Package ‘tvdiff’

December 31, 2020

Type Package

Title Numerical differentiation of noisy, nonsmooth data

Version 0.0.9000

Description Numerical differentiation of noisy, nonsmooth data based on
Total Variation Regularized Numerical Differentiation algorithm by Rick Chartrand.

License CC0

Encoding UTF-8

LazyData true

Suggests knitr,
rmarkdown,
testthat,
covr

VignetteBuilder knitr

RoxygenNote 7.1.1

Imports Matrix,
Rcpp

Depends R (>= 2.10)

LinkingTo Rcpp,
RcppArmadillo

SystemRequirements C++11

R topics documented:

`largedemodata` *Large demo dataset*

Description

"Data obtained from a whole-room calorimeter, courtesy of Edward L. Melanson of the University of Colorado Denver. The metabolic rate of a subject within the calorimeter can be determined via respirometry, the measurement of oxygen consumption, and carbon dioxide production within the room." "The data consists of samples taken every second for most of a day, for a total of 82,799 samples."

Usage

```
largedemodata
```

Format

A data frame with 82,799 rows and 2 variables:

x seconds

obs observed measurement

Source

https://sites.google.com/site/dnartrahckcir/home/tvdiff-code

References

Rick Chartrand, "Numerical differentiation of noisy, nonsmooth data," ISRN Applied Mathematics, Vol. 2011, Article ID 164564, 2011

`pcgsolve` *Preconditioned conjugate gradient method*

Description

Preconditioned conjugate gradient method for solving system of linear equations Ax = b, where A is symmetric and positive definite.

Usage

```
pcgsolve(b, M, alph, L, dx, tol = 1e-06, maxIter = 1000L)
```

Arguments

`b`	vector, with same dimension as number of rows of A.
`M`	matrix, preconditioner matrix defined interal to `TVRegDiffR`.
`alph`	numeric, regularization parameter used in `TVRegDiffR`.
`L`	matrix, linearized diffusion matrix internal to `TVRegDiffR`.
`dx`	numeric, grid spacing used in `TVRegDiffR`.
`tol`	numeric, threshold for convergence, default is `1e-6`.
`maxIter`	numeric, maximum iteration, default is `1000`.

Details

Code is a slightly modified version of `pcgsolve`

Value

A vector representing solution x.

`smalldemodata` *Small demo dataset*

Description

A test dataset based on the function $f(x) = |x - 0.5|$ with Gaussian noise with a standard deviation of 0.05.

Usage

```
smalldemodata
```

Format

A data frame with 100 rows and 3 variables:

x `seq(0,1,length.out = 100)`

true `abs(x -0.5)`

obs `abs(x -0.5) + rnorm(length(x),mean = 0 ,sd = 0.05)`

Source

https://sites.google.com/site/dnartrahckcir/home/tvdiff-code

References

Rick Chartrand, "Numerical differentiation of noisy, nonsmooth data," ISRN Applied Mathematics, Vol. 2011, Article ID 164564, 2011

`toyproblem` *Toy problem*

Description

A test dataset based on the piecewise function $f(x) = (x + 2)^2$ when $x < 0$ and $f(x) = (x - 2)^2$ when $x > 0$ with Gaussian noise with a standard deviation of 0.5.

Usage

```
toyproblem
```

Format

A data frame with 100 rows and 4 variables:

x `seq(-4,4,length.out = 100)`

y_true true value

y_obs observation with noise

dydx_true true derivative

`TVRegDiffR` *Total Variation Regularized Numerical Differentiation (TVDiff)*

Description

Estimate the derivative of noisy data using total variation regularized differentiation.

Usage

```
TVRegDiffR(
  data,
  iter,
  alph,
  u0 = NULL,
  scale = "small",
  ep = 1e-06,
  dx = 1/length(data),
  plotflag = 0,
  tol = 1e-06,
  maxit = 1000
)
```

Arguments

`data`	Vector of data to be differentiated.
`iter`	Number of iterations to run the main loop. A stopping condition based on the norm of the gradient vector g below would be an easy modification. No default value.
`alph`	Regularization parameter. This is the main parameter to fiddle with. Start by varying by orders of magnitude until reasonable results are obtained. A value to the nearest power of 10 is usally adequate. No default value. Higher values increase regularization strenght and improve conditioning.
`u0`	Initialization of the iteration. Default value is the naive derivative (without scaling), of appropriate length (this being different for the two methods). Although the solution is theoretically independent of the intialization, a poor choice can exacerbate conditioning issues when the linear system is solved.
`scale`	'large' or 'small' (case insensitive). Default is 'small'. 'small' has somewhat better boundary behavior, but becomes unwieldly for data larger than 1000 entries or so. 'large' has simpler numerics but is more efficient for large-scale problems. 'large' is more readily modified for higher-order derivatives, since the implicit differentiation matrix is square.

`ep` Parameter for avoiding division by zero. Default value is 1e-6. Results should not be very sensitive to the value. Larger values improve conditioning and therefore speed, while smaller values give more accurate results with sharper jumps.

`dx` Grid spacing, used in the definition of the derivative operators. Default is the reciprocal of the data size.

`plotflag` Flag whether to display plot at each iteration. Default is 0 (no). Useful, but adds significant running time.

`tol` R Version Only: Tolerance passed to preconditiond conjugate gradient solver

`maxit` R Version Only: Maximum iterations passed to preconditiond conjugate gradient solver

Details

C++ code for preconditioned conjugate gradient method adapted from `cPCG::pcgsolve`

Value

Estimate of the regularized derivative of data. Due to different grid assumptions, length(u) = length(data) + 1 if scale = 'small', otherwise length(u) = length(data).

Author(s)

R translation: Nathaniel Price (<`natbprice@gmail.com`>)

Original Matlab Code: Rick Chartrand (<`rickc@lanl.gov`>)

References

Rick Chartrand, "Numerical differentiation of noisy, nonsmooth data," ISRN Applied Mathematics, Vol. 2011, Article ID 164564, 2011.

Examples

```
# Load small demo data
data("smalldemodata")

# Unpack data
x <- smalldemodata$x
obs <- smalldemodata$obs
true <- smalldemodata$true
dydx_true <- rep(-1, length(x))
dydx_true[x > 0.5] <- 1
dx <- x[2] - x[1]

# Extimate derivative
dydx <- TVRegDiffR(
  data = obs,
  iter = 100,
  alph = 0.2,
  scale = "small",
  ep = 1e-6,
  dx = dx
)
dydx <- dydx[-1]
```

Index