Package 'CARrampsOcl'

February 19, 2015

Type Package

Title Reparameterized and marginalized posterior sampling for conditional autoregressive models, OpenCL implementation

Version 0.1.4

Date 2013-08-18

Author Kate Cowles and Michael Seedorff and Alex Sawyer

Maintainer Kate Cowles <kate-cowles@uiowa.edu>

SystemRequirements OpenCL library; double-precision AMD or Nvidia GPU; GNU make

Imports OpenCL, fields

Suggests coda

Description This package fits Bayesian conditional autoregressive models for spatial and spatiotemporal data on a lattice. It uses OpenCL kernels running on GPUs to perform rejection sampling to obtain independent samples from the joint posterior distribution of model parameters.

License GPL (>= 3)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-10-25 18:55:12

R topics documented:

`CARrampsOcl-package` *Draws independent samples from joint posterior in Bayesian CAR models*

Description

This package fits Bayesian conditional autoregressive models (also called intrinsic Gaussian Markov random fields or IGMRFs) for spatial and spatiotemporal data on a lattice. It uses graphical processing units (GPUs) to perform rejection sampling to obtain independent samples from the joint posterior distribution of model parameters, including spatial(temporal) random effects, precision parameters, and regression coefficients for some polynomial trend surfaces.

The CARrampsOcl package can handle models with up to three structure matrices to accommodate different patterns and strengths of spatial association in different dimensions. Such models were used by Besag and Higdon (1999) and by Reich, Hodges, and Carlin (2007) are discussed in detail in Kunsch (1994) and He, Hodges, and Carlin (2007).

Rue and Held (2005), Chapter 3, describe (polynomial) intrinsic GMRFs of first, second, and higher orders. If all of the structure matrices representing association in a model are random walk 1 or 2, then the CARrampsOcl package can estimate the coefficients of the polynomial trend surface, the design matrix of which spans the null space of the Kronecker sum of the structure matrices.

Details

Package:	CARrampsOcl
Type:	Package
Version:	0.1.3
Date:	2013-04-04
License:	GPL>=3
LazyLoad:	yes

Author(s)

Kate Cowles and Michael Seedorff and Alex Sawyer Maintainer: Kate Cowles <kate-cowles@uiowa.edu>

References

Besag, J. E. and Higdon, D. M. (1993), "Bayesian inference for agricultural Field experiments." Bull. Int. Statist. Inst., 55, no.1, 121-136.

Cowles, M.K. (2011) "Back to the future: New hardware + an old algorithm equals fast sampling for Bayesian spatial models." Department of Statistics and Actuarial Science, The University of Iowa.

Cowles M.K., Yan, J., Smith, B. (2009), "Reparameterized and Marginalized Posterior and Predictive Sampling for Complex Bayesian Geostatistical Models," Journal of Computational and Graphical Statistics, 18(2), 262-282.

He, Y., Hodges, J.S., and Carlin, B.P. (2007), "Reconsidering the variance parameterization in multiple precision models," Bayesian Analysis, 2, 529-556.

Kunsch, H.R. (1994), "Robust priors for smoothing and image restoration," Annals of the Institute of Statistical Mathematics, 55, no. 1, 1-19.

Reich, B.J., Hodges, J.S., and Carlin, B.P. (2007), "Spatial analyses of periodontal data using conditionally autoregressive priors having two classes of neighbor relations," J. Amer. Statist. Assoc., 102, 44-55.

Rue, H. and Held, L. (2005). Gaussian Markov Random Fields: Theory and Applications, volume 104 of Monographs on Statistics and Applied Probability. Chapman & Hall, London.

