

## DUAL REGULARIZED TOTAL LEAST SQUARES SOLUTION FROM TWO-PARAMETER TRUST-REGION ALGORITHM

GEUNSEOP LEE

**ABSTRACT.** For the overdetermined linear system, when both the data matrix and the observed data are contaminated by noise, Total Least Squares method is an appropriate approach. Since an ill-conditioned data matrix with noise causes a large perturbation in the solution, some kind of regularization technique is required to filter out such noise. In this paper, we consider a Dual regularized Total Least Squares problem. Unlike the Tikhonov regularization which constrains the size of the solution, a Dual regularized Total Least Squares problem considers two constraints; one constrains the size of the error in the data matrix, the other constrains the size of the error in the observed data. Our method derives two nonlinear equations to construct the iterative method. However, since the Jacobian matrix of two nonlinear equations is not guaranteed to be nonsingular, we adopt a trust-region based iteration method to obtain the solution.

### 1. Introduction

Consider the overdetermined linear system

$$X\mathbf{y} \approx \mathbf{b},$$

with the data matrix  $X \in \mathbf{R}^{m \times n}$ , the solution  $\mathbf{y} \in \mathbf{R}^n$ , and the observed data  $\mathbf{b} \in \mathbf{R}^m$ . If we assume that  $\mathbf{b}$  has noise, but  $X$  is known exactly, we can estimate a solution  $\mathbf{y}$  using the least squares method, which solves

$$(1) \quad \mathbf{y}_{LS} = \arg \min_{\mathbf{y}} \|\mathbf{b} - X\mathbf{y}\|_2.$$

Unfortunately, in many cases, the data matrix  $X$  is ill-conditioned, which causes the output  $\mathbf{y}_{LS}$  very sensitive to noise in  $\mathbf{b}$ . In particular, if the matrix  $X$  is ill conditioned, then the smallest singular value of  $X$  gradually decays to zero. Thus even a small perturbation in the data leads to a large perturbation at the solution  $\mathbf{y}_{LS}$ . To obtain a meaningful and stable solution to (1) despite noise, the solution has to be obtained with *a priori* information. Therefore, we consider a regularization technique that filters out such noise, and smooths

---

Received March 7, 2016; Revised August 6, 2016.

2010 *Mathematics Subject Classification.* Primary 65F10, 65F22.

*Key words and phrases.* dual regularized total least squares method, trust region algorithm.

out the solution. One approach is Tikhonov regularization. Tikhonov regularization picks a linear operator  $L$  so that the solution  $\mathbf{y}$  is constrained to satisfy the condition  $\|L\mathbf{y}\|_2 \leq \delta$ . Here  $L$  is, in general, the identity, a first-order derivative operator, or a second-order derivative operator. The parameter  $\delta$ , which controls the size or smoothness of the solution, comes from knowledge of the underlying physical model. Thus, the least squares solution  $\mathbf{y}_{LS}(\delta)$  with a Tikhonov regularization is obtained from

$$(2) \quad \mathbf{y}_{LS}(\delta) = \arg \min_{\mathbf{y}} \|\mathbf{b} - X\mathbf{y}\|_2, \quad \text{subject to } \|L\mathbf{y}\|_2 \leq \delta.$$

However, in many cases,  $X$  may likely have a perturbation. If error exists in both the right hand side  $\mathbf{b}$  and the data matrix  $X$ , the Total Least Squares (TLS) method is more appropriate than the ordinary least squares model (1).

The TLS approach finds an error matrix  $E_{TLS}$  and an error vector  $\mathbf{r}_{TLS}$  subject to satisfy the following minimization problem

$$(3) \quad \begin{pmatrix} E_{TLS} & \mathbf{r}_{TLS} \end{pmatrix} = \min \begin{pmatrix} E & \mathbf{r} \end{pmatrix} \|_F,$$

such that for some  $\mathbf{y} \in \mathbf{R}^n$ ,

$$(4) \quad (X + E)\mathbf{y} = \mathbf{b} + \mathbf{r}.$$

Similar to (1), the TLS solution  $\mathbf{y}_{TLS}$  is obtained by solving the TLS problem

$$\mathbf{y}_{TLS} = \arg \min_{\mathbf{y}} \begin{pmatrix} E & \mathbf{r} \end{pmatrix} \|_F^2 = \arg \min_{\mathbf{y}} \|\mathbf{b} - (X + E)\mathbf{y}\|_2^2 + \|E\|_F^2.$$

A regularization technique is still required since the solution  $\mathbf{y}_{TLS}$  from (3) and (4) is even more sensitive to noise. The TLS solution  $\mathbf{y}_{TLS}(\delta)$  with Tikhonov regularization is obtained from

$$(5) \quad \mathbf{y}_{TLS}(\delta) = \arg \min_{\mathbf{y}} \|\mathbf{b} - (X + E)\mathbf{y}\|_2^2 + \|E\|_F^2, \quad \text{subject to } \|L\mathbf{y}\|_2 \leq \delta,$$

and its Lagrange multiplier formulation is

$$\mathcal{L}(\mathbf{y}, E, \mathbf{r}, \delta) = \|\mathbf{b} - (X + E)\mathbf{y}\|_2^2 + \|E\|_F^2 + \eta(\|L\mathbf{y}\|_2 - \delta),$$

where  $\eta$  represents a Lagrange multiplier.

