

# Package ‘DSSP’

October 12, 2022

**Type** Package

**Title** Implementation of the Direct Sampling Spatial Prior

**Version** 0.1.1

**Date** 2022-07-07

**Maintainer** Gentry White <gentry.white@qut.edu.au>

**Description** Draw samples from the direct sampling spatial prior model as described in G. White, D. Sun, P. Speckman (2019) <[arXiv:1906.05575](https://arxiv.org/abs/1906.05575)>. The basic model assumes a Gaussian likelihood and derives a spatial prior based on thin-plate splines.

**License** GPL (>= 3)

**URL** <https://github.com/gentrywhite/DSSP>

**BugReports** <https://github.com/gentrywhite/DSSP/issues>

**Imports** mcmcse, posterior, rust, sp

**Suggests** cowplot, ggplot2, gstat, interp, knitr, rmarkdown, testthat (>= 3.0.0)

**LinkingTo** Rcpp, RcppArmadillo

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.0

**NeedsCompilation** yes

**Author** Gentry White [aut, cre] (<<https://orcid.org/0000-0002-1170-9299>>),  
Rex Parsons [aut] (<<https://orcid.org/0000-0002-6053-8174>>)

**Repository** CRAN

**Date/Publication** 2022-07-12 11:00:06 UTC

## R topics documented:

DSSP-package	2
DSSP	3
make.M	4
plot.dsspMod	5
predict.dsspMod	6
residuals.dsspMod	7
sample.delta	8
sample.eta	9
sample.nu	10
summary.dsspMod	11
tps.rbf	12
<b>Index</b>	<b>14</b>

---

DSSP-package	<i>Implementation of the Direct Sampling Spatial Prior</i>
--------------	--

---

### Description

Draw samples from the direct sampling spatial prior model as described in G. White, D. Sun, P. Speckman (2019) <arXiv:1906.05575>. The basic model assumes a Gaussian likelihood and derives a spatial prior based on thin-plate splines.

### Package Content

Index of help topics:

DSSP	DSSP
DSSP-package	Implementation of the Direct Sampling Spatial Prior
make.M	Precision Matrix Function
plot.dsspMod	Diagnostic, Density and Contour Plots
predict.dsspMod	Predictions from a model with new data.
residuals.dsspMod	Get residuals from 'dsspMod' model
sample.delta	Function to sample from the posterior of the variance parameter
sample.eta	Function to sample from the posterior of the smoothing parameter eta conditioned on the data y.
sample.nu	Function to sample from the posterior of the spatial effects
summary.dsspMod	Summarise a 'dsspMod' model
tps.rbf	TPS radial basis function

### Maintainer

Gentry White <gentry.white@qut.edu.au>

**Author(s)**

NA

DSSP

*DSSP***Description**

This function samples from the log-posterior of all parameters in the model and returns a list object containing the samples. It performs a few compatibility checks on the inputs, then calls the `sample.eta()`, `sample.delta()`, and `sample.nu()`.

**Usage**

```
DSSP(formula, data, N, pars, log_prior = function(x) -x, coords = NULL)
```

**Arguments**

<code>formula</code>	a two sided linear formula with the response on left and the covariates on the right.
<code>data</code>	a <code>data.frame</code> or <code>sp::SpatialPointsDataFrame</code> containing the response variable, covariates and coordinates.
<code>N</code>	is the number of random samples to be drawn from the joint posterior for <code>eta</code> , <code>delta</code> , and <code>nu</code> .
<code>pars</code>	a vector of the prior shape and rate parameters for the inverse-gamma prior distribution of <code>delta</code> , the variance parameter for the Gaussian likelihood.
<code>log_prior</code>	a function evaluating the log of the prior density of <code>eta</code> . Default to be <code>function(x) -x</code> .
<code>coords</code>	spatial coordinates passed as the value argument to <code>sp::coordinates()</code> .

