Package 'GMMBoost'

October 12, 2022

Type Package

Title Likelihood-Based Boosting for Generalized Mixed Models

Version 1.1.3

Date 2020-02-03

Author Andreas Groll

Maintainer Andreas Groll <groll@mathematik.uni-muenchen.de>

Description Likelihood-based boosting approaches for generalized mixed models are provided.

Imports minqa, magic

License GPL-2

Repository CRAN

Date/Publication 2020-02-03 11:00:02 UTC

NeedsCompilation no

R topics documented:

bGAMM *Fit Generalized Semiparametric Mixed-Effects Models*

Description

Fit a semiparametric mixed model or a generalized semiparametric mixed model.

Usage

```
bGAMM(fix=formula, add=formula, rnd=formula,
      data, lambda, family = NULL, control = list())
```

Arguments

fix	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. For categorical covariables use as.factor(.) in the formula. Note, that the corresponding dummies are treated as a group and are updated blockwise
add	a one-sided linear formula object describing the additive part of the model, with the additive terms on the right side of a ~ operator, separated by + operators. The smooth terms are expanded in B-spline basis functions, with a difference penalty apllied on adjacent spline coefficients.
rnd	a two-sided linear formula object describing the random-effects part of the model, with the grouping factor on the left of a ~ operator and the random terms, separated by + operators, on the right.
data	the data frame containing the variables named in formula.
lambda	the smoothing parameter that controls the smoothness of the additive terms. The optimal smoothing parameter is a tuning parameter of the procedure that has to be determined, e.g. by use of information criteria or cross validation.
family	a GLM family, see glm and family. If family is missing then a linear mixed model is fit; otherwise a generalized linear mixed model is fit.
control	a list of control values for the estimation algorithm to replace the default values returned by the function bGAMMControl. Defaults to an empty list.

Value

Generic functions such as print, predict, summary and plot have methods to show the results of the fit. The predict function uses also estimates of random effects for prediction, if possible (i.e. for known subjects of the grouping factor). The plot function shows the estimated smooth functions. Single functions can be specified by a suitable vector in the which argument. Default is which=Null and all smooth functions (up to a maximum of nine) are shown.

call	a list containing an image of the bGLMM call that produced the object.
coefficients	a vector containing the estimated fixed effects

ranef	a vector containing the estimated random effects.
spline.weights	a vector containing the estimated spline coefficients.
StdDev	a scalar or matrix containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively.
fitted.values	a vector of fitted values.
phi	estimated scale parameter, if overdispersion=TRUE is used. Otherwise, it is equal to one.
HatMatrix	hat matrix corresponding to the final fit.
IC	a matrix containing the evaluated information criterion for the different covariates (columns) and for each boosting iteration (rows).
IC_sel	a vector containing the evaluated information criterion for the selected covariate at different boosting iterations.
components	a vector containing the selected components at different boosting iterations.
opt	number of optimal boosting steps with respect to AIC or BIC, respectively, if OPT=TRUE. Otherwise, opt is equal to the number of iterations. Note, that the boosting algorithm is also stopped, if it has converged with respect to the parameter estimates [coefficients,ranef] or with respect to the IC_sel.
Deltamatrix	a matrix containing the estimates of fixed and random effects (columns) for each boosting iteration (rows).
Q_long	a list containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively, for each boosting iteration.
fixerror	a vector with standrad errors for the fixed effects.
ranerror	a vector with standrad errors for the random effects.
smootherror	a matrix with pointwise standard errors for the smooth function estimates.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

References

Groll, A. and G. Tutz (2012). Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting. *Methods of Information in Medicine* 51(2), 168–177.

See Also

bGAMMControl

Examples

```
data("soccer")

gamm1 <- bGAMM(points ~ ball.possession + tackles,
         ~ transfer.spendings + transfer.receits
         + unfair.score + ave.attend + sold.out,
```

```
          rnd = list(team=~1), data = soccer, lambda = 1e+5,
          family = poisson(link = log), control = list(steps=200, overdispersion=TRUE,
          start=c(1,rep(0,25))))

plot(gamm1)

# see also demo("bGAMM-soccer")
```

---

bGAMMControl *Control Values for* bGAMM *fit*

Description

The values supplied in the function call replace the defaults and a list with all possible arguments is returned. The returned list is used as the `control` argument to the bGAMM function.

