Summary Conclusions: Computation of Minimum Volume Covering Ellipsoids*

Peng Sun† and Robert M. Freund‡
†*Duke University Fuqua School of Business, Chapel Hill, North Carolina*
‡*MIT Sloan School of Management, Cambridge, MA*

***Abstract*— We present a practical algorithm for computing the minimum volume n-dimensional ellipsoid that must contain m given points $a_1, \ldots, a_m \in \mathbb{R}^n$. 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.**

***Index Terms*— Ellipsoid, Newton's method, interior-point method, barrier method, active set, semidefinite program, data mining, robust statistics, clustering analysis.**

I. Introduction

This paper is concerned with computing the minimum volume ellipsoid in n-dimensional space $\mathbb{R}^n$ containing m given points $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$. 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 [3]. In the area of robust statistics and data mining, efficiently finding outliers is a challenge that has attracted much research interest [8]. Indeed, one can identify data outliers quickly if one can compute the minimum volume ellipsoid quickly, since 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 data sets; invariance under linear transformation is important in a multi-dimensional setting where different coefficients have different units of measurement. Traditional distance-based clustering methods such as k-mean or k-median methods are not scale-invariant. However, clustering using minimum volume ellipsoids, which use the minimum volume covering ellipsoid to cover all points in each cluster and minimizes the total volume of these covering ellipsoids, has the linear-transformation-invariance property.

*This paper contains substantial material from the full-length paper [10] "Computation of Minimum Volume Covering Ellipsoids" which has been accepted for publication in *Operations Research*. This research has been partially supported through the Singapore-MIT Alliance.

The minimum volume covering ellipsoid problem has been studied for over fifty years. As early as 1948, Fritz John [5] discussed this problem in his work on optimality conditions. Barnes [1] provides an algorithm for this problem based on matrix eigenvalue decomposition. Khachiyan and Todd [7] 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 is $O(m^{3.5} \ln\left(\frac{mR}{\epsilon}\right) \ln\left(\frac{n \ln R}{\epsilon}\right))$ 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. Nesterov and Nemirovski [9] obtain a complexity bound of $O(m^{3.5} \ln(\frac{mR}{\epsilon}))$ operations, and more recently Khachiyan [6] has reduced this to $O(m^{3.5} \ln(\frac{m}{\epsilon}))$ operations. Zhang [15] presents interior-point algorithms for MVIE, based on various equation system reduction schemes. In [16], Zhang and Gao extend their earlier results and compare different practical algorithms for the maximum volume inscribed ellipsoid problem. Boyd et. al [14] and Toh [11] both 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, herein we 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.

The paper is organized as follows. In the rest of this section, we present notation. In Section II, we review formulations of the minimum volume covering ellipsoid problem and issues in solving the problem via available interior-point software. In Section III we present our algorithm for solving the MVCE. In section IV-B, we review dual problem formulations and the conditional gradient method for solving the dual, and in Section V we develop active set strategies. Computational results are presented in Section VI and Section VII contains concluding remarks.

A. 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: $U := \mathrm{diag}(u)$ and $T := \mathrm{diag}(t)$. The Euclidean norm $\sqrt{y^T y}$ is denoted by $\|y\|$. For a given symmetric positive definite matrix M, define the M-norm by $\|v\|_M := \sqrt{v^T M v}$.

II. 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 $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$:

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

In order 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,

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

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 $A^T A$ 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

$$\mathbf{E}_{Q,c} := \{ x \in R^n | \, (x - c)^T \, Q \, (x - c) \leq 1 \} \ ;$$

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 $\frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)} \frac{1}{\sqrt{\det Q}}$, see [4] for example. Here $\Gamma(\cdot)$ is the standard gamma function of calculus.

By the change of variables: $M = Q^{\frac{1}{2}}$ and $z = Q^{\frac{1}{2}} c$, the convex optimization formulation of the minimum volume covering ellipsoid problem is:

$$\begin{array}{lll} (\mathrm{MVCE}^1) & \min_{M,z} \ \psi(M, z) & := -\ln \det M \\ & \text{s.t.} & (Ma_i - z)^T (Ma_i - z) \leq 1 \ , \ i = 1, \ldots, m \\ & & M \succ 0 \ , \end{array} \tag{1}$$

If $(\bar{M}, \bar{z})$ is a solution of MVCE^1, we recover the parameters for the optimal ellipsoid by setting $(\bar{Q}, \bar{c}) = (\bar{M}^2, \bar{M}^{-1} \bar{z})$.

