Package ‘PEIP’

February 19, 2015

Type Package
Title Functions for Aster Book on Inverse Theory
Version 1.7
Date 2012-05-08
Depends R (>= 2.12)
Imports bvls, Matrix, RSEIS, pracma
Author Jonathan M. Lees
Maintainer Jonathan M. Lees <jonathan.lees@unc.edu>
Description Several Functions for Aster Book on Inverse Theory. These functions are transla-
tions of MATLAB code developed by the authors to illustrate the concepts of Inverse the-
ory as applied to geophysics. There is one function, rlsqr, written by Kee-
hoon Kim, that is a wrapper for the FORTRAN based Paige and Saunders LSQR routine.
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-01-07 20:08:11

R topics documented:

PEIP-package .................................................. 2
Ainv ............................................................ 3
art ............................................................... 4
astercode ...................................................... 5
bartl ............................................................. 5
bayes ........................................................... 6
blf2 ............................................................. 7
cgls ............................................................. 8
chis ............................................................. 9
chi2cdf ......................................................... 9
chi2inv ......................................................... 10
dcost .......................................................... 11
error.bar ....................................................... 11
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>flipGSVD</td>
<td>13</td>
</tr>
<tr>
<td>gcval</td>
<td>14</td>
</tr>
<tr>
<td>gcv_function</td>
<td>15</td>
</tr>
<tr>
<td>get_l_rough</td>
<td>15</td>
</tr>
<tr>
<td>ginv</td>
<td>16</td>
</tr>
<tr>
<td>GSVD</td>
<td>17</td>
</tr>
<tr>
<td>idcost</td>
<td>18</td>
</tr>
<tr>
<td>imagesc</td>
<td>18</td>
</tr>
<tr>
<td>interp2grid</td>
<td>19</td>
</tr>
<tr>
<td>irls</td>
<td>20</td>
</tr>
<tr>
<td>irlsIreg</td>
<td>21</td>
</tr>
<tr>
<td>kac</td>
<td>22</td>
</tr>
<tr>
<td>linesconst</td>
<td>23</td>
</tr>
<tr>
<td>lmarq</td>
<td>24</td>
</tr>
<tr>
<td>loadMAT</td>
<td>24</td>
</tr>
<tr>
<td>l_curve_corner</td>
<td>25</td>
</tr>
<tr>
<td>l_curve_tgsvd</td>
<td>26</td>
</tr>
<tr>
<td>l_curve_tikh_gsvd</td>
<td>27</td>
</tr>
<tr>
<td>l_curve_tikh_svd</td>
<td>28</td>
</tr>
<tr>
<td>mcmc</td>
<td>29</td>
</tr>
<tr>
<td>Mnorm</td>
<td>29</td>
</tr>
<tr>
<td>nnz</td>
<td>30</td>
</tr>
<tr>
<td>occam</td>
<td>31</td>
</tr>
<tr>
<td>phi</td>
<td>32</td>
</tr>
<tr>
<td>phiinv</td>
<td>32</td>
</tr>
<tr>
<td>picard_vals</td>
<td>33</td>
</tr>
<tr>
<td>plotconst</td>
<td>34</td>
</tr>
<tr>
<td>quadlin</td>
<td>34</td>
</tr>
<tr>
<td>rlsqr</td>
<td>35</td>
</tr>
<tr>
<td>rnk</td>
<td>38</td>
</tr>
<tr>
<td>sirt</td>
<td>39</td>
</tr>
<tr>
<td>tinv</td>
<td>40</td>
</tr>
<tr>
<td>USV</td>
<td>41</td>
</tr>
<tr>
<td>Vnorm</td>
<td>42</td>
</tr>
<tr>
<td>wGSVD</td>
<td>42</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEIP-package</td>
<td>44</td>
</tr>
</tbody>
</table>

**Description**

Auxilliary functions and routines for running the examples and excersizes described in the book on inverse theory.

**Details**
These functions are used in conjunction with the example described in the PEIP book.

There is one C-code routine, interp2grid. This is introduced to replicate the MATLAB code interp2. It does not work exactly as the matlab code prescribes.

In the PEIP library one LAPACK routine is called: dggsvd. In R, LAPACK routines are stored in slightly different locations on Linux, Windows and Mac computers. Be aware. This will come up in examples from Chapter 4.

Almost all examples work as scripts run with virtually no user input, e.g.

