Package 'geoGAM'

October 29, 2016

Type Package

Title Select Sparse Geoadditive Models for Spatial Prediction

Version 0.1-1

Date 2016-10-27

Depends R(>= 2.14.0)

Imports mboost, mgcv, grpreg, MASS

Description A model building procedure to select a sparse 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 and smoothing splines. The resulting covariate set after gradient boosting is further reduced through cross validated backward selection 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 is provided.

License GPL (>= 2)

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

Maintainer Madlene Nussbaum <madlene.nussbaum@env.ethz.ch>

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-29 10:48:22

R topics documented:

berne *Berne – soil mapping case study*

Description

The Berne data set 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 pedogenetic 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 data set 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 for traditional polygon soil maps 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 data set 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 data sets can be found in *Nussbaum et al. in prep.*

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 et al. (in prep). Comparison of statistical approaches for digital soil mapping.

Examples

```
data(berne)
```

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

Description

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

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 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 et al. (in prep). Comparison of statistical approaches for digital soil mapping.

Examples

```
## Not run:
data(berne.grid)

# plot spatial object
library(raster)
```

```
coordinates(berne.grid) <- ~x+y
proj4string(berne.grid) <- CRS("+init=epsg:21781")
gridded(berne.grid) <- TRUE

plot( raster(berne.grid, layer = "tr_se_mrrtf2m"))

## End(Not run)
```

`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

To obtain predictive distribution for continuous responses `bootstrap` implements a model based bootstrap approach (*Davison-Hinkley-2008, pp. 262, 285*) for `geoGAM` models. Errors are simulated from Gaussian distribution with `R` repetitions involving the following steps:

1. simulating new response $Y(s)^*$ from Gaussian distribution $\mathcal{N}(\hat{f}(\mathbf{x}, s), \hat{\sigma}^2)$ with fitted values of the final model $\hat{f}(\mathbf{x}, s)$ and residual variance $\hat{\sigma}^2$,

2. selecting geoadditive model with geoGAM for $Y(s)^*$,
3. computing prediction error

$$\sigma_i^* = \hat{f}(\mathbf{x}, s)^* - (\hat{f}(\mathbf{x}, s) + \mathcal{N}(0, \hat{\sigma}^{*2}))$$

with fitted values $\hat{f}(\mathbf{x}, s)^*$ and residual variance $\hat{\sigma}^{*2}$ of the model built on the simulated response $Y(s)^*$.

Simulated point predictions are then computed by $\hat{f}(\mathbf{x}, s)^* - \sigma_i^*$.

This results in a predictive distribution for each site. For `back.transform = log` or `sqrt` the simulated predictions are backtransformed after the steps above (see `predict.geoGAM` for more information). Prediction intervals can be achieved by computing the desired percentiles (e.g. 2.5 % and 97.5 % percentiles to get lower and upper limits of 95 % prediction intervals.)

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)

Madlene Nussbaum, <madlene.nussbaum@env.ethz.ch>

References

Nussbaum, M., Papritz, A., and others (in prep). Mapping of soil properties at high resolution by geo-additive models.

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

```
## Not run:
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)


## compute model based bootstrap with 100 repetitions
quakes.boot <- bootstrap(quakes.geogam,
                         newdata = quakes.pred,
                         R = 100)


# 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")

## End(Not run)
```

---

geoGAM *Select sparse geoadditive model*

Description

Selects a sparse geoadditive model from a large set of covariates with the aim of spatial prediction. As covariates categoric and continuous are handeled and smooth spatial surfaces can be included for model selection.

