_0^0)_i$ proportional to the distance $d_i := \|a_i - \bar{a}\|_{Q(e)}$ for $i \in \mathscr{S}_0$, normalizing so that $e^T(u_0^0) = n/2$. If the initial active set is chosen via CGI, we set $(u_0^0)_i$ proportional to the output values u_i of the conditional gradient algorithm for $i \in \mathscr{S}_0$, normalizing so that $e^T(u_0^0) = n/2$. The reason for this normalization is that it follows from (22) that u optimal for (22) must satisfy $e^Tu = n/2$.

We now discuss how u_0^k is determined for $k \geqslant 1$. At the end of the previous active-set step, Algorithm DRN has computed $\bar{u}_k$ for the active set $\mathscr{S}_k$. If the active set has just been expanded so that $\mathscr{S}_{k+1} = \mathscr{S}_k \cup \Delta\mathscr{S}$, we set u_0^{k+1} to be a combination

$$\alpha\begin{pmatrix}\bar{u}_k\\0\end{pmatrix} + (1-\alpha)\begin{pmatrix}0\\\bar{d}\end{pmatrix},$$

where the indices are partitioned here into $\mathscr{S}_k$ and $\Delta\mathscr{S}$ and $\bar{d}_i = \|a_i - c_k\|_{Q_k}$. We found that $\alpha = 0.75$ worked well in practice. Then, we normalize so that $e^T(u_0^{k+1}) = n/2$.

If the active set has just been shrunk, we simply renormalize $\bar{u}_k$ so that the remaining indices satisfy $\sum_{i\in\mathscr{S}_{k+1}}(u_0^{k+1})_i = n/2$.

5.3. Updating the Active Set

5.3.1. Expanding the Active Set. Suppose that the current active set is $\mathscr{S}_k$ and that we have just run Algorithm DRN on this set, obtaining $(\bar{u}_k, Q_k, c_k)$ as output. We consider expanding the active set to $\mathscr{S}_{k+1} = \mathscr{S}_k \cup \Delta\mathscr{S}$ for some set $\Delta\mathscr{S}$. When we expand the active set, we choose points $a_i \notin \mathscr{S}_k$, whose distances from the current center c_k are largest, using the current ellipsoidal norm to define the distances: $d_i := \|a_i - c_k\|_{Q_k}$. We would like to add a reasonable number of points to the active set whose distances d_i satisfy $d_i \geqslant 1$ (otherwise, a_i would remain inactive in the current active set), and are large. We sort the d_is to determine a priority ranking. The simple strategy of choosing the $l \geqslant 1$ farthest points to enter the active set does not work well because, for example, the second-farthest point may be nearby and dominated by the farthest point. Intuitively, we want the points that we add to the active set to be spread around the current ellipsoid $\mathbf{E}_{Q_k, c_k}$. This is handled in our code as follows: After sorting points according to the d_is and considering only points a_i with $d_i > 1$, if there are fewer than 30 such points we simply include all of them in $\Delta\mathscr{S}$. Otherwise, the first point to be added to $\Delta\mathscr{S}$ is the point a_i with the largest d_i. After that, we examine points one by one in descending order of d_i, and we add a_j to $\Delta\mathscr{S}$ if $\sum_{i\in\Delta\mathscr{S}}(a_j - c_k)^TQ_k(a_i - c_k) < 0$. In this way, the points that wind up in $\Delta\mathscr{S}$ will tend to make larger angles with other points in $\Delta\mathscr{S}$ (measured with respect to the matrix Q_k), and so will hopefully be spread around the ellipsoid $\mathbf{E}_{Q_k, c_k}$.

5.3.2. Shrinking the Active Set. There are several ways to delete points from the current active set with the guarantee that they will not enter the active set again. One way is to use linear programming to test if a given point in the active set lies in the convex hull of the other points in the active set, but this approach is obviously too expensive. Another possibility is to check if any of the points in the active set lie in the inscribed Löwner-John ellipsoid $\mathbf{E}_{(n^2Q_k), c_k}$, which is guaranteed to be contained in the convex hull of the current active set (see John 1948). Checking this is relatively inexpensive, but is not effective in higher dimensions because it simply deletes too few points.

We used the following simple heuristic to delete points: When the cardinality of the active set first reaches $100, 150, 200, \ldots,$ we delete all points a_i whose current distance from the current center (using the current ellipsoidal norm) satisfies $d_i < 0.9999$.

Finite termination of active-set methods typically can be proved under the assumption that each subproblem is solved exactly; see some of the discussions on this in Powell (1982) or Gill and Murray (1974), for example.

6. Computational Results

To perform computational tests, we generated datasets of varying dimension n and number of points m. The datasets were generated using independent random multinomial Gaussian distribution or several Gaussian distributions, to mimic the datapoints from one or more clusters, as might be encountered in practice. All computation was done in MATLAB 6.5.0.180913a Release 13 on a Pentium IV 1.5 GHz PC with 1 GB RAM, running LINUX.

6.1. Small- and Medium-Size Problems

Table 1 shows computational results for the solution of the minimum-volume covering ellipsoid problem on small- and medium-sized problems, namely $4 \leqslant n \leqslant 20$ and $20 \leqslant m \leqslant 500$. We tested three different algorithms: (i) solution via Algorithm DRN described in §3, (ii) solution via formulation RD solved by using a modified version of SDPT3 (modified to handle the parameterized family of barrier functions in (2) with θ absent from the first term), and (iii) solution via formulation MVCE^3 using the same modified version of SDPT3. In Table 1 and elsewhere, we refer to these three approaches simply as DRN, RD-SDPT3, and MVCE^3-SDPT3. All three methods were run on the full problems, i.e., without any active-set methodology. The starting point used for Algorithm DRN was as described in §3.3. We tried a variety of different ways to choose starting points for Algorithms RD-SDPT3 and MVCE^3-SDPT3, but ultimately found no obvious advantage over the default starting-point methodology built into SDPT3. All feasibility and relative duality gap tolerances were set to $\epsilon = 10^{-7}$. The "Iterations" columns in Table 1 show the number of IPM/Newton iterations for each method. Table 1 also shows the geometric means of iterations and solution times. Note that in Table 1 there were two problem instances of size $n = 10$ and $m = 500$ for which MVCE^3-SDPT3 terminated prematurely due to numerical problems.

