Package 'GPCMlasso'

September 8, 2026

Type Package

Title Regularized Explanatory Generalized Partial Credit Models

Version 0.2-0

Date 2026-09-07

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

Description Fits explanatory generalized partial credit models and related ordinal item response models with global and item-specific covariate effects. Penalized marginal maximum likelihood estimation is used for variable selection, detection of differential item functioning, and clustering of item-specific covariate effects by fusion penalties. The package extends the regularization approach for differential item functioning in generalized partial credit models proposed by Schauberger and Mair (2020) <doi:10.3758/s13428-019-01224-2>.

License GPL (>= 2)

Imports Rcpp (>= 0.12.4), TeachingDemos, cubature, caret, statmod, mvtnorm, mirt, methods

Depends ltm

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Gunther Schauberger [aut, cre]

Repository CRAN

Date/Publication 2026-09-08 11:10:03 UTC

Contents

GPCMlasso-package 2
ctrl_GPCMlasso 4
GPCMlasso 7
plot.GPCMlasso 11
predict.GPCMlasso 12
tenseness 14

tenseness_small 15
trait.posterior 17

Index **19**

`GPCMlasso-package` *GPCMlasso: Regularized Explanatory Generalized Partial Credit Models*

Description

The **GPCMlasso** package fits generalized partial credit models and related ordinal item response models with explanatory variables. It supports models with global covariate effects, item-specific covariate effects, and regularization terms for variable selection and the clustering of item-specific effects by fusion penalties.

Details

The original functionality of the package focuses on the detection of differential item functioning (DIF) in generalized partial credit models. In this framework, DIF effects are explicitly parametrized and selected by penalized marginal maximum likelihood estimation. The package additionally provides tools for fitting explanatory generalized partial credit models with global and item-specific covariate effects, including full-fusion penalties to identify clusters of items with equal covariate effects.

Several variables can be considered simultaneously, and both continuous and categorical covariates can be included. The package covers generalized partial credit models and special cases such as partial credit models, rating scale models, and models for binary item responses.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, G. and Mair, P. (2020). A regularization approach for the detection of differential item functioning in generalized partial credit models. *Behavior Research Methods*, 52, 279–294. doi:10.3758/s13428019012242

See Also

`GPCMlasso`, `ctrl_GPCMlasso`, `plot.GPCMlasso`, `predict.GPCMlasso`, `trait.posterior`

Examples

```
data(tenseness_small)

## Formula with five ordinal items and two covariates
form_cov <- as.formula(
  paste(
    "cbind(",
    paste(colnames(tenseness_small)[1:5], collapse = ","),
    ") ~ Gender + Age"
  )
)

## Not run:
## ------------------------------------------------------------
## Classical DIF analysis
## ------------------------------------------------------------
fit_dif <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    trace = FALSE
  )
)

summary(fit_dif)
plot(
  fit_dif,
  type = "DIF",
  panels_per_page = 2,
  columns = 2,
  ask_new = FALSE
)

## ------------------------------------------------------------
## Explanatory GPCM-X with global and item-specific effects
## and fusion penalty
## ------------------------------------------------------------
set.seed(1860)

fit_fusion <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  main.effects = TRUE,
  cv = TRUE,
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    folds = 2,
```

```
    penalize.main.effects = TRUE,
    include.item.effects = TRUE,
    fuse.per.variable = TRUE,
    scale.covariate.effects = FALSE,
    trace = FALSE
  )
)

summary(
  fit_fusion,
  effect.scale = "effect"
)

## Available covariate panels
fit_fusion$design_list$x.names

plot(
  fit_fusion,
  type = "Variable",
  panels = fit_fusion$design_list$x.names,
  panels_per_page = 2,
  columns = 2,
  variable.scale = "effect",
  select = "cv",
  ask_new = FALSE
)

## End(Not run)
```

