Package ‘gasper’

February 11, 2023

Type Package

Title Graph Signal Processing

Version 1.1.3

Description Provides the standard operations for signal processing on graphs:
graph Fourier transform, spectral graph wavelet transform,
visualization tools. It also implements a data driven method
for graph signal denoising/regression, for details see
De Loynes, Navarro, Olivier (2019) <arxiv:1906.01882>.
The package also provides an interface to the SuiteSparse Matrix Collection,
<https://sparse.tamu.edu/>, a large and widely used set of sparse matrix
benchmarks collected from a wide range of applications.

URL https://github.com/fabnavarro/gasper

BugReports https://github.com/fabnavarro/gasper/issues

License LGPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports Rcpp, ggplot2, methods, Matrix, RSpectra, httr, curl

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Fabien Navarro [aut, cre],
Basile de Loynes [aut],
Baptiste Olivier [aut]

Maintainer Fabien Navarro <fabien.navarro@math.cnrs.fr>

Repository CRAN

Date/Publication 2023-02-11 16:10:02 UTC

R topics documented:

`adjacency_mat` *Compute the adjacency matrix of the gaussian weighted graph*

Description

Compute the adjacency matrix of the gaussian weighted graph

Usage

```
adjacency_mat(
  pts,
  f = function(x) {
     exp(-x^2/8)
 },
  s = 0
)
```

Arguments

pts	coordinates of N points in R^3.
f	is a scalar potential ($\exp(-x^2/2t^2)$ for gaussian potential)
s	is a threhold to sparisfy the matrix

See Also

laplacian_mat, swissroll

Examples

```
pts <- swissroll(N=100, seed=0, a=1, b=4)
W <- adjacency_mat(pts)
```

---

analysis *Analysis operator.*

Description

Compute the analysis operator for coefficient y.

Usage

```
analysis(y, tf)
```

Arguments

y	Graph signal to analyze.
tf	frame coefficients.

Value

coef Transform coefficients.

See Also

synthesis, tight_frame

`betathresh`	*Apply Beta Threshold.*

Description

Apply Beta Threshold.

Usage

```
betathresh(y, t, b)
```

Arguments

`y`	Noisy Data.
`t`	Threshold.
`b`	Thresholding type (b=1: soft, b=2: JS).

Value

`x` Filtered result.

`download_graph`	*Download sparse matrix form the SuiteSparse Matrix Collection.*

Description

If coordinates are associated with the graphs, they are automatically downloaded and added to the output. See https://sparse.tamu.edu/ for the list of groups and graph names.

Usage

```
download_graph(matrixname, groupname)
```

Arguments

`matrixname`	Name of the graph to download.
`groupname`	Name of the group that provides the graph.

Value

`matrixname` a list contening the sparse matrix `sA`, `xy` coordinates (if any), `dim` the number of rows, columns and numerically nonzero elements and `info`, the path to a plain txt file containing information associated with `sA` (accessible for example via `file.show(matrixname$info)`).

References

Davis, T. A., & Hu, Y. (2011). The University of Florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS), 38(1), 1-25.

Examples

```
matrixname <- "grid1"
groupname <- "AG-Monien"
download_graph(matrixname,groupname)
file.show(grid1$info)
```

---

eigendec — *Spectral decomposition of a symetric matrix*

Description

Eigen decomposition of dense symmetric/hermitian matrix M using divide-and-conquer methods that provides slightly different results than the standard method, but is considerably faster for large matrices.

Usage

```
eigendec(M)
```

Arguments

`M`	a matrix.

---

eigensort — *Spectral decomposition of a symetric matrix.*

Description

Computes eigenvalues and eigenvectors of matrices (output sorted in increasing order).

Usage

```
eigensort(x)
```

Arguments

`x`	Symetric matrix (sparse or dense) whose spectral decomposition is to be computed.

Examples

```
A <- matrix(1, ncol=2, nrow=2)
dec <- eigensort(A)
```

forward_sgwt *Compute Forward Spectral Graph Wavelet Transform.*

Description

Compute forward SGWT for signal f (without frame calculation). The calculation corresponds to the frame defined by the 'tight_frame' function (without explicit calculation of the latter).

Usage

```
forward_sgwt(f, evalues, evectors, b = 2)
```

Arguments

`f`	Graph signal to analyze.
`evalues`	Eigenvalues of the Laplacian matrix.
`evectors`	Eigenvectors of the Laplacian matrix.
`b`	Parameter that control the number of scales.

Value

`wc` wavelet coefficients.

