Package: BTLLasso (via r-universe)

October 30, 2026

Type Package

Title Modelling Heterogeneity in Paired Comparison Data

Version 0.1-14

Date 2025-07-22

Maintainer Gunther Schauberger <gunther.schauberger@tum.de>

Description Performs 'BTLLasso' as described by Schauberger and Tutz (2019) <doi:10.18637/jss.v088.i09> and Schauberger and Tutz (2017) <doi:10.1177/1471082X17693086>. BTLLasso is a method to include different types of variables in paired comparison models and, therefore, to allow for heterogeneity between subjects. Variables can be subject-specific, object-specific and subject-object-specific and can have an influence on the attractiveness/strength of the objects. Suitable L1 penalty terms are used to cluster certain effects and to reduce the complexity of the models.

License GPL (>= 2)

Imports Rcpp (>= 0.11.3), stringr, psychotools

Depends Matrix, parallel

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Author Gunther Schauberger [aut, cre]

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

Date/Publication 2025-07-22 09:43:01 UTC

RemoteUrl https://github.com/cran/BTLLasso

RemoteRef HEAD

RemoteSha 3ec5286f24b6a8e404da653144ca60d75501ebaa

Contents

BTLLasso-package . . . 2
boot.BTLLasso . . . 5
BTLLasso . . . 9
Buli1415 . . . 13
Buli1516 . . . 15
Buli1617 . . . 17
Buli1718 . . . 19
BuliResponse . . . 21
ctrl.BTLLasso . . . 22
cv.BTLLasso . . . 26
GLES . . . 31
GLESsmall . . . 33
paths . . . 35
plot.boot.BTLLasso . . . 38
plot.BTLLasso . . . 41
predict.BTLLasso . . . 45
print.boot.BTLLasso . . . 48
print.BTLLasso . . . 51
print.cv.BTLLasso . . . 54
response.BTLLasso . . . 57
SimData . . . 59

Index **61**

`BTLLasso-package` *BTLLasso*

Description

Performs BTLLasso, a method to model heterogeneity in paired comparison data. Different types of covariates are allowd to have an influence on the attractivity/strength of the objects. Covariates can be subject-specific, object-specific or subject-object-specific. L1 penalties are used to reduce the complexity of the model by enforcing clusters of equal effects or by elimination of irrelevant covariates. Several additional functions are provided, such as cross-validation, bootstrap intervals, and plot functions.

Author(s)

Maintainer: Gunther Schauberger <gunther.schauberger@tum.de>

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso, cv.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`boot.BTLLasso` *Bootstrap function for BTLLasso*

Description

Performs bootstrap for BTLLasso to get bootstrap intervals. Main input argument is a `cv.BTLLasso` object. The bootstrap is (recommended to be) performed on level of the cross-validation. Therefore, within every bootstrap iteration the complete cross-validation procedure from the `cv.BTLLasso` object is performed. A `plot` function can be applied to the resulting `boot.BTLLasso` object to plot bootstrap intervals.

Usage

```
boot.BTLLasso(
  model,
  B = 500,
  lambda = NULL,
  cores = 1,
  trace = TRUE,
  trace.cv = TRUE,
  with.cv = TRUE
)
```

Arguments

model	A cv.BTLLasso object.
B	Number of bootstrap iterations.
lambda	Vector of tuning parameters. If not specified (default), tuning parameters from cv.BTLLasso object are used. See also details.
cores	Number of cores for (parallelized) computation.
trace	Should the trace of the BTLLasso algorithm be printed?
trace.cv	Should the trace fo the cross-validation be printed? If parallelized, the trace is not working on Windows machines.
with.cv	Should cross-validation be performed separately on every bootstrap sample? If FALSE, the tuning parameter is fixed to the value chosen in the cv.BTLLasso object.

Details

The method can be highly time-consuming, for high numbers of tuning parameters, high numbers of folds in the cross-validation and high number of bootstrap iterations B. The number of tuning parameters can be reduced by specifying lambda in the boot.BTLLasso function. You can control if the range of prespecified tuning parameters was to small by looking at the output values lambda.max.alert and lambda.min.alert. They are set TRUE if the smallest or largest of the specifed lambda values was chosen in at least one bootstrap iteration.

Value

cv.model	cv.BTLLasso object
estimatesB	Matrix containing all B estimates for original parameters. For internal use.
estimatesBrepar	Matrix containing all B estimates for reparameterized (symmetric side constraints) parameters.
lambdaB	vector of used tuning parameters
lambda.max.alert	Was the largest value of lambda chosen in at least one bootstrap iteration?
lambda.min.alert	Was the smallest value of lambda chosen in at least one bootstrap iteration?
number.na	Total number of failed bootstrap iterations.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso, cv.BTLLasso, plot.boot.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

BTLLasso *Function to perform BTLLasso*

Description

Performs BTLLasso, a method to model heterogeneity in paired comparison data. Different types of covariates are allowed to have an influence on the attractivity/strength of the objects. Covariates can be subject-specific, object-specific or subject-object-specific. L1 penalties are used to reduce the complexiy of the model by enforcing clusters of equal effects or by elimination of irrelevant covariates.