`ctrl_GPCMlasso` *Control function for GPCMlasso*

Description

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

Usage

```
ctrl_GPCMlasso(
  log.lambda = TRUE,
  lambda = NULL,
  l.lambda = 50,
  lambda.min = 0.1,
  penalize.main.effects = FALSE,
  fuse.per.variable = FALSE,
  scale.covariate.effects = TRUE,
  include.item.effects = TRUE,
  adaptive = TRUE,
  weight.penalties = TRUE,
```

```
  ada.lambda = 1e-04,
  ada.power = 1,
  Q = 15,
  lambda2 = 1e-04,
  cvalue = 1e-05,
  trace = TRUE,
  folds = 10,
  cores = 25,
  null_thresh = 0.01,
  gradtol = 1e-06,
  steptol = 1e-06,
  iterlim = 500,
  precision = 3,
  all.dummies = FALSE,
  ctrl.gpcm = list()
)
```

Arguments

`log.lambda`	Logical. Should the grid of tuning parameters be created on a log scale?
`lambda`	Optional vector of tuning parameters. If `NULL`, a grid of length `l.lambda` is created automatically.
`l.lambda`	Length of the tuning parameter grid.
`lambda.min`	Minimal value used for the automatically created tuning parameter grid.
`penalize.main.effects`	Logical. Should global covariate effects be penalized?
`fuse.per.variable`	Logical. Should item-specific effects referring to the same covariate be fused across items?
`scale.covariate.effects`	Logical. If `TRUE`, covariate effects are multiplied by item discrimination parameters as in the original **GPCMlasso** parameterization. If `FALSE`, covariate effects enter the linear predictor directly.
`include.item.effects`	Logical. If `TRUE`, item-specific covariate effects are included. If `FALSE`, only global covariate effects are estimated.
`adaptive`	Logical. Should adaptive lasso weights be used?
`weight.penalties`	Logical. Should penalties be weighted according to the number of penalty terms and corresponding parameters?
`ada.lambda`	Tuning parameter for ridge-regularized estimation of adaptive weights.
`ada.power`	Power used for adaptive weights.
`Q`	Number of nodes used in Gauss-Hermite quadrature.
`lambda2`	Ridge penalty on all coefficients except slope parameters.
`cvalue`	Internal parameter for the quadratic approximation of the L_1 penalty.

trace	Logical. Should progress information be printed?
folds	Number of folds for cross-validation.
cores	Number of cores used for parallel computation.
null_thresh	Threshold used to distinguish zero from non-zero values.
gradtol	Parameter for optimization accuracy. See nlm.
steptol	Parameter for optimization accuracy. See nlm.
iterlim	Parameter for optimization accuracy. See nlm.
precision	Number of decimal places used to round coefficient estimates in selected output methods.
all.dummies	Logical. Should all dummy variables be included for factors with more than two categories?
ctrl.gpcm	List of control arguments for the initial gpcm fit used to obtain starting values.

Value

A list of control parameters used by GPCMlasso.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, G. and Mair, P. (2020). A regularization approach for the detection of differential item functioning in generalized partial credit models. *Behavior Research Methods*, 52, 279–294. doi:10.3758/s13428019012242

See Also

GPCMlasso, plot.GPCMlasso

Examples

```
ctrl <- ctrl_GPCMlasso()
str(ctrl)

ctrl_small <- ctrl_GPCMlasso(
  l.lambda = 5,
  cores = 1,
  trace = FALSE
)

ctrl_fusion <- ctrl_GPCMlasso(
  l.lambda = 5,
  cores = 1,
  penalize.main.effects = TRUE,
  include.item.effects = TRUE,
```

```
  fuse.per.variable = TRUE,
  scale.covariate.effects = FALSE,
  trace = FALSE
)
```

GPCMlasso *Fit Regularized Generalized Partial Credit Models*

Description

Fits generalized partial credit models and related ordinal item response models with optional explanatory variables. The function can be used for differential item functioning (DIF) detection and for explanatory item response modelling with global and item-specific covariate effects.

