Level choice in truncated total least squares

Diana Sima Sabine Van Huffel

Department of Electrical Engineering
Katholieke Universiteit Leuven, Belgium

TLS workshop, Leuven, August 2006

Outline

Total Least Squares

Noisy linear system $Ax \approx b$

- A is a given $m \times n$ matrix ($m \geq n$)
- b is an m-dimensional given vector

Total Least Squares finds the *nearest compatible system*

$$\text{TLS:} \quad \min_{\Delta A, \Delta b, x} \left\| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \right\|_F^2 \quad \text{s.t.} \quad (A + \Delta A)x = b + \Delta b$$

Solution method: Rank reduction of $\begin{bmatrix} A & b \end{bmatrix}$ by one.

TLS is classically solved using the SVD of $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$.
$\rightsquigarrow$ the right singular vector in V corresponding to the smallest singular value gives the TLS solution $x_{\text{TLS}} := -v_{1:n,n+1} / v_{n+1,n+1}$.

Total Least Squares

Noisy linear system $Ax \approx b$

- A is a given $m \times n$ matrix ($m \geq n$)
- b is an m-dimensional given vector

Total Least Squares finds the *nearest compatible system*

$$\text{TLS}: \quad \min_{\Delta A, \Delta b, x} \| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \|_F^2 \quad \text{s.t.} \quad (A + \Delta A)x = b + \Delta b$$

Solution method: Rank reduction of $\begin{bmatrix} A & b \end{bmatrix}$ by one.

TLS is classically solved using the SVD of $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$.
$\rightsquigarrow$ the right singular vector in V corresponding to the smallest singular value gives the TLS solution $x_{\text{TLS}} := -v_{1:n,n+1} / v_{n+1,n+1}$.

Total Least Squares

Noisy linear system $Ax \approx b$

- A is a given $m \times n$ matrix ($m \geq n$)
- b is an m-dimensional given vector

Total Least Squares finds the *nearest compatible system*

TLS: $$\min_{\Delta A, \Delta b, x} \left\| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \right\|_F^2 \quad \text{s.t.} \quad (A + \Delta A)x = b + \Delta b$$

Solution method: Rank reduction of $\begin{bmatrix} A & b \end{bmatrix}$ by one.

TLS is classically solved using the SVD of $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$.
$\rightsquigarrow$ the right singular vector in V corresponding to the smallest singular value gives the TLS solution $x_{\text{TLS}} := -v_{1:n,n+1} / v_{n+1,n+1}$.

Total Least Squares

Possible problems

- non-uniqueness: non-unique smallest singular value
- multicollinearities: linearly dependent columns in A
- non-genericity: non-existence of the solution x (*e.g.*, when b is orthogonal to the left singular subspace corresp. to smallest singular value of A)

Modifying the TLS method

It is possible to identify each problematic situation (by inspecting the SVDs of A and $\begin{bmatrix} A & b \end{bmatrix}$) and to add extra constraints such that a unique minimum norm TLS solution x_{TLS} is found.

Total Least Squares

Possible problems

- non-uniqueness: non-unique smallest singular value
- multicollinearities: linearly dependent columns in A
- non-genericity: non-existence of the solution x (*e.g.*, when b is orthogonal to the left singular subspace corresp. to smallest singular value of A)

Modifying the TLS method

It is possible to identify each problematic situation (by inspecting the SVDs of A and $\begin{bmatrix} A & b \end{bmatrix}$) and to add extra constraints such that a unique minimum norm TLS solution x_{TLS} is found.

Ill-posed linear algebraic systems.

When $Ax \approx b$ originates from an ill-posed problem

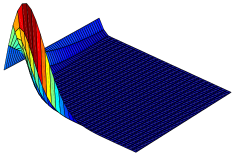
- $\begin{bmatrix} A & b \end{bmatrix}$ is ill-conditioned
- there is no clear gap between singular values of $\begin{bmatrix} A & b \end{bmatrix}$
- singular vectors corresponding to decreasing singular values contain increasing number of sign changes
- b can be almost orthogonal onto singular subspaces of A

Example: TLS solution of an ill-posed problem

Example

$A_0 x_0 \approx b_0$ – a slightly incompatible ill-posed system

- A_0 – a smooth integral kernel
- x_0 – discretized smooth function
- Singular values of the data matrix $\begin{bmatrix} A_0 & b_0 \end{bmatrix} = U_0 \Sigma_0 V_0^\top$
- $U_0^\top b_0 \rightsquigarrow$ close-to-nongenericity

