Package ‘geoGAM’

July 22, 2025

Type Package

Title Select Sparse Geoadditive Models for Spatial Prediction

Version 0.1-3

Date 2023-10-30

Depends R(>= 2.14.0)

Imports mboost, mgcv, grpreg, MASS

Suggests raster, sp

Description A model building procedure to build parsimonious geoadditive model from a large number of covariates. Continuous, binary and ordered categorical responses are supported. The model building is based on component wise gradient boosting with linear effects, smoothing splines and a smooth spatial surface to model spatial autocorrelation. The resulting covariate set after gradient boosting is further reduced through backward elimination and aggregation of factor levels. The package provides a model based bootstrap method to simulate prediction intervals for point predictions. A test data set of a soil mapping case study in Berne (Switzerland) is provided. Nussbaum, M., Walthert, L., Fraefel, M., Greiner, L., and Papritz, A. (2017) <doi:10.5194/soil-3-191-2017>.

License GPL (>= 2)

Author Madlene Nussbaum [cre, aut],
Andreas Papritz [ths]

Maintainer Madlene Nussbaum <m.nussbaum@uu.nl>

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2023-11-14 18:00:07 UTC

Contents

`berne` *Berne – soil mapping case study*

Description

The Berne dataset contains soil responses and a large set of explanatory covariates. The study area is located to the Northwest of the city of Berne and covers agricultural area. Soil responses included are soil pH (4 depth intervals calculated from soil horizon), drainage classes (3 ordered classes) and presence of waterlogging characteristics down to a specified depth (binary response).

Covariates cover environmental conditions by representing climate, topography, parent material and soil.

Usage

```
data("berne")
```

Format

A data frame with 1052 observations on the following 238 variables.

`site_id_unique` ID of original profile sampling

`x` easting, Swiss grid in m, EPSG: 21781 (CH1903/LV03)

`y` northing, Swiss grid in m, EPSG: 21781 (CH1903/LV03)

`dataset` Factor splitting dataset for `calibration` and independent `validation`. `validation` was assigned at random by using weights to ensure even spatial coverage of the agricultural area.

`dclass` Drainage class, ordered Factor.

`waterlog.30` Presence of waterlogging characteristics down to 30 cm (1: presence, 0: absence)

`waterlog.50` Presence of waterlogging characteristics down to 50 cm (1: presence, 0: absence)

`waterlog.100` Presence of waterlogging characteristics down to 100 cm (1: presence, 0: absence)

`ph.0.10` Soil pH in 0-10 cm depth.

`ph.10.30` Soil pH in 10-30 cm depth.

`ph.30.50` Soil pH in 30-50 cm depth.

`ph.50.100` Soil pH in 50-100 cm depth.

`timeset` Factor with range of sampling year and label for sampling type for soil pH. no label: $CaCl_2$ laboratory measurements, `field`: field estimate by indicator solution, `ptf`: H_20 laboratory measurements transferred by pedotransfer function (univariate linear regression) to level of $CaCl_2$ measures.

`cl_mt_etap_pe` columns 14 to 238 contain environmental covariates representing soil forming factors. For more information see Details below.

`cl_mt_etap_ro`

`cl_mt_gh_1`

```
cl_mt_gh_10

cl_mt_gh_11

cl_mt_gh_12

cl_mt_gh_2

cl_mt_gh_3

cl_mt_gh_4

cl_mt_gh_5

cl_mt_gh_6

cl_mt_gh_7

cl_mt_gh_8

cl_mt_gh_9

cl_mt_gh_y

cl_mt_pet_pe

cl_mt_pet_ro

cl_mt_rr_1

cl_mt_rr_10

cl_mt_rr_11

cl_mt_rr_12

cl_mt_rr_2

cl_mt_rr_3

cl_mt_rr_4

cl_mt_rr_5

cl_mt_rr_6

cl_mt_rr_7

cl_mt_rr_8

cl_mt_rr_9

cl_mt_rr_y

cl_mt_swb_pe

cl_mt_swb_ro

cl_mt_td_1

cl_mt_td_10

cl_mt_td_11

cl_mt_td_12

cl_mt_td_2

cl_mt_tt_1

cl_mt_tt_11

cl_mt_tt_12
```