Examples

```
# load data
  data(iowaSW06)

# construct structure matrices
  Q1<- makeRW2Q(33)       # RW2 within rows (east-west)
  Q2<- makeRW2Q(24)       # RW2 within columns (north-south)

  iowaQ <- list( list( type="Gen", content=Q1 ), list( type="Gen", content=Q2))
# dimenstions of Q1, Q2,  in that order
  na<- nrow(Q1)
  nb<- nrow(Q2)


# construct the design matrix with with as many columns as there are
# in null space of kronecker prod of Q's

  X2 <- cbind( rep(1,nb), 1:nb)
  X1 <- cbind( rep(1,na), 1:na)
  X <-  kronecker( X2, X1)

# parameters of gamma prior densities on tausqy, tausqphi[1], tausqphi[2]
  alpha2 = beta2 <- c(.1, .1, .1)

# number of samples
  nsamp = 100

#random seed
  myseed = 314

  output <- CARrampsOcl.fit(alpha=alpha2, beta=beta2,
            Q=iowaQ, y=iowaSW06,  nsamp=nsamp,
```

```
            seed=myseed,
            fixed = FALSE, randeffs=TRUE, coefs=TRUE,designMat=X,
            mult= 50)

# summarize marginal posterior densities of precision parameters
  library(coda)
  summary(as.mcmc( output$params ))

# summarize marginal posterior densities of regression coefficients
#    intercept, slope within rows (west-to-east linear trend),
#    slope for columns (north to south linear trend),
#    interaction between rows  and columns
  summary(as.mcmc(output$regcoefs))

# summary statistics for site-specific random effects at first 10 sites
  print( cbind( output$phi$phimean, output$phi$phisd)[1:10,] )

# plot the raw data and the posterior means of the site-specific random effects
  plot2Q( output, numcols=16)
```

`CARrampsOcl.fit` *Fit Bayesian normal conditional autoregressive model*

Description

This function fits CAR models to data. It draws independent samples from the posterior distribution of precision parameters and regression coefficients for certain polynomial trend surfaces. If the `randeffs` argument is set to true, it will also produce the estimated mean and standard deviation of the marginal posterior density of each random effect.

Usage

```
CARrampsOcl.fit(alpha, beta, Q, y, nsamp, seed, fixed = FALSE, coefs = FALSE,
randeffs = FALSE, designMat = NULL, mult = 20, filename = "params.txt")
```

Arguments

`alpha`	Vector of `alpha` parameters for gamma prior densities on precisions. The first element is for the prior density on the measurement error precision, and the remaining entries are for the the prior density(ies) on the random effects precision(s), in the same order as the Q matrices in the list argument Q.
`beta`	Vector of `beta` parameters for gamma prior densities on precisions. The first element is for the prior density on the measurement error precision, and the remaining entries are for the the prior density(ies) on the random effects precisions, in the same order as the Q matrices in the list argument Q.
`Q`	List of specifications of structure matrices for the CAR model. Each element of the list Q is itself a list with 2 elements: \ `type`: one of c("CAR1","RW1","Gen") indicating a conditional autoregressive structure of order 1 on a rectangular grid,

a random walk 1 structure on a line, or a general (any other) structure, and \ content: for "CAR1," a two-vector giving the dimensions of the rectangular grid; for "RW1," a scalar giving the dimension; and for "Gen," a matrix – the symmetric structure matrix.

`y` Vector of observed data values. If there is more than one structure matrix in the list called Q, then the data must be ordered accordingly. For example, suppose the data are measured on a rectangular lattice with r rows and c columns, and that Q[[1]] represents the within-row neighborhood structure while Q[[2]] represents the within-column neighborhood structure. Then the data must be in row-major order.

`nsamp` Integer representing the number of desired draws from the joint posterior density of the model parameters.

`seed` Seed for the random number generator. Must be a positive integer.

`fixed` Logical value (TRUE or FALSE). CARrampsOcl.fit utilizes rejection sampling to draw from the marginalized joint posterior density. This involves generating a batch of candidate values, of which some are rejected and some are accepted. If `fixed` is TRUE, then a single batch of candidates of size `nsamp` is generated and the number of accepted candidates returned is likely to be much smaller than `nsamp`. If `fixed` is FALSE (the default), then additional batches of candidates will be generated iteratively until at least `nsamp` samples have been accepted.

