Package 'slp'

October 14, 2022

Version 1.0-5

Author Wesley Burr, with contributions from Karim Rahim

Copyright file COPYRIGHTS

Maintainer Wesley Burr <wesley.burr@gmail.com>

Title Discrete Prolate Spheroidal (Slepian) Sequence Regression Smoothers

Description Interface for creation of 'slp' class smoother objects for use in Generalized Additive Models (as implemented by packages 'gam' and 'mgcv').

Depends R (>= 2.15.1)

License GPL (>= 2)

Imports mgcv (>= 1.7.18)

Suggests gam (>= 1.09)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-08-28 18:54:58

R topics documented:

slp-package *slp: Discrete Prolate Spheroidal (Slepian) Sequence Regression Smoothers*

Description

Interface for creation of 'slp' class smoother objects for use in Generalized Additive Models as implemented by packages gam and mgcv.

Details

Package:	slp
Type:	Package
Version:	1.0-5
Date:	2016-08-28
License:	GPL-2

Author(s)

Wesley Burr, with contributions from Karim Rahim

Maintainer: Wesley Burr <wesley.burr@gmail.com>

References

Thomson, D.J. (1982) Spectrum estimation and harmonic analysis. *Proceedings of the IEEE* Volume **70**, number 9, pp. 1055–1096.

Thomson, D.J. (2001) Inverse Constrained Projection Filters. *Proc. SPIE 4478*, Wavelets: Applications in Signal and Image Processing IX, 172 (December 5, 2001); doi:10.1117/12.449708

Hastie, T. J. (1991) *Generalized additive models.* Chapter 7 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth \& Brooks/Cole.

Hastie, T. and Tibshirani, R. (1990) *Generalized Additive Models.* London: Chapman and Hall.

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73(1):3-36

Wood, S.N. (2004) Stable and efficient multiple smoothing parameter estimation for generalized additive models. J. Amer. Statist. Ass. 99:673-686.

Wood, S.N. (2006) *Generalized Additive Models: an introduction with R*, CRC

.dpss *Compute Discrete Prolate Spheroidal Sequences*

Description

Compute Discrete Prolate Spheroidal (Slepian) Sequences for use as time-based smoother. This approach uses the tridiagonal method and exploits symmetry. Note the odd order tapers are normalized so that the slope at the centre is positive, in accordance with Slepian (1978) and Thomson (1982). This differs from Percival and Walden (1993). This code follows Chapter 8.3 of Percival and Walden (1993) using LAPACK function calls, Anderson (1999).

Usage

```
.dpss(n,k,nw)
```

Arguments

n	A positive integer, typically the length of the time series-regression data set.
k	A positive integer, the number of basis vectors for the smoother, often 2*nw - 2.
nw	A positive double-precision number, the time-bandwidth parameter. The frequency domain analogue of the maximum period of interest.

Value

v	A 'n' by 'k' matrix of basis vectors of class 'dpss'. Each column is the appropriate Slepian sequence of order 'k-1'.

References

Anderson, E. (1999). *LAPACK Users' guide (Vol. 9).* SIAM.

Percival, D.B. and Walden, A.T. (1993) *Spectral analysis for physical applications.* Cambridge University Press.

Slepian, D. (1978) Prolate spheroidal wave functions, Fourier analysis, and uncertainty. V–The discrete case. *Bell System Technical Journal* Volume **57**, pp. 1371–1430

Thomson, D.J (1982) Spectrum estimation and harmonic analysis. *Proceedings of the IEEE* Volume **70**, number 9, pp. 1055–1096.

basis *slp: Pre-computed basis sets*

Description

Shared name for a number of pre-computed basis sets included with the `slp` package. For large `N`, significant speed-ups can be obtained by pre-computing the basis set and simply loading it from disk.

Format

Each file named `basis_N_XXX_W_X_K_XX.RData`, with the `X` entries integers indicating the `N`, `W` (in `df/year`) and `K` parameters, can be loaded via `data(...)` as a `basis` object in the environment of your choice.