cl_mt_tt_3

cl_mt_tt_4

cl_mt_tt_5

cl_mt_tt_6

cl_mt_tt_7

cl_mt_tt_8

cl_mt_tt_9

cl_mt_tt_y

ge_caco3

ge_geo500h1id

ge_geo500h3id

ge_gt_ch_fil

ge_lgm

ge_vszone

sl_nutr_fil

sl_physio_neu

sl_retention_fil

sl_skelett_r_fil

sl_wet_fil

tr_be_gwn25_hdist

tr_be_gwn25_vdist

tr_be_twi2m_7s_tcilow

tr_be_twi2m_s60_tcilow

tr_ch_3_80_10

tr_ch_3_80_10s

tr_ch_3_80_20s

tr_cindx10_25

tr_cindx50_25

tr_curv_all

tr_curv_plan

tr_curv_prof

tr_enessk

tr_es25

tr_flowlength_up

tr_global_rad_ch

tr_lsf

tr_mrrtf25

tr_mrvbf25

tr_ndom_veg2m_fm

tr_nego

tr_nnessk

tr_ns25

tr_ns25_145mn

tr_ns25_145sd

tr_ns25_75mn

tr_ns25_75sd

tr_poso

tr_protindx

tr_se_alti10m_c

tr_se_alti25m_c

tr_se_alti2m_fmean_10c

tr_se_alti2m_fmean_25c

tr_se_alti2m_fmean_50c

tr_se_alti2m_fmean_5c

tr_se_alti2m_std_10c

tr_se_alti2m_std_25c

tr_se_alti2m_std_50c

tr_se_alti2m_std_5c

tr_se_alti50m_c

tr_se_alti6m_c

tr_se_conv2m

tr_se_curv10m

tr_se_curv25m

tr_se_curv2m

tr_se_curv2m_s15

tr_se_curv2m_s30

tr_se_curv2m_s60

tr_se_curv2m_s7

tr_se_curv2m_std_10c

tr_se_curv2m_std_25c

tr_se_curv2m_std_50c

tr_se_curv2m_std_5c

tr_se_curv50m

tr_se_curv6m

tr_se_curvplan10m

tr_se_curvplan25m

tr_se_curvplan2m

tr_se_curvplan2m_grass_17c

tr_se_curvplan2m_grass_45c

tr_se_curvplan2m_grass_9c

tr_se_curvplan2m_s15

tr_se_curvplan2m_s30

tr_se_curvplan2m_s60

tr_se_curvplan2m_s7

tr_se_curvplan2m_std_10c

tr_se_curvplan2m_std_25c

tr_se_curvplan2m_std_50c

tr_se_curvplan2m_std_5c

tr_se_curvplan50m

tr_se_curvplan6m

tr_se_curvprof10m

tr_se_curvprof25m

tr_se_curvprof2m

tr_se_curvprof2m_grass_17c

tr_se_curvprof2m_grass_45c

tr_se_curvprof2m_grass_9c

tr_se_curvprof2m_s15

tr_se_curvprof2m_s30

tr_se_curvprof2m_s60

tr_se_curvprof2m_s7

tr_se_curvprof2m_std_10c

tr_se_curvprof2m_std_25c

tr_se_curvprof2m_std_50c

tr_se_curvprof2m_std_5c

tr_se_curvprof50m

tr_se_curvprof6m

tr_se_diss2m_10c

tr_se_diss2m_25c

tr_se_diss2m_50c

tr_se_diss2m_5c

tr_se_e_aspect10m

tr_se_e_aspect25m

tr_se_e_aspect2m

tr_se_e_aspect2m_10c

tr_se_e_aspect2m_25c

tr_se_e_aspect2m_50c

tr_se_e_aspect2m_5c

tr_se_e_aspect2m_grass_17c

tr_se_e_aspect2m_grass_45c

tr_se_e_aspect2m_grass_9c

tr_se_e_aspect50m

tr_se_e_aspect6m

tr_se_mrrtf2m

tr_se_mrvbf2m

tr_se_n_aspect10m

tr_se_n_aspect25m

tr_se_n_aspect2m

tr_se_n_aspect2m_10c

tr_se_n_aspect2m_25c

tr_se_n_aspect2m_50c

tr_se_n_aspect2m_5c

tr_se_n_aspect2m_grass_17c

tr_se_n_aspect2m_grass_45c

tr_se_n_aspect2m_grass_9c

tr_se_n_aspect50m

tr_se_n_aspect6m

tr_se_no2m_r500

tr_se_po2m_r500

tr_se_rough2m_10c

tr_se_rough2m_25c

tr_se_rough2m_50c

tr_se_rough2m_5c

tr_se_rough2m_rect3c

tr_se_sar2m

tr_se_sca2m

tr_se_slope10m

tr_se_slope25m

tr_se_slope2m

tr_se_slope2m_grass_17c

tr_se_slope2m_grass_45c

tr_se_slope2m_grass_9c

tr_se_slope2m_s15

tr_se_slope2m_s30

tr_se_slope2m_s60

tr_se_slope2m_s7

tr_se_slope2m_std_10c

tr_se_slope2m_std_25c

tr_se_slope2m_std_50c

tr_se_slope2m_std_5c

tr_se_slope50m

tr_se_slope6m

tr_se_toposcale2m_r3_r50_i10s

tr_se_tpi_2m_10c

tr_se_tpi_2m_25c

tr_se_tpi_2m_50c

tr_se_tpi_2m_5c

tr_se_tri2m_altern_3c

tr_se_tsc10_2m

tr_se_twi2m

tr_se_twi2m_s15

tr_se_twi2m_s30

tr_se_twi2m_s60

tr_se_twi2m_s7

tr_se_vrm2m

tr_se_vrm2m_r10c

tr_slope25_l2g

tr_terrtextur

tr_tpi2000c

tr_tpi5000c

tr_tpi500c

tr_tsc25_18

tr_tsc25_40

tr_twi2

tr_twi_normal

tr_vdcn25

Details

Soil data

The soil data originates from various soil sampling campaigns since 1968. Most of the data was collected in conventional soil surveys in the 1970ties in the course of amelioration and farm land exchanges. As frequently observed in legacy soil data sampling site allocation followed a purposive sampling strategy identifying typical soils in an area in the course of polygon soil mapping.

`dclass` contains drainage classes of three levels. Swiss soil classification differentiates stagnic (I), gleyic (G) and anoxic/reduced (R) soil profile qualifiers with each 4, 6 resp. 5 levels. To reduce complexity the qualifiers I, G and R were aggregated to the degree of hydromorphic characteristic of a site with the ordered levels `well` (qualifier labels I1–I2, G1–G3, R1 and no hydromorphic qualifier), `moderate` well drained (I3–I4, G4) and `poor` drained (G5–G6, R2–R5).

`waterlog` indicates the `presence` or `absence` of waterlogging characteristics down 30, 50 and 100 cm soil depth. The responses were based on horizon qualifiers 'gg' or 'r' of the Swiss classification (*Brunner et al. 1997*) as those were considered to limit plant growth. A horizon was given the qualifier 'gg' if it was strongly gleyic predominantly oxidized (rich in Fe^{3+}) and 'r' if it was anoxic predominantly reduced (poor in Fe^{3+}).

`pH` was mostly sampled following genetic soil horizons. To ensure comparability between sites pH was transferred to fixed depth intervals of 0–10, 10–30, 30–50 and 50–100 cm by weighting soil horizons falling into a given interval. The data contains laboratory measurements that solved samples in $CaCl_2$ or H_20. The latter were transferred to the level of $CaCl_2$ measurements by univariate linear regression (label `ptf` in `timeset`). Further, the dataset contains estimates of pH in the field by an indicator solution (Hellige pH, label `field` in `timeset`). The column `timeset` can be used to partly correct for the long sampling period and the different sampling methods.

Environmental covariates

The numerous covariates were assembled from the available spatial data in the case study area. Each covariate name was given a prefix:

- `cl_` climate covariates as precipitation, temperature, radiation
- `tr_` terrain attributes, covariates derived from digital elevation models
- `ge_` covariates from geological maps
- `sl_` covariates from an overview soil map

References to the used datasets can be found in *Nussbaum et al. 2017b.*

References

Brunner, J., Jaeggli, F., Nievergelt, J., and Peyer, K. (1997). Kartieren und Beurteilen von Landwirtschaftsboeden. FAL Schriftenreihe 24, Eidgenoessische Forschungsanstalt fuer Agraroekologie und Landbau, Zuerich-Reckenholz (FAL).

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E., and Papritz, A., 2017b. Evaluation of digital soil mapping approaches with large sets of environmental covariates, SOIL Discuss., https://www.soil-discuss.net/soil-2017-14/, in review.

Examples

```
data(berne)
```

berne.grid *Berne – very small extract of prediction grid*

Description

The Berne grid dataset contains values of spatial covariates on the nodes of a 20 m grid. The dataset is intended for spatial continouous predictions of soil properties modelled from the sampling locations in berne dataset.

Usage

```
data("berne")
```

Format

A data frame with 4594 observations on the following 228 variables.

`id` node identifier number.

`x` easting, Swiss grid in m, EPSG: 21781 (CH1903/LV03)

`y` northing, Swiss grid in m, EPSG: 21781 (CH1903/LV03)

`cl_mt_etap_pe` columns 4 to 228 contain environmental covariates representing soil forming factors. For more information see `Details` in berne.