Usage

```
geoGAM(response, covariates = names(data)[!(names(data) %in% c(response,coords))],
       data, coords = NULL, weights = rep(1, nrow(data)),
       offset = T, max.stop = 300, non.stationary = F,
       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

Generic model representation

Response Y at location s measured on interval scale is modeled by the regression function f from the environmental covariates $\mathbf{x}$ as

$$Y(\mathbf{x}, s) = f(\mathbf{x}, s) + \epsilon$$

with ϵ being a zero mean spatially uncorrelated error that follows a given distribution (e.g. Gaussian). For positively skewed $Y(\mathbf{x}, s)$ transformed by natural logarithm ϵ follows a lognormal distribution.

The probability of presence of a event or an observed class r is modeled similarly using the inverse logistic transformation (e.g. *Tutz 2012*, p. 37) by

$$P(Y(\mathbf{s}) = r|\mathbf{x}, s) = logit^{-1}\Big(f(\mathbf{x}, s) \Big) = \frac{exp(\ f(\mathbf{x}, s)\ )}{1 + exp(\ f(\mathbf{x}, s)\ )}.$$

Analogous ordered classes at location s are modeled by cumulative probabilities (*Tutz-2012*, pp. 244, cumulative logit model, proportional odds model) of the class being equal or smaller than r with

$$P(Y(\mathbf{s}) \leq r|\mathbf{x}, s) = \frac{exp(\ f(\nu_r, \mathbf{x}, s)\ )}{1 + exp(\ f(\nu_r, \mathbf{x}, s)\ )},$$

where $\nu_1 ... \nu_{r-1}$ are class specific thresholds or model intercepts. This specification follows the proportional odds assumption by estimating equal covariate effects independent from the target category. \

.

Model selection

Sparse models are desirable if models should be open to feasibility checks in regard of meaningful relationships between response and (environmental) covariates. A fully automated model selection procedure is required for large number of responses or simulation of predictive distributions (`bootstrap.geoGAM`). The following steps are performed in `geoGAM`:

1. To ensure stability of the algorithm continuous covariates are centered and scaled by their range.
2. (optional with `offset = TRUE`) The group Lasso (least absolute shrinkage and selection operator, *Breheny and Huang 2015*, `grpreg`) – an algorithm that likely excludes non-relevant covariates and treats categorical covariates as groups – is used to select relevant categorical covariates.

   For ordered responses step wise proportional odds logistic regression (`polr`) in both directions with BIC is used, because no Lasso is available.
3. By component wise gradient boosting a subset of relevant continuous and categoric covariates and spatial effects is selected. Boosting is a slow stage-wise additive learning algorithm. It expands $f(\mathbf{x}, s)$ in a set of base procedures (baselearners) and approximates $f(\mathbf{x}, s)$ by a finite sum of them. Best fitting baselearners update the model only in small steps.

   The default step length ($\nu = 0.1$) is used as this is not a sensitive parameter as long as it is small (*Hofner et al. 2012*). Early stopping of the algorithm (`mstop`) is achieved by minimizing cross validation error. For continuous covariates non-parametric penalized smoothing spline baselearners (*Kneib et al. 2009*) are used. Categorical covariates are assigned to linear baselearners. To capture the spatial structure in the data smooth spatial surface established by a bivariate tensor-product P-spline of spatial coordinates are added. Spatially varying effects are estimated by baselearners formed by the product continuous covariates and the smooth spatial surface (`non.stationary = TRUE`).

   To equalize the inclusion probability of baselearners each should have the same flexibility to fit the data (*Hofner et al. 2011*). Unbiased baselearner selection is controlled by assigning 5 degrees of freedom to each. Where linear categorical covariates did not reach 6 levels (5 degrees of freedom) we aggregated them to grouped baselearners.

   With option `offset = TRUE` fitted values of group Lasso and proportional odds regression respectively (step 1) are used as an offset to the boosting algorithm. Setting an offset accelerates model building and ensures inclusion of relevant categorical covariates due to the limit of equal degrees of freedom of 5.
4. To remove baselearners with strongly shrunken coefficients the effect size is evaluated. As effect size the range of coefficients for categorical covariates is computed and the magnitude of smooth baselearners is obtained after removal of extreme values from the partial effects (outlier < 1. quantile - 1.5*interquantile distance and > 3. quantile + 1.5* interquantile distance, *Frigge et al. 1989*, `boxplot`).

   The optimal effect size is found by fitting geoadditive models by retaining the same degrees of freedom as in gradient boosting for the non-parametric smooth terms. The categorical covariates used to compute the offset (step 1) are included as linear parametric effects.
5. The model is further reduced by stepwise removal of covariates. The covariate to drop is chosen by largest `p` value of `F` tests for categorical terms and approximate `F` test (*Wood 2011*) for smooth terms.

6. Similar levels of categorical covariates are merged stepwise based on highest p values from two sample t tests (t.test) of partial residuals. For interactions by smooth terms and categorical covariates the t-test of the interaction coefficient obtained from linear regression computed on the partial residuals are used.

The optimal model building parameters (λ in group Lasso, number of boosting iterations (ν), magnitude of baselearners, number of remaining covariates and degree of aggregation of factor levels) is done by minimizing root mean squared error (RMSE) for interval scaled, Brier Score for binary and ranked probability score (*Wilks 2011, chap. 8*) for ordered responses computed from 10fold cross-validation done with the same subsets.

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)

Madlene Nussbaum, <madlene.nussbaum@env.ethz.ch>

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.

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

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., 2012. Model-based boosting in R, a hands-on tutorial using the R package mboost. Technical report no. 120, Ludwig-Maximilians-Universitaet Muenchen, Department of Statistics. http://epub.ub.uni-muenchen.de/12754/

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

Nussbaum, M., Papritz, A., and others, in prep. Mapping of soil properties at high resolution by geo-additive models.

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")


## Not run:

## 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 = T,
                      sets = mboost::cv(rep(1, nrow(d.cal)), type = "kfold"),
                      validation.data = d.val)
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 = F,
                          sets = sample( cut(seq(1,nrow(d.cal)),breaks=10,labels=FALSE) ),
                             validation.data = d.val)
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 = T,
```

```
                        non.stationary = T,
                        seed = 1,
                        validation.data = d.val)
summary(dclass.geogam)
summary(dclass.geogam, what = "path")


## End(Not run)
```

---

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)

Madlene Nussbaum, <madlene.nussbaum@env.ethz.ch>

References

Nussbaum, M., Papritz, A., and others (in prep). Mapping of soil properties at high resolution by geo-additive models.

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 = F, ...)
```

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.
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. For lognormal responses (back.transform = 'log') unbiased back transformation (*Cressie 1993*,p 135) was computed by

$$\tilde{Y}(s) = exp(\hat{f}(\mathbf{x}, s) - \frac{1}{2}\hat{\sigma}^2)$$

with $\hat{\sigma}^2$ being the estimated variance of $\hat{f}(\mathbf{x}, s)$ (see predict.gam with se.fit=TRUE). For responses with square root transformation (back.transform = 'sqrt') unbiased backtransform was computed by

$$\tilde{Y}(s) = \hat{f}(\mathbf{x}, s)^2 - \hat{\sigma}^2$$

.
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)

Madlene Nussbaum, <madlene.nussbaum@env.ethz.ch>

References

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

Nussbaum, M., Papritz, A., and others (in prep). Mapping of soil properties at high resolution by geo-additive models.

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" )


## Not run:


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

library(raster)

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 binary 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)

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

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

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

# assign Swiss CH1903 / LV03 projection
proj4string(sp.grid) <- CRS("+init=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")


## End(Not run)
```

Index