Usage

```
GPCMlasso(
  formula,
  data,
  DSF = FALSE,
  model = c("PCM", "RSM", "GPCM", "GRSM", "RM", "2PL"),
  control = ctrl_GPCMlasso(),
  cv = FALSE,
  main.effects = TRUE
)
```

Arguments

`formula`	Formula specifying the item responses and optional covariates. Item responses must be supplied on the left-hand side using `cbind()`. Covariates are specified on the right-hand side. If the right-hand side is `~ 0`, a model without covariates is fitted.
`data`	Data frame containing the ordinal item response data and all covariates.
`DSF`	Logical. Should differential step functioning (DSF) be considered? If `DSF = TRUE`, one parameter per step between two adjacent response categories is introduced. For binary items, DSF and DIF coincide.
`model`	Character string specifying the underlying item response model. Available options are `"PCM"`, `"RSM"`, `"GPCM"`, `"GRSM"`, `"RM"`, and `"2PL"`. Generalized models allow item-specific discrimination parameters. `"RM"` and `"2PL"` are available for binary item responses only.
`control`	Control object specifying algorithmic and numerical optimization options, usually created by `ctrl_GPCMlasso`.
`cv`	Logical. Should cross-validation be performed? Cross-validation can be used for tuning parameter selection.
`main.effects`	Logical. Should global covariate effects be included for models with covariates? Additional item-specific effects can be enabled via `include.item.effects` in `ctrl_GPCMlasso`.

Details

A joint parametric model is set up based on an item response model chosen by the user. Several covariates can be considered simultaneously. For DIF models, item-specific covariate effects are introduced and selected by a lasso-type penalty. For explanatory GPCM-X models, global covariate effects and item-specific deviations can be included, and fusion penalties can be used to cluster item-specific effects referring to the same covariate.

Value

An object of class `"GPCMlasso"`, a list containing:

`coefficients`	Matrix of estimated parameters, one row per tuning parameter value.
`logLik`	Vector of log-likelihood values, one per tuning parameter.
`cv_error`	Vector of cross-validation errors. Only relevant if `cv = TRUE`.
`call`	The matched function call.
`model`	The selected item response model.
`data`	The data used for model fitting.
`control`	The control list used for model fitting.
`DSF`	Logical value indicating whether DSF was used.
`formula`	The model formula.
`item.names`	Item names.
`Y`	Matrix containing the item responses.
`design_list`	List of design matrices and auxiliary objects used internally.
`AIC`	Vector of AIC values, one per tuning parameter.
`BIC`	Vector of BIC values, one per tuning parameter.
`cAIC`	Vector of corrected AIC values, one per tuning parameter.
`df`	Vector of degrees of freedom, one per tuning parameter.
`coef.rescal`	Matrix of parameters rescaled to their original scales.
`main.effects`	Logical value indicating whether global covariate effects were included.
`scale_cols`	Internal indicator specifying which design columns are multiplied by item discrimination parameters.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

References

Schauberger, G. and Mair, P. (2020). A regularization approach for the detection of differential item functioning in generalized partial credit models. *Behavior Research Methods*, 52, 279–294. doi:10.3758/s13428019012242

See Also

GPCMlasso-package, ctrl_GPCMlasso, plot.GPCMlasso, predict.GPCMlasso, trait.posterior

Examples

```
data(tenseness_small)

## Response formula with five ordinal items
form_items <- as.formula(
  paste(
    "cbind(",
    paste(colnames(tenseness_small)[1:5], collapse = ","),
    ") ~ 0"
  )
)

## Formula including covariates
form_cov <- as.formula(
  paste(
    "cbind(",
    paste(colnames(tenseness_small)[1:5], collapse = ","),
    ") ~ Gender + Age"
  )
)

## Not run:
## ------------------------------------------------------------
## Example 1: Classical DIF analysis as in Schauberger and Mair
## ------------------------------------------------------------
##
## A GPCM is fitted with item-specific covariate effects.
## Selected non-zero item-specific effects indicate DIF.

fit_dif <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    trace = FALSE
  )
)

fit_dif
summary(fit_dif)

## Plot DIF paths for selected item panels
plot(
  fit_dif,
  type = "DIF",
  panels = 1:2,
```

```
  panels_per_page = 2,
  columns = 2,
  ask_new = FALSE
)

## Predicted category probabilities and posterior trait estimates
pred_dif <- predict(fit_dif, type = "response")
theta_dif <- trait.posterior(fit_dif, cores = 1)


## ------------------------------------------------------------
## Example 2: Explanatory GPCM-X with global, item-specific,
## and fused covariate effects
## ------------------------------------------------------------
##
## This model includes global covariate effects and item-specific
## deviations from these global effects. The fusion penalty encourages
## clusters of items with equal covariate effects.
##
## Since information criteria are not uniquely defined for the fusion
## model, cross-validation is used for tuning parameter selection.

set.seed(1860)

fit_fusion <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  main.effects = TRUE,
  cv = TRUE,
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    folds = 2,
    penalize.main.effects = TRUE,
    include.item.effects = TRUE,
    fuse.per.variable = TRUE,
    scale.covariate.effects = FALSE,
    trace = FALSE
  )
)

fit_fusion

## Summary on the interpretable response-effect scale
summary(
  fit_fusion,
  effect.scale = "effect"
)

## Available variable panels
fit_fusion$design_list$x.names
```

```
## Plot covariate effect paths for all available covariate panels
plot(
  fit_fusion,
  type = "Variable",
  panels = fit_fusion$design_list$x.names,
  panels_per_page = 2,
  columns = 2,
  variable.scale = "effect",
  select = "cv",
  ask_new = FALSE
)

## End(Not run)
```