Usage

```
BTLLasso(
  Y,
  X = NULL,
  Z1 = NULL,
  Z2 = NULL,
  lambda = NULL,
  control = ctrl.BTLLasso(),
  trace = TRUE
)
```

Arguments

Y	A response.BTLLasso object created by response.BTLLasso.
X	Matrix containing all **subject-specific covariates** that are to be included with **object-specific effects**. One row represents one subject, one column represents one covariate. X has to be standardized.
Z1	Matrix containing all **object-subject-specific covariates** that are to be included with **object-specific effects**. One row represents one subject, one column represents one combination between covariate and object. Column names have to follow the scheme 'firstvar.object1',...,'firstvar.objectm',...,'lastvar.objectm'. The object names 'object1',...,'objectm' have to be identical to the object names used in the response.BTLLasso object Y. The variable names and the object names have to be separated by '.'. The rownames of the matrix', Z.name, 'have to be equal to the subjects specified in the response object. Z1 has to be standardized.
Z2	Matrix containing all **object-subject-specific covariates or object-specific covariates** that are to be included with **global effects**. One row represents one subject, one column represents one combination between covariate and object. Column names have to follow the scheme 'firstvar.object1',...,'firstvar.objectm',...,'lastvar.objectm'. The object names 'object1',...,'objectm' have to be identical to the object names used in the response.BTLLasso object Y. The variable names and the object names have to be separated by '.'. The rownames of the matrix', Z.name, 'have to be equal to the subjects specified in the response object. Z2 has to be standardized.
lambda	Vector of tuning parameters. If NULL, automatically a grid of tuning parameters is created.
control	Function for control arguments, mostly for internal use. See also ctrl.BTLLasso.
trace	Should the trace of the BTLLasso algorithm be printed?

Value

coefs	Matrix containing all (original) coefficients, one row per tuning parameter, one column per coefficient.
coefs.repar	Matrix containing all reparameterized (for symmetric side constraint) coefficients, one row per tuning parameter, one column per coefficient.
logLik	Vector of log-likelihoods, one value per tuning parameter.
design	List containing design matrix and several additional information like, e.g., number and names of covariates.
Y	Response object.
penalty	List containing all penalty matrices and some further information on penalties.
response	Vector containing 0-1 coded response.
X	X matrix containing subject-specific covariates.
Z1	Z1 matrix containing subject-object-specific covariates.
Z2	Z2 matrix containing (subject)-object-specific covariates.

lambda	Vector of tuning parameters.
control	Control argument, specified by ctrl.BTLLasso.
df	Vector containing degrees of freedom for all models along the grid of tuning parameters.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

cv.BTLLasso, boot.BTLLasso, ctrl.BTLLasso, plot.BTLLasso, paths, print.BTLLasso, predict.BTLLasso, coef

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
```

```
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)
```

```
Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

Buli1415 *Bundesliga Data 2014/15 (Buli1415)*

Description

Data from the German Bundesliga from the season 2014/15. The data contain all 306 matches of the season treated as paired comparisons with 5 (Y5) or 3 (Y3) different response categories. Additionally, different match-specific covariates are given as, for example, the percentage of ball possession or the total running distance per team and per match.

Format

A list containing data from the German Bundesliga with 306 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y5 A response.BTLLasso object with 5 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Y3 A response.BTLLasso object with 3 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Z1 Matrix containing all team-match-specific covariates

- Distance: Total amount of km run
- BallPossession: Percentage of ball possession
- TacklingRate: Rate of won tacklings
- ShotsonGoal: Total number of shots on goal
- CompletionRate: Percentage of passes reaching teammates
- FoulsSuffered: Number of fouls suffered
- Offside: Number of offsides (in attack)

Source

https://www.kicker.de/

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

Buli1516, Buli1617, Buli1718

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(Buli1415)

Y <- Buli1415$Y5
Z1 <- scale(Buli1415$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)

par(op)

## End(Not run)
```

Buli1516 *Bundesliga Data 2015/16 (Buli1516)*

Description

Data from the German Bundesliga from the season 2015/16. The data contain all 306 matches of the season treated as paired comparisons with 5 (Y5) or 3 (Y3) different response categories. Additionally, different match-specific covariates are given as, for example, the percentage of ball possession or the total running distance per team and per match.

Format

A list containing data from the German Bundesliga with 306 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y5 A response.BTLLasso object with 5 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison

- with.order Vector containing information that each match has to be considered including an order effect.

Y3 A response.BTLLasso object with 3 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Z1 Matrix containing all team-match-specific covariates

- Distance: Total amount of km run
- BallPossession: Percentage of ball possession
- TacklingRate: Rate of won tacklings
- ShotsonGoal: Total number of shots on goal
- CompletionRate: Percentage of passes reaching teammates
- FoulsSuffered: Number of fouls suffered
- Offside: Number of offsides (in attack)

Z2 Matrix containing all the average market values of the teams as a team-specific covariate

@references Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

Source

https://www.kicker.de/

See Also

Buli1415, Buli1617, Buli1718

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(Buli1516)

Y <- Buli1516$Y5
Z1 <- scale(Buli1516$Z1, scale = FALSE)
```

```
ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)

par(op)

## End(Not run)
```

Buli1617 *Bundesliga Data 2016/17 (Buli1617)*

Description

Data from the German Bundesliga from the season 2016/17. The data contain all 306 matches of the season treated as paired comparisons with 5 (Y5) or 3 (Y3) different response categories. Additionally, different match-specific covariates are given as, for example, the percentage of ball possession or the total running distance per team and per match.