Usage

```
bGAMMControl(nue=0.1,add.fix=NULL,start=NULL,q_start=NULL,
             OPT=TRUE,nbasis=20,spline.degree=3,
             diff.ord=2,sel.method="aic",steps=500,
             method="EM",overdispersion=FALSE)
```

Arguments

nue	weakness of the learner. Choose 0 < nue =< 1. Default is 0.1.
add.fix	a vector specifying smooth terms, which are excluded from selection.
start	a vector containing starting values for fixed and random effects of suitable length. Default is a vector full of zeros.
q_start	a scalar or matrix of suitable dimension, specifying starting values for the random-effects variance-covariance matrix. Default is a scalar 0.1 or diagonal matrix with 0.1 in the diagonal.
OPT	logical scalar. When TRUE the estimates at the optimal number of boosting steps, chosen by information criteria, are derived. If FALSE, the estimates at the maximal number of boosting steps are derived. Default is TRUE.
nbasis	the number of b-spline basis functions for the modeling of smooth terms. Default is 20.
spline.degree	the degree of the B-spline polynomials. Default is 3.
diff.ord	the order of the difference penalty; must be lower than the degree of the B-spline polynomials (see previous argument). Default is 2.
sel.method	two different information criteria, "aic" or "bic", can be chosen, on which the selection step is based on. Default is "aic".
steps	the number of boosting interations. Default is 500.

method	two methods for the computation of the random-effects variance-covariance parameter estimates can be chosen, an EM-type estimate and an REML-type estimate. The REML-type estimate uses the bobyqa function for optimization. Default is EM.
overdispersion	logical scalar. If FALSE, no scale parameter is derived, if TRUE, in each boosting iteration a scale parameter is estimated by use of Pearson residuals. This can be used to fit overdispersed Poisson models. Default is FALSE.

Value

a list with components for each of the possible arguments.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

See Also

bGAMM, bobyqa

Examples

```
# decrease the maximum number of boosting iterations
# and use BIC for selection
bGAMMControl(steps = 100, sel.method = "BIC")
```

---

bGLMM *Fit Generalized Mixed-Effects Models*

Description

Fit a linear mixed model or a generalized linear mixed model.

Usage

```
bGLMM(fix=formula, rnd=formula, data, family = NULL, control = list())
```

Arguments

fix	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. For categorical covariables use as.factor(.) in the formula. Note, that the corresponding dummies are treated as a group and are updated blockwise
rnd	a two-sided linear formula object describing the random-effects part of the model, with the grouping factor on the left of a ~ operator and the random terms, separated by + operators, on the right.

data	the data frame containing the variables named in formula.
family	a GLM family, see glm and family. If family is missing then a linear mixed model is fit; otherwise a generalized linear mixed model is fit.
control	a list of control values for the estimation algorithm to replace the default values returned by the function bGLMMControl. Defaults to an empty list.

Value

Generic functions such as print, predict and summary have methods to show the results of the fit. The predict function uses also estimates of random effects for prediction, if possible (i.e. for known subjects of the grouping factor).

call	a list containing an image of the bGLMM call that produced the object.
coefficients	a vector containing the estimated fixed effects
ranef	a vector containing the estimated random effects.
StdDev	a scalar or matrix containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively.
fitted.values	a vector of fitted values.
phi	estimated scale parameter, if overdispersion=TRUE is used. Otherwise, it is equal to one.
HatMatrix	hat matrix corresponding to the final fit.
IC	a matrix containing the evaluated information criterion for the different covariates (columns) and for each boosting iteration (rows).
IC_sel	a vector containing the evaluated information criterion for the selected covariate at different boosting iterations.
components	a vector containing the selected components at different boosting iterations.
opt	number of optimal boosting steps with respect to AIC or BIC, respectively, if OPT=TRUE. Otherwise, opt is equal to the number of iterations. Note, that the boosting algorithm is also stopped, if it has converged with respect to the parameter estimates [coefficients,ranef] or with respect to the IC_sel.
Deltamatrix	a matrix containing the estimates of fixed and random effects (columns) for each boosting iteration (rows).
Q_long	a list containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively, for each boosting iteration.
fixerror	a vector with standrad errors for the fixed effects.
ranerror	a vector with standrad errors for the random effects.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

References

Tutz, G. and A. Groll (2010). Generalized linear mixed models based on boosting. In T. Kneib and G. Tutz (Eds.), *Statistical Modelling and Regression Structures - Festschrift in the Honour of Ludwig Fahrmeir.* Physica.

