Package ‘factorEx’

November 28, 2025

Type Package

Title Design and Analysis for Factorial Experiments

Version 1.1.0

Maintainer Naoki Egami <`naoki.egami5@gmail.com`>

Description Provides design-based and model-based estimators for the population average marginal component effects in general factorial experiments, including conjoint analysis. The package also implements a series of recommendations offered in de la Cuesta, Egami, and Imai (2022) <doi:10.1017/pan.2020.40>, and Egami and Imai (2019) <doi:10.1080/016214

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5.0), arm, genlasso

Imports prodlim, sandwich, igraph, pbmcapply, pbapply, mvtnorm, stringr, doParallel, foreach, estimatr

URL `https://github.com/naoki-egami/factorEx`

BugReports `https://github.com/naoki-egami/factorEx/issues`

NeedsCompilation no

Author Naoki Egami [aut, cre],
Brandon de la Cuesta [aut],
Kosuke Imai [aut]

Repository CRAN

Date/Publication 2025-11-28 09:50:07 UTC

Contents

`decompose_pAMCE` *Decompose the difference between the pAMCEs*

Description

See examples in ‘model_pAMCE‘.

Usage

```
decompose_pAMCE(out, effect_name, target_diff)
```

Arguments

`out`	An object of class "pAMCE", a result of a call to ’model_pAMCE’.
`effect_name`	Effect for which the function decomposes the difference in the pAMCEs. The first element should be a factor name and the second element should be a level name.
`target_diff`	Two target profile distributions for which the function compares the pAMCEs. If missing, the function comapres the first target profile distribution and the in-sample profile distribution.

Value

`decompose_pAMCE` returns ‘data.frame‘ showing the decomposition of the difference between the pAMCEs.

`design_pAMCE` *Estimating the population AMCE using a design-based approach*

Description

`design_pAMCE` implements the design-based approach to estimate the pAMCE. See de la Cuesta, Egami, and Imai (2022) for details. More examples are available at the GitHub page of `factorEx`.

Usage

```
design_pAMCE(
  formula,
  factor_name,
  data,
  pair = FALSE,
  pair_id = NULL,
  cross_int = FALSE,
  cluster_id = NULL,
  target_dist,
  target_type,
  partial_joint_name
)
```

Arguments

`formula`	Formula
`factor_name`	Factors for which the function estimates the pAMCEs. If not specified, the function estimates for all factors.
`data`	Data
`pair`	Whether we use a paired-choice conjoint design.
`pair_id`	Unique identifiers for pairs in the paired-choice conjoint design (optional).
`cross_int`	Include interactions across profiles. Default is FALSE.
`cluster_id`	Unique identifiers for computing cluster standard errors (optional).
`target_dist`	Target profile distributions to be used. See Examples in the GitHub page for details.
`target_type`	Types of target profile distributions. ‘marginal‘, ’partial_joint’, or ‘target_data‘.
`partial_joint_name`	(when ‘target_type = "partial_joint"‘) Names of factors representing partial joint distributions. See Examples in the GitHub page for details.

Value

`design_pAMCE` returns an object of `pAMCE` class.

- `AMCE`: Estimates of the pAMCE for all factors.
- `design_weight`: Weight for each observation used in the weighted difference-in-means.
- `approach`: "design_based"
- `input`: Input into the function.
- `...`: Values for internal use.

References

de la Cuesta, Egami, and Imai. (2022). Improving the External Validity of Conjoint Analysis: The Essential Role of Profile Distribution. Political Analysis.

Egami and Imai. (2019). Causal Interaction in Factorial Experiments: Application to Conjoint Analysis. Journal of the American Statistical Association, Vol.114, No.526 (June), pp. 529–540.

Examples

```
# Small example
data("OnoBurden")
OnoBurden_data_pr_s <- OnoBurden$OnoBurden_data_pr[1:500, ]
# randomization based on marginal population design
target_dist_marginal_s <- OnoBurden$target_dist_marginal[c("gender", "age")]

# design-based estimation
out_design_mar_s <-
  design_pAMCE(formula = Y ~ gender + age,
               factor_name = "gender",
               data = OnoBurden_data_pr_s,
               pair_id = OnoBurden_data_pr_s$pair_id,
               cluster_id = OnoBurden_data_pr_s$id,
               target_dist  = target_dist_marginal_s, target_type = "marginal")
summary(out_design_mar_s)

  #  Example
  data("OnoBurden")
  OnoBurden_data_pr <- OnoBurden$OnoBurden_data_pr
  # randomization based on marginal population design
  target_dist_marginal <- OnoBurden$target_dist_marginal

  # design-based estimation
  out_design_mar <-
   design_pAMCE(formula = Y ~ gender + age + family + race + experience + party + pos_security,
                 factor_name = c("gender", "age", "experience"),
                 data = OnoBurden_data_pr,
                 pair_id = OnoBurden_data_pr$pair_id,
                 cluster_id = OnoBurden_data_pr$id,
                 target_dist  = target_dist_marginal, target_type = "marginal")
 summary(out_design_mar)
```

