Minimum volume covering ellipsoids

Elizabeth Harris[1]

(Received 30 January 2023; revised 8 January 2024)

Abstract

We present a new initialisation of an adaptive batch strategy to compute the ε-approximate minimum volume covering ellipsoid (MVCE) for a set of n points. We focus on moderately sized datasets (up to dimension $d = 100$ and $n = 1\,000\,000$. The adaptive batch strategy works in an optimisation-deletion-adaptation cycle: we solve the MVCE problem using a smaller number of points, we delete points from consideration that are guaranteed to not lie on the boundary of the MVCE, and then carefully select a new batch of points. We propose a new initialisation, which involves selecting the points corresponding to some highest leverage scores. We show using numerical examples that this new initialisation tends to improve computation time as well as reduce the total number of cycles, as compared with initialising with a random selection of points.

DOI:10.21914/anziamj.v64.17956, Published 2024-05-04, as part of the Proceedings of the 20th Biennial Computational Techniques and Applications Conference. ISSN 1445-8810. (Print two pages per sheet of paper.)

Contents

1 Introduction

The minimum volume covering ellipsoid (MVCE) problem arises in many applied and theoretical areas. Statistical applications include outlier detection [20], clustering [14], and the closely related D-optimal design problem [18]. In fact, the MVCE problem and the D-optimal design problem are dual to one another [17, 21]. Containing ellipsoids are used in parameter identification and control theory, to describe uncertainty sets for parameters and state vectors [3, 16]. Minimum volume covering ellipsoids are also used in computational geometry and computer graphics [4], in particular, for collision detection [2].

Although the MVCE problem is easy to formulate, it is not easy to solve. Consequently, many algorithms for computing the MVCEs and D-optimal designs have been studied in the literature. These include Frank–Wolfe type algorithms [5, 23, 1, 10, 12], the Dual Reduced Newton algorithm [19], and the Randomised Exchange algorithm [6]. However, these algorithms cannot deal with big high-dimensional datasets.

One approach is to combine a solution algorithm with the active set method [19, 9, 11]. Instead of running the solution algorithm on the full dataset, we apply the solution algorithm multiple times on smaller subsets of the full dataset. In each iteration, we add points to the active set, run the solution algorithm on this active set, and repeat until we find the solution to the original problem. The rationale behind this approach is that only a small number of points are required to compute the MVCE. In fact, the MVCE is determined by at most $\mathsf{d}(\mathsf{d}+3)/2$ points on its boundary [8].

The active set method has been shown to work well for moderately sized datasets [19, 9, 11], however, it may fail for larger datasets. This is because at each iteration, the active set generally increases, and hence may eventually become too large for the solution algorithm to handle. Rosa and Harman [13] solved this problem by using an adaptive batch strategy. The adaptive batch strategy works in an optimisation-deletion-adaptation cycle (or iteration). In the optimisation step, the MVCE problem is solved using a smaller number of points s. Points that are guaranteed to not lie on the boundary of the MVCE are deleted from consideration. Then in the adaptation step, a new batch of points of size s is selected, that is more likely to contain boundary points of the MVCE. Keeping the batch size constant ensures the solution algorithm can solve each subproblem at each iteration [13].

We focus on the initialisation of the adaptive batch strategy. Rosa and Harman [13] initialised by selecting a random subset of points of size s. With this initialisation, the adaptive batch strategy usually converged in three iterations. We improve this further. In Section 3.1, we present an initialisation that (a) keeps the batch size fairly small, and (b) usually converges in only one iteration. Our initialisation is based on statistical leverage scores, which have long been used in statistical regression diagnostics to identify outliers [15]. As we show in Section 3.1, points with high leverage are more likely to lie on the boundary of the MVCE. Therefore, we initialise by selecting the s points corresponding to the s highest leverage scores.

2 The minimum volume covering ellipsoid problem

Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a set of data points in $\mathbb{R}^d$. Then the minimum volume covering ellipsoid is the ellipsoid that covers $\mathcal{X}$, which attains the minimum volume of all covering ellipsoids of $\mathcal{X}$. We assume throughout that there exists a non-degenerate minimum volume covering ellipsoid.