Here, Tikhonov regularization demands a reliable boundary  $\delta$ , which is unknown in many applications. Instead, the other kinds of boundaries,  $\gamma$  and  $\phi$ , are possibly given, so that from (4)

$$(6) \quad \|E\|_F \leq \gamma,$$

$$(7) \quad \|\mathbf{r}\|_2 \leq \phi.$$

The TLS solution  $\mathbf{y}_{TLS}(\gamma, \phi)$  under the constraints (6) and (7), which satisfies

$$(X + E)\mathbf{y}_{TLS} = \mathbf{b} + \mathbf{r}$$

as well, can be found by solving

$$(8) \quad \mathbf{y}_{TLS}(\gamma, \phi) = \arg \min_{\mathbf{y}} \|L\mathbf{y}\|_2, \quad \text{subject to } \|E\|_F \leq \gamma, \|\mathbf{r}\|_2 \leq \phi.$$

Lu, Pereverzev, and Tautenhahn [5] named this approach a Dual Regularized Total Least Squares (Dual RTLS) problem and solved the problem with an

iterative method. However, their pioneering work on this topic has limitations since they assumed the special conditions where  $L = I, \gamma = 0$ , or  $\phi = 0$ , which are not realistic for most TLS models. Another work from Lu et al. [4, 6] considered a more general case where  $L \neq I, \gamma \neq 0$ , and  $\phi \neq 0$ . However, the convergence of their new approach is guaranteed only when  $\gamma = 0$  which is theoretically the same condition from [5]. Additionally, for less computational complexity, they approximated their equations by using the model functions [6]. However, their computational efforts are still much more complicated than the other regularized TLS approaches that solve (5). Our contribution is to explore the method when  $L \neq I, \gamma \neq 0$ , and  $\phi \neq 0$ , with a less complicated and straightforward approach. Also, we give convergence results for our method. This paper is organized as follows: in §2 we give a brief summary of the Dual RTLS methods proposed [4, 5, 6], in §3 we derive the two parameter trust region algorithm to solve (8), in §4 we present experimental results, and in §5 we give a conclusion.

### 2. Dual regularized total least squares

In this section, we briefly introduce the Dual RTLS methods proposed by Lu et al. [4, 5, 6]. The Lagrange multiplier formulation of a Dual RTLS problem based on (8) is

$$(9) \quad \mathcal{L}(\mathbf{y}, E, \alpha, \beta) = \|\mathbf{L}\mathbf{y}\|_2 + \alpha(\|\mathbf{r}\|_2^2 - \phi^2) + \beta(\|E\|_F^2 - \gamma^2),$$

where  $\alpha > 0$  and  $\beta > 0$  indicate Lagrange multipliers. Then, the following theorem, proved in [5], characterizes the solution  $\mathbf{y}_{TLS}(\gamma, \phi)$  in (8).

**Theorem 2.1.** *If the constraints (6) and (7) are active, the TLS solution  $\mathbf{y}_{TLS}(\gamma, \phi) = \mathbf{y}(\lambda^*, \mu^*)$  of (8) solves the two-parameter linear system*

$$(10) \quad (X^T X + \lambda^* L^T L - \mu^* I)\mathbf{y}(\lambda^*, \mu^*) = X^T \mathbf{b},$$

where the optimal parameters  $\lambda^*$  and  $\mu^*$  satisfies

$$(11) \quad \lambda^* = \frac{\beta + \alpha\|\mathbf{y}\|_2^2}{\alpha\beta},$$

$$(12) \quad \mu^* = \frac{\alpha\|\mathbf{b} - X\mathbf{y}\|_2^2}{\beta + \alpha\|\mathbf{y}\|_2^2},$$

respectively. Moreover

$$(13) \quad \|\mathbf{b} - X\mathbf{y}(\lambda^*, \mu^*)\|_2 = \phi + \gamma\|\mathbf{y}(\lambda^*, \mu^*)\|_2,$$

$$(14) \quad \mu^* = \frac{\gamma(\phi + \gamma\|\mathbf{y}(\lambda^*, \mu^*)\|_2)}{\|\mathbf{y}(\lambda^*, \mu^*)\|_2}.$$

To obtain the solution  $\mathbf{y}(\lambda^*, \mu^*)$ , Lu et al. assume some conditions including special [5] or more general conditions [4, 6]. First, assume that  $\gamma = 0$ . This case is basically equivalent to the ordinary least squares problem; no error is

found at the data matrix. Thus, we can reduce a Dual RTLS problem defined in (8) to

$$\min \|L\mathbf{y}\|_2 \quad \text{subject to} \quad \|\mathbf{b} - X\mathbf{y}\|_2 \leq \phi.$$

Thus if the constraint  $\|\mathbf{b} - X\mathbf{y}\|_2 \leq \phi$  is active, then the solution  $\mathbf{y}_{TLS}$  is the solution of the linear system

$$(15) \quad (X^T X + \lambda^* L^T L)\mathbf{y}(\lambda^*) = X^T \mathbf{b},$$

with the carefully chosen parameter  $\lambda^*$ . By solving the nonlinear equation

$$f(\lambda) = \|\mathbf{b} - X\mathbf{y}(\lambda)\|_2^2 - \phi^2 = 0,$$

we can choose the optimal parameter  $\lambda^*$ . However, the function  $f(\lambda)$  is convex for a small  $\lambda$ , and is concave for a large  $\lambda$ . Thus the modified function  $h(r)$  is more convenient to compute, which is given by

$$h(r) := f(1/r) = 0.$$

The function  $h(r)$  is monotonically decreasing, and possesses the unique solution  $\lambda^*$ . This leads to use a Newton-like algorithm to find the optimized  $\lambda^*$ . Detailed pseudo-code for the solution  $\mathbf{y}(\lambda^*)$ , when  $\gamma = 0$ , is found at [4].

The second special case is where  $L = I$ . In this case, the form of the linear system (10) becomes equivalent to (15) when two constraints (6) and (7) are active, such that

$$(X^T + (\lambda^* - \mu^*)I)\mathbf{y}(\lambda^*, \mu^*) = X^T \mathbf{b}.$$

Thus the procedure for finding the optimal parameter pair  $(\lambda^*, \mu^*)$  is very similar to the case where  $\gamma = 0$ . Unfortunately, those special cases amount to ordinary regularized least squares problems, which do not expand our menu of options. Thus it is somewhat limited to solve more practical problems.