`coefs` Logical value. If `coefs` is TRUE, then regression coefficients for a polynomial trend surface will be estimated. The design matrix must be provided in the argument `designmat`.

`randeffs` Logical. If true, random effects corresponding to each observation in the dataset will be calculated and returned.

`designMat` The design matrix for regression. It must be set to NULL if `coefs` is FALSE. The only regression coefficients that CARramps can estimate are for polynomial trend surfaces. The degree of the polynomial trend surface must agree with the structure matrix or matrices in the list Q. For a CAR(1) or RandomWalk(1) structure matrix, the appropriate design matrix is an intercept (column of ones) only. RandomWalk(2) structure matrices correspond to linear trend surfaces, and the appropriate design matrix consists of an intercept and a column of consecutive integers. The example code for this function demonstrates how to construct the design matrix for a two-dimensional linear trend surface, which corresponds to a Q list consisting of two RandomWalk(2) structure matrices.

`mult` The `mult` argument and the `nsamp` argument together determine the size of the batches of candidates generated for rejection sampling, which is `mult` times `nsamp`. Leaving `mult` at its default value of 20 is generally safe. Decreasing `mult` for very small datasets with high acceptance rates, or increasing it for large datasets with low acceptance rates, may speed computing.

`filename` The name for the file in which batches of accepted samples of precision parameters will be saved while the function is running. Future releases of the CARrampsOcl package will be able to use this output file to resume sampling after interruption.

Value

params	Matrix of samples drawn from joint posterior density of the precision parameters. The first column is the measurement error precision, and subsequent columns are the spatial precisions in the same order as the Q matrices.
phi	List with two components: phimean is vector of means of posterior densities random effects; phisd is vector of standard deviations of marginal posterior densities of random effects. Both are in the same order as the observations the y vector.
preds	Always NULL in current release of CARramps. Future releases will enable estimation of means and standard deviations from posterior predictive distributions, which will be output here.
regcoefs	Matrix of samples drawn from joint posterior density of regression coefficients. There is one column for each coefficient.
y	Data vector
acptrate	Rejection sampling acceptance rate
n	Number of observations in y

Author(s)

Kate Cowles

Examples

```
# load data
  data(iowaSW06)

# construct random walk 2 structure matrix for each dimension
  Q1<- makeRW2Q(33)       # for rows
  Q2<- makeRW2Q(24)       # for columns


# dimensions of Q1, Q2,  in that order
  na<- nrow(Q1)
  nb<- nrow(Q2)

  Q <- list( list(type="Gen",content=Q1), list(type="Gen",content=Q2) )

# construct the design matrix with with as many columns as there are
# in null space of kronecker prod of Q's

  X2 <- cbind( rep(1,nb), 1:nb)
  X1 <- cbind( rep(1,na), 1:na)
  X <-  kronecker( X2, X1)

# parameters of gamma prior densities on tausqy, tausqphi[1], tausqphi[2]
  alpha2 = beta2 <- c(.1, .1, .1)

# number of samples
  nsamp = 100
```

```
#random seed
  myseed = 314

  output <- CARrampsOcl.fit(alpha=alpha2,
            beta=beta2, Q=Q, y=iowaSW06,  nsamp=nsamp,
            seed=myseed,
            fixed = FALSE, randeffs=TRUE, coefs=TRUE,designMat=X,
            mult= 50)

# summarize marginal posterior densities of precision parameters
  library(coda)
  summary(as.mcmc( output$params ))

# summarize marginal posterior densities of regression coefficients
#    intercept, slope within rows (west-to-east linear trend),
#    slope for columns (north to south linear trend),
#    interaction between rows  and columns
  summary(as.mcmc( output$regcoefs))

# summary statistics for site-specific random effects at first 10 sites
  print( cbind( output$phi$phimean, output$phi$phisd)[1:10,] )

# plot the raw data and the posterior means of the site-specific random effects
  plot2Q( output, numcols=32, col = rev(terrain.colors(32)),
  rev.inds = c(FALSE, TRUE))
```

`iowaSW06` *Iowa normalized difference vegetation index (NDVI) data*

Description

Normalized difference vegetation index (NDVI) values derived from satellite image data from southwest Iowa and eastern Nebraska in July 2006. These are 792 values, representing NDVI at each pixel on a rectangle with 33 rows and 24 columns. The cities of Des Moines, Omaha/Council Bluffs, and Lincoln are included in the rectangular area.

