Package: gmrf (via r-universe)

September 14, 2024

Title Lightweight helper for fitting GMRF models in mgcv

Version 0.0.1

Description A set of functions to help with the fitting of gmrf models in mgcv. The package is intended for the fitting GMRF models such as random walks and seasonal effects. The code is written as simply as possible to make is easy to see how GMRFs are constructed.

Depends R (>= 3.1.0), mgcv (>= 1.8-3), Matrix (>= 1.1-4)

Imports spdep (>= 0.5-77), MASS (>= 7.3-35)

Suggests knitr (>= 1.8)

License MIT + file LICENSE

LazyData true

Repository https://faskally.r-universe.dev

RemoteUrl https://github.com/faskally/gmrf

RemoteRef HEAD

RemoteSha 8167605af9a2b7c4ca8843d8c9c77bf7f8e0a4d0

Contents

`Dar` *Differences defining a cyclic auto regressive process or equivalently, an AR with toroidal edge effects.*

Description

Details

Usage

```
Dar(n, phi)
```

Arguments

`n`	the size of the GMRF
`phi`	the autoregressive parameters

Details

Description - This function does stuff.

Value

a Matrix

`Dharm` *Differences defining a cyclic nth order random walk*

Description

Details

Usage

```
Dharm(n, wavelength = n, cyclic = FALSE)
```

Arguments

`n`	the size of the GMRF
`wavelength`	the wavelength of the harmonic. default is for this this to be n, i.e. one cycle.
`cyclic`	logical: should the smoother be cyclic, i.e. corrected for toroidal edge effects.

Details

Description - This function does stuff.

Value

a Matrix

`Dnb` *Differences defining a regional model*

Description

Details

Usage

```
Dnb(nb)
```

Arguments

`nb`	the neighbourhood structure of a regional layout

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each connection / edge

Drw — *Differences defining a cyclic nth order random walk*

Description

Details

Usage

```
Drw(n, order = 2, cyclic = FALSE)
```

Arguments

n	the size of the GMRF.
order	the order of the random walk.
cyclic	logical: should the differences be cyclic, i.e. corrected for toroidal edge effects.

Details

Description - This function does stuff.

Value

a Matrix

Drw1 — *Differences defining a 1st order random walk*

Description

Details

Usage

```
Drw1(n, cyclic = FALSE)
```

Arguments

n	the size of the GMRF.
cyclic	logical: should the differences be cyclic, i.e. corrected for toroidal edge effects .

Details

Description - This function does stuff.

Value

a Matrix

Dseasonal *Differences defining a seasonal model*

Description

This model treats the sum over m time points to be stationary

Usage

```
Dseasonal(n, m)
```

Arguments

n	the size of the GMRF
m	the length of the seasonal period

Details

Description - This function does stuff.

Value

a Matrix

getCnb *Get a constraint matrix for a regional GMRF*

Description

Details. Each group of regions sums to zero.

Usage

```
getCnb(Q)
```

Arguments

Q	a precision matrix for a regional GMRF

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each distinct group

getFactorsnb *Get a constraint matrix for a regional GMRF*

Description

Details. Each group of regions sums to zero.

Usage

```
getFactorsnb(Q)
```

Arguments

Q	a precision matrix for a regional GMRF

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each distinct group

getQar *Differences defining a cyclic auto regressive process or equivalently, an AR with toroidal edge effects.*

Description

Details

Usage

```
getQar(n, phi, weights = NULL)
```

Arguments

n	the size of the GMRF
phi	the autoregressive parameters
weights	weights to be applied to the node differences in effect allowing different variances at each time step.

Details

Description - This function does stuff.

Value

a Matrix

`getQnb` *Differences defining a regional model*

Description

Details

Usage

```
getQnb(nb, weights = NULL)
```

Arguments

`nb`	the neighbourhood structure of a regional layout
`weights`	weights to be applied to the node differences in effect allowing different connections between regions to vary differently. An example of a weight could be the length of the shared border between regions

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each connection / edge

---

`getQpoly` *Differences defining a regional model from spatial polygons...*

Description

Details

Usage

```
getQpoly(poly, weights = NULL)
```

Arguments

`poly`	a spatial polygon
`weights`	weights to be applied to the node differences in effect allowing different connections between regions to vary differently. An example of a weight could be the length of the shared border between regions

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each connection / edge

---

`getQrw` *Compute RWn precision matrix*

Description

Details

Usage

```
getQrw(n, order = 2, weights = NULL, cyclic = FALSE)
```

Arguments

`n`	the size of the GMRF
`order`	the order of the random walk
`weights`	weights to be applied to the node differences (see details)
`cyclic`	logical: should the smoother be cyclic, i.e. corrected for toroidal edge effects.

Details

Description - This function does stuff.

Value

what does it return

Examples

```
require(gmrf)
n <- 100
idx <- 1:n
idy <- c(5:40, 60:n)
set.seed(64)
Q <- getQrw(n, order = 2)
x <- simQ(exp(1) * Q)
# simulate the first 3/4, say
y <- x + rnorm(n) * 3.5
y <- y[idy]
## set up variables for smoothing
rownames(Q) <- colnames(Q) <- idx
## fit an RW2 smoother with restricted df
```

```
g1 <- gam(y ~ s(idy, bs = "gmrf", xt = list(penalty = Q), k = length(y)-1), method="REML")
summary(g1)
plot(idy, y, xlim = range(idx), ylim = range(x,y))
lines(idx, x, col = "blue", lwd = 2)
pred <- predict(g1, newdata = list(idy = idx), se = TRUE)
lines(idx, pred$fit, col = "red", lwd = 2)
lines(idx, pred$fit + 2*pred$se.fit, col = "red", lty = 2)
lines(idx, pred$fit - 2*pred$se.fit, col = "red", lty = 2)
```

`getQrw1` *Compute RW1 precision matrix*

Description

If weights are supplied

Usage

```
getQrw1(n, weights = NULL, cyclic = FALSE)
```

Arguments

`n`	the size of the GMRF
`weights`	weights to be applied to the node differences (see details)
`cyclic`	logical: should the smoother be cyclic, i.e. corrected for toroidal edge effects.