There are many equivalent mathematical definitions of an ellipsoid. We define the ellipsoid

$$\mathcal{E}(\mathbf{Q}, \mathbf{x}_c) := \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{x} - \mathbf{x}_c)^\top \mathbf{Q} (\mathbf{x} - \mathbf{x}_c) \leqslant d\},$$

where $\mathbf{x}_c$ is the centre of the ellipsoid, and shape matrix $\mathbf{Q}$ is a d-dimensional symmetric positive definite matrix.

The volume of the ellipsoid $\mathcal{E}(\mathbf{Q}, \mathbf{x}_c)$ is

$$\operatorname{vol}(\mathcal{E}(\mathbf{Q}, \mathbf{x}_c)) = d^{d/2} \Omega_d \det(\mathbf{Q})^{-1/2},$$

where Ω_d is the volume of the unit ball in $\mathbb{R}^d$ [e.g., 22]. Minimising the ellipsoid's volume is equivalent to minimising the logarithm of its volume. After removing constant terms, we are left with minimising

$$-\log\det(\mathbf{Q}).$$

We now write the mathematical formulation of the minimum volume covering ellipsoid problem. Suppose we have a finite set of points $\mathcal{X}$. Then its minimum volume covering ellipsoid is found by solving

$$\begin{aligned} \underset{\mathbf{Q} \succ 0, \mathbf{x}_c \in \mathbb{R}^d}{\text{minimise}} \quad & -\log\det(\mathbf{Q}) \\ \text{subject to} \quad & (\mathbf{x}_i - \mathbf{x}_c)^\top \mathbf{Q} (\mathbf{x}_i - \mathbf{x}_c) \leqslant d, \quad i = 1, \ldots, n. \end{aligned} \tag{P1}$$

Here, $\mathbf{Q} \succ 0$ means that $\mathbf{Q}$ is a positive definite matrix. Although the objective function is convex, (P1) itself is not convex [22].

We reformulate (P1) so that it is convex. We first define the set

$$\hat{\mathcal{X}} := \{\hat{x}_1, \dots, \hat{x}_n\} \subseteq \mathbb{R}^{d+1},$$

where

$$\hat{x}_i := \begin{bmatrix} x_i \\ 1 \end{bmatrix},$$

for all $i = 1, \dots, n$. The points $\hat{x}_i$ are copies of the points x_i embedded in the hyperplane

$$\mathcal{H} := \left\{ \begin{bmatrix} x \\ 1 \end{bmatrix} : x \in \mathbb{R}^d \right\} \subseteq \mathbb{R}^{d+1}.$$

We then compute the MVCE of $\hat{\mathcal{X}}$, with x_c fixed at the origin. If we intersect this centred MVCE with with $\mathcal{H}$, then we obtain the MVCE of $\mathcal{X}$ [21, 10].

From now on, we only consider centred MVCEs. To find the centred MVCE of $\mathcal{X}$, we set $x_c = 0$, and solve

$$\begin{aligned} \underset{Q \succ 0}{\text{minimise}} \quad & -\log\det(Q) \\ \text{subject to} \quad & x_i^\top Q x_i \leqslant d, \quad i = 1, \dots, n. \end{aligned} \tag{P}$$

We let Q^* denote the optimal solution to (P). Then $\mathcal{E}^*(\mathcal{X}) := \mathcal{E}(Q^*, 0)$ is the minimum volume covering ellipsoid. We call (P) the MVCE problem.

The dual problem to (P) is the D-optimal design problem [17, 21]. This problem is concave, and is formulated as

$$\begin{aligned} \underset{u \in \mathbb{R}^n}{\text{maximise}} \quad & \log\det\left(\sum_{i=1}^n u_i x_i x_i^\top \right) \\ \text{subject to} \quad & \sum_{i=1}^n u_i = 1, \quad u \geqslant 0, \end{aligned} \tag{D}$$

where $\mathbf{u}$ is the design vector. A feasible solution $\mathbf{u}$ is optimal if it satisfies the optimality conditions

$$\mathbf{x}_i^\top \left(\sum_{i=1}^{n} u_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \mathbf{x}_i \leqslant d\,,$$
$$\mathbf{x}_i^\top \left(\sum_{i=1}^{n} u_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \mathbf{x}_i = d \quad \text{if } u_i > 0\,,$$

for all $i = 1, \ldots, n$. Hence for any $\mathbf{u}$ feasible for (D), the symmetric positive definite matrix

$$\mathbf{Q}(\mathbf{u}) := \left(\sum_{i=1}^{n} u_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1}$$

is feasible for (P). Further, if $\mathbf{u}^*$ is an optimal solution to (D), then the optimal solution to (P) is $\mathbf{Q}^* := \mathbf{Q}(\mathbf{u}^*)$.