Currently, these pre-computed bases are used as speed-up aids within `{slp(...)}` and `{s(..., bs='slp', ...)}`. A full list of available bases can be obtained by examining `slpSavedObjects`, an additional `data(...)` object included with the package.

checkSaved *Determine if combination of N, W and K is available on disk, or must be computed*

Description

Given values for `N`, `W` and `K`, determine if the combination is available on disk as a `data()` load. If so, significant time savings can be obtained, especially for large `N`.

Usage

```
checkSaved(N, W, K)
```

Arguments

`N`	the length of the time index array, in units of `deltat`.
`W`	the time bandwidth, **in units of** `df/year`. This is **not** the same units as a `slp(...)` call, and `W` should be an integer, e.g., 6 `df/year`.
`K`	the number of basis vectors requested.

Details

Does a lookup against `data(slpSavedObjects)` to determine whether the combination of `N`, `W` and `K` is saved on disk as part of the package, and can be loaded. It is possible to create your own basis objects for particular choices of `N`, `W` and `K` and save them as part of the library directory, updating `slpSavedObjects` as you do so.

Value

Logical (TRUE or FALSE), indicating availability or lack thereof.

Examples

```
# Examples using pkg:gam
library("slp")
checkSaved(N = 730, W = 6, K = 24)
checkSaved(N = 365, W = 6, K = 13)
```

`Predict.matrix.slp.smooth` *Create model matrix for prediction, for model using slp smoother*

Description

Re-generate the basis matrix for a particular `N, W` Slepian sequence family member, with the additional property that the smoother captures/passes constants without distortion. Simply re-arranges `object`. Not intended to be used directly by user.

Usage

```
## S3 method for class 'slp.smooth'
Predict.matrix(object,data)
```

Arguments

`object`	a smooth specification object, usually generated by a model term `s(..., bs = 'slp', ..., xt = list(...))`, and for this type, **requiring** an additional `xt = list()` object containing parameters. For examples, see below.
`data`	a list containing just the data required by this term, with names corresponding to `object[['term']]`. Typically just a single time index array.

Details

Presumably because most basis sets are larger in size than their computational burden, `mgcv` passes objects around without including the actual basis vectors. For example, if using basis `cr`, the parameters are included in `object`, and then the bases re-computed as needed.

As the `slp` basis is significantly more computational in nature, the basis vectors are saved as part of the object. While `mgcv` deletes the main set of time-aligned vectors, this routine restores such vectors so that `predict` and `plot` work correctly.

Value

A corrected (re-assembled) version of `object`, which contains the `X` basis vectors in a format that can be used by `predict` or `plot`.

See Also

smooth.construct, Predict.matrix

---

slp.gam *Generate a Basis Matrix for Discrete Prolate Spheroidal (Slepian) Sequences*

Description

Generate the basis matrix for a particular N, W Slepian sequence family member, with the additional property that the smoother passes constants without distortion. Can be quite slow execution due to the latter property.

Based on ns for implementation with gam.

Parallel implementation for mgcv included in package as slp.mgcv.

Usage

```
slp(x, W = NA, K = NA, deltat = 1, naive = FALSE, intercept = FALSE,
    customSVD = TRUE, forceC = FALSE, returnS = FALSE)
```

Arguments

x	the predictor variable. Missing values are allowed. Assumed to be contiguous; if not, then converted to a contiguous series to determine appropriate N, K and W, then the basis vectors are back-converted at the termination of the routine. Should be in units of deltat.
W	the time bandwidth. Computed as the frequency domain analogue of the maximum period of interest for a time series-regression problem using "smooth functions of time". For example, a period choice of 2 months converts to 60 days and W = 1/60 cycles per day. Alternatively, if the interest is in a period of 7 cycles per year, then W = 7 / 365.2425 = 0.0192 cycles per day.
K	the number of basis vectors requested. If not provided, then W must be, and K is set to approximately floor(2 * N * W - 1). This parameter is approximately equivalent to df for ns with fixed dimension. Note: if you specify K higher than 2 * N * W + 1 performance will suffer significantly. The actual number of basis vectors returned is K-1 for the case of intercept = FALSE, and K for intercept = TRUE.
deltat	the time step for the input x. Restricted to 1 and 6 days for ease of logic checking, as these are the most traditional choices. Assumes that W is in the same units, and has no real impact beyond this, so it is trivial to make deltat symbolically equal an arbitrary choice and convert W to match.
naive	a flag for returning the naive (default) Slepian basis vectors v (TRUE) rather than the mean-passing SLP2 or SLP3 variants (FALSE).
intercept	a flag for choosing between a SLP2 or SLP3 basis. Type-2 bases capture (absorb) means of target series, while Type-3 bases ignore (pass) means.