**Author(s)**

Jonathan M. Lees<jonathan.lees.edu> Maintainer: Jonathan M. Lees<jonathan.lees.edu>

**References**


---

**Ainv**

An Inverse Solution

**Description**

QR decomposition solution to Ax=b

**Usage**

\[ \text{Ainv}(\text{GAB, } x, \text{ tol } = 1e-12) \]

**Arguments**

- **GAB**: design matrix
- **x**: right hand side
- **tol**: tolerance for singularity

**Details**

need something to make up for the lame-o matlab code that does this \( h = G\backslash x \) to get the inverse
**Value**

Inverse Solution

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

---

**Description**

ART algorithm for solving sparse linear inverse problems

**Usage**

\[ \text{art}(A, b, \text{tolx}, \text{maxiter}) \]

**Arguments**

- **A**: Constraint matrix
- **b**: right hand side
- **tolx**: difference tolerance for successive iterations (stopping criteria)
- **maxiter**: maximum iterations (stopping criteria).

**Details**

Alpha is a damping factor. If alpha<1, then we won’t take full steps in the ART direction. Using a smaller value of alpha (say alpha=.75) can help with convergence on some problems.

**Value**

\[ x \] solution

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

Compile AsterCode

Description

Compile (source) the PEIP code in local session. Used it the Package is not installed.

Usage

astercode(dir = "RLIB")

Arguments

dir directory where code is stored

Details

This function was created and used before the R-code was packaged as a library.

Value

side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Bartlett window

Description

Bartlett (triangle) window of length m

Usage

bartl(m)

Arguments

m integer, length of vector

Value

vector
**Author(s)**
Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

**Examples**
bartl(1)

---

**Description**

Given a linear inverse problem $Gm=d$, a prior mean $m_{prior}$ and covariance matrix $covm$, data $d$, and data covariance matrix $covd$, this function computes the MAP solution and the corresponding covariance matrix.

**Usage**

`bayes(G, mprior, covm, d, covd)`

**Arguments**

- **G**: Design Matrix
- **mprior**: vector, prior model
- **covm**: vector, model covariance
- **d**: vector, right hand side
- **covd**: vector, data covariance

**Value**

vector model

**Author(s)**
Jonathan M. Lees<jonathan.lees@unc.edu>

**References**
blf2  Bounded least squares

Description

Bounded least squares

Usage

blf2(A, b, c, delta, l, u)

Arguments

<table>
<thead>
<tr>
<th>A</th>
<th>Design Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>Right hand side</td>
</tr>
<tr>
<td>c</td>
<td>matrix weight on x</td>
</tr>
<tr>
<td>delta</td>
<td>tolerance</td>
</tr>
<tr>
<td>l</td>
<td>lower bound</td>
</tr>
<tr>
<td>u</td>
<td>upper bound</td>
</tr>
</tbody>
</table>

Details

Solves the problem: min/max c'x where \( \| Ax - b \| \leq \delta \) and \( l \leq x \leq u \).

Value

x  solution

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References


Conjugate gradient Least squares

**Description**

Conjugate gradient Least squares

**Usage**

cgls(Gmat, dee, niter)

**Arguments**

- Gmat: input matrix
- dee: right hand side
- niter: max number of iterations

**Details**

Performs niter iterations of the CGLS algorithm on the least squares problem min norm(G*m-d). Gmat should be a sparse matrix.

**Value**

- x: matrix of models
- rho: misfit norms
- eta: model norms

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**

**chi**

*Chi function*

---

**Description**

Chi function

**Usage**

\[
\text{chi}(x, n)
\]

**Arguments**

- \(x\) value
- \(n\) degrees of freedom

**Value**

function evaluated

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**References**


---

**chiRcdf**

*Chi-Sq CDF*

---

**Description**

Computes the Chi^2 CDF, using a transformation to N(0,1) on page 333 of Thisted, Elements of Statistical Computing.

**Usage**

\[
\text{chi2cdf}(x, n)
\]

**Arguments**

- \(x\) end value of chi^2 pdf to integrate to. (scalar)
- \(n\) degrees of freedom (scalar)
Note that x and m must be scalars.

probability that Chi^2 random variable is less than or equal to x (scalar).

Jonathan M. Lees<jonathan.lees@unc.edu>


Inverse Chi-Sq

Usage

chi2inv(x, n)

Arguments

probability that Chi^2 random variable is less than or equal to x (scalar).

degrees of freedom

Computes the inverse Chi^2 distribution corresponding to a given probability that a Chi^2 random variable with the given degrees of freedom is less than or equal to x. Uses chi2cdf.m.

corresponding value of x for given probability.