Usage

```
data(iowaSW06)
```

Format

A vector of 792 integer values. The data are in row-major order.

Source

http://glcf.umiacs.umd.edu/data/gimms/

References

Pinzon, J., Brown, M.E. and Tucker, C.J., 2005. Satellite time series correction of orbital drift artifacts using empirical mode decomposition. In: N. Huang (Editor), Hilbert-Huang Transform: Introduction and Applications, pp. 167-186.

Tucker, C.J., J. E. Pinzon, M. E. Brown, D. Slayback, E. W. Pak, R. Mahoney, E. Vermote and N. El Saleous (2005), An Extended AVHRR 8-km NDVI Data Set Compatible with MODIS and SPOT Vegetation NDVI Data. International Journal of Remote Sensing, Vol 26:20, pp 4485-5598.

See Also

plot2Q

Examples

```
data(iowaSW06)
```

---

iowaSW97_06small — *Southwest Iowa 10-year normalized difference vegetation index NDVI values*

Description

Normalized difference vegetation index (NDVI) values derived from satellite image data from southwest Iowa and eastern Nebraska in July of each year from 1997 through 2006. These are 2040 values, representing NDVI at each pixel on a rectangle with 17 rows and 12 columns at each of 10 times.

Usage

```
data(iowaSW97_06small)
```

Format

A vector of 2040 integer values. The data are in row-major order within year.

Source

http://glcf.umiacs.umd.edu/data/gimms/

References

Pinzon, J., Brown, M.E. and Tucker, C.J., 2005. Satellite time series correction of orbital drift artifacts using empirical mode decomposition. In: N. Huang (Editor), Hilbert-Huang Transform: Introduction and Applications, pp. 167-186.

Tucker, C.J., J. E. Pinzon, M. E. Brown, D. Slayback, E. W. Pak, R. Mahoney, E. Vermote and N. El Saleous (2005), An Extended AVHRR 8-km NDVI Data Set Compatible with MODIS and SPOT Vegetation NDVI Data. International Journal of Remote Sensing, Vol 26:20, pp 4485-5598.

See Also

plot3Q

Examples

```
data(iowaSW97_06small)
```

---

makeQ *Function to construct a structure matrix for 2-dimensional CAR(1) model.*

Description

Function to construct a structure matrix for 2-dimensional CAR(1) model.

Usage

```
makeQ(nr, nc)
```

Arguments

nr	number of rows in lattice
nc	number of columns in lattice

Value

structure matrix of dimension (nr * nc) by (nr * nc)

Author(s)

Kate Cowles

Examples

```
# Construct 300 x 300 structure matrix for intrinsic CAR(1) for
# lattice data on a rectangle with 15 rows and 20 columns
makeQ( 15, 20)
```

makeRW1Q *Function to construct structure matrix for 1-dimenstional random walk 1.*

Description

Function to construct structure matrix for 1-dimenstional random walk 1.

Usage

```
makeRW1Q(n)
```

Arguments

n	dimension of structure matrix

Value

structure matrix of dimension n by n

Author(s)

Kate Cowles

Examples

```
# Construct the structure matrix for a RW(1) on a line at 15 equally-spaced
# points
makeRW1Q(15)
```

makeRW2Q *Function to construct structure matrix for 1-dimensional random walk 2.*

Description

Function to construct structure matrix for 1-dimensional random walk 2.

