Package: spCF (via r-universe)

May 29, 2026

Type Package

Title Coarse-to-Fine Spatial Modeling

Version 0.1.1

Imports FNN, fields, nloptr, dbscan, ranger, withr, Rcpp

LinkingTo Rcpp

Suggests sp, sf, knitr, rmarkdown, CARBayesdata

Description Provides functions for coarse-to-fine spatial modeling (CFSM), enabling fast spatial prediction, regression, and uncertainty quantification. This method is suitable for moderate to large samples. For further details, see Murakami et al. (2026) <doi:10.1111/gean.70034>.

License GPL (>=2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/pak/sysreqs cmake

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

Date/Publication 2026-05-06 13:24:29 UTC

RemoteUrl https://github.com/dmuraka/spcf

RemoteRef HEAD

RemoteSha 4125c1b7d7596efef838c26d7cc0e55b0e5c7aee

Contents

cf_glm *Coarse-to-fine spatial generalized linear mixed models (CF-GLMMs)*

Description

Prediction and regression via CF-GLMMs.

Usage

```
cf_glm(
  y,
  x = NULL,
  coords,
  offset = NULL,
  x0 = NULL,
  coords0 = NULL,
  offset0 = NULL,
  mod_hv
)
```

Arguments

`y`	Vector of response variables (N x 1) including continuous, count, and binary responses, following an exponential family distribution.
`x`	Matrix of covariates (N x K).
`coords`	Matrix of 2-dimensional point coordinates (N x 2).
`offset`	Optional. Vector of offset variable (N x 1) to be included in the linear predictor. It is consistent with that of `glm`.
`x0`	Optional. Matrix of covariates at prediction sites (N0 x K).
`coords0`	Optional. Matrix of 2-dimensional point coordinates at prediction sites (N0 x 2).
`offset0`	Optional. Vector of offset variables at prediction sites (N0 x 1)
`mod_hv`	Output object of the `cf_glm_hv` function.

Value

A list with the following elements:

beta Regression coefficients, their standard errors, and the lower and upper limits of the 95 percent confidence intervals.

sd_summary Standard deviation of the regression term (xb), spatial process (spatial_scale1, spatial_scale2,...), additional learning, and residuals.

e_summary Error statistics for the validation samples: pseudo R-squared, root mean squared error (RMSE), and mean absolute error (MAE).

pred Predictive means and standard deviations (sample sites).

pred0 Predictive means and standard deviations (prediction sites).

pred_q Predictive quantiles on the response scale at the sample sites. A data frame whose columns `q0.005`, `q0.025`, `q0.05`, `q0.1`, ..., `q0.9`, `q0.95`, `q0.975`, `q0.995` give the corresponding quantile levels, obtained by Gaussian approximation on the link scale followed by inverse-link transformation.

pred0_q Predictive quantiles on the response scale at the prediction sites. Column structure is identical to `pred_q`. `NULL` when prediction sites are not supplied.

bands Bandwidth values for each scale. The i-th bandwidth is used for the spatial process corresponding to the i-th column of the Z matrix.

Z Predictive mean of the spatial process in each scale (sample sites; list).

Z_sd Predictive standard deviation of the spatial process in each scale (sample sites; list).

Z0 Predictive mean of the spatial process in each scale (prediction sites; list).

Z0_sd Predictive standard deviation of the spatial process in each scale (prediction sites; list).

Other Other internal output objects.

Author(s)

Daisuke Murakami

References

Murakami, D., Comber, A., Yoshida, T., Tsutsumida, N., Brunsdon, C., & Nakaya, T. (2025). Coarse-to-fine spatial GLMMs for scalable prediction and multiscale analysis. *ArXiv*.