The first observation from Table 1 is that MVCE^3-SDPT3 has vastly inferior solution times to DRN and to RD-SDPT3. This is almost surely due to the very large Schur-complement matrix $(m(n+1) \times m(n+1))$ that must be formed and factorized to solve MVCE^3 via SDPT3.

The second observation from Table 1 is that DRN needs to take roughly one-half as many Newton steps as RD-SDPT3. Examining the output of SDPT3 in greater detail to assess the reasons for this, we found that, particularly in the first 10 iterations, RD-SDPT3 routinely had slow convergence to primal and/or dual feasibility. (In interior-point codes such as SDPT3, slow convergence to feasibility is indicated by step sizes that are much less than 1.) However, in the last few iterations of RD-SDPT3, the iterates of RD-SDPT3 converged as quickly as for DRN. This probably means that SDPT3 is not as capable of capitalizing on good starting-point information, but it also could mean that the directions produced by DRN are somehow better. Of course, the performance of RD-SDPT3 could potentially improve if a more successful starting-point methodology is found, but so far such a methodology has eluded us even after testing several different approaches.

The computational effort per iteration for DRN is dominated by factorizing and solving an $m \times m$ matrix, whereas for RD-SDPT3 it is dominated by factorizing and solving an

$$\left(\frac{n^2+3n+4}{2}\right) \times \left(\frac{n^2+3n+4}{2}\right)$$

matrix. When $m/n^2 \ll 1/2$, we might expect DRN to dominate RD-SDPT3; when $m/n^2 \gg 1/2$, we might expect RD-SDPT3 to dominate DRN. This intuition is verified by the numbers in Table 2. The rightmost column in Table 2 shows the ratio of the DRN solution times to the RD-SDPT3 solution times. Table 2 shows the trend that the relative advantage of DRN over SDPT3 diminishes as m/n^2 grows. However, for $m/n^2 \leqslant 1/2$, DRN outperforms RD-SDPT3.

6.2. Solving Large Problems Using DRN and Active-Set Strategies

The computational results in §6.1 are for small- to medium-size problems; for larger-sized problems, an active-set strategy is necessary to achieve good computational performance. Recall from Remark 8 that the minimum-volume ellipsoid is determined by at most $(n^2 + 3n)/2$ points. Furthermore, our computational experience indicates that the number of points that determine the minimum-volume ellipsoid tends to be closer to $n^2/4$ in practice. Based on the analysis reported in Table 2, this suggests that Algorithm DRN should be used to solve the active-set subproblems at each major iteration, because its performance is superior to RD-SDPT3 when $m_k/n^2 \leqslant 1/4 < 1/2$, where m_k is the number of points in the active set at iteration k.

Table 3 summarizes the computational performance of Algorithm DRN coupled with the active-set strategy described in §5, for dimensions n and m in the ranges $10 \leqslant n \leqslant 30$ and $1{,}000 \leqslant m \leqslant 30{,}000$, over samples of 10 randomly generated problems. (For completeness, Table 4 in the appendix shows the computational performance measures for every problem solved.) The average performance measures in Table 3 are computed using the geometric mean of the 10 randomly generated problems.

Table 1. Performance of Algorithms DRN, RD-SDPT3, and MVCE^3-SDPT3 on small- and medium-sized problem instances of the minimum-volume covering ellipsoid problem.

Dimensions		Algorithm					
		DRN		RD-SDPT3		MVCE^3-SDPT3	
n	*m*	Iterations	Solution time (seconds)	Iterations	Solution time (seconds)	Iterations	Solution time (seconds)
4	20	10	0.06	18	0.91	12	1.19
4	20	11	0.03	22	0.93	12	1.03
4	20	11	0.03	15*	0.65*	13	1.1
4	20	9	0.02	17	0.75	15	1.28
4	20	10	0.02	17	0.74	10	0.84
4	20	9	0.02	14	0.61	14	1.19
4	20	10	0.02	16	0.69	10	0.86
4	20	10	0.02	18	0.8	13	1.11
4	20	10	0.03	16	0.69	13	1.11
4	20	10	0.02	19	0.82	13	1.1
Geometric mean		9.980	0.0252	17.073	0.7524	12.406	1.072
4	60	11	0.08	18	1.12	12	2.75
4	60	10	0.07	23	1.44	11	2.58
4	60	11	0.08	18	1.12	13	2.99
4	60	12	0.09	19	1.18	13	3.01
4	60	10	0.07	18	1.11	13	2.97
4	60	11	0.07	20	1.25	14	3.17
4	60	12	0.08	16	0.98	12	2.81
4	60	11	0.08	18	1.12	13	2.98
4	60	14	0.1	24	1.48	16	3.64
4	60	11	0.08	16*	1.01*	11	2.56
Geometric mean		11.250	0.0795	18.841	1.171	12.727	2.932
10	200	14	1.46	28	4.96	12	96.89
10	200	13	1.32	26	4.45	14	112.27
10	200	16	1.61	21	3.64	14	112.07
10	200	13	1.32	21	3.64	14	112.42
10	200	15	1.52	26	4.5	13	104.45
10	200	15	1.52	21	3.61	14	112.47
10	200	14	1.41	29	4.97	13	104.79
10	200	15	1.51	24	4.15	13	105.06
10	200	18	1.82	24	4.15	14	112.09
10	200	12	1.22	22	3.79	14	112.14
Geometric mean		14.411	1.462	24.038	4.157	13.483	108.34
10	500	16	19.08	23	14.08	15**	2,008.13**
10	500	16	19.13	27	16.54	16	1,970.17
10	500	15	17.98	23	14.13	15	1,869.81
10	500	17	20.35	33	20.17	18	2,281.87
10	500	15	17.95	29	17.68	16	1,985.44
10	500	18	21.51	30	18.38	16	1,985.84
10	500	15	17.91	20	12.29	15**	2,105.83**
10	500	16	19.18	19	11.68	14	1,726.04
10	500	17	20.32	25	15.36	15	1,838.34
10	500	16	19.16	28	17.05	15	1,890.58
Geometric mean		16.073	19.225	25.338	15.52	15.468	1,960.957
20	500	16	19.46	35	29.25		
20	500	17	20.75	21	17.66		
20	500	15	18.28	29	24.31		
20	500	15	18.69	27	22.64		
20	500	14	18.03	29	24.36	OUT OF MEMORY	
20	500	17	21.99	30	25.1		
20	500	14	18.17	29	24.33		
20	500	16	20.79	27	22.6		
20	500	15	19.47	31	25.95		
20	500	16	20.79	36	30.11		
Geometric mean		15.466	19.599	29.114	24.396		