A. Solution via Available Interior-Point Software

MVCE^1 can be re-written as a log-determinant maximization problem subject to linear equations and second-order cone constraints:

$$\begin{array}{lll} (\mathrm{MVCE}^2) & \min_{M,z,y,w} & -\ln \det M \\ & \text{s.t.} & Ma_i - z - y_i = 0 \ , \ i = 1, \ldots, m \\ & & w_i = 1 \ , \ i = 1, \ldots, m \\ & & (y_i, w_i) \in C_2^n \ , \ i = 1, \ldots, m \\ & & M \succ 0 \ , \end{array}$$

where C_2^n denotes the second-order cone $\{(y, w) \in \mathbb{R}^{n+1} \mid \|y\| \leq w\}$. The format of MVCE^2 is suitable for solution using a slightly modified version of the software SDPT3 (see [12], [13]), where the software is modified in order to handle the parameterized family of barrier functions:

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

for $\theta > 0$. 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 the computational burden becomes prohibitive.

One can construct the following dual problem of MVCE^1, see Section IV for details and further exposition:

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

Notice 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.

In order to solve large practical instances of the minimum volume covering ellipsoid problem ($n \geq 20, m \geq 1{,}000$, 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 Section III, 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 Section V.

III. 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.

A. Newton Step

By adding a logarithmic barrier function to the problem formulation MVCE[1], we obtain the formulation

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

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

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

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

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

$$Ut = \theta e \quad (7)$$

$$u,\ t \geq 0 \quad (8)$$

$$M \succ 0\ . \quad (9)$$

We could attempt to solve (4)-(9) for (M,z,t,u) directly 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:

$$z = \frac{MAu}{e^Tu}\ . \quad (10)$$

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

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

The following proposition demonstrates an important property of the matrix arising in (11):

Proposition 2: Under Assumption 1, if $u > 0$, then $\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right) \succ 0$. ■

The following remark presents a closed form solution for the equation system (11), see Lemma 4 of Zhang and Gao [16]:

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

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

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

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

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

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

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

and

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

$$\begin{array}{rcl} h(u) + t &=& e \\ Ut &=& \theta e \\ u,t &\geq& 0\ , \end{array} \quad (15)$$

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

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

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 $(\Delta u,\Delta t)$:

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

This system will have the unique solution:

$$\begin{array}{rcl} \Delta u &=& \left(\nabla_u h(u) - U^{-1}T\right)^{-1}\left(r_1 - U^{-1}r_2\right) \\ \Delta t &=& U^{-1}r_2 - U^{-1}T\Delta u\ , \end{array} \quad (18)$$

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

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

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

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

(ii) $\nabla_u h(u) \preceq 0$

(iii) $h(u) = \text{diag}\,(\Sigma(u))$.

■

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 $\left(\nabla_u h(u) - U^{-1}T\right) \prec 0$, and so is nonsingular. ■

Now let us put all of this together. In order 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 $\theta \geq 0$,

(i) form and factorize the matrix $M^{-2}(u) = \left[2\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)\right]$
(ii) form the matrix $\Sigma(u) = \left(A - \frac{Aue^T}{e^Tu}\right)^T M^2(u) \left(A - \frac{Aue^T}{e^Tu}\right)$
(iii) form $\nabla_u h(u) = -2\left(\frac{\Sigma(u)}{e^Tu} + \Sigma(u) \circ \Sigma(u)\right)$ and factorize $\left(\nabla_u h(u) - U^{-1}T\right)$
(iv) 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 (i) and (iii) above. The matrix $\left(AUA^T - \frac{Auu^TA^T}{e^Tu}\right)$ in step (i) is $n \times n$; it requires mn^2 operations to form and n^3 steps to factorize, while the matrix $\left(\nabla_u h(u) - U^{-1}T\right)$ in step (iv) 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$_\theta^2$. 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.