We compute the minimum volume covering ellipsoid using (D), since there are many fast and reliable algorithms for calculating optimal designs for medium sized problems. Since (D) (and (P)) cannot usually be solved exactly, we focus on deriving approximate solutions. To ensure the chosen algorithm terminates with a guaranteed quality of solution, we define some approximate optimality conditions. A feasible $\mathbf{u}$ for (D) is called ε-primal feasible if $\mathbf{Q}(\mathbf{u})$ satisfies

$$\mathbf{x}_i^\top \mathbf{Q}(\mathbf{u}) \mathbf{x}_i \leqslant (1+\varepsilon)\, d\,,$$

for all $i = 1, \ldots, n$. If $\mathbf{Q}(\mathbf{u})$ additionally satisfies

$$\mathbf{x}_i^\top \mathbf{Q}(\mathbf{u}) \mathbf{x}_i \geqslant (1-\varepsilon)\, d \quad \text{if } u_i > 0\,,$$

for all $i = 1, \ldots, n$, then we say that $\mathbf{u}$ is ε-approximately optimal. If we have an ε-primal feasible (or ε-approximately optimal) solution $\mathbf{u}$, then $\mathbf{u}$ and $(1+\varepsilon)^{-1}\mathbf{Q}(\mathbf{u})$ are both within $d \log(1+\varepsilon)$ of being optimal in (D) and (P), respectively [22, Proposition 2.9].

Algorithm 1 Adaptive Batch Strategy

Require: $\mathcal{X} = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, $s < n$, $\varepsilon \in (0, 1)$

1: Initialisation. Determine an initial subset of points $\bar{\mathcal{X}}$ of size s.
2: Optimisation. Solve (D) for the set of points $\bar{\mathcal{X}}$. Output design $\bar{u}$.
3: Check optimality conditions. If $x_i^\top Q(\bar{u}) x_i \leqslant d(1+\varepsilon)$ for all i, stop. Return $Q^* = Q(\bar{u})$. Otherwise go to Step 4.
4: Deletion. Delete inessential points from $\mathcal{X}$.
5: Adaptation. Select a new subset $\bar{\mathcal{X}}$ of size s that is more likely to contain boundary points of $\mathcal{E}^*(\mathcal{X})$. Go to Step 2.

Ensure: Q^*

3 Adaptive batch strategy

The minimum volume covering ellipsoid is only determined by points on its boundary. For centrally symmetric sets, the minimum volume covering ellipsoid is determined by at most $(d^2 + d)/2$ boundary points [8]. These points are called the core set.

This suggests that to solve the MVCE problem, we do not need to consider all the points in $\mathcal{X}$. In the Adaptive Batch Strategy [13], we only consider s points from $\mathcal{X}$ at a time, where s is chosen to be much smaller than the total number of points n. In Algorithm 1, we outline the Adaptive Batch Strategy. We now explain each step in detail.

3.1 Initialisation

Rosa and Harman [13] initialised by uniformly sampling s points from n. This initialisation is computationally inexpensive, and simple to implement. Despite these advantages, its main disadvantage is that it does not focus on capturing points from the core set.

In our new initialisation, we aim to select as many points from the core set as possible, while keeping s small. To this end, we briefly introduce the concept

of statistical leverage. The leverage score has long been used in statistical regression diagnostics to identify outliers [15]. Let the points in the dataset $\mathcal{X}$ be contained in the rows of $\mathbf{X} \in \mathbb{R}^{n \times d}$. Then the ith leverage score is precisely the ith diagonal entry of the hat matrix $\mathbf{H}$, that is,

$$\ell_i := \boldsymbol{e}_i^\top \mathbf{H} \boldsymbol{e}_i \,,$$

for all $i = 1, \ldots, n$, where

$$\mathbf{H} := \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \,.$$

Points of equal leverage are points lying on one ellipsoid. To see this, observe that for $i = 1, \ldots, n$, we have

$$\ell_i = \boldsymbol{e}_i^\top \mathbf{H} \boldsymbol{e}_i = \boldsymbol{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \boldsymbol{x}_i \,.$$

