Package 'stcos'

April 25, 2019

Type Package

Title Space-Time Change of Support

Version 0.1.0

Date 2019-04-23

Author Andrew M. Raim [aut, cre],
Scott H. Holan [aut, res],
Jonathan R. Bradley [aut, res],
Christopher K. Wikle [aut, res]

Maintainer Andrew M. Raim <andrew.raim@gmail.com>

URL https://github.com/holans/ST-COS

Description
Spatio-temporal change of support (STCOS) methods are designed for statistical inference on geographic and/or time domains that differ from those on which the data were observed. 'stcos' implements a parsimonious class of Bayesian hierarchical spatio-temporal models for STCOS with Gaussian outcomes introduced by Bradley, Wikle, and Holan <doi:10.1002/sta4.94>.

License GPL (>= 2)

Imports R6

Depends R (>= 3.3), Rcpp, Matrix, sf

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 6.1.1

Encoding UTF-8

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-04-25 16:00:03 UTC

R topics documented:

`acs_sf` *Shapes and ACS estimates for Boone County, MO.*

Description

An `sf` object with ACS estimates for:

- Boone County, Missouri
- Table B19013
- Block group level geography
- Years 2013 - 2017

Usage

```
acs5_2013

acs5_2014

acs5_2015

acs5_2016

acs5_2017
```

Format

`sf` objects.

Details

Shapefiles were gathered via the `tigris` package, and ACS estimates were downloaded from the American FactFinder `http://factfinder.census.gov`. Data was assembled on 2/28/2019. See `data-prep-aff.R` in the Columbia example code for details.

columbia_neighbs *City of Columbia neighborhoods.*

Description

An sf object containing the geometry of four neighborhoods in the City of Columbia, Boone County, Missouri. Based on shapefiles provided by the Office of Information Technology / GIS, City of Columbia, Missouri.

Usage

```
columbia_neighbs
```

Format

An sf object with 4 features (neighborhoods).

DIC *Deviance Information Criterion*

Description

A function to compute the Deviance Information Criterion (DIC) on an stcos object.

Usage

```
DIC(object)
```

Arguments

object	A result from the Gibbs sampler.

Value

DIC computed from saved draws

See Also

gibbs.stcos gibbs.stcos.raw

Examples

```
## Not run:
out1 <- gibbs.stcos(sp, R = 10000, burn = 0, thin = 1)
DIC(out1)

## End(Not run)
```

gibbs *Gibbs Sampler for STCOS Model*

Description

Run the Gibbs sampling algorithm for the STCOS model. `gibbs.stcos` presents a simplified interface, while `gibbs.stcos.raw` allows all inputs to be specified separately.

Usage

```
gibbs.stcos(prep, R, report.period = R + 1, burn = 0, thin = 1,
  hyper = NULL, init = NULL)

gibbs.stcos.raw(z, v, H, S, K.inv, R, report.period = R + 1, burn = 0,
  thin = 1, init = NULL, fixed = NULL, hyper = NULL)
```

Arguments

`prep`	An `STCOSprep` object.
`R`	Number of MCMC reps.
`report.period`	Gibbs sampler will report progress each time this many iterations are completed.
`burn`	Burn this many of `R` the draws, before saving history.
`thin`	After burn-ikn period, save one out of every `thin` draws.
`hyper`	A list containing the following hyperparameter values: `a.sig2mu`, `a.sig2K`, `a.sig2xi`, `b.sig2mu`, `b.sig2K`, `b.sig2xi`. Any hyperparameters which are not specified are set to a default value of 2.
`init`	A list containing the following initial values for the MCMC: `sig2mu`, `sig2xi`, `sig2K`, `mu_B`, `eta`, `xi`. Any values which are not specified are set to arbitrary choices.
`z`	Vector which represents the outcome; assumed to be direct estimates from the survey.
`v`	Vector which represents direct variance estimates from the survey.
`H`	Matrix of overlaps between source and fine-level supports.
`S`	Design matrix for basis decomposition.
`K.inv`	Inverse of the K matrix, which is the covariance of the random coefficient η
`fixed`	A list specifying which parameters to keep fixed in the MCMC. This can normally be left blank. If elements `sig2mu`, `sig2xi`, or `sig2K` are specified they should be boolean, where TRUE means fixed (i.e. not drawn). If elements `mu_B`, `eta`, or `xi` are specified, they should each be a vector of indicies; the specified indices are to be treated as fixed (i.e. not drawn).