Notice that MVCE$_\theta^2$ is itself an optimization problem of a self-concordant barrier function, see [9]. 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. We are able to show that the answer to this question is negative. However, we can also 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 $c = M^{-1}z = \frac{Au}{e^Tu}$, which states that the center of the optimal ellipsoid is a convex weighting of the points $a_1, \ldots, a_m$, with the weights being the normalized dual variables $\frac{u_i}{e^Tu}, i = 1, \ldots, m$. It is also easy to see that when $\theta = 0$, the complementarity condition $u_i t_i = \theta = 0$ 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 $a_1, \ldots, a_m$ to be a random sample of m i.i.d. random variables, then with $u := \frac{e}{m}$ we have $M^{-2}(u) = \frac{2}{m}\left(A - \frac{Aee^T}{m}\right)\left(A - \frac{Aee^T}{m}\right)^T$ is proportional to the sample covariance matrix.

B. Algorithm DRN

Based on the Newton step procedure outlined in Subsection III-A, we construct the following basic interior-point algorithm for solving the MVCE$_\theta^2$ 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. OBJ$:= -\ln\det M(u)$ If $\|e - h(u) - t\| \leq \epsilon_1$ and $\frac{u^Tt}{\text{OBJ}} \leq \epsilon_2$, STOP. Return u, $Q := [M(u)]^2$, and $c := [M(u)]^{-1}z(u)$.

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

Step 3: Step-size Computation and Step. Compute $\bar{\beta} \leftarrow \max\{\beta \mid (u, t) + \beta(\Delta u, \Delta t) \geq 0\}$ 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 Subsection III-C. 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 criterion in a bit more detail below in Subsection III-D. 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.

C. 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 $u^0 = \frac{n}{2m}e$ (where the factor $\frac{n}{2m}$ 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 $h(u^0) \leq (0.95)e$, 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 $h(u^0) \not\leq (0.95)e$, observe that since $h(\alpha u) = \frac{1}{\alpha}h(u)$ from (16), we can rescale u^0 to ensure strict feasibility of the algorithm as follows: compute

$$\begin{array}{rcl} \alpha & = & \frac{\max\{h_1(u^0), \ldots, h_m(u^0)\}}{0.95} \\ u^0 & \leftarrow & \alpha u^0 \\ t^0 & \leftarrow & e - h(u^0)\ . \end{array} \tag{20}$$

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 $(u, t) = (u^0, t^0)$.

D. Stopping Criteria

The following result is the basis of the stopping criteria of Algorithm DRN:

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

Proposition 7 states that the optimality gap of a feasible solution $(M, z) := (M(u), z(u))$ of MVCE[1] is at most $u^T t$, where $t = e - h(u) \geq 0$. The stopping criteria of Algorithm DRN, specified in Step 1, is to check that primal feasibility is satisfied to a pre-specified tolerance ϵ_1, and then to check if the relative optimality gap is not greater than the pre-specified tolerance ϵ_2. 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.

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

A. Dual Formulations

Using standard Lagrangian duality constructs, one can construct the following dual problem of MVCE[1]:

$$\begin{array}{lll} \max_u & \phi(u) & := \left(\frac{n}{2}\ln 2 + \frac{n}{2}\right) + \frac{1}{2}\ln\det\left[AUA^T - \frac{Auu^TA^T}{e^Tu}\right] - e^Tu \\ \text{s.t.} & u \geq 0 \ . & \end{array} \tag{21}$$

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

Re-write u as $u = \tilde{\lambda}\tilde{u}$ where $\tilde{\lambda} = e^Tu$ and $\tilde{u} \in \mathbb{R}^m_+$. Through some arithmetic matipulations, we can re-write (21) as follows:

$$\begin{array}{lll} \max_{\tilde{\lambda},\tilde{u}} & \left(\frac{n}{2}\ln 2 + \frac{n}{2}\right) + \frac{n+1}{2}\ln\tilde{\lambda} + \frac{1}{2}\ln\det\begin{pmatrix} A\tilde{U}A^T & A\tilde{u} \\ \tilde{u}^TA^T & e^T\tilde{u}\end{pmatrix} - \frac{1}{2}\ln\tilde{\lambda} - \tilde{\lambda} \\ \text{s.t.} & e^T\tilde{u} = 1 \\ & \tilde{\lambda} \geq 0,\ \tilde{u} \geq 0 \ . \end{array} \tag{22}$$