The last and a more general case where neither  $\gamma = 0$  nor  $L = I$  is introduced in [4]. Due to the conditions  $\gamma \neq 0$  and  $L \neq I$ , the parameters  $\lambda^*$  and  $\mu^*$  are required separately. In a straightforward manner, we can calculate  $\mu^*$  from its definition (12). However, since the computation of  $\lambda^*$  is complicated computationally, a model function estimation is used to approximate the procedure. Assume that the cost function of a Dual RTLS problem is given by

$$(16) \quad J_{\lambda, \mu}(\mathbf{y}) = \|\mathbf{b} - X\mathbf{y}\|_2^2 + \lambda \|L\mathbf{y}\|_2^2 + \mu \|\mathbf{y}\|_2^2.$$

Then, with the solution  $\mathbf{y}(\lambda^*, \mu^*)$ , the partial derivatives of  $J_{\lambda^*, \mu^*}(\mathbf{y})$  respect to  $\lambda^*$  and  $\mu^*$  are given by

$$(17) \quad \begin{aligned} \frac{\partial J_{\lambda^*, \mu^*}(\mathbf{y})}{\partial \lambda^*} &= F'_{\lambda^*} = \|L\mathbf{y}(\lambda^*, \mu^*)\|_2^2, \\ \frac{\partial J_{\lambda^*, \mu^*}(\mathbf{y})}{\partial \mu^*} &= F'_{\mu^*} = \|\mathbf{y}(\lambda^*, \mu^*)\|_2^2. \end{aligned}$$

From (13) and (17), the equation (16) can be reformulated as

$$J_{\lambda^*, \mu^*}(\mathbf{y}) - \lambda^* F'_{\lambda^*} - \mu^* F'_{\mu^*} = (\phi + \gamma \sqrt{F'_{\mu^*}})^2.$$

Here,  $J_{\lambda^*, \mu^*}(\mathbf{y})$  can be approximated from its definition (16) such that

$$(18) \quad J_{\lambda^*, \mu^*}(\mathbf{y}) \approx \|\mathbf{y}(\lambda^*, \mu^*)\|_2^2 - (\mu^* + T)F'_{\mu^*} - \lambda^* F'_{\lambda^*},$$

where  $T$  is a positive constant, which satisfies  $\|X\mathbf{y}\|_2^2 = T\|\mathbf{y}\|_2^2$ . By using a model function  $m(\lambda^*, \mu^*)$  which approximates  $J_{\lambda^*, \mu^*}(\mathbf{y})$ , we can reformulate the equation (18) by substituting  $m(\lambda^*, \mu^*)$  into  $J_{\lambda^*, \mu^*}(\mathbf{y})$ , such that

$$(19) \quad m(\lambda^*, \mu^*) \approx \|\mathbf{y}(\lambda^*, \mu^*)\|_2^2 - (\mu^* + T)m_{\mu^*}(\lambda^*, \mu^*) - \lambda^* m_{\lambda^*}(\lambda^*, \mu^*).$$

Then, the equation (19) is simplified by substituting constants  $C, D$  and  $T$  into complicated dependents, such that

$$m_{\mu^*}(\lambda^*, \mu^*) = \|\mathbf{y}(\lambda^*, \mu^*)\|_2^2 + \frac{C}{\lambda^*} + \frac{D}{T + \mu^*},$$

where  $C$  and  $D$  are given by

$$C = -(\lambda^*)^2 \|L\mathbf{y}(\lambda^*, \mu^*)\|_2^2,$$

$$D = -\frac{\|\mathbf{y}(\lambda^*, \mu^*)\|_2^2 - J_{\lambda^*, \mu^*}(\mathbf{y}) - \lambda^* \|L\mathbf{y}(\lambda^*, \mu^*)\|_2^2}{\|\mathbf{y}(\lambda^*, \mu^*)\|_2^2}.$$

Therefore, a parameter  $\lambda$  is updated from computing constants  $C, D, T$ , and a parameter  $\mu$  in (17). It continues until the optimal parameter pair  $(\lambda, \mu)$  close to  $(\lambda^*, \mu^*)$  converges. As stated earlier, however, if  $\gamma \neq 0$ , then the algorithm possibly fails to converge [4].

### 3. Derivation of the algorithm

As stated in §2, the solution of (9) is equivalent to the solution of the linear system (10) with the carefully chosen two parameters  $(\lambda^*, \mu^*)$ . It is further extended that the solution  $\mathbf{y}(\lambda^*, \mu^*)$  is equivalent to the eigenvector of the following augmented linear system (20) correspond to the smallest eigenvalue  $\mu^*$ .

**Theorem 3.1.** *The solution  $\mathbf{y}_{TLS}(\gamma, \phi)$  of (8) solves the parameterized eigen-pair problem*

$$(20) \quad A(\lambda^*, \mu^*) \begin{pmatrix} \mathbf{y}(\lambda^*, \mu^*) \\ -1 \end{pmatrix} = 0$$

for the optimal parameter pair  $(\lambda^*, \mu^*)$ , where the solution-dependent matrix  $A(\lambda, \mu)$  is given by

$$(21) \quad A(\lambda, \mu) = \begin{pmatrix} X^T X + \lambda L^T L - \mu I & X^T \mathbf{b} \\ \mathbf{b}^T X & \mathbf{b}^T \mathbf{b} - \lambda \|L\mathbf{y}(\lambda, \mu)\|_2^2 - \mu \end{pmatrix}.$$

Conversely, the smallest eigenpair of  $A(\lambda^*, \mu^*)$  is  $(\mu^*, \begin{pmatrix} \mathbf{y}(\lambda^*, \mu^*) \\ -1 \end{pmatrix})$ .