Value

An `stcos` object which contains draws from the sampler.

Examples

```
## Not run:
basis <- SpaceTimeBisquareBasis$new(x, y, t, w.s, w.t)
sp <- STCOSPrep$new(fine_domain = dom.fine,
    fine_domain_geo_name = "GEO_ID",
    basis = basis, basis_mc_reps = 500)
out1 <- gibbs.stcos(sp, R = 10000, burn = 0, thin = 1)

out2 <- gibbs.stcos.raw(z = sp$get_z(), S = sp$get_reduced_S(),
    v = sp$get_v(), K.inv = sp$get_Kinv(), H = sp$get_H(),
    R = 10000, burn = 0, thin = 1)

## End(Not run)
```

`licols` *licols*

Description

An R version of a Matlab `licols` function given in https://www.mathworks.com/matlabcentral/answers/108835-how-to-get-only-linearly-independent-rows-in-a-matrix-or-to-remove-linear-dependenc
Extract a linearly independent set of columns of a given matrix `X`.

Usage

```
licols(X, tol = 1e-10, quiet = FALSE)
```

Arguments

`X`	A matrix.
`tol`	A tolerance for rank estimation. Default is 1e-10.
`quiet`	A boolean; if FALSE, print a warning about computation time if `X` is large.

Value

`Xsub` contains the extracted columns of `X` and `idx` contains the indices (into X) of those columns. The elapsed time is stored in `elapsed.sec`.

Examples

```
x <- sample(1:3, size = 20, replace = TRUE)
Z <- model.matrix(~ as.factor(x) - 1)
X <- cbind(1, Z)
licols(X)
```

`mle.stcos` *MLE for STCOS Model*

Description

MLE for STCOS Model

Usage

```
mle.stcos(z, v, H, S, K.inv, init = NULL, optim.control = list())
```

Arguments

`z`	Vector which represents the outcome; assumed to be direct estimates from the survey.
`v`	Vector which represents direct variance estimates from the survey.
`H`	Matrix of overlaps between source and fine-level supports.
`S`	Design matrix for basis decomposition.
`K.inv`	Inverse of the K matrix, which is the covariance of the random coefficient η
`init`	A list containing the following initial values for the MCMC: `sig2xi`. If not specified, we select an arbitrary initial value.
`optim.control`	This is passed as the `control` argument to `optim`. Note that the value `fnscale` is ignored if specified.

Value

A list containing maximum likelihood estimates.

Examples

```
## Not run:
z <- sp$get_z()
v <- sp$get_v()
H <- sp$get_H()
S.reduced <- sp$get_reduced_S()
K.inv <- sp$get_Kinv(2005:2015)

mle.out <- mle.stcos(z, v, S.reduced, H, K.inv)

sig2K.hat <- mle.out$sig2K.hat
sig2xi.hat <- mle.out$sig2xi.hat
mu.hat -> mle.out$mu.hat

## End(Not run)
```

SpaceTimeBisquareBasis

SpaceTime Bisquare Basis

Description

An R6Class representing the space-time bisquare basis.

Usage

```
basis <- SpaceTimeBisquareBasis$new(knots.x, knots.y, knots.t,
    w.s, w.t)
basis$compute(x, y, time)
basis$get_dim()
basis$get_cutpoints()
basis$get_rl()
basis$get_ws()
basis$get_wt()
```

Arguments

- `knots.x` x-coordinate of knot points.
- `knots.y` y-coordinate of knot points.
- `knots.t` time coordinate of knot points.
- `w.s` (Original, before transformation) spatial radius.
- `w.t` Temporal radius.
- `x` Vector of x-coordinates for points on which to evaluate the basis.
- `y` Vector of y-coordinates for points on which to evaluate the basis.
- `time` Vector of time coordinates for points on which to evaluate the basis.