Format

A list containing data from the German Bundesliga with 306 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y5 A response.BTLLasso object with 5 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Y3 A response.BTLLasso object with 3 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison

- with.order Vector containing information that each match has to be considered including an order effect.

Z1 Matrix containing all team-match-specific covariates

- Distance: Total amount of km run
- BallPossession: Percentage of ball possession
- TacklingRate: Rate of won tacklings
- ShotsonGoal: Total number of shots on goal
- CompletionRate: Percentage of passes reaching teammates
- FoulsSuffered: Number of fouls suffered
- Offside: Number of offsides (in attack)
- Corners: Number of corners (in attack)

@references Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

Source

https://www.kicker.de/

See Also

Buli1415, Buli1516, Buli1718

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(Buli1617)

Y <- Buli1617$Y5
Z1 <- scale(Buli1617$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
```

```
plot(m.buli)

par(op)

## End(Not run)
```

Buli1718 *Bundesliga Data 2017/18 (Buli1718)*

Description

Data from the German Bundesliga from the season 2017/18. The data contain all 306 matches of the season treated as paired comparisons with 5 (Y5) or 3 (Y3) different response categories. Additionally, different match-specific covariates are given as, for example, the percentage of ball possession or the total running distance per team and per match.

Format

A list containing data from the German Bundesliga with 306 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y5 A response.BTLLasso object with 5 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Y3 A response.BTLLasso object with 3 response categories for the Buli1516 data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named team per paired comparison (home team)
- second.object: Vector containing the second-named team per paired comparison (away team)
- subject: Vector containing a match-day identifier per paired comparison
- with.order Vector containing information that each match has to be considered including an order effect.

Z1 Matrix containing all team-match-specific covariates

- Distance: Total amount of km run
- BallPossession: Percentage of ball possession
- TacklingRate: Rate of won tacklings
- ShotsonGoal: Total number of shots on goal
- CompletionRate: Percentage of passes reaching teammates
- FoulsSuffered: Number of fouls suffered

- Offside: Number of offsides (in attack)
- Corners: Number of corners (in attack)

@references Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

Source

https://www.kicker.de/

See Also

Buli1415, Buli1516, Buli1617

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(Buli1718)

Y <- Buli1718$Y5
Z1 <- scale(Buli1718$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)

par(op)

## End(Not run)
```

`BuliResponse` *Bundesliga Data Response Data (BuliResponse)*

Description

Data from the German Bundesliga from the season 2015/16. The data contain all variables from the 306 matches that are necessary to create the respective `response.BTLLasso` object from the data set `Buli1516`. The purpose of the data set is to provide an example how `response.BTLLasso` objects can be created.

Format

A data set containing all information that is necessary to create a response object for the Bundesliga data `link{Buli1516}`

Result Ordinal, 5-categorical results from Bundesliga season 2015/16.

TeamHome Abbreviation of home team.

TeamAway Abbreviation of away team.

Matchday Matchdays from 1 to 34.

@references Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

Source

https://www.kicker.de/

Examples

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

Y.Buli <- response.BTLLasso(response = BuliResponse$Result,
                            first.object = BuliResponse$TeamHome,
                            second.object = BuliResponse$TeamAway,
                            subject = BuliResponse$Matchday)

## End(Not run)
```

`ctrl.BTLLasso` *Control function for BTLLasso*

Description

Control parameters for different penalty terms and for tuning the fitting algorithm.

Usage