See Also

`cf_glm_hv`, `sp_scalewise`

Examples

```
################ Example 1: Count data modeling/Disease mapping/smoothing
set.seed(1234)
require( CARBayesdata )
require( sf )
data(pollutionhealthdata)
data(GGHB.IZ)

### Data
dat      <- pollutionhealthdata[pollutionhealthdata$year==2011,]
y        <- dat[,"observed"]             # count data
x        <- dat[,c("pm10","jsa","price")]
offset   <- log(dat[,"expected"])
coords   <- st_coordinates(st_centroid(GGHB.IZ))

### Holdout validation optimizing the number of spatial scales
mod_hv   <- cf_glm_hv(y = y, x = x, offset=offset, coords = coords, family=poisson())
```

```
### Spatial modeling and prediction
mod      <- cf_glm(y = y, x = x, coords = coords, mod_hv = mod_hv)
mod

### Mapping predictive mean and standard deviations (SD)
GGHB.IZ$y      <- y
GGHB.IZ$pred   <- mod$pred$pred
GGHB.IZ$pred_sd<- mod$pred$pred_sd
plot(GGHB.IZ[,c("pred")],lwd=0.2,axes=TRUE, key.pos=4,nbreaks=50)   # Predictive mean
plot(GGHB.IZ[,c("pred_sd")],lwd=0.2,axes=TRUE, key.pos=4,nbreaks=50)# Predictive SD

### Multiscale spatial pattern/feature extraction
mod_s1      <- sp_scalewise(mod,bw_range=c(4000,Inf)) # Large scale (4000 <= bandwidth)
mod_s2      <- sp_scalewise(mod,bw_range=c(0,4000))   # Small scale (bandwidth <= 4000)
GGHB.IZ$z1  <- mod_s1$pred$pred
GGHB.IZ$z2  <- mod_s2$pred$pred
plot(GGHB.IZ[,c("z1","z2")],lwd=0.2,axes=TRUE,key.pos=4, nbreaks=50)# Extracted features


################ Example 2: Binary data modeling/spatial prediction
set.seed(1234)
require(sp); require(sf)
data(meuse)
data(meuse.grid)

### Data
y        <- ifelse(meuse$ffreq==1, 1, 0 )# binary data
coords   <- meuse[,c("x","y")]
x        <- meuse[,"dist"]

### Data at prediction sites
coords0  <- meuse.grid[,c("x","y")]
x0       <- meuse.grid[,"dist"]

### Holdout validation optimizing the number of spatial scales
mod_hv   <- cf_glm_hv(y = y, x = x, coords = coords, family=binomial())

### Spatial modeling and prediction
mod      <- cf_glm(y = y, x=x, coords = coords, x0=x0, coords0 = coords0,
                   mod_hv = mod_hv)
mod

### Mapping predictive mean and standard deviations (SD)
meuse.grid$pred   <- mod$pred0$pred
meuse.grid$pred_sd<- mod$pred0$pred_sd
meuse.grid_sf     <- st_as_sf(meuse.grid, coords = c("x","y"))
plot(meuse.grid_sf[,"pred"], pch = 15, cex = 0.8, nbreaks = 20)   # Predictive mean
plot(meuse.grid_sf[,"pred_sd"], pch = 15, cex = 0.8, nbreaks = 20)# Predictive SD

### Multiscale spatial pattern/feature extraction
mod_s1<- sp_scalewise(mod,bw_range=c(1000,Inf)) # Large scale (1000 <= bandwidth)
mod_s2<- sp_scalewise(mod,bw_range=c(0,1000))   # Small scale (0 <= bandwidth <= 1000)
meuse.grid_sf$z1    <- mod_s1$pred0$pred
```

```
meuse.grid_sf$z2    <- mod_s2$pred0$pred
plot(meuse.grid_sf[,c("z1","z2")], pch = 15,
     cex = 0.5, nbreaks = 20,axes=TRUE) # Predictive means
```

`cf_glm_hv` *Holdout validation for coarse-to-fine training of spatial generalized linear mixed models (GLMMs)*

Description

Trains a coarse-to-fine spatial GLMMs (CF-GLMMs) and optimizes the spatial scale through progressive holdout validation.

Usage

```
cf_glm_hv(
  y,
  x = NULL,
  coords,
  offset = NULL,
  train_rat = 0.75,
  id_train = NULL,
  alpha = 0.9,
  kernel = "exp",
  family = gaussian(),
  seed = 1234
)
```

Arguments

`y`	Vector of response variables (N x 1) including continuous, count, and binary responses, following an exponential family distribution.
`x`	Matrix of covariates (N x K).
`coords`	Matrix of 2-dimensional point coordinates (N x 2).
`offset`	Optional. Vector of offset variable (N x 1) to be included in the linear predictor. It is consistent with that of `glm`.
`train_rat`	Training sample ratio (default: 0.75). For small to moderate samples (N <= 30000), samples closest to the k-means centers are used for validation samples. For larger samples, training samples are drawn at random.
`id_train`	Optional. If specified, the corresponding samples are used as training samples. Otherwise, training samples are chosen based on ‘train_rat‘.
`alpha`	Decay ratio of the kernel bandwidth in the coarse-to-fine training (default: 0.9). As it approaches one, the optimization becomes more stringent but requires longer computation time.

`kernel` Kernel type for modeling spatial dependence. ‘"exp"‘ for the exponential kernel (default) and ‘"gau"‘ for the Gaussian kernel.