Here, $(\mathbf{X}^\top \mathbf{X})^{-1}$ is a symmetric positive definite matrix, and hence is a shape matrix. So each $\boldsymbol{x}_i$ is a point lying on the ellipsoid,

$$\mathcal{E}\left(\frac{d}{\ell_i}(\mathbf{X}^\top \mathbf{X})^{-1}, 0\right),$$

as long as $\boldsymbol{x}_i$ is non-zero. Otherwise, the ellipsoid is just the origin.

We therefore think of leverage scores as distances from the origin. We also conclude that points with high leverage are more likely to be in the core set of $\mathcal{X}$. We therefore initialise as follows. Given a dataset $\mathcal{X}$, we calculate the leverage score of each data point. We then order the data points by decreasing leverage score, and select the first s points. These points form another set $\bar{\mathcal{X}}$.

3.2 Optimisation

In our implementation of the optimisation step, we compute the solution to the D-optimal design problem to find the MVCE of each batch $\bar{\mathcal{X}}$. Specifically, we use the Wolfe–Atwood (WA) algorithm [1, 23]. We stop the algorithm once

we achieve a design $\bar{\mathbf{u}}$ that is ε-approximately optimal, where $\varepsilon = 10^{-9}$. The algorithm outputs $\bar{\mathbf{u}}$, as well as the scaled Cholesky factorisation

$$\sum_{i=1}^{s} \bar{u}_i \mathbf{x}_i \mathbf{x}_i^\top = \bar{c} \bar{\mathbf{R}}^\top \bar{\mathbf{R}},$$

where $\bar{c} \in \mathbb{R}$ and $\bar{\mathbf{R}} \in \mathbb{R}^{d \times d}$ is a non-singular upper triangular matrix [22]. Hence

$$\bar{\mathbf{Q}} = \mathbf{Q}(\bar{\mathbf{u}}) = \left(\sum_{i=1}^{s} \bar{u}_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} = \left(\bar{c} \bar{\mathbf{R}}^\top \bar{\mathbf{R}} \right)^{-1} = \frac{1}{\bar{c}} \bar{\mathbf{R}}^{-1} \bar{\mathbf{R}}^{-\top}.$$

3.3 Optimality conditions

We check if the current solution is ε-approximately optimal for the entire dataset $\mathcal{X}$. To do this, we only need to check for ε-primal feasibility. Consider the design vector $\mathbf{u} \in \mathbb{R}^n$, where

$$u_i = \begin{cases} \bar{u}_i, & \text{if } \mathbf{x}_i \in \bar{\mathcal{X}}, \\ 0, & \text{otherwise.} \end{cases}$$

Then $\mathbf{u}$ is an ε-approximately optimal solution to the full problem if

$$\mathbf{x}_i^\top \mathbf{Q}(\mathbf{u}) \mathbf{x}_i = \mathbf{x}_i^\top \mathbf{Q}(\bar{\mathbf{u}}) \mathbf{x}_i \leqslant (1 + \varepsilon) d, \tag{1}$$

for all $i = 1, \ldots, n$, and

$$\mathbf{x}_i^\top \mathbf{Q}(\mathbf{u}) \mathbf{x}_i = \mathbf{x}_i^\top \mathbf{Q}(\bar{\mathbf{u}}) \mathbf{x}_i \geqslant (1 - \varepsilon) d \quad \text{if } u_i > 0, \tag{2}$$

for all $i = 1, \ldots, n$. Conditions (1) and (2) are satisfied for the points in $\bar{\mathcal{X}}$, so we only need to consider the points not in $\bar{\mathcal{X}}$. Since $u_i = 0$ when $\mathbf{x}_i \notin \bar{\mathcal{X}}$, only condition (1) needs to be checked.