`plot.GPCMlasso` *Plot function for GPCMlasso*

Description

Plot function for a `GPCMlasso` object. Plots coefficient paths along the tuning parameter lambda.

Usage

```
## S3 method for class 'GPCMlasso'
plot(x, select = c("BIC", "AIC", "cAIC", "cv"),
type = c("DIF", "Variable"), variable.scale = c("effect", "internal"),
log.lambda = TRUE, panels_per_page = 1, panels = "all",
columns = NULL, ask_new = TRUE, lambda.lines = TRUE,
equal_range = TRUE, add.to.lambda = 0, items_per_page = NULL,
items = NULL, plots = NULL, ...)
```

Arguments

`x`	`GPCMlasso` object.
`select`	Specifies which criterion to use for the optimal model.
`type`	Which plot type should be used. `"DIF"` means that one plot panel per item is shown. `"Variable"` means that one plot panel per variable is shown.
`variable.scale`	Only relevant for `type = "Variable"`. `"effect"` plots interpretable response-effect paths. `"internal"` plots the internal GPCMlasso parameter paths.
`log.lambda`	A logical value indicating whether lambda or log(lambda) should be used.
`panels_per_page`	Number of plot panels per page.
`panels`	Panels to be shown. If `"all"`, all panels are shown. For `type = "DIF"`, `panels` refers to item panels. For `type = "Variable"`, `panels` refers to covariate panels. Panels can be specified by numeric indices or names.
`columns`	Number of columns when several plot panels are shown on one page.

ask_new	If TRUE, the user is asked before plotting the next page.
lambda.lines	Whether vertical lines should be drawn for all lambda values.
equal_range	Whether equal y-axis ranges should be used.
add.to.lambda	Constant added to lambda before log transformation.
items_per_page	Deprecated. Use panels_per_page instead.
items	Deprecated. Use panels instead.
plots	Deprecated. Use panels instead.
...	Further plot arguments.

Details

For type = "DIF", one plot panel per item is created and the internal item-specific parameters are shown. The argument panels can be used to restrict the output to selected item panels.

For type = "Variable", one plot panel per covariate is created. The argument panels can be used to restrict the output to selected covariate panels. By default, the paths are shown on the response-effect scale, i.e. global effect minus item-specific DIF parameter. This corresponds to the interpretable effect of a covariate on the local logit under the internal DIF parameterization.

For type = "Variable", effect-scale plots are recommended for interpretation.

---

predict.GPCMlasso *Predict function for GPCMlasso*

Description

Predict category probabilities or linear predictors from a fitted GPCMlasso model.

Usage

```
## S3 method for class 'GPCMlasso'
predict(
  object,
  coefs = NULL,
  newdata = NULL,
  theta = NULL,
  type = c("link", "response"),
  trait.cores = 1,
  ...
)
```

Arguments

object	GPCMlasso object.
coefs	Optional vector of coefficients, can be filled with a specific row from object$coefficients. If not specified, coefs are chosen to be the BIC-optimal coefficients or, if cross-validation was performed, the optimal coefficients according to cross-validation.
newdata	Optional data frame containing the covariates used for prediction. The right-hand side of object$formula is evaluated on newdata. The resulting columns must match the covariate columns used in the fitted model.
theta	Optional vector of trait values used for prediction. Required when newdata is supplied.
type	"link" returns the linear predictor. "response" returns predicted category probabilities.
trait.cores	Number of cores used for computing posterior trait estimates within prediction on the original data. The default is 1, which avoids the overhead of starting parallel workers for small prediction tasks.
...	Further predict arguments.