Jonathan M. Lees<jonathan.lees@unc.edu>
**dcost**

**cosine transform**

**Description**
Computes the column-by-column discrete cosine transform of X.

**Usage**
dcost(x)

**Arguments**
- **x**: Time series matrix

**Value**
cosine transformade data

**Author(s)**
Jonathan M. Lees<j jonathan.lees@unc.edu>

---

**error.bar**

**Plot Error Bar**

**Description**
Plot Error Bar

**Usage**
error.bar(x, y, lo, hi, pch = 1, col = 1, barw = 0.1, add = FALSE, ...)

---

**References**

**See Also**
chi, chi2cdf
**Arguments**

- **x**: X-values
- **y**: Y-values
- **lo**: Lower limit of error bars
- **hi**: Upper limit of error bars
- **pch**: plotting character
- **col**: color
- **barw**: width of the bar
- **add**: logical, add=FALSE starts a new plot
- **...**: other plotting parameters

**Value**

graphical side effects

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**Examples**

```r
x = 1:10
y = 2*x+5
zup = rnorm(10)

zup = zup-min(zup)+.5
zdown = rnorm(10)
zdown = zdown-min(zdown)+.2
```

#### example with same error on either side:

```r
error.bar(x, y, y-zup, y+zup, pch = 1, col = 'brown', barw = 0.1, add = FALSE)
```

#### example with different error on either side:

```r
error.bar(x, y, y-zdown, y+zup, pch = 1, col = 'brown', barw = 0.1, add = FALSE)
```
**Description**

Flip (reverse order) output of GSVD

**Usage**

```
flipGSVD(vs, d1 = c(50, 50), d2 = c(48, 50))
```

**Arguments**

- **vs**: list output of GSVD
- **d1**: dimensionals of A
- **d2**: dimensions of B

**Details**

This flipping of the matrix is done to agree with the Matlab code.

**Value**

- **U**
- **V**
- **X**
- **C**
- **S**

**Note**

The GSVD routines are from LAPACK.

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**See Also**

GSVD
### gcval

**Get c-val**

**Description**

Extract the smallest regularization parameter.

**Usage**

\[ \text{gcval}(U, s, b, \text{npoints}) \]

**Arguments**

- **U**: U matrix from gsvd(G, L)
- **s**: \([\text{diag}(C) \; \text{diag}(S)]\) which are the lambdas and mus from the gsvd
- **b**: the data to try and match
- **npoints**: number of alphas to estimate

**Details**

Evaluate the GCV function gcval_function at npoints points.

**Value**

List:

- **reg_min**: alpha with the minimal g (scalar)
- **g**: \(\| \text{Gm}_\text{(alpha,L)} - d \|^2 / (\text{Tr}(I - \text{GG}^\#))\)
- **alpha**: alpha for the corresponding g

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

**See Also**

gcval_function
**gcv_function**

**Description**

Auxiliary routine for GCV calculations

**Usage**

```
gcv_function(alpha, gamma2, beta)
```

**Arguments**

- **alpha**: parameter
- **gamma2**: square of the gamma from the gsvd
- **beta**: projected data to fit

**Value**

vector, \( g - \| G_{m}(\alpha,L) - d \|^{2} / (\text{Tr}(I - GG^{\#})^{2} \)

**Author(s)**

Jonathan M. Lees<jjonathan.lees@unc.edu>

**References**


---

**get_1_rough**  
**One-D Roughening**

**Description**

returns a 1D differentiating matrix operating on a series with \( n \) points.

**Usage**

```
get_1_rough(n, deg)
```

**Arguments**

- **n**: number of data points
- **deg**: order of the derivative to approximate
ginv

Details
Used to get first and 2nd order roughening matrices

Value
Matrix: discrete differentiation matrix

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>

References

Description
Get inverse of matrix x or solve Ax=b.

Usage
ginv(G, x, tol = 1e-12)

Arguments
G Design Matrix
x right hand side
tol tolerance

Details
This function used as alternative to matlab code that does this h = G\x to get the inverse

Value
inverse

Note
Be careful about the usage of tolerance

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>
GSVD

See Also

solve, Ainv

Description

Generalized svd

Usage

GSVD(A, B)

Arguments

A U*E1*t(Q)
B V*E2*t(Q)

Value

U matrix
V matrix
X matrix
C matrix
S matrix

Note

lapack

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>
idcost  

Inverse cosine transform

Description
Takes the column-by-column inverse discrete cosine transform of Y.