*Indicates premature termination of SDPT3 with termination code "Stop: progress is too slow."

**Indicates premature termination of SDPT3 with termination code "Stop: failure to solve the Schur-complement equation by using the Sherman-Morrison update."

Table 2. Geometric mean of solution times of Algorithms DRN and RD-SDPT3 as a function of the dimensions for random samples of 10 problems.

			Geometric mean of solution time (seconds)		Ratio $\frac{\text{DRN}}{\text{RD-SDPT3}}$
n	m	m/n^2	DRN	RD-SDPT3	
10	50	0.5	0.062	1.221	0.051
10	100	1	0.224	1.933	0.116
10	200	2	1.335	3.827	0.349
10	400	4	9.294	10.589	0.878
10	600	6	30.911	17.826	1.734
10	800	8	78.889	30.482	2.588
20	200	0.5	1.361	6.98	0.195
20	300	0.75	4.067	11.047	0.368
20	400	1	9.104	19.381	0.470
20	600	1.5	30.393	34.774	0.874
20	800	2	74.428	60.157	1.237
20	1,000	2.5	153.018	95.412	1.604
20	1,200	3	288.510	143.678	2.008
30	450	0.5	13.571	33.813	0.401
30	900	1	149.895	114.429	1.310
30	1,350	1.5	408.515	270.31	1.511
30	1,800	2	1,012.718	478.09	2.118

The table presents results using the two initialization schemes SCI and CGI that were described in §§5.1.1 and 5.1.2. The "Iterations" columns report the number of outer iterations, that is, the number of different subproblems solved, and the "Final Active Set" columns report the number of points present in the last active-set subproblem. (Note that the active set is the current working set of points, as opposed to the set of points that lie on the boundary of the optimal ellipsoid, which we call the set of "binding points.") The "Initialization Time" columns report the time taken by the algorithm to initialize the active set using the CGI and the SCI initialization schemes. The "Total Solution Time" columns report the total time to solve the problems. As before, all subproblems were solved to a feasibility tolerance and a relative duality gap tolerance of $\epsilon = 10^{-7}$. Note that the Final Active Set numbers are different for the two initialization schemes. This reflects the fact that the two initialization schemes start with different active sets, and hence terminate with different active sets as well.

The table indicates that SCI dominates CGI in terms of Total Solution Time, becoming more advantageous for larger problem dimensions. This is probably due to the fact that CGI requires roughly mn^3 operations as opposed to mn^2 for SCI, but also it appears from the numbers in Table 3 that the active set generated by CGI is just not as good, as evidenced by the fact that if the initialization times are subtracted from the total times, then SCI still wins by a large margin, especially on the larger problems.

The Total Solution Times reported in Table 3 for the largest problems ($n = 30$, $m = 30{,}000$) clearly indicate that Algorithm DRN coupled with a suitable active-set strategy solves these problems to a high degree of accuracy ($\epsilon_1 = \epsilon_2 = 10^{-7}$) in well under 30 seconds on a personal computer.

7. Some Unsuccessful Strategies

In developing Algorithm DRN and the active-set strategies discussed in §5, we tested and discarded a wide variety of computational strategies in favor of more efficient methods. Some of these strategies were mentioned in §5; here is a short list of other strategies that were not previously discussed.

• *Directly Linearizing the Optimality Conditions*. The direction computed by Algorithm DRN at each iteration is the result of first eliminating the variables M, z from Equations (4)–(9) and then linearizing the remaining nonlinear system, finally yielding a system of linear equations of dimension m. An alternative strategy is to directly linearize Equations (4)–(9) and then eliminate variables t and u and solve the remaining system of linear equations for M, z. Using this strategy, one would only need to factorize a matrix of dimension $(n(n+3))/2$, which should be less than m. Unfortunately, we found that forming the matrix of the resulting linear system required roughly m^2n^4 operations, which dominates the factorization procedure for each Newton step. In computational tests, we

Table 3. Summary performance of Algorithm DRN with an active-set strategy using CGI and SCI initialization schemes on large-problem instances of the minimum-volume covering ellipsoid problem for random samples of 10 problems.