Methods

- `new` Create a new `SpaceTimeBisquareBasis` object.
- `get_dim` Get the number of cutpoints used to construct this basis.
- `get_cutpoints` Get the cutpoints used to construct this basis.
- `get_rl` Get the transformed spatial radius. The transformation is based on quantiles of distances between knots.
- `get_ws` Get the original spatial radius used to construct this basis. This is transformed before it is applied to account for the geography being used.
- `get_wt` Get the temporal radius used to construct this basis.
- `compute` Evaluate this basis on specific points.

Examples

```
set.seed(1234)
seq.x <- seq(0, 1, length.out = 3)
seq.y <- seq(0, 1, length.out = 3)
seq.t <- seq(0, 1, length.out = 3)
knots = expand.grid(seq.x, seq.y, seq.t)
x <- runif(50)
y <- runif(50)
t <- sample(1:3, size = 50, replace = TRUE)

basis <- SpaceTimeBisquareBasis$new(knots[,1], knots[,2], knots[,3], w.s = 0.5, w.t = 1)
basis$compute(x, y, t)
basis$get_dim()
basis$get_cutpoints()
basis$get_rl()
basis$get_ws()
basis$get_wt()

# Plot the (spatial) knots and the points at which we evaluated the basis
plot(knots[,1], knots[,2], pch = 4, cex = 1.5, col = "red")
text(x, y, labels = t, cex = 0.75)

# Draw a circle representing the basis' radius around one of the knot points
tseq <- seq(0, 2*pi, length=100)
rad <- basis$get_rl()
coords <- cbind(rad * cos(tseq) + seq.x[2], rad * sin(tseq) + seq.y[2])
lines(coords, col = "red")
```

SpatialBisquareBasis *Spacial Bisquare Basis*

Description

An R6Class representing the spacial bisquare basis.

Usage

```
basis <- SpaceTimeBisquareBasis$new(knots.x, knots.y, w)
basis$compute(x, y, time)
basis$get_dim()
basis$get_cutpoints()
basis$get_rl()
basis$get_w()
```

Arguments

- `knots.x` x-coordinate of knot points.
- `knots.y` y-coordinate of knot points.

- w (Original, before transformation) radius.
- x Vector of x-coordinates for points on which to evaluate the basis.
- y Vector of y-coordinates for points on which to evaluate the basis.

Methods

- new Create a new SpatialBisquareBasis object.
- get_dim Get the number of cutpoints used to construct this basis.
- get_cutpoints Get the cutpoints used to construct this basis.
- get_rl Get the transformed spatial radius. The transformation is based on quantiles of distances between knots.
- get_w Get the original radius used to construct this basis. This is transformed before it is applied to account for the geography being used.
- compute Evaluate this basis on specific points.

Examples

```
set.seed(1234)
seq.x <- seq(0, 1, length.out = 3)
seq.y <- seq(0, 1, length.out = 3)
knots = merge(seq.x, seq.y)
x <- runif(50)
y <- runif(50)

basis <- SpatialBisquareBasis$new(knots[,1], knots[,2], w = 0.5)
basis$compute(x, y)
basis$get_dim()
basis$get_cutpoints()
basis$get_rl()
basis$get_w()

# Plot the knots and the points at which we evaluated the basis
plot(knots[,1], knots[,2], pch = 4, cex = 1.5, col = "red")
points(x, y, cex = 0.5)

# Draw a circle representing the basis' radius around one of the knot points
tseq <- seq(0, 2*pi, length=100)
rad <- basis$get_rl()
coords <- cbind(rad * cos(tseq) + seq.x[2], rad * sin(tseq) + seq.y[2])
lines(coords, col = "red")
```

---

stcos *stcos: Space-Time Change of Support*

Description

An R Package for Space-Time Change of Support (STCOS) modeling.

References

Jonathan R. Bradley, Christopher K. Wikle, and Scott H. Holan (2015). Spatio-temporal change of support with application to American Community Survey multi-year period estimates. STAT 4 pp.255-270. https://doi.org/10.1002/sta4.94.

