On the complex least squares problem with constrained phase

Ivan Markovsky

School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
im@ecs.soton.ac.uk

Abstract

The problem of solving approximately in the least squares sense an overdetermined linear system of equations with complex valued coefficients is considered, where the elements of the solution vector are constrained to have the same phase. A direct solution to this problem is given in [Linear Algebra and Its Applications, Vol. 433, pp. 1719–1721]. An alternative direct solution that reduces the problem to a generalized eigenvalue problem is derived in this paper. The new solution is related to generalized low-rank matrix approximation and makes possible one to use existing robust and efficient algorithms.

Keywords: Linear system of equations, Phase constraint, Low-rank approximation, Total least squares.

1 Introduction

The considered problem is: given a complex valued $m \times n$ matrix A and $m \times 1$ vector b, find a real valued $n \times 1$ vector x and a number ϕ, such that the equation's error or residual of the overdetermined linear system of equations

$$Axe^{\mathbf{i}\phi} \approx b, \qquad (\mathbf{i} \text{ is the imaginary unit})$$

is minimized in the least squares sense, *i.e.*,

$$\text{minimize} \quad \text{over } x \in \mathbb{R}^n \text{ and } \phi \in (-\pi, \pi] \quad \|Axe^{\mathbf{i}\phi} - b\|. \tag{1}$$

Problem (1) is a complex linear least squares problem with constraint that the phase of all elements of the solution have the same phase. An application of (1) to magnetic resonance imaging is discussed in [Byd10, Section 3].

As formulated, (1) is a nonlinear optimization problem. General purpose local optimization methods can be used for solving it, however, this approach has the usual disadvantages of local optimization methods: need of initial approximation, no guarantee of global optimality, convergence issues, and no insight in the geometry of the solutions set. In [Byd10] the following closed form solution of (1) is derived

$$\widehat{x} = \big(\Re(A^H A)\big)^{+} \Re(A^H b e^{-\mathbf{i}\phi}) \tag{2}$$

$$\widehat{\phi} = \frac{1}{2} \angle\big((A^H b)^{\top} \Re(A^H A)^{+} (A^H b)\big), \tag{3}$$

where $\Re(A)/\Im(A)$ is the real/imaginary part, $\angle(A)$ is the angle, A^H is the complex conjugate transpose, and A^+ is the pseudoinverse of A. Moreover, in the case when a solution of (1) is not unique, (2, 3) is a least norm element of the solution set, *i.e.*, a solution (x, ϕ), such that $\|x\|$ is minimized. Expression (2) is the result of minimizing the cost function $\|Axe^{\mathbf{i}\phi} - b\|$ with respect to x, for a fixed ϕ. This is a linear least squares problems (with complex valued data and real valued solution). Then minimization of the cost function with respect to ϕ, for x fixed to its optimal value (2), leads through a nontrivial chain of steps to (3).

2 An alternative solution

Problem (1) is equivalent[1] to the problem

$$\text{minimize} \quad \text{over } x \in \mathbb{R}^n \text{ and } \phi' \in (-\pi, \pi] \quad \|Ax - be^{\mathbf{i}\phi'}\|, \tag{4}$$

where $\phi' = -\phi$. With

$$y_1 := \Re(e^{\mathbf{i}\phi'}) = \cos(\phi') = \cos(\phi) \qquad \text{and} \qquad y_2 := \Im(e^{\mathbf{i}\phi'}) = \sin(\phi') = -\sin(\phi),$$

we have

$$\begin{bmatrix} \Re(be^{\mathbf{i}\phi'}) \\ \Im(be^{\mathbf{i}\phi'}) \end{bmatrix} = \begin{bmatrix} \Re(b) & -\Im(b) \\ \Im(b) & \Re(b) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Then, (4) is furthermore equivalent to the problem

$$\text{minimize} \quad \text{over } x \in \mathbb{R}^n \text{ and } y \in \mathbb{R}^2 \quad \left\| \begin{bmatrix} \Re(A) \\ \Im(A) \end{bmatrix} x - \begin{bmatrix} \Re(b) & -\Im(b) \\ \Im(b) & \Re(b) \end{bmatrix} y \right\| \quad \text{subject to} \quad \|y\| = 1,$$

or

$$\text{minimize} \quad \text{over } z \in \mathbb{R}^{n+2} \quad z^\top C^\top C z \quad \text{subject to} \quad z^\top D^\top D z = 1, \tag{5}$$

with

$$C := \begin{bmatrix} \Re(A) & \Re(b) & -\Im(b) \\ \Im(A) & \Im(b) & \Re(b) \end{bmatrix} \in \mathbb{R}^{2m \times (n+2)} \qquad \text{and} \qquad D := \begin{bmatrix} 0 & 0 \\ 0 & I_2 \end{bmatrix} \in \mathbb{R}^{(n+2)\times(n+2)}. \tag{6}$$

