Package ‘R1magic’

August 29, 2016

Type Package

Title Compressive Sampling: Sparse Signal Recovery Utilities

Version 0.3.2

Date 2015-04-19

Maintainer Mehmet Suzen <mehmet.suzen@physics.org>

Depends stats, utils

Description Utilities for sparse signal recovery suitable for compressed sensing. L1, L2 and TV penalties, DFT basis matrix, simple sparse signal generator, mutual cumulative coherence between two matrices and examples, Lp complex norm, scaling back regression coefficients.

License GPL (>= 3)

LazyLoad yes

NeedsCompilation no

Author Mehmet Suzen [aut, cre]

Repository CRAN

Date/Publication 2015-04-19 22:18:58

R topics documented:

`R1magic-package` *Compressive Sampling: Sparse signal recovery utilities*

Description

Utilities for sparse signal recovery suitable for compressed sensing. L1, L2 and TV penalties, DFT basis matrix, simple sparse signal generator, mutual cumulative coherence between two matrices and examples, Lp complex norm, scaling back regression coefficients.

Details

Package:	R1magic
Type:	Package
Version:	0.3.2
Date:	2014-04-19
License:	GPL (>= 3)
LazyLoad:	yes

Author(s)

Mehmet Suzen Maintainer: Mehmet Suzen <mehmet.suzen@physics.org>

References

Emmanuel Candes, Justin Romberg, and Terence Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. (IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006)

Emmanuel Candes and Justin Romberg, Quantitative robust uncertainty principles and optimally sparse decompositions. (Foundations of Comput. Math., 6(2), pp. 227 - 254, April 2006)

David Donoho, Compressed sensing. (IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, April 2006)

Examples

```
CompareL1_L2_TV1(100,10,0.1);
```

CompareL1_L2_TV1 *Compare L1, L2 and TV on a sparse signal.*

Description

Compare L1, L2 and TV on a sparse signal.

Usage

```
CompareL1_L2_TV1(N, M, per)
```

Arguments

N	Size of the sparse signal to generate , integer.
M	Number of measurements.
per	Percentage of spikes.

Author(s)

Mehmet Suzen

DFTMatrix0 *Generate Discrete Fourier Transform Matrix using DFTMatrixPlain.*

Description

Generate Discrete Fourier Transform Matrix (NxN).

Usage

```
DFTMatrix0(N)
```

Arguments

N	Integer value determines the dimension of the square matrix.

Value

It returns a NxN square matrix.

Author(s)

Mehmet Suzen

See Also

DFTMatrixPlain

Examples

```
DFTMatrix0(2)
```

`DFTMatrixPlain` *Generate Plain Discrete Fourier Transform Matrix without the coefficient*

Description

Generate plain Discrete Fourier Transform Matrix (NxN) without a coefficient.

Usage

```
DFTMatrixPlain(N)
```

Arguments

`N`	Integer value defines the dimension of the square plain DFT matrix.

Value

It returns a NxN square matrix.

Author(s)

Mehmet Suzen

Examples

```
DFTMatrixPlain(2)
```

`GaussianMatrix` *Generate Gaussian Random Matrix*

Description

Generate Gaussian Random Matrix (zero mean and standard deviation one.)

Usage

```
GaussianMatrix(N, M)
```

Arguments

`N`	Integer value determines number of rows.
`M`	Integer value determines number of columns.

Value

Returns MxN matrix.

Author(s)

Mehmet Suzen

Examples

```
GaussianMatrix(3,2)
```

`Lnorm` *L-p norm of a given complex vector*

Description

L-p norm of a given complex vector

Usage

```
Lnorm(X, p)
```

Arguments

`X,`	a complex vector, can be real too.
`p,`	norm value

Value

L-p norm of the complex vector

Author(s)

Mehmet Suzen

`mutualCoherence` *Cumulative mutual coherence*

Description

Generate vector of cumulative mutual coherence of a given matrix up to a given order. \ Mutual Cumulative Coherence of a Matrix A at order k is defined as $M(A,k) = max_p max_{p \neq q, q \in \Omega} \sum_q | < a_p, a_q > | / (|a_p||a_q|)$

Usage

```
mutualCoherence(A, k)
```

Arguments

`A`	A matrix.
`k`	Integer value determines number of columns or the order of mutual coherence function to .

Value

Returns k-vector

Author(s)

Mehmet Suzen

References

Compressed sensing in diffuse optical tomography \ M. Suzen, A.Giannoula and T. Durduran, \ Opt. Express 18, 23676-23690 (2010) \ J. A. Tropp \ Greed is good: algorithmic results for sparse approximation, \IEEE Trans. Inf. Theory 50, 2231-2242 (2004)

Examples

```
set.seed(42)
B <- matrix(rnorm(100), 10, 10) # Gaussian Random Matrix
mutualCoherence(B, 3) # mutual coherence up to order k
```

objective1TV — *1-D Total Variation Penalized Objective Function*

Description

1-D Total Variation Penalized Objective Function

Usage

```
objective1TV(x, T, phi, y, lambda)
```

Arguments

x	Initial value of the vector to be recovered. Sparse representation of the vector (N x 1 matrix) X=Tx, where X is the original vector
T	sparsity bases (N x N matrix)
phi	Measurement matrix (M x N).
y	Measurement vector (Mx1).
lambda	Penalty coefficient.