customSVD — a flag for using the built-in svd (case FALSE) or a modified version of DGESDD LAPACK 3.5.0. The modified version provides significant speed improvements as it skips a number of unnecessary steps for the particular edge case needed by slp.

forceC — a flag for forced computation of the basis vectors. Several combinations of commonly used N, W, K parameters have been pre-computed and included with the package. If this parameter is set to TRUE, the routine will compute the basis vectors regardless of whether they are available in pre-computed form. See checkSaved{checkSaved} for further details.

returnS — a flag for returning the projection matrix S rather than the basis vectors that form the same. Intended to be used outside of model-fitting environments.

Details

slp is based around the routine .dpss, which generates a family of Discrete Prolate Spheroidal (Slepian) Sequences. These vectors are orthonormal, have alternating even/odd parity, and form the optimally concentrated basis set for the subspace of R^N corresponding to the bandwidth W. Full details are given in Slepian (1978). These basis functions have natural boundary conditions, and lack any form of knot structure. This version is returned for naive = TRUE.

The dpss basis vectors can be adapted to provide the additional useful property of capturing or passing constants perfectly. That is, the smoother matrix S formed from the returned rectangular matrix will either reproduce constants at near round-off precision, i.e., S %*% rep(1, N) = rep(1, N), for naive = FALSE with intercept = TRUE, or will pass constants, i.e., S %*% rep(1, N) = rep(0, N), for naive = FALSE with intercept = FALSE.

The primary use is in modeling formula to directly specify a Slepian time-based smoothing term in a model: see the examples.

For large N this routine can be **very** slow. If you are computing models with large N, we highly recommend pre-computing the basis object, then using it in your models without recomputation. The third example below demonstrates this approach.

Value

A matrix of dimension length(x) * K or length(x) * (K-1) where either K was supplied, or W was supplied and K converted. Note that the basis vectors are computed on a contiguous grid based on x, and then back-converted to the time structure of x.

Attributes are returned that correspond to the arguments to ns, and explicitly give K, W, etc.

References

Thomson, D.J (1982) Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*. Volume **70**, number 9, pp. 1055-1096.

Slepian, David (1978) Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty V: the Discrete Case. *Bell System Technical Journal*. Volume **57**, pp. 1371-1429.

See Also

ns

Examples

```
# Examples using pkg:gam
library("gam")
library("slp")
N <- 730
W <- 14 / N
K <- 28         # will actually use 27 df when intercept = FALSE
x <- rnorm(n = N, sd = 1)
y <- x + rnorm(n = N, sd = 2) + 5.0
t <- seq(1, N)

# note: all three examples share identical results

# example with in-call computation, using K (df)
fit1 <- gam(y ~ x + slp(t, K = K, forceC = TRUE), family = gaussian)

# example with in-call computation, using W
fit2 <- gam(y ~ x + slp(t, W = W, forceC = TRUE), family = gaussian)

# example with out-of-call computation, using K
timeBasis <- slp(t, K = K, forceC = TRUE)
fit3 <- gam(y ~ x + timeBasis, family = gaussian)

# the same computations can be done using pre-computed basis vectors
# for significant speed-ups, especially for large N - see `checkSaved'
# for more details
fit4 <- gam(y ~ x + slp(t, W = W, forceC = FALSE))
```

`slp.mgcv` *Generate a Basis Matrix for Discrete Prolate Spheroidal (Slepian) Sequences*

Description

Generate the basis matrix for a particular `N`, `W` Slepian sequence family member, with the additional property that the smoother captures/passes constants without distortion. Can be quite slow in execution due to the latter property.