`diagnose_pAMCE` *Diagnose modeling assumptions for the model-based approach*

Description

Diagnose modeling assumptions for the model-based approach

Usage

```
diagnose_pAMCE(x, factor_name)
```

Arguments

`x`	An object of class "pAMCE", a result of a call to 'model_pAMCE'.
`factor_name`	A factor for which the function diagnoses modeling assumptions.

Value

No return value, called for side effects

`model_pAMCE` *Estimating the population AMCE using a model-based approach*

Description

`model_pAMCE` implements the model-based approach to estimate the pAMCE. See de la Cuesta, Egami, and Imai (2022) for details. More examples are available at the GitHub page of `factorEx`.

Usage

```
model_pAMCE(
  formula,
  formula_three = NULL,
  data,
  reg = TRUE,
  ord_fac,
  pair = FALSE,
  pair_id = NULL,
  cross_int = FALSE,
  cluster_id = NULL,
  target_dist,
  target_type,
  difference = FALSE,
  cv_type = "cv.1Std",
  nfolds = 5,
  boot = 100,
  seed = 1234,
  numCores = NULL
)
```

Arguments

`formula`	Formula
`formula_three`	Formula for three-way interactions (optional)
`data`	Data
`reg`	TRUE (regularization) or FALSE (no regularization). Default is TRUE
`ord_fac`	Whether we assume each factor is ordered. When not specified, we assume all of them are ordered
`pair`	Whether we use a paired-choice conjoint design
`pair_id`	Unique identifiers for pairs in the paired-choice conjoint design (optional)
`cross_int`	Include interactions across profiles. Default is FALSE.

`cluster_id`	Unique identifiers for computing cluster standard errors (optional).
`target_dist`	Target profile distributions to be used. This argument should be ‘list‘
`target_type`	Types of target profile distributions. ‘marginal‘ or ‘target_data‘. See Examples for details.
`difference`	Whether we compute the differences between the multiple pAMCEs. Default is FALSE.
`cv_type`	(optimal only when ‘reg = TRUE“) ‘cv.1Std“ (stronger regularization; default) or ‘cv.min‘ (weaker regularization).
`nfolds`	Number of cross validation folds. Default is 5.
`boot`	The number of bootstrap samples.
`seed`	Seed for bootstrap.
`numCores`	Number of cores to be used for parallel computing. If not specified, detect the number of available cores internally.

Value

`model_pAMCE` returns an object of `pAMCE` class.

- `AMCE`: Estimates of the pAMCE for all factors.
- `boot_AMCE`: Estimates of the pAMCE for all factors in each bootstrap sample.
- `boot_coef`: Estimates of coefficients for the linear probability model in each bootstrap sample.
- `approach`: "model_based"
- `input`: Input into the function.
- `...`: Values for internal use.

References

de la Cuesta, Egami, and Imai. (2022). Improving the External Validity of Conjoint Analysis: The Essential Role of Profile Distribution. Political Analysis.

Egami and Imai. (2019). Causal Interaction in Factorial Experiments: Application to Conjoint Analysis. Journal of the American Statistical Association, Vol.114, No.526 (June), pp. 529–540.

Examples

```
# Small example
target_dist_marginal <- OnoBurden$target_dist_marginal
OnoBurden_data <- OnoBurden$OnoBurden_data
OnoBurden_data_small <- OnoBurden_data[1:300, ]
target_dist_marginal_small <- target_dist_marginal[c("gender", "race")]

# model-based estimation without regularization
out_model_s <-
    model_pAMCE(formula = Y ~ gender + race,
         data = OnoBurden_data_small, reg = FALSE,
         pair_id = OnoBurden_data_small$pair_id,
```

```
           cluster_id = OnoBurden_data_small$id,
           target_dist  = target_dist_marginal_small,
           target_type = "marginal")


 # model-based estimation with regularization
 out_model <-
  model_pAMCE(formula = Y ~ gender + age + family + race + experience + party + pos_security,
                 data = OnoBurden_data_small, reg = TRUE,
                 pair_id = OnoBurden_data_small$pair_id,
                 cluster_id = OnoBurden_data_small$id,
                 target_dist  = target_dist_marginal,
                 target_type = "marginal",
                 nfolds = 2,   # suggest "nfolds = 5" in practice
                 numCores = 1, # when numCores = NULL, it automatically uses all cores.
                 boot = 5)  # suggest "boot = 500" in practice
summary(out_model, factor_name = c("gender"))

# decompose the difference in the pAMCEs
decompose_pAMCE(out_model, effect_name = c("gender", "Female"))
```

`OnoBurden` *Dataset from Ono and Burden (2018)*

Description

Dataset from Ono and Burden (2018)

Usage

`OnoBurden`

Format

A list containing the conjoint data, target profile distributions based on (1) marginals, (2) a combination of marginal and partial joint distributions, and (3) the full joint distribution.