```
ctrl.BTLLasso(
  l.lambda = 30,
  log.lambda = TRUE,
  lambda.min = 0.05,
  adaptive = TRUE,
  scale = TRUE,
  norm = c("L1", "L2"),
  epsilon = 1e-04,
  lambda2 = 1e-04,
  c = 1e-09,
  precision = 3,
  weight.penalties = TRUE,
  include.intercepts = TRUE,
  order.effect = FALSE,
  object.order.effect = FALSE,
  order.center = FALSE,
  name.order = "Order",
  penalize.intercepts = FALSE,
  penalize.X = TRUE,
  penalize.Z2 = FALSE,
  penalize.Z1.absolute = TRUE,
  penalize.Z1.diffs = TRUE,
  penalize.order.effect.absolute = TRUE,
  penalize.order.effect.diffs = FALSE
)
```

Arguments

`l.lambda`	Number of tuning parameters. Applies only if `lambda = NULL` in the main function.
`log.lambda`	Should the grid of tuning parameters be created on a logarithmic scale rather than equidistant. Applies only if `lambda = NULL` in the main function.
`lambda.min`	Minimal value for tuning parameter. Applies only if `lambda = NULL` in the main function.
`adaptive`	Should adaptive lasso be used? Default is TRUE.

`scale` Should the covariates be scaled so that they are on comparable scales? Default is TRUE. Variables will be properly scaled AND centered. Please note that results will refer to scaled covariates. If `adaptive = TRUE` scaling is not necessary to keep penalties comparable.

`norm` Specifies the norm used in the penalty term. Currently, only 'L1' and 'L2' are possible. Default is to 'L1', only 'L1' allows for clustering and variable selection.

`epsilon` Threshold value for convergence of the algorithm.

`lambda2` Tuning parameter for ridge penalty on all coefficients. Should be small, only used to stabilize results.

`c` Internal parameter for the quadratic approximation of the L1 penalty. Should be sufficiently small.

`precision` Precision for final parameter estimates, specifies number of decimals.

`weight.penalties`
Should the penalties across the different model components (i.e. intercepts, order effects, X, Z1, Z2) be weighted according to the number of penalties included? Default is `TRUE` to minimize the risk of selection bias across different model components.

`include.intercepts`
Should intercepts be included in the model?

`order.effect` Should a global order effect (corresponding to home effect in sports applications) be included in the model?

`object.order.effect`
Should object-specific order effects (corresponding to home effects in sports applications) be included in the model?

`order.center` Should (in case of object-specific order effects) the order effects be centered in the design matrix? Centering is equivalent to the coding scheme of effect coding instead of dummy coding.

`name.order` How should the order effect(s) be called in plots or prints.

`penalize.intercepts`
Should intercepts be penalized? If `TRUE`, all pairwise differences between intercepts are penalized.

`penalize.X` Should effects from X matrix be penalized? If `TRUE`, all pairwise differences corresponding to one covariate are penalized. Can also be used with a character vector as input. Then, the character vector contains the names of the variables from X whose parameters should be penalized.

`penalize.Z2` Should absolute values of effects from Z2 matrix be penalized? Can also be used with a character vector as input. Then, the character vector contains the names of the variables from Z2 whose parameters should be penalized.

`penalize.Z1.absolute`
Should absolute values of effects from Z1 matrix be penalized? Can also be used with a character vector as input. Then, the character vector contains the names of the variables from Z1 whose parameters should be penalized.

`penalize.Z1.diffs`
: Should differences of effects from Z1 matrix be penalized? If `TRUE`, all pairwise differences corresponding to one covariate are penalized. Can also be used with a character vector as input. Then, the character vector contains the names of the variables from Z1 whose parameters should be penalized.

`penalize.order.effect.absolute`
: Should absolute values of order effect(s) be penalized? Only relevant if either `object.order.effect = TRUE` or `order.effect = TRUE`.

`penalize.order.effect.diffs`
: Should differences of order effects be penalized? If `TRUE`, all pairwise differences are penalized. Only relevant if `object.order.effect = TRUE`

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

`BTLLasso`, `cv.BTLLasso`

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim
```

```
par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')
```

```
##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

---

cv.BTLLasso *Cross-validation function for BTLLasso*

Description

Performs cross-validation of BTLLasso, including the BTLLasso algorithm for the whole data set.

Usage