*Proof.* We simply follow the procedure of proof by [8]. Note that  $\mathbf{y}$  is used instead of  $\mathbf{y}(\lambda, \mu)$  for convenience. Obviously, the first block of (21) comes from (10). From the definition of  $\mu^*$  in (12), we get

$$\mu^*(\beta + \alpha\|\mathbf{y}\|_2^2) = \alpha(\|X\mathbf{y}\|_2^2 - \mathbf{y}^T X^T \mathbf{b} - \mathbf{b}^T X\mathbf{y} + \mathbf{b}^T \mathbf{b}).$$

If we reformulate (10) such that

$$\|X\mathbf{y}\|_2^2 + \lambda^*\|L\mathbf{y}\|_2^2 - \mu^*\|\mathbf{y}\|_2^2 = \mathbf{y}^T X^T \mathbf{b},$$

and gather two equations, then

$$\mu^*(\beta + \alpha\|\mathbf{y}\|_2^2) = \alpha(\|X\mathbf{y}\|_2^2 - \|X\mathbf{y}\|_2^2 - \lambda^*\|L\mathbf{y}\|_2^2 + \mu^*\|\mathbf{y}\|_2^2 - \mathbf{b}^T X\mathbf{y} + \mathbf{b}^T \mathbf{b}),$$

$$\mu^*\beta = \alpha(\mathbf{b}^T \mathbf{b} - \lambda^*\|L\mathbf{y}\|_2^2 - \mathbf{b}^T X\mathbf{y}),$$

$$\mu^* = \frac{\alpha}{\beta} \left( \begin{array}{cc} \mathbf{b}^T X & \mathbf{b}^T \mathbf{b} - \lambda^*\|L\mathbf{y}\|_2^2 \end{array} \right) \left( \begin{array}{c} \mathbf{y} \\ -1 \end{array} \right).$$

Therefore,

$$(22) \quad \left( \begin{array}{cc} 1 & 0 \\ 0 & \frac{\alpha}{\beta} \end{array} \right) \left( \begin{array}{cc} M(\lambda^*, \mu^*) & X^T \mathbf{b} \\ \mathbf{b}^T X & \mathbf{b}^T \mathbf{b} - \lambda^*\|L\mathbf{y}\|_2^2 - \mu^* \end{array} \right) \left( \begin{array}{c} \mathbf{y} \\ -1 \end{array} \right) = 0,$$

where  $M(\lambda, \mu) = X^T X + \lambda L^T L - \mu I$ . Since the first matrix of (22) is nonsingular because  $\alpha \neq 0$  and  $\beta \neq 0$ , we see that (20) holds. For the proof of the opposite direction, the procedure is similar to that of [8].  $\square$

**Theorem 3.2.** *Assume that the null space  $\mathcal{N}(X)$  of  $X \in \mathbf{R}^{m \times n}$  and the null space  $\mathcal{N}(L)$  of  $L \in \mathbf{R}^{p \times n}$  satisfy  $\mathcal{N}(X) \cap \mathcal{N}(L) = \mathbf{0}$ , where  $m \geq n$  and  $p \leq n$ . Assume also that the smallest eigenvalue  $\tau$  of  $M(\lambda, 0)$  has the eigenspace spanned by the left orthogonal matrix  $V(\lambda) \in \mathbf{R}^{n \times k}$ . If*

$$V(\lambda)^T X^T \mathbf{b} \neq 0$$

*then smallest eigenvalue  $\mu$  of  $A(\lambda, 0)$  in (22) satisfies  $\mu < \tau$  and  $M(\lambda, \mu)$  is positive definite.*

*Proof.* See [3].  $\square$

According to Theorem 3.2,  $M(\lambda, \mu)$  is positive definite when  $(\lambda, \mu)$  comes close to  $(\lambda^*, \mu^*)$ . Then  $A(\lambda, \mu)$  is factored into

$$(23) \quad A(\lambda, \mu) = \left( \begin{array}{cc} I & 0 \\ \mathbf{b}^T X M(\lambda, \mu)^{-1} & 1 \end{array} \right) \left( \begin{array}{cc} M(\lambda, \mu) & X^T \mathbf{b} \\ 0 & \rho_1(\lambda, \mu) \end{array} \right),$$

where  $\rho_1(\lambda, \mu) = \mathbf{b}^T \mathbf{b} - \lambda\|L\mathbf{y}(\lambda, \mu)\|_2^2 - \mu - \mathbf{b}^T X\mathbf{y}(\lambda, \mu)$ . However, the solution  $\mathbf{y}(\lambda, \mu)$  in (22) is not trivial only if  $A(\lambda, \mu)$  is singular. Since the first matrix in (23) is nonsingular, the second matrix has to be singular. Thus, we can clearly construct one nonlinear equation in order to have the solution in (22), such that

$$(24) \quad \rho_1(\lambda, \mu) = \mathbf{b}^T \mathbf{b} - \lambda\|L\mathbf{y}(\lambda, \mu)\|_2^2 - \mu - \mathbf{b}^T X\mathbf{y}(\lambda, \mu) = 0.$$

Also, we can get another equation from (13) such that

$$(25) \quad \rho_2(\lambda, \mu) = \|\mathbf{b} - X\mathbf{y}(\lambda, \mu)\|_2 - \phi - \gamma\|\mathbf{y}(\lambda, \mu)\|_2 = 0.$$