Dimensions		CGI				SCI			
n	m	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)
20	1,000	7.49	75.63	0.15	1.2	6.28	75.24	0.01	1.63
10	10,000	5.82	37.92	0.35	0.88	8.98	39.19	0.05	1.13
20	10,000	11.77	112.57	1.2	4.7	9.79	111.95	0.08	4.1
20	20,000	10.53	97.97	2.57	5.52	11.92	101.55	0.18	5.4
20	30,000	11.36	115.21	3.48	7.76	12.51	106.16	0.23	6.36
30	10,000	15.4	201.69	3.34	16.86	12.66	193.14	0.15	14.96
30	20,000	15.71	213.87	6.9	23.91	12.4	214.48	0.29	18.22
30	30,000	16.69	201.59	10.08	28.83	13.44	195.18	0.44	19.82

found that not only is the overall computation time much slower than that of Algorithm DRN, but the number of Newton iterations was consistently higher as well, suggesting that the directions produced by this method are inferior to those of Algorithm DRN.

• *Expanding the Active Set.* The strategy for expanding the active set was described in detail in §5.3.1, where we described how points are chosen to be added to the active set based on the idea of having the new points be far from the center c_k as measured in the ellipsoidal distance d_i for each point a_i (see §5.3.1 for details) and also spread out in all directions around the ellipsoid. This was originally accomplished by examining points one by one in descending order of d_i, and then including a_j into $\Delta\mathcal{S}$ if $(a_j - c_k)^T Q_k (a_i - c_k) < 0$ for all $i \in \Delta\mathcal{S}$. However, computational testing revealed that this resulted in too few points being added at each outer iteration.

• *Hybrid of SCI and CGI.* We tested several hybrids of SCI and CGI. In one approach, SCI was used to obtain $m/3$ points, and then these points were used to initialize the conditional gradient algorithm, which would then be run for $l = n+1$ iterations to then produce the initial active set $\mathcal{S}_0$ and u_0^0. The motivation for this strategy was that this would essentially cut the computation time of CGI by one-third. Another idea that we tested was to start the CGI from a point u_0 whose ith component was proportional to the distance from a_i to the sample mean $\bar{a} := (1/m)Ae$. Neither of these approaches proved to be very effective.

• *Conditional Gradient Algorithm for the Solution of RD.* Khachiyan (1996) provides the best-known complexity bound for solving the minimum-volume covering ellipsoid problem, and his method can be interpreted as the conditional gradient algorithm applied to the formulation RD. However, as one might expect, this algorithm is not effective in practice.

8. Concluding Remarks

Algorithms and associated software for conic formulations of convex optimization problems that use primal-dual interior-point methods are intended for general convex problems presented in such conic format. While these algorithms generally perform well in practice, they are not designed to be able to consider any special structure of certain classes of problems, such as the minimum-volume covering ellipsoid problem. Herein, we have presented Algorithm DRN for solving the minimum covering ellipsoid problem, which is itself an interior-point-type algorithm, and which is designed around the optimality conditions of the problem augmented with a logarithmic barrier term, although it does not quite fall into the existing interior-point algorithm theoretical framework of self-concordant functions. We have shown that this algorithm performs very well for problems of moderate size. When the number of points to be covered is large, we show how Algorithm DRN can be used with an active-set strategy (where the active-set strategy is also designed specifically for the minimum-volume covering ellipsoid problem), and we report computational results on large problems that validate the efficiency of these approaches. (Incidentally, we also did some computational testing of algorithms applied to problems of large dimension ($n = 100$, 200, and 500) with a moderate number of points ($m = 1{,}000$). Consistent with the implications discussed in §6.1, we found that DRN outperformed RD-SDPT3 by factors of 3 to 20.)

From a practical point of view, most applications of the minimum-volume ellipsoid are based on the ideal situation in which there are no outliers in the data. To make the minimum-volume ellipsoid problem more amenable in the presence of outliers, it is necessary to explore problem formulations that allow points that lie outside of the ellipsoid, such as in the following problem formulation, which penalizes such points:

$$\begin{aligned}
(\text{MVCEP}) \quad \min_{M,z,\xi} \quad & -\ln\det M + Pe^T\xi \\
\text{s.t.} \quad & (Ma_i - z)^T(Ma_i - z) \leqslant 1 + \xi_i, \quad i = 1,\ldots,m, \\
& \xi \geqslant 0, \\
& M \succ 0,
\end{aligned}$$

in which P is a user-specified penalizing coefficient. Formulation (MVCEP) could also be solved by a slight modification of Algorithm DRN, with the active-set strategy if need be. This formulation has the potential of identifying outliers in the data, which has been an important focus in data mining; see Knorr and Zamar (2001). However, from the point of view of determining a "robust" minimum-volume ellipsoid, MVCEP still has the drawback that the shape of the optimal ellipsoid is potentially determined in part by points that lie outside of the optimal ellipsoid. Future work in this area could include developing formulations and solution methods for this problem that include nonconvex penalizing terms such as $P\sum_{i=1}^m \operatorname{sign}(\xi_i)$.