Value

Returns a vector.

Author(s)

Mehmet Suzen

objectiveL1 — *Objective function for ridge L1 penalty*

Description

Objective function for ridge L1 penalty

Usage

```
objectiveL1(x, T, phi, y, lambda)
```

Arguments

x,	unknown vector
T,	transform bases
phi,	measurement matrix
y,	measurement vector
lambda,	penalty term

Note

Thank you Jason Xu of Washington University for pointing out complex number handling

Author(s)

Mehmet Suzen

`objectiveL2` *Objective function for Tikhinov L2 penalty*

Description

Objective function for Tikhinov L2 penalty

Usage

```
objectiveL2(x, T, phi, y, lambda)
```

Arguments

`x,`	unknown vector
`T,`	transform bases
`phi,`	measurement matrix
`y,`	measurement vector
`lambda,`	penalty term

Note

Thank you Jason Xu of Washington University for pointing out complex number handling

Author(s)

Mehmet Suzen

`oo` *Frequency expression for DFT*

Description

Frequency expression for DFT

Usage

```
oo(p, omega)
```

Arguments

`p`	Exponent
`omega`	Omega expression for DFT

Author(s)

Mehmet Suzen

`scaleBack.lm` *Transform back multiple regression coefficients to unscaled regression coefficients Original question posed by Mark Seeto on the R mailing list.*

Description

Transform back multiple regression coefficients to unscaled regression coefficients Original question posed by Mark Seeto on the R mailing list.

Usage

```
scaleBack.lm(X, Y, betas.scaled)
```

Arguments

`X,`	unscaled design matrix without the intercept, m by n matrix
`Y,`	unscaled response, m by 1 matrix
`betas.scaled,`	coefficients vector of multiple regression, first term is the intercept

Note

2015-04-10

Author(s)

M.Suzen

Examples

```
set.seed(4242)
 X            <- matrix(rnorm(12), 4, 3)
 Y            <- matrix(rnorm(4), 4, 1)
 betas.scaled <- matrix(rnorm(3), 3, 1)
 betas        <- scaleBack.lm(X, Y, betas.scaled)
```

---

solve1TV *1-D Total Variation Penalized Nonlinear Minimization*

Description

1-D Total Variation Penalized Nonlinear Minimization

Usage

```
solve1TV(phi,y,T,x0,lambda=0.1)
```

Arguments

x0	Initial value of the vector to be recovered. Sparse representation of the vector (N x 1 matrix) X=Tx, where X is the original vector
T	sparsity bases (N x N matrix)
phi	Measurement matrix (M x N).
y	Measurement vector (Mx1).
lambda	Penalty coefficient. Defaults 0.1

Value

Returns nlm object.

Author(s)

Mehmet Suzen

solveL1 *l1 Penalized Nonlinear Minimization*

Description

l1 Penalized Nonlinear Minimization

Usage

```
solveL1(phi,y,T,x0,lambda=0.1)
```

Arguments

x0	Initial value of the vector to be recovered. Sparse representation of the vector (N x 1 matrix) X=Tx, where X is the original vector
T	sparsity bases (N x N matrix)
phi	Measurement matrix (M x N).
y	Measurement vector (Mx1).
lambda	Penalty coefficient. Defaults 0.1

Value

Returns nlm object.

Author(s)

Mehmet Suzen

solveL2 *l2 Penalized Nonlinear Minimization*

Description

l2 Penalized Nonlinear Minimization

Usage

```
solveL2(phi,y,T,x0,lambda=0.1)
```

Arguments

x0	Initial value of the vector to be recovered. Sparse representation of the vector (N x 1 matrix) X=Tx, where X is the original vector
T	sparsity bases (N x N matrix)
phi	Measurement matrix (M x N).
y	Measurement vector (Mx1).
lambda	Penalty coefficient. Defaults 0.1

Value

Returns nlm object.

Author(s)

Mehmet Suzen

`sparseSignal` *Sparse digital signal Generator.*

Description

Sparse digital signal Generator with given thresholds.

Usage

```
sparseSignal(N, s, b = 1, delta = 1e-07, nlev = 0.05, slev = 0.9)
```

Arguments

`N`	Number of signal components, vector size.
`s`	Number of spikes, significatn components
`b`	Signal bandwidth, defaults 1.
`delta`	Length of discrete distances among components, defaults 1e-7.
`nlev`	Maximum value of insignificant component, relative to b, defaults to 0.05
`slev`	Maximum value of significant component, relative to b, defaults to 0.9

Author(s)

Mehmet Suzen

`TV1` *1-D total variation of a vector.*

Description

1-D total variation of a vector.

Usage

```
TV1(x)
```

Arguments

`x`	A vector.

Author(s)

Mehmet Suzen

Index