Therefore, by assuming that  $\mathbf{y}(\lambda, \mu)$  is differentiable, the partial derivatives of the function pair  $\rho(\lambda, \mu) = \begin{pmatrix} \rho_1(\lambda, \mu) \\ \rho_2(\lambda, \mu) \end{pmatrix}$  respect to  $\lambda$  and  $\mu$  are

$$(26) \quad \frac{\partial \rho_1}{\partial \lambda} = -2\lambda\mathbf{y}^T(\lambda, \mu)L^T L\mathbf{y}_\lambda,$$

$$(27) \quad \frac{\partial \rho_1}{\partial \mu} = -1 + 2\lambda\mathbf{y}^T(\lambda, \mu)\mathbf{y}_\lambda - \|\mathbf{y}(\lambda, \mu)\|_2^2,$$

$$(28) \quad \frac{\partial \rho_2}{\partial \lambda} = \frac{\lambda\mathbf{y}^T(\lambda, \mu)L^T L\mathbf{y}_\lambda - \mu\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{b} - X\mathbf{y}(\lambda, \mu)\|_2} - \gamma\frac{\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{y}(\lambda, \mu)\|_2},$$

$$(29) \quad \frac{\partial \rho_2}{\partial \mu} = \frac{\lambda\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu) - \mu\mathbf{y}_\mu^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{b} - X\mathbf{y}(\lambda, \mu)\|_2} - \gamma\frac{\mathbf{y}_\mu^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{y}(\lambda, \mu)\|_2},$$

where  $\mathbf{y}_\lambda = \frac{\partial \mathbf{y}(\lambda, \mu)}{\partial \lambda}$  and  $\mathbf{y}_\mu = \frac{\partial \mathbf{y}(\lambda, \mu)}{\partial \mu}$  are found from solving the linear systems

$$(30) \quad M(\lambda, \mu)\mathbf{y}_\lambda = -L^T L\mathbf{y}(\lambda, \mu),$$

$$(31) \quad M(\lambda, \mu)\mathbf{y}_\mu = \mathbf{y}(\lambda, \mu),$$

respectively.

From those partial derivatives, we have a Jacobian matrix

$$(32) \quad J_\rho(\lambda, \mu) = \begin{pmatrix} -2\lambda\mathbf{y}^T(\lambda, \mu)L^T L\mathbf{y}_\lambda & -1 + 2\lambda\mathbf{y}^T(\lambda, \mu)\mathbf{y}_\lambda - \|\mathbf{y}(\lambda, \mu)\|_2^2 \\ \frac{\lambda\mathbf{y}^T(\lambda, \mu)L^T L\mathbf{y}_\lambda - \mu\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{b} - X\mathbf{y}(\lambda, \mu)\|_2} - \gamma\frac{\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{y}(\lambda, \mu)\|_2} & \frac{\lambda\mathbf{y}_\lambda^T\mathbf{y}(\lambda, \mu) - \mu\mathbf{y}_\mu^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{b} - X\mathbf{y}(\lambda, \mu)\|_2} - \gamma\frac{\mathbf{y}_\mu^T\mathbf{y}(\lambda, \mu)}{\|\mathbf{y}(\lambda, \mu)\|_2} \end{pmatrix}$$

based on the equations (26), (27), (28), and (29).

However, a Jacobian matrix  $J_\rho(\lambda, \mu)$  in (32) is not necessarily nonsingular. Therefore, a Newton-like iterative method may or may not converge to a unique solution. Thus we develop the trust-region based method that allows  $J_\rho(\lambda, \mu)$  to be singular.

Unlike line search methods such as a Newton method or a steepest descent method, a trust-region method defines a region at the current iteration, which makes a model of an object function to be *trusted*. It then computes the step to make minimize a model in a region [7].

Assume that the quadratic model function  $m_k(\mathbf{p}^k)$  for the trust-region sub-problem is defined as

$$(33) \quad \begin{aligned} \min_{\mathbf{p}^k \in \mathbf{R}^2} m_k(\mathbf{p}^k) &= \min_{\mathbf{p}^k \in \mathbf{R}^2} \|J_\rho(\lambda^k, \mu^k)\mathbf{p}^k + \rho(\lambda^k, \mu^k)\|_2^2, \\ &\text{subject to } \|\mathbf{p}^k\|_2 \leq \Delta^k, \end{aligned}$$

where  $\Delta^k$  represents the trust-region radius, and  $\mathbf{p}^k = \begin{pmatrix} \delta\lambda^k \\ \delta\mu^k \end{pmatrix}$  indicates the trust-region step. If  $J_\rho(\lambda^k, \mu^k)$  is singular, the solution  $\mathbf{p}^k$  of  $m_k(\mathbf{p}^k)$  is equivalent to the solution of the least squares problem

$$(34) \quad \min_{\mathbf{p}^k(w) \in \mathbf{R}^2} \left\| \begin{pmatrix} J_\rho(\lambda^k, \mu^k) \\ \sqrt{w}I \end{pmatrix} \mathbf{p}^k(w) + \begin{pmatrix} \rho(\lambda^k, \mu^k) \\ 0 \end{pmatrix} \right\|_2^2$$

with the carefully chosen regularization parameter  $w \geq 0$ . Thus the solution  $\mathbf{p}^k(w)$  is obtained by solving the normal equation

$$(J_\rho(\lambda^k, \mu^k)^T J_\rho(\lambda^k, \mu^k) + wI) \mathbf{p}^k(w) = -J_\rho(\lambda^k, \mu^k)^T \rho(\lambda^k, \mu^k).$$

To seek the sufficiently large scalar  $w$  which makes  $(J_\rho(\lambda^k, \mu^k)^T J_\rho(\lambda^k, \mu^k) + wI)$  positive definite under the constraint  $\|\mathbf{p}^k\|_2 \leq \Delta^k$ , we adopt a Newton method. One nonlinear function is constructed from the fact that the constraint (33) must hold, which is given by

$$(35) \quad g(w) = (\Delta^k)^2 - \|\mathbf{p}^k(w)\|_2^2.$$

We know that the function  $g(w)$  is monotone, and its derivative respect to  $w$  is nonzero, thus  $g(w)$  has a unique solution  $w^*$ . A derivative  $g_w(w)$  of the function  $g(w)$  respect to  $w$  is found by solving

$$(36) \quad g_w(w) = -2\mathbf{p}^k(w)^T \mathbf{p}_w^k(w),$$

where  $\mathbf{p}_w^k(w) = \frac{\partial \mathbf{p}^k(w)}{\partial w}$  is obtained from the linear system

$$(J_\rho(\lambda^k, \mu^k)^T J_\rho(\lambda^k, \mu^k) + wI) \mathbf{p}_w^k(w) = -\mathbf{p}^k(w).$$

