Package 'ipcr'

November 8, 2020

Type Package

Title Individual Parameter Contribution Regression

Version 0.2.0

Maintainer Manuel Arnold <arnoldmz@hu-berlin.de>

Description Identify and estimate parameter heterogeneity in structural equation models and other parametric models. This package is still under development. Please report any bugs.

Depends R (>= 3.1.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports ggplot2, glmnet, lavaan, lmtest, OpenMx, sandwich

RoxygenNote 7.1.1

R topics documented:

`get_ipcs` *Calcuate Individual Parameter Contributions*

Description

This functions calculates the individual parameter contributions (IPCs) of a fitted model.

Usage

```
get_ipcs(x)
```

Arguments

x	a fitted model object.

Details

`get_ipcs` is a convenience function for obtaining IPCs. The more powerful `ipcr` function also provides IPCs and can be used to predict differences in model parameters by regressing the IPCs on covariates.

Value

A `data.frame` containing the of a fitted model. The dimensions of this `data.frame` are $n * k$, where n denotes the number of observations and k the number of parameters. The columns should be named as in `coef`.

References

Arnold, M., Oberski, D. L., Brandmaier, A. M., & Voelkle, M. C. (2019). Identifying heterogeneity in dynamic panel models with individual parameter contribution regression. *Structural Equation Modeling, 27*, 613-628. doi: 10.1080/10705511.2019.1667240

See Also

`ipcr`

`get_scores` *Extract Scores*

Description

This functions extracts the scores of a fitted model. Arguments are passed to the generic function `estfun`.

Usage

```
get_scores(x, ...)
```

Arguments

x	a fitted model object.
...	arguments passed to methods.

Value

A `data.frame` containing the empirical estimating functions. Typically, this should be an $n * k$ matrix corresponding to n observations and k parameters. The columns should be named as in `coef` or `terms`, respectively.

The estimating function (or score function) for a model is the derivative of the objective function with respect to the parameter vector. The empirical estimating functions is the evaluation of the estimating function at the observed data (n observations) and the estimated parameters (of dimension k).

References

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. *Journal of Statistical Software, 16*(9), 1-16. doi: 10.18637/jss.v016.i09

Zeileis, A., Köll, S., Graham, N. (2020). Various versatile variances: An object- oriented implementation of clustered covariances in R. *Journal of Statistical Software, 95*(1), 1-36. doi 10.18637/jss.v095.i01

See Also

`estfun`

`ipcr` *Individual Parameter Contribution Regression*

Description

Explain and predict differences in model parameters with individual parameter contribution (IPC) regression. IPC regression allows studying heterogeneity in parameter estimates as a linear function of covariates. `ipcr` was mainly written for investigating parameter heterogeneity in structural equation models fitted with **lavaan** or **OpenMx** but also for models estimated with `lm` and `glm`.

Usage

```
ipcr(
  fit,
  covariates = NULL,
  iterate = FALSE,
  iteration_info = FALSE,
  conv = 1e-04,
  max_it = 50,
  regularization = FALSE,
  s = "lambda.min",
  alpha = 1,
  weights = NULL,
  nlambda = 100,
  standardize = TRUE,
  nfolds = 10,
  linear_MxModel = TRUE
)
```

Arguments

`fit`	a fitted model object.
`covariates`	a vector, matrix, or data.frame with one or more covariates used to predict differences in the model parameters. Interaction and polynomial terms can be included as new variables which may require centering.
`iterate`	a logical value; if TRUE iterated IPC regression is performed. Currently, iterated IPC regression is only available for models fitted with **lavaan** or **OpenMx**.
`iteration_info`	a logical value; if TRUE the parameter values for each iteration with corresponding log-likelihood value are stored in a matrix. Requesting this matrix increases the runtime.

conv	an integer used as a stopping criterion for iterated IPC regression. The criterion is the largest difference in any parameter estimate between iterations.
max_it	the maximum number of iterations for iterated IPC regressions.
regularization	a logical value; if TRUE regularized linear regression models are fitted via penalized maximum likelihood using k-fold cross-validation.
s	a character."lambda.min" (default) gives the minimum mean cross-validated error. The other option is lambda.1se", which gives the most regularized model such that the error is within one standard error of the minimum. For regularized IPC regression only.
alpha	The elastic net mixing parameter with $0 \leq \alpha \leq 1$. alpha = 1 is the lasso penalty (default) and alpha = 0 the rigde penalty. For regularized IPC regression only.
weights	observation weights for regularization. Can be total counts if responses are proportion matrices. Default is 1 for each observation. For regularized IPC regression only.
nlambda	the number of penalty terms. The default is 100. For regularized IPC regression only.
standardize	a logical value; if TRUE variables are standardized prior to regularization. This only affects regularization; standard/iterated IPC regression coefficients are not standardized.
nfolds	number of folds - default is 10. Although nfolds can be as large as the sample size (leave-one-out cross-validation), it is not recommended for large datasets. Smallest value allowable is 3. For regularized IPC regression only.
linear_MxModel	a logical value indicating if the structural equation model contains non-linear functions of model parameters (FALSE) or not (TRUE, default). TRUE speeds up the runtime of the linear model. Only relevant for models fitted with **OpenMx**.

Details

IPCs are rough approximations of individual-specific parameter values. The IPCs of individual i are defined as

$$IPC_i = \theta + A(\theta)^(-1)S(\theta, y_i),$$

where θ are the estimated model parameters, $S(\theta, y_i)$ is the estimating function (e.g, the first derivative of the log-likelihood), and $A(\theta)$ is the expectation of the negative derivative of the estimating function (often called Hessian matrix). By regressing IPCs on covariates, parameter differences can be predicted.