Usage

```
makeRW2Q(n)
```

Arguments

n	dimension of structure matrix

Value

RW2 structure matrix of dimension n by n

Author(s)

Kate Cowles

See Also

CARrampsOcl.fit

Examples

```
# construct structure matrix for RW(2) on a line at 15 equally-spaced points
makeRW2Q(15)
```

---

plot2Q *Function to produce image plot of 2-dimensional data modeled with 2 separate structure matrices.*

Description

Function to produce image plot of 2-dimensional data modeled with 2 separate structure matrices.

Usage

```
plot2Q(objname, numcols = 64, col = rev(terrain.colors(numcols)),
rev.inds = c(FALSE, FALSE))
```

Arguments

objname	name of output object produced by CARrampsOcl.fit
numcols	number of shades from the color palette to be used
col	color palette to be used in plotting; the default plots high values in green and low values in pink.
rev.inds	Should the plotting indices on the two-dimensional plot be reversed? Setting rev.inds = c(TRUE,FALSE) flips the plot from left to right; rev.inds = c(FALSE,TRUE) turns the plot upside down.

Details

This function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the posterior distributions of the corresponding random effects.

Value

This function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the posterior distributions of the corresponding random effects.

Author(s)

Kate Cowles

Examples

```
# load data
  data(iowaSW06)

# construct structure matrix
  Q1<- makeRW2Q(33)       # for rows
  Q2<- makeRW2Q(24)       # for columns


# dimensions of Q1, Q2,  in that order
    na<- nrow(Q1)
    nb<- nrow(Q2)

Q <- list( list(type="Gen",content=Q1), list(type="Gen",content=Q2) )

# construct the design matrix with with as many columns as there are
# in null space of kronecker prod of Q's

X2 <- cbind( rep(1,nb), 1:nb)
X1 <- cbind( rep(1,na), 1:na)
X <-  kronecker( X2, X1)

# parameters of gamma prior densities on tausqy, tausqphi[1], tausqphi[2]
alpha2 = beta2 <- c(.1, .1, .1)

# number of samples
nsamp = 100

#random seed
myseed = 314

output <- CARrampsOcl.fit(alpha=alpha2,
            beta=beta2, Q=Q, y=iowaSW06,  nsamp=nsamp,
            seed=myseed,
            fixed = FALSE, randeffs=TRUE, coefs=TRUE,designMat=X,
```

```
            mult= 50)

# plot the raw data and the posterior means of the site-specific random effects

plot2Q( output, numcols=32, col = rev(terrain.colors(32)), rev.inds = c(FALSE, TRUE))
```

---

plot3Q *Plot 3-dimensional data modeled with 3 separate structure matrices*

Description

Plot 3-dimensional data modeled with 3 separate structure matrices.

Usage

```
plot3Q(objname, numcols = 64, col = rev(terrain.colors(numcols)),
plotdims = c(1, 2), rev.inds = c(FALSE, FALSE), blocks = NULL,
animate = FALSE, intv = 3,
title = c("Raw data", "Estimated underlying truth"), sub = NULL)
```

Arguments

objname	name of output object produced by CARrampsOcl.fit
numcols	number of shades from the color palette to be used
col	color palette to be used in plotting; the default plots high values in green and low values in pink.
plotdims	This function produces a sequence of two-dimensional plots. The plotdims argument is a 2-vector identifying which two dimensions appear in the rows and columns of each two-dimensional plot. The numbers refer to the order of the structure matrices specified in the Q argument to CARrampsOcl.fit. The default is the first two dimensions.
rev.inds	Should the plotting indices on each two-dimensional plot be reversed? Setting rev.inds = c(TRUE,FALSE) flips the plots from left to right; rev.inds = c(FALSE,TRUE) turns the plots upside down.
blocks	Subset which two-dimensional plots are displayed. Default is that the corresponding two-dimensional plot is displayed for each value of the third dimension. To subset, set blocks to a set of integers. For example, blocks = 1:5 will display the first 5 plots only.
animate	If animate = FALSE, the user is prompted to press a key to move to the next plot. Otherwise, the next plot will automatically appear after intv seconds.
intv	How many seconds to wait before displaying next plot (ignored if animate=FALSE).
title	Vector of character values; the titles of the two plots.
sub	Vector of subtitles for plots.