It is well known that a solution of problem (5) can be obtained from the generalized eigenvalue decomposition (GEVD) of the pair of matrices $(C^\top C, D)$. More specifically, the smallest generalized eigenvalue $\lambda_{\min}$ of $(C^\top C, D)$ is equal to the minimum value of (5), *i.e.*,

$$\lambda_{\min} = \|A\widehat{x}e^{\mathbf{i}\widehat{\phi}} - b\|^2.$$

If $\lambda_{\min}$ is simple, a corresponding generalized eigenvector $z_{\min}$ is of the form

$$z_{\min} = \alpha \begin{bmatrix} \widehat{x} \\ -\cos(\widehat{\phi}) \\ \sin(\widehat{\phi}) \end{bmatrix},$$

for some $\alpha \in \mathbb{R}$. We have the following result.

Theorem 1. *Let $\lambda_{\min}$ be the smallest generalized eigenvalue of the pair of matrices $(C^\top C, D)$, defined in (6), and let $z_{\min}$ be a corresponding generalized eigenvector. Assuming that $\lambda_{\min}$ is a simple eigenvalue, problem (1) has unique solution, given by*

$$\widehat{x} = \frac{1}{\|z_2\|} z_1, \quad \widehat{\phi} = \angle(-z_{2,1} + \mathbf{i} z_{2,2}), \qquad \text{where} \quad z_{\min} =: \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \begin{matrix} \}\, n \\ \}\, 2 \end{matrix}. \tag{7}$$

Remarks:

1. *GEVD vs GSVD* Since the original data are the matrix A and the vector b, the generalized singular value decomposition (GSVD) of the pair (C, D) can be used instead of the GEVD of the pair $(C^\top C, D)$. This avoids "squaring" the data and is recommended from a numerical point of view.

2. *Link to low-rank approximation and total least squares* Problem (5) is equivalent to the generalized low-rank approximation problem [GHS87]

$$\text{minimize} \quad \text{over } \widehat{C} \in \mathbb{R}^{2m\times(n+2)} \quad \left\|(C - \widehat{C})D\right\|_{\mathrm{F}} \quad \text{subject to} \quad \operatorname{rank}(\widehat{C}) \le n+1 \quad \text{and} \quad \widehat{C}D^\perp = CD^\perp, \tag{8}$$

[1] Two optimization problems are equivalent if the solution of the first can be obtained from the solution of the second by a one-to-one transformation. Of practical interest are equivalent problems for which the transformation is "simple".

where

$$D^{\perp} = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n+2)\times(n+2)}$$

and $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. Indeed, the constraints of (8) imply that

$$\big\|(C-\widehat{C})D\big\|_{\mathrm{F}} = \|b-\widehat{b}\|, \qquad \text{where} \quad \widehat{b} = Axe^{\mathbf{i}\phi}.$$

The normalization (7) is reminiscent to the generic solution of the total least squares problems [MV07]. The solution of total least squares problems, however, involves a normalization by scaling with the last element of a vector $z_{\min}$ in the approximate kernel of the data matrix C, while the solution of (1) involves normalization by scaling with the norm of the last two elements of the vector $z_{\min}$.

3. *Uniqueness of the solution and minimum norm solutions* Obviously, x is nonunique when A has nontrivial null space. This source of nonuniqueness is fixed in [Byd10] by choosing from the solutions set a least norm solution. A least norm solution of (1), however, may also be nonunique due to possible nonuniquess of ϕ. Consider the following example,

$$A = \begin{bmatrix} 1 & \mathbf{i} \\ -\mathbf{i} & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ -\mathbf{i} \end{bmatrix},$$

which has two least norm solutions

$$\widehat{x}e^{\mathbf{i}\phi_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \text{and} \qquad \widehat{x}'e^{\mathbf{i}\phi_2} = \begin{bmatrix} 0 \\ -\mathbf{i} \end{bmatrix}.$$

Moreover, there is a trivial source of nonuniqueness in x and ϕ due to $xe^{\mathbf{i}\phi} = -xe^{\mathbf{i}(\phi\pm\pi)}$ with both ϕ and one of the angles $\phi \pm \pi$ in the interval $(-\pi, \pi]$.