`plot.pAMCE` *Plotting the estimated population AMCEs*

Description

Plotting the estimated population AMCEs

Usage

```
## S3 method for class 'pAMCE'
plot(
  x,
  factor_name,
  target_dist_name,
  legend_pos = "topright",
  main = "Estimated population AMCEs",
  xlim,
  mar = 12,
  diagnose = FALSE,
  ...
)
```

Arguments

`x`	An object of class "pAMCE", a result of a call to 'model_pAMCE' or 'design_pAMCE'
`factor_name`	Factors for which the function visualizes the pAMCEs
`target_dist_name`	Names of the target profile distributions to be used
`legend_pos`	Position of the legend. Default is 'topright'
`main`	Title of the plot
`xlim`	Range for the x-axis
`mar`	Space on the left side of the plot. Default is 12
`diagnose`	Whether we plot diagnostic checks recommended in de la Cuesta, Egami, and Imai (2019). Default is FALSE
`...`	Other graphical parameters

Value

No return value, called for side effects

`plot_decompose` *Plot decomposition of the difference between pAMCEs*

Description

See examples in ‘model_pAMCE‘.

Usage

```
plot_decompose(x, effect_name, target_diff, mar = 12)
```

Arguments

`x`	An object of class "pAMCE", a result of a call to 'model_pAMCE'.
`effect_name`	Effect for which the function decomposes the difference in the pAMCEs. The first element should be a factor name and the second element should be a level name.
`target_diff`	Two target profile distributions for which the function compares the pAMCEs. If missing, the function comapres the first target profile distribution and the in-sample profile distribution.
`mar`	Space on the left side of the plot. Default is 12.

Value

No return value, called for side effects

`plot_diagnose` *Plotting diagnostic checks*

Description

Plotting diagnostic checks

Usage

```
plot_diagnose(
  x,
  factor_name,
  legend_pos = "topright",
  target_dist_name,
  xlim,
  mar
)
```

Arguments

`x`	An object of class "pAMCE", a result of a call to 'model_pAMCE.'
`factor_name`	A factor for which the function diagnoses modeling assumptions.
`legend_pos`	Position of the legend. Default is 'topright'.
`target_dist_name`	Names of the target profile distributions to be used.
`xlim`	Range for the x-axis.
`mar`	Space on the left side of the plot. Default is 12.

Value

No return value, called for side effects

`summary.pAMCE` *Summarizing the estimated population AMCEs*

Description

Summarizing the estimated population AMCEs

Usage

```
## S3 method for class 'pAMCE'
summary(object, factor_name, target_dist_name, sample = FALSE, ...)
```

Arguments

`object`	An object of class "pAMCE", a result of a call to 'model_pAMCE' or 'design_pAMCE'
`factor_name`	Factors for which the function visualizes the pAMCEs
`target_dist_name`	Names of the target profile distributions to be used
`sample`	Whether to print the sample AMCEs, which is estimated based on the profile distribution used for randomization
`...`	Other parameters

Value

No return value, called for side effects

`weights_pAMCE` *Computing weights for estimating the population AMCE using a design-based approach*

Description

Computing weights for estimating the population AMCE using a design-based approach

Usage

```
weights_pAMCE(
  formula,
  factor_name,
  data,
  pair,
  pair_id,
  cross_int,
```

```
  target_dist,
  target_type,
  partial_joint_name
)
```

Arguments

`formula`	Formula
`factor_name`	A factor for which the function computes weights
`data`	Data
`pair`	Whether we use a paired-choice conjoint design
`pair_id`	Unique identifiers for pairs in the paired-choice conjoint design (optional)
`cross_int`	Include interactions across profiles. Default is FALSE
`target_dist`	Target profile distributions to be used. This argument should be ‘list‘
`target_type`	Types of target profile distributions. ‘marginal‘, ‘partial_joint‘ or ‘target_data‘. See Examples for details
`partial_joint_name`	Names of factors representing partial joint distributions. See Examples for details.

Value

No return value, called for side effects

Index