Package 'Bodi'

March 23, 2022

Title Boosting Diversity in Regression Ensembles

Version 0.1.0

Description A gradient boosting-based algorithm by incorporating a diversity term to guide the gradient boosting iterations, see Bourel, Cugliari, Goude, Poggi (2021) <https://hal.archives-ouvertes.fr/hal-03041309/>.

License MIT + file LICENSE

Encoding UTF-8

Imports mgcv, ranger, rpart, gbm, opera

RoxygenNote 7.1.2

NeedsCompilation no

Maintainer Yannig Goude <yannig.goude@edf.fr>

Author Yannig Goude [aut, cre] (<https://orcid.org/0000-0003-2028-5536>),
Mathias Bourel [aut] (<https://orcid.org/0000-0002-7472-7179>),
Jairo Cugliari [aut] (<https://orcid.org/0000-0001-6779-0084>),
Jean-Michel Poggi [aut] (<https://orcid.org/0000-0002-8222-1653>)

Repository CRAN

Date/Publication 2022-03-23 12:40:02 UTC

R topics documented:

Bodi-package	*Bodi: Boosting Diversity Algorithm*

Description

We provide an implementation of the boosting diversity algorithm. This is a gradient boosting-based algorithm by incorporating a diversity term to guide the gradient boosting iterations. The idea is to trade off some individual optimality for global enhancement. The improvement is obtained with progressively generated predictors by boosting diversity. See Borel et al. (2021) <https://hal.archives-ouvertes.fr/hal-03041309v1>

Author(s)

Yannig Goude [aut, cre], Mathias Bourel [aut], Jairo Cugliari [aut], Jean-Michel Poggi [aut]

Mantainer: Yannig Goude <yannig.goude@edf.fr>

References

- Mathias Bourel, Jairo Cugliari, Yannig Goude, Jean-Michel Poggi. Boosting Diversity in Regression Ensembles. https://hal.archives-ouvertes.fr/hal-03041309v1 (2021).

---

boosting_diversity	*Diversity Boosting Algorithm*

Description

Train a set of initial learners by promoting diversity among them. For this, a gradient descent strategy is adopted where a specialized loss function induces diversity which yields on a reduction of the mean-square-error of the aggregated learner.

Usage

```
boosting_diversity(
  target,
  cov,
  data0,
  data1,
  sample_size = 0.5,
  grad_step = 1,
  diversity_weight = 1,
  Nstep = 10,
  model = "gam",
  sampling = "random",
  Nblock = 10,
```

```
        aggregation_type = "uniform",
        param = list(),
        theorical_dw = FALSE,
        model_list = NULL,
        w_list = NULL,
        param_list = NULL,
        cov_list = NULL
)
```

Arguments

`target`	name of the target variable
`cov`	the model equation, a character string provided in the formula syntax. For example, for a linear model including covariates X_1 and X_2 it will be "X1+X2" and for a GAM with smooth effects it will be "s(X1)+s(X2)"
`data0`	the learning set
`data1`	the test set
`sample_size`	the size of the bootstrap sample as a proportion of the learning set size. sample_size=0.5 means that the resamples are of size n/2 where n is the number of rows of data0.
`grad_step`	step of the gradient descent
`diversity_weight`	
	the weight of the diversity encouraging penalty (kappa in the paper)
`Nstep`	the number of iterations of the diversity boosting algorithm (N in the paper)
`model`	the type of base learner used in the algorithm if using a single base learner (model_list=NULL). Currently it could be either "gam" for an additive model, "rf" for a random forest, ""gbm" for gradient boosting machines, "rpart" for single CART trees.
`sampling`	the type of sampling procedure used in the resampling step. Could be either "random" for uniform random sampling with replacement or "blocks" for uniform sampling with replacement of blocks of consecutive data points. Default is "random".
`Nblock`	number of blocks for the block sampling. Equal to 10 by default.
`aggregation_type`	
	type of aggregation used for the ensemble method, default is uniform weights but it could be also "MLpol" an aggregation algorithm from the opera package
`param`	a list containing the parameters of the model chosen. It could be e.g. the number of trees for "rf", the depth of the tree for "rpart"...
`theorical_dw`	set to TRUE if one want to use the theoretical upper bound of the diversity weight kappa
`model_list`	a list of model among the possible ones (see the description of model argument). In that case the week learner is sample at each step in the list. "Still "experimental", be careful.
`w_list`	the prior weights of each model in the model_list
`param_list`	list of parameters of each model in the model_list
`cov_list`	list of covariates of each model in the model_list

Value

a list including the boosted models, the ensemble forecast

fitted_ensemble

                  Fitted values (in-sample predictions) for the ensemble method (matrix).

forecast_ensemble

                  Forecast (out-sample predictions) for the ensemble method (matrix).

fitted   Fitted values of the last boosting iteration (vector).

forecast   Forecast of the last boosting iteration (vector).

err_oob   Estimated out-of-bag errors by iteration (vector).

diversiy_oob   Estimated out-of-bag diversity (vector).

Author(s)

Yannig Goude <yannig.goude@edf.fr>

Examples

```
all <- na.omit(airquality)
smp <- sample(nrow(all), floor(.8 * nrow(all)))
boosting_diversity("Ozone", "Solar.R+Wind+Temp+Month+Day",
                   data0 = all[smp, ], data1 = all[-smp, ])
```

---

  buildBlock   *buildBlock*

Description

Compute blocks of consecutive data for blockwise CV or sampling.

Usage

```
buildBlock(Nblock, data0)
```

Arguments

Nblock   number of blocks

data0   the learning set

Value

A list of vectors containing the indices of each block.

Author(s)

Yannig Goude <yannig.goude@edf.fr>

Examples

```
buildBlock(4, data.frame(id = 1:15))
```

Index