Now, for each point $\mathbf{x}_i \in \mathcal{X}$, its current ellipsoidal distance is

$$d_{\bar{u}}(\mathbf{x}_i) := \mathbf{x}_i^\top \bar{\mathbf{Q}} \mathbf{x}_i = \frac{1}{\bar{c}} \left\| \bar{\mathbf{R}}^{-1} \mathbf{x}_i \right\|_2^2.$$

Therefore, if for all $\mathbf{x}_i \in \mathcal{X}$, we have

$$d_{\bar{\mathbf{u}}}(\mathbf{x}_i) \leqslant d(1+\varepsilon),$$

then we stop, and output the shape matrix

$$\mathbf{Q}^* = \bar{\mathbf{Q}} = \frac{1}{\bar{c}} \bar{\mathbf{R}}^{-1} \bar{\mathbf{R}}^{-\top}.$$

3.4 Deletion

Harman and Pronzato [7] show that points with current ellipsoidal distance

$$d_{\bar{\mathbf{u}}}(\mathbf{x}_i) < d\left(1 + \frac{\delta - \sqrt{\delta(4+\delta-4/d)}}{2}\right),$$

where

$$\delta = \max_{\mathbf{x}_i \in \mathcal{X}} \{d_{\bar{\mathbf{u}}}(\mathbf{x}_i) - d\},$$

can never lie on the boundary of $\mathcal{E}^*(\mathcal{X})$. Therefore we can safely eliminate these points from $\mathcal{X}$.

3.5 Adaptation

In Section 3.1, we discuss that points with higher leverage are more likely to be in the core set of $\mathcal{X}$. Similarly, points with higher current ellipsoidal distance $d_{\bar{\mathbf{u}}}(\mathbf{x}_i)$ are more likely to be in the core set of $\mathcal{X}$. This suggests that we should select the s points with largest $d_{\bar{\mathbf{u}}}(\mathbf{x}_i)$.

However, this choice of batch does not guarantee convergence of the Adaptive Batch Strategy. Instead, Rosa and Harman [13] select the s_{supp} points that support the current design $\bar{\mathbf{u}}$ (i.e., points with $\bar{u}_i > 0$). These points are a core set for the current ellipsoid, so in a sense, define the current ellipsoid. Then to these points, they add the $s - s_{\text{supp}}$ points with highest current ellipsoidal distance $d_{\bar{\mathbf{u}}}(\mathbf{x}_i)$. As long as $s_{\text{supp}} < s$, the Adaptive Batch Strategy converges [13, Theorem 1].

Table 1: Iteration summary results of the computational experiments. The values are obtained from the geometric mean of 100 independent runs. Entries with '—' indicate that the algorithm did not converge for this choice of s.

		1%		2%		5%		10%	
d	n	lev	unif	lev	unif	lev	unif	lev	unif
10	100 000	1.16	2.41	1.04	2.01	1.00	2.00	1.00	2.00
	1 000 000	1.02	2.01	1.00	2.00	1.00	2.00	1.00	2.00
30	100 000	2.00	3.00	1.24	3.00	1.00	2.18	1.00	2.00
	1 000 000	1.00	2.28	1.00	2.02	1.00	2.00	1.00	2.00
50	100 000	2.19	4.00	2.00	3.00	1.06	2.92	1.01	2.03
	1 000 000	1.05	2.99	1.00	2.18	1.00	2.00	1.00	2.00
80	100 000	4.10	6.03	2.00	4.00	1.52	3.00	1.00	2.33
	1 000 000	1.39	3.00	1.01	2.82	1.00	2.00	1.00	2.00
100	100 000	—	—	3.66	5.12	1.96	3.01	1.01	2.76
	1 000 000	1.80	3.01	1.01	3.00	1.00	2.01	1.00	2.00

4 Results

We compare the Adaptive Batch Strategy using our leverage score based initialisation with the uniform sampling initialisation used by Rosa and Harman [13]. We then compare both Adaptive Batch Strategies with directly applying the WA algorithm to the entire dataset. All computations are performed on a personal laptop with a 64 bit MacOS 12 operating system, and a 2.4 GHz Quad-Core Intel Core i5 processor with 8 GB of RAM. The algorithms are run using MATLAB (R2021a).

We generate each dataset $\mathcal{X}$ from a mixture of independent multivariate Gaussian distributions [13]. We choose values of $n = 100\,000$ and $1\,000\,000$, and vary d between 10 and 100. Then for each Adaptive Batch Strategy, we vary the batch size s between 1% and 10% of n.