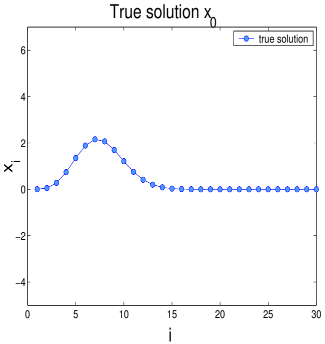


Example: TLS solution of an ill-posed problem

Example

$A_0 x_0 \approx b_0$ – a slightly incompatible ill-posed system

- A_0 – a smooth integral kernel
- x_0 – discretized smooth function
- Singular values of the data matrix $\begin{bmatrix} A_0 & b_0 \end{bmatrix} = U_0 \Sigma_0 V_0^\top$
- $U_0^\top b_0 \rightsquigarrow$ close-to-nongenericity

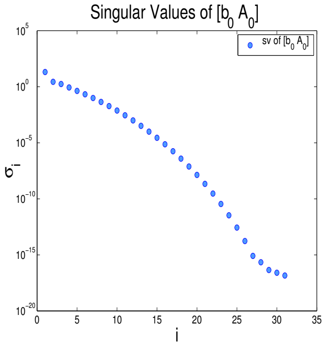


Example: TLS solution of an ill-posed problem

Example

$A_0 x_0 \approx b_0$ – a slightly incompatible ill-posed system

- A_0 – a smooth integral kernel
- x_0 – discretized smooth function
- Singular values of the data matrix $\begin{bmatrix} A_0 & b_0 \end{bmatrix} = U_0 \Sigma_0 V_0^\top$
- $U_0^\top b_0 \rightsquigarrow$ close-to-nongenericity

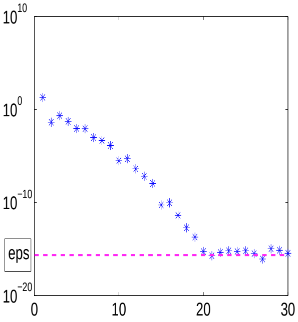


Example: TLS solution of an ill-posed problem

Example

$A_0 x_0 \approx b_0$ – a slightly incompatible ill-posed system

- A_0 – a smooth integral kernel
- x_0 – discretized smooth function
- Singular values of the data matrix $\begin{bmatrix} A_0 & b_0 \end{bmatrix} = U_0 \Sigma_0 V_0^\top$
- $U_0^\top b_0 \rightsquigarrow$ close-to-nongenericity

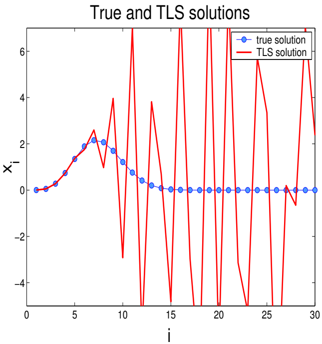


Example: TLS solution of an ill-posed problem

Example

- $A_0 x_0 = b_0$, an $m \times n$ exact system
- x_0 – a discretized smooth function
- $\left[\begin{array}{cc} b & A \end{array}\right] = \left[\begin{array}{cc} b_0 & A_0 \end{array}\right] + noise$
- x – TLS solution of $Ax \approx b$
- Singular values of the data matrix $\left[\begin{array}{cc} A & b \end{array}\right] = U\Sigma V^\top$
  $\rightsquigarrow$ almost equal smallest s.v.
- $U^\top b$

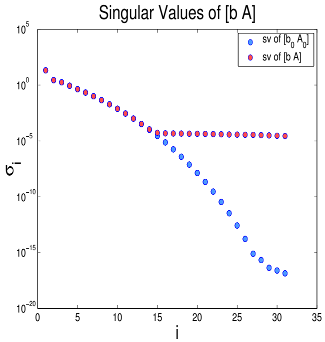
Example: TLS solution of an ill-posed problem



Example

- $A_0 x_0 = b_0$, an $m \times n$ exact system
- x_0 – a discretized smooth function
- $\begin{bmatrix} b & A \end{bmatrix} = \begin{bmatrix} b_0 & A_0 \end{bmatrix} + noise$
- x – TLS solution of $Ax \approx b$
- Singular values of the data matrix $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$ $\rightsquigarrow$ almost equal smallest s.v.
- $U^\top b$