Usage
idcost(Y)

Arguments
Y  
Input cosine transform

Value
Time series

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>

References

See Also
dcost

imagesc  

Image Display

Description
Display image in matlab format, i.e. flip and transpose.

Usage
imagesc(G, col = grey((1:99)/100), ...)
interp2grid

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Image matrix</td>
</tr>
<tr>
<td>col</td>
<td>color scale</td>
</tr>
<tr>
<td>...</td>
<td>graphical parameters</td>
</tr>
</tbody>
</table>

Details

Program flips image and transposes.

Value

graphical side effects

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

interp2grid

**Bilinear and Bicubic Interpolation to Grid**

Description

This code was includes bicubic interpolation and bilinear interpolation adapted from Numerical Recipes in C: The art of scientific computing [http://www.nrbook.com/nr3/](http://www.nrbook.com/nr3/) (chapter 3... bicubic interpolation) and a bicubic interpolation from [http://www.paulinternet.nl/?page=bicubic](http://www.paulinternet.nl/) in java code.

Inputs are a list of points to interpolate to from raster objects of class 'asc' (adehabitat package), 'RasterLayer' (raster package) or 'SpatialGridDataFrame' (sp package).

Usage

```
interp2grid(mat,xout,yout,xin=NULL,yin=NULL,type=2)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mat</td>
<td>a matrix of data that can be a raster matrix of class 'asc' (adehabitat package), 'RasterLayer' (raster package) or 'SpatialGridDataFrame' (sp package) NA values are not permitted. data must be complete.</td>
</tr>
<tr>
<td>xout</td>
<td>a vector of data representing x coordinates of the output grid. Resulting grid must have square cell sizes if mat is of class 'asc', 'RasterLayer' or 'SpatialGridDataFrame'.</td>
</tr>
<tr>
<td>yout</td>
<td>a vector of data representing x coordinates of the output grid. Resulting grid must have square cell sizes if mat is of class 'asc', 'RasterLayer' or 'SpatialGridDataFrame'.</td>
</tr>
</tbody>
</table>
**x**in  
A vector identifying the locations of the columns of the input data matrix. These are automatically populated if mat is of class 'asc', 'RasterLayer' or 'SpatialGridDataFrame'.

**y**in  
A vector identifying the locations of the rows of the input data matrix. These are automatically populated if mat is of class 'asc', 'RasterLayer' or 'SpatialGridDataFrame'.

**type**  
An integer value representing the type of interpolation method used.
1 - bilinear adapted from Numerical Recipes in C  
2 - bicubic adapted from Numerical Recipes in C  
3 - bicubic adapted from online java code

**Value**

Returns a matrix of the originating class.

**Author(s)**

Jeremy VanDerWal <jjvanderwal@gmail.com>

**Examples**

```r
tx = seq(0, 3, 0.1)
ty = seq(0, 3, 0.1)

# Create a matrix of random values
mat = matrix(rnorm(100), nrow=10)
txin = seq(0, 3, length=4)tyin = seq(0, 3, length=4)

# Interpolation methods
bilinear = interp2grid(mat, tx, ty, xin, yin, type=1)
bicubic2 = interp2grid(mat, tx, ty, xin, yin, type=2)
bicubic3 = interp2grid(mat, tx, ty, xin, yin, type=3)

# Visualize the interpolated results
par(mfrow=c(2,2), cex=1)
image(mat, main='base', zlim=c(0,16), col=heat.colors(100))
image(bilinear, main='bilinear', zlim=c(0,16), col=heat.colors(100))
image(bicubic2, main='bicubic2', zlim=c(0,16), col=heat.colors(100))
image(bicubic3, main='bicubic3', zlim=c(0,16), col=heat.colors(100))
```

**description**

Iteratively reweight least squares

**Description**

Uses the iteratively reweight least squares strategy to find an approximate \( L_p \) solution to \( Ax=b \).
irlsl1reg

Usage

    irls(A, b, tolr, tolx, p, maxiter)

Arguments

A     Matrix of the system of equations.
b     Right hand side of the system of equations
tolr  Tolerance below which residuals are ignored
tolx  Stopping tolerance. Stop when (norm(newx-x)/(1+norm(x)) < tolx)
p     Specifies which p-norm to use (most often, p=1.)
maxiter Limit on number of iterations of IRLS

Details

Use to get L-1 norm solution of inverse problems.

Value

    x     Approximate L_p solution

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References


__________________________

irlsl1reg                 L1 least squares with sparsity
__________________________

Description

Solves the system Gm=d using sparsity regularization on Lm. Solves the L1 regularized least squares problem: min norm(G*m-d,2)^2+alpha*norm(L*m,1)

Usage

    irlsl1reg(G, d, L, alpha, maxiter = 100, tolx = 1e-04, tolr = 1e-06)
### Arguments

- **G**: design matrix
- **d**: right hand side
- **L**: regularization matrix
- **alpha**: regularization parameter
- **maxiter**: Maximum number of IRLS iterations
- **tolx**: Tolerance on successive iterates
- **tolr**: Tolerance below which we consider an element of L*m to be effectively zero

### Value

- **m**: model vector

### Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

### References


---

**kac**  
*Kaczmarz*

### Description

Implements Kaczmarz’s algorithm to solve a system of equations iteratively

### Usage

