Toeplitz covariance estimation with applications to MUSIC

Daniel Cederberg, *Graduate Student Member, IEEE*

***Abstract*—We consider the maximum likelihood estimation of a covariance matrix that is known to be Toeplitz. The corresponding optimization problem is nonconvex, and to compute a local minimizer we propose an efficient implementation of Newton's method. The cost per iteration of a straightforward implementation is of order $\mathcal{O}(n^4)$, where $n+1$ is the dimension of the covariance matrix. By exploiting fast Fourier transforms to assemble the gradient and Hessian we reduce the iteration cost to order $\mathcal{O}(n^3)$. Through extensive simulations we demonstrate that for direction-of-arrival estimation, the performance of MUSIC is significantly enhanced by using it together with the maximum likelihood Toeplitz covariance estimate. The simulations also demonstrate that our open-source implementation is orders of magnitude faster than two existing Toeplitz covariance estimators.**

***Index Terms*—Toeplitz covariance matrix, maximum-likelihood estimation, array processing, fast Fourier transform.**

I. Introduction

The accuracy of many algorithms in signal processing strongly depends on the quality of an estimate of the signal's covariance matrix. Obtaining an accurate estimate of the covariance matrix can be difficult when faced with few measurements of the signal or low signal-to-noise ratio. Sometimes a priori information on the structure of the covariance matrix can be taken into account to compute a more accurate esimate, possibly boosting the performance of algorithms relying on the estimate.

A. Toeplitz covariance matrices

In signal processing it is often assumed that the signal is stationary which implies that the covariance matrix is Toeplitz. The list of applications of Toeplitz covariance matrices is extensive and includes:

1) Enhancement of the performance of direction-of-arrival estimation algorithms such as beamforming [1], [2], [3] or the multiple signal classification algorithm (MUSIC) [4], [5], [6], [7].
2) Recovery of second-order statistics through compressive covariance sensing [8], [9], [10].
3) Target detection in radar systems [11], [12], [13], [14], [15].

Given the diverse range of applications in which Toeplitz covariance matrices arise, it is important to be able to estimate them accurately.

B. Toeplitz maximum likelihood estimation

In this paper we consider the maximum likelihood (ML) estimation of a covariance matrix that is known to be Toeplitz. Let $z(1), z(2), \ldots, z(K) \in \mathbf{C}^{n+1}$ be K independent observations of a complex circular zero-mean Gaussian random vector z with covariance matrix $T \in \mathbf{H}^{n+1}_{++}$, where T is assumed to be a Toeplitz matrix and $\mathbf{H}^{n+1}_{++}$ is the space of positive definite Hermitian matrices of dimension $n+1$. The probability density function of $z \sim \mathcal{CN}(0, T)$ is [16, Chapter 15]

$$p(z; T) = \frac{1}{\pi^K \det T} \exp(-z^H T^{-1} z).$$

The log-likelihood value of the observations (up to additive and multiplicative constants) is

$$-\mathbf{log\ det}\, T - \mathbf{Tr}(T^{-1} S),$$

where $S = \frac{1}{K} \sum_{k=1}^{K} z(k) z(k)^H$ is the sample covariance matrix. The Toeplitz matrix that maximizes the likelihood of the observations is found by solving the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{log\ det}\, T + \mathbf{Tr}(T^{-1} S) \\ \text{subject to} & T \text{ being Toeplitz,} \end{array} \tag{1}$$

with variable $T \in \mathbf{H}^{n+1}$. The objective function of (1) is nonconvex so in this paper the word *solution* will be used in a rather informal way. By a solution of (1) we simply mean a point at which the gradient of the objective function vanishes.

C. Existing algorithms

Maximum likelihood estimation of Toeplitz covariance matrices can be traced back to the seminal paper of Burg et al. [17], who used adhoc-principles to derive an iterative method for finding a point satisfying the optimality conditions of problem (1). Since then several works on ML Toeplitz covariance estimation, including [18], [19], [20], [21], [22], [23], have proposed iterative algorithms based on the majorization-minimization framework [24]. Of these references the first four focus exclusively on Toeplitz matrices with a positive semidefinite circular extension (consequences of this restriction are discussed in [25], [22]). In [26] an algorithm based on outer approximation was proposed for solving (1).

Another line of research is based on the presumption that attempting to solve (1) is prohibitively expensive for practical applications and proposes closed-form Toeplitz covariance estimators [5], [11].

While the above references justify their algorithms from the ML principle, Toeplitz covariance estimators derived from other principles have also been proposed (see, for example, [14], [27]).

The author is with the department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: dance858@stanford.edu).

D. This paper

We propose to solve problem (1) using Newton's method. In every iteration of Newton's method the gradient and Hessian of the objective function must be evaluated. The cost of a straightforward evaluation of the gradient and Hessian turns out to be of order $\mathcal{O}(n^3)$ and $\mathcal{O}(n^4)$, respectively, where $(n+1)$ is the dimension of the covariance matrix. We show how fast Fourier transforms allow us to reduce the complexities to $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$, respectively. Our implementation is available online and is orders of magnitude faster than several other algorithms for Toeplitz covariance estimation.

To benchmark our method against existing techniques for Toeplitz covariance estimation, we first assess the methods in terms of the Euclidean distance to the true covariance matrix. However, we firmly believe that the efficacy of a method for covariance matrix estimation should be evaluated by studying its impact on specific applications. In light of this, we consider the problem of estimating the arrival angles of several sources emitting signals that impinge on a uniform linear array of sensors. Through extensive simulations we demonstrate that the performance of one of the most popular direction-of-arrival estimation algorithms, MUSIC, is significantly enhanced by using it together with the covariance matrix estimate obtained from our method. MUSIC together with the sample covariance matrix is known to perform well asymptotically, but its poor performance in a regime with low signal-to-noise ratio, few samples, or a small signal spatial separation has several times been documented [28], [29], [4], [7]. In contrast, MUSIC with the covariance matrix estimate obtained from our method often performs relatively well in this more difficult regime, and in many cases it even attains the Cramér–Rao bound for uncorrelated signals.

Outline of paper: The remaining part of the paper is structured as follows. In §II we summarize Newton's method and present expressions for the gradient and Hessian of the objective function. In §III we outline how structure can be exploited in the implementation. In particular, we show how to efficiently evaluate the gradient and Hessian. Numerical experiments are presented in §IV and the paper is concluded in §V.

Notation: The space of Hermitian matrices of dimension $(n+1)$ is denoted by $\mathbf{H}^{n+1}$, and the Hermitian conjugate of a matrix A is denoted by A^H. We let $i = \sqrt{-1}$. The real and imaginary parts of $z \in \mathbf{C}$ are $\Re z$ and $\Im z$, respectively, and $\overline{z}$ denotes the complex conjugate. If A is a matrix or a vector, $\Re A$, $\Im A$ and $\overline{A}$ refers to elementwise operations. For $a = (a_0, a_1, \dots, a_n)^T \in \mathbf{C}^{n+1}$ we let $[a]_{1:n} = (a_1, \dots, a_n)^T \in \mathbf{C}^n$. Similar notation is used for matrices: if $A \in \mathbf{C}^{(n+1)\times(n+1)}$, then $[A]_{1:n,1:n} \in \mathbf{C}^{n\times n}$ is the matrix obtained by excluding the first row and column of A. Similarily, $[A]_{1:n,0:n} \in \mathbf{C}^{n\times(n+1)}$ is the matrix obtained by excluding the first row of A.

II. Algorithm

In this section we describe Newton's method applied to a reformulation of problem (1).

