Author(s) : W. Hart, W. Matt
Submitter :
Language : Fortran

Library: KERNLIB
Submitted: 01.01.1975
Revised: 04.02.1986

Constrained and Unconstrained Linear Least Squares Fitting

The `TL` package finds the least squares solution to a set of unweighted linear equations, possibly subject to a set of equality constraints. The solution is found by Householder triangularisation (see Ref. 1 for details) with parameter elimination if constraints are present. This write-up ends with a few words on generalised least squares fitting (unequal weighting) which is a simple application of the `TL` package.

All matrices are assumed to be stored **row-wise and without gaps**, **contrary** to the Fortran convention, i.e., if the Fortran statement `DIMENSION A(NJ,NI)` reserves memory for the matrix **A** the element A_{ij} is found in word `A(J,I)`.

Structure:

`SUBROUTINE` subprograms
User Entry Names: `TLSC`, `TLS`, `TLERR`, `TLRES`
Internal Entry Names: `TLSMSQ`, `TLSWOP`, `TLUK`, `TLSTEP`, `TLPIV`

Usage:

General Description

Consider the set of M linear equations

$$\sum_{j=1}^{N} A_{ij}x_j = b_i \qquad (i = 1, 2, \dots, M \text{ with } N \le M)$$

to be solved such that the Euclidian norm $||\mathbf{Ax} - \mathbf{b}||_2 = S^2$ is minimised. Instead of determining $\mathbf{x}$ from the Normal Equation $\mathbf{x} = (\mathbf{A'A})^{-1}\mathbf{A'b}$ it is found by applying successive Householder transformations ($\mathbf{Q}$) which reduce $\mathbf{A}$ to upper triangular form without changing the norm of the columns of $\mathbf{A}$ or the vector $\mathbf{b}$. This is beneficial from the point of view of stability and flexibility of application. Writing

$$\mathbf{QA} = \mathbf{R} = \begin{bmatrix} \mathbf{R_1} \\ \mathbf{O} \end{bmatrix} \begin{matrix} \} \; N \text{ rows} \\ \} \; M-N \text{ rows} \end{matrix} \qquad \text{and} \qquad \mathbf{Qb} = \mathbf{y} = \begin{bmatrix} \mathbf{y_1} \\ \mathbf{y_2} \end{bmatrix} \begin{matrix} \} \; N \text{ rows} \\ \} \; M-N \text{ rows} \end{matrix}$$

we have that $||\mathbf{Rx} - \mathbf{y}||_2 = ||\mathbf{Ax} - \mathbf{b}||_2$ and the vector $\mathbf{x}$ is obtained by backward substitution in $\mathbf{R_1x} = \mathbf{y_1}$. As a byproduct, the sum of squares of residuals is directly calculated as $S^2 = ||\mathbf{y_2}||_2$.

Now consider $\mathbf{A}$ and $\mathbf{b}$ to be composed of M_1 constraints to be satisfied exactly, followed by $M - M_1$ equations to be minimised. Writing

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{A_2} \end{bmatrix} \begin{matrix} \} \; M_1 \text{ rows} \\ \} \; M-M_1 \text{ rows} \end{matrix} \quad , \quad \mathbf{b} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{bmatrix} \begin{matrix} \} \; M_1 \text{ rows} \\ \} \; M-M_1 \text{ rows} \end{matrix}$$

then $||\mathbf{A_2x} - \mathbf{b_2}||_2 = S^2$ has to be minimized subject to $\mathbf{A_1x} - \mathbf{b_1} = 0$.

This problem is solved by eliminating M_1 parameters and then evaluating the reduced set of parameters (see Ref. 2 for details).

An attractive feature of the unitary Householder transformations is that when each parameter is eliminated ("solved for") column pivoting allows the selection of that parameter which gives the maximum reduction in the current value of S^2. Thus it is possible to terminate the calculation whenever S^2 or its current reduction become acceptably small. This can be exploited when iterating. If there is more than one RHS vector, then **x** and **b** become $N \times L$ and $M \times L$ matrices with the pivoting strategy applied to the first column of **b**.

The triangular form of $\mathbf{R_1}$ allows the error matrix, **E**, of the fitted parameters to be derived directly from $\mathbf{R_1}$ itself without inverting. The equation is

$$\mathbf{E} = \mathbf{R_1^{-1}}(\mathbf{R_1^{-1}})'.$$

Moreover, the vector of fitted residuals is most easily computed by applying the inverse Householder transformation to $\mathbf{y_2}$, i.e.

$$\mathbf{Ax} - \mathbf{b} = \mathbf{Q}^{-1}\begin{bmatrix}\mathbf{O}\\ \mathbf{y_2}\end{bmatrix}.$$

Note that these residuals do *not have to be calculated* to find the fitted S^2 which is output from the fitting routines.

In all routines described below, the dimensionality of the problem is transmitted via the common block

```
COMMON /TLSDIM/ M1,M,N,L,IER
```

The parameter `IER` returns the number of parameters solved for, or else `-1001` if either `M1` $>$ `N`, `N` $>$ `M` or **A** has rank less than `N`.