Details

Description - This function does stuff.

Value

what does it return

`getRegionalGMRF` *Differences defining a regional model*

Description

This one is for back compatability

Usage

```
getRegionalGMRF(nbmat, weights = NULL)
```

Arguments

nbmat	the neighbourhood structure of a regional layout as a matrix
weights	weights to be applied to the node differences in effect allowing different connections between regions to vary differently. An example of a weight could be the length of the shared border between regions

Details

Description - This function does stuff.

Value

a Matrix with a column for each region and a row for each connection / edge

---

gmrf *Lightweight add-on for mgcv to allow slighly easier use of GMRF smoothing models*

Description

Lightweight add-on for mgcv to allow slighly easier use of GMRF smoothing models

---

Predict.matrix.gmrf.smooth *Predict from a gmrf smoother*

Description

This function gives predictions from a gmrf smoother

Usage

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

Arguments

object	an object of class '"gmrf.smooth"' produced by the 'smooth.construct' method.
data	a list containing just the data (including any 'by' variable) required by this term, with names corresponding to 'object$term' (and 'object$by'). The 'by' variable is the last element.

Value

A design matrix

simeQ *Simulate from an inhomogenous GMRF*

Description

Required the eigen decomposition of a precision matrix

Usage

```
simeQ(eQ, tol = 1e-09, rank = NULL, k = 1)
```

Arguments

eQ	an eigen decomposition of a symmetric positive semi-definate matrix corresponding to the precision matrix of an inhomogenous GMRF
tol	tolerance (relative to largest eigen value) for numerical lack of positive-definiteness in 'Q'.
rank	the rank of the precision matrix. If this is not supplied it it is estimated by comparing the eigen values to tol * largest eigenvalue.
k	optional scaling of the precision matrix. This can be used to save recomputing the eigen decomposition for different smoothing parameters.

Details

Note if rank is suplied and is less than that true rank, the simulation will be from a reduced rank GMRF.

Value

a single draw from with the appropriate conditional covariance structure

Examples

```
## create a rw2 GMRF precision matrix and simulate
Q <- getQrw(100, order = 2)
eQ <- eigen(Q)
x <- simeQ(eQ, k = exp(5))
plot(x, type = "l")
```

simQ *Simulate from an inhomogenous GMRF*

Description

Details

Usage

```
simQ(Q, tol = 1e-09, rank = NULL)
```

Arguments

Q	a symmetric positive semi-definate matrix corresponding to the precision matrix of an inhomogenous GMRF
tol	tolerance (relative to largest eigen value) for numerical lack of positive-definiteness in 'Q'.
rank	the rank of the precision matrix. If this is not supplied it it is estimated by comparing the eigen values to tol * largest eigenvalue

Details

This function does stuff.

Value

a single draw from with the appropriate covariance structure

Examples

```
## create a rw2 GMRF precision matrix and simulate
Q <- getQrw(100, order = 2)
x <- simQ(Q)
plot(x)
##
## simulate an AR1 GMRF with toroidal edge correction
Q <- getQar(100, phi = 0.8)
x <- simQ(exp(5)*Q)
plot(x, type = "l")
```

`smooth.construct.gmrf.smooth.spec` *Construct a smoother using a GMRF smoothing prior*

Description

This function is used internally in mgcv when fitting a smoother of type gmrf. More details to be added.

Usage

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

Arguments

`object`	a smooth specification object, usually generated by a term 's(...,bs="gmrf", penalty = list(Q = ...))'. 'x' is a factor variable giving labels for the nodes in the GMRF graph, and the 'xt' argument is obligatory: see details.
`data`	a list containing just the data (including any 'by' variable) required by this term, with names corresponding to 'object$term' (and 'object$by'). The 'by' variable is the last element.
`knots`	not used.

Value

An object of class '"gmrf.smooth"'.

Examples

```
require(gmrf)
n <- 100
idx <- 1:n
idy <- c(5:40, 60:n)
set.seed(64)
Q <- getQrw(n, order = 2)
x <- simQ(exp(1) * Q)
# simulate the first 3/4, say
y <- x + rnorm(n) * 3.5
y <- y[idy]
## set up variables for smoothing
rownames(Q) <- colnames(Q) <- idx
## fit an RW2 smoother with restricted df
g1 <- gam(y ~ s(idy, bs = "gmrf", xt = list(penalty = Q), k = length(y)-1), method="REML")
summary(g1)
plot(idy, y, xlim = range(idx), ylim = range(x,y))
lines(idx, x, col = "blue", lwd = 2)
pred <- predict(g1, newdata = list(idy = idx), se = TRUE)
```

```
lines(idx, pred$fit, col = "red", lwd = 2)
lines(idx, pred$fit + 2*pred$se.fit, col = "red", lty = 2)
lines(idx, pred$fit - 2*pred$se.fit, col = "red", lty = 2)
```

Index