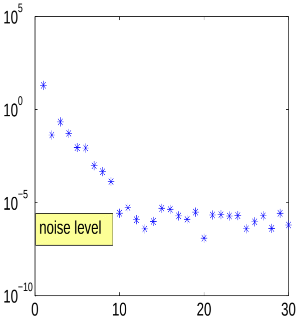
Example: TLS solution of an ill-posed problem



Example

- $A_0 x_0 = b_0$, an $m \times n$ exact system
- x_0 – a discretized smooth function
- $\begin{bmatrix} b & A \end{bmatrix} = \begin{bmatrix} b_0 & A_0 \end{bmatrix} + noise$
- x – TLS solution of $Ax \approx b$
- Singular values of the data matrix $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$ $\rightsquigarrow$ almost equal smallest s.v.
- $U^\top b$

Example: TLS solution of an ill-posed problem



Example

- $A_0 x_0 = b_0$, an $m \times n$ exact system
- x_0 – a discretized smooth function
- $\begin{bmatrix} b & A \end{bmatrix} = \begin{bmatrix} b_0 & A_0 \end{bmatrix} + noise$
- x – TLS solution of $Ax \approx b$
- Singular values of the data matrix $\begin{bmatrix} A & b \end{bmatrix} = U \Sigma V^\top$
  $\rightsquigarrow$ almost equal smallest s.v.
- $U^\top b$

Truncated Total Least Squares

Truncated TLS

- let $k \leq n$ be a truncation level
- compute the nearest rank k approximation of $\begin{bmatrix} A & b \end{bmatrix}$, $\begin{bmatrix} A_k & b_k \end{bmatrix}$, using the SVD
- solve in the TLS sense the 'truncated' problem $A_k x \approx b_k$.

Truncation goals

noise removal, numerical stabilization ...

Note

In the SVD of $\begin{bmatrix} A_k & b_k \end{bmatrix}$ there can be multiple singular values. In particular, the smallest nonzero singular value can be multiple. Thus, non-uniqueness and non-genericity issues can occur!

Truncated Total Least Squares

Truncated TLS

- let $k \leq n$ be a truncation level
- compute the nearest rank k approximation of $\begin{bmatrix} A & b \end{bmatrix}$, $\begin{bmatrix} A_k & b_k \end{bmatrix}$, using the SVD
- solve in the TLS sense the 'truncated' problem $A_k x \approx b_k$.

Truncation goals

noise removal, numerical stabilization ...

Note

In the SVD of $\begin{bmatrix} A_k & b_k \end{bmatrix}$ there can be multiple singular values. In particular, the smallest nonzero singular value can be multiple. Thus, non-uniqueness and non-genericity issues can occur!

Truncated Total Least Squares

Truncated TLS

- let $k \leq n$ be a truncation level
- compute the nearest rank k approximation of $\begin{bmatrix} A & b \end{bmatrix}$, $\begin{bmatrix} A_k & b_k \end{bmatrix}$, using the SVD
- solve in the TLS sense the 'truncated' problem $A_k x \approx b_k$.

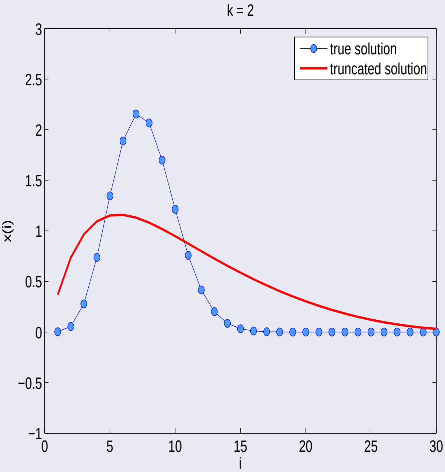
Truncation goals

noise removal, numerical stabilization . . .

Note

In the SVD of $\begin{bmatrix} A_k & b_k \end{bmatrix}$ there can be multiple singular values. In particular, the smallest nonzero singular value can be multiple. Thus, non-uniqueness and non-genericity issues can occur!