`cl_mt_etap_ro`

`cl_mt_gh_1`

`cl_mt_gh_10`

`cl_mt_gh_11`

`cl_mt_gh_12`

`cl_mt_gh_2`

`cl_mt_gh_3`

`cl_mt_gh_4`

`cl_mt_gh_5`

`cl_mt_gh_6`

`cl_mt_gh_7`

`cl_mt_gh_8`

`cl_mt_gh_9`

`cl_mt_gh_y`

`cl_mt_pet_pe`

`cl_mt_pet_ro`

`cl_mt_rr_1`

`cl_mt_rr_10`

cl_mt_rr_11

cl_mt_rr_12

cl_mt_rr_2

cl_mt_rr_3

cl_mt_rr_4

cl_mt_rr_5

cl_mt_rr_6

cl_mt_rr_7

cl_mt_rr_8

cl_mt_rr_9

cl_mt_rr_y

cl_mt_swb_pe

cl_mt_swb_ro

cl_mt_td_1

cl_mt_td_10

cl_mt_td_11

cl_mt_td_12

cl_mt_td_2

cl_mt_tt_1

cl_mt_tt_11

cl_mt_tt_12

cl_mt_tt_3

cl_mt_tt_4

cl_mt_tt_5

cl_mt_tt_6

cl_mt_tt_7

cl_mt_tt_8

cl_mt_tt_9

cl_mt_tt_y

ge_caco3

ge_geo500h1id

ge_geo500h3id

ge_gt_ch_fil

ge_lgm

ge_vszone

sl_nutr_fil

sl_physio_neu

```
sl_retention_fil
sl_skelett_r_fil
sl_wet_fil
tr_be_gwn25_hdist
tr_be_gwn25_vdist
tr_be_twi2m_7s_tcilow
tr_be_twi2m_s60_tcilow
tr_ch_3_80_10
tr_ch_3_80_10s
tr_ch_3_80_20s
tr_cindx10_25
tr_cindx50_25
tr_curv_all
tr_curv_plan
tr_curv_prof
tr_enessk
tr_es25
tr_flowlength_up
tr_global_rad_ch
tr_lsf
tr_mrrtf25
tr_mrvbf25
tr_ndom_veg2m_fm
tr_nego
tr_nnessk
tr_ns25
tr_ns25_145mn
tr_ns25_145sd
tr_ns25_75mn
tr_ns25_75sd
tr_poso
tr_protindx
tr_se_alti10m_c
tr_se_alti25m_c
tr_se_alti2m_fmean_10c
tr_se_alti2m_fmean_25c
tr_se_alti2m_fmean_50c
```

tr_se_alti2m_fmean_5c

tr_se_alti2m_std_10c

tr_se_alti2m_std_25c

tr_se_alti2m_std_50c

tr_se_alti2m_std_5c

tr_se_alti50m_c

tr_se_alti6m_c

tr_se_conv2m

tr_se_curv10m

tr_se_curv25m

tr_se_curv2m

tr_se_curv2m_s15

tr_se_curv2m_s30

tr_se_curv2m_s60

tr_se_curv2m_s7

tr_se_curv2m_std_10c

tr_se_curv2m_std_25c

tr_se_curv2m_std_50c

tr_se_curv2m_std_5c

tr_se_curv50m

tr_se_curv6m

tr_se_curvplan10m

tr_se_curvplan25m

tr_se_curvplan2m

tr_se_curvplan2m_grass_17c

tr_se_curvplan2m_grass_45c

tr_se_curvplan2m_grass_9c

tr_se_curvplan2m_s15

tr_se_curvplan2m_s30

tr_se_curvplan2m_s60

tr_se_curvplan2m_s7

tr_se_curvplan2m_std_10c

tr_se_curvplan2m_std_25c

tr_se_curvplan2m_std_50c

tr_se_curvplan2m_std_5c

tr_se_curvplan50m

tr_se_curvplan6m

tr_se_curvprof10m

tr_se_curvprof25m

tr_se_curvprof2m

tr_se_curvprof2m_grass_17c

tr_se_curvprof2m_grass_45c

tr_se_curvprof2m_grass_9c

tr_se_curvprof2m_s15

tr_se_curvprof2m_s30

tr_se_curvprof2m_s60

tr_se_curvprof2m_s7

tr_se_curvprof2m_std_10c

tr_se_curvprof2m_std_25c

tr_se_curvprof2m_std_50c

tr_se_curvprof2m_std_5c

tr_se_curvprof50m

tr_se_curvprof6m

tr_se_diss2m_10c

tr_se_diss2m_25c

tr_se_diss2m_50c

tr_se_diss2m_5c

tr_se_e_aspect10m

tr_se_e_aspect25m

tr_se_e_aspect2m

tr_se_e_aspect2m_10c

tr_se_e_aspect2m_25c

tr_se_e_aspect2m_50c

tr_se_e_aspect2m_5c

tr_se_e_aspect2m_grass_17c

tr_se_e_aspect2m_grass_45c

tr_se_e_aspect2m_grass_9c

tr_se_e_aspect50m

tr_se_e_aspect6m

tr_se_mrrtf2m

tr_se_mrvbf2m

tr_se_n_aspect10m

tr_se_n_aspect25m

tr_se_n_aspect2m

tr_se_n_aspect2m_10c

tr_se_n_aspect2m_25c

tr_se_n_aspect2m_50c

tr_se_n_aspect2m_5c

tr_se_n_aspect2m_grass_17c

tr_se_n_aspect2m_grass_45c

tr_se_n_aspect2m_grass_9c

tr_se_n_aspect50m

tr_se_n_aspect6m

tr_se_no2m_r500

tr_se_po2m_r500

tr_se_rough2m_10c

tr_se_rough2m_25c

tr_se_rough2m_50c

tr_se_rough2m_5c

tr_se_rough2m_rect3c

tr_se_sar2m

tr_se_sca2m

tr_se_slope10m

tr_se_slope25m

tr_se_slope2m

tr_se_slope2m_grass_17c

tr_se_slope2m_grass_45c

tr_se_slope2m_grass_9c

tr_se_slope2m_s15

tr_se_slope2m_s30

tr_se_slope2m_s60

tr_se_slope2m_s7

tr_se_slope2m_std_10c

tr_se_slope2m_std_25c

tr_se_slope2m_std_50c

tr_se_slope2m_std_5c

tr_se_slope50m

tr_se_slope6m

tr_se_toposcale2m_r3_r50_i10s

tr_se_tpi_2m_10c

tr_se_tpi_2m_25c

tr_se_tpi_2m_50c

tr_se_tpi_2m_5c

tr_se_tri2m_altern_3c

tr_se_tsc10_2m

tr_se_twi2m

tr_se_twi2m_s15

tr_se_twi2m_s30

tr_se_twi2m_s60

tr_se_twi2m_s7

tr_se_vrm2m

tr_se_vrm2m_r10c

tr_slope25_l2g

tr_terrtextur

tr_tpi2000c

tr_tpi5000c

tr_tpi500c

tr_tsc25_18

tr_tsc25_40

tr_twi2

tr_twi_normal

tr_vdcn25

Details

Due to CRAN file size restrictions the grid for spatial predictions only shows a very small excerpt of the original study area.