Details

For each value of a third dimension, this function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Value

For each value in `blocks`, this function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Author(s)

Alex Sawyer and Kate Cowles

Examples

```
## Not run:
data(iowaSW97_06small)
na<- 17
nb <- 12
nc <- 10
Q <- list( list(type="RW1",content=na), list(type="RW1",content=nb),
list(type="RW1",content=10))
alpha <- beta <- rep(0.01,4)
X<- matrix( rep(1, na * nb * nc), ncol=1)

# Add noise to data to see whether fitting method can extract true image
y <- iowaSW97_06small/100 + rnorm(na*nb*nc,sd=6)

outputSW <- CARrampsOcl.fit(alpha, beta, Q, y, nsamp=50,
     seed=2, fixed = FALSE, coefs = TRUE,
     randeffs = TRUE, designMat = X, mult = 2000, filename = "params.txt")
require(coda)
summary( as.mcmc(1/sqrt(outputSW$params)))

plot3Q( objname=outputSW, numcols=64, col=rev(terrain.colors(64)),
plotdims=c(1,2), rev.inds=c(FALSE,TRUE), blocks=NULL, animate=TRUE,
intv=3, title=c("Raw data","Estimated underlying truth"), sub=1997:2006 )

## End(Not run)
```

---

plotCAR1 *Function to plot 2-dimensional data modeled using a single structure matrix.*

Description

Function to plot 2-dimensional data modeled using a single structure matrix.

Usage

```
plotCAR1(objname, numcols = 64, col = rev(terrain.colors(numcols)), cardims,
   rev.inds = c(FALSE, FALSE),
   title = c("Raw data", "Estimated underlying truth"),
   sub = NULL)
```

Arguments

objname	name of output object produced by CARrampsOcl.fit
numcols	number of shades from the color palette to be used
col	color palette to be used in plotting; the default plots high values in green and low values in pink.
cardims	Two-vector specifying the number of rows and columns in the CAR1 structure matrix; same as dimension of CAR1 Q matrix provided to CARrampsOcl.fit.
rev.inds	Should the plotting indices on each two-dimensional plot be reversed? Setting rev.inds = c(TRUE,FALSE) flips the plots from left to right; rev.inds = c(FALSE,TRUE) turns the plots upside down.
title	Vector of character values; the titles of the two plots.
sub	subtitle for plots; vector of length equal to the number of pages of plots to be displayed

Details

This function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Value

This function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Author(s)

Kate Cowles

Examples

```
# load data
## Not run:
data(iowaSW06)

Q <- list( list(type="CAR1",content=c(33,24)) )

# construct the design matrix with with as many columns as there are
# in null space of kronecker prod of Q's
```

```
X <-  matrix( rep(1,33*24), ncol=1)

# parameters of gamma prior densities on tausqy, tausqphi[1], tausqphi[2]
alpha2 = beta2 <- c(.1, .1)
# number of samples
nsamp = 100

#random seed
myseed = 314

output <- CARrampsOcl.fit(alpha=alpha2,
            beta=beta2, Q=Q, y=iowaSW06,  nsamp=nsamp,
            seed=myseed,
            fixed = FALSE, randeffs=TRUE, coefs=TRUE,designMat=X,
            mult= 50)

# plot the raw data and the posterior means of the site-specific random effects

plotCAR1( output, numcols=32, col = rev(terrain.colors(32)),
    cardims = c(33 ,24 ), rev.inds = c(FALSE, TRUE))

## End(Not run)
```

`plotCAR1plus1Q` *Plot 3-dimensional data modeled with a 2-dimensional CAR1 plus a 1-dimensional structure matrix*