Example: effect of truncation levels



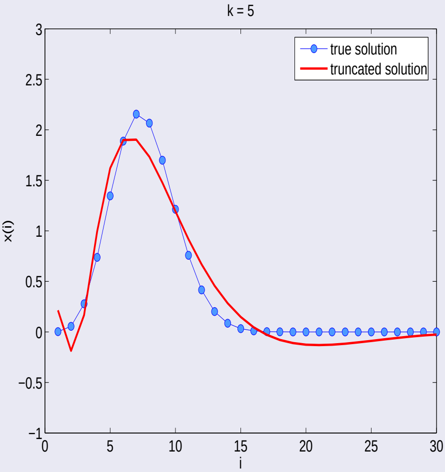
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



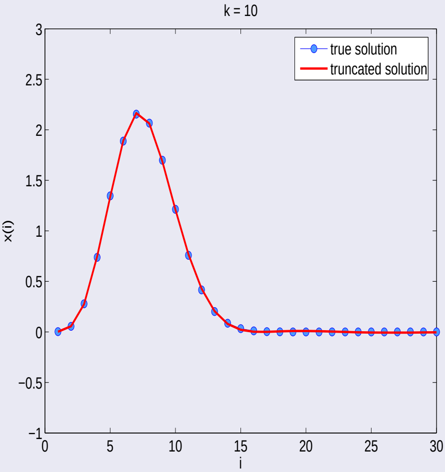
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



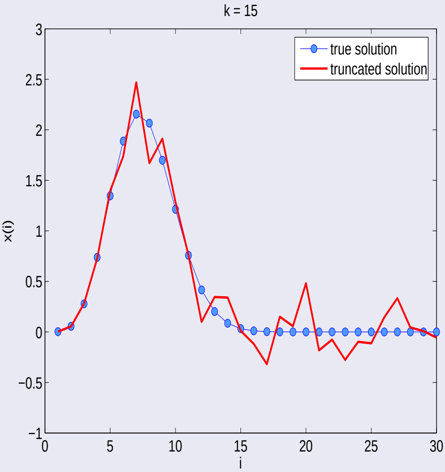
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



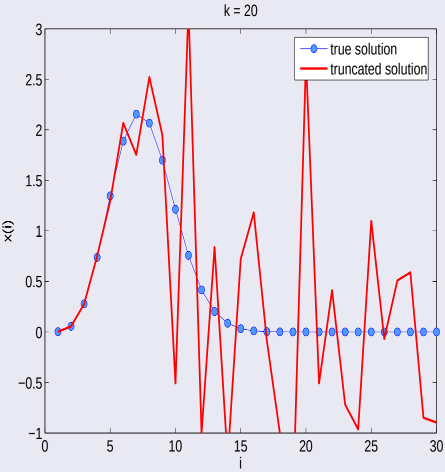
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



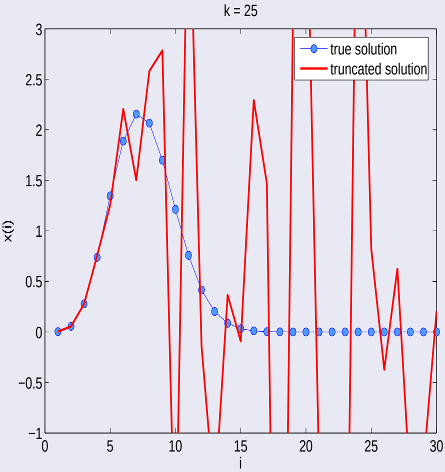
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



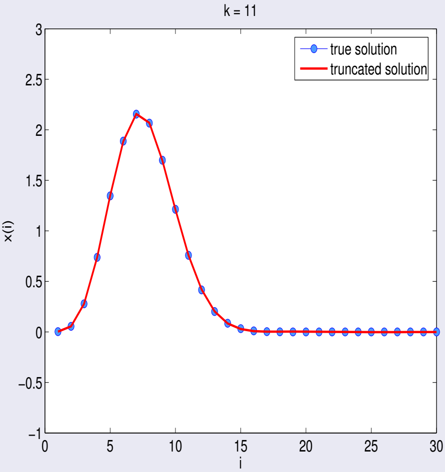
Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Example: effect of truncation levels



Example

The noisy ill-posed problem $Ax \approx b$.
Optimal truncation level at $k = 11$.

Choosing the truncation level

- online performance evaluation
- efficient computations

Truncation during bidiagonalization

Bidiagonalization of $\begin{bmatrix} b & A \end{bmatrix}$

- *Fierro et al.* proposed a Golub-Kahan bidiagonalization algorithm instead of the SVD-based method for Truncated TLS
- motivation: partial bidiagonalization provides (suboptimal) lower rank approximation
- *Paige & Strakoš* studied the properties of the bidiagonalization of $\begin{bmatrix} b & A \end{bmatrix}$
- for well-posed problems, partial bidiagonalization yields core problems that can avoid non-uniquness, non-genericity issues for TLS!

