Package ‘LaplaceDeconv’

January 27, 2016

Type Package

Title Laplace Deconvolution with Noisy Discrete Non-Equally Spaced Observations on a Finite Time Interval

Version 1.0.4

Date 2015-10-29

Author Yves Rozenholc (University Paris Descartes), Marianna Pensky (University of Central Florida)

Maintainer Yves Rozenholc <yves.rozenholc@parisdescartes.fr>

Description Solves the problem of Laplace deconvolution with noisy discrete non-equally spaced observations on a finite time interval based on expansions of the convolution kernel, the unknown function and the observed signal over Laguerre functions basis. It implements the methodology proposed in the paper ``Laplace deconvolution on the basis of time domain data and its application to Dynamic Contrast Enhanced imaging'' by F. Comte, C-A. Cuenod, M. Pensky and Y. Rozenholc in ArXiv (http://arxiv.org/abs/1405.7107).

License MIT + file LICENSE

Depends parallel, polynom, orthopolynom

Encoding UTF-8

RoxygenNote 5.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-27 20:39:49

R topics documented:

BuildLaguerreSystem 2
EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda 3
LagLaplDeconv 4
LaguerreLaplaceConvolution 6
LaplaceConvolution 7
MakeLaguerreMatrix 8

Index **10**

BuildLaguerreSystem *function BuildLaguerreSystem*

Description

build the matrix associated to the Laguerre function basis at given times, its QR decomposition, the coefficients of the kernel function of the Laplace convolution and the Toeplitz matrix associated to the Laplace convolution with the kernel g.

Usage

```
BuildLaguerreSystem(a, g, times, Mmax, tol = 1e-07)
```

Arguments

`a,`	numeric, the scale parameter of the Laguerre functions basis
`g,`	numeric vector, the observed values of the known Laplace convolution kernel at the observation times
`times,`	numeric vector, the observation times
`Mmax,`	numeric, the maximal degree
`tol,`	tolerance in qr decomposition (default 1e-7)

Value

a list containing:

- `gcoef`, numeric vector, the coefficients of g in the Laguerre functions basis
- `GM`, numeric matrix, the Toeplitz matrix of the Laplace convolution with g
- `Phi`, numeric matrix, the Laguerre function basis
- `Phi.qr`, the QR decomposition of Phi
- `Mmax`, numeric, the maximal degree
- `a`, numeric, the scale of the Laguerre function basis
- `g`, numeric vector, the kernel
- `times`, numeric vector, the observation times

Author(s)

Y. Rozenholc and M. Pensky

Examples

```
## Not run:

 library(LaplaceDeconv)
 t = seq(0,10,l=50)
 g = exp(-5*t)
 # compute the elements needed for the Laplace deconvolution with kernel observations g at times t
 bFctLagMat = BuildLaguerreSystem(a=1/2,g,t,M=10)

## End(Not run)
```

`EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda` *Data from two Dynamical Contrast Enhanced Computer Tomography (DCE-CT) image sequences realized one patient of the cohort REMISCAN before treatment starts (t0) and after 15 days of treatment (t1). These are the real data used to illustrate the paper in References.*

Description

Each dataset contains a variable `ex_dcemri` which is a list containing #'

- `AIF`, numeric vector, the averaged enhancement curve recorded in the aorta, the convolution kernel
- `TUM_1`, `TUM_2`, `TUM_3`, numeric vectors, three averaged enhancement curves recorded in homogenous regions in the tumor, the observations
- `times`, numeric vector, the observation times
- `sigma`, numeric, the noise level
- `delta`, numeric, equals 0, a time shift assumed to be less than 1 time unit

Usage

```
data(EX_DCEMRI_t0)
```

Author(s)

C-A. Cuenod and Y. Rozenholc

References

"Laplace deconvolution on the basis of time domain data and its application to Dynamic Contrast Enhanced imaging" by F. Comte, C-A. Cuenod, M. Pensky, Y. Rozenholc (ArXiv http://arxiv.org/abs/1405.7107)

REMISCAN - Project number IDRCB 2007-A00518-45/P060407/STIC 2006; Research Ethics Board (REB) approved- cohort funding by INCa (1M Euros) and promoted by the AP-HP (Assistance Publique Hôpitaux de Paris). Inclusion target: 100 patients. Start in 2007. Closed since July 2015.

LagLaplDeconv *function LagLaplDeconv*

Description

Main function of this package : computes the Laplace deconvolution with noisy discrete non-equally spaced observations on a finite time interval (see reference).