```
kac(A, b, tolx, maxiter)
```

### Arguments

- **A**: Constraint matrix
- **b**: right hand side
- **tolx**: difference tolerance for successive iterations (stopping criteria)
- **maxiter**: maximum iterations (stopping criteria)

### Value

- **x**: solution
linesconst

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>

References

linesconst  Plot constant model

Description
Add to plotting model in piecewise constant form over n subintervals, where n is the length of x.

Usage
linesconst(x, l, r, ...)

Arguments
x model to be plotted
l left endpoint of plot
r right endpoint of plot
... graphical parameters

Details
Used for plotting vector models

Value
graphical side effects

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>

See Also
plotconst
lmarq  \textit{Lev-Marquardt Inversion}

\textbf{Description}

Use the Levenberg-Marquardt algorithm to minimize \( f(p) = \sum (F_i(p)^2) \)

\textbf{Usage}

\[ \text{lmarq}(\text{afun}, \text{ajac}, p0, \text{tol}, \text{maxiter}) \]

\textbf{Arguments}

- \text{afun}: name of the function \( F(x) \)
- \text{ajac}: name of the Jacobian function \( J(x) \)
- \text{p0}: initial guess
- \text{tol}: stopping tolerance
- \text{maxiter}: maximum number of iterations allowed

\textbf{Value}

- \text{pstar}: best solution found.
- \text{iter}: iteration count.

\textbf{Author(s)}

Jonathan M. Lees<jonathan.lees@unc.edu>

loadMAT \textit{Load a Matlab matfile}

\textbf{Description}

Load a Matlab matfile, rename the internal parameters to get R-objects

\textbf{Usage}

\[ \text{loadMAT}(\text{fn}, \text{pos}=1) \]

\textbf{Arguments}

- \text{fn}: file name of MATfile
- \text{pos}: integer, position in search path, default=1
Details

Program reads in previously saved mat-files and extracts the data, and renames the variables to match the book.

Value

Whatever is in the MATfile

Note

Matfiles are created using the matlab2R routines

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

---

**l_curve_corner**  
**L Curve Corner**

Description

Retrieve corner of L-curve

Usage