Bidiagonal form of the core decomposition.

$$P^{\top}\left[\begin{array}{cc} b & AQ \end{array}\right] = \left[\begin{array}{c||c|c} b_1 & A_{11} & 0 \\ \hline 0 & 0 & A_{22} \end{array}\right], \quad \left[\begin{array}{cc} b_1 & A_{11} \end{array}\right] = \left[\begin{array}{c||cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & \cdot & \cdot & \cdot & \\ & & & \beta_p & \alpha_p \\ & & & & (\beta_{p+1}) \end{array}\right]$$

where $\alpha_i \beta_i \neq 0$ and β_{p+1} is zero for compatible systems $Ax = b$.

Properties of the core bidiagonal reduction (*Paige & Strakoš*)

- A_{11} is minimally dimensioned
- A_{11} has only distinct and nonzero singular values
- A_{22} need not be bidiagonalized.
- solving the reduced bidiagonal problem $A_{11} x_1 \approx b_1$ with the TLS algorithm and transforming back to the full solution $x = Q \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$ gives the minimum norm TLS solution of $Ax \approx b$.

Truncation during bidiagonalization

Truncation algorithm based on bidiagonalization

1: $k = 0$, $u_0 = b$, $v_0 = 0$
2: **repeat**
3: $k = k + 1$
4: compute the k^{th} bidiagonalization step of $\begin{bmatrix} b & A \end{bmatrix}$:

$$\alpha_i v_i = A^\top u_i - \beta_i v_{i-1}, \qquad \beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i$$

5: compute value of a truncation criterion at k
6: **until** $k = n + 1$ or truncation criterion is satisfied

Truncation during bidiagonalization

Truncation algorithm based on bidiagonalization

1: $k = 0$, $u_0 = b$, $v_0 = 0$
2: **repeat**
3: $\quad k = k + 1$
4: $\quad$ compute the k^{th} bidiagonalization step of $\begin{bmatrix} b & A \end{bmatrix}$:

$$\alpha_i v_i = A^\top u_i - \beta_i v_{i-1}, \qquad \beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i$$

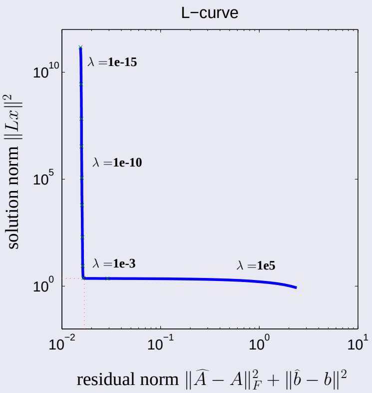
5: $\quad$ compute value of a truncation criterion at k
6: **until** $k = n + 1$ or truncation criterion is satisfied

Methods for choosing the truncation level

L-Curve

- the norm of truncated solution $\|x_k\|_2$ is plotted against norm of residual error $\left\| \left[\begin{array}{cc} b & A \end{array} \right] - \left[\begin{array}{cc} b_k & A_k \end{array} \right] \right\|_F$ for various k's
- the k corresponding to the *corner* is chosen

L-Curve



L−curve
10^{10}
10^{5}
10^{0}
10^{-2}
10^{-1}
10^{0}
10^{1}
λ =1e-15
λ =1e-10
λ =1e-3
λ =1e5
solution norm $\|Lx\|^2$
residual norm $\|\widehat{A} - A\|_F^2 + \|\hat{b} - b\|^2$

Methods for choosing the truncation level

Cross Validation

- involves repeatingly splitting the rows of $\begin{bmatrix} A & b \end{bmatrix}$ into estimation and validation sets
- computing from each validation data the TTLS solution for various k levels
- evaluating the residual error on the validation data
- choosing the best level on the averaged validation tests

simplification of this scenario is not possible, thus the classical CV is not efficient and not implementable online, during bidiagonalization

Methods for choosing the truncation level

Generalized Cross Validation

Regularization for a nonlinear model:

$$GCV: \quad \min_k \frac{\text{residual sum of squares of the model } k \text{ fit}}{(\text{number of degrees of freedom in model } k)^2}$$

We think of the errors-in-variables model

$$(A + \Delta A)x = b + \Delta b, \qquad \Delta A,\ \Delta b \text{ and } x \text{ unknown,}$$

as a nonlinear model, because of the bilinear term $\Delta A \cdot x$.