3 Computational algorithms

Solution (2, 3) gives a straightforward procedure for computing a least norm solution of problem (1) (see function `cls1` in the appendix). The corresponding computational cost is $O(n^2m + n^3)$. Theorem 1 gives two alternative procedures—one based on the GEVD and one based on the GSVD (see functions `cls2` and `cls3` in the appendix). The computational costs are $O\big((n+2)^2m + (n+2)^3\big)$, for `cls2`, and $O\big(m^3 + (n+2)^2m^2 + (n+2)^2m + (n+2)^3\big)$, for `cls3`. Note, however, that `cls2` and `cls3` compute the full GEVD and GSVD, respectively, while only the smallest generalized eigenvalue/eigenvector or singular value/singular vector pair is needed for solving (1). This suggests a way of reducing the computational complexity by a factor of magnitude.

The equivalence between problem (1) and the generalized low-rank approximation problem (8), noted in remark 2 above, allows us to use the algorithm from [GHS87] for solving problem (1). The resulting Algorithm 1 is implemented in the function `cls4` and has computational cost $O\big((n+2)^2m\big)$.

Algorithm 1 Solution of problem (1) using the algorithm from [GHS87].

Input: $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^{m\times 1}$

1: QR factorization of C, $QR = C$.

2: Define $R =: \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}\begin{matrix} \}\, n \\ \}\, 2 \end{matrix}$, where $R_{11} \in \mathbb{R}^{n\times n}$.

3: SVD of R_{22}, $U\Sigma V^{\top} = R_{22}$.

4: Let $\widehat{\phi} := \angle(v_{12} - \mathbf{i}v_{22})$.

5: Let $\widehat{x} := R_{11}^{-1}R_{12}\begin{bmatrix} v_{12} \\ v_{22} \end{bmatrix}$.

Output: $\widehat{x}e^{\mathbf{i}\widehat{\phi}}$

function	method	computational cost
`cls1`	(2, 3)	$O(n^2m + n^3)$
`cls2`	full GEVD	$O\big((n+2)^2m + (n+2)^3\big)$
`cls3`	full GSVD	$O\big(m^3 + (n+2)^2m^2 + (n+2)^2m + (n+2)^3\big)$
`cls4`	Algorithm 1	$O\big((n+2)^2m\big)$

Table 1: Summary of methods for solving the complex least-squares problem (1).

Numerical examples

Generically, the four solution methods implemented in the functions `cls1`, ..., `cls4` compute the same result, which is equal to the unique solution of problem (1). The test script given in the appendix illustrates this fact numerically on examples with random complex data. (In the numerical experiments, the difference between the computed solutions is of the order of magnitude of the machine precision.) Moreover, as predicted by the theoretical computation costs, the method based on Algorithm 1 is the fastest of the four methods when both the number of equations and the number of unknowns is growing, see Figure 1.

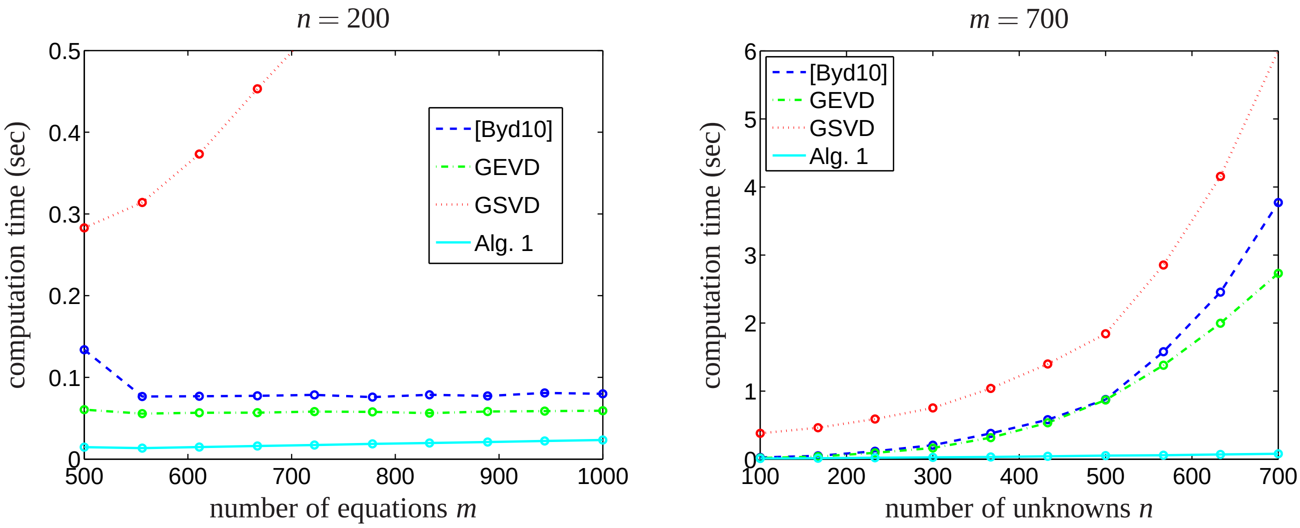

Figure 1: Computation time for the four methods, implemented in the functions `cls1`, ..., `cls4`.