```r
l_curve_corner(rho, eta, reg_param)
```

Arguments

- `rho`: misfit
- `eta`: model norm or seminorm
- `reg_param`: regularization parameter

Value

- `reg_corner`: the value of `reg_param` with maximum curvature
- `ireg_corner`: the index of the value in `reg_param` with maximum curvature
- `kappa`: the curvature for each `reg_param`

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Description

L curve parematers and models for truncated gsvd regularization.

Usage

l_curve_tgsvd(u, d, x, Lam, G, L)

Arguments

U U, output of GSVD
d output of GSVD
X output of GSVD
Lam output of GSVD
G output of GSVD
L output of GSVD

Value

List:
eta the solution seminorm ||Lm||
rho the residual norm ||G m - dl||
reg_param corresponding regularization parameters
m corresponding suite of models for truncated GSVD

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

Description

L-curve tikh gsvd

Usage

l_curve_tikh_gsvd(U, d, X, Lam, Mu, G, L, npoints, varargin = NULL)

Arguments

U           from the gsvd
d           data vector for the problem G*m=d
X           from the gsvd
Lam         from the gsvd
Mu          from the gsvd
G           system matrix
L           roughening matrix
npoints     alpha_min, alpha_max: if specified, constrain the logarithmically spaced regularization parameter range, otherwise an attempt is made to estimate them from the range of generalized singular values

Details

Uses output of GSVD

Value

eta          - the solution seminorm ||Lm||
rho          - the residual norm ||G m - d||
reg_param    - corresponding regularization parameters
m            - corresponding suite of models for truncated GSVD

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>
Description

L-curve for Tikhonov regularization

Usage

\[ l\text{\_curve\_tikh\_svd}(U, s, d, npoints, varargin = NULL) \]

Arguments

- \texttt{U} matrix of data space basis vectors from the svd
- \texttt{s} vector of singular values
- \texttt{d} the data vector
- \texttt{npoints} the number of logarithmically spaced regularization parameters
- \texttt{varargin} \texttt{alpha\_min, alpha\_max}: if specified, constrain the logarithmically spaced regularization parameter range, otherwise an attempt is made to estimate them from the range of singular values

Details

Calculates the L-curve

Value

- \texttt{eta} the solution norm \( \|m\| \) or seminorm \( \|Lm\| \)
- \texttt{rho} the residual norm \( \|Gm - d\| \)
- \texttt{reg\_param} corresponding regularization parameters

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>
**mcmc**

*Maximum likelihood Models*

**Description**

Maximum likelihood Models

**Usage**

\[
\text{mcmc}(\text{alogprior}, \text{aloglikelihood}, \text{agenerate}, \text{alogproposal}, m0, niter)
\]

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alogprior</td>
<td>Name of a function that computes the log of the prior distribution.</td>
</tr>
<tr>
<td>aloglikelihood</td>
<td>Name of a function that computes the log of the likelihood.</td>
</tr>
<tr>
<td>agenerate</td>
<td>Name of a function that generates a random model from the current model using the proposal distribution (r(x,y)).</td>
</tr>
<tr>
<td>alogproposal</td>
<td>Name of a function that computes the log of the proposal distribution (r(x,y)).</td>
</tr>
<tr>
<td>m0</td>
<td>Initial model</td>
</tr>
<tr>
<td>niter</td>
<td>Number of iterations to perform</td>
</tr>
</tbody>
</table>

**Value**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mout</td>
<td>MCMC samples</td>
</tr>
<tr>
<td>mMAP</td>
<td>Best model found in the MCMC simulation.</td>
</tr>
<tr>
<td>accrate</td>
<td>Acceptance rate</td>
</tr>
</tbody>
</table>

**Author(s)**

Jonathan M. Lees<jonathan.lees@unc.edu>

---

**mnorm**

*Matrix Norm*

**Description**

Matrix Norm

**Usage**

\[
mnorm(X, k = 2)
\]
Arguments

- $x$ matrix
- $k$ norm number

Details

returns the largest singular value of the matrix or vector

Value

Scalar Norm

Note

if $k=1$, absolute value; $k=2$ 2-norm (rms); $k>2$, largest singular value.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

---

nnz $Non$-zeros

Description

Number of non-zero elements in a vector

Usage

nnz($h$)

Arguments

- $h$ vector

Value

integer

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>
Description

Occam’s inversion

Usage

occam(afun, ajac, L, d, m0, delta)

Arguments

afun character, function handle that computes the forward problem
ajac character, function handle that computes the Jacobian of the forward problem
L regularization matrix
d data that should be fit
m0 guess at the model
delta cutoff to use for the discrepancy principle portion

Value

vector, model found

Note

This is a simple brute force way to do the line search. Much more sophisticated methods are available. Note: we’ve restricted the line search to the range from 1.0e-20 to 1. This seems to work well in practice, but might need to be adjusted for a particular problem.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

bayes
phi \hspace{1cm} \textit{Integral of Normal Distribution}

\textbf{Description}

normal distribution and returns the value of the integral

\textbf{Usage}

\texttt{phi(x)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{0.5cm} \text{endpoint of integration (scalar)}
\end{itemize}

\textbf{Value}

value of integral

\textbf{Author(s)}

Jonathan M. Lees<jonathan.lees@unc.edu>

\textbf{See Also}

\texttt{erf}

---

\texttt{phiinv} \hspace{1cm} \textit{Inverse Normal Distribution Integral}

\textbf{Description}

Calculates the inverse normal distribution from the value of the integral

\textbf{Usage}

\texttt{phiinv(x)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{0.5cm} \text{endpoint value of integration (scalar)}
\end{itemize}

\textbf{Value}

value of integral (scalar)
**picard_vals**

**Author(s)**
Jonathan M. Lees<jonathan.lees@unc.edu>

**See Also**
phi

---

**Description**
Picard plot parameters for subsequent plotting

**Usage**

`picard_vals(U, sm, d)`

**Arguments**
- **U**: the U matrix from the SVD or GSVD
- **sm**: singular values in decreasing order, or the lambdas divided by the mus in decreasing order
- **d**: data to fit

**Value**
List:
- **utd**: the columns of U transposed times d
- **utd_norm**: utd./sm

**Author(s)**
Jonathan M. Lees<jonathan.lees@unc.edu>
plotconst  
*Plot constant model*

**Description**

Plots a model in piecewise constant form over n subintervals, where n is the length of x.

**Usage**