The environmental covariates for prediction of soil properties from dataset berne were extracted at the nodes of a 20 m grid. For higher resolution geodata sets no averaging over the area of the 20x20 pixel was done. Berne.grid therefore has the same spatial support for each covariate as berne.

For more information on the environmental covariates see berne.

References

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E., and Papritz, A.: Evaluation of digital soil mapping approaches with large sets of environmental covariates, SOIL, 4, 1-22, doi:10.5194/soil-4-1-2018, 2018.

Examples

```
data(berne.grid)
```

`bootstrap.geoGAM` *Bootstrapped predictive distribution*

Description

Method for class `geoGAM` to compute model based bootstrap for point predictions. Returns complete predictive distribution of which prediction intervals can be computed.

Usage

```
## Default S3 method:
bootstrap(object, ...)

## S3 method for class 'geoGAM'
bootstrap(object, newdata, R = 100,
          back.transform = c("none", "log", "sqrt"),
          seed = NULL, cores = detectCores(), ...)
```

Arguments

`object`	geoGAM object
`newdata`	data frame in which to look for covariates with which to predict.
`R`	number of bootstrap replicates, single positive integer.
`back.transform`	sould to `log` or `sqrt` transformed responses unbiased back transformation be applied? Default is `none`.
`seed`	seed for simulation of new response. Set seed for reproducible results.
`cores`	number of cores to be used for parallel computing.
`...`	further arguments.

Details

Soil properties are predicted for new locations $\mathbf{s}_+$ from the final `geoGAM` fit by $\tilde{Y}(\mathbf{s}_+) = \hat{f}(\mathbf{x}(\mathbf{s}_+))$, see function `predict.geoGAM`. To model the predictive distributions for continuous responses `bootstrap.geoGAM` uses a non-parametric, model-based bootstrapping approach (*Davison and Hinkley 1997*, pp. 262, 285) as follows:

1. New values of the response are simulated according to $Y(\mathbf{s})^* = \hat{f}(\mathbf{x}(\mathbf{s})) + \epsilon$, where $\hat{f}(\mathbf{x}(\mathbf{s}))$ are the fitted values of the final model and ϵ are errors randomly sampled with replacement from the centred, homoscedastic residuals of the final model *Wood 2006*, p. 129).
2. `geoGAM` is fitted to $Y(\mathbf{s})^*$.
3. Prediction errors are computed according to $\delta_+^* = \hat{f}(\mathbf{x}(\mathbf{s}_+))^* - (\,\hat{f}(\mathbf{x}(\mathbf{s}_+)) + \epsilon\,)$, where $\hat{f}(\mathbf{x}(\mathbf{s}_+))^*$ are predicted values at new locations $\mathbf{s}_+$ of the model built with the simulated response $Y(\mathbf{s})^*$ in step B above, and the errors ϵ are again randomly sampled from the centred, homoscedastic residuals of the final model (see step A).

Prediction intervals are computed according to

$$[\hat{f}(\mathbf{x}(\mathbf{s}_+)) - \delta^*_{+\,(1-\alpha)}\,;\hat{f}(\mathbf{x}(\mathbf{s}_+)) - \delta^*_{+\,(\alpha)}]$$

where $\delta^*_{+\,(\alpha)}$ and $\delta^*_{+\,(1-\alpha)}$ are the α- and $(1-\alpha)$-quantiles of δ^*_+, pooled over all 1000 bootstrap repetitions.

Predictive distributions for binary and ordinal responses are directly obtained from a final geoGAM fit by predicting probabilities of occurrence $\widetilde{\mathrm{Prob}}(Y(\mathbf{s}) = r \,|\, \mathbf{x}(\mathbf{s}))$ (*Davison and Hinkley 1997*, p. 358).

Value

Data frame of `nrows(newdata)` rows and `R + 2` columns with `x` and `y` indicating coordinates of the location and `P1` to `P...R` the prediction at this location from `1...R` replications.

Author(s)

M. Nussbaum

References

Nussbaum, M., Walthert, L., Fraefel, M., Greiner, L., and Papritz, A.: Mapping of soil properties at high resolution in Switzerland using boosted geoadditive models, SOIL, 3, 191-210, doi:10.5194/soil-3-191-2017, 2017.

Davison, A. C. and Hinkley, D. V., 2008. Bootstrap Methods and Their Applications. Cambridge University Press.

See Also

To create geoGAM objects see `geoGAM` and to predict without simulation of the predictive distribution see `predict.geoGAM`.

Examples

```
data(quakes)

# group stations to ensure min 20 observations per factor level
# and reduce number of levels for speed
quakes$stations <- factor( cut( quakes$stations, breaks = c(0,15,19,23,30,39,132)) )

# Artificially split data to create prediction data set
set.seed(1)
quakes.pred <- quakes[ ss <- sample(1:nrow(quakes), 500), ]
quakes <- quakes[ -ss, ]

quakes.geogam <- geoGAM(response = "mag",
                        covariates = c("stations", "depth"),
                        coords = c("lat", "long"),
                        data = quakes,
                        max.stop = 20,
                          cores = 1)


## compute model based bootstrap with 10 repetitions (use at least 100)
quakes.boot <- bootstrap(quakes.geogam,
                         newdata = quakes.pred,
                         R = 10, cores = 1)


# plot predictive distribution for site in row 9
hist( as.numeric( quakes.boot[ 9, -c(1:2)] ), col = "grey",
    main = paste("Predictive distribution at", paste( quakes.boot[9, 1:2], collapse = "/" )),
      xlab = "predicted magnitude")

# compute 95 % prediction interval and add to plot
quant95 <- quantile( as.numeric( quakes.boot[ 9, -c(1:2)] ), probs = c(0.025, 0.975) )
abline(v = quant95[1], lty = "dashed")
abline(v = quant95[2], lty = "dashed")
```

geoGAM *Select sparse geoadditive model*

Description

Selects a parsimonious geoadditive model from a large set of covariates with the aim of (spatial) prediction.

Usage

```
geoGAM(response, covariates = names(data)[!(names(data) %in% c(response,coords))],
       data, coords = NULL, weights = rep(1, nrow(data)),
       offset = TRUE, max.stop = 300, non.stationary = FALSE,
       sets = NULL, seed = NULL, validation.data = NULL,
       verbose = 0, cores = min(detectCores(),10))
```

Arguments

response	name of response as character. Responses currently supported: gaussian, binary, ordered.
covariates	character vector of all covariates (factor, continuous). If not given, all columns of data are used.
data	data frame containing response, coordinates and covariates.
coords	character vector of column names indicating spatial coordinates.
weights	weights used for model fitting.
offset	logical, use offset for component wise gradient boosting algorithm.

`max.stop` maximal number of boosting iterations.

`non.stationary` logical, include non-stationary effects in model selection. This allows for spatial varying coefficients for continuous covariates, but increases computational effort.

`sets` give predefined cross validation sets.

`seed` set random seed for splitting of the cross validation sets, if no `sets` are given.