IPCs are often slightly biased. This bias can be removed with a procedure termed iterated IPC regression, which re-calculates the IPCs until the regression coefficients of the IPC regression models converge. Iterated IPCs often show larger variability than standard IPCs. `iterate=TRUE` performs iterated IPC regression. `regularization=TRUE` adds another list with regularized linear models to the `ipcr` object. All arguments related to regularization are passed to `cv.glmnet`. If requested, iterated IPCs will be used for the regularized IPC regression models.

Value

`ipcr` returns an object of `class` "`ipcr`". An `ipcr` function call returns a list which may consist of the following elements:

`info`	a list with information about the `ipcr` function call
`IPCs`	a `data.frame` with IPCs

regression_list	a list with an (iterated) IPC regression model for each model parameter
regularized_regression_list	a list with a regularized (iterated) IPC regression model for each model parameter
output	formatted output that can be examined with print and summary

The function summary prints a summary of the IPC regression equations. print shows the arguments specified in the ipcr function call. plot visualizes the correlation between IPCs and covariates in the form of a heatmap.

The generic functions AIC, BIC, coef, confint, effects, fitted, logLik, nobs, predict, residuals, sigma, and vcvo extract various information from the value returned by ipcr. Heteroskedastic robust IPC regression can be performed with the functions {coeftest} and coefci from **lmtest**. By default, these functions extract information for all model parameters. By specifying an additional parameter argument with the names of one or several of the model parameters as in coef, the return values can be limited to the corresponding model parameters.

References

Arnold, M., Oberski, D. L., Brandmaier, A. M., & Voelkle, M. C. (2019). Identifying heterogeneity in dynamic panel models with individual parameter contribution regression. *Structural Equation Modeling, 27*, 613-628. doi: 10.1080/10705511.2019.1667240

See Also

get_ipcs, plot.ipcr, and summary.ipcr

Examples

```
# Structural equation model example using the lavaan package

## Load Holzinger and Swineford (1939) data provided by the lavaan package
HS_data <- lavaan::HolzingerSwineford1939

## Remove observations with missing values
HS_data <- HS_data[stats::complete.cases(HS_data), ]

## lavaan model syntac for a single group model
m <- 'visual =~ x1 + x2 + x3
      textual =~ x4 + x5 + x6
      speed =~ x7 + x8 + x9'

## Fit the model
fit <- lavaan::cfa(model = m, data = HS_data)

## Prepare a data.frame with covariates
covariates <- HS_data[, c("sex", "ageyr", "agemo", "school", "grade")]

## Regress parameters on covariates with the ipcr function
res <- ipcr(fit = fit, covariates = covariates)

## Plot heatmap with the correlation between parameters and predictors
plot(res)

## Show results (standard IPC regression)
summary(res)

## IPC regression with LASSO regularization
```

```
res_reg <- ipcr(fit = fit, covariates = covariates, regularization = TRUE)

## Show results (regularized standard IPC regression)
summary(res_reg)
```

`plot.ipcr` *Plot Correlations between Covariates and IPCs*

Description

This functions plots a heat map that visualizes the correlation between covariates and IPCs.

Usage

```
## S3 method for class 'ipcr'
plot(x, ...)
```

Arguments

`x`	an ipcr object.
`...`	other arguments.

Details

This function is a wrapper for `ggplot`. Currently, arguments passed to `ggplot` cannot be changed.

`plot_differences` *Plot Estimated Conditional Differences in a Model Parameter*

Description

This functions plots the conditional values of the model parameters as a function of the covariates.

Usage

```
plot_differences(
  x,
  parameter = NULL,
  covariate = NULL,
  confidence_level = 0.95,
  ...
)
```

Arguments

x	an ipcr object.
parameter	a string. The name of a model parameter as in coef. Per default all model parameters are plotted
covariate	a string. The name of a covariate. Per default, all the effects of all covariates are plotted.
confidence_level	a numeric. The confidence interval plotted. 0.95 (resulting) in 95% confidence interval is the default.
...	other arguments.

Details

Note that regression lines (for continuous covariates), means (for dummy variables), and confidence intervals are plotted, using the estimates of the model parameters and the corresponding variances and covariances of the parameter estimates. The plots show the estimated parameter value as a function of a covariate with all other covariates set equal to zero. (Mean) centering of the covariates may increase the interpretability of the plots.

This function is a wrapper for ggplot.

See Also

plot.ipcr,

Examples

```
# Generate data
## Covariates
z1 <- rep(0:1, each = 50)
z2 <- rnorm(n = 100)
covariates <- data.frame(z1 = z1, z2 = z2)
## Model data
x <- rnorm(n = 100)
y <- 0.5 + 0.75*z1 + rnorm(n = 100, sd = sqrt(0.75))
d <- data.frame(x = x, y = y)

# Fit a linear regression
m <- lm(y ~ x, data = d)

# Investigate model with IPC regression
ipc <- ipcr(m, covariates = covariates)

# Plotting parameter values as a function of the covariates
## All plots
plot_differences(ipc)
## Plot the values of the regression slope x as a function of the covariate z1
plot_differences(ipc, parameter = "x", covariate = "z1")
```

summary.ipcr *Inidividual Parameter Contribution Regression Summary*

Description

This functions returns the coefficients of the individual parameter contribution (IPC) regression equations.

Usage

```
## S3 method for class 'ipcr'
summary(object, regularization = TRUE, digits = 3, verbose = FALSE, ...)
```

Arguments

object	an ipcr object.
regularization	a logical value; whether to show results for regularized IPC regression (if computed) or non-regularized results.
digits	integer indicating the number of decimal places to be used.
verbose	a logical value; if TRUE ipcr settings, non-regularized, and regularized results are shown.
...	further arguments passed to and from methods.

Index