Appendix

Proofs of Propositions 2, 5, and 7

Proof of Proposition 2. Let $v = U^{1/2}e$. Then, note that

$$\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right) = AU^{1/2}\left[I - \frac{vv^T}{v^Tv}\right]U^{1/2}A^T \succeq 0$$

because $P := [I - vv^T/v^Tv] \succeq 0$ (P is a projection matrix, $P^2 = P$, $P = P^T$). Now suppose that $y^T(AUA^T - Auu^TA^T/e^Tu)y = 0$, $y \neq 0$. This can only happen if $U^{1/2}A^Ty = \theta v$ for some scalar θ, i.e., $U^{1/2}A^Ty = \theta U^{1/2}e$. Therefore, $A^Ty = \theta e$, $y \neq 0$, which violates Assumption 1. □

Proof of Proposition 5. Define $C(u) := M^2(u) = [2(AUA^T - Auu^TA^T/e^Tu)]^{-1}$ and $\tilde{a}_i(u) := a_i - Au/e^Tu$. From the definition of $\Sigma(u)$, we have

$$\sigma_{ij}(u) := [\Sigma(u)]_{ij} = (\tilde{a}_i(u))^T M^2(u)(\tilde{a}_j(u)), \tag{25}$$

and thus $h_i(u) = [\Sigma(u)]_{ii}$ from (20), which shows part (iii) of Proposition 5. To compute $\nabla_u h(u)$, we employ the chain rule. We have

$$\frac{\partial \tilde{a}_i(u)}{\partial u_j} = \frac{Au}{(e^Tu)^2} - \frac{a_j}{e^Tu} = \frac{-\tilde{a}_j}{e^Tu}$$

and

$$\begin{aligned}\frac{\partial C(u)}{\partial u_j} &= -2C(u)\left[a_j a_j^T + \frac{Auu^TA^T}{(e^Tu)^2} - \frac{a_j u^T A^T}{e^Tu} - \frac{Aua_j^T}{e^Tu}\right]C(u)\\ &= -2C(u)\tilde{a}_j(u)(\tilde{a}_j(u))^T C(u).\end{aligned}$$

Now invoke the chain rule on (25):

$$\begin{aligned}\frac{\partial h_i(u)}{\partial u_j} &= \frac{-2}{e^Tu}(\tilde{a}_i(u))^T C(u)\tilde{a}_j(u)\\ &\quad - 2(\tilde{a}_i(u))^T C(u)\tilde{a}_j(u)(\tilde{a}_j(u))^T C(u)\tilde{a}_i(u)\\ &= -2\left(\frac{\sigma_{ij}(u)}{e^Tu} + (\sigma_{ij}(u))^2\right).\end{aligned}$$

Therefore, $\nabla_u h(u) = -2(\Sigma(u)/(e^Tu) + \Sigma(u) \circ \Sigma(u))$, proving part (i). Now note that $\Sigma(u) \succeq 0$, $e^Tu > 0$, and $\Sigma(u) \circ \Sigma(u) \succeq 0$ because the Hadamard product of symmetric positive semidefinite matrices is also symmetric positive semidefinite; see Theorem 7.5.3 of Horn and Johnson (1985). Therefore, $\nabla_u h(u) \preceq 0$, proving part (ii). □

Proof of Proposition 7. Let $(M, z) = (M(u), z(u))$. $M(u) \succ 0$ from Proposition 2, and from (16) we see that $h(u) \leqslant e$ is the same as $(Ma^i - z)^T(Ma^i - z) \leqslant 1$, $i = 1, \ldots, m$, whereby (M, z) is feasible for MVCE^2. Note that u is feasible for the dual problem (21), with duality gap

$$\begin{aligned}\psi(M,z) - \phi(u) &= -\ln\det M - \frac{n}{2}\ln 2 - \frac{n}{2}\\ &\quad - \frac{1}{2}\ln\det\left[AUA^T - \frac{Auu^TA^T}{e^Tu}\right] + e^Tu\\ &= -\frac{n}{2} + e^Tu \qquad \text{(from (13))}\\ &= -\frac{n}{2} + u^Tt + u^Th(u).\end{aligned}$$

If $u^Th(u) = n/2$, then $\psi(M,z) - u^Tt = \phi(u)$, which is a dual-feasible objective function value and hence a lower bound on the optimal objective function value of MVCE^2. We therefore need to show that $u^Th(u) = n/2$, which we do now:

$$\begin{aligned}u^Th(u) &= \sum_{i=1}^m u_i(Ma_i - z)^T(Ma_i - z)\\ &= \sum_{i=1}^m u_i\left(a_i - \frac{Au}{e^Tu}\right)^T M^2\left(a_i - \frac{Au}{e^Tu}\right)\\ &= \sum_{i=1}^m u_i\left(a_i - \frac{Au}{e^Tu}\right)\left(a_i - \frac{Au}{e^Tu}\right)^T \cdot M^2\\ &= \left(A - \frac{Aue^T}{e^Tu}\right)U\left(A - \frac{Aue^T}{e^Tu}\right)^T \cdot M^2\\ &= \left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\cdot M^2 = \frac{1}{2}M^{-2}\cdot M^2 = \frac{n}{2},\end{aligned}$$

where "$\cdot$" denotes the trace operator $A \cdot B = \operatorname{trace}(A^TB) = \sum_{i=1}^n (A^TB)_{ii}$. □

Extended Version of Table 3

Table 4 shows the computational performance measures for all problems solved by Algorithm DRN coupled with the active-set strategy. As discussed in §6.2, the table presents results using the two initialization schemes SCI and CGI, which were described in §§5.1.1 and 5.1.2. The "Iterations" columns report the number of outer iterations, that is, the number of different subproblems solved, and the "Final Active Set" columns report the number of points present in the last active-set subproblem. The "Initialization Time" columns report the time taken by the algorithm to initialize the active set using the CGI and the SCI initialization schemes. The "Total Solution Time" columns report the total time to solve the problems. As before, all subproblems were solved to a feasibility tolerance and a relative duality gap tolerance of $\epsilon = 10^{-7}$.

The DRN Direction Is Not a Newton Direction of a Self-Concordant Function

In this subsection, we show that the (Newton) direction produced by Algorithm DRN is not the Newton direction of a self-concordant function. However, it is the Newton direction of a function that is almost self-concordant.