Specifically, if we have a QR decomposition such that

$$\begin{pmatrix} J_\rho(\lambda^k, \mu^k) \\ \sqrt{w}I \end{pmatrix} = Q(w)R(w),$$

then  $\mathbf{p}_w^k(w) = -R(w)^{-T} R(w)^{-1} \mathbf{p}^k(w)$ . Thus  $g_w(w)$  in (36) becomes

$$g_w(w) = 2\|R(w)^{-1} \mathbf{p}^k(w)\|_2^2,$$

and the parameter  $w$  in the trust-region subproblem is updated by the Newton step

$$w^{i+1} = w^i - g(w^i)/g_w(w^i).$$

It continues until the optimal parameter  $w_i$  close to  $w^*$  converges.

If the Jacobian matrix  $J_\rho(\lambda^k, \mu^k)$  is nonsingular, the trust-region subproblem becomes quite simple. In this case, we set  $w = 0$ , and  $\mathbf{p}^k$  is computed by solving  $\mathbf{p}^k = -J_\rho(\lambda^k, \mu^k)^{-1} \rho(\lambda^k, \mu^k)$ , which is referred to as the *full step*. Pseudo-code for the trust-region based Dual RTLS algorithm is summarized at Algorithm 1, which comes from small modification of the algorithm [7, p. 87]. Also, pseudo-code for the solution of the trust-region subproblem is given at Algorithm 2 for the case where  $J_\rho(\lambda^k, \mu^k)$  is singular. Algorithm 2 is primarily the Newton-based 2-by-2 iterative method, and thus requires small computational expense. Even for the practical implementation, two or three iterations are sufficient



for a fairly loose approximation to  $\mathbf{p}^k(w_i)$  rather than highly accurate solution according to [7]. Note that  $\rho_k$  is the ratio between actual reduction and the predicted reduction, and is defined as

$$(37) \quad \rho_k = \frac{\|\mathbf{p}^k\|_2}{m_k(0) - m_k(\mathbf{p}^k)},$$

and  $\hat{\Delta}$  is an overall bound on the step lengths.

Since Algorithm 1 is from small modification of the trust region algorithm introduced in [7, pp. 261–262], it follows that the global convergence of Algorithm 1 is also obtained.

#### 4. Experimental results

We now present the numerical experiments that compare the performance of Algorithm 1 and the model function based Dual RTLS algorithm [4], which applies to a general case where  $L \neq I, \gamma \neq 0$ , and  $\phi \neq 0$ . Specifically, the discretization of a Fredholm integral equation of the first kind is used for the evaluations. The experiments were run on a Intel Core i5 computer with 3.10 GHz and 8GB memory under MATLAB version 7.10.0.499.

The discretization of a Fredholm integral equation problem is defined as, in general,

$$(38) \quad \int_b^a x(s, t)y(t)dt = b(s), \quad s \in [a, b],$$

where  $x(s, t)$  is the continuous kernel function,  $y(t)$  is the known solution that we seek, and  $b(s)$  is the observed data, respectively [1, p. 875]. An integral equation is used to describe many physical phenomenas. In other words, it is often a more convenient and powerful formulation for a model of real world problem with their compact and completely self-contained form [1]. Also, an integral equation enables us to use a particular boundary condition. We call an equation (38) a Fredholm integral equation when the limit of the integral  $[a, b]$  is fixed. For the discretization of an integral equation, a set of linear equations is obtained from (38) by using quadrature rule, which is given by

$$(39) \quad \sum_{j=1}^n w_j X(s_i, t_j)y(t_j) = b(s_i), \quad i = 1, \dots, m,$$

where  $w_j$  represents a weight. The equation (39) can be expressed as a matrix form

$$\begin{pmatrix} w_1 X(s_1, t_1) & \dots & w_n X(s_1, t_n) \\ w_1 X(s_2, t_1) & \dots & w_n X(s_2, t_n) \\ \vdots & & \vdots \\ w_1 X(s_m, t_1) & \dots & w_n X(s_m, t_n) \end{pmatrix} \begin{pmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_n) \end{pmatrix} = \begin{pmatrix} b(s_1) \\ b(s_2) \\ \vdots \\ b(s_m) \end{pmatrix},$$

---

**Algorithm 1** Generalized Dual RTLS algorithm
 

---

**Input**  $X, L, \mathbf{b}, tol, \hat{\Delta}, \eta$

1. Choose initial guesses  $\lambda^0, \mu^0$  and  $k = 0$ .

2. Solve  $M(\lambda^0, \mu^0)\mathbf{y}^0 = X^T\mathbf{b}$ .

3. **While**  $\epsilon \geq tol$

Solve  $M(\lambda^k, \mu^k)\mathbf{y}^k = X^T\mathbf{b}$

Solve  $M(\lambda^k, \mu^k)\mathbf{y}_\lambda^k = -L^T L\mathbf{y}^k$

Solve  $M(\lambda^k, \mu^k)\mathbf{y}_\mu^k = \mathbf{y}^k$

Set  $\rho(\lambda^k, \mu^k) = [\rho_1(\lambda^k, \mu^k), \rho_2(\lambda^k, \mu^k)]^T$

Compute the Jacobian matrix  $J_\rho(\lambda^k, \mu^k) = \begin{pmatrix} \frac{\partial \rho_1}{\partial \lambda} & \frac{\partial \rho_1}{\partial \mu} \\ \frac{\partial \rho_2}{\partial \lambda} & \frac{\partial \rho_2}{\partial \mu} \end{pmatrix}$

**If** the rank of  $J_\rho(\lambda^k, \mu^k) < 2$

  Compute  $\mathbf{p}^k$  from Algorithm 2 with input parameters

$J_\rho(\lambda^k, \mu^k), \Delta^k, tol$ .