```
cv.BTLLasso(
  Y,
  X = NULL,
  Z1 = NULL,
  Z2 = NULL,
  folds = 10,
  lambda = NULL,
  control = ctrl.BTLLasso(),
  cores = folds,
  trace = TRUE,
  trace.cv = TRUE,
  cv.crit = c("RPS", "Deviance")
)
```

Arguments

Y	A response.BTLLasso object created by response.BTLLasso.
X	Matrix containing all **subject-specific covariates** that are to be included with **object-specific effects**. One row represents one subject, one column represents one covariate. X has to be standardized.
Z1	Matrix containing all **object-subject-specific covariates** that are to be included with **object-specific effects**. One row represents one subject, one column represents one combination between covariate and object. Column names have to follow the scheme 'firstvar.object1',...,'firstvar.objectm',...,'lastvar.objectm'. The object names 'object1',...,'objectm' have to be identical to the object names used in the response.BTLLasso object Y. The variable names and the object names have to be separated by '.'. The rownames of the matrix', Z.name, 'have to be equal to the subjects specified in the response object. Z1 has to be standardized.
Z2	Matrix containing all **object-subject-specific covariates or object-specific covariates** that are to be included with **global effects**. One row represents one subject, one column represents one combination between covariate and object. Column names have to follow the scheme 'firstvar.object1',...,'firstvar.objectm',...,'lastvar.objectm'. The object names 'object1',...,'objectm' have to be identical to the object names used in the response.BTLLasso object Y. The variable names and the object names have to be separated by '.'. The rownames of the matrix', Z.name, 'have to be equal to the subjects specified in the response object. Z2 has to be standardized.
folds	Number of folds for the crossvalidation. Default is 10.
lambda	Vector of tuning parameters. If NULL, automatically a grid of tuning parameters is created.
control	Function for control arguments, mostly for internal use. See also ctrl.BTLLasso.
cores	Number of cores used for (parallelized) cross-validation. By default, equal to the number of folds.
trace	Should the trace of the BTLLasso algorithm be printed?

trace.cv	Should the trace fo the cross-validation be printed? If parallelized, the trace is not working on Windows machines.
cv.crit	Which criterion should be used to evaluate cross-validation. Choice is between Ranked probability score and deviance. Only RPS considers the ordinal structure of the response.

Details

Cross-validation can be performed parallel, default is 10-fold cross-validation on 10 cores. Output is a cv.BTLLasso object which can then be used for bootstrap intervalls using boot.BTLLasso.

Value

coefs	Matrix containing all (original) coefficients, one row per tuning parameter, one column per coefficient.
coefs.repar	Matrix containing all reparameterized (for symmetric side constraint) coefficients, one row per tuning parameter, one column per coefficient.
logLik	Vector of log-likelihoods, one value per tuning parameter.
design	List containing design matrix and several additional information like, e.g., number and names of covariates.
Y	Response object.
penalty	List containing all penalty matrices and some further information on penalties
response	Vector containing 0-1 coded response.
X	X matrix containing subject-specific covariates.
Z1	Z1 matrix containing subject-object-specific covariates.
Z2	Z2 matrix containing (subject)-object-specific covariates.
lambda	Vector of tuning parameters.
control	Control argument, specified by ctrl.BTLLasso.
criterion	Vector containing values of the chosen cross-validation criterion, one value per tuning parameter.
folds	Number of folds in cross validation.
cv.crit	Cross-validation criterion, either RPS or Deviance.
df	Vector containing degrees of freedom for all models along the grid of tuning parameters.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso, boot.BTLLasso, ctrl.BTLLasso, plot.BTLLasso, paths, print.cv.BTLLasso, predict.BTLLasso, coef

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
```

```
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

GLES *German Longitudinal Election Study (GLES)*

Description

Data from the German Longitudinal Election Study (GLES), see Rattinger et al. (2014). The GLES is a long-term study of the German electoral process. It collects pre- and post-election data for several federal elections, the data used here originate from the pre-election study for 2013.

Format

A list containing data from the German Longitudinal Election Study with 2003 (partly incomplete) observations. The list contains both information on the response (paired comparisons) and different covariates.

Y A response.BTLLasso object for the GLES data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named party per paired comparison
- second.object: Vector containing the second-named party per paired comparison
- subject: Vector containing a person identifier per paired comparison
- with.order Automatically generated vector containing information on order effect. Irrelevant, because no order effect needs to be included in the analysis of GLES data.

X Matrix containing all eight person-specific covariates

- Age: Age in years
- Gender (0: male, 1: female)
- EastWest (0: West Germany, 1: East Germany)
- PersEcon: Personal economic situation, 1: good or very good, 0: else
- Abitur: School leaving certificate, 1: Abitur/A levels, 0: else
- Unemployment: 1: currently unemployed, 0: else
- Church: Frequency of attendence in a church/synagogue/mosque/..., 1: at least once a month, 0: else
- Migration: Are you a migrant / not German since birth? 1: yes, 0: no

Z1 Matrix containing all four person-party-specific covariates

- Climate: Self-perceived distance of each person to all five parties with respect to ones attitude towards climate change.
- SocioEcon: Self-perceived distance of each person to all five parties with respect to ones attitude towards socio-economic issues.
- Immigration: Self-perceived distance of each person to all five parties with respect to ones attitude towards immigration.

Source

https://www.gesis.org/en/gles/about-gles

References

Rattinger, H., S. Rossteutscher, R. Schmitt-Beck, B. Wessels, and C. Wolf (2014): Pre-election cross section (GLES 2013). *GESIS Data Archive, Cologne ZA5700 Data file Version 2.0.0.*

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(GLES)
Y <- GLES$Y
X <- scale(GLES$X, scale = FALSE)

subs <- c("(in years)","female (1); male (0)","East Germany (1); West Germany (0)",
          "(very) good (1); else (0)", "Abitur/A levels (1); else (0)",
          "currently unemployed (1); else (0)","at least once a month (1); else (0)",
          "yes (1); no (0)")

set.seed(5)
m.gles <- cv.BTLLasso(Y = Y, X = X, control = ctrl.BTLLasso(l.lambda = 50))
```

```
par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles, subs.X = subs)

par(op)

## End(Not run)
```

GLESsmall *Subset of the GLES data set with 200 observations and 4 covariates.*

Description

This is a subset of the GLES data set from the German Longitudinal Election Study (GLES), see Rattinger et al. (2014). The subset contains only 200 of the 2003 observations and only a small part of the covariates. The GLES is a long-term study of the German electoral process. It collects pre- and post-election data for several federal elections, the data used here originate from the pre-election study for 2013.

Format

A list containing data from the German Longitudinal Election Study with 200 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y A response.BTLLasso object for the GLES data including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named party per paired comparison
- second.object: Vector containing the second-named party per paired comparison
- subject: Vector containing a person identifier per paired comparison
- with.order Automatically generated vector containing information on order effect. Irrelevant, because no order effect needs to be included in the analysis of GLES data.

X Matrix containing all eight person-specific covariates

- Age: Age in years
- Gender (0: male, 1: female)

Z1 Matrix containing all four person-party-specific covariates

- Climate: Self-perceived distance of each person to all five parties with respect to ones attitude towards climate change.
- Immigration: Self-perceived distance of each person to all five parties with respect to ones attitude towards immigration.

Source

https://www.gesis.org/en/gles/about-gles

References

Rattinger, H., S. Rossteutscher, R. Schmitt-Beck, B. Wessels, and C. Wolf (2014): Pre-election cross section (GLES 2013). *GESIS Data Archive, Cologne ZA5700 Data file Version 2.0.0.*

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, to appear

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

See Also

GLES

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')

par(op)

## End(Not run)
```

paths *Plot covariate paths for BTLLasso*

Description

Plots paths for every covariate of a BTLLasso object or a cv.BTLLasso object. In contrast to `plot.BTLLasso`, only one plot is created, every covariate is illustrated by one path. For `cv.BTLLasso` objects, the optimal model according to the cross-validation is marked by a vertical dashed line.

Usage

```
paths(model, y.axis = c("penalty", "L2"), x.axis = c("loglambda", "lambda"))
```

Arguments

`model`	`BTLLasso` or `cv.BTLLasso` object
`y.axis`	Two possible values for the y-axis. Variables can either be plotted with regard to their contribution to the total penalty term (`y.axis='penalty'`) or with regard to the L_2 norm of the corresponding parameter vector (`y.axis='L2'`).
`x.axis`	Should the paths be plotted against log(lambda+1) or against lambda?'

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

`BTLLasso`, `cv.BTLLasso`, `plot.BTLLasso`

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
```

```
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
```

```
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)
```

```
## End(Not run)
```

`plot.boot.BTLLasso` *Plot bootstrap intervals for BTLLasso*

Description

Plots bootstrap intervals for every single coefficient based on bootstrap estimates calculated by `boot.BTLLasso`. Bootstrap intervals are separated by covariates, every covariate is plotted separately.