```r
plotconst(x, l, r, ...)
```

**Arguments**

- `x`  
  model to be plotted
- `l`  
  left endpoint of plot
- `r`  
  right endpoint of plot
- `...`  
  graphical parameters

**Details**

Used for plotting vector models

**Value**

graphical side effects

**Author(s)**

Jonathan M. Lees <jonathan.lees@unc.edu>

**See Also**

`linesconst`

---

quadlin  
*Lagrange multiplier technique*

**Description**

Quadratic Linearization

**Usage**

```r
quadlin(Q, A, b)
```
Arguments

- **Q**: positive definite symmetric matrix
- **A**: matrix with linearly independent rows
- **b**: data vector

Details

Solves the problem: \( \min \left( \frac{1}{2} \right) t(x)^*Q*x \) with \( Ax = b \) using the Lagrange multiplier technique, where \( Q \) is assumed to be symmetric and positive definite and the rows of \( A \) are linearly independent.

Value

- **list**: vector of solution values
- **lambda**: Lagrange multiplier

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Description

R Version of LSQR routine written in Fortran by Paige and Saunders.

Usage

```r
rlsqr(G=matrix(), u=vector(), wantse = 0, damp = 0, atol = 0, btol = 0, conlim = 0, itnlim = 100, nout = 0)
```

Arguments

- **G**: Design Matrix
- **u**: data vector
- **wantse**: weighting?
- **damp**: Damping parameter
- **atol**: a Tolerance
- **btol**: b Tolerance
- **conlim**: con
- **itnlim**: Iteration limit
- **nout**: integer, output file (Not Available)
Details

This code is an R wrapper for performing the LSQR routine by Paige and Saunders. The LSQR program is a popular inversion program for solving the least squares problem, Ax=b.

Value

Solution vector

Note

The fortran code has write statements that are not available in R-code according to the standard R documentation.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>, Keehoon Kim<keehoon@email.unc.edu>

References


See Also

sirt, art

Examples

```r
## Not run:

library(png)
library(PEIP)
shift=function(x,ns)
{
  nsig=c(rep(0,ns),x[1:(length(x)-ns)])
  return(nsig)
}

imagesc<-function(G, col=grey((1:99)/100), ...)
{
  ###### plot an image after flipping and transposing
  ###### to match the matlab code
  d = dim(G)
  b = G[d[1]:1,]
  image(t(b), col=col, ...)
}