`validation.data`
data frame containing response, coordinates and covariates to compute independent validation statistics. This data set is used to calculate predictive performance at the end of model selection only.

`verbose` Should screen output be generated? 0 = none, >0 create output.

`cores` number of cores to be used for parallel computing

Details

Summary

`geoGAM` models smooth nonlinear relations between responses and single covariates and combines these model terms additively. Residual spatial autocorrelation is captured by a smooth function of spatial coordinates and nonstationary effects are included by interactions between covariates and smooth spatial functions. The core of fully automated model building for geoGAM is componentwise gradient boosting. The model selection procedures aims at obtaining sparse models that are open to check feasibilty of modelled relationships (*Nussbaum et al. 2017a*).

`geoGAM` to date models continuous, binary and ordinal responses. It is able to cope with numerous continuous and categorical covariates.

Generic model representation

GAM expand the (possibly transformed) conditional expectation of a response at given covariates s as an additive series

$$g\left(\mathrm{E}[Y(\mathbf{s})\,|\,\mathbf{x}(\mathbf{s})]\right) = \nu + f(\mathbf{x}(\mathbf{s})) = \nu + \sum_j f_j(x_j(\mathbf{s})),$$

where ν is a constant and $f_j(x_j(\mathbf{s}))$ are linear terms or unspecified “smooth” nonlinear functions of single covariates $x_j(\mathbf{s})$ (e.g. smoothing spline, kernel or any other scatterplot smoother) and $g(\cdot)$ is again a link function. A generalized additive model (GAM) is based on the following components (*Hastie and Tibshirani 1990, Chapt. 6*):

1. *Response distribution*: Given $\mathbf{x}(\mathbf{s}) = x_1(\mathbf{s}), x_2(\mathbf{s}), ..., x_p(\mathbf{s})$, the $Y(\mathbf{s})$ are conditionally independent observations from simple exponential family distributions.
2. *Link function*: $g(\cdot)$ relates the expectation $\mu(\mathbf{x}(\mathbf{s})) = \mathrm{E}[Y(\mathbf{s})|\mathbf{x}(\mathbf{s})]$ of the response distribution to
3. the *additive predictor* $\sum_j f_j(x_j(\mathbf{s}))$.

geoGAM extend GAM by allowing a more complex form of the additive predictor (*Kneib et al. 2009*, *Hothorn et al. 2011*): First, one can add a smooth function $f_{\mathbf{s}}(\mathbf{s})$ of the spatial coordinates (smooth spatial surface) to the additive predictor to account for residual autocorrelation. More complex relationships between Y and $\mathbf{x}$ can be modelled by adding terms like $f_j(x_j(\mathbf{s})) \cdot f_k(x_k(\mathbf{s}))$

– capturing the effect of interactions between covariates – and $f_{\mathbf{s}}(\mathbf{s}) \cdot f_j(x_k(\mathbf{s}))$ – accounting for spatially changing dependence between Y and $\mathbf{x}$. Hence, in its full generality, a generalized additive model for spatial data is represented by

$$g(\mu(\mathbf{x}(\mathbf{s}))) = \nu + f(\mathbf{x}(\mathbf{s})) =$$

$$\nu + \underbrace{\sum_u f_{j_u}(x_{j_u}(\mathbf{s})) + \sum_v f_{j_v}(x_{j_v}(\mathbf{s})) \cdot f_{k_v}(x_{k_v}(\mathbf{s}))}_{\text{global marginal and interaction effects}}$$

$$+ \underbrace{\sum_w f_{\mathbf{s}_w}(\mathbf{s}) \cdot f_{j_w}(x_{j_w}(\mathbf{s}))}_{\text{nonstationary effects}} + \underbrace{f_{\mathbf{s}}(\mathbf{s})}_{\text{autocorrelation}} .$$

Kneib et al. (2009) called the above equation a geoadditive model, a name coined before by *Kammann and Wand 2003* for a combination of additive models with a geostatistical error model. It remains to specify what response distributions and link functions should be used for the various response types: For (possibly transformed) *continuous* responses one uses often a normal response distribution combined with the identity link $g\left(\mu(\mathbf{x}(\mathbf{s}))\right) = \mu(\mathbf{x}(\mathbf{s}))$. For binary data (coded as 0 and 1), one assumes a Bernoulli distribution and uses often a logit link

$$g\left(\mu(\mathbf{x}(\mathbf{s}))\right) = \log\left(\frac{\mu(\mathbf{x}(\mathbf{s}))}{1-\mu(\mathbf{x}(\mathbf{s}))}\right),$$

where

$$\mu(\mathbf{x}(\mathbf{s})) = \mathrm{Prob}[Y(\mathbf{s}) = 1 \,|\, \mathbf{x}(\mathbf{s})] = \frac{\exp(\nu + f(\mathbf{x}(\mathbf{s})))}{1+\exp(\nu + f(\mathbf{x}(\mathbf{s})))}.$$

For ordinal data, with ordered response levels, $1, 2, \ldots, k$, the cumulative logit or proportional odds model (*Tutz 2012*, Sect. 9.1) is used. For any given level $r \in (1, 2, \ldots, k)$, the logarithm of the odds of the event $Y(\mathbf{s}) \leq r \,|\, \mathbf{x}(\mathbf{s})$ is then modelled by

$$\log\left(\frac{\mathrm{Prob}[Y(\mathbf{s}) \leq r \,|\, \mathbf{x}(\mathbf{s}))]}{\mathrm{Prob}[Y(\mathbf{s}) > r \,|\, \mathbf{x}(\mathbf{s}))]}\right) = \nu_r + f(\mathbf{x}(\mathbf{s})),$$

with ν_r a sequence of level-specific constants satisfying $\nu_1 \leq \nu_2 \leq \ldots \leq \nu_r$. Conversely,

$$\mathrm{Prob}[Y(\mathbf{s}) \leq r \,|\, \mathbf{x}(\mathbf{s})] = \frac{\exp(\nu_r + f(\mathbf{x}(\mathbf{s})))}{1+\exp(\nu_r + f(\mathbf{x}(\mathbf{s})))}.$$

Note that $\mathrm{Prob}[Y(\mathbf{s}) \leq r \,|\, \mathbf{x}(\mathbf{s})]$ depends on r only through the constant ν_r. Hence, the ratio of the odds of two events $Y(\mathbf{s}) \leq r \,|\, \mathbf{x}(\mathbf{s})$ and $(\mathbf{s}) \leq r \,|\, \tilde{\mathbf{x}}(\mathbf{s})$ is the same for all r (*Tutz 2012*, p. 245).

Model building (selection of covariates)

To build parsimonious models that can readily be checked for agreement understanding in regards to the analized subject. The following steps 1–6 are implemented in `geoGAM` toa achieve sparse models in a fully automated way. In several of these steps tuning parameters are optimized by 10-fold cross-validation with fixed subsets using either root mean squared error (RMSE), continuous responses), Brier score (BS), binary responses) or ranked probability score (RPS), ordinal responses) as optimization criteria (see *Wilks, 2011*). To improve the stability of the algorithm continuous covariates are first scaled (by difference of maximum and minimum value) and centred.