`family` Description of the error distribution and link function consistent with the ’family’ argument in the `glm` function. Functionality has been confirmed for gaussian(), poisson(), and binomial(). For other families, functionality has only been verified preliminarily.

`seed` Random seed used for the training/validation split when ‘id_train‘ is not supplied. Defaults to ‘1234‘, which makes the split reproducible across calls. Set to ‘NULL‘ to allow each call to draw a different split (useful for assessing sensitivity to the split).

Value

A list with the following elements:

loss_hv Deviance loss for validation samples.

loss_hv_all All the deviance losses obtained in each learning step.

id_train ID of training samples.

other List of other outcomes, which are internally used.

Author(s)

Daisuke Murakami

References

Murakami, D., Comber, A., Yoshida, T., Tsutsumida, N., Brunsdon, C., & Nakaya, T. (2025). Coarse-to-fine spatial GLMMs for scalable prediction and multiscale analysis. *ArXiv*.

See Also

`cf_glm`

`cf_lm` *Coarse-to-fine spatial modeling (CFSM) for Gaussian response*

Description

Prediction and regression via coarse-to-fine spatial modeling.

Usage

```
cf_lm(y, x = NULL, coords, x0 = NULL, coords0 = NULL, mod_hv)
```

Arguments

y	Vector of response variables (N x 1).
x	Matrix of covariates (N x K).
coords	Matrix of 2-dimensional point coordinates (N x 2).
x0	Optional. Matrix of covariates at prediction sites (N0 x K).
coords0	Optional. Matrix of 2-dimensional point coordinates at prediction sites (N0 x 2).
mod_hv	Output object of the `cf_lm_hv` function.

Value

A list with the following elements:

beta Regression coefficients, their standard errors, and the lower and upper limits of the 95 percent confidence intervals.

sd_summary Standard deviation of the regression term (xb), spatial process (spatial_scale1, spatial_scale2,...), additionally learned components (effective if ‘cf_lm_hv/add_learn‘ is not ‘none‘), and residuals.

e_summary R-squared for the validation samples (validation_R2), root mean squared error for the validation samples (validation_RMSE), and the residual standard deviation (residual_SD).

pred Predictive means and standard deviations (sample sites).

pred0 Predictive means and standard deviations (prediction sites).

bands Bandwidth values for each scale. The i-th bandwidth is used to describe the spatial process corresponding to the i-th column of the Z matrix.

Z Predictive means of the single-scale processes at each scale, corresponding to each bandwidth value (sample sites; list).

Z_sd Predictive standard deviation of the spatial processes corresponding to in each bandwidth (sample sites; list).

Z0 Predictive mean of the spatial process corresponding to each bandwidth (prediction sites; list).

Z0_sd Predictive standard deviation of the spatial process corresponding to in each bandwidth (prediction sites; list).

Other Other internal output objects.

Author(s)

Daisuke Murakami

References

Murakami, D., Comber, A., Yoshida, T., Tsutsumida, N., Brunsdon, C., & Nakaya, T. (2026). Coarse-to-fine spatial modeling: A scalable, machine-learning-compatible framework. *Geographical Analysis*, 58(2), e70034. https://onlinelibrary.wiley.com/doi/10.1111/gean.70034