Details

Predictions are based on a selected coefficient vector. If no coefficient vector is supplied, the cross-validation optimal coefficient vector is used when cross-validation was performed; otherwise, the BIC-optimal coefficient vector is used.

For predictions on the original data, posterior trait estimates are obtained from trait.posterior(). For predictions on newdata, trait values must be supplied via the argument theta, because posterior trait estimation for new persons requires response data and is not inferred automatically from covariates alone.

Results are lists of vectors with length equal to the number of response categories in case of probabilities (type = "response") or the number of adjacent-category logits in case of linear predictors (type = "link").

Author(s)

Gunther Schauberger
<gunther@stat.uni-muenchen.de>

See Also

GPCMlasso

tenseness *Tenseness Data from the Freiburg Complaint Checklist*

Description

Data from the Freiburg Complaint Checklist. The data contain all 8 items corresponding to the scale *Tenseness* from the standardization sample of the Freiburg Complaint Checklist.

Format

A data frame containing data from the Freiburg Complaint Checklist. The item responses refer to the scale *Tenseness* and are measured on a 5-point Likert scale, where low values correspond to low frequencies or low intensities of the respective complaint and high values correspond to high frequencies or high intensities.

The data frame contains the following variables:

Clammy_hands Do you have clammy hands?

Sweat_attacks Do you have sudden attacks of sweating?

Clumsiness Do you notice that you behave clumsily?

Wavering_hands Do your hands shake frequently, for example when lighting a cigarette or when holding a cup?

Restless_hands Do you notice that your hands are restless?

Restless_feet Do you notice that your feet are restless?

Twitching_eyes Do you notice involuntary twitching of your eyes?

Twitching_mouth Do you notice involuntary twitching of your mouth?

Gender Gender of the person.

Household Whether the person lives alone or together with somebody else.

Income Income, categorized from 1 (low income) to 11 (high income). For simplicity, income can be treated as a metric variable.

WestEast Whether the person is from East Germany, the former GDR.

Abitur Whether the person has Abitur, corresponding to A-levels.

Age Age of the person.

Source

ZPID (2013). PsychData of the Leibniz Institute for Psychology Information ZPID. Trier: Center for Research Data in Psychology.

Fahrenberg, J. (2010). Freiburg Complaint Checklist [Freiburger Beschwerdenliste (FBL)]. Goettingen: Hogrefe.

Examples

```
data(tenseness)
str(tenseness)
```

tenseness_small *Subset of Tenseness Data from the Freiburg Complaint Checklist*

Description

A small subset of the Freiburg Complaint Checklist data. The data contain 5 items, out of the 8 items in the full data set, corresponding to the scale *Tenseness*, together with two covariates.

Format

A data frame containing a subset of the Freiburg Complaint Checklist data with 200 observations. The complete data set is available as `tenseness`. The item responses refer to the scale *Tenseness* and are measured on a 5-point Likert scale, where low values correspond to low frequencies or low intensities of the respective complaint and high values correspond to high frequencies or high intensities.

The data frame contains the following variables:

Clammy_hands Do you have clammy hands?

Sweat_attacks Do you have sudden attacks of sweating?

Clumsiness Do you notice that you behave clumsily?

Wavering_hands Do your hands shake frequently, for example when lighting a cigarette or when holding a cup?

Restless_hands Do you notice that your hands are restless?

Gender Gender of the person.

Age Age of the person.

Source

ZPID (2013). PsychData of the Leibniz Institute for Psychology Information ZPID. Trier: Center for Research Data in Psychology.

Fahrenberg, J. (2010). Freiburg Complaint Checklist [Freiburger Beschwerdenliste (FBL)]. Goettingen: Hogrefe.