References

Göbel, F., Blanchard, G., von Luxburg, U. (2018). Construction of tight frames on graphs and application to denoising. In Handbook of Big Data Analytics (pp. 503-522). Springer, Cham.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.

See Also

`inverse_sgwt`, `tight_frame`

`full` *Convert symmetric sparse matrix to full matrix.*

Description

Convert a symmetric sparse matrix sA to full matrix A.

Usage

```
full(sA)
```

Arguments

`sA` Sparse matrix to convert.

See Also

fullup

fullup

Examples

```
sA <- pittsburgh$sA
A <- full(sA)
```

`fullup` *Convert symmetric sparse matrix to full matrix.*

Description

Convert a symmetric sparse matrix sA stored as upper triangular matrix to full matrix A.

Usage

```
fullup(sA)
```

Arguments

`sA` Sparse upper triangular matrix to convert.

See Also

full

Examples

```
data(grid1)
A <- fullup(grid1$sA)
```

`grid1` *grid1 graph.*

Description

A dataset containing grid1 graph.

Usage

```
grid1
```

Format

list of 2 elements

xy coordinates

sA adjacency matrix

`GVN` *Graph Von Neumann Estimator.*

Description

Graph equivalent of the Von Neummann variance estimator.

Usage

```
GVN(y, A, L)
```

Arguments

`y`	Noisy data.
`A`	Adjacency matrix.
`L`	Laplacian matrix.

References

von Neumann, J. (1941). Distribution of the ratio of the mean square successive difference to the variance. *Ann. Math. Statistics*, 35(3), 433–451.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

See Also

HPFVN

Examples

```
data(minnesota)
A <- minnesota$A
L <- laplacian_mat(A)
x <- minnesota$xy[ ,1]
n <- length(x)
f <- sin(x)
sigma <- 0.1
noise <- rnorm(n, sd = sigma)
y <- f + noise
sigma^2
GVN(y, A, L)
```

HPFVN *High Pass Filter Von Neumann Estimator.*

Description

Graph extension of the Von Neummann variance estimator using finest scale coefficients.

Usage

```
HPFVN(wcn, evalues, b)
```

Arguments

wcn	Noisy wavelet coefficients.
evalues	Laplacian spectrum.
b	Parameter that control the number of scales.

References

von Neumann, J. (1941). Distribution of the ratio of the mean square successive difference to the variance. *Ann. Math. Statistics*, 35(3), 433–451.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

See Also

GVN

Examples

```
data(grid1)
A <- grid1$sA
L <- laplacian_mat(A)
x <- grid1$xy[ ,1]
n <- length(x)
val1 <- eigensort(L)
evalues <- val1$evalues
evectors <- val1$evectors
f <- sin(x)
sigma <- 0.1
noise <- rnorm(n, sd = sigma)
y <- f + noise
b <- 2
wcn <- forward_sgwt(y, evalues, evectors, b=b)
sigma^2
HPFVN(wcn, evalues, b)
```

---

inverse_sgwt *Compute Inverse Spectral Graph Wavelet Transform.*

Description

Compute inverse (adjoint) SGWT for signal f (without frame calculation). The calculation is perform for the frame defined by the 'tight_frame' function. The tightness of the underlying frame implies that the computation is obtained by simply applying the adjoint linear transformation to the wavelet coefficients.

Usage

```
inverse_sgwt(wc, evalues, evectors, b = 2)
```

Arguments

`wc`	Wavelet coefficients.
`evalues`	Eigenvalues of the Laplacian matrix.
`evectors`	Eigenvectors of the Laplacian matrix.
`b`	Parameter that control the number of scales.

Value

`f` SGWT adjoint applied to wc.

References

Göbel, F., Blanchard, G., von Luxburg, U. (2018). Construction of tight frames on graphs and application to denoising. In Handbook of Big Data Analytics (pp. 503-522). Springer, Cham.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.

See Also

`forward_sgwt`, `tight_frame`

`laplacian_mat` *Laplacian matrix.*

Description

Compute the (unormalized) laplacian matrix from the adjacency matrix.

Usage

```
laplacian_mat(W)
```

Arguments

`W`	Adjacency matrix.

Value

`L` (unormalized) Laplacian matrix.

`matmult` *Matrix multiplication*

Description

Matrix multiplication

Usage

```
matmult(A, B)
```

Arguments

`A`	a matrix.
`B`	a matrix.

minnesota — *Minnesota road network.*

Description

A dataset containing the minnesota road network (as well as the two signals).