A Hermitian Toeplitz matrix of dimension $n+1$ is determined by its first row, which is specified by $2n+1$ real parameters. For $x = (x_0, x_1, \dots, x_n)^T \in \mathbf{R}^{n+1}$ and $y = (y_1, y_2, \dots, y_n)^T \in \mathbf{R}^n$ define $\mathcal{T} : \mathbf{R}^{n+1} \times \mathbf{R}^n \mapsto \mathbf{H}^{n+1}$ by

$$\mathcal{T}(x,y) = \begin{bmatrix} 2x_0 & x_1 + iy_1 & \dots & x_n + iy_n \\ x_1 - iy_1 & 2x_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & x_1 + iy_1 \\ x_n - iy_n & \dots & x_1 - iy_1 & 2x_0 \end{bmatrix}. \tag{2}$$

In words, $\mathcal{T}(x,y)$ is the Hermitian Toeplitz matrix whose first row is specified by (x,y). Problem (1) is equivalent to

$$\text{minimize } f(x,y) = \mathbf{log\ det}\ \mathcal{T}(x,y) + \mathbf{Tr}(\mathcal{T}(x,y)^{-1}S), \tag{3}$$

with variable $(x,y) \in \mathbf{R}^{n+1} \times \mathbf{R}^n$. The domain of the objective function is

$$\mathbf{dom}\, f = \{(x,y) \in \mathbf{R}^{n+1} \times \mathbf{R}^n \mid \mathcal{T}(x,y) \succ 0\}.$$

A. Newton's method

To solve (3) we propose to use Newton's method. Given an initial point $(x_0, y_0) \in \mathbf{dom}\, f$, the iterates are computed according to

$$\begin{aligned} (\Delta x_k, \Delta y_k) &= -B_k^{-1} \nabla f(x_k, y_k) \\ (x_{k+1}, y_{k+1}) &= (x_k, y_k) + t_k(\Delta x_k, \Delta y_k), \qquad k \geq 0, \end{aligned}$$

where B_k is equal to the true Hessian $\nabla^2 f(x_k, y_k)$ if $\nabla^2 f(x_k, y_k) \succ 0$. If $\nabla^2 f(x_k, y_k)$ is not positive definite, then we let $B_k = \nabla^2 f(x_k, y_k) + \tau_k I$ for succesively larger values of $\tau_k > 0$ until $B_k \succ 0$ [30, page 51]. The step size t_k is chosen through a backtracking line search [30, page 37]. It depends on two constants α, β with $0 < \alpha < 0.5$, $0 < \beta < 1$. Starting with $t_k = 1$, the step size is multiplied by β until the descent condition

$$\begin{aligned} f(x_k + t_k \Delta x_k, y_k + t_k \Delta y_k) &\leq f(x_k, y_k) \\ &+ \alpha t_k \nabla f(x_k, y_k)^T (\Delta x_k, \Delta y_k) \end{aligned} \tag{4}$$

is satisfied. Typical values on the line search parameters are $\alpha = 0.05$ and $\beta = 0.8$.

The main cost of every iteration is to compute the gradient and Hessian of f, and a Cholesky factorization of B_k. We will see that the bottleneck is to evaluate the gradient and Hessian.

B. Derivatives

Now we present expressions for $\nabla f(x,y)$ and $\nabla^2 f(x,y)$ at a point $(x,y) \in \mathbf{dom}\, f$. In particular, we assume that $\mathcal{T}(x,y) \succ 0$.

Let $E_k \in \mathbf{R}^{(n+1)\times(n+1)}$ be the matrix with ones only on subdiagonal k, and zeroes elsewhere. The matrix E_k is known as a *shift matrix*, since when applied to a vector $a = (a_0, a_1, \dots, a_n)^T \in \mathbf{C}^{n+1}$ it shifts the components of a over k positions:

$$E_k a = (\underbrace{0, \dots, 0}_{k \text{ zeros}}, a_0, a_1, \dots, a_{n-k})^T.$$

The operator $\mathcal{T}$ defined in (2) can be expressed as

$$\mathcal{T}(x,y) = \sum_{k=0}^{n} x_k(E_k^T + E_k) + i \sum_{k=1}^{n} y_k(E_k^T - E_k).$$

Partition the objective function as $f(x,y) = f_1(x,y) + f_2(x,y)$ where $f_1(x,y) = \mathbf{log\,det}\,\mathcal{T}(x,y)$ and $f_2(x,y) = \mathbf{Tr}(\mathcal{T}(x,y)^{-1}S)$. Since $\mathcal{T}(x,y)^{-1} \succ 0$, a Cholesky factorization $\mathcal{T}(x,y)^{-1} = RR^H$ with $R \in \mathbf{C}^{(n+1)\times(n+1)}$ lower triangular exists. While the matrix R depends on x we will simply denote it by R instead of $R(x)$. Also let $L \in \mathbf{C}^{(n+1)\times(n+1)}$ be a Cholesky factor of S (*i.e.* $S = LL^H$), and let $A = RR^H L \in \mathbf{C}^{(n+1)\times(n+1)}$. Denote the columns of R and A by r_k and a_k, $0 \le k \le n$, respectively. Deriving the partial derivatives of f is a tedious but straightforward task (see the appendix). For example, one can show that for $0 \le j \le n$:

$$\frac{\partial f_1(x,y)}{\partial x_j} = 2\Re\,\mathbf{Tr}(R^H E_j R)$$
$$\frac{\partial f_2(x,y)}{\partial x_j} = -2\Re\,\mathbf{Tr}(A^H E_j A),$$

and for $0 \le j \le n$, $0 \le t \le n$:

$$\frac{\partial^2 f_1(x,y)}{\partial x_t \partial x_j} = -\,2\Re\,\mathbf{Tr}(R^H E_t^T R R^H E_j R + R^H E_t R R^H E_j R)$$
$$\frac{\partial^2 f_2(x,y)}{\partial x_t \partial x_j} = 2\Re\,\mathbf{Tr}(A^H E_t^T R R^H E_j^T A + A^H E_t^T R R^H E_j A) + 2\Re\,\mathbf{Tr}(A^H E_t R R^H E_j^T A + A^H E_t R R^H E_j A).$$

The remaining partial derivatives have similar expressions.

The complexity of evaluating the gradient and Hessian can be summarized as follows. The cost of computing the Cholesky factorization of $\mathcal{T}(x,y)^{-1}$ is of order $\mathcal{O}(n^2)$ using the Levinson–Durbin algorithm (see §III-A). The cost of computing A is of order $\mathcal{O}(n^3)$. To evaluate the gradient and Hessians one must *e.g.* compute the quantities

$$\mathbf{Tr}(R^H E_j R) = \sum_{k=0}^{n} r_k^H E_j r_k$$
$$\mathbf{Tr}(A^H E_t R R^H E_j A) = \sum_{k=0}^{n}\sum_{\ell=0}^{n} (a_\ell^H E_t r_k)(r_k^H E_j a_\ell)$$

for $0 \le k \le n$, $0 \le j \le n$. The complexity of computing $r_k^H E_j r_k$ for $0 \le k \le n$, $0 \le j \le n$ is of order $\mathcal{O}(n^3)$. The complexity of computing $a_\ell^H E_t r_k$ for $0 \le \ell \le n$, $0 \le k \le n$, $0 \le t \le n$ is of order $\mathcal{O}(n^4)$.

To summarize, the complexity of evaluating the gradient and Hessian is of order $\mathcal{O}(n^3)$ and $\mathcal{O}(n^4)$, respectively.

C. Derivatives expressed using crosscorrelation

To evaluate the gradient and Hessian more efficiently, it is convenient to recognize that the partial derivatives can be expressed using the notion of *crosscorrelation*. For $0 \le k \le n$, $0 \le \ell \le n$, $0 \le j \le n$, let

$$c_{r,r}^{(j)}(k,k) = \overline{r_k^H E_j r_k}, \qquad c_{a,a}^{(j)}(k,k) = \overline{a_k^H E_j a_k}$$
$$c_{r,a}^{(j)}(k,\ell) = \overline{r_k^H E_j a_\ell}, \qquad c_{a,r}^{(j)}(\ell,k) = \overline{a_\ell^H E_j r_k}.$$

Define crosscorrelation vectors

$$c_{r,r}(k,k) = (c_{r,r}^{(0)}(k,k),\ c_{r,r}^{(1)}(k,k), \ldots, c_{r,r}^{(n)}(k,k))^T \in \mathbf{C}^{n+1}$$
$$c_{a,a}(k,k) = (c_{a,a}^{(0)}(k,k),\ c_{a,a}^{(1)}(k,k), \ldots, c_{a,a}^{(n)}(k,k))^T \in \mathbf{C}^{n+1}$$
$$c_{r,a}(k,\ell) = (c_{r,a}^{(0)}(k,\ell),\ c_{r,a}^{(1)}(k,\ell), \ldots, c_{r,a}^{(n)}(k,\ell))^T \in \mathbf{C}^{n+1}$$
$$c_{a,r}(\ell,k) = (c_{a,r}^{(0)}(\ell,k),\ c_{a,r}^{(1)}(\ell,k), \ldots, c_{a,r}^{(n)}(\ell,k))^T \in \mathbf{C}^{n+1}.$$

In the appendix we show that the gradient and Hessian can be expressed in terms of the crosscorrelation vectors as

$$\nabla_x f_1(x,y) = 2\Re \sum_{k=0}^{n} c_{r,r}(k,k)$$
$$\nabla_x f_2(x,y) = -2\Re \sum_{k=0}^{n} c_{a,a}(k,k)$$
$$\nabla_{xx}^2 f_1(x,y) = -2\Re \sum_{k=0}^{n}\sum_{\ell=0}^{n} (\overline{c_{r,r}(k,\ell)} + c_{r,r}(\ell,k)) c_{r,r}(k,\ell)^T$$
$$\nabla_{xx}^2 f_2(x,y) = 2\Re \sum_{k=0}^{n}\sum_{\ell=0}^{n} (\overline{c_{r,a}(k,\ell)} + c_{a,r}(\ell,k))(\overline{c_{r,a}(k,\ell)} + c_{a,r}(\ell,k))^H.$$

Similar expressions hold for the remaining partial derivatives.