See Also

`GPCMlasso`, `ctrl_GPCMlasso`, `plot.GPCMlasso`, `trait.posterior`

Examples

```
data(tenseness_small)

## Formula with five ordinal items and two covariates
form_cov <- as.formula(
  paste(
    "cbind(",
    paste(colnames(tenseness_small)[1:5], collapse = ","),
```

```
    ") ~ Gender + Age"
  )
)

## Not run:
## ------------------------------------------------------------
## Classical DIF analysis
## ------------------------------------------------------------
fit_dif <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    trace = FALSE
  )
)

summary(fit_dif)
plot(
  fit_dif,
  type = "DIF",
  panels_per_page = 2,
  columns = 2,
  ask_new = FALSE
)

## ------------------------------------------------------------
## Explanatory GPCM-X with global and item-specific effects
## and fusion penalty
## ------------------------------------------------------------
set.seed(1860)

fit_fusion <- GPCMlasso(
  formula = form_cov,
  data = tenseness_small,
  model = "GPCM",
  main.effects = TRUE,
  cv = TRUE,
  control = ctrl_GPCMlasso(
    cores = 1,
    l.lambda = 10,
    folds = 2,
    penalize.main.effects = TRUE,
    include.item.effects = TRUE,
    fuse.per.variable = TRUE,
    scale.covariate.effects = FALSE,
    trace = FALSE
  )
)

summary(
```

```
  fit_fusion,
  effect.scale = "effect"
)

## Available covariate panels
fit_fusion$design_list$x.names

plot(
  fit_fusion,
  type = "Variable",
  panels = fit_fusion$design_list$x.names,
  panels_per_page = 2,
  columns = 2,
  variable.scale = "effect",
  select = "cv",
  ask_new = FALSE
)

## End(Not run)
```

`trait.posterior` *Calculate Posterior Estimates for Trait Parameters*

Description

Calculates posterior estimates for trait/person parameters for a fitted `GPCMlasso` model using the assumed Gaussian distribution of the person parameters.

Usage

```
trait.posterior(model, coefs = NULL, cores = 25, tol = 1e-04)
```

Arguments

`model`	Object of class `GPCMlasso`.
`coefs`	Optional vector of coefficients. If `coefs = NULL`, the coefficients from the BIC-optimal model are used, or, if cross-validation was performed, the coefficients from the cross-validation optimal model are used.
`cores`	Number of cores used for parallel computation.
`tol`	Deprecated. Kept for backward compatibility.

Details

The function computes posterior means of the latent trait parameters by Gauss-Hermite quadrature. If no coefficient vector is supplied, the cross-validation optimal coefficient vector is used when cross-validation was performed; otherwise, the BIC-optimal coefficient vector is used.

Value

Numeric vector containing posterior estimates of the trait/person parameters.

Author(s)

Gunther Schauberger
<gunther.schauberger@tum.de>

See Also

GPCMlasso, predict.GPCMlasso

Examples

```
data(tenseness_small)

form0 <- as.formula(
  paste(
    "cbind(",
    paste(colnames(tenseness_small)[1:5], collapse = ","),
    ") ~ 0"
  )
)

## Not run:
rsm0 <- GPCMlasso(
  formula = form0,
  data = tenseness_small,
  model = "RSM",
  control = ctrl_GPCMlasso(cores = 1, trace = FALSE)
)

theta_hat <- trait.posterior(rsm0, cores = 1)
summary(theta_hat)

## End(Not run)
```

Index

∗ **DIF**
GPCMlasso-package, 2
∗ **GPCM**
GPCMlasso-package, 2
∗ **IRT**
GPCMlasso-package, 2
∗ **PCM**
GPCMlasso-package, 2
∗ **datasets**
tenseness, 14
tenseness_small, 15
∗ **fusion-lasso**
GPCMlasso-package, 2
∗ **package**
GPCMlasso-package, 2
∗ **variable-selection**
GPCMlasso-package, 2

ctrl_GPCMlasso, *2*, 4, *7*, *9*, *15*

gpcm, *6*
GPCMlasso, *2*, *6*, 7, *13*, *15*, *18*
GPCMlasso-package, 2

nlm, *6*

plot.GPCMlasso, *2*, *6*, *9*, 11, *15*
predict.GPCMlasso, *2*, *9*, 12, *18*

tenseness, 14, *15*
tenseness_small, 15
trait.posterior, *2*, *9*, *15*, 17