Constrained Least Squares Fitting

```
CALL TLSC(A,B,AUX,IPIV,EPS,X)
```

`A`	(`REAL`) The combined constraint / derivative matrix of dimension `M` $\times$ `N`, the upper `M1` rows being the constraints.
`B`	(`REAL`) The combined constraint / measurement matrix of dimension `M` $\times$ `L`, the upper `M1` rows being the constraints.
`X`	(`REAL`) The matrix of dimension `N` $\times$ `L` returning the `L` least squares solutions.
`AUX`	(`REAL`) Working array of length `N` $+ \max($`N`$,$`L`$)$. On output `AUX(J),(J=1,L)` contain the minimised sum of squares.
`IPIV`	(`INTEGER`) Working array of length `N` which holds the exchange information (column pivoting is employed if necessary).
`EPS`	(`REAL`) Parameter specifying a pivoting criterium. There is no exchange of columns I and 1 unless `EPS` $*$ `PIVOT(I)` $>$ `PIVOT(1)`. Typically `EPS` $\simeq$ `0.1`.

Subroutines called: `TLSMSQ`, `TLSWOP`, `TLUK`, `TLSTEP`.

When constraint equations are present, the full pivoting strategy cannot be adopted and so all parameters are solved for, i.e., `IER` returns the value `N` or `-1001`. Under these circumstances `EPS` is used to reduce the amount of pivoting to those cases where it is felt to be absolutely necessary.

Unconstrained Least Squares Fitting

```
CALL TLS(A,B,AUX,IPIV,EPS,X)
```

A (REAL) M $\times$ N derivative matrix.

B (REAL) M $\times$ L matrix of measurements.

X (REAL) N $\times$ L parameter solution matrix.

AUX (REAL) Working array as for TLSC.

IPIV (INTEGER) Working array as for TLSC.

EPS (REAL) Input parameter used for prematurely terminating the calculation:
> 0 : Termination when r.m.s. residual $<$ |EPS|,
< 0 : Termination when the reduction in the residual $<$ |EPS|,
$= 0$: Unconditionally solve for all X_j.

Subroutines called: TLSMSQ, TLSWOP, TLUK, TLSTEP, TLPIV.

As previously indicated, full pivoting is possible without constraints, hence the allowance for premature exit.

Fitted Error Matrix

```
CALL TLERR(A,E,AUX,IPIV)
```

The parameter and subroutine arguments defined previously in COMMON /TLSDIM/ require the output values from a call to TLS or TLSC. E is an $N \times N$ matrix which, upon return, will contain the unnormalised covariance matrix of the fitted parameters, $(\mathbf{A}'\mathbf{A})^{-1}$. A may be overwritten by E and the routine may be called independently from TLS/TLSC by setting IER to zero.

Subroutines called: TLUK, TLSTEP.

Fitted Residuals

```
CALL TLRES(A,B,AUX)
```

All the arguments and common variables require the output values from a call to TLS or TLSC. Upon return, B will give the matrix of residuals, i.e., for each set of least squares equations the column vector $\mathbf{Ax} - \mathbf{b}$.

Subroutine called: TLSTEP.

Notes:

1. The pivoting and exit criteria of TLS are calculated using the first vector of measurements; therefore it is wise to have EPS $= 0$ if L > 1.
2. TLERR and/or TLRES may be called in any order after TLS or TLSC.
3. TLS or TLSC may be used for solving simultaneous linear equations by setting M $=$ N or M1 $=$ N.
4. Useful examples in the application of these routines can be found in the HYDRA Geometry / Kinematics processors.

Generalized Least Squares Fitting

The problem is to minimise $(\mathbf{Ax}-\mathbf{b})'\mathbf{G}(\mathbf{Ax}-\mathbf{b})$ where $\mathbf{G}$, the weight matrix, is the inverse of the error matrix of the measurement vector $\mathbf{b}$. Once again Householder triangularisation offers an attractive alternative to the Normal Equation solution $\mathbf{x}=(\mathbf{A}'\mathbf{G}\mathbf{A})^{-1}\mathbf{A}'\mathbf{G}\mathbf{b}$. The first step is to perform the Choleski decomposition of $\mathbf{G}$, which is positive semi-definite (see `TR` (F112)), such that $\mathbf{G}=\mathbf{U}'\mathbf{U}$, $\mathbf{U}$ being upper triangular. The problem is then reduced to minimising $||\mathbf{A}^1\mathbf{x}-\mathbf{b}^1||_2$, where $\mathbf{A}^1=\mathbf{U}\mathbf{A}$ and $\mathbf{b}^1=\mathbf{U}\mathbf{b}$, which is just the unweighted case previously described. This has the feature that if $\mathbf{A}$ has already been triangularised then the product $\mathbf{UA}$ remains triangular and only back substitution is necessary to find the weighted least squares solution.

References:

1. G. Golub, Numerical methods for solving linear least squares problems, Numer. Math. **7** (1965) 206–216.

2. Å. Björck and G. Golub, Iterative refinement of linear least square solutions by Householder transformation, BIT **7** (1967) 322–337.

•