For a given $u > 0$ and $t > 0$, the DRN direction is given by the formula

$$\Delta u = (\nabla_u h(u) - U^{-1}T)^{-1}(e - \theta U^{-1}e - h(u)); \tag{26}$$

see (18).

Let

$$D_f := \left\{u \in \mathbb{R}^m \;\middle|\; \left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right) \succ 0\right\},$$

and for $u \in D_f$ define the function

$$f(u) := -\frac{1}{2}\ln\det\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right), \tag{27}$$

and recall the definition of $h(u)$ in (16).

Table 4. Performance of Algorithm DRN with an active-set strategy using CGI and SCI initialization schemes on large-problem instances of the minimum-volume covering ellipsoid problem.

Dimensions		CGI				SCI			
n	*m*	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)
20	1,000	8	76	0.17	1.18	7	74	0.02	1.62
20	1,000	6	83	0.15	1.2	5	84	0.01	1.23
20	1,000	7	79	0.15	1.14	8	79	0.02	2.14
20	1,000	9	74	0.15	1.33	10	71	0.01	2.5
20	1,000	6	57	0.14	0.64	5	57	0.01	1.4
20	1,000	9	72	0.15	1.27	7	72	0.01	1.72
20	1,000	6	74	0.13	1	5	75	0.01	1.16
20	1,000	7	72	0.14	1.12	7	72	0.01	1.71
20	1,000	9	71	0.14	1.33	7	71	0.02	1.85
20	1,000	9	107	0.14	2.38	4	106	0.01	1.38
Geometric mean		7.49	75.63	0.15	1.20	6.28	75.24	0.01	1.63
10	10,000	6	29	0.34	0.74	11	32	0.05	1.57
10	10,000	6	37	0.34	0.96	7	41	0.05	1.16
10	10,000	7	52	0.36	1.01	9	52	0.04	1.16
10	10,000	5	33	0.37	0.84	10	37	0.06	1.4
10	10,000	7	34	0.38	0.95	9	63	0.04	1.08
10	10,000	4	34	0.38	0.76	12	34	0.05	1.18
10	10,000	6	41	0.33	1.05	7	33	0.05	0.8
10	10,000	7	50	0.33	1.07	6	42	0.04	0.66
10	10,000	5	47	0.33	0.82	12	38	0.04	2.02
10	10,000	6	30	0.33	0.66	9	30	0.04	0.88
Geometric mean		5.82	37.92	0.35	0.88	8.98	39.19	0.05	1.13
20	10,000	11	97	1.22	3.29	13	116	0.08	4.95
20	10,000	14	123	1.21	6.76	7	119	0.08	3.02
20	10,000	14	127	1.19	4.69	14	116	0.08	6.13
20	10,000	9	98	1.19	3.38	12	106	0.08	4.65
20	10,000	13	119	1.23	5.91	8	118	0.08	3.77
20	10,000	13	141	1.19	6.64	7	119	0.08	3.05
20	10,000	13	137	1.19	6.58	7	129	0.08	3.09
20	10,000	10	88	1.18	3.1	15	81	0.08	5.66
20	10,000	10	87	1.18	3.21	15	97	0.08	5.55
20	10,000	12	125	1.23	5.79	6	128	0.08	2.85
Geometric mean		11.77	112.57	1.20	4.70	9.79	111.95	0.08	4.10
20	20,000	12	97	2.35	5.13	18	96	0.15	7.45
20	20,000	11	92	2.34	5.7	9	92	0.16	4.31
20	20,000	10	92	2.55	5.15	15	84	0.17	5.68
20	20,000	11	84	2.58	5.22	11	105	0.17	4.41
20	20,000	11	98	2.59	5.76	12	106	0.19	5.49
20	20,000	11	92	2.73	5.69	16	104	0.19	6.73
20	20,000	12	118	2.72	6.47	8	112	0.18	4
20	20,000	11	108	2.56	5.53	11	104	0.19	5.5
20	20,000	8	97	2.57	4.92	8	98	0.18	3.71
20	20,000	9	106	2.76	5.74	16	119	0.18	8.62
Geometric mean		10.53	97.97	2.57	5.52	11.92	101.55	0.18	5.40
20	30,000	10	111	3.63	7.09	12	108	0.26	6.32
20	30,000	10	126	3.77	8.29	12	108	0.23	6.78
20	30,000	12	135	3.72	9.34	13	123	0.25	7.38
20	30,000	15	154	3.59	10.94	8	132	0.25	5.14
20	30,000	12	127	3.56	8.62	16	112	0.23	7.54
20	30,000	12	108	3.37	6.97	13	93	0.22	5.04
20	30,000	14	135	3.32	9.98	11	147	0.22	6.79
20	30,000	9	79	3.28	5.52	13	81	0.22	5.84
20	30,000	10	95	3.29	5.98	15	88	0.22	7.01
20	30,000	11	102	3.28	6.64	14	88	0.22	6.29
Geometric mean		11.36	115.21	3.48	7.76	12.51	106.16	0.23	6.36

Table 4. (Continued.)