GCV computation

$$GCV: \quad \min_k \frac{\text{residual sum of squares of the model } k \text{ fit}}{(\text{number of degrees of freedom in model } k)^2}$$

Residual sum of squares – expressed using the bidiagonal reduction:

$$\begin{aligned} \text{RSS} &= \left\| \begin{bmatrix} b & A \end{bmatrix} - \begin{bmatrix} b_k & A_k \end{bmatrix} \right\|_F^2 \\ &= \left\| \begin{bmatrix} b & A \end{bmatrix} \right\|_F^2 - \left\| \begin{bmatrix} b_{11}^{(k)} & A_{11}^{(k)} \end{bmatrix} \right\|_F^2 + \sigma_{\min}\left(\begin{bmatrix} b_{11}^{(k)} & A_{11}^{(k)} \end{bmatrix} \right)^2 \end{aligned}$$

For each k, the RSS computation requires the sum of squares of the α, β elements in the current bidiagonal matrix and the smallest s.v. of the $(k+1) \times k$ bidiagonal matrix $\begin{bmatrix} b_{11}^{(k)} & A_{11}^{(k)} \end{bmatrix}$.

GCV computation

$$GCV: \quad \min_k \frac{\text{residual sum of squares of the model } k \text{ fit}}{(\text{number of degrees of freedom in model } k)^2}$$

number of degrees of freedom = total number of noisy variables - effective number of parameters = $m(n+1) - p_k^{\text{eff}}$

p_k^{eff} is the trace of the influence matrix that makes the link between the reconstructed model and the noisy data:

$$p_k^{\text{eff}} = \mathrm{Tr}\,\frac{\partial \mathrm{vec}\left[\begin{array}{cc} b_k & A_k \end{array}\right]}{\partial \mathrm{vec}\left[\begin{array}{cc} b & A \end{array}\right]} = \frac{1}{2}\mathrm{Tr}\left\{\left(\frac{A_{11}^{(k)\top} A_{11}^{(k)}}{(\sigma'')^2} - I_k + 8(v_1'')^2 x_1^k x_1^{k\top}\right)^{-1}\right\},$$

For each k, the number of degrees of freedom computation requires the inversion of $k \times k$ tridiagonal + rank-one matrix.

Summary

- Truncated Total Least Squares in ill-posed linear systems
- bidiagonalization is used for efficient computations and online optimal truncation level selection
- choice of truncation level: adapted several classical methods

Summary

- Truncated Total Least Squares in ill-posed linear systems
- bidiagonalization is used for efficient computations and online optimal truncation level selection
- choice of truncation level: adapted several classical methods

Summary

- Truncated Total Least Squares in ill-posed linear systems
- bidiagonalization is used for efficient computations and online optimal truncation level selection
- choice of truncation level: adapted several classical methods

References

C. Paige and Z. Strakos (2006) "Core problems in linear algebraic systems", *SIMAX* 27.

R. D. Fierro, G. H. Golub, P. C. Hansen and D. P. O'Leary (1997) "Regularization by truncated total least squares", *SIAM J. Sci. Comput.* 18.

Similarities and differences between Truncated TLS and Core TLS

- Truncated TLS discards smallest $n-k$ singular values of $\begin{bmatrix} b & A \end{bmatrix}$, but keeps the repeats of large singular values, if any.
- Core TLS discards $n-p$ singular values of A, which are only zeros and repeats.

Note

For ill-posed problems, the large singular values are in general distinct, gradually decreasing.

Algorithmic note

Fierro et al. proposed a Lanczos bidiagonalization algorithm for Truncated TLS that is **identical** to the bidiagonalization proposed by *Paige & Strakoš* for Core TLS.

back

The TTLS corrected model for truncation level k

$$\begin{bmatrix} b_k & A_k \end{bmatrix} = \begin{bmatrix} b & A \end{bmatrix} - \frac{(Ax_{\text{TTLS},k} - b)\begin{bmatrix} -1 & x_{\text{TTLS},k}{}^\top \end{bmatrix}}{\|x_{\text{TTLS},k}\|^2 + 1}.$$

back