Gathering terms in the objective function of (22) and optimizing with respect to $\tilde{\lambda}$ yields $\tilde{\lambda} = \frac{n}{2}$, which when substituted yields the refined dual problem RD of (3). We refer to RD as a refinement of (22) because (22) has been optimized with respect to the scalar variable $\tilde{\lambda}$. The $D-$optimal experimental design problem can be formulated as an instance of RD, see for example [14], and hence the computational methods developed herein are applicable to large-scale experimental design problems.

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

$$\begin{array}{lll} \text{(PL)} & \min_Y & \left(\frac{n}{2}\ln n\right) - \left(\frac{n+1}{2}\ln(n+1)\right) - \frac{1}{2}\ln\det Y \\ & \text{s.t.} & \begin{pmatrix} a_i \\ 1\end{pmatrix}^T [Y] \begin{pmatrix} a_i \\ 1\end{pmatrix} \leq 1 \quad , \ i = 1, \ldots, m \\ & & Y \in \mathbb{S}^{n+1}_{++} \ . \end{array} \tag{23}$$

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 $H := \{(x, x_{n+1}) \mid x_{n+1} = 1\}$. [7], [6], and [9] 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 $(a_i, 1)^T, i = 1, \ldots, m$.

B. 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 [2] in the early solution stages. Khachiyan [6] 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 [6] can be interpreted as a version of the conditional gradient method applied to this problem. Here we re-state this algorithm in our notation and interpretation. Let $S^{(m-1)}$ denote the standard simplex in $\mathbb{R}^m$, namely $S^{(m-1)} := \{u \in \mathbb{R}^m \mid u \geq 0, e^Tu = 1\}$, and let

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

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:

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

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 $\max_{u \in S^{(m-1)}} g(\bar{u})^Tu$, whose optimal solution is given by the j^{th} 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:

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

Khachiyan [6] 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 [6] for details). This leads to the following algorithm:

Algorithm CONDITIONAL GRADIENT

Step 0: Initialization. Choose an initial values of (u^0) satisfying $u^0 \geq 0, e^Tu^0 = 1$. Set $u \leftarrow u^0$.

Step 1: Solve subproblem. Compute $g_i(u) = \left(a_i^T \ 1\right) V(u)^{-1} \begin{pmatrix} a_i \\ 1\end{pmatrix}, i = 1, \ldots, m$. 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 $g(u)$.

V. 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 [5] states that the number of such boundary points is not too large:

Remark 8: [5] The minimum volume covering ellipsoid is determined by a subset of at most $\frac{n^2+3n}{2}$ points.

This motivates the design of active-set strategies for solving MVCE[1] 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[1] 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 $\|a_j - c_k\|_{Q_k} \leq 1 + \epsilon_1$ for $i \in \mathcal{M} \setminus \mathcal{S}_k$, 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**.

In order 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.

A. Initial Active Set

One naïve approach is to randomly choose $l \geq n+1$ 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 $\mathcal{M} \setminus \{a_i\}$. 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 which we call Sample Covariance Initialization (SCI) and Conditional Gradient Initialization (CGI), both of which we now describe.

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 data points $a_1, \ldots, a_m$. 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 $\bar{a} := \frac{1}{m}Ae$. 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. In order 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.

2) Conditional Gradient Initialization (CGI): The strategy in the conditional gradient initialization scheme is to run the conditional gradient algorithm for a small number of iterations starting at the barycenter $u = \frac{1}{m}e$ 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 Section IV-B. We accumulate these points to form the initial active set $\mathcal{S}_0$. 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 $l \geq n+1$. Because of this, we chose $l = n+1$ as the number conditional gradient steps to run.

We compare the computational performance of SCI versus CGI in Section VI.

B. Determining u_0^k

We first discuss the issue of determining u_0^0. If the initial active set $\mathcal{S}_0$ is chosen via SCI, we set $(u_0^0)_i$ proportional to the distance $d_i := \|a_i - \bar{a}\|_{Q(e)}$ for $i \in \mathcal{S}_0$, normalizing so that $e^T(u_0^0) = \frac{n}{2}$. 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 $i \in \mathcal{S}_0$, normalizing so that $e^T(u_0^0) = \frac{n}{2}$.