Usage

```
LagLaplDeconv(Y, g, times = 1:length(Y), sigma, cpen = 2, atab = -1,
  Mmax = 25, ncores.max = Inf, verbose = FALSE, withplot = FALSE)
```

Arguments

`Y`	numeric vector, the observed noisy observations of the Laplace convolution
`g`	numeric vector, the known kernel of the Laplace convolution
`times`	numeric vector, the observation times (default `1:length(Y)`)
`sigma`	numeric, the noise level
`cpen`	numeric, the penalization constant (default 2)
`atab`	numeric vector, an array of value for a (default -1, see Details)
`Mmax`	integer, the maximum degree (default 25)
`ncores.max`	numeric, number of used cores (default `Inf`)
`verbose`	boolean to control information output (default `FALSE`)
`withplot`	boolean to control plot output (default `FALSE`)

Details

`atab` defines the values of the scale parameter used to build the Laguerre functions basis. If `atab` is of length 1 and negative then `atab<-seq(0.4,3,by=0.05/abs(atab))/sqrt(Tmax/10)` where `Tmax=max(times)`.

Value

a list containing:

- `f.hat`, numeric vector, the estimate at observation times
- `q.hat`, numeric vector, the reconstructed convolution of the kernel g and the estimate f.hat, computed at observation times
- `f.coef`, numeric vector, the coefficients of f.hat in the selected Laguerre function basis
- `info`, only for internal use — not documented
- `a.hat`, numeric, the selected value of the parameter a
- `M0`, numeric, the dimension of the selected model
- `sigma`, numeric, the noise level
- `cpen`, numeric, the penalization constant used in the penalty

Author(s)

Y. Rozenholc and M. Pensky

References

Laplace deconvolution on the basis of time domain data and its application to Dynamic Contrast Enhanced imaging by F. Comte, C-A. Cuenod, M. Pensky, Y. Rozenholc (ArXiv http://arxiv.org/abs/1405.7107)

Examples

```
## Not run: 
 #### AN ARTICIAL EXAMPLE ####

 library(LaplaceDeconv)
 par(mfrow=c(1,1))
 set.seed(29102015)

 sigma=0.02
 a = 1
 t = seq(0,5,l=100)
 g = 20*t^2*exp(-5*t)
 f.coef = c(0.4,0.02,0.01)

 # compute the Laplace convolution from g, kernel computed at times t, and the function
 # described by its decomposition in Laguerre function basis with scale a :
 fg = LaguerreLaplaceConvolution(t,g,f.coef,a)

 # the noisy observations :
 Y = fg+sigma*rnorm(length(fg))

 # estimation of f from the observation and the kernel :
 L = LagLaplDeconv(Y,g,t,sigma)
 matplot(t,cbind(g,MakeLaguerreMatrix(a,3)(t)%*%f.coef,fg,L$q.hat,L$f.hat,Y),lty=1,
   type=c('b',rep('l',4),'p'),ylab='',pch='x')

 # display results of estimation
 legend('topright',lty=c(rep(1,5),0),pch=c('x',rep('',4),'x'),
   legend=c(
     'g: partially observed kernel',
     'f: unknown',
     'q=fxg: unknown convolution',
     expression(hat(q)*': plug-in convolution'),
     expression(hat(f)*': estimation of f'),
     'Y: observations'),
   col=1:6)

## End(Not run)

 ## Not run: 
 #### A REAL EXAMPLE USING DCE-MRI DATA FROM A TUMOR ####

 library(LaplaceDeconv)
```

```
par(mfrow=c(1,2))

# load data from patient before the treatment
data(EX_DCEMRI_t0)

# display AIF and tumoral enhancements
matplot(ex_dcemri$times,
  cbind(ex_dcemri$AIF,ex_dcemri$TUM_1,ex_dcemri$TUM_2,ex_dcemri$TUM_3),
  ylab='',lty=1,type=c('b',rep('p',3)),pch='+',main='Observations')
legend('topright',pch='+',legend=c('AIF','TUM_1','TUM_2','TUM_3'),col=1:4)

# estimation of the contrast agent survival functions
L1 = LagLaplDeconv(ex_dcemri$TUM_1,ex_dcemri$AIF,ex_dcemri$times,ex_dcemri$sigma)
L2 = LagLaplDeconv(ex_dcemri$TUM_2,ex_dcemri$AIF,ex_dcemri$times,ex_dcemri$sigma)
L3 = LagLaplDeconv(ex_dcemri$TUM_3,ex_dcemri$AIF,ex_dcemri$times,ex_dcemri$sigma)

matlines(ex_dcemri$times,cbind(L1$q.hat,L2$q.hat,L3$q.hat),type='l',lty=1,col=2:4)

# display results of estimation
matplot(ex_dcemri$times,cbind(L1$f.hat,L2$f.hat,L3$f.hat),type='l',lty=1,col=2:4,
  ylab='survival',main='Contrast agent survival fcts')
legend('topright',lty=1,col=2:4,
  legend=c(
    paste0('TUM_1 - a.hat=',round(L1$a.hat,digits=2)),
    paste0('TUM_2 - a.hat=',round(L2$a.hat,digits=2)),
    paste0('TUM_3 - a.hat=',round(L3$a.hat,digits=2))
    )
  )

## End(Not run)
```

LaguerreLaplaceConvolution

function LaguerreLaplaceConvolution

Description

computes the Laplace convolution of two functions f and g observed at discrete times t. Use trapezoidal formula and an expansion of f in the Laguerre function basis.