Description

Plot 3-dimensional data modeled with a 2-dimensional CAR1 plus a 1-dimensional structure matrix

Usage

```
plotCAR1plus1Q(objname, numcols = 64, col = rev(terrain.colors(numcols)),
    plotdim = 1, cardims, rev.inds = c(FALSE, FALSE), blocks = NULL,
    animate = FALSE, intv = 3,
    title = c("Raw data", "Estimated underlying truth"), sub = NULL)
```

Arguments

`objname`	name of output object produced by `CARrampsOcl.fit`
`numcols`	number of shades from the color palette to be used
`col`	color palette to be used in plotting; the default plots high values in green and low values in pink.
`plotdim`	This function produces a sequence of two-dimensional plots. The `plotdim` argument is a scalar identifying which dimension appears in the rows and columns of each two-dimensional plot. The number refers to the order of the structure matrices specified in the `Q` argument to `CARrampsOcl.fit`. The default is the first dimension.

cardims	Two-vector specifying the number of rows and columns in the CAR1 structure matrix; same as dimension of CAR1 Q matrix provided to CARrampsOcl.fit.
rev.inds	Should the plotting indices on each two-dimensional plot be reversed? Setting rev.inds = c(TRUE,FALSE) flips the plots from left to right; rev.inds = c(FALSE,TRUE) turns the plots upside down.
blocks	Subset which two-dimensional plots are displayed. Default is that the corresponding two-dimensional plot is displayed for each value of the third dimension. To subset, set blocks to a set of integers. For example, blocks = 1:5 will display the first 5 plots only.
animate	If animate = FALSE, the user is prompted to press a key to move to the next plot. Otherwise, the next plot will automatically appear after intv seconds.
intv	How many seconds to wait before displaying next plot (ignored if animate=FALSE).
title	Vector of character values; the titles of the two plots.
sub	Vector of subtitles for plots.

Details

For each value in blocks, this function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Value

For each value in blocks, this function plots two two-dimensional plots side-by-side. The left plot is of the raw data input into the CARrampsOcl.fit function, and the right plot is of the estimated means of the of the posterior distributions of the corresponding random effects.

Note

Uses image.plot from the fields package.

Author(s)

Kate Cowles

Examples

```
data(iowaSW97_06small)

na<- 17
nb <- 12
nc <- 10
Q <- list( list(type="CAR1",content=c(na,nb)), list(type="RW1",content=nc))
alpha <- beta <- rep(0.01,3)
X<- matrix( rep(1, na * nb * nc), ncol=1)

y <- iowaSW97_06small/100 + rnorm(na*nb*nc,sd=6)   # add noise
print(system.time(outputSW2 <- CARrampsOcl.fit(alpha, beta, Q, y, nsamp=50,
     seed=2, fixed = FALSE, coefs = TRUE,
```

```
     randeffs = TRUE, designMat = X, mult = 2000, filename = "params.txt")))

plotCAR1plus1Q( objname=outputSW2, numcols=64, col=rev(terrain.colors(64)),
   plotdim=1, cardims=c(na,nb),
   rev.inds=c(FALSE,TRUE), blocks=NULL, animate=FALSE, intv=3,
   title=c("Raw data","Estimated underlying truth"), sub=1997:2006 )
```

`rdirichlet` *Generate random samples from dirichlet distribution.*

Description

Generate random samples from dirichlet distribution.

Usage

```
rdirichlet(n, parms)
```

Arguments

`n`	number of vectors to be drawn
`parms`	vector of parameters of dirichlet distribution

Value

n draws from the dirichlet density with parameters in `parms`

Author(s)

Kate Cowles

Examples

```
## The function is currently defined as
function (n, parms)
{
# generate n random vectors from dirichlet
# rejection envelope

    l <- length(parms)
    x <- matrix(rgamma(l * n, parms), ncol = l, byrow = TRUE)
    sm <- x %*% rep(1, l)
    return(x/as.vector(sm))
  }
```

Index