We now discuss how u_0^k is determined for $k \geq 1$. At the end of the previous active set step, algorithm DRN has computed $\bar{u}_k$ for the active set $\mathcal{S}_k$. If the active set has just been expanded so that $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \Delta\mathcal{S}$, 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 $\mathcal{S}_k$ and $\Delta\mathcal{S}$ 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 $e^T(u_0^{k+1}) = \frac{n}{2}$.

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

C. Updating the Active Set

1) Expanding the Active Set: Suppose that the current active set is $\mathcal{S}_k$ 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 $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \Delta\mathcal{S}$ for some set $\Delta\mathcal{S}$. When we expand the active set, we choose points $a_i \notin \mathcal{S}_k$ 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 $d_i \geq 1$ (otherwise a_i would remain inactive in the current active set), and are large. Intuitively we want the points added 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_i's 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 $\Delta\mathcal{S}$. Otherwise, the first point to be added to $\Delta\mathcal{S}$ 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 $\Delta\mathcal{S}$ if $\sum_{i\in\Delta\mathcal{S}}(a_j - c_k)^T Q_k(a_i - c_k) < 0$. In this way, the points that wind up in $\Delta\mathcal{S}$ will tend to make larger angles with other points in $\Delta\mathcal{S}$ (measured with respect to the matrix Q_k).

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, for example, using linear programming or the inscribed Löwner-John ellipsoid. However these approaches are either too expensive or not efficient enough. 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 $d_i < 0.9999$.

VI. Computational Results

In order to perform computational tests, we generated data sets of varying dimension n and number of points m. The data sets were generated using independent random multinomial Gaussian distribution or several Gaussian distributions, in order to mimic the data points 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 1.5GHz PC with 1G RAM, running LINUX.

A. Small and Medium-Size Problems

Table I shows computational results for the solution of the minimum volume covering ellipsoid problem on small- and medium-sized problems, namely $4 \leq n \leq 20$ and $20 \leq m \leq 500$). We tested three different algorithms: (i) solution via the DRN algorithm described in Section III, (ii) solution via formulation RD solved 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^2 using the same modified version of SDPT3. In Table I and elsewhere we refer to these three approaches simply as DRN, RD-SDPT3, and MVCE^2-SDPT3. All three methods were run on the full problems, i.e., without any active-set methodology. The starting point used for the DRN algorithm was as described in Subsection III-C. We tried a variety of different ways to choose starting points for algorithms RD-SDPT3 and MVCE^2-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 I show the geometric mean of the number of IPM/Newton iterations for each method over 10 problem cases while the "Solution Time" columns show the geometric means of solution times of these 10 problems.

TABLE I

Performance of algorithms DRN, RD-SDPT3, and MVCE^2-SDPT3, on small and medium-sized problem instances of the minimum volume covering ellipsoid problem. (Each number of iterations or solution time is the geometric mean of 10 problems.)

Dimensions		Algorithm					
		DRN		RD-SDPT3		MVCE^2-SDPT3	
n	m	Iterations	Solution Time (seconds)	Iterations	Solution Time (seconds)	Iterations	Solution Time (seconds)
4	20	9.980	0.0248	17.073	0.7524	12.406	1.072
4	60	11.167	0.0797	18.841	1.171	12.727	2.932
10	200	13.752	1.370	24.038	4.157	13.483	108.340
10	500	15.758	18.381	25.338	15.520	15.468	1960.957
20	500	14.378	17.065	29.114	24.396	OUT OF MEMORY	

The first observation from Table I is that MVCE^2-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^2 via SDPT3.

The second observation from Table I is that DRN needs to take roughly one half as many Newton steps as RD-SDPT3. Examining the output of SDPT3 in greater detail in order 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 of 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 $\frac{m}{n^2} << \frac{1}{2}$ we might expect DRN to dominate RD-SDPT3 due its superior choice of direction. However, when $\frac{m}{n^2} >> \frac{1}{2}$ we might expect RD-SDPT3 to dominate DRN.

