Package 'spedecon'

May 9, 2026

Type Package

Title Smoothness-Penalized Deconvolution for Density Estimation Under Measurement Error

Version 0.1

Date 2024-01-11

URL https://www.davidjkent.org

Description Implements the Smoothness-Penalized Deconvolution method for estimating a probability density under measurement error of Kent and Rupert (2023) <doi:10.1080/01621459.2023.2259028>. The estimator is formed by computing a histogram of the error-contaminated data, and then finding an estimate that minimizes a reconstruction error plus a smoothness-inducing penalty term. The primary function, sped(), takes the data and error distribution, and returns the estimator as a function.

License GPL-3

Imports quadprog, splines, methods, stats, graphics

NeedsCompilation no

Encoding UTF-8

RoxygenNote 7.3.0

Author David Kent [aut, cre] (ORCID: <https://orcid.org/0000-0003-2880-1216>)

Maintainer David Kent <dk657@cornell.edu>

Repository CRAN

Date/Publication 2024-01-12 10:00:02 UTC

Contents

`compute_ephemera` *Pre-computations for sped*

Description

`compute_ephemera()` does data-independent pre-computations for sped and can speed up repeated applications

Usage

```
compute_ephemera(gtwid, hn, padding, spline_dim, perknot = 2)
```

Arguments

`gtwid`	Object of class spedecon_gtwid describing the density of Z in the model $Y = X + Z$
`hn`	Object of class `histogram` holding any histogram with the desired bins. The bins must be equally-spaced, i.e. `hn$equidist` must be `TRUE`, but otherwise only `hn$breaks` and `hn$mids` are used.
`padding`	Support of spline space is extended by `padding/2` beyond the data on each side
`spline_dim`	Numeric integer, dimension of spline space
`perknot`	Number of positivity constraints per knot

Details

The computations in sped rely on several matrices and vectors that are determined by the error density, spline space, and histogram bins, but do not depend on the data. Computing these is the most time-intensive element of the process, so if the estimator will be applied several times to different data, but the same error density, spline space, and histogram bins (likely in simulations), gains can be had by pre-computing those matrices and vectors just one time.

For comparison, the sped function internally uses `padding = 0.4`, and `perknot = 2`.

Value

Object of class `spedecon_ephemera`, a list containing the pre-computed values.

References

Kent D, Ruppert D (2023). "Smoothness-Penalized Deconvolution (SPeD) of a Density Estimate." *Journal of the American Statistical Association*, to appear. ISSN 0162-1459, doi:10.1080/01621459.2023.2259028

Examples

```
alpha <- 1e-3; n <- 1e3; s <- 0.3
Y <- rgamma(n,5,2) + rnorm(n,0,s)
gtwid <- gaussian_gtwid(sd=s)
hn <- hist(Y,breaks="FD",plot=FALSE)
ephemera <- compute_ephemera(gtwid=gtwid,hn=hn,padding=0.4,spline_dim=30,perknot=2)
sol1 <- sped(Y,gtwid,1e-3,ephemera=ephemera) # fast
sol2 <- sped(Y,gtwid,1e-3) # slow
attr(sol1,"coef") - attr(sol2,"coef")
```

---

gaussian_gtwid *Fourier transform of Gaussian density*

Description

Returns a spedecon_gtwid object representing the Fourier transform of a mean-zero Gaussian density

Usage

```
gaussian_gtwid(sd)
```

Arguments

sd	Standard deviation

Value

Object of class spedecon_gtwid

Examples

```
gtwid <- gaussian_gtwid(sd = 1)
```

---

laplace_gtwid *Fourier transform of Laplace density*

Description

Returns a spedecon_gtwid object representing the Fourier transform of a mean-zero Laplace density with scale b

Usage

```
laplace_gtwid(b)
```

Arguments

b	Scale parameter

Value

Object of class spedecon_gtwid

Examples

```
gtwid <- laplace_gtwid(b = 1)
```

---

new_spedecon_gtwid *Creates object of class* spedecon_gtwid

Description

Constructor for class spedecon_gtwid. Use helper functions gaussian_gtwid(), laplace_gtwid(), and uniform_gtwid() instead whenever possible.

Usage

```
new_spedecon_gtwid(gtwid, family)
```

Arguments

gtwid	Function representing the Fourier transform
family	List with at least one entry family[["family"]] naming the family of distributions, and possibly other entries stating the values of the parameters in that family.

Details

The spedecon_gtwid class is meant to represent the Fourier transform of a probability density. The basic type is a function. It also has a family attribute which can hold the name and parameters of the family of distributions.

Value

Object of class spedecon_gtwid

See Also

Use gaussian_gtwid(), laplace_gtwid(), or uniform_gtwid() instead whenever possible.

new_spedecon_spline_sped_fit *Creates object of class* spedecon_spline_sped_fit

Description

Internal use only. Constructor for class spedecon_spline_sped_fit

Usage

```
new_spedecon_spline_sped_fit(coef, basis, alpha, constraint)
```

Arguments

coef	Numeric vector; the coefficients of the spline.
basis	Object of class spedecon_spline_basis, representing a basis for the spline space.
alpha	Positive numeric, the alpha that was used for the fit.
constraint	The type of constraint used.