In Table 1, we summarise the iteration complexity of the Adaptive Batch Strategies using the two different initialisations. In the tables, we refer to these

Table 2: Time summary results of the computational experiments. The values are obtained from the geometric mean of 100 independent runs. All times shown are in seconds. Entries with '—' indicate that the algorithm did not converge for this choice of s.

		1%		2%		5%		10%		
d	n	lev	unif	lev	unif	lev	unif	lev	unif	full
10	100 000	0.2	0.2	0.2	0.3	0.3	0.5	0.3	0.6	0.8
	1 000 000	0.6	0.9	0.7	1.1	0.9	1.5	1.1	1.8	2.9
30	100 000	1.9	1.9	1.1	2.2	1.9	3.4	2.2	3.8	4.4
	1 000 000	3.5	6.1	4.7	6.8	6.1	9.5	7.9	12.8	28.9
50	100 000	10.4	13.4	11.6	12.5	8.4	19.1	9.0	14.4	23.9
	1 000 000	13.5	30.8	15.2	25.1	22.1	32.0	27.2	43.2	186.3
80	100 000	42.4	46.8	23.6	35.3	28.6	35.0	21.3	37.2	46.4
	1 000 000	47.8	78.8	42.1	94.9	55.8	90.5	76.6	124.2	456.1
100	100 000	—	—	56.7	72.3	53.1	67.0	34.3	78.0	109.3
	1 000 000	87.3	118.0	67.0	159.1	90.7	140.1	120.7	195.4	1207.0

strategies as lev and unif. In text, we refer to these strategies as uniform ABS and leverage ABS. Notice that regardless of initialisation strategy, as batch size increases, the number of iterations decreases or stays the same. Also, the leverage ABS converges in fewer iterations compared to the uniform ABS. In fact, our leverage ABS often converges in only one iteration, further confirming that points with high leverage often correspond to points in the core set of the MVCE.

In Table 2, we summarise the total computation time of both Adaptive Batch Strategies and the full algorithm (i.e. directly applying the WA algorithm to the full dataset). We first note that both Adaptive Batch Strategies performed better than directly applying the WA algorithm (provided that the Adaptive Batch Strategy converged). We finally note that although our leverage initialisation is more computationally expensive, Table 2 shows that it is more efficient.

References

[1] C. L. Atwood. "Sequences converging to D-optimal designs of experiments". In: *Ann. Statist.* 1.2 (1973), pp. 342–352. DOI: 10.1214/aos/1176342371 (cit. on pp. C198, C204).

[2] Y.-J. Chen, M.-Y. Ju, and K.-S. Hwang. "A virtual torque-based approach to kinematic control of redundant manipulators". In: *IEEE Trans. Indust. Elec.* 64.2 (2017), pp. 1728–1736. DOI: 10.1109/TIE.2016.2548439 (cit. on p. C198).

[3] F. L. Chernousko. "Ellipsoidal state estimation for dynamical systems". In: *Nonlin. Anal.* 63.5-7 (2005), pp. 872–879. DOI: 10.1016/j.na.2005.01.009 (cit. on p. C198).

[4] D. Eberly. *3D game engine design: A practical approach to real-time computer graphics*. CRC Press, 2007. DOI: 10.1201/b18212 (cit. on p. C198).

[5] M. Frank and P. Wolfe. "An algorithm for quadratic programming". In: *Naval Res. Log. Quart.* 3.1-2 (1956), pp. 95–110. DOI: 10.1002/nav.3800030109 (cit. on p. C198).

[6] R. Harman, L. Filová, and P. Richtárik. "A randomized exchange algorithm for computing optimal approximate designs of experiments". In: *J. Am. Stat. Ass.* 115.529 (2020), pp. 348–361. DOI: 10.1080/01621459.2018.1546588 (cit. on p. C198).

[7] R. Harman and L. Pronzato. "Improvements on removing nonoptimal support points in D-optimum design algorithms". In: *Stat. Prob. Lett.* 77.1 (2007), pp. 90–94. DOI: 10.1016/j.spl.2006.05.014 (cit. on p. C206).

[8] F. John. "Extremum problems with inequalities as subsidiary conditions". In: *Traces and emergence of nonlinear programming*. Springer Basel, 2014, pp. 197–215. DOI: 10.1007/978-3-0348-0439-4_9 (cit. on pp. C199, C203).