Usage

```
## S3 method for class 'boot.BTLLasso'
plot(
  x,
  quantiles = c(0.025, 0.975),
  plots_per_page = 1,
  ask_new = TRUE,
  rescale = FALSE,
  which = "all",
  include.zero = TRUE,
  rows = NULL,
  subs.X = NULL,
  subs.Z1 = NULL,
  main.Z2 = "Obj-spec. Covariates",
  ...
)
```

Arguments

`x`	boot.BTLLasso object
`quantiles`	Which empirical quantiles of the bootstrap estimates should be plotted?
`plots_per_page`	Number of plots per page, internally specified by `par(mfrow=...)`.
`ask_new`	If TRUE, the user is asked before each plot.
`rescale`	Should the parameter estimates be rescaled for plotting? Only applies if `scale = TRUE` was specified in `cv.BTLLasso`.
`which`	Integer vector to specify which parameters/variables to plot.
`include.zero`	Should all plots contain zero?
`rows`	Optional argument for the number of rows in the plot. Only applies if `plots_per_page>1`.
`subs.X`	Optional vector of subtitles for variables in `X`. Can be used to note the encoding of the single covariates, especially for dummy variables.
`subs.Z1`	Optional vector of subtitles for variables in `Z1`. Can be used to note the encoding of the single covariates, especially for dummy variables.
`main.Z2`	Optional character containg main for plot containing intervals for Z2 parameters.
`...`	other parameters to be passed through to plot function.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

boot.BTLLasso, BTLLasso, cv.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)
```

```
head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
```

```
                            penalize.order.effect.diffs = TRUE,
                            penalize.order.effect.absolute = FALSE,
                            order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`plot.BTLLasso` *Plot parameter paths for BTLLasso*

Description

Plots single paths for every parameter of a `BTLLasso` object or a `cv.BTLLasso` object. In contrast, to `paths`, one plot per covariate is created, every single parameter is illustrated by one path. For `cv.BTLLasso` objects, the optimal model according to the cross-validation is marked by a vertical dashed line.

Usage

```
## S3 method for class 'BTLLasso'
plot(
  x,
  plots_per_page = 1,
  ask_new = TRUE,
  rescale = FALSE,
  which = "all",
```

```
  equal.ranges = FALSE,
  x.axis = c("loglambda", "lambda"),
  rows = NULL,
  subs.X = NULL,
  subs.Z1 = NULL,
  main.Z2 = "Obj-spec. Covariates",
  ...
)
```

Arguments

x	BTLLasso or cv.BTLLasso object
plots_per_page	Number of plots per page, internally specified by par(mfrow=...).
ask_new	If TRUE, the user is asked before each plot.
rescale	Should the parameter estimates be rescaled for plotting? Only applies if scale = TRUE was specified in BTLLasso or cv.BTLLasso.
which	Integer vector to specify which parameters/variables to plot.
equal.ranges	Should all single plots (for different covariates) have equal ranges on the y-axes. FALSE by default.
x.axis	Should the paths be plotted against log(lambda+1) or against lambda?
rows	Optional argument for the number of rows in the plot. Only applies if plots_per_page>1.
subs.X	Optional vector of subtitles for variables in X. Can be used to note the encoding of the single covariates, especially for dummy variables.
subs.Z1	Optional vector of subtitles for variables in Z1. Can be used to note the encoding of the single covariates, especially for dummy variables.
main.Z2	Optional character containg main for plot containing intervals for Z2 parameters.
...	Further plot arguments.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso, cv.BTLLasso, paths

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
```

```
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")
```

```
set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

---

predict.BTLLasso *Predict function for BTLLasso*

Description

Predict function for a BTLLasso object or a cv.BTLLasso object. Predictions can be linear predictors, probabilities or the values of the latent traits for both competitors in the paired comparisons.

Usage

```
## S3 method for class 'BTLLasso'
predict(object, newdata = list(), type = c("link", "response", "trait"), ...)
```

Arguments

object	BTLLasso or cv.BTLLasso object
newdata	List possibly containing slots Y, X, Z1 and Z2 to use new data for prediction.
type	Type "link" gives the linear predictors for separate categories, type "response" gives the respective probabilities. Type "trait" gives the estimated latent traits of both competitors/objects in the paired comparisons.
...	Further predict arguments.

Details

Results are lists of matrices with prediction for every single tuning parameter for BTLLasso objects and a single matrix for cv.BTLLasso objects.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso, cv.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`print.boot.BTLLasso` *Print function for boot.BTLLasso objects*

Description

Prints the most important output of `boot.BTLLasso` objects.