In cases of nonunique solution, the four methods need not compute the same solution. The behavior of the methods based on the GEVD, GSVD, and Algorithm 1 in case of nonunique solution is not analysed in this paper. As noted in remark 3, even a least norm solution may not be unique.

4 Conclusions

A new closed-form solution of the complex least squares problem with constrained phase was derived by reformulating the original optimization problem as an equivalent one that can be solved by a standard GEVD or GSVD. The result shows that the complex least squares problem with constrained phase is a special generalized low-rank approximation problem and makes possible the use of existing robust and efficient methods developed in the literature. In case of nonunique solution, a least norm solution may still be nonunique. Nongeneric problems and efficient methods avoiding the full GEVD and GSVD computation are topics for future research.

References

[Byd10] M. Bydder. Solution of a complex least squares problem with constrained phase. *Linear Algebra Appl.*, 433(11–12):1719–1721, December 2010.

[GHS87] G. Golub, A. Hoffman, and G. Stewart. A generalization of the Eckart–Young–Mirsky matrix approximation theorem. *Linear Algebra Appl.*, 88/89:317–327, 1987.

[MV07] I. Markovsky and S. Van Huffel. Overview of total least squares methods. *Signal Processing*, 87:2283–2302, 2007.

A Matlab code for complex least squares approximation with constrained phase

5a ⟨*Solution by (2, 3)* 5a⟩≡

```
function cx = cls1(A, b)
invM = pinv(real(A' * A)); Atb = A' * b;
phi  = 1 / 2 * angle((Atb).' * invM * Atb);
x    = invM * real(Atb * exp(-i * phi));
cx   = x * exp(i * phi);
```

5b ⟨*Solution by GEVD* 5b⟩≡

```
function cx = cls2(A, b)
⟨Define C, D, and n 5c⟩
[v, l] = eig(C' * C, D); l = diag(l);
l(find(l < 0)) = inf; % ignore nevative values
[ml, mi] = min(l); z = v(:, mi);
phi = angle(-z(end - 1) + i * z(end));
x   = z(1:(end - 2)) / norm(z((end - 1):end));
cx  = x * exp(i * phi);
```

5c ⟨*Define C, D, and n* 5c⟩≡ (5)

```
C = [real(A) real(b) -imag(b);
     imag(A) imag(b)  real(b)];
n = size(A, 2); D = diag([zeros(1, n), 1, 1]);
```

5d ⟨*Solution by GSVD* 5d⟩≡

```
function cx = cls3(A, b)
⟨Define C, D, and n 5c⟩
[u, v] = gsvd(C, D); z = v(:, 1);
phi = angle(-z(end - 1) + i * z(end));
x   = pinv(C(:, 1:n)) * [real(b * exp(- i * phi));
                         imag(b * exp(- i * phi))];
cx  = x * exp(i * phi);
```

5e ⟨*Solution by Algorithm 1* 5e⟩≡

```
function cx = cls4(A, b)
⟨Define C, D, and n 5c⟩
R   = triu(qr(C, 0));
[u, s, v] = svd(R((n + 1):(n + 2), (n + 1):end));
phi = angle(v(1, 2) - i * v(2, 2));
x   = R(1:n, 1:n) \ (R(1:n, (n + 1):end) * [v(1, 2); v(2, 2)]);
cx  = x * exp(i * phi);
```

5f ⟨*Test example* 5f⟩≡

```
% Generate random data
m = 5; n = 2;
A = rand(m, n) + i * rand(m, n);
b = rand(m, 1) + i * rand(m, 1);
% Apply the methods
for i = 1:4,
  eval(sprintf('tic, x = cls%d(A, b); t(%d) = toc;', i, i))
  eval(sprintf('e(%d) = norm(A * x - b); nx(%d) = norm(x);', i, i));
  fprintf('cls%d: ||Ax - b|| = %10.8f, ||x|| = %10.8f, computed in %4.2f sec\n', ...
          i, e(i), nx(i), t(i));
end
```