Andrew M. Raim, Scott H. Holan, Jonathan R. Bradley, and Christopher K. Wikle (2017). A model selection study for spatio-temporal change of support. In JSM Proceedings, Government Statistics Section. Alexandria, VA: American Statistical Association, pp.1524-1540.

`STCOSPrep` *STCOS Preparation*

Description

An `R6Class` for preparing an STCOS analysis.

Usage

```
sp <- STCOSPrep$new(fine_domain, fine_domain_geo_name, basis, basis_mc_reps = 500,
                    report_period = 100)

sp$add_obs(domain, period, estimate_name, variance_name, geo_name)

S <- sp$get_S()

sp$set_basis_reduction(f)

sp$get_reduced_S()

sp$get_Kinv(times, X = NULL, autoreg = TRUE)

sp$get_obs(idx)

sp$get_basis()

sp$get_geo()
```

Arguments

- `fine_domain` An `sf` object; the fine-level support for the analysis.
- `fine_domain_geo_name` The name of the field in `fine_domain` which represents a unique geographical ID.
- `basis` An object of type `SpaceTimeBisquareBasis`.
- `basis_mc_reps` Number of Monte Carlo reps to use when computing area-level basis function decomposition.

- report_period Gibbs sampler will report progress each time this many iterations are completed.
- domain An sf object; a source support for the analysis.
- period A vector of times from which the estimates were computed. For example, to indicate 2015 ACS 5-year estimates, use period = 2011:2015.
- estimate_name Name of the field which contains direct estimates.
- variance_name Name of the field which contains direct variance estimates.
- geo_name Name of the fiend which represents unique geographical ID.
- idx A vector of indices.
- f A function which performs a dimension reduction.
- times A vector of times relevant to the analysis.
- X A fixed covariate, if one is available.
- autoreg A boolean; if TRUE, assume an autoregressive covariance structure in time. Otherwise assume independence between times.

Methods

- $new Create a new STCOSPrep object, which does not yet contain any source supports.
- $add_obs Add a source support.
- $get_obs Get a source support(s) which have already been added.
- $domain2model Compute H and S matrix fragments for a given source support.
- $get_z Get vector of direct estimates from added source supports.
- $get_v Get vector of direct variance estimates from added source supports.
- $get_H Get matrix of overlaps between fine-level support and added source supports.
- $get_S Get design matrix based on basis function decomposition, based on added source supports.
- $get_reduced_S Same as get_S, except first apply the dimension reduction function (which is the identity function by default).
- $get_geo Get vector of GEO IDs from added source supports.
- $set_basis_reduction Set the dimension reduction function to be applied to S and related matrices.
- $get_basis Get the basis function which has been used to construct this object.
- $get_Kinv Compute the K.inv matrix.

Examples

```
## Not run:
sp <- STCOSPrep$new(dom.fine, "GEO_ID", basis, 500)

# Add source support data
sp$add_obs(acs1.2015, 2015, "DirectEst", "DirectVar", "GEO_ID")
...
sp$add_obs(acs5.2015, 2011:2015, "DirectEst", "DirectVar", "GEO_ID")
```

```
# Reduce dimension of the design matrix with basis expansion
S <- sp$get_S()
eig <- eigen(t(S) %*% S)
idx.S <- which(cumsum(eig$values) / sum(eig$values) < 0.75)
Tx.S <- t(eig$vectors[idx.S,])
f <- function(S) { S %*% Tx.S }
sp$set_basis_reduction(f)

# Get quantities needed for MCMC
S.reduced <- sp$get_reduced_S()
z <- sp$get_z()
v <- sp$get_v()
H <- sp$get_H()

# compute K.inv matrix using Random-Walk method
K.inv <- sp$get_Kinv(2011:2015)

# Retrieve some of the source supports
sp$get_obs(idx = c(2,4,7))

# Return the basis function object from which sp was constructed
sp$get_basis()

# Return the GEO IDs which have been processed so far
sp$get_geo()

## End(Not run)
```

Index