img= readPNG('original_image.png')
im = matrix(as.vector(img), 40000, 1)
```
rlsqr

load('rtest.RDATA')

# Blurring the image (DATA == d)

d0=G
d=d0

# Solving damped least squares problem
# ( G ) * x = ( d )
# ( damp * I ) ( 0 )

damp = lambda
lambda=0

# data vector (u)
u=d

# tolerance of iteration
atol=1.0e-6;btol=1.0e-6
atol=.Machine$double.eps;btol=.Machine$double eps

# taking condition number of G into consideration ( 0 means to ignore it )
conlim=0

# the number of iteration of LSQR
itnlim=1000;

# Running LSQR

# Executing LSQR
lx=rlsqr(G=G,u=u,damp=lambda,itnlim=itnlim,atol=atol,btol=btol,conlim=0)

# Results of LSQR

layout(matrix(1:3, ncol=3))
imagesc(matrix(img,200,200),main='Original Image',axes=FALSE, xlab='',ylab='')
imagesc(matrix(d,200,200),main='Blurred Image',axes=FALSE,xlab='',ylab='')
Rank of Matrix

Description

Return the rank of a matrix. Not to be confused with the R function rank.

Usage

rnk(G, tol = 1e-14)

Arguments

G    Matrix

tol  machine tolerance for small numbers

Details

Number of singular values greater than tol.

Value

integer, number of non-zero singular values

Note

duplicate the matlab function rank

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

svd
Examples

hilbert <- function(n) { i <- 1:n; 1 / outer(i - 1, i, "+") }
X <- hilbert(9)[,1:6]
rmk(X)

sirt

SIRT Algorithm for sparse matrix inversion

Description

Row action method for inversion of matrices

Usage

sirt(A, b, tolx, maxiter)

Arguments

A Design Matrix
b vector, Right hand side
tolx numeric, tolerance for stopping
maxiter integer, Maximum iterations

Details

Iterates until conversion

Value

Solution vector

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

References

See Lees, 1989

See Also

art, kac
tinv

Inverse T-distribution

Description
Inverse T-distribution, qt

Usage
tinv(p, nu)

Arguments
p   P-value
nu  degrees of freedom

Details
Wrapper for qt

Value
Quantile for T-distribution

Author(s)
Jonathan M. Lees<jonathan.lees@unc.edu>

See Also
qt

Examples
tinv(.4, 10)
Description

Singular Value Decomposition

Usage

USV(G)

Arguments

G Matrix

Details

returns matrices U, S, V according to matlab convention.

Value

list:

U Matrix
S Matrix, singular values
V Matrix

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

svd

Examples

hilbert <- function(n) { i <- 1:n; 1 / outer(i - 1, i, "+") }
X <- hilbert(9)[,1:6]

h = USV(X)

print( h$U )
**Vnorm**  
*Vector 2-Norm*

Description

Vector 2-Norm.

Usage

Vnorm(x)

Arguments

x  
numeric vector

Value

Numeric scale norm

Note

This function is intended to duplicated the matlab function norm.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

Examples

V = Vnorm(rnorm(10))

---

**wGSVD**  
*wrapper for GSVD*

Description

wrapper for GSVD, dggsvd, generalized SVD

Usage

wGSVD(A, B)
wGSVD

Arguments

A matrix, \( A = U^*E_1^*Q \)
B matrix, \( V^*E_2^*Q \)

Details

Uses LAPACK routine dggsvd

Value

list: U, V, X, C, S

Note

This function calls the LAPACK routine installed on the system. For LINUX, MAC and WIN-
DOWS there may be slightly different implementation.

The order of the singular values returned are according to Matlab convention, smallest to largest.

Author(s)

Jonathan M. Lees<jonathan.lees@unc.edu>

See Also

svd
## Index

*Topic **misc**
- Ainv, 3
- art, 4
- astercode, 5
- bartl, 5
- bayes, 6
- blf2, 7
- cglsl, 8
- chi, 9
- chi2cdf, 9
- chi2inv, 10
- dcost, 11
- error.bar, 11
- flipGSVD, 13
- gcval, 14
- get_1_rough, 15
- ginv, 16
- GSVD, 17
- idcost, 18
- imagesc, 18
- irls, 20
- irlsl1reg, 21
- kac, 22
- l_curve_corner, 25
- l_curve_tgsvd, 26
- l_curve_tikh_gsvd, 27
- l_curve_tikh_svd, 28
- linesconst, 23
- lmarq, 24
- loadMAT, 24
- mcmc, 29
- Mnorm, 29
- nnz, 30
- occam, 31
- phi, 32
- phiinv, 32
- picard_vals, 33
- plotconst, 34

*Topic **package**
- PEIP-package, 2

- Ainv, 3
- art, 4
- astercode, 5
- bartl, 5
- bayes, 6
- blf2, 7
- cglsl, 8
- chi, 9
- chi2cdf, 9
- chi2inv, 10
- dcost, 11
- error.bar, 11
- flipGSVD, 13
- gcval, 14
- get_1_rough, 15
- ginv, 16
- GSVD, 17
- idcost, 18
- imagesc, 18
- interpRgrid, 19
- irls, 20
- irlsl1reg, 21

- quadlin, 34
- rlsqrl, 35
- rnk, 38
- sirt, 39
- tinv, 40
- USV, 41
- Vnorm, 42
- wGSVD, 42

---

44
INDEX

kac, 22

l_curve_corner, 25
l_curve_tgsvd, 26
l_curve_tikh_gsvd, 27
l_curve_tikh_svd, 28
linesconst, 23
lmarq, 24
loadMAT, 24

mcmc, 29
Mnorm, 29

nnz, 30

occam, 31

PEIP (PEIP-package), 2
PEIP-package, 2
phi, 32
phiinv, 32
picard_vals, 33
plotconst, 34

quadlin, 34

rlsqr, 35
rnk, 38

sirt, 39

tinv, 40

USV, 41

Vnorm, 42

wGsvd, 42