Usage

```
minnesota
```

Format

list of 6 elements

xy coordinates

A adjacency matrix

sA sparse version of A

f1 eta=0.01 and k=2

f2 eta = 0.001 and k=4

labels labels

Source

D. Gleich. The MatlabBGL Matlab library. https://www.cs.purdue.edu/homes/dgleich/packages/matlab_bgl/index.html.

NYCdata — *NYC network.*

Description

A dataset containing the NYC network.

Usage

```
NYCdata
```

Format

list of 2 elements

A NYC adjacency matrix (built using exponential weights between two nodes).

f median price observed from miles travelled to the given drop off point.

`pittsburgh` *Pittsburgh network.*

Description

A dataset containing the pittsubrgh network.

Usage

```
pittsburgh
```

Format

list of 7 elements

A pittsburgh adjacency matrix

sA pittsburgh sparse adjacency matrix

xy coordinates

f signal used in Trend filtering on graphs

y noisy signal used in Trend filtering on graph

f1 eta=0.01, k=5

geo geometry

Source

The sources come from different codes provided by Yu-Xiang Wang (UC Santa Barbara) and are associated with the article: "Trend Filtering on Graphs, JMLR, 2016". https://sites.cs.ucsb.edu/~yuxiangw/resources.html.

`plot_filter` *Plot tight-frame filters.*

Description

Plot tight-frame kernels/filters.

Usage

```
plot_filter(lmax, b, N = 1000)
```

Arguments

`lmax`	Largest eigenvalues of the Laplacian matrix.
`b`	Parameter that control the number of scales.
`N`	Number of discretization points (by default N=1000).

plot_graph *Graph plot*

Description

Graph plot

Usage

```
plot_graph(z, size = 0.75)
```

Arguments

z	Graph data.
size	Dot size.

See Also

plot_signal

Examples

```
data(grid1)
plot_graph(grid1)
```

plot_signal *Plot a signal on top of a given graph*

Description

Plot a signal on top of a given graph

Usage

```
plot_signal(z, f, size = 0.75, limits = range(f))
```

Arguments

z	Graph data.
f	Signal to plot.
size	Dot size.
limits	Set colormap limits.

See Also

plot_graph

Examples

```
f <- rnorm(length(grid1$xy[,1]))
plot_signal(grid1, f)
```

PSNR *Peak Signal to Noise Ratio.*

Description

Compute the Peack Signal to Noise Ratio, defined by:

$$PSNR(x, y) = 10 \log_{10}(\max(\max(x), \max(y))^2 / |x - y|^2)$$

Usage

```
PSNR(x, y)
```

Arguments

x	Original reference signal/image.
y	Restored or noisy signal/image.

Value

Peak Signal to Noise ratio.

See Also

SNR

Examples

```
x <- cos(seq(0, 10, length=100))
y <- x + rnorm(100, sd=0.5)
PSNR(x, y)
```

`randsignal` *Generate random signal with varying regularity.*

Description

Generate $f = A^k x_\eta / r^k$, with A the adjacency matrix and x_η realization of Bernoulli random variables of parameter η and r the largest eigenvalue (in magnitude). The generation is carried out in sparse matrices in order to scale up.

Usage

```
randsignal(eta, k, A, r)
```

Arguments

`eta`	Smoothness parameter.
`k`	Smoothness parameter.
`A`	Adjacency matrix.
`r`	Optional argument corresponding to the largest eigenvalue of A (in magnitude).

Value

`f` output signal.

`rlogo` *R logo graph.*

Description

A dataset containing a graph based on the R logo.

Usage

```
rlogo
```

Format

list of 2 elements

xy coordinates

sA adjacency matrix

smoothmodulus — *Modulus of smoothness.*

Description

Compute the modulus of smoothness of a graph signal.

Usage

```
smoothmodulus(f, A)
```

Arguments

f	Signal.
A	Adjacency matrix (sparse or dense).

Examples

```
data(grid1)
A <- grid1$A
x <- grid1$xy[ ,1]
f <- sin(x)
smoothmodulus(f, A)
```

SNR — *Signal to Noise Ratio.*

Description

Signal to Noise Ratio.

Usage

```
SNR(x, y)
```

Arguments

x	Original reference signal.
y	Restored or noisy signal.

Value

Signal to Noise ratio.

See Also

PSNR

Examples

```
x <- cos(seq(0, 10, length=100))
y <- x + rnorm(100, sd=0.5)
SNR(x, y)
```

SUREthresh *Stein's Unbiased Risk Estimate.*

Description

Adaptive Threshold Selection Using Principle of SURE (The irreductible variance term is not included, it does not change the position of the minimum).