Representing the gradient and Hessian using the notion of crosscorrelation does not immediately make it cheaper to evaluate them. However, this representation is essential for the next section where we use a fast Fourier transform to compute the crosscorrelations, allowing us to improve the complexity for evaluating the gradient and Hessian to order $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$, respectively.

III. IMPLEMENTATION

In this section we first discuss how the Levinson–Durbin algorithm can be used to speed up certain steps of Newton's method that require the Cholesky factorization of the inverse of a positive definite Toeplitz matrix. Next we show how the gradient and Hessian can be evaluated efficiently using fast Fourier transforms. The technique we use is inspired by [31], which presents a method for evaluating the derivatives of a logarithmic barrier function for the cone of positive semidefinite real-valued Toeplitz matrices. Finally we discuss how to use the problem structure to find a good initial point for Newton's method.

A. Levinson-Durbin algorithm

Since $\mathbf{dom}\, f$ is not all of $\mathbf{R}^{n+1} \times \mathbf{R}^n$ it is important to make sure that $(x_k, y_k) + t_k(\Delta x_k, \Delta y_k) \in \mathbf{dom}\, f$ before the line search condition (4) is evaluated. Checking if $(x_k + t_k\Delta x_k, y_k + t_k\Delta y_k) \in \mathbf{dom}\, f$ can be done by attempting a Cholesky factorization of $\mathcal{T}(x_k + t_k\Delta x_k, y_k + t_k\Delta y_k)$ or its inverse. The cost of computing the Cholesky decomposition of $\mathcal{T}(x_k + t_k\Delta x_k, y_k + t_k\Delta y_k)^{-1}$ is of order $\mathcal{O}(n^2)$ using the Levinson–Durbin algorithm. If the Toeplitz matrix is not positive definite the factorization can be terminated early by studying the magnitude of the so-called *reflection coefficients* (see, for example, [32, §C8.2]).

B. Efficient evaluation of gradient and Hessian

The method for evaluating the gradient and Hessian is based on the discrete Fourier transform (DFT), which can be defined in matrix form as follows. The DFT of length N of a vector $z \in \mathbf{C}^{n+1}$ is the vector $Z \in \mathbf{C}^N$ given by $Z = Wz$, where $W \in \mathbf{C}^{N\times(n+1)}$ is the DFT matrix defined elementwise by

$$W_{\ell k} = e^{-2\pi i \ell k/N}, \qquad 0 \leq \ell \leq N-1,\ 0 \leq k \leq n. \tag{5}$$

Note that with this definition, a vector and its DFT do not necessarily have the same length. An alternative interpretation of Z is that it is the DFT of length N of a sequence of length N obtained by appending $N-n-1$ zeros to z.

If $N \geq 2(n+1)$ the crosscorrelation between two vectors of length $n+1$ can be computed using the DFT [33, §8]. For $0 \leq k \leq n$, let $R_k = Wr_k$ and $A_k = Wa_k$ be the DFTs of r_k and a_k, respectively. Then

$$c_{r,r}(k,k) = \frac{1}{N} W^H (R_k \circ \overline{R_k}) \tag{6}$$

$$c_{a,a}(k,k) = \frac{1}{N} W^H (A_k \circ \overline{A_k}), \tag{7}$$

where $\circ$ denotes the Hadamard product. The cost of computing R_k and A_k for all $0 \leq k \leq n$ is of order $\mathcal{O}(n^2 \log n)$ if a fast Fourier transform is used and if N is of order $\mathcal{O}(n)$. In the complexity analysis of the remainder of the paper we will assume that $N = \mathcal{O}(n)$; for instance, $N = 2(n+1)$.

Computing the gradient: From (6) it follows that

$$\begin{aligned}\nabla_x f_1(x,y) &= 2\Re \sum_{k=0}^{n} c_{r,r}(k,k) = \frac{2}{N} \Re \sum_{k=0}^{n} W^H (R_k \circ \overline{R_k}) \\ &= \frac{2}{N} \Re\bigg(W^H \sum_{k=0}^{n} R_k \circ \overline{R}_k \bigg).\end{aligned}$$

A similar argument shows that

$$\begin{aligned}\nabla_y f_1(x,y) &= -\frac{2}{N} \Im \bigg[W^H \bigg(\sum_{k=0}^{n} R_k \circ \overline{R}_k \bigg) \bigg]_{1:n} \\ \nabla_x f_2(x,y) &= -\frac{2}{N} \Re \bigg(W^H \bigg(\sum_{k=0}^{n} A_k \circ \overline{A}_k \bigg) \bigg) \\ \nabla_y f_2(x,y) &= \frac{2}{N} \Im \bigg[W^H \bigg(\sum_{k=0}^{n} A_k \circ \overline{A}_k \bigg) \bigg]_{1:n}.\end{aligned}$$

The cost of computing the sums $\sum_{k=0}^{n} R_k \circ \overline{R_k}$ and $\sum_{k=0}^{n} A_k \circ \overline{A}_k$ is of order $\mathcal{O}(n^2)$. Multiplying a vector with W^H incurs a cost of order $\mathcal{O}(n \log n)$ if a fast inverse Fourier transform is used. Consequently, given the matrix A, these expressions allow us to compute the gradient at a cost of order $\mathcal{O}(n^2)$.

Computing the Hessian: In the appendix we derive the following expressions that allow us to compute the Hessian of f efficiently. Define $F \in \mathbf{C}^{N\times N}$ and $G \in \mathbf{C}^{N \times N}$ by

$$F = \sum_{k=0}^{n} R_k R_k^H, \qquad G = \sum_{k=0}^{n} A_k A_k^H.$$

Then the Hessians can be written as

$$\begin{aligned}\nabla^2_{xx} f_1(x,y) &= -\frac{4}{N^2} \Re \Big(W^H \Re((F \circ F^T) W) \Big) \\ \nabla^2_{yy} f_1(x,y) &= \frac{4}{N^2} \Im \Big[W^H \Im((F \circ F^T) W) \Big]_{1:n,1:n} \\ \nabla^2_{yx} f_1(x,y) &= \frac{4}{N^2} \Im \Big[W^H \Re((F \circ F^T) W) \Big]_{1:n,0:n} \\ \nabla^2_{xx} f_2(x,y) &= \frac{4}{N^2} \Re \Big(W^H \Re((F \circ G^T + F^T \circ G) W) \Big) \\ \nabla^2_{yy} f_2(x,y) &= -\frac{4}{N^2} \Im \Big[W^H \Im((F \circ G^T + F^T \circ G) W) \Big]_{1:n,1:n} \\ \nabla^2_{yx} f_2(x,y) &= -\frac{4}{N^2} \Im \Big[W^H \Re((F \circ G^T + F^T \circ G) W) \Big]_{1:n,0:n}.\end{aligned}$$

The cost of computing F and G is of order $\mathcal{O}(n^3)$. Computing $F \circ F^T$ and $F \circ G^T + F^T \circ G$ incurs a cost of order $\mathcal{O}(n^2)$. The complexity of multiplying an $N \times N$ matrix by W^H and W is of order $\mathcal{O}(n^2 \log n)$ if a fast Fourier transform is used. Hence, these expressions allow us to evaluate the Hessian at a cost of order $\mathcal{O}(n^3)$.