1. Boosting (see step 2 below) is more stable and converges more quickly when the effects of categorical covariates (factors) are accounted for as model offset. Therefore, the group lasso (least absolute shrinkage and selection operator, *Breheny and Huang 2015*, grpreg)) – an algorithm that likely excludes non-relevant covariates and treats factors as groups – is used to select important factors for the offset. For ordinal responses stepwise proportional odds logistic regression in both directions with BIC (e. g. *Faraway 2005*, p. 126) is used to select the offset covariates because lasso cannot be used for such responses.

2. Next, a subset of relevant factors, continuous covariates and spatial effects is selected by componentwise gradient boosting. Boosting is a slow stagewise additive learning algorithm. It expands $f(\mathbf{x}(\mathbf{s}))$ in a set of base procedures (baselearners) and approximates the additive predictor by a finite sum of them as follows (*Buehlmann and Hothorn 2007*):

   (a) Initialize $\hat{f}(\mathbf{x}(\mathbf{s}))^{[m]}$ with offset of step 1 above and set $m = 0$.

   (b) Increase m by 1. Compute the negative gradient vector $\mathbf{U}^{[m]}$ (e.g. residuals) for a loss function $l(\cdot)$.

   (c) Fit all baselearners $g(\mathbf{x}(\mathbf{s}))_{1..p}$ to $\mathbf{U}^{[m]}$ and select the baselearner, say $g(\mathbf{x}(\mathbf{s}))_j^{[m]}$ that minimizes $l(\cdot)$.

   (d) Update $\hat{f}(\mathbf{x}(\mathbf{s}))^{[m]} = \hat{f}(\mathbf{x}(\mathbf{s}))^{[m-1]} + v \cdot g(\mathbf{x}(\mathbf{s}))_j^{[m]}$ with step size $v \leq 1$.

   (e) Iterate steps (b) to (d) until $m = m_{stop}$ (main tuning parameter).

   The following settings are used in above algorithm: As loss functions $l(\cdot)$ L_2 is used for continuous, negative binomial likelihood for binary (*Buehlmann and Hothorn 2007*) and proportional odds likelihood for ordinal responses (*Schmid et al. 2011*). Early stopping of the boosting algorithm is achieved by determining optimal m_{stop} by cross-validation. Default step length ($v = 0.1$) is used. This is not a sensitive parameter as long as it is clearly below 1 (*Hofner et al. 2014*). For continuous covariates penalized smoothing spline baselearners (*Kneib et al. 2009*) are used. Factors are treated as linear baselearners. To capture residual autocorrelation a bivariate tensor-product P-spline of spatial coordinates (*Wood 2006, pp. 162*) is added to the additive predictor. Spatially varying effects are modelled by baselearners formed by multiplication of continuous covariates with tensor-product P-splines of spatial coordinates (*Wood 2006, pp. 168*). Uneven degree of freedom of baselearners biases baselearner selection (*Hofner et al. 2011b*). Therefore, each baselearner is penalized to 5 degrees of freedom (df). Factors with less than 6 levels ($df < 5$) are aggregated to grouped baselearners. By using an offset, effects of important factors with more than 6 levels are implicitly accounted for without penalization.

3. At m_{stop} (see step 2 above), many included baselearners may have very small effects only. To remove these the effect size e_j of each baselearner $f_j(x_j(\mathbf{s}))$ is computed. For factors the effect size e_j is the largest difference between effects of two levels and for continuous covariates it is equal to the maximum contrast of estimated partial effects (after removal of extreme values as in boxplots, *Frigge et al. 1989*). Generalized additive models (GAM, *Wood 2011*) are fitted including smooth and factor effects depending on the effect size e_j of the corresponding baselearner j. The procedure iterates through e_j and excludes covariates with e_j smaller than a threshold effect size e_t. Optimal e_t is determined by 10-fold cross-validation of GAM. In these GAM fits smooth effects are penalized to 5 degrees of freedom as imposed by componentwise gradient boosting (step 2 above). The factors selected as offset in step 1 are included in the main GAM, that is now fitted without offset.

4. The GAM is further reduced by stepwise removal of covariates by cross-validation. The candidate covariate to drop is chosen by largest p value of F tests for linear factors and approximate F test (*Wood 2011*) for smooth terms.

5. Factor levels with similar estimated effects are merged stepwise again by cross-validation based on largest p values from two sample t-tests of partial residuals.
6. The final model (used to compute spatial predictions) results ideally in a parsimonious GAM. Because of step 5, factors have possibly a reduced number of coefficients. Effects of continuous covariates are modelled by smooth functions and – if at all present – spatially structured residual variation (autocorrelation) is represented by a smooth spatial surface. To avoid overfitting both types of smooth effects are penalized to 5 degrees of freedom (as imposed by step 2).

Value

Object of class `geoGAM`:

`offset.grplasso` Cross validation for grouped LASSO, object of class `cv.grpreg` of package `grpreg`). Empty for `offset` = `FALSE`.

`offset.factors` Character vector of factor names chosen for the offset computation. Empty for `offset` = `FALSE`.

`gamboost` Gradient boosting with smooth components, object of class `gamboost` of package `mboost`.

`gamboost.cv` Cross validation for gradient boosting, object of class `cvrisk` of package `mboost`.

`gamboost.mstop` Mstop used for gamboost.

`gamback.cv` List of cross validation error for tuning parameter magnitude.

`gamback.backward` List of cross validation error path for backward selection of `gam` fit.

`gamback.aggregation` List(s) of cross validation error path for aggregation of factor levels.

`gam.final` Final selected geoadditive model fit, object of class `gam`.

`gam.final.cv` Data frame with original response and cross validation predictions.

`gam.final.extern` Data frame with original response data and predictions of `gam.final`.

`data` Original data frame for model calibration.

`parameters` List of parameters handed to geoGAM (used for subsequent bootstrap of prediction intervals).

Author(s)

M. Nussbaum

References

Breheny, P. and Huang, J., 2015. Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. Statistics and Computing, 25, 173–187.

Buehlmann, P. and Hothorn, T., 2007. Boosting algorithms: Regularization, prediction and model fitting, Stat Sci, 22, 477–505, doi:10.1214/07-sts242.

Faraway, J. J., 2005. Linear Models with R, vol. 63 of Texts in Statistical Science, Chapman & Hall/CRC, Boca Raton.

Frigge, M., Hoaglin, D. C., and Iglewicz, B., 1989. Some implementations of the boxplot. The American Statistician, 43(1), 50–54.

Hastie, T. J. and Tibshirani, R. J., 1990. Generalized Additive Models, vol. 43 of Monographs on Statistics and Applied Probability, Chapman and Hall, London.