Dimensions		CGI				SCI			
n	m	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)	Iterations	Final active set	Initialization time (seconds)	Total solution time (seconds)
30	10,000	14	194	2.97	12.63	11	174	0.14	8.98
30	10,000	19	224	2.97	26.83	17	219	0.13	24.95
30	10,000	18	223	3.06	20.67	16	198	0.14	16.13
30	10,000	15	189	3.42	13.2	12	195	0.15	13.05
30	10,000	13	199	3.55	13.72	15	177	0.15	16.76
30	10,000	15	172	3.55	13.43	13	172	0.15	11.51
30	10,000	15	195	3.43	16.02	14	170	0.16	15.22
30	10,000	14	158	3.47	12.94	15	162	0.15	16.66
30	10,000	17	254	3.55	25.05	8	244	0.15	14.78
30	10,000	15	227	3.52	20.94	9	240	0.15	16.46
Geometric mean		15.40	201.69	3.34	16.86	12.66	193.14	0.15	14.96
30	20,000	14	184	7.05	17.62	20	195	0.29	24.19
30	20,000	14	198	6.65	20.09	14	192	0.28	15.95
30	20,000	15	189	6.85	18.89	11	169	0.29	12.45
30	20,000	17	234	6.85	25.05	18	228	0.3	26.17
30	20,000	17	216	6.81	28.61	10	239	0.29	20.02
30	20,000	19	220	6.97	27.79	13	205	0.29	19.18
30	20,000	17	239	6.93	31.68	8	239	0.3	14.44
30	20,000	14	227	6.96	24.6	9	227	0.3	17.13
30	20,000	17	241	6.93	31.7	11	262	0.29	22.86
30	20,000	14	200	7.05	18.58	15	205	0.25	14.76
Geometric mean		15.71	213.87	6.90	23.91	12.40	214.48	0.29	18.22
30	30,000	14	167	8.96	19.5	15	178	0.43	17.96
30	30,000	16	245	10.14	34.7	11	244	0.43	24.54
30	30,000	14	175	8.72	18.58	13	147	0.44	12.28
30	30,000	17	147	10.39	22.06	19	145	0.43	16.19
30	30,000	17	239	10.57	34.78	11	222	0.43	20.64
30	30,000	18	169	10.52	25.7	13	170	0.45	13.65
30	30,000	18	248	10.53	45.20	10	248	0.42	25.08
30	30,000	19	212	10.47	33.21	20	193	0.44	26.96
30	30,000	20	268	10.47	41.96	11	246	0.44	24.69
30	30,000	15	185	10.2	25.39	15	195	0.45	22.69
Geometric mean		16.69	201.59	10.08	28.83	13.44	195.18	0.44	19.82

PROPOSITION 9. *Suppose that $u \in D_f$. Then, $\nabla f(u) = -h(u)$.*

PROOF. This follows by direct but tedious application of the chain rule. □

Our analysis relies on the non-self-concordance of $f(u)$, which is stated in the next proposition, and whose proof appears at the end of this section.

PROPOSITION 10. *$f(u)$ is not a self-concordant function on D_f.*

Now consider the following optimization problem, which is equivalent to the program (21) with a logarithmic barrier term added:

$$(\mathrm{D}^1_\theta) \quad \min_u \; f(u) + e^T u - \theta \sum_{i=1}^m \ln(u_i) \quad \text{s.t.} \; u > 0. \tag{28}$$

At any point $u > 0$, the Newton direction for D^1_θ is given by

$$\tilde{\Delta} u = (\nabla^2 f(u) + \theta U^{-2})^{-1}(-\nabla f(u) - e + \theta U^{-1} e), \tag{29}$$

and note from Proposition 10 that the objective function of D^1_θ is not a self-concordant function for $\theta > 0$ and sufficiently small. The following proposition shows that when u, t satisfy $Ut = \theta e$, then the directions Δu of (26) and $\tilde{\Delta} u$ of (29) are the same.

PROPOSITION 11. *Suppose that $u, t > 0$ and that $Ut = \theta e$. Then, $\Delta u = \tilde{\Delta} u$.*

PROOF. From Proposition 9 we have $\nabla f(u) = -h(u)$ and so $\nabla^2 f(u) = -\nabla_u h(u)$. Substituting these equalities and the hypothesis of this proposition into (29) yields

$$\tilde{\Delta} u = (-\nabla_u h(u) + U^{-1} T)^{-1}(h(u) - e + \theta U^{-1} e) = \Delta u. \quad \square$$

From Proposition 11, we see that at points (u, t) satisfying $Ut = \theta e$, the DRN direction is exactly the Newton direction of the non-self-concordant objective function of D^1_θ for $\theta > 0$ and sufficiently small.

Note that if the DRN direction were instead derived as a "dual only" direction, then it would correspond to the Newton direction of problem D^1_θ for all $u > 0$, and so would correspond to the Newton direction of a non-self-concordant function for all $u > 0$. To see this, let us rewrite Equations (15) as

$$h(u) + \theta U^{-1}e = e, \quad u > 0, \tag{30}$$

and the Newton direction $\bar{\Delta}u$ at a point $u > 0$ for (30) is

$$\bar{\Delta}u = (\nabla_u h(u) - \theta U^{-2})^{-1}(e - h(u) - \theta U^{-1}e).$$

It then follows from Proposition 9 that $\bar{\Delta}u = \tilde{\Delta}u$, and so the "dual only" version of Algorithm DRN is the Newton direction of a non-self-concordant function for $\theta > 0$ and sufficiently small.

Finally, we point out that while $f(u)$ is not a self-concordant function, the function

$$f(u) - \frac{1}{2}\ln(e^T u) = -\frac{1}{2}\ln\det[AUA^T - Auu^T A^T],$$

is known to be a self-concordant function (see Nesterov and Nemirovskii 1994), and so $f(u)$ is very closely related to a self-concordant function.