**Details**

The direct sampling spatial prior model assumes that the spatial model can be written as the likelihood parameterised with mean vector `nu` and variance `delta`

$$(y|nu, delta) N(nu, delta * I)$$

where `I` is the identity matrix. The prior for the vector of spatial effects `nu` is improper but is proportional to

$$\pi(nu|eta) \propto (\det(M)/2\pi)^{1/2} * \exp(-etanu' M nu/2),$$

the prior for `delta` is assumed to be a inverse-gamma distribution

$$(delta) IG(a, b)$$

and the prior for `eta` can be specified for the user as any valid density function for `eta > 0`.

**Value**

A list containing N samples of nu, eta, delta, and the original data X and Y.

**Examples**

```
## Use the Meuse River dataset from the package 'gstat'

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y

f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter

## Draw 100 samples from the posterior of eta given the data y.
OUTPUT <- DSSP(
  formula = log(zinc) ~ 1, data = meuse.all, N = 100,
  pars = c(0.001, 0.001), log_prior = f
)
```

---

make.M

*Precision Matrix Function*

---

**Description**

This function creates the precision matrix for the spatial prior based on thin-plate splines and returns the matrix M, and its eigenvalues and eigenvectors

**Usage**

```
make.M(X, covariates)
```

**Arguments**

X	a matrix of spatial coordinates. It is recommended that the coordinates be scaled and centred.
covariates	the observed values for the covariates (including intercept).

**Details**

The M matrix is the precision matrix for the spatial effects from the direct sampling spatial prior (DSSP) model. M is based on thin plate splines basis functions, see White et. al. 2019 for more details on how the matrix M is constructed.

**Value**

A list containing the precision matrix M and the object M.eigen containing eigenvalues and eigenvectors for the matrix M.

**Examples**

```
## Use the Meuse River dataset from the package 'gstat'

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y
X <- scale(coordinates(meuse.all))
make.M(X)
```

---

plot.dsspMod

*Diagnostic, Density and Contour Plots*


---

**Description**

Diagnostic, Density and Contour Plots

**Usage**

```
## S3 method for class 'dsspMod'
plot(
  x,
  robust_residuals = TRUE,
  contour_plots = TRUE,
  nx = 100,
  ny = 100,
  nlevels = 5,
  ...
)
```

**Arguments**

x	an object of class dsspMod
robust_residuals	whether to use robust residuals (median of predicted). Default to be TRUE.
contour_plots	whether or not to return a second panel with contour plots. Defaults to TRUE
nx	dimension of output grid in x direction. Used for interpolation (akime::interp()).
ny	dimension of output grid in y direction. Used for interpolation (akime::interp()).
nlevels	number of levels used in contour plot.
...	additional arguments that are passed to ggplot2::scale_fill_distiller().

**Value**

a list containing the plots printed (individually and together in grid)

**Examples**

```

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y

f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter

## Draw 100 samples from the posterior of eta given the data y.
OUTPUT <- DSSP(
  formula = log(zinc) ~ 1, data = meuse.all, N = 100,
  pars = c(0.001, 0.001), log_prior = f
)
plot(OUTPUT, contour_plots = FALSE)

```

---

predict.dsspMod	<i>Predictions from a model with new data.</i>
-----------------	--

---

**Description**

Predictions from a model with new data.

**Usage**

```
## S3 method for class 'dsspMod'
predict(object, newdata, ...)
```

**Arguments**

object	a fitted dsspMod object.
newdata	a data frame for which to evaluate predictions.
...	optional and ignored arguments.

**Value**

returns matrix with posterior densities for each row in the input data.

**Examples**

```

data("meuse.all", package = "gstat")
sp::coordinates(meuse.all) <- ~ x + y
meuse.fit <- DSSP(
  formula = log(zinc) ~ 1, data = meuse.all[1:155, ], N = 100, function(x) -2 * log(1 + x),
  pars = c(0.001, 0.001)
)
preds <- predict(meuse.fit, meuse.all[156:164, ])

```

---

residuals.dsspMod	<i>Get residuals from dsspMod model</i>
-------------------	---

---

## Description

Get residuals from dsspMod model

## Usage

```
## S3 method for class 'dsspMod'  
residuals(object, newdata, robust = TRUE, ...)
```

## Arguments

object	an object of class dsspMod
newdata	a data frame for which to estimate residuals.
robust	whether or not to use median (rather than mean) of posterior density to as estimate calculate residuals.
...	additional arguments which are ignored.