Usage

```
## S3 method for class 'boot.BTLLasso'
print(x, quantiles = c(0.025, 0.975), rescale = FALSE, ...)
```

Arguments

`x`	`boot.BTLLasso` object
`quantiles`	Which empirical quantiles of the bootstrap estimates should be printed?
`rescale`	Should the parameter estimates be rescaled for plotting? Only applies if `scale = TRUE` was specified in `BTLLasso` or `cv.BTLLasso`.
`...`	possible further arguments for print command

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

boot.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`print.BTLLasso` *Print function for BTLLasso objects*

Description

Prints the most important output of `BTLLasso` objects.

Usage

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

Arguments

`x`	`BTLLasso` object
`...`	possible further arguments for print command

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`print.cv.BTLLasso` *Print function for cv.BTLLasso objects*

Description

Prints the most important output of `cv.BTLLasso` objects.

Usage

```
## S3 method for class 'cv.BTLLasso'
print(x, rescale = FALSE, ...)
```

Arguments

`x`	`cv.BTLLasso` object
`rescale`	Should the parameter estimates be rescaled for plotting? Only applies if `scale = TRUE` was specified in `BTLLasso` or `cv.BTLLasso`.
`...`	possible further arguments for print command

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

cv.BTLLasso

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

##############################
##### Example with simulated data set containing X, Z1 and Z2
##############################
data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)
```

```
## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)


##############################
##### Example with small version from GLES data set
##############################
data(GLESsmall)

## extract data and center covariates for better interpretability
Y <- GLESsmall$Y
X <- scale(GLESsmall$X, scale = FALSE)
Z1 <- scale(GLESsmall$Z1, scale = FALSE)

## vector of subtitles, containing the coding of the X covariates
subs.X <- c('', 'female (1); male (0)')

## Cross-validate BTLLasso model
m.gles.cv <- cv.BTLLasso(Y = Y, X = X, Z1 = Z1)
m.gles.cv

coef(m.gles.cv)
logLik(m.gles.cv)

head(predict(m.gles.cv, type="response"))
head(predict(m.gles.cv, type="trait"))

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.gles.cv, subs.X = subs.X, plots_per_page = 4, which = 2:5)
paths(m.gles.cv, y.axis = 'L2')


##############################
##### Example with Bundesliga data set
##############################
data(Buli1516)

Y <- Buli1516$Y5

Z1 <- scale(Buli1516$Z1, scale = FALSE)

ctrl.buli <- ctrl.BTLLasso(object.order.effect = TRUE,
                           name.order = "Home",
                           penalize.order.effect.diffs = TRUE,
                           penalize.order.effect.absolute = FALSE,
                           order.center = TRUE, lambda2 = 1e-2)

set.seed(1860)
m.buli <- cv.BTLLasso(Y = Y, Z1 = Z1, control = ctrl.buli)
```

```
m.buli

par(xpd = TRUE, mar = c(5,4,4,6))
plot(m.buli)


##############################
##### Example with Topmodel data set
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)
plot(m.models, plots_per_page = 6)

par(op)

## End(Not run)
```

`response.BTLLasso` *Create response object for BTLLasso*

Description

Create a response object for `BTLLasso` and `cv.BTLLasso`

Usage

```
response.BTLLasso(
  response,
  first.object = NULL,
  second.object = NULL,
  subject = NULL,
  with.order = rep(TRUE, length(response))
)
```

Arguments

`response`	Vector containing results (binary or ordinal) of single paired comparisons. Alternatively, also a `paircomp` object as defined in the package `psychotools` could be used. In this case, none of the further arguments are needed.
`first.object`	Vector (character or factor, same length as response) indicating the first object of the respective paired comparison from response.

`second.object` Vector (character or factor, same length as response) indicating the second object of the respective paired comparison from response.

`subject` Vector (character, same length as response) indicating the subject that generated the respective paired comparison from response.

`with.order` Boolean vector containing indicators for each paired comparison if an order effect was present. By default, an order effect is assumed for each comparison. This option is relevant whenever only some of the paired comparisons had an order effect and others did not, for example if some matches are played on neutral ground. This option is only effective if either `order.effect = TRUE` or `object.order.effect = TRUE`.

Value

Object of class `response.BTLLasso`

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, Gunther and Tutz, Gerhard (2019): BTLLasso - A Common Framework and Software Package for the Inclusion and Selection of Covariates in Bradley-Terry Models, *Journal of Statistical Software*, 88(9), 1-29, doi:10.18637/jss.v088.i09

Schauberger, Gunther and Tutz, Gerhard (2017): Subject-specific modelling of paired comparison data: A lasso-type penalty approach, *Statistical Modelling*, 17(3), 223 - 243

Schauberger, Gunther, Groll Andreas and Tutz, Gerhard (2018): Analysis of the importance of on-field covariates in the German Bundesliga, *Journal of Applied Statistics*, 45(9), 1561 - 1578

See Also

`BTLLasso`, `cv.BTLLasso`

Examples

```
## Not run:
##############################
##### Example how response object for Bundesliga data Buli1516 was created
##############################

data(BuliResponse)

Y.Buli <- response.BTLLasso(response = BuliResponse$Result,
                            first.object = BuliResponse$TeamHome,
                            second.object = BuliResponse$TeamAway,
                            subject = BuliResponse$Matchday)
```

```
##############################
##### Example to create response object from paircomp object
##############################
data("Topmodel2007", package = "psychotree")