See Also

cf_glm, cf_lm_hv, sp_scalewise

Examples

```
set.seed(123)
require(sp); require(sf)
data(meuse)
data(meuse.grid)

### Data
y        <- log(meuse[,"zinc"])
coords   <- meuse[,c("x","y")]
x        <- data.frame(dist   = meuse[,"dist"],
                       ffreq2 = as.integer(meuse$ffreq == 2),
                       ffreq3 = as.integer(meuse$ffreq == 3))

### Data at prediction sites
coords0  <- meuse.grid[,c("x","y")]
x0       <- data.frame(dist   = meuse.grid[,"dist"],
                       ffreq2 = as.integer(meuse.grid$ffreq == 2),
                       ffreq3 = as.integer(meuse.grid$ffreq == 3))

### Holdout validation optimizing the number of spatial scales
mod_hv   <- cf_lm_hv(y = y, x = x, coords = coords, add_learn = "none")

### Spatial modeling and prediction
mod      <- cf_lm(y = y, x = x, x0 = x0, coords = coords, coords0 = coords0,
                  mod_hv = mod_hv)
mod

### Mapping predictive mean and standard deviations (SD)
meuse.grid$pred   <- mod$pred0$pred
meuse.grid$pred_sd<- mod$pred0$pred_sd
meuse.grid_sf     <- st_as_sf(meuse.grid, coords = c("x","y"))
plot(meuse.grid_sf[,"pred"], pch = 15, cex = 0.5, nbreaks = 20)   # Predictive mean
plot(meuse.grid_sf[,"pred_sd"], pch = 15, cex = 0.5, nbreaks = 20)# Predictive SD

### Multiscale spatial pattern/feature extraction
mod_s1<- sp_scalewise(mod,bw_range=c(1000,Inf)) # Large scale (1000 <= bandwidth)
mod_s2<- sp_scalewise(mod,bw_range=c(500,1000)) # Middle scale (500 <= bandwidth <= 1000)
mod_s3<- sp_scalewise(mod,bw_range=c(0,500))    # Small scale (bandwidth <= 500)
z1    <- mod_s1$pred0$pred                      # Predictive mean
z2    <- mod_s2$pred0$pred
z3    <- mod_s3$pred0$pred
z1_sd <- mod_s1$pred0$pred_sd                   # Predictive SD
z2_sd <- mod_s2$pred0$pred_sd
z3_sd <- mod_s3$pred0$pred_sd
meuse.grid_sf3  <- cbind(meuse.grid_sf, z1, z2, z3, z1_sd, z2_sd, z3_sd)
plot(meuse.grid_sf3[,c("z1","z2","z3")], pch = 15,
     cex = 0.5, nbreaks = 20,key.pos=4,axes=TRUE) # Predictive means
plot(meuse.grid_sf3[,c("z1_sd","z2_sd","z3_sd")], pch = 15,
```

```
cex = 0.5, nbreaks = 20,key.pos=4,axes=TRUE) # Predictive SD
```

---

cf_lm_hv *Holdout validation for the Gaussian coarse-to-fine spatial modeling (CFSM)*

Description

Trains the CFSM-based Gaussian spatial regression and optimizes the number of spatial scales through sequential holdout validation.

Usage

```
cf_lm_hv(
  y,
  x = NULL,
  coords,
  train_rat = 0.75,
  id_train = NULL,
  alpha = 0.9,
  kernel = "exp",
  add_learn = "none",
  seed = 123
)
```

Arguments

y	Vector of response variables (N x 1).
x	Matrix of covariates (N x K).
coords	Matrix of 2-dimensional point coordinates (N x 2).
train_rat	Training sample ratio (default: 0.75). For small to moderate samples (N <= 30000), samples closest to the k-means centers are used for validation samples. For larger samples, training samples are drawn at random.
id_train	Optional. If specified, the corresponding samples are used as training samples. Otherwise, training samples are chosen based on ‘train_rat‘.
alpha	Decay ratio of the kernel bandwidth in the coarse-to-fine training (default: 0.9). As it approaches one, the optimization becomes more stringent but requires longer computation time.
kernel	Kernel type for modeling spatial dependence. ‘"exp"‘ for the exponential kernel (default) and ‘"gau"‘ for the Gaussian kernel.
add_learn	If ‘"rf"‘, random forest is additionally trained to capture non-linear patterns and/or higher-order interactions. Default is ‘"none"‘, meaning no additional training.

seed — Random seed used for the training/validation split when ‘id_train‘ is not supplied. Defaults to ‘123‘, which makes the split reproducible across calls. Set to ‘NULL‘ to allow each call to draw a different split (useful for assessing sensitivity to the split).

Value

A list with the following elements:

sse_hv Sum-of-squared error (SSE) for validation samples.

sse_hv_all All the SSEs obtained in each learning step.

id_train ID of training samples.

other List of other outcomes, which are internally used.

Author(s)

Daisuke Murakami

References

Murakami, D., Comber, A., Yoshida, T., Tsutsumida, N., Brunsdon, C., & Nakaya, T. (2025). Coarse-to-fine spatial GLMMs for scalable prediction and multiscale analysis. *ArXiv*.

See Also

`cf_lm`

---

sp_scalewise — *Extract scale-wise spatial processes*

Description

Evaluate mean and variance of the spatial process with bandwidth values within a pre-specified range

Usage

```
sp_scalewise(mod, bw_range = c(0, Inf))
```

Arguments

mod — Output object from the `cf_lm` or `cf_glm` function.

bw_range — Range of bandwidth values of the simulated spatial processes. For example, if bw_range = c(10, 20), spatial processes with bandwidths between 10 and 20 are synthesized and simulated. The default is c(0, Inf), which synthesizes all scales.

Value

A list with the following elements:

pred Means and standard deviations of the spatial process (sample sites).

pred0 Means and standard deviations of the spatial process (prediction sites). `NULL` when `mod` was fitted without prediction sites.

Author(s)

Daisuke Murakami

See Also

`cf_lm`, `cf_glm`

Index