**else**

  Solve  $J_\rho(\lambda^k, \mu^k)\mathbf{p}^k = -\rho(\lambda^k, \mu^k)$

Evaluate  $\rho_k$  from (37).

**If**  $\rho_k < \frac{1}{4}$

$\Delta^{k+1} = \frac{1}{4}\Delta^k$

**else**

**If**  $\rho_k > \frac{3}{4}$  and  $\left\| \begin{pmatrix} \delta\lambda^k \\ \delta\mu^k \end{pmatrix} \right\|_2 = \delta_k$

$\Delta^{k+1} = \min(2\Delta^k, \hat{\Delta})$

**else**

$\Delta^{k+1} = \Delta^k$

**If**  $\rho_k > \eta$

$\begin{pmatrix} \lambda^{k+1} \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda^k \\ \mu^k \end{pmatrix} + \mathbf{p}^k,$

**else**

$\begin{pmatrix} \lambda^{k+1} \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} \lambda^k \\ \mu^k \end{pmatrix}$

$k = k + 1$

$\epsilon = \|\mathbf{p}^k\|_2$

4. return  $\mathbf{y}^k$

---

or simply

$$(40) \quad X\mathbf{y} = \mathbf{b}.$$

There are some possible perturbations to both side of (40) due to an approximation error during the discretization or a model error. Moreover, in many cases, the matrix  $X$  is very ill-conditioned, thus the Dual RTLS method is a good approach to obtain the solution  $\mathbf{y}$ .

To simulate the discretization of a Fredholm integral equation problems, we generate square matrices using the 'SHAW', 'PHILLIPS', 'HEAT(1)', and

**Algorithm 2** Trust-Region Subproblem**Input**  $J_\rho, \Delta^k, tol$ 1. Choose initial guesses  $w^1$  and  $i = 1$ .2. **While**  $\epsilon \geq tol$ Find  $\left( \frac{J_\rho}{\sqrt{w^i} I} \right) = Q(w^i)R(w^i)$ Solve  $R(w^i)^T R(w^i) \mathbf{p}^k(w^i) = -J_\phi^T \rho^k$ Set  $g(w^i) = (\Delta^k)^2 - \|\mathbf{p}^k(w^i)\|_2$ Solve  $\mathbf{p}_w^k(w^i) = -R(w^i)^{-T} R(w^i)^{-1} \mathbf{p}^k(w^i)$ Set  $g_w(w^i) = 2\|R(w^i)^{-1} \mathbf{p}^k(w^i)\|_2^2$  $w^{i+1} = w^i - g(w^i)/g_w(w^i)$  $\epsilon = g(w^i)/g_w(w^i)$  $i = i + 1$ 3. return  $\mathbf{p}^k(w^i)$ 

‘DERIV2(1)’ functions from Hansen’s regularization tools [2] that generate ill-posed data matrices  $X$  in (40) with various sizes. Then we generate noise-contaminated matrices  $\hat{X}$  from  $X$ . The noise vector  $\mathbf{r}$  added to  $\mathbf{b}$  is generated such that

$$\mathbf{r} = \sigma \max(\mathbf{b}) \bar{\mathbf{r}} / \|\bar{\mathbf{r}}\|_2,$$

where  $\bar{\mathbf{r}}$  is generated from Matlab ‘randn’ function, and  $\max(\mathbf{b})$  is the largest value in  $\mathbf{b}$ .

We take initial parameter  $\lambda^0 = 1.0e - 1$  for ‘SHAW’ and ‘PHILLIPS’ functions, and  $\lambda^0 = 1.0e - 3$  for ‘HEAT(1)’ and ‘DERIV2(1)’ functions. We also set

$$(41) \quad \mu^0 = \frac{\gamma(\phi + \gamma\|\mathbf{b}\|_2)}{\|\mathbf{b}\|_2}$$

for Algorithm 1. The boundaries  $\gamma$  and  $\phi$  are set to be  $\gamma = \tau\|X - \hat{X}\|_F$  and  $\phi = \tau\|\mathbf{b} - \hat{X}\mathbf{y}_{true}\|_2$  respectively, where the scale factor  $\tau = 1.0$ . The approximation of the first order derivative operator is chosen for the regularization matrix  $L$ . All algorithms are terminated if the stopping criterion  $\epsilon$ , which is defined as

$$\epsilon = \sqrt{(\delta\lambda)^2 + (\delta\mu)^2},$$

satisfies  $\epsilon \leq 1.0e - 8$ . Additionally, we set  $\hat{\Delta} = 1.0$  and  $\eta = 0$  for Algorithm 1, and  $tol = 1.0e - 3$  and  $w^1 = 1.0e - 2$  for Algorithm 2.

In our experiment, first we compare the relative errors and convergence steps with a fixed noise level  $\sigma = 0.05$ . Tables 1, 2, 3, and 4 show the results of the experiments. Note that the results are the average of 100 times experiments. As can be seen, relative errors of both algorithms have no significant differences except ‘PHILLIPS’ function. However, the experiments reveal that the Dual

TABLE 1. Comparison of Algorithm 1 and the Dual RTLS algorithm with SHAW function, where noise level  $\sigma = 0.05$ , and  $\lambda^0 = 0.1$ .