## Value

vector containing residuals with same length as rows in data used.

## Examples

```
library(sp)  
library(gstat)  
data(meuse.all)  
coordinates(meuse.all) <- ~ x + y  
  
f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter  
  
## Draw 100 samples from the posterior of eta given the data y.  
OUTPUT <- DSSP(  
  formula = log(zinc) ~ 1, data = meuse.all, N = 100,  
  pars = c(0.001, 0.001), log_prior = f  
)  
residuals(OUTPUT)
```

sample.delta

*Function to sample from the posterior of the variance parameter***Description**

This function samples from the log-posterior density of the variance parameter from the likelihood

**Usage**

```
sample.delta(eta, ND, EV, Q, pars)
```

**Arguments**

eta	samples of the smoothing parameter from the sample.eta function.
ND	the rank of the precision matrix, the default value is n-3 for spatial data.
EV	eigenvalues of the precision matrix spatial prior from the function make.M().
Q	the data vector from the cross-product of observed data, Y, and eigenvalues from the M matrix, V.
pars	a vector of the prior shape and rate parameters for the inverse-gamma prior distribution of delta.

**Value**

N samples drawn from the posterior of  $\pi(\text{delta}|\text{eta}, y)$ .

**Examples**

```
## Use the Meuse River dataset from the package 'gstat'

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y
X <- scale(coordinates(meuse.all))
tmp <- make.M(X)

M <- tmp$M

Y <- scale(log(meuse.all$zinc))

ND <- nrow(X) - 3
M.list <- make.M(X) ## Only Needs to return the eigenvalues and vectors
M <- M.list$M
EV <- M.list$M.eigen$values
V <- M.list$M.eigen$vectors
Q <- crossprod(Y, V)

f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter
```



```

## Draw 100 samples from the posterior of eta given the data y.

ETA <- sample.eta(100, ND, EV, Q, f, UL = 1000)
DELTA <- sample.delta(ETA, ND, EV, Q, pars = c(0.001, 0.001))
## Old Slow Version of sample.nu()
## sample.delta<-function(eta,nd,ev,Q,pars)
## {
##   N<-length(eta)
##   f.beta<-function(x)
##   {
##     lambda<-1/(1+x*ev)
##     b<-tcrossprod(Q,diag(1-lambda))
##     beta<-0.5*tcrossprod(Q,b)+pars[2]
##     return(beta)
##   }
##   alpha<-pars[1]+nd*0.5
##   beta<-sapply(eta,f.beta)
##   delta<-1/rgamma(N,shape=alpha,rate=beta)
##   return(delta)
## }

```

---

sample.eta

*Function to sample from the posterior of the smoothing parameter eta conditioned on the data y.*

---

### Description

This function samples from the log-posterior density of the smoothing parameter from the thin-plate splines based spatial prior using a ratio-of-uniform sampler.

### Usage

```
sample.eta(N, ND, EV, Q, UL = 1000, log_prior)
```

### Arguments

N	the number of samples desired.
ND	the rank of the precision matrix, the default value is n-3 for spatial data.
EV	eigenvalues of the precision matrix spatial prior from the function make.M().
Q	the data vector from the cross-product of observed data, Y, and eigenvalues from the M matrix, V.
UL	the upper limit for the smoothing parameter value; used for the ratio-of-uniform sampler, default is 1000.
log_prior	a function of x evaluating the log of the prior density for eta

### Value

N samples drawn from the posterior of eta given the data y  $\pi(\eta|y)$ .

## Examples

```
## Use the Meuse River dataset from the package 'gstat'

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y
X <- scale(coordinates(meuse.all))
tmp <- make.M(X)

EV <- tmp$M.eigen$values
V <- tmp$M.eigen$vectors

M <- tmp$M

Y <- scale(log(meuse.all$zinc))
Q <- crossprod(Y, V)

ND <- nrow(X) - 3
f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter

## Draw 100 samples from the posterior of eta given the data y.
sample.eta(100, ND, EV, Q, UL = 1000, f)
```

---

sample.nu

*Function to sample from the posterior of the spatial effects*

---

## Description

This function samples from the posterior density of the spatial effects from the direct sampling spatial prior (DSSP) model.