Hofner, B., Hothorn, T., Kneib, T., and Schmid, M., 2011. A framework for unbiased model selection based on boosting. Journal of Computational and Graphical Statistics, 20(4), 956–971.

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M., 2014. Model-based boosting in R: A hands-on tutorial using the R package mboost, Computation Stat, 29, 3–35, doi:10.1007/s00180-012-0382-5.

Hothorn, T., Mueller, J., Schroder, B., Kneib, T., and Brandl, R., 2011. Decomposing environmental, spatial, and spatiotemporal components of species distributions, Ecol Monogr, 81, 329–347.

Kneib, T., Hothorn, T., and Tutz, G., 2009. Variable selection and model choice in geoadditive regression models. Biometrics, 65(2), 626–634.

Nussbaum, M., Walthert, L., Fraefel, M., Greiner, L., and Papritz, A.: Mapping of soil properties at high resolution in Switzerland using boosted geoadditive models, SOIL, 3, 191-210, doi:10.5194/soil-3-191-2017, 2017.

Schmid, M., Hothorn, T., Maloney, K. O., Weller, D. E., and Potapov, S., 2011. Geoadditive regression modeling of stream biological condition, Environ Ecol Stat, 18, 709–733, doi:10.1007/s10651-010-0158-4.

Tutz, G., 2012, Regression for Categorical Data, Cambridge University Press, doi:10.1017/cbo9780511842061.

Wilks, D. S., 2011. Statistical Methods in the Atmospheric Sciences, Academic Press, 3 edn.

Wood, S. N., 2006. Generalized Additive Models: An Introduction with R, Chapman and Hall/CRC.

Wood, S. N., 2011. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B), 73(1), 3–36.

See Also

The model selection is based on packages `grpreg` (function `cv.grpreg`), `MASS` (function `polr`), `mboost` (functions `gamboost`, `cv`, `cvrisk`) and `mgcv` (function `gam`). For further information please see documentation and vignettes for these packages.

Examples

```
### small examples with earthquake data

data(quakes)
set.seed(2)
quakes <- quakes[ sample(1:nrow(quakes), 50), ]

quakes.geogam <- geoGAM(response = "mag",
                        covariates = c("depth", "stations"),
                        data = quakes,
                        seed = 2,
                        max.stop = 5,
                        cores = 1)
summary(quakes.geogam)


data(quakes)

# create grouped factor with reduced number of levels
quakes$stations <- factor( cut( quakes$stations, breaks = c(0,15,19,23,30,39,132)) )

quakes.geogam <- geoGAM(response = "mag",
                        covariates = c("stations", "depth"),
                        coords = c("lat", "long"),
                        data = quakes,
                        max.stop = 10,
                        cores = 1)

summary(quakes.geogam)
summary(quakes.geogam, what = "path")




## Use soil data set of soil mapping study area near Berne

data(berne)
set.seed(1)

# Split data sets and
# remove rows with missing values in response and covariates

d.cal <- berne[ berne$dataset == "calibration" & complete.cases(berne), ]
d.val <- berne[ berne$dataset == "validation" & complete.cases(berne), ]


### Model selection for continuous response
ph10.geogam <- geoGAM(response = "ph.0.10",
                      covariates = names(d.cal)[14:ncol(d.cal)],
                      coords = c("x", "y"),
                      data = d.cal,
                      offset = TRUE,
                      sets = mboost::cv(rep(1, nrow(d.cal)), type = "kfold"),
                      validation.data = d.val,
                      cores = 1)
summary(ph10.geogam)
summary(ph10.geogam, what = "path")


### Model selection for binary response
waterlog100.geogam <- geoGAM(response = "waterlog.100",
                             covariates = names(d.cal)[c(14:54, 56:ncol(d.cal))],
                             coords = c("x", "y"),
                             data = d.cal,
                             offset = FALSE,
                         sets = sample( cut(seq(1,nrow(d.cal)),breaks=10,labels=FALSE) ),
                             validation.data = d.val,
                             cores = 1)
summary(waterlog100.geogam)
summary(waterlog100.geogam, what = "path")


### Model selection for ordered response
dclass.geogam <- geoGAM(response = "dclass",
                        covariates = names(d.cal)[14:ncol(d.cal)],
                        coords = c("x", "y"),
                        data = d.cal,
                        offset = TRUE,
                        non.stationary = TRUE,
                        seed = 1,
                        validation.data = d.val,
                        cores = 1)
summary(dclass.geogam)
summary(dclass.geogam, what = "path")
```

methods *Methods for* geoGAM *objects*

Description

Methods for models fitted by geoGAM().

Usage

```
## S3 method for class 'geoGAM'
summary(object, ..., what = c("final", "path"))

## S3 method for class 'geoGAM'
print(x, ...)

## S3 method for class 'geoGAM'
plot(x, ..., what = c("final", "path"))
```

Arguments

object	an object of class geoGAM
x	an object of class geoGAM
...	other arguments passed to summary.gam, plot.gam or plot.mboost
what	print summary or plot partial effects of final selected model or print summary or plot gradient boosting path of model selection path.

Details

`summary` with `what = "final"` calls `summary.gam` to display a summary of the final (geo)additive model. `plot` with `what = "final"` calls `plot.gam` to plot partial residual plots of the final model.

`summary` with `what = "path"` give a summary of covariates selected in each step of model building. `plot` with `what = "path"` calls `plot.mboost` to plot the path of the gradient boosting algorithm.

Value

For `what == "final"` summary returns a list of 3:

`summary.gam`	containing the values of `summary.gam`.
`summary.validation$cv`	cross validation statistics.
`summary.validation$validation`	validation set statistics.

For `what == "path"` summary returns a list of 13:

`response`	name of response.
`family`	family used for `geoGAM` fit.
`n.obs`	number of observations used for model fitting.
`n.obs.val`	number of observations used for model validation.
`n.covariates`	number of initial covariates including factors.
`n.cov.chosen`	number of covariates in final model.
`list.factors`	list of factors chosen as offset.
`mstop`	number of optimal iterations of gradient boosting.
`list.baselearners`	list of covariate names selected by gradient boosting.
`list.effect.size`	list of covariate names after cross validation of effect size in gradient boosting.
`list.backward`	list of covariate names after backward selection.
`list.aggregation`	list of aggregated factor levels.
`list.gam.final`	list of covariate names in final model.

Author(s)

M. Nussbaum

References

Nussbaum, M., Walthert, L., Fraefel, M., Greiner, L., and Papritz, A.: Mapping of soil properties at high resolution in Switzerland using boosted geoadditive models, SOIL, 3, 191-210, doi:10.5194/soil-3-191-2017, 2017.