At first glance it seems like it is necessary to *e.g.* separately compute both $W^H \Re(D)$ and $W^H \Im(D)$, where $D = (F \circ F^T)W$, to evaluate $\nabla^2 f_1(x,y)$. Multiplication by W^H is implemented using a fast inverse Fourier transform (IFFT), which suggests that one has to compute $2(n+1)$ IFFTs to evaluate $\nabla^2 f_1(x,y)$. However, by using conjugate symmetry of the IFFT (see the appendix for more details), both $W^H \Re(D)$ and $W^H \Im(D)$ can be recovered from $W^H D$. It is therefore only necessary to compute $n+1$ IFFTs to evaluate $W^H \Re(D)$ and $W^H \Im(D)$.

C. Initialization

Newton's method is often very efficient in a neighborhood of a local minimizer, and its performance can greatly benefit from a good starting point. We have found that the following strategy to compute a starting point works well. A preliminary starting point is computed by averaging the sample covariance matrix along its diagonals. The resulting Toeplitz matrix is not guaranteed to be positive definite, *i.e.* it may fail to belong to $\mathbf{dom}\, f$. If the preliminary starting point does not belong to $\mathbf{dom}\, f$ we compute a new starting point as the maximum likelihood estimate of a *circulant* covariance matrix. The maximum likelihood estimate of a circulant covariance matrix can be computed efficiently using $K+1$ fast Fourier transforms, where K is the number of measurements [34].

IV. Simulations

In this section we present a numerical study of the proposed method for Toeplitz covariance matrix estimation. Our implementation, which we call NML for 'Newton maximum likelihood estimator', is written in C and includes all the enhancements presented in the previous section. It uses BLAS [35] and LAPACK [36] for linear algebra operations and FFTW [37] to compute discrete Fourier transforms. NML is available online at

https://github.com/dance858/Toeplitz-covariance-estimation along with the code to run the numerical examples. The experiments were performed on an Intel Core i5-8250U 1.60GHz CPU.

We consider two types of experiments. In the first experiment we assess several methods for Toeplitz covariance matrix estimation in terms of the Euclidean distance to the true covariance matrix. In the second experiment we consider direction-of-arrival estimation and demonstrate that the performance of MUSIC is enhanced by replacing the sample covariance matrix with the covariance matrix estimate obtained by solving (1) with NML. Comparisons are made with the following methods:

1) *Diagonal averaging* (DA) [38, §4.5], [7], [39]. A simple estimator is obtained by averaging the sample covariance along its diagonals to obtain a Toeplitz matrix. This estimate is not guaranteed to be positive definite and in our experiments we noticed that it sometimes failed to be positive definite for small sample sizes.
2) *Asymptotically maximum likelihood estimator* (AML) [5]. This estimator approximates the maximum likelihood estimator as the number of samples tends to infinity. It is computable in closed form, at a cost of order $\mathcal{O}(n^4)$. The estimator requires that $K \geq n+1$, *i.e.* that more data points are available than the dimension of the covariance matrix. The obtained estimate is not guaranteed to be positive definite and in our experiments we noticed that it sometimes failed to be positive definite for small sample sizes.
3) *Maximum likelihood estimator via ATOM* (ATOM) [23], [22]. An estimate is obtained by solving the maximum likelihood estimation problem (1) using a majorization-minimization (MM) algorithm. We use the method referred to as ATOM2 in [23]. In every iteration this algorithm solves an SDP as a subproblem using Dykstra's algorithm, and the complexity per iteration of Dykstra's algorithm is $\mathcal{O}(n^3)$. ATOM requires the user to specify a parameter (denoted by γ_0 in [23]). As recommended in [23], we used $\gamma_0 = 0.1$ (we also tried $\gamma_0 \in \{0.001, 0.01, 1, 10\}$ but $\gamma_0 = 0.1$ performed the best). We thank the authors of [23] for providing an implementation of ATOM.

A. Distance to true covariance matrix

In the first experiment we compare the algorithms above in terms of the mean-squared error

$$\text{MSE} = \mathbf{E}[\|t - \hat{t}\|_2^2], \tag{8}$$

where $t \in \mathbf{C}^{n+1}$ is the first row of the true Toeplitz covariance matrix, and $\hat{t} \in \mathbf{C}^{n+1}$ is the first row of the estimated Toeplitz matrix. The mean-squared error is estimated numerically using 500 Monte Carlo runs. In the experiment below we also report a benchmark based on a Cramér-Rao bound (CRB) found in [40].

Data generation: To form the true Toeplitz covariance matrix $T \in \mathbf{C}^{n+1}$, we draw $2(n+1)$ independent samples $a(1), \dots, a(n+1), b(1), \dots, b(n+1) \in \mathbf{R}$ from a real-valued zero-mean Gaussian distribution with unit variance. The first row of T is then chosen to be equal to the autocorrelation of the complex-valued sequence $\{a(k) + ib(k),\, 1 \leq k \leq n+1\}$. The Toeplitz matrix computed this way is guaranteed to be positive definite [38, §1.3].

To generate observations of $z \sim \mathcal{CN}(0, T)$, we first compute the Cholesky decomposition $T = UU^H$. For $1 \leq k \leq K$, let $z(k) = U(u(k) + iv(k))$, where $u(k) \in \mathbf{R}^{n+1}$ and $v(k) \in \mathbf{R}^{n+1}$ are independently drawn from a real-valued zero-mean Gaussian distribution with covariance matrix $\frac{1}{2}I$. Then $z(1), \dots, z(K)$ are independent observations of $z \sim \mathcal{CN}(0, T)$ [16, Theorem 15.1].

Numerical results: We let $n = 49$ so the covariance matrix to be estimated is of size 50×50. Figure 1a shows the mean-squared error versus the number of samples K. The average runtime for each estimator is shown in Table I.

In Figure 1a we see that the mean-squared error of both NML and ATOM closely align with the CRB. Interestingly, AML exhibits poor performance in terms of the performance metric (8). Nevertheless, in the next section we will see that in applications, the poor performance with respect to this performance metric does not necessarily reflect the quality of the covariance estimate obtained from AML.

In [5] the authors underscore the computational efficiency as a salient characteristic of AML. They also note that attempting to solve the Toeplitz ML estimation problem (1) is inherently complex, rendering exact solutions of limited relevance in practical scenarios. While this assertion might have been accurate over two decades ago, Table I suggests that NML is more than two orders of magnitude faster than AML. In Figure 1b we investigate this further by showing the average runtime of the estimators for $n+1 \in \{5, 15, 25, \dots, 105\}$. The runtime is averaged over 20 problem instances for each value of n and $K = n+1$ samples are used. In Figure 1b we see that NML is at least an order of magnitude faster than both AML and ATOM, regardless of the dimension n.

TABLE I
AVERAGE RUNTIME IN SECONDS.

NML	AML	DA	ATOM
0.0076	1.521	0.0002	2.380

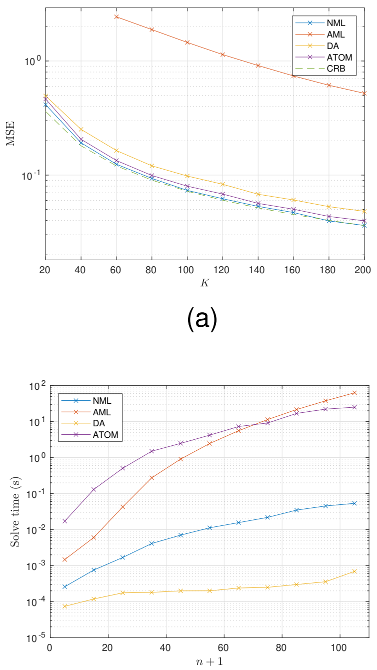
B. Direction-of-arrival estimation

We consider a scenario with M sources emitting signals that impinge on an array of $n+1$ sensors. The arrival angle of source k relative the array normal direction is θ_k, and the output of the array is $z(t) \in \mathbf{C}^{n+1}$. A common model for the array output is [38, §6.2]

$$z(t) = A(\theta)s(t) + e(t), \tag{9}$$

where $\theta = (\theta_1, \dots, \theta_M)^T \in \mathbf{R}^M$ is the vector of arrival angles, $A(\theta) \in \mathbf{C}^{(n+1)\times M}$ is a steering matrix whose exact expression depends on the geometry of the array, $s(t) = (s_1(t), \dots, s_M(t))^T \in \mathbf{C}^M$ is the vector of source signals, and $e(t) \in \mathbf{C}^{n+1}$ represents noise. Let $P \in \mathbf{H}^{M\times M}$ denote the source signal covariance matrix, and assume that the noise is independent of the source signals and is white with variance σ^2. Then the covariance matrix of the array output is

$$T = A(\theta)PA(\theta)^H + \sigma^2 I. \tag{10}$$



(a)

(b)

Fig. 1. (a) MSE (8) versus number of samples. (b) Average runtime versus dimension.