See Also

bGLMMControl

Examples

```
data("soccer")
## linear mixed models
lm1 <- bGLMM(points ~ transfer.spendings + I(transfer.spendings^2)
       + ave.unfair.score + transfer.receits + ball.possession
       + tackles + ave.attend + sold.out, rnd = list(team=~1), data = soccer)

lm2 <- bGLMM(points~transfer.spendings + I(transfer.spendings^2)
       + ave.unfair.score + transfer.receits + ball.possession
       + tackles + ave.attend + sold.out, rnd = list(team=~1 + ave.attend),
       data = soccer, control = list(steps=10, lin=c("(Intercept)","ave.attend"),
       method="REML", nue=1, sel.method="bic"))

## linear mixed models with categorical covariates
lm3 <- bGLMM(points ~ transfer.spendings + I(transfer.spendings^2)
       + as.factor(red.card) + as.factor(yellow.red.card)
       + transfer.receits + ball.possession + tackles + ave.attend
       + sold.out, rnd = list(team=~1), data = soccer, control = list(steps=10))


## generalized linear mixed model
glm1 <- bGLMM(points~transfer.spendings  + I(transfer.spendings^2)
        + ave.unfair.score + transfer.receits + ball.possession
        + tackles + ave.attend + sold.out, rnd = list(team=~1),
        family = poisson(link = log), data = soccer,
        control = list(start=c(5,rep(0,31))))
```

---

bGLMMControl *Control Values for* bGLMM *fit*

Description

The values supplied in the function call replace the defaults and a list with all possible arguments is returned. The returned list is used as the control argument to the bGLMM function.

Usage

```
bGLMMControl(nue=0.1, lin="(Intercept)", start=NULL, q_start=NULL, OPT=TRUE,
             sel.method="aic", steps=500, method="EM",
             overdispersion=FALSE,print.iter=TRUE)
```

Arguments

nue	weakness of the learner. Choose 0 < nue =< 1. Default is 0.1.
lin	a vector specifying fixed effects, which are excluded from selection.
start	a vector containing starting values for fixed and random effects of suitable length. Default is a vector full of zeros.
q_start	a scalar or matrix of suitable dimension, specifying starting values for the random-effects variance-covariance matrix. Default is a scalar 0.1 or diagonal matrix with 0.1 in the diagonal.
OPT	logical scalar. When TRUE the estimates at the optimal number of boosting steps, chosen by information criteria, are derived. If FALSE, the estimates at the maximal number of boosting steps are derived. Default is TRUE.
sel.method	two different information criteria, "aic" or "bic", can be chosen, on which the selection step is based on. Default is "aic".
steps	the number of boosting interations. Default is 500.
method	two methods for the computation of the random-effects variance-covariance parameter estimates can be chosen, an EM-type estimate and an REML-type estimate. The REML-type estimate uses the bobyqa function for optimization. Default is EM.
overdispersion	logical scalar. If FALSE, no scale parameter is derived, if TRUE, in each boosting iteration a scale parameter is estimated by use of Pearson residuals. This can be used to fit overdispersed Poisson models. Default is FALSE.
print.iter	logical. Should the number of interations be printed?. Default is TRUE.

Value

a list with components for each of the possible arguments.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

See Also

bGLMM, bobyqa

Examples

```
# decrease the maximum number of boosting iterations
# and use BIC for selection
bGLMMControl(steps = 100, sel.method = "BIC")
```

GMMBoost *Likelihood-Based Boosting for Generalized Mixed Models*

Description

This packages provides likelihood-based boosting approaches for Generalized mixed models

Details

Package:	GMMBoost
Type:	Package
Version:	1.1.3
Date:	2019-02-03
License:	GPL-2
LazyLoad:	yes

for loading a dataset type data(nameofdataset)

Author(s)

Andreas Groll

References

Special thanks goes to Manuel Eugster, Sebastian Kaiser, Fabian Scheipl and Felix Heinzl, who helped to create this package and whose insightful advices helped to improve the package.