## Usage

```
sample.nu(Y, eta, delta, EV, V)
```

## Arguments

Y	vector of observed data.
eta	samples of the smoothing parameter from the <code>sample.eta</code> function.
delta	samples of the variance parameter from the <code>sample.delta</code> function.
EV	eigenvalues of the precision matrix spatial prior from the function <code>make.M()</code> .
V	eigenvectors of the precision matrix spatial prior from the function <code>make.M()</code> .

## Value

A matrix of samples with each column a random draw from the posterior of the spatial effects from the DSSP model  $\pi(nu|eta, delta, y)$ .

**Examples**

```
## Use the Meuse River dataset from the package 'gstat'

library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y
X <- scale(coordinates(meuse.all))
tmp <- make.M(X)

EV <- tmp$M.eigen$values
V <- tmp$M.eigen$vectors

Y <- scale(log(meuse.all$zinc))
Q <- crossprod(Y, V)

ND <- nrow(X) - 3
f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter
## Draw 100 samples from the posterior of eta given the data y.

ETA <- sample.eta(100, ND, EV, Q, f, UL = 1000)
DELTA <- sample.delta(ETA, ND, EV, Q, pars = c(0.001, 0.001))
NU <- sample.nu(Y, ETA, DELTA, EV, V)
```

summary.dsspMod

*Summarise a dsspMod model***Description**

Summarise a dsspMod model

**Usage**

```
## S3 method for class 'dsspMod'
summary(object, prob = 0.95, robust = FALSE, mc_se = FALSE, ...)
```

**Arguments**

object	an object of class dsspMod
prob	the desired probability to be covered by the credible intervals. The default is 0.95.
robust	whether or not to use the median (rather than the mean) to calculate the estimates that summarise the posterior. Default to FALSE.
mc_se	whether or not to include the uncertainty in Estimate caused by sampling should be shown in the summary. Defaults to FALSE.
...	additional arguments which are ignored.

**Value**

An object of class "dsspModsummary". Provides a summary of the the Direct Sampling Spatial Prior (DSSP) model. Includes details of the formula used to fit the model, and a summary of the model (*eta*, *delta*) and the covariates.

**Examples**

```
library(sp)
library(gstat)
data(meuse.all)
coordinates(meuse.all) <- ~ x + y

f <- function(x) -x ## log-prior for exponential distribution for the smoothing parameter

## Draw 100 samples from the posterior of eta given the data y.
OUTPUT <- DSSP(
  formula = log(zinc) ~ 1, data = meuse.all, N = 100,
  pars = c(0.001, 0.001), log_prior = f
)
summary(OUTPUT)
```

---

 tps.rbf

---

*TPS radial basis function*


---

**Description**

Function to compute the thin-plate splines radial basis function for internal use by the function `make.M()`.

**Usage**

```
tps.rbf(x, is.even)
```

**Arguments**

<code>x</code>	is a Euclidean distance between two points.
<code>is.even</code>	is a logical argument indicating TRUE if the dimension of the space where the thin-plate spline smoother is being fitted is even.

**Details**

This function computes the thin-plate spline radial basis function depending on the if `d` is odd or even.

**Value**

The resulting value of the thin-plate spline radial basis function.

**Examples**

```
## Use the Meuse River dataset from the package 'gstat'  
  
library(sp)  
library(gstat)  
data(meuse.all)  
coordinates(meuse.all) <- ~ x + y  
X <- scale(coordinates(meuse.all))  
D <- as.matrix(dist(X))  
K <- tps.rbf(D, TRUE)
```

# Index

## \* **package**

DSSP-package, [2](#)

DSSP, [3](#)

DSSP-package, [2](#)

make.M, [4](#)

plot.dsspMod, [5](#)

predict.dsspMod, [6](#)

residuals.dsspMod, [7](#)

sample.delta, [8](#)

sample.eta, [9](#)

sample.nu, [10](#)

summary.dsspMod, [11](#)

tps.rbf, [12](#)