See Also

geoGAM, gam, predict.gam

Examples

```
### small example with earthquake data

data(quakes)
set.seed(2)

quakes <- quakes[ sample(1:nrow(quakes), 50), ]

quakes.geogam <- geoGAM(response = "mag",
                        covariates = c("depth", "stations"),
                        data = quakes,
                        seed = 2,
                        max.stop = 5,
                        cores = 1)

summary(quakes.geogam)
summary(quakes.geogam, what = "path")

plot(quakes.geogam)
plot(quakes.geogam, what = "path")
```

---

predict.geoGAM *Prediction from fitted geoGAM model*

Description

Takes a fitted geoGAM object and produces point predictions for a new set of covariate values. If no new data is provided fitted values are returned. Centering and scaling is applied with the same parameters as for the calibration data set given to geoGAM. Factor levels are aggregated according to the final model fit.

Usage

```
## S3 method for class 'geoGAM'
predict(object, newdata,
        type = c("response", "link", "probs", "class"),
        back.transform = c("none", "log", "sqrt"),
        threshold = 0.5, se.fit = FALSE, ...)
```

Arguments

object	an object of class geoGAM

`newdata` An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used. If newdata is provided then it should contain all the variables needed for prediction: a warning is generated if not. Factors aggregated by the function `geoGAM` will be aggregated in the same way for prediction within this function.

`type` Type of prediction.

`back.transform` Should to `log` or `sqrt` transformed responses unbiased back transformation be applied? Default is `none`. Ignored for categorical responses.

`threshold` Ignored for `type = c("response", "link", "probs")` and for `type = "class"` for responses with more than two levels.

`se.fit` logical. Default is FALSE.

`...` further arguments to `predict()`.

Details

Returns point predictions for new locations s from linear and smooth trends $\hat{f}(\mathbf{x}, s)$ estimated by penalized least squares geoGAM by calling the function `predict.gam`.

Back transformation of log and sqrt

For lognormal responses (`back.transform = 'log'`) in full analogy to lognormal kriging (*Cressie-2006*, Eq. 20) the predictions are backtransformed by

$$\mathrm{E}[Y(\mathbf{s}) \,|\, \mathbf{x}] = \exp\left(\hat{f}(\mathbf{x}(\mathbf{s})) + \frac{1}{2}\hat{\sigma}^2 - \frac{1}{2}\mathrm{Var}[\hat{f}(\mathbf{x}(\mathbf{s})]\right)$$

with $\hat{f}(\mathbf{x}(\mathbf{s}))$ being the prediction of the log-transformed response, $\hat{\sigma}^2$ the estimated residual variance of the final `geoGAM` fit (see `predict.gam` with `se.fit=TRUE`) and Var$[\hat{f}(\mathbf{x}(\mathbf{s}))]$ the variance of $\hat{f}(\mathbf{x}(\mathbf{s}))$ as provided again by the final `geoGAM`.

For responses with square root transformation (`back.transform = 'sqrt'`) unbiased backtransform is computed by (*Nussbaum et al. 2017b*)

$$\tilde{Y}(s) = \hat{f}(\mathbf{x}(\mathbf{s}))^2 + \hat{\sigma}^2 - Var[\hat{f}(\mathbf{x}(\mathbf{s}))]$$

with $\hat{f}(\mathbf{x}(\mathbf{s}))^2$ being the prediction of the sqrt-transformed response, $\hat{\sigma}^2$ the estimated residual variance of the fitted model and $Var[\hat{f}(\mathbf{x}(\mathbf{s}))]$ the variance of $\hat{f}(\mathbf{x}(\mathbf{s}))$ as provided again by `geoGAM`.

Discretization of probability predictions

For binary and ordered responses predictions yield predicted occurrence probabilities $\tilde{P}(Y(\mathbf{s}) = \mathbf{r}|\mathbf{x}, s)$ for response classes $\mathbf{r}$.

To obtain binary class predictions a `threshold` can be given. A threshold of 0.5 (default) maximizes percentage correct of predicted classes. For binary responses of rare events this threshold may not be optimal. Maximizing on e.g. Gilbert Skill Score (GSS, Wilks, 2011, chap. 8) on cross-validation predictions of the final geoGAM might be a better strategy. GSS is excluding the correct predictions of the more abundant class and is preferably used in case of unequal distribution of binary responses (direct implementation of such a cross validation procedure planed.)

For ordered responses `predict` with `type = 'class'` selects the class to which the median of the probability distribution over the ordered categories is assigned (*Tutz 2012, p. 475*).

Value

Vector of point predictions for the sites in newdata is returned, with unbiased back transformation applied according to option back.transform.

If se.fit = TRUE then a 2 item list is returned with items fit and se.fit containing predictions and associated standard error estimates as computed by predict.gam.

Author(s)

M. Nussbaum

References

Cressie, N. A. C., 1993. Statistics for Spatial Data, John Wiley and Sons.

Nussbaum, M., Walthert, L., Fraefel, M., Greiner, L., and Papritz, A.: Mapping of soil properties at high resolution in Switzerland using boosted geoadditive models, SOIL, 3, 191-210, doi:10.5194/soil-3-191-2017, 2017.

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E., and Papritz, A.: Evaluation of digital soil mapping approaches with large sets of environmental covariates, SOIL, 4, 1-22, doi:10.5194/soil-4-1-2018, 2018.

Tutz, G., 2012. Regression for Categorical Data, Cambridge University Press.

Wilks, D. S., 2011. Statistical Methods in the Atmospheric Sciences, Academic Press.

See Also

geoGAM, gam, predict.gam, summary.geoGAM, plot.geoGAM

Examples

```
data(quakes)
set.seed(2)

quakes <- quakes[ ss <- sample(1:nrow(quakes), 50), ]

# Artificially split data to create prediction data set
quakes.pred <- quakes[ -ss, ]

quakes.geogam <- geoGAM(response = "mag",
                        covariates = c("depth", "stations"),
                        data = quakes,
                        max.stop = 5,
                        cores = 1)

predicted <- predict(quakes.geogam, newdata = quakes.pred, type = "response" )




## Use soil data set of soil mapping study area near Berne
```

```
data(berne)
data(berne.grid)

# Split data sets and
# remove rows with missing values in response and covariates

d.cal <- berne[ berne$dataset == "calibration" & complete.cases(berne), ]

### Model selection for numeric response
ph10.geogam <- geoGAM(response = "ph.0.10",
                      covariates = names(d.cal)[14:ncol(d.cal)],
                      coords = c("x", "y"),
                      data = d.cal,
                      seed = 1,
                      cores = 1)

# Create GRID output with predictions
sp.grid <- berne.grid[, c("x", "y")]

sp.grid$pred.ph.0.10 <- predict(ph10.geogam, newdata = berne.grid)

if(requireNamespace("raster")){

  require("sp")

  # transform to sp object
  coordinates(sp.grid) <- ~ x + y

  # assign Swiss CH1903 / LV03 projection
  proj4string(sp.grid) <- CRS("EPSG:21781")

  # transform to grid
  gridded(sp.grid) <- TRUE

  plot(sp.grid)

  # optionally save result to GeoTiff
  # writeRaster(raster(sp.grid, layer = "pred.ph.0.10"),
  #             filename= "raspH10.tif", datatype = "FLT4S", format ="GTiff")

}
```

Index