To estimate the arrival angles given a finite set of samples $z(1), \ldots, z(K)$ of the array output, the multiple signal classification (MUSIC) algorithm can be used. To infer the arrival angles, MUSIC processes an estimate of T (see, for example, [38, page 161] for more details). The sample covariance matrix is commonly used as an estimate of T.

If the sensors are equally spaced along a straight line and if the source signal covariance matrix P is assumed to be diagonal, it turns out that the covariance matrix T given by (10) is Toeplitz. This justifies using a Toeplitz covariance estimate together with MUSIC in lieu of the sample covariance, since the true covariance matrix is Toeplitz.

Cramér–Rao bounds have been derived to assess the performance of algorithms for direction-of-arrival estimation. With no assumptions on P the CRB can be found in *e.g.* [38, B.6.33], which we will refer to as the unstructured CRB or U-CRB. It has been shown that MUSIC approaches the U-CRB asymptotically as $K \to \infty$ for uncorrelated signals [41]. However, for a uniform linear array, the a priori knowledge that the signals are uncorrelated can be taken into account to obtain a lower CRB [42, §8.4], which we will refer to as the structured CRB or S-CRB. In the simulations below we will see that MUSIC together with the Toeplitz covariance estimate computed using `NML` in many cases attain the S-CRB.

When a uniform linear array is employed, a variant of the MUSIC algorithm known as Root-MUSIC [38, page 161] can be used. MUSIC and Root-MUSIC have similar asymptotic properties, but in certain scenarios Root-MUSIC has been found to perform better [43], [29]. In the experiment below we therefore focus on the impact of using a Toeplitz structured covariance estimate for Root-MUSIC. A similar setting was considered in [5].

Setup: We consider a scenario with $M = 5$ sources. The noise $e(t)$ in (9) is drawn from a complex circular Gaussian distribution with zero mean and covariance matrix $\sigma^2 I$. The source signal covariance matrix is $P = pI$ for some $p > 0$, and the signal-to-noise ratio (SNR) in dB is defined as SNR $= 10\log_{10}(p/\sigma^2)$. In all experiments we let $p = 1$ and vary σ^2 to obtain an appropriate SNR. As performance metrics, we consider the mean-squared angle estimation error

$$\mathbf{E}[\|\theta - \hat{\theta}\|_2^2], \tag{11}$$

and the mean-squared normalized Frobenius distance error

$$\mathbf{E}\left[\frac{\|T - \hat{T}\|_F^2}{\|T\|_F^2}\right]. \tag{12}$$

In (11), θ is the vector of true arrival angles and $\hat{\theta}$ is an estimate obtained by using MUSIC together with the sample covariance (`SC`) or a Toeplitz structured estimate obtained from `NML`, `AML` or `DA`. In (12), T is the true covariance matrix given by (10) and $\hat{T}$ is the estimate. The expectations in (11) and (12) are estimated using 1000 Monte Carlo runs.

Both `ATOM` and `NML` are based on the same optimization problem. Since the experiments in §IV-A indicate that `NML` is significantly faster than `ATOM`, we do not include `ATOM` in the experiments below.

Numerical results: We consider several different scenarios where we vary the number of samples K, the signal-to-noise ratio, the number of sensors m, and the angle separation $\Delta\theta$. In the first three scenarios the true arrival angles are $(-10°, -5°, 0°, 5°, 10°)$. The results can be summarized as follows (when we refer to *e.g.* `NML` below, we mean MUSIC together with the covariance estimate obtained from `NML`).

1) Figures 2a and 2b show the mean-squared error (11) and the average Frobenius error (12), respectively, versus K. The SNR is 5 dB and $m = 15$. The number of samples varies between 15 (which is the smallest number of samples required for `AML`) and 500. Figures 2c and 2d correspond to an identical setting except that the SNR is 10 dB.

   In figures 2a and 2c we make the following observations. MUSIC with the sample covariance approaches U-CRB as K increases, whereas `NML` and `AML` approach S-CRB. Although both `NML` and `AML` approach S-CRB as K increases, `NML` sometimes has much greater angle estimation accuracy in the regime with few samples. For instance, for SNR = 10 db and 15 samples (which corresponds to the first data point in Figure 2c), the estimation error of `AML` is three orders of magnitude larger than the estimation error of `NML`.

   In Figure 2a it is interesting to note that `DA` has some advantage over `SC` for SNR = 5 dB and $K \leq 250$. However, in Figure 2c where SNR = 10 dB, the advantage of `DA` compared to `SC` no longer exists. In particular, `DA` does not approach U-CRB for SNR = 10 dB.

   Lastly, it is enlightening to compare the two performance metrics (11) and (12) using Figure 2. `NML` achieves the smallest angle estimation error as well as the smallest Frobenius distance error. On the other hand, `AML` is second best in terms of angle estimation error but performs among the worst in terms of Frobenius distance error. Conversely, DA which performs worst in terms of angle estimation error, performs second best in terms of Frobenius distance error. This indicates that the Frobenius distance error is not closely related to the accuracy of MUSIC, and the Frobenius distance to the true covariance matrix may therefore in this context not be an appropriate measure of the quality of a covariance matrix estimate. A similar observation was made in [5]. In the remaining experiments the Frobenius distance error is therefore omitted.

2) Figure 3a shows the mean-squared error (11) versus SNR, for fixed $m = 15$ and $K = 100$. Figure 3b corresponds to an identical setting except that the number of samples is $K = 200$.

   In Figure 3a we make the observation that in the regime with low SNR, the covariance estimates from both NML and AML significantly improve the accuracy of MUSIC. For SNR $\geq$ 5 dB, NML attains the S-CRB whereas AML is close to S-CRB but never seems to attain it. Finally, we note that in the regime with low SNR, MUSIC with the covariance estimate from DA yields some improvement over MUSIC with the sample covariance, but in the regime with high SNR it performs significantly worse and never attains the U-CRB. Similar observations can be made in Figure 3b.

3) Figure 4a shows the mean-squared error (11) versus the number of sensors, for fixed SNR = 5 dB and $K = 100$. Figure 4b corresponds to an identical setting except that the SNR is equal to 10 dB.

   In figures 4a and 4b we see that NML and AML can both enhance MUSIC in certain intervals. Interestingly, NML is always better or on par with SC and attains the S-CRB (which for large m coincides with U-CRB) whereas AML does not attain S-CRB. Furthermore, the performance of DA is always worse or on par with SC.

In the fourth scenario the true arrival angles are $(-2\Delta\theta, -\Delta\theta, 0, \Delta\theta, 2\Delta\theta)$, with $\Delta\theta$ varying from 2° to 8°. Figures 5a and 5b show the mean-squared error (11) versus $\Delta\theta$ for $m = 20$, SNR = 10 dB, $K = 100$, and $m = 20$, SNR = 10 dB, $K = 200$, respectively. We see that NML and AML can significantly improve the capability of resolving sources that are close. For instance, in Figure 5b the estimation error for MUSIC together with the sample covariance at $\Delta\theta \approx 3°$ (the fourth data point) is two orders of magnitude larger than the estimation error of MUSIC together with the covariance estimate from NML and AML.

The average runtimes were 1.4, 12.8 and 0.08 milliseconds for NML, AML and DA, respectively.

V. Discussion

In this paper we have shown how to efficiently implement Newton's method for maximum likelihood estimation of a Toeplitz covariance matrix. The proposed method is orders of magnitude faster than other algorithms for Toeplitz covariance estimation. For direction-of-arrival estimation with 15 sensors, our open-source implementation can solve the corresponding optimization problem in a millisecond, possibly allowing it to be used to enhance real-time implementations of algorithms for direction-of-arrival estimation.