B. Solving Large Problems using DRN and Active-Set Strategies

The computational results in Subsection VI-A 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 $\frac{n^2+3n}{2}$ points. Furthermore, our computational experience indicates that the number of points that determine the minimum-volume ellipsoid tends to be closer to $\frac{n^2}{4}$ in practice. This suggests that the DRN algorithm should be used to solve the active-set subproblems

TABLE II

SUMMARY PERFORMANCE OF DRN ALGORITHM WITH AN ACTIVE-SET STRATEGY USING SCI INITIALIZATION SCHEME ON LARGE PROBLEM INSTANCES OF THE MINIMUM VOLUME COVERING ELLIPSOID PROBLEM, FOR RANDOM SAMPLES OF 10 PROBLEMS.

Dimensions		SCI			
n	m	Iterations	Final Active Set	Initialization Time (seconds)	Total Solution Time (seconds)
10	10,000	9.18	42.59	0.04	0.97
20	1,000	6.51	82.76	0.01	1.28
20	10,000	10.69	117.65	0.08	3.79
20	20,000	12.44	110.67	0.15	4.53
20	30,000	12.85	121.41	0.22	5.45
30	10,000	13.65	198.53	0.13	12.55
30	20,000	14.98	204.49	0.24	16.07
30	30,000	15.98	200.93	0.36	17.64

at each major iteration, since its performance is superior to RD-SDPT3 when $\frac{m_k}{n^2} \leq \frac{1}{4} < \frac{1}{2}$, where m_k is the number of points in the active set at iteration k.

Table II summarizes the computational performance of the DRN algorithm coupled with the active-set strategy described in Section V, for dimensions n and m in the ranges $10 \leq n \leq 30$ and $1,000 \leq m \leq 30,000$, over samples of 10 randomly generated problems. The average performance measures in Table II are computed using the geometric mean of the 10 randomly generated problems. The table presents results using the initialization scheme SCI which was described in Subsections V-A.1. The "Iterations" column reports the number of outer iterations, that is, the number of different subproblems solved, and the "Final Active Set" column reports the number of points present in the last active set subproblem. (Note: 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." Clearly the final active set is a superset of the set of binding points.) The "Initialization Time" columns report the time taken by the algorithm to initialize the active set using the SCI initialization scheme. 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 $\epsilon = 10^{-7}$. Notice 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 Total Solution Times reported in Table II for the largest problems ($n = 30$, $m = 30,000$) clearly indicate that the DRN algorithm 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.

VII. 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 the DRN algorithm 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 the DRN algorithm 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 which validate the efficiency of these approaches.

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:

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

in which P is a user-specified penalizing coefficient. Formulation (MVCEP) could also be solved by a slight modification of the DRN algorithm, 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 [8]. 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 non-convex penalizing terms such as $P\sum_{i=1}^{m} \text{sign}\,(\xi_i)$.

REFERENCES

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

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

[3] C. Croux, G. Haesbroeck, and P.J. Rousseeuw. Location adjustment for the minimum volume ellipsoid estimator. *Statistics and Computing*, to appear.

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

[5] F. John. Extreme problems with inequalities as subsidiary conditions. *Studies and Essays presented to R. Courant on his 60th Birthday, January 8, 1948*, pages 187–204, 1948.

[6] L. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21(2):307–320, 1996.

[7] L. Khachiyan and M. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159, 1993.

[8] E. Knorr, R. Ng, and R. Zamar. Robust space transformations for distance-based operations. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 126–135, 2001.

[9] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1993.

[10] P. Sun and R.M. Freund. *Computation of Minimum Volume Covering Ellipsoids*. to appear in *Operations Research*.

[11] K. Toh. Primal-dual path-following algorithms for determinant maximization problems with linear matrix inequalities. *Computational Optimization and Applications*, 14:309–330, 1999.

[12] K. Toh, M. Todd, and R. Tütüncü. Sdpt3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[13] R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, to appear.

[14] L. Vandenberghe, S. Boyd, and S. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

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

[16] Y. Zhang and L. Gao. On numerical solution of the maximum volume ellipsoid problem. *SIAM Journal on Optimization*, 14(1):53–76, 2003.