Data matrix size		Algorithm 1	Dual RTLS
400× 400	CPU time	0.0264	0.2205
	iteration	4.93	89.33
	relative error	0.1207	0.1202
800× 800	CPU time	0.2028	1.7557
	iteration	6.69	150.97
	relative error	0.0797	0.0794
1200× 1200	CPU time	0.4581	6.0323
	iteration	6.00	206.16
	relative error	0.0647	0.0643

TABLE 2. Comparison of Algorithm 1 and the Dual RTLS algorithm with PHILLIPS function, where noise level  $\sigma = 0.05$ , and  $\lambda^0 = 0.1$ .

Data matrix size		Algorithm 1	Dual RTLS
400× 400	CPU time	0.0365	2.0512
	iteration	7.04	344.22
	relative error	0.0231	0.3936
800× 800	CPU time	0.1795	8.4950
	iteration	6.64	356.12
	relative error	0.0206	2.6654
1200× 1200	CPU time	0.5101	21.3426
	iteration	6.76	344.50
	relative error	0.0170	1.0682

RTLS algorithm fails to converge within a predefined maximum iteration number (400) when we use ‘DERIV2(1)’ function. The outputs from ‘PHILLIPS’ function also shows that the Dual RTLS algorithm produces noise-dominated outputs when it fails to converge within a maximum iteration number. Contrarily, Algorithm 1 produces more reliable and robust outputs with faster execution time and less iteration numbers compared to the Dual RTLS algorithm.

Second, we compare the relative errors and convergence steps with variant noise level ranged from 0.001 to 0.1. The kernel function and the ground-truth solution are generated from ‘SHAW’ function, where  $X \in \mathbf{R}^{800 \times 800}$ . All initial parameters are same with the earlier experiment. According to Table 5, an interesting result is found that with lower noise levels, the Dual RTLS algorithm fails to converge to the solution within a maximum iteration number, whereas

TABLE 3. Comparison of Algorithm 1 and the Dual RTLS algorithm with HEAT(1) function, where noise level  $\sigma = 0.05$ , and  $\lambda^0 = 0.001$ .

Data matrix size		Algorithm 1	Dual RTLS
400× 400	CPU time	0.0179	0.3100
	iteration	3.68	132.70
	relative error	0.0883	0.0874
800× 800	CPU time	0.1093	2.4105
	iteration	3.64	220.08
	relative error	0.0667	0.0661
1200× 1200	CPU time	0.3081	7.7222
	iteration	3.90	282.26
	relative error	0.0565	0.0559

TABLE 4. Comparison of Algorithm 1 and the Dual RTLS algorithm with DERIV2(1) function, where noise level  $\sigma = 0.05$ , and  $\lambda^0 = 0.001$ .

Data matrix size		Algorithm 1	Dual RTLS
400× 400	CPU time	0.0189	0.9060
	iteration	4.02	400
	relative error	0.0208	0.0208
800× 800	CPU time	0.1133	4.3240
	iteration	4.00	400
	relative error	0.0153	0.0154
1200× 1200	CPU time	0.3087	10.9272
	iteration	4.00	400
	relative error	0.0128	0.0129

Algorithm 1 robustly ensure convergence to the solution. When  $\sigma = 0.1$ , the Dual RTLS fails to produce a meaningful solution.

## 5. Conclusion

We introduced the circumstances if the different kinds of prior informations of the regularization are given, which is referred to as a dual regularized Total Least Squares problem. Specifically, unlike Tikhonov regularization, a dual regularized Total Least Squares problem defines two constraints; the one is the constraint for the size of the data matrix error, and the other is the constraint for the size of the observed data error. In this paper, we derived two non-linear equations from a dual regularized Total Least Squares problem. However, since the Jacobian matrix of two non-linear equations is not guaranteed to be nonsingular, we suggested the trust-region based iteration algorithm.

TABLE 5. Comparison of Algorithm 1 and the Dual RTLS algorithm with SHAW data, where  $X \in \mathbf{R}^{800 \times 800}$ .

Noise Level		Algorithm 1	Dual RTLS
0.001	CPU time	0.3092	4.4719
	iteration	6	400
	relative error	0.0322	0.0362
0.005	CPU time	0.1425	4.4148
	iteration	5	400
	relative error	0.0410	0.0410
0.01	CPU time	0.1024	4.2838
	iteration	3.54	392.10
	relative error	0.0443	0.0442
0.05	CPU time	0.1946	1.6449
	iteration	6.52	151.08
	relative error	0.0806	0.0789
0.1	CPU time	0.5181	1.3686
	iteration	9.06	63.70
	relative error	0.1759	1.6005

Our experimental results showed that our algorithm is convergent with fewer iteration numbers.

### References

- [1] G. Arfken, *Mathematical Methods for Physicists*, Third edition, 1985, Academic Press, Orlando, FL, USA.
- [2] P. C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numerical Algorithms **6** (1998), 1–35.
- [3] G. Lee, H. Fu, and J. L. Barlow, *Fast high-resolution image reconstruction using Tikhonov regularization based total least squares*, SIAM J. Sci. Comput. **35** (2013), no. 1, B275–B290.
- [4] S. Lu, S. V. Pereverzev, and U. Tautenhahn, *Dual regularized total least squares and multi-parameter regularization*, Comput. Methods Appl. Math. **8** (2008), no. 3, 253–262.
- [5] ———, *Regularized total least squares: computational aspects and error bounds*, SIAM J. Matrix Anal. Appl. **31** (2009), Issue 3.
- [6] ———, *A model function method in regularized total least squares*, Appl. Anal. **89** (2010), no. 11, 1693–1703.
- [7] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, Berlin, 2000.
- [8] R. Renaut and H. Guo, *Efficient algorithms for solution of regularized total least squares*, SIAM J. Matrix Anal. Appl. **26** (2004), no. 2, 457–476.

GEUNSEOP LEE  
 TEST AND PACKAGE CENTER  
 SAMSUNG ELECTRONICS CO. LTD  
 ASAN 31487, KOREA  
*E-mail address:* geunseoplee.la@gmail.com