Proof of Proposition 10. For each $u \in D_f$ and any direction d, define the univariate function $f_{u,d}(\alpha) := f(u + \alpha d)$. Then, $f(\cdot)$ is self-concordant or not, depending on whether the quantity

$$\frac{|f'''_{u,d}(0)|}{(f''_{u,d}(0))^{3/2}}$$

can be bounded independent of u and d; see Nesterov and Nemirovskii (1994). Here we will show that this quantity cannot be bounded, thereby demonstrating that $f(\cdot)$ is not self-concordant. We can write

$$\begin{aligned} f_{u,d}(\alpha) &= -\frac{1}{2}\ln\det\begin{pmatrix} A(U+\alpha D)A^T & A(u+\alpha d) \\ (u+\alpha d)^T A^T & e^T(u+\alpha d) \end{pmatrix} \\ &\quad + \frac{1}{2}\ln(e^T(u+\alpha d)) \\ &= -\frac{1}{2}\ln\det(M + \alpha N) + \frac{1}{2}\ln(e^T u + \alpha e^T d), \end{aligned}$$

where

$$M = \begin{bmatrix} AUA^T & Au \\ u^T A^T & e^T u \end{bmatrix} \quad\text{and}\quad N = \begin{bmatrix} ADA^T & Ad \\ d^T A^T & e^T d \end{bmatrix}.$$

It then follows from the rules of differentiation that

$$f'_{u,d}(0) = -\frac{1}{2}\mathrm{Tr}(M^{-1}N) + \frac{e^T d}{e^T u},$$

$$f''_{u,d}(0) = \frac{1}{2}\mathrm{Tr}(M^{-1}NM^{-1}N) - \left(\frac{e^T d}{e^T u}\right)^2,$$

$$f'''_{u,d}(0) = -\left(\mathrm{Tr}(M^{-1}NM^{-1}NM^{-1}N) - \left(\frac{e^T d}{e^T u}\right)^3\right),$$

where "Tr(B)" denotes the trace of a matrix B.

Now let

$$A = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix}, \quad u = (1 \ \ 1 \ \ 1 \ \ 1 \ \ 1)^T,$$

and $d = (0 \ \ 0 \ \ 0 \ \ \delta \ \ 1-\delta)^T$

for a given scalar δ. Then,

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \delta & \delta \\ 0 & \delta & 1 \end{bmatrix},$$

$$M^{-1}N = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2}\delta & \frac{1}{2}\delta \\ 0 & \frac{1}{5}\delta & \frac{1}{5} \end{bmatrix},$$

and $e^T d/e^T u = 1/5$, and direct substiution of these equalities yields

$$f''_{u,d}(0) = \frac{9}{40}\delta^2 \quad\text{and}\quad f'''_{u,d}(0) = \frac{11}{40}\delta^3 + \frac{3}{50}\delta^2.$$

Therefore, for $\delta > 0$ we have

$$\frac{|f'''_{u,d}(0)|}{(f''_{u,d}(0))^{3/2}} = \frac{(11/40)\delta^3 + (3/50)\delta^2}{(9/40)^{3/2}\delta^3},$$

which goes to $+\infty$ as $\delta \downarrow 0$. □

Acknowledgments

The authors thank Kim Chuan Toh for his assistance with the software SDPT3 and for his guidance on computational matters. The authors also thank Yurii Nesterov for guidance on issues regarding self-concordant functions, and thank Xin Chen for input on aspects of mathematical analysis. The authors are grateful to the referees for their many useful comments and suggestions.

References

Barnes, E. 1982. An algorithm for separating patterns by ellipsoids. *IBM J. Res. Develop.* **26**(6) 759–764.

Bertsekas, D. 1999. *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Croux, C., G. Haesbroeck, P. J. Rousseeuw. 2002. Location adjustment for the minimum volume ellipsoid estimator. *Statist. Comput.* **12**(3) 191–200.

Gill, P. E., W. Murray, eds. 1974. *Numerical Methods for Constrained Optimization*. Academic Press, London, U.K.

Grötschel, M., L. Lovasz, A. Schrijver. 1998. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, Germany.

Han, J., M. Kamber. 2001. *Data Mining, Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA.

Horn, R., C. Johnson. 1985. *Matrix Analysis*. Cambridge University Press, Cambridge, U.K.

John, F. 1948. Extreme problems with inequalities as subsidiary conditions. *Studies and Essays Presented to R. Courant on his 60th Birthday. January 8, 1948*. Wiley Interscience, New York, 187–204.

Khachiyan, L. 1996. Rounding of polytopes in the real number model of computation. *Math. Oper. Res.* **21**(2) 307–320.

Khachiyan, L., M. Todd. 1993. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Math. Programming* **61** 137–159.

Knorr, E., R. Ng, R. Zamar. 2001. Robust space transformations for distance-based operations. *Proc. Seventh ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining*. ACM, New York, 126–135.

Nesterov, Y., A. Nemirovskii. 1994. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, PA.

Powel, M. J. D., ed. 1982. *Nonlinear Optimization 1981*. Academic Press, London, U.K.

Toh, K. 1999. Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. *Computational Optim. Appl.* **14** 309–330.

Toh, K., M. Todd, R. Tütüncü. 1999. Sdpt3—A Matlab software package for semidefinite programming. *Optim. Methods Software* **11** 545–581.

Tütüncü, R., K. Toh, M. Todd. 2003. Solving semidefinite-quadratic-linear programs using SDPT-3. *Math. Programming* **95**(2) 189–217.

Vandenberghe, L., S. Boyd, S. Wu. 1998. Determinant maximization with linear matrix inequality constraints. *SIAM J. Matrix Anal. Appl.* **19**(2) 499–533.

Zhang, Y. 1998. An interior-point algorithm for the maximum-volume ellipsoid problem. Technical report TR98-15, Department of Computational and Applied Mathematics, Rice University, Houston, TX.

Zhang, Y., L. Gao. 2003. On numerical solution of the maximum volume ellipsoid problem. *SIAM J. Optim.* **14**(1) 53–76.