See Also

bGLMM,OrdinalBoost,bGAMM

knee *Clinical pain study on knee data*

Description

The knee data set illustrates the effect of a medical spray on the pressure pain in the knee due to sports injuries.

Usage

```
data(soccer)
```

Format

A data frame with 381 patients, each with three replicates, and the following 7 variables:

`pain` the magnitude of pressure pain in the knee given in 5 categories (1: lowest pain; 5: strongest pain).

`time` the number of replication

`id` number of patient

`th` the therapy (1: spray; 0: placebo)

`age` age of the patient in years

`sex` sex of the patient (1: male; 0: female)

`pain.start` the magnitude of pressure pain in the knee at the beginning of the study

References

Tutz, G. (2000). *Die Analyse kategorialer Daten - eine anwendungsorientierte Einfuehrung in Logit-Modellierung und kategoriale Regression*. Muenchen: Oldenbourg Verlag.

Tutz, G. and A. Groll (2011). Binary and ordinal random effects models including variable selection. Technical Report **97**, Ludwig-Maximilians-University.

See Also

`OrdinalBoost`, `glmmLasso`.

`OrdinalBoost` *Fit Generalized Mixed-Effects Models*

Description

Fit a generalized linear mixed model with ordinal response.

Usage

```
OrdinalBoost(fix=formula, rnd=formula, data,model="sequential",control=list())
```

Arguments

`fix`	a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. For categorical covariables use `as.factor(.)` in the formula. Note, that the corresponding dummies are treated as a group and are updated blockwise
`rnd`	a two-sided linear formula object describing the random-effects part of the model, with the grouping factor on the left of a ~ operator and the random terms, separated by + operators, on the right.
`data`	the data frame containing the variables named in `formula`.

model	Two models for repeatedly assessed ordinal scores, based on the threshold concept, are available, the "sequential" and the "cumulative" model. Default is "sequential".
control	a list of control values for the estimation algorithm to replace the default values returned by the function OrdinalBoostControl. Defaults to an empty list.

Value

Generic functions such as print, predict and summary have methods to show the results of the fit. The predict function shows the estimated probabilities for the different categories for each observation, either for the data set of the OrdinalBoost object or for newdata. Default is newdata=Null. It uses also estimates of random effects for prediction, if possible (i.e. for known subjects of the grouping factor).

call	a list containing an image of the OrdinalBoost call that produced the object.
coefficients	a vector containing the estimated fixed effects
ranef	a vector containing the estimated random effects.
StdDev	a scalar or matrix containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively.
fitted.values	a vector of fitted values.
HatMatrix	hat matrix corresponding to the final fit.
IC	a matrix containing the evaluated information criterion for the different covariates (columns) and for each boosting iteration (rows).
IC_sel	a vector containing the evaluated information criterion for the selected covariate at different boosting iterations.
components	a vector containing the selected components at different boosting iterations.
opt	number of optimal boosting steps with respect to AIC or BIC, respectively, if OPT=TRUE. Otherwise, opt is equal to the number of iterations. Note, that the boosting algorithm is also stopped, if it has converged with respect to the parameter estimates [coefficients,ranef] or with respect to the IC_sel.
Deltamatrix	a matrix containing the estimates of fixed and random effects (columns) for each boosting iteration (rows).
Q_long	a list containing the estimates of the random effects standard deviation or variance-covariance parameters, respectively, for each boosting iteration.
fixerror	a vector with standrad errors for the fixed effects.
ranerror	a vector with standrad errors for the random effects.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

References

Tutz, G. and A. Groll (2012). Likelihood-based boosting in binary and ordinal random effects models. *Journal of Computational and Graphical Statistics*. To appear.

See Also

OrdinalBoostControl

Examples

```
## Not run:
data(knee)

# fit a sequential model
# (here only one step is performed in order to
# save computational time)

glm1 <- OrdinalBoost(pain ~ time + th + age + sex, rnd = list(id=~1),
        data = knee, model = "sequential", control = list(steps=1))

# see also demo("OrdinalBoost-knee") for more extensive examples

## End(Not run)
```

OrdinalBoostControl *Control Values for* OrdinalBoost *fit*

Description

The values supplied in the function call replace the defaults and a list with all possible arguments is returned. The returned list is used as the control argument to the bGLMM function.