[9] L. Källberg and D. Andrén. *Active set strategies for the computation of minimum-volume enclosing ellipsoids*. Tech. rep. Mälardalen University, 2019. URL: http://www.es.mdu.se/publications/5680- (cit. on p. C199).

[10] L. G. Khachiyan. "Rounding of polytopes in the real number model of computation". In: *Math. Op. Res.* 21.2 (1996), pp. 307–320. DOI: 10.1287/moor.21.2.307 (cit. on pp. C198, C201).

[11] J. Kudela. "Minimum-volume covering ellipsoids: Improving the efficiency of the Wolfe–Atwood algorithm for large-scale instances by pooling and batching". In: *MENDEL* 25.2 (2019), pp. 19–26. DOI: 10.13164/mendel.2019.2.019 (cit. on p. C199).

[12] P. Kumar and E. A. Yildirim. "Minimum-volume enclosing ellipsoids and core sets". In: *J. Op. Theor. Appl.* 126.1 (2005), pp. 1–21. DOI: 10.1007/s10957-005-2653-6 (cit. on p. C198).

[13] S. Rosa and R. Harman. "Computing minimum-volume enclosing ellipsoids for large datasets". In: *Comput. Stat. Data Anal.* 171, 107452 (2022). DOI: 10.1016/j.csda.2022.107452 (cit. on pp. C199, C203, C206, C207).

[14] J. B. Rosen. "Pattern separation by convex programming". In: *J. Math. Anal. Appl.* 10.1 (1965), pp. 123–134. DOI: 10.1016/0022-247X(65)90150-2 (cit. on p. C198).

[15] P. J. Rousseeuw and M. Hubert. "Robust statistics for outlier detection". In: *WIREs: Data mining and knowledge discovery* 1.1 (2011), pp. 73–79. DOI: 10.1002/widm.2 (cit. on pp. C199, C204).

[16] F. Schweppe. "Recursive state estimation: Unknown but bounded errors and system inputs". In: *IEEE Trans. Auto. Control* 13.1 (1968), pp. 22–28. DOI: 10.1109/TAC.1968.1098790 (cit. on p. C198).

[17] R. Sibson. "Discussion of Dr Wynn's and of Dr Laycock's papers". In: *J. Roy. Stat. Soc. B* 34.2 (1972), pp. 181–183. DOI: 10.1111/j.2517-6161.1972.tb00898.x (cit. on pp. C198, C201).

[18] S. D. Silvey. *Optimal design: An introduction to the theory for parameter estimation*. Ettore Majorana International Science Series. London, Chapman and Hall, 1980. DOI: 10.1007/978-94-009-5912-5 (cit. on p. C198).

[19] P. Sun and R. M. Freund. "Computation of minimum-volume covering ellipsoids". In: *Op. Res.* 52.5 (2004), pp. 690–706. DOI: 10.1287/opre.1040.0115 (cit. on pp. C198, C199).

[20] D. M. Titterington. "Estimation of correlation coefficients by ellipsoidal trimming". In: *J. Roy. Stat. Soc.: C* 27.3 (1978), pp. 227–234. DOI: 10.2307/2347157 (cit. on p. C198).

[21] D. M. Titterington. "Optimal design: Some geometrical aspects of D-optimality". In: *Biometrika* 62.2 (1975), pp. 313–320. DOI: 10.2307/2335366 (cit. on pp. C198, C201).

[22] M. J. Todd. *Minimum-volume ellipsoids: Theory and algorithms*. MOS-SIAM Series on Optimization. SIAM, 2016. DOI: 10.1137/1.9781611974386 (cit. on pp. C200, C202, C205).

[23] P. Wolfe. "Convergence theory in nonlinear programming". In: ed. by J. Abadie. Integer and Nonlinear Programming. North Holland, Amsterdam, 1970, pp. 1–36. DOI: 10.1007/BF00932858 (cit. on pp. C198, C204).

Author address

1. **Elizabeth Harris**, School of Mathematical and Physical Sciences, University of Newcastle, Callaghan, NSW 2308, Australia
mailto:e.harris@uon.edu.au
orcid:0000-0002-8799-345X