We compared the impact of our covariance estimator on MUSIC with the impact of other typical Toeplitz covariance estimators. The empirical results clearly demonstrate the superior performance of our method. Remarkably, the angle estimation error of MUSIC together with NML is smaller (or no worse) than the estimation error of MUSIC together with any other covariance estimator for essentially *every* configuration of the experimental parameters we tested. The merits of NML are most pronounced in the regime with low SNR and few samples, and our findings suggest that the covariance estimate obtained from NML also improves MUSIC's capability to resolve sources with small spatial separation.

The numerical experiments raise some important questions. First of all, they seem to challenge a common belief that diagonal averaging of the sample covariance is a practical tool to enhance the performance of MUSIC [39], [7]. Secondly, they challenge the presumption that solving the Toeplitz maximum likelihood estimation problem exactly is prohibitively expensive for practical applications [5].

We hope that the availability of our solver will encourage the community to apply the proposed Toeplitz covariance estimator to other applications.

Appendix A

In this appendix we present derivations that have been omitted from the main text. Due to space limitations we only present derivations related to $\nabla^2_{xx} f_2(x, y)$ (the omitted derivations are similar). The following identities will be useful. If C is Hermitian, then for any matrix B of compatible dimension:

$$\mathbf{Tr}\left\{C(B^H + B)\right\} = 2\Re\,\mathbf{Tr}(CB). \tag{13}$$

We will use the fact that for any matrix B:

$$\mathbf{Tr}(B^H) = \overline{\mathbf{Tr}(B)}. \tag{14}$$

For complex-valued vectors a and b:

$$\Re(\overline{a}b^T) = \Re(ab^H) \tag{15}$$

$$\Im(\overline{a}b^T) = -\Im(ab^H) \tag{16}$$

$$(a \circ \overline{b})(a \circ \overline{b})^H = (aa^H) \circ (bb^H)^T \tag{17}$$

$$(a \circ \overline{b})(\overline{a} \circ b)^H = (aa^T) \circ (bb^T)^H \tag{18}$$

$$(a \circ \overline{b})(\overline{a} \circ b)^T = (aa^H) \circ (bb^H)^T. \tag{19}$$

We recall the notation that $\mathcal{T}(x, y)^{-1} = RR^H$, $S = LL^H$ and $A = RR^H L$.

A. Derivation of partial derivatives

Below we write $\mathcal{T}$ instead of $\mathcal{T}(x, y)$. From standard calculus it follows that for $0 \leq j \leq n$:

$$\begin{aligned}\frac{\partial f_2(x, y)}{\partial x_j} &= -\,\mathbf{Tr}\left(\mathcal{T}^{-1}(E_j^T + E_j)\mathcal{T}^{-1}S\right)\\ &\overset{(13)}{=} -2\Re\,\mathbf{Tr}(\mathcal{T}^{-1}S\mathcal{T}^{-1}E_j) = -2\Re\,\mathbf{Tr}(A^H E_j A).\end{aligned}$$

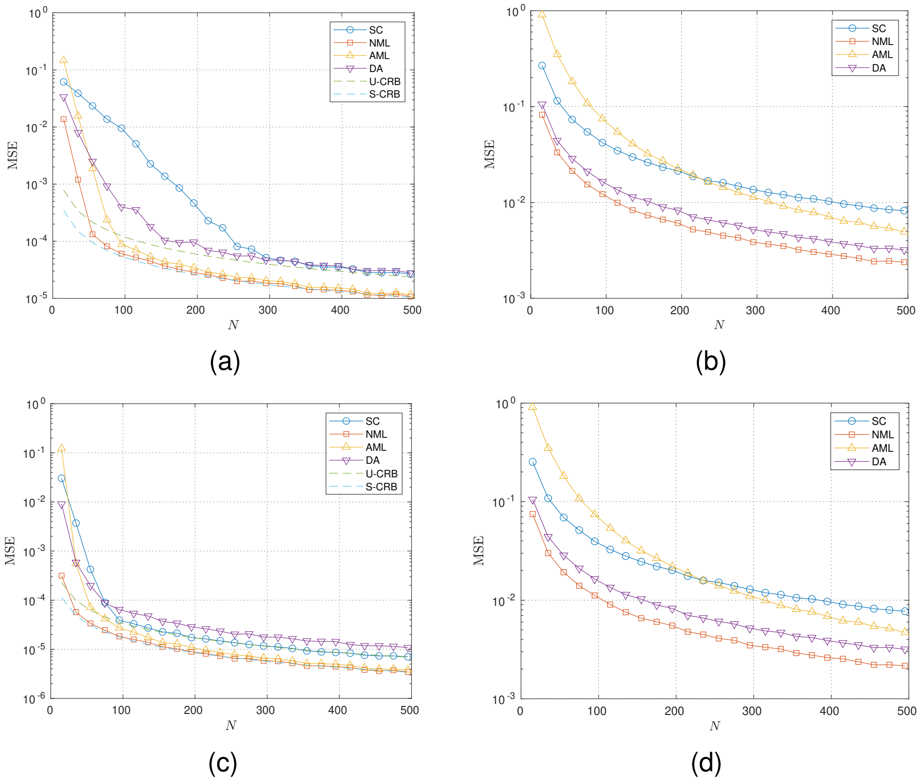


Fig. 2. (a) MSE (11) versus number of samples when $m = 15$ and SNR = 5 dB. (b) Average normalized Frobenius distance (12) versus number of samples when $m = 15$ and SNR = 5 dB. (c) MSE (11) versus number of samples when $m = 15$ and SNR = 10 dB. (d) Average normalized Frobenius distance (12) versus number of samples when $m = 15$ and SNR = 10 dB.

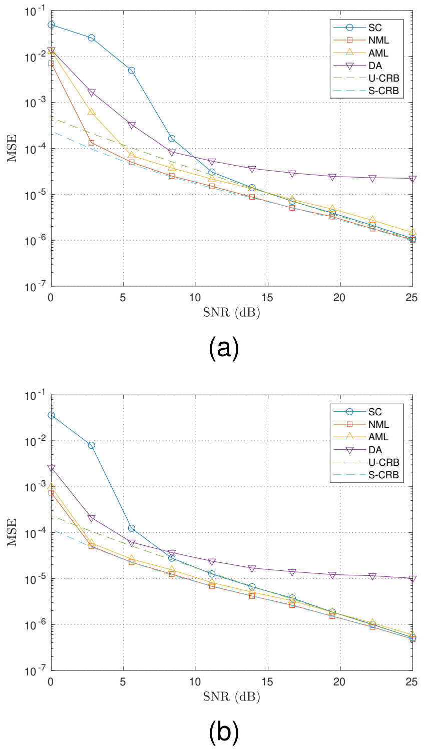


Fig. 3. (a) MSE versus SNR when $m = 15$ and $K = 100$. (b) MSE versus SNR when $m = 15$ and $K = 200$.

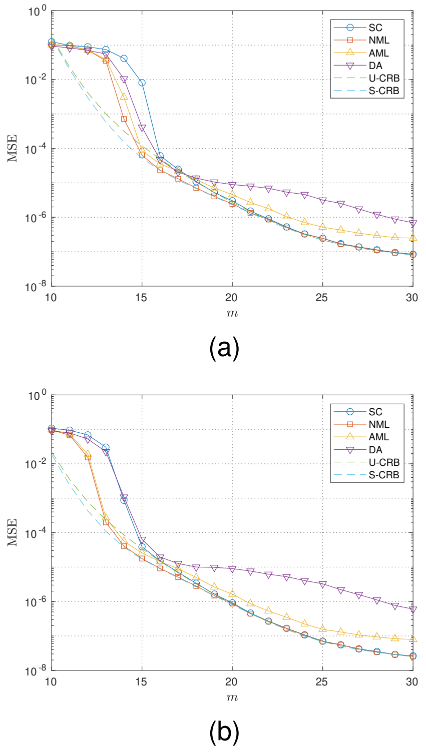
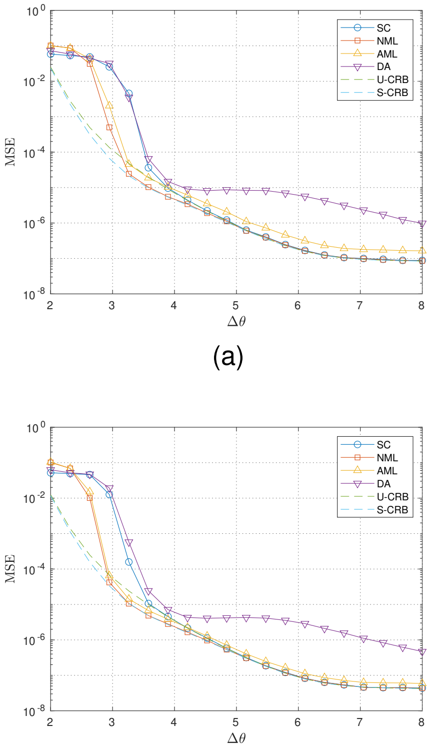


Fig. 4. (a) MSE versus number of sensors when SNR = 5 dB and $K = 100$. (b) MSE versus number of sensors when SNR = 10 dB and $K = 100$.