Usage

```
SUREthresh(wcn, tresh, diagWWt, b, sigma, hatsigma, policy, keepwc = TRUE)
```

Arguments

wcn	Noisy wavelet coefficents.
tresh	Threshold values.
diagWWt	Weights.
b	Thresholding type (b=1: soft, b=2: JS).
sigma	Sd of the noise.
hatsigma	Estimator of the sd (if any).
policy	Dependent or uniform.
keepwc	Boolean allowing to export the coefficients of the frame after thresholding (TRUE by default).

Value

`res` a dataframe contening SURE, hatSURE and their respective minima.

References

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

See Also

SURE_MSEthresh

SURE_MSEthresh *Stein's Unbiased Risk Estimate.*

Description

Adaptive Threshold Selection Using Principle of SURE (The irreductible variance term is not included, it does not change the position of the minimum).

Usage

```
SURE_MSEthresh(
  wcn,
  wcf,
  tresh,
  diagWWt,
  b,
  sigma,
  hatsigma,
  policy,
  keepwc = TRUE
)
```

Arguments

`wcn`	Noisy wavelet coefficents.
`wcf`	True wavelet coefficients.
`tresh`	Threshold values.
`diagWWt`	Weights.
`b`	Thresholding type (b=1: soft, b=2: JS).
`sigma`	Sd of the noise.
`hatsigma`	Estimator of the sd (if any).
`policy`	Dependent or uniform.
`keepwc`	Boolean allowing to export the coefficients of the frame after thresholding (TRUE by default).

Details

Note: - the calculation of the MSE is also included for comparison purpose.

Value

`res` a dataframe contening MSE, SURE, hatSURE and their respective minima

References

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

See Also

`SUREthresh`

`swissroll` *Swiss roll graph generation*

Description

Map the square $[0, 1]^2$ in swiss roll for all x, y in $[0, 1]^2$, set

$$Sx = \pi\sqrt{(b^2 - a^2)x + a^2)}$$

$$Sy = \pi(b^2 - a^2)y/2$$

Usage

```
swissroll(N = 500, seed = NULL, a = 1, b = 4)
```

Arguments

`N`	Number of points drawn.
`seed`	Optionally specify a RNG seed (for reproducible experiments).
`a, b`	Shape parameters.

Value

N x 3 array for 3d points.

See Also

`adjacency_mat`

Examples

```
pts <- swissroll(N=500, seed=0, a=1, b=4)
```

`synthesis` *Synthesis operator.*

Description

Compute the synthesis operator for coefficient coeff.

Usage

```
synthesis(coeff, tf)
```

Arguments

`coeff`	Transform coefficients.
`tf`	Frame coefficients.

Value

`y` Synthesis signal.

See Also

`analysis`, `tight_frame`

`tight_frame` *Tight-frame computation.*

Description

Constructs tight-frame.

Usage

```
tight_frame(evalues, evectors, b = 2)
```

Arguments

`evalues`	Eigenvalues of the Laplacian matrix.
`evectors`	Eigenvectors of the Laplacian matrix.
`b`	Parameter that control the number of scales.

References

Coulhon, T., Kerkyacharian, G., & Petrushev, P. (2012). Heat kernel generated frames in the setting of Dirichlet spaces. Journal of Fourier Analysis and Applications, 18(5), 995-1066.

Göbel, F., Blanchard, G., von Luxburg, U. (2018). Construction of tight frames on graphs and application to denoising. In Handbook of Big Data Analytics (pp. 503-522). Springer, Cham.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

`zetav` *Evaluates tight-frame kernel functions*

Description

Evaluates kernel associated with the particular (Littlewood-Paley type) tight-frame construction, based on a unity partition, proposed in the reference papers.

Usage

```
zetav(x, k, b)
```

Arguments

`x`	Support to evaluate the kernel (vector).
`k`	Scale index (scalar).
`b`	Parameter that control the number of scales (scalar).

References

Coulhon, T., Kerkyacharian, G., & Petrushev, P. (2012). Heat kernel generated frames in the setting of Dirichlet spaces. Journal of Fourier Analysis and Applications, 18(5), 995-1066.

Göbel, F., Blanchard, G., von Luxburg, U. (2018). Construction of tight frames on graphs and application to denoising. In Handbook of Big Data Analytics (pp. 503-522). Springer, Cham.

de Loynes, B., Navarro, F., Olivier, B. (2021). Data-driven thresholding in denoising with Spectral Graph Wavelet Transform. Journal of Computational and Applied Mathematics, Vol. 389.

Index