Usage

```
LaguerreLaplaceConvolution(t, g, f.coef, a)
```

Arguments

`t,`	numeric vector, the observation times
`g,`	numeric vector, the observed values of the known Laplace convolution kernel at the observation times

`f.coef,` numeric vector, the coefficients in the Laguerre function basis of the function f to convole with g

`a,` numeric, the scale of the Laguerre functions basis

Value

return the Laplace convolution of f and g using Trapezoidal formula and expansion of f in the Laguerre function basis

Author(s)

Y. Rozenholc and M. Pensky

Examples

```
## Not run:

 library(LaplaceDeconv)
 a = 1/2
 t = seq(0,10,l=100)
 g = exp(-5*t)
 f.coef = c(1,0.25,0.1)
 # compute the Laplace convolution from g, kernel computed at times t, and the function described by
 # its decomposition in Laguerre function basis with scale a
 fg = LaguerreLaplaceConvolution(t,g,f.coef,a)
 matplot(t,cbind(MakeLaguerreMatrix(a,3)(t)%*%f.coef,g,fg),lty=1,type='l',ylab='')
 legend('topright',lty=1,legend=c('f','g','fxg'),col=1:3)

## End(Not run)
```

`LaplaceConvolution` *function LaplaceConvolution*

Description

computes the Laplace convolution of two functions f and g observed at discrete times t. Use trapezoidal formula and spline approximation of f.

Usage

```
LaplaceConvolution(t, g, f)
```

Arguments

`t,` numeric vector, the observation times

`g,` numeric vector, the observed values of the known Laplace convolution kernel at the observation times

`f,` numeric vector, the coefficients the values of the function f to convole with g

Value

return the Laplace convolution of f and g using Trapezoidal formula and spline approximation for F

Author(s)

Y. Rozenholc and M. Pensky

Examples

```
## Not run:

 library(LaplaceDeconv)
 t = seq(0,10,l=100)
 g = exp(-5*t)
 f = t^2*exp(-t)
 # compute the Laplace convolution from functions computed at times t : f and g
 fg = LaplaceConvolution(t,g,f)
 matplot(t,cbind(f,g,fg),lty=1,type='l')
 legend('topright',lty=1,legend=c('f','g','fxg'),col=1:3)

## End(Not run)
```

`MakeLaguerreMatrix` *function MakeLaguerreMatrix*

Description

return a constructor of the Laguerre function basis.

Usage

```
MakeLaguerreMatrix(a = 1/2, M)
```

Arguments

`a,`	numeric, the scale of the Laguerre functions basis
`M,`	numeric, the maximal degree

Value

a constructor of the M-first Laguerre functions with scale a

Author(s)

Y. Rozenholc and M. Pensky

Examples

```
## Not run:

 library(LaplaceDeconv)
 # build the constructor
 bFctLagMat = MakeLaguerreMatrix(M=10)
 # compute the Laguerre function basis at times seq(0,15,l=100)
 bFctLagMat(seq(0,15,l=100))

## End(Not run)
```

Index

∗Topic **data**
EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda, 3

AIF, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3

BuildLaguerreSystem, 2

delta, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3

ex_dcemri (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3
EX_DCEMRI_t0, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3
EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda, 3
EX_DCEMRI_t1, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3

LagLaplaceDeconv, (LagLaplDeconv), 4
LagLaplaceDeconvolution, (LagLaplDeconv), 4
LagLaplDeconv, 4
LaguerreLaplaceConvolution, 6
LaguerreLaplaceDeconvolution, (LagLaplDeconv), 4
LaguerrePenalizedQR (LagLaplDeconv), 4
LaplaceConvolution, 7
LaplaceDeconv, (LagLaplDeconv), 4
LaplaceDeconvolution, (LagLaplDeconv), 4
LaplDeconv, (LagLaplDeconv), 4

MakeLaguerreMatrix, 8

sigma, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3

times, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3

TUM_1, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3
TUM_2, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3
TUM_3, (EX_DCEMRI_t0.rda, EX_DCEMRI_t1.rda), 3