(a)

(b)

Fig. 5. (a) MSE versus angle separation when $m = 20$, SNR = 10 dB and $K = 100$. (b) MSE versus angle separation when $m = 20$, SNR = 10 dB and $K = 200$.

Furthermore,

$$\frac{\partial^2 f_2(x,y)}{\partial x_t \partial x_j} = \mathbf{Tr}(\mathcal{T}^{-1}(E_t^T + E_t)\mathcal{T}^{-1}(E_j^T + E_j)\mathcal{T}^{-1}S)$$
$$+ \mathbf{Tr}(\mathcal{T}^{-1}(E_j^T + E_j)\mathcal{T}^{-1}(E_t^T + E_t)\mathcal{T}^{-1}S)$$
$$\overset{(14)}{=} 2\Re\,\mathbf{Tr}(\mathcal{T}^{-1}S\mathcal{T}^{-1}E_t^T\mathcal{T}^{-1}E_j^T + \mathcal{T}^{-1}S\mathcal{T}^{-1}E_t^T\mathcal{T}^{-1}E_j)$$
$$+ 2\Re\,\mathbf{Tr}(\mathcal{T}^{-1}S\mathcal{T}^{-1}E_t\mathcal{T}^{-1}E_j^T + \mathcal{T}^{-1}S\mathcal{T}^{-1}E_t\mathcal{T}^{-1}E_j)$$
$$= 2\Re\,\mathbf{Tr}(A^H E_t^T R R^H E_j^T A) + 2\Re\,\mathbf{Tr}(A^H E_t^T R R^H E_j A)$$
$$+ 2\Re\,\mathbf{Tr}(A^H E_t R R^H E_j^T A) + 2\Re\,\mathbf{Tr}(A^H E_t R R^H E_j A).$$

B. Hessian expressed using crosscorrelation

Note that

$$\mathbf{Tr}(A^H E_t^T R R^H E_j^T A) = \sum_{k=0}^{n}\sum_{\ell=0}^{n}(a_\ell^H E_t^T r_k)(r_k^H E_j^T a_\ell)$$
$$= \sum_{k=0}^{n}\sum_{\ell=0}^{n}\overline{c_{r,a}^{(t)}(k,\ell)}\cdot\overline{c_{a,r}^{(j)}(\ell,k)}$$
$$\mathbf{Tr}(A^H E_t^T R R^H E_j A) = \sum_{k=0}^{n}\sum_{\ell=0}^{n}(a_\ell^H E_t^T r_k)(r_k^H E_j a_\ell)$$
$$= \sum_{k=0}^{n}\sum_{\ell=0}^{n}\overline{c_{r,a}^{(t)}(k,\ell)}c_{r,a}^{(j)}(k,\ell)$$
$$\mathbf{Tr}(A^H E_t R R^H E_j^T A) = \sum_{k=0}^{n}\sum_{\ell=0}^{n}(a_\ell^H E_t r_k)(r_k^H E_j^T a_\ell)$$
$$= \sum_{k=0}^{n}\sum_{\ell=0}^{n}c_{a,r}^{(t)}(\ell,k)\overline{c_{a,r}^{(j)}(\ell,k)}$$
$$\mathbf{Tr}(A^H E_t R R^H E_j A) = \sum_{k=0}^{n}\sum_{\ell=0}^{n}(a_\ell^H E_t r_k)(r_k^H E_j a_\ell)$$
$$= \sum_{k=0}^{n}\sum_{\ell=0}^{n}c_{a,r}^{(t)}(\ell,k)c_{r,a}^{(j)}(k,\ell).$$

Hence, for $0 \le j \le n$, $0 \le t \le n$:

$$\frac{\partial^2 f_2(x,y)}{\partial x_t \partial x_j} = 2\Re\Bigg(\sum_{k=0}^{n}\sum_{\ell=0}^{n}\overline{c_{r,a}^{(t)}(k,\ell)}\cdot\overline{c_{a,r}^{(j)}(\ell,k)} + \overline{c_{r,a}^{(t)}(k,\ell)}\cdot$$
$$c_{r,a}^{(j)}(k,\ell) + c_{a,r}^{(t)}(\ell,k)\cdot\overline{c_{a,r}^{(j)}(\ell,k)} + c_{a,r}^{(t)}(\ell,k)\cdot c_{r,a}^{(j)}(k,\ell)\Bigg)$$
$$= 2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}(\overline{c_{r,a}^{(t)}(k,\ell)} + c_{a,r}^{(t)}(\ell,k))(\overline{c_{a,r}^{(j)}(\ell,k)} + c_{r,a}^{(j)}(k,\ell)).$$

By assembling partial derivatives we conclude that

$$\nabla_{xx}^2 f_2(x,y) = 2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}(\overline{c_{r,a}(k,\ell)} + c_{a,r}(\ell,k))(\overline{c_{r,a}(k,\ell)}$$
$$+ c_{a,r}(\ell,k))^H.$$

C. Efficient evaluation of Hessian

By using (15)-(19) one can show that

$$2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}\overline{c_{r,a}(k,\ell)}c_{r,a}(k,\ell)^T = \frac{2}{N^2}\Re\Big(W^H(F\circ G^T)W\Big)$$
$$2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}\overline{c_{r,a}(k,\ell)}c_{a,r}(\ell,k)^H = \frac{2}{N^2}\Re\Big(W^H(F\circ G^T)\overline{W}\Big)$$
$$2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}c_{a,r}(\ell,k)c_{r,a}(k,\ell)^T = \frac{2}{N^2}\Re\Big(W^H(G\circ F^T)\overline{W}\Big)$$
$$2\Re\sum_{k=0}^{n}\sum_{\ell=0}^{n}c_{a,r}(\ell,k)c_{a,r}(\ell,k)^H = \frac{2}{N^2}\Re\Big(W^H(G\circ F^T)W\Big).$$

Hence, $\nabla_{xx}^2 f_2(x,y) = \frac{2}{N^2}\Re(W^H D(W + \overline{W}))$ where $D = F\circ G^T + F^T \circ G$. From symmetry it follows that D is real. Hence, $D(W + \overline{W}) = DW + \overline{DW} = 2\Re(DW)$, implying that $\nabla_{xx}^2 f(x,y) = \frac{4}{N^2}\Re(W^H\Re(DW))$.

D. Conjugate symmetry of the IFFT

Consider the complex-valued sequence $w_k = c_k + ib_k$, $0 \le k \le N-1$, where $c_k \in \mathbf{R}$ and $b_k \in \mathbf{R}$ are the real and imaginary parts of w_k, respectively. Denote the inverse discrete Fourier transforms of $(w_0, \ldots, w_{N-1}) \in \mathbf{C}^N$, $(c_0, \ldots, c_{N-1}) \in \mathbf{R}^N$ and $(b_0, \ldots, b_{N-1}) \in \mathbf{R}^N$ by $(W_0, \ldots, W_{N-1}) \in \mathbf{C}^N$, $(C_0, \ldots, C_{N-1}) \in \mathbf{C}^N$ and $(B_0, \ldots, B_{N-1}) \in \mathbf{C}^N$, respectively. Then it holds that $A_0 = \Re W_0$, $B_0 = \Im W_0$, and for $1 \le k \le N-1$:

$$A_k = \frac{1}{2}(\Re W_k + \Re W_{N-k}) + i\frac{1}{2}(\Im W_k - \Im W_{N-k})$$
$$B_k = \frac{1}{2}(\Im W_k + \Im W_{N-k}) + i\frac{1}{2}(\Re W_{N-k} - \Re W_k).$$

Hence, the inverse discrete Fourier transform of the separate real and imaginary parts of a complex-valued sequence can be recovered from the inverse discrete Fourier transform of the complex-valued sequence.