Details

The basic type of an object of type spedecon_spline_sped_fit is a function; one can therefore evaluate, plot, etc. and ignore the other attributes if desired. The function is represented as a spline, and has attributes coef and basis, which represents the coefficients and basis respectively. coef is a numeric vector, while basis is an object of class spedecon_spline_basis, which is essentially just a list holding the knots and order of the spline space. A spedecon_spline_sped_fit object also has attributes alpha and constraint which record the penalty parameter and constraint method used for the fit.

Value

Object of class spedecon_spline_sped_fit

sped *Smoothness-Penalized Deconvolution*

Description

sped() computes the Smoothness-Penalized Deconvolution estimate on the provided data and error distribution

Usage

```
sped(
  Y,
  gtwid,
  alpha,
  constraint = "constrainedQP",
  spline_dim = 30,
  hn = NULL,
  ephemera = NULL,
  ...
)
```

Arguments

`Y`	Numeric vector of data from the model $Y = X + Z$
`gtwid`	Object of class spedecon_gtwid describing the density of Z in the model $Y = X + Z$. It should almost always be created by one of the helper functions `gaussian_gtwid()`, `laplace_gtwid()`, or `uniform_gtwid()`.
`alpha`	Positive numeric penalty parameter
`constraint`	String, controls whether and how the solution is constrained to be a pdf. One of `"constrainedQP"`, `"projection"`, or `"unconstrained"` for constrained quadratic program, metric projection, or unconstrained, respectively
`spline_dim`	Numeric integer, dimension of spline space
`hn`	(optional) Object of class `histogram` holding pre-computed histogram computed from the data Y
`ephemera`	(optional) Object of class `spedecon_ephemera` holding pre-computed computational bits
`...`	(optional) Other arguments

Details

This function computes the "Smoothness-Penalized Deconvolution" (SPeD) estimate of a density under additive measurement error. The essential inputs to the function are the data `Y`, the Fourier transform `gtwid` of the error density, and the penalty parameter `alpha`; more details follow here, but for a full description of the estimator please consult Kent and Ruppert (2023).

The data model is that we observe an iid sample distributed like $Y = X + Z$, with Z an error independent of X. We wish to estimate the density $f(x)$ of X. It is assumed that we know the probability density of the errors Z, call it $g(z)$.

The estimator begins with a density estimate $h_n(y)$ of the density of Y, and minimizes the objective function

$$\|g * v - h_n\|^2 + \alpha \|v^{(2)}\|^2$$

in v, with v ranging over a space of cubic splines with equally-spaced knots; the dimension of this space can be adjusted with the argument `spline_dim`. The SPeD estimate is not naturally a pdf, so it must be constrained. When `constraint = "constrainedQP"`, the constraint is imposed directly into quadratic program minimizing the objective; when `constraint = "projection"`, the

unconstrained estimate is computed and then projected onto the space of pdfs. The preliminary density estimate h_n is computed internally as a histogram using Freedman-Diaconis choice of bin width, but a user-supplied histogram computed with `hist()` may be provided via the `hn` argument.

The computations require the *Fourier transform* $\tilde{g}(t)$ of the probability density, and this must be supplied as an object of type `spedecon_gtwid`, which can be produced for common error densities using the helper functions `gaussian_gtwid()`, `laplace_gtwid()`, and `uniform_gtwid()`.

If the estimator will be re-computed many times for many realizations of data, substantial time can be saved by pre-computing all the auxiliary matrices and vectors one time, and supplying them through the `ephemera` argument. This can be done whenever the repeated computations all use the same error density, same histogram bins, and same spline space, as those are what define the required matrices and vectors. A helper function `compute_ephemera()` is provided to pre-compute these.

Value

Object of class spedecon_spline_sped_fit

References

Kent D, Ruppert D (2023). "Smoothness-Penalized Deconvolution (SPeD) of a Density Estimate." *Journal of the American Statistical Association*, to appear. ISSN 0162-1459, doi:10.1080/01621459.2023.2259028

Examples

```
alpha <- 1e-3
n <- 1e3; s <- 0.3
Y <- rgamma(n,5,2) + rnorm(n,0,s) # Data, contaminated with Gaussian errors
sol <- sped(Y,gtwid=gaussian_gtwid(sd=s),1e-3)
plot(sol,n=1e3) # Plot the resulting estimate
curve(dgamma(x,5,2),col=2,n=1e3,add=TRUE) # The target density f() of X
sol(c(2,3,4)) # We can evaluate sol; it is a function
```

---

uniform_gtwid *Fourier transform of Uniform density*

Description

Returns a `spedecon_gtwid` object representing the Fourier transform of a Uniform[-a,a] density

Usage

```
uniform_gtwid(a)
```

Arguments

`a`	Half-width

Value

Object of class spedecon_gtwid

Examples

```
gtwid <- uniform_gtwid(a = 1)
```

Index