Y.models <- response.BTLLasso(Topmodel2007$preference)
X.models <- scale(model.matrix(preference~., data = Topmodel2007)[,-1])
rownames(X.models) <- paste0("Subject",1:nrow(X.models))
colnames(X.models) <- c("Gender","Age","KnowShow","WatchShow","WatchFinal")

set.seed(5)
m.models <- cv.BTLLasso(Y = Y.models, X = X.models)

## End(Not run)
```

SimData *Simulated data set for illustration*

Description

This data set is a simulated data set including all possible types of covariates (X, Z1 and Z2) and is intended to serve for illustration purpose. The data set contains paired comparisons between four objects with five different response categories from 200 subjects.

Format

A list containing simulated data for 200 observations. The list contains both information on the response (paired comparisons) and different covariates.

Y A response.BTLLasso object with simulated responses including

- response: Ordinal paired comparison response vector
- first.object: Vector containing the first-named object per paired comparison
- second.object: Vector containing the second-named object per paired comparison
- subject: Vector containing a subject identifier per paired comparison
- with.order Automatically generated vector containing information on order effect. Each paired comparison is associated with an order effect.

X Matrix containing both subject-specific covariates

- X_var1
- X_var2

Z1 Matrix containing both subject-object-specific covariates

- Z1_var1
- Z1_var2

Z2 Matrix containing both object-specific covariates

- Z2_var1
- Z2_var2

Examples

```
## Not run:
op <- par(no.readonly = TRUE)

data(SimData)

## Specify control argument
## -> allow for object-specific order effects and penalize intercepts
ctrl <- ctrl.BTLLasso(penalize.intercepts = TRUE, object.order.effect = TRUE,
                      penalize.order.effect.diffs = TRUE)

## Simple BTLLasso model for tuning parameters lambda
m.sim <- BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                  Z2 = SimData$Z2, control = ctrl)
m.sim

par(xpd = TRUE)
plot(m.sim)


## Cross-validate BTLLasso model for tuning parameters lambda
set.seed(1860)
m.sim.cv <- cv.BTLLasso(Y = SimData$Y, X = SimData$X, Z1 = SimData$Z1,
                        Z2 = SimData$Z2, control = ctrl)
m.sim.cv
coef(m.sim.cv)
logLik(m.sim.cv)

head(predict(m.sim.cv, type="response"))
head(predict(m.sim.cv, type="trait"))

plot(m.sim.cv, plots_per_page = 4)


## Example for bootstrap intervals for illustration only
## Don't calculate bootstrap intervals with B = 20!!!!
set.seed(1860)
m.sim.boot <- boot.BTLLasso(m.sim.cv, B = 20, cores = 20)
m.sim.boot
plot(m.sim.boot, plots_per_page = 4)

par(op)

## End(Not run)
```

Index

∗ **BTLLasso**
boot.BTLLasso, 5
BTLLasso, 9
BTLLasso-package, 2
ctrl.BTLLasso, 22
cv.BTLLasso, 26
paths, 35
plot.boot.BTLLasso, 38
plot.BTLLasso, 41
predict.BTLLasso, 45
print.boot.BTLLasso, 48
print.BTLLasso, 51
print.cv.BTLLasso, 54
∗ **BTL**
BTLLasso-package, 2
∗ **Bradley-Terry**
BTLLasso-package, 2
∗ **bootstrap**
boot.BTLLasso, 5
plot.boot.BTLLasso, 38
∗ **control**
ctrl.BTLLasso, 22
∗ **covariate**
paths, 35
∗ **cross**
cv.BTLLasso, 26
∗ **datasets**
Buli1415, 13
Buli1516, 15
Buli1617, 17
Buli1718, 19
BuliResponse, 21
GLES, 31
GLESsmall, 33
SimData, 59
∗ **interval**
boot.BTLLasso, 5
plot.boot.BTLLasso, 38
∗ **package**
BTLLasso-package, 2
∗ **parameter**
plot.BTLLasso, 41
predict.BTLLasso, 45
∗ **paths**
paths, 35
plot.BTLLasso, 41
predict.BTLLasso, 45
∗ **validation**
cv.BTLLasso, 26

boot.BTLLasso, 5, *11*, *28, 29*, *38, 39*, *49*
BTLLasso, *3*, *7*, 9, *24*, *29*, *35*, *39*, *42*, *46*, *52*, *58*
BTLLasso-package, 2
Buli1415, 13, *16*, *18*, *20*
Buli1516, *14*, 15, *18*, *20, 21*
Buli1617, *14*, *16*, 17, *20*
Buli1718, *14*, *16*, *18*, 19
BuliResponse, 21

coef, *11*, *29*
ctrl.BTLLasso, *10, 11*, 22, *27–29*
cv.BTLLasso, *3*, *7*, *11*, *24*, 26, *35*, *39*, *42*, *46*, *55*, *58*

GLES, 31, *33, 34*
GLESsmall, 33

paircomp, *57*
paths, *11*, *29*, 35, *41, 42*
plot, *5*
plot.boot.BTLLasso, *7*, 38
plot.BTLLasso, *11*, *29*, *35*, 41
predict.BTLLasso, *11*, *29*, 45
print.boot.BTLLasso, 48
print.BTLLasso, *11*, 51
print.cv.BTLLasso, *29*, 54

response.BTLLasso, *10*, *27*, 57

SimData, 59