References

[1] Babak Mohammadzadeh Asl and Ali Mahloojifar. A low-complexity adaptive beamformer for ultrasound imaging using structured covariance matrix. *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, 59, 2012.

[2] D.R. Fuhrmann. Application of Toeplitz covariance estimation to adaptive beamforming and detection. *IEEE Trans. Signal Process.*, 39(10):2194–2198, 1991.

[3] D.A Gray, B. D. O Anderson, and P. K. Sim. Estimation of structured covariances with application to array beamforming. *Circuits, Syst., Signal Process.*, 6(4), 1987.

[4] Y.I. Abramovich, D.A. Gray, A.Y. Gorokhov, and N.K. Spencer. Positive-definite Toeplitz completion in DOA estimation for nonuniform linear antenna arrays. I. Fully augmentable arrays. *IEEE Trans. Signal Process.*, 46(9):2458–2471, 1998.

[5] Hongbin Li, P. Stoica, and Jian Li. Computationally efficient maximum likelihood estimation of structured covariance matrices. *IEEE Trans. Signal Process.*, 47(5):1314–1323, 1999.

[6] Yanyan Liu, Xiaoying Sun, and Shishun Zhao. A covariance matrix shrinkage method with Toeplitz rectified target for DOA estimation under the uniform linear array. *AEU - Int. J. Electron. Commun.*, 81:50–55, 2017.

[7] P. Vallet and P. Loubaton. Toeplitz rectification and DOA estimation with music. In *2014 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pages 2237–2241, 2014.

[8] Heng Qiao and Piya Pal. Gridless Line Spectrum Estimation and Low-Rank Toeplitz Matrix Compression Using Structured Samplers: A Regularization-Free Approach. *IEEE Trans. Signal Process.*, 65(9):2221–2236, 2017.

[9] Daniel Romero, Dyonisius Dony Ariananda, Zhi Tian, and Geert Leus. Compressive Covariance Sensing: Structure-based compressive sensing beyond sparsity. *IEEE Signal Process. Mag.*, 33(1):78–93, 2016.

[10] Yuxin Chen, Yuejie Chi, and Andrea J. Goldsmith. Exact and Stable Covariance Estimation From Quadratic Sampling via Convex Programming. *IEEE Trans. Inf. Theory*, 61(7):4034–4059, 2015.

[11] Bosung Kang, Vishal Monga, and Muralidhar Rangaswamy. Computationally efficient Toeplitz approximation of structured covariance under a rank constraint. *IEEE Trans. Aerosp. Electron. Syst.*, 51(1):775–785, 2015.

[12] Naixin Kang, Zheran Shang, and Qinglei Du. Knowledge-Aided Structured Covariance Matrix Estimator Applied for Radar Sensor Signal Detection. *Sensors*, 19(3), 2019.

[13] Marc Arnaudon, Frédéric Barbaresco, and Le Yang. Riemannian Medians and Means With Applications to Radar Signal Processing. *IEEE J. Sel. Topics Signal Process.*, 7(4):595–604, 2013.

[14] Xiaolin Du, Augusto Aubry, Antonio De Maio, and Guolong Cui. Toeplitz Structured Covariance Matrix Estimation for Radar Applications. *IEEE Signal Process. Lett.*, 27:595–599, 2020.

[15] E. Conte, M. Lops, and G. Ricci. Adaptive detection schemes in compound-Gaussian clutter. *IEEE Trans. Aerosp. Electron. Syst.*, 34(4):1058–1069, 1998.

[16] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1997.

[17] J.P. Burg, D.G. Luenberger, and D.L. Wenger. Estimation of structured covariance matrices. *Proc. IEEE*, 70(9):963–974, 1982.

[18] A. Dembo, C.L. Mallows, and L.A. Shepp. Embedding nonnegative definite Toeplitz matrices in nonnegative definite circulant matrices, with application to covariance estimation. *IEEE Trans. Inf. Theory*, 35(6):1206–1212, 1989.

[19] M.I. Miller and D.L. Snyder. The role of likelihood and entropy in incomplete-data problems: Applications to estimating point-process intensities and toeplitz constrained covariances. *Proc. IEEE*, 75(7):892–907, 1987.

[20] D. Fuhrmann, M. Turmon, and M.I. Miller. Efficient implementation of the EM algorithm for maximum likelihood estimation of Toeplitz covariance matrices. *Proc. 22nd Conf. Inf. Science and Systems, Princeton University, Princeton*, 1988.

[21] Prabhu Babu. MELT—Maximum-Likelihood Estimation of Low-Rank Toeplitz Covariance Matrix. *IEEE Signal Process. Lett.*, 23(11):1587–1591, 2016.

[22] Augusto Aubry, Prabhu Babu, Antonio De Maio, and RikhabChand Jyothi. ATOM for MLE of Toeplitz Structured Covariance Matrices for RADAR Applications. In *2022 IEEE Radar Conf. (RadarConf22)*, pages 1–6, 2022.

[23] Augusto Aubry, Prabhu Babu, Antonio De Maio, and Massimo Rosamilia. New Methods for MLE of Toeplitz Structured Covariance Matrices with Applications to RADAR Problems. arXiv:2307.03923. Submitted to *IEEE Trans. Signal Process.*, 2023.

[24] Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE Trans. Signal Process.*, 65(3):794–816, 2017.

[25] D.R. Fuhrmann and M.I. Miller. On the existence of positive-definite maximum-likelihood estimates of structured covariance matrices. *IEEE Trans. Inf. Theory*, 34(4):722–729, 1988.

[26] L. Davis, R. Evans, and E. Polak. Maximum likelihood estimation of positive definite Hermitian Toeplitz matrices using outer approximations. In *Ninth IEEE Signal Process. Workshop on Statistical Signal and Array Process. (Cat. No.98TH8381)*, pages 49–52, 1998.

[27] T. Tony Cai, Zhao Ren, and Harrison H. Zhou. Optimal rates of convergence for estimating toeplitz covariance matrices. *Probability Theory and Related Fields*, 156(1):101–143, June 2013.

[28] Ben A. Johnson, Yuri I. Abramovich, and Xavier Mestre. MUSIC, G-MUSIC, and Maximum-Likelihood Performance Breakdown. *IEEE Trans. Signal Process.*, 56(8):3944–3958, 2008.

[29] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Process. Mag.*, 13(4):67–94, 1996.

[30] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, New York, NY, USA, 2e edition, 2006.

[31] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Math. Prog.*, vol. 93, 2002.

[32] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall international series in systems and control engineering. Prentice Hall, 1989.

[33] Alan V. Oppenheim and Ronald W Schafer. *Discrete-Time signal processing*. Prentice Hall, 1e edition, 1989.

[34] D.R. Fuhrmann and T.A. Barton. Estimation of Block-Toeplitz Covariance Matrices. In *1990 Conf. Record Twenty-Fourth Asilomar Conf. Signals, Syst., Comput, 1990.*, volume 2, pages 779–, 1990.

[35] L Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.*, 28(2):135–151, 2002.

[36] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Soc. Ind., Appl. Math., Philadelphia, PA, third edition, 1999.

[37] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *pieee*, 93(2):216–231, February 2005.

[38] P. Stoica and R.L. Moses. *Spectral Analysis of Signals*. Pearson Prentice Hall, 2005.

[39] J. Cadzow. Signal enhancement using canonical projection operators. In *ICASSP '87. IEEE Int. Conf. Acoust., Speech, Signal Process.*, volume 12, pages 673–676, 1987.

[40] M.J. Turmon and M.I. Miller. Maximum-likelihood estimation of complex sinusoids and Toeplitz covariances. *IEEE Trans. Signal Process.*, 42(5):1074–1086, 1994.

[41] P. Stoica and Arye Nehorai. MUSIC, maximum likelihood, and Cramer-Rao bound. *IEEE Trans. Acoust., Speech, Signal Process*, 37(5):720–741, 1989.

[42] H.L. Van Trees. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. Detection, Estimation, and Modulation Theory. Wiley, 2002.

[43] B.D. Rao and K.V.S. Hari. Performance analysis of Root-Music. *IEEE Trans. Acous., Speech, Signal Process.*, 37(12):1939–1949, 1989.

Daniel Cederberg has an M.S. degree in applied mathematics from Linköping University. He is currently working towards the Ph.D. degree at Stanford University. His research interests include numerical optimization and its applications to signal processing.