Based on `smooth.construct.cr.smooth.spec` for implementation with `mgcv`.

Usage

```
## S3 method for class 'slp.smooth.spec'
smooth.construct(object,data,knots)
```

Arguments

`object`	a smooth specification object, usually generated by a model term `s(..., bs = 'slp', ..., xt = list(...))`, and for this type, **requiring** an additional `xt = list()` object containing parameters. For examples, see below.

data	a list containing just the data required by this term, with names corresponding to object[['term']]. Typically just a single time index array.
knots	a list containing any knots supplied for basis setup – should be NULL, as slp basis objects are not generated from knots.

Details

slp is based on .dpss, which generates a family of Discrete Prolate Spheroidal (Slepian) Sequences. These vectors are orthonormal, have alternating even/odd parity, and form the optimally concentrated basis set for the subspace of R^N corresponding to the bandwidth W. Full details are given in Slepian (1978). These basis functions have natural boundary conditions, and lack any form of knot structure. This version is returned for naive = TRUE.

The dpss basis vectors can be adapted to provide the additional useful property of capturing or passing constants perfectly. That is, the smoother matrix S formed from the returned rectangular matrix will either reproduce constants at near round-off precision, i.e., S %*% rep(1, N) = rep(1, N), for naive = FALSE with intercept = TRUE, or will pass constants, i.e., S %*% rep(1, N) = rep(0, N), for naive = FALSE with intercept = FALSE.

The primary use is in modeling formula to directly specify a Slepian time-based smoothing term in a model: see the examples.

For large N this routine can be **very** slow. If you are computing models with large N, we highly recommend pre-computing the basis object, then using it in your models without recomputation. The third example below demonstrates this approach.

Value

An object of class slp.smooth. In addition to the usual elements of a smooth class (see smooth.construct), this object will contain:

C	a constraint matrix which restricts mgcv from modifying the (already) orthogonal and mean-passing/capturing basis vectors.
K	the user-specified number of basis vectors, or the computed K from user-supplied W.
W	the user-specified bandwidth W, or the computed W from user-supplied K.
fullBasis	the full-span computed, normalized basis set, before contiguity is taken into account. Used by predict when given an object of type slp.smooth.
contiguous	a logical variable declaring whether or not the input time array was considered to be contiguous by the basis computation procedure.
wx	the "corrected" input time array; if contiguous == FALSE then this will be the same as data[object[['term']]].

References

Wood S.N. (2006) Generalized Additive Models: An Introduction with R. Chapman and Hall/CRC Press.

Hastie T.J. & Tibshirani, R.J. (1990) Generalized Additive Models. Chapman and Hall.

Thomson, D.J (1982) Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*. Volume **70**, number 9, pp. 1055-1096.

Slepian, David (1978) Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty V: the Discrete Case. *Bell System Technical Journal*. Volume **57**, pp. 1371-1429.

See Also

`smooth.construct`

Examples

```
# Examples using pkg:mgcv
library("mgcv")
library("slp")

N <- 730
W <- 8 / N
K <- 16         # will actually use 15 df as intercept = FALSE
x <- rnorm(n = N, sd = 1)
y <- x + rnorm(n = N, sd = 2) + 5.0
t <- seq(1, N)

# note: all three examples share identical results

# example with in-call computation, using K (df)
fit1 <- gam(y ~ x + s(t, bs = 'slp', xt = list(K = K)), family = gaussian)

# example with in-call computation, using W
fit2 <- gam(y ~ x + s(t, bs = 'slp', xt = list(W = W)), family = gaussian)
```

`slpSavedObjects` *slp: Listing of available pre-computed basis sets*

Description

List of pre-computed basis sets included with the `slp` package. For large `N`, significant speed-ups can be obtained by pre-computing the basis set and simply loading it from disk.

Usage

```
slpSavedObjects
```

Format

A list containing the `N`, `W` (in `df/year`) and `K` of the pre-computed basis sets. Each combination is included as a `data(...)` loadable `.RData` file.

Index