Usage

```
OrdinalBoostControl(nue=0.1, lin=NULL, katvar=NULL, start=NULL, q_start=NULL,
                OPT=TRUE, sel.method="aic", steps=100, method="EM", maxIter=500,
                    print.iter.final=FALSE, eps.final=1e-5)
```

Arguments

nue	weakness of the learner. Choose 0 < nue =< 1. Default is 0.1.
lin	a vector specifying fixed effects, which are excluded from selection.
katvar	a vector specifying category-specific covariates, which are also excluded from selection.
start	a vector containing starting values for fixed and random effects of suitable length. Default is a vector full of zeros.
q_start	a scalar or matrix of suitable dimension, specifying starting values for the random-effects variance-covariance matrix. Default is a scalar 0.1 or diagonal matrix with 0.1 in the diagonal.

OPT	logical scalar. When TRUE the estimates at the optimal number of boosting steps, chosen by information criteria, are derived. If FALSE, the estimates at the maximal number of boosting steps are derived. Default is TRUE.
sel.method	two different information criteria, "aic" or "bic", can be chosen, on which the selection step is based on. Default is "aic".
steps	the number of boosting interations. Default is 100.
method	two methods for the computation of the random-effects variance-covariance parameter estimates can be chosen, an EM-type estimate and an REML-type estimate. The REML-type estimate uses the bobyqa function for optimization. Default is EM.
maxIter	the number of interations for the final Fisher scoring reestimation procedure. Default is 500.
print.iter.final	logical. Should the number of interations in the final re-estimation step be printed?. Default is FALSE.
eps.final	controls the speed of convergence in the final re-estimation. Default is 1e-5.

Value

a list with components for each of the possible arguments.

Author(s)

Andreas Groll <andreas.groll@stat.uni-muenchen.de>

See Also

OrdinalBoost, bobyqa

Examples

```
# decrease the maximum number of boosting iterations
# and use BIC for selection
OrdinalBoostControl(steps = 10, sel.method = "BIC")
```

---

soccer *German Bundesliga data for the seasons 2008-2010*

Description

The soccer data contains different covariables for the teams that played in the first Germna soccer division, the Bundesliga, in the seasons 2007/2008 until 2009/2010.

Usage

```
data(soccer)
```

Format

A data frame with 54 observations on the following 16 variables.

`pos` the final league rank of a soccer team at the end of the season

`team` soccer teams

`points` number of the points a team has earned during the season

`transfer.spendings` the amount (in Euro) that a team has spent for new players at the start of the season

`transfer.receits` the amount (in Euro) that a team has earned for the selling of players at the start of the season

`yellow.card` number of the yellow cards a team has received during the season

`yellow.red.card` number of the yellow-red cards a team has received during the season

`red.card` number of the red cards a team has received during the season

`unfair.score` unfairness score which is derived by the number of yellow cards (1 unfairness point), yellow-red cards (2 unfairness points) and red cards (3 unfairness points) a team has received during the season

`ave.unfair.score` average unfairness score per match

`ball.possession` average percentage of ball possession per match

`tackles` average percentage of head-to-head duels won per match

`capacity` capacity of the team's soccer stadium

`total.attend` total attendance of a soccer team for the whole season

`ave.attend` average attendance of a soccer team per match

`sold.out` number of stadium sold outs during a season

References

Groll, A. and G. Tutz (2011a). Regularization for generalized additive mixed models by likelihood-based boosting. Technical Report **110**, Ludwig-Maximilians-University.

Groll, A. and G. Tutz (2012). Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting. *Methods of Information in Medicine.* To appear.

Groll, A. and G. Tutz (2011c). Variable selection for generalized linear mixed models by L1-penalized estimation. Technical Report **108**, Ludwig-Maximilians-University.

We are grateful to Jasmin Abedieh for providing the German Bundesliga data, which were part of her bachelor thesis.

See Also

bGLMM,bGAMM.

Index