Package ‘CRTgeeDR’

May 7, 2026

Title Doubly Robust Inverse Probability Weighted Augmented GEE Estimator

Version 2.0.1

Maintainer Melanie Prague <`mprague@hsph.harvard.edu`>

Description Implements a semi-parametric GEE estimator accounting for missing data with Inverse-probability weighting (IPW) and for imbalance in covariates with augmentation (AUG). The estimator IPW-AUG-GEE is Doubly robust (DR).

License GPL (>= 2)

Depends R (>= 2.10), Matrix, MASS, ggplot2, grDevices, graphics, stats, methods

LazyData true

NeedsCompilation no

Author Melanie Prague [aut, cre],
Paul Gilbert [ctb] (Author of R package numDeriv, which has been acknowledged in numDeriv.R),
Ravi Varadhan [ctb] (Author of R package numDeriv, which has been acknowledged in numDeriv.R),
Ming Wang [ctb] (Author of R package geesmv, which has been acknowledged in getFay.R),
Lee McDaniel [ctb] (Author of R package geeM, which has been modfied and references in multiple R files),
Nick Henderson [ctb] (Author of R package geeM, which has been modfied and references in multiple R files)

Repository CRAN

Date/Publication 2022-09-06 07:28:24 UTC

Contents

CRTgeeDR 2
data.sim 2
fitted.CRTgeeDR 3
geeDREstimation 3

getCI 8
getOMPlot 8
getPSPlot 9
predict.CRTgeeDR 9
print.CRTgeeDR 10
print.summary.CRTgeeDR 10
summary.CRTgeeDR 11

Index **12**

`CRTgeeDR` *Doubly Robust Inverse Probability Weighted Augmented GEE estimator*

Description

The CRTgeeDR package allows you to estimates parameters in a regression model (with possibly a link function). It allows treatment augmentation and IPW for missing data alone.

Details

The only function you're likely to need from **CRTgeeDR** is `geeDREstimation`. Otherwise refer to the help documentation.

`data.sim` *The data.sim Dataset.*

Description

HIV risk of infection after STI/HIV intervention in a cluster randomized trial.

Format

A data frame with 10000 rows and 8 variables

Details

A dataset containing the HIV risk scores and presence of risky behaviors (yes/no) and other covarites of 10000 subjects among 100 communities. The variables are as follows:

- IDPAT subject id
- CLUSTER cluster id
- TRT treatment status, 1 is received STI/HIV intervention
- X1 A covariate following a N(0,1)
- JOB employement status
- MARRIED marital status

- AGE age
- HIV.KNOW Score for HIV knowlege
- RELIGION religiosity score
- OUTCOME Binary outcome - 1 if the subject is at high risk of HIV infection, 0 otherwise. NA if missing.
- MISSING 1 if the ouctome is missing - 0 otherwise.

`fitted.CRTgeeDR` *Fit CRTgeeDR object.*

Description

Fit CRTgeeDR object to a dataset

Usage

```
## S3 method for class 'CRTgeeDR'
fitted(object, ...)
```

Arguments

`object`	CRTgeeDR object
`...`	ignored

`geeDREstimation` *Doubly Robust Inverse Probability Weighted Augmented GEE Estimator*

Description

This function implements a GEE estimator. It implements classical GEE, IPW-GEE, augmented GEE and IPW-Augmented GEE (Doubly robust).

Usage

```
geeDREstimation(formula, id, data = parent.frame(), family = gaussian,
  corstr = "independence", Mv = 1, weights = NULL, aug = NULL,
  pi.a = 1/2, corr.mat = NULL, init.beta = NULL, init.alpha = NULL,
  init.phi = 1, scale.fix = FALSE, sandwich = TRUE, maxit = 20,
  tol = 1e-05, print.log = FALSE, typeweights = "VW", nameTRT = "TRT",
  model.weights = NULL, model.augmentation.trt = NULL,
  model.augmentation.ctrl = NULL, stepwise.augmentation = FALSE,
  stepwise.weights = FALSE, nameMISS = "MISSING", nameY = "OUTCOME",
  sandwich.nuisance = FALSE, fay.adjustment = FALSE, fay.bound = 0.75)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`id`	a vector which identifies the clusters. The length of "id" should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
`data`	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which CRTgeeDR is called.
`family`	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See family for details of family functions.)
`corstr`	a character string specifying the correlation structure. The following are permitted: ’"independence"’, ’"exchangeable"’, ’"ar1"’, ’"unstructured"’ and ’"userdefined"’
`Mv`	for "m-dependent", the value for m
`weights`	A vector of weights for each observation. If an observation has weight 0, it is excluded from the calculations of any parameters. Observations with a NA anywhere (even in variables not included in the model) will be assigned a weight of 0.
`aug`	A list of vector (one for A=1 treated, one for A=0 control) for each observation representing E(Y|X,A=a).
`pi.a`	A number, Probability of treatment attribution P(A=1)
`corr.mat`	The correlation matrix for "fixed". Matrix should be symmetric with dimensions >= the maximum cluster size. If the correlation structure is "userdefined", then this is a matrix describing which correlations are the same.
`init.beta`	an optional vector with the initial values of beta. If not specified, then the intercept will be set to InvLink(mean(response)). init.beta must be specified if not using an intercept.
`init.alpha`	an optional scalar or vector giving the initial values for the correlation. If provided along with Mv>1 or unstructured correlation, then the user must ensure that the vector is of the appropriate length.
`init.phi`	an optional initial overdispersion parameter. If not supplied, initialized to 1.
`scale.fix`	if set to TRUE, then the scale parameter is fixed at the value of init.phi.
`sandwich`	if set to TRUE, the sandwich variance is provided together with the naive estimator of variance.
`maxit`	maximum number of iterations.
`tol`	tolerance in calculation of coefficients.
`print.log`	if set to TRUE, a report is printed.
`typeweights`	a character string specifying the weights implementation. The following are permitted: "GENMOD" for $W^{1/2}V^{-1}W^{1/2}$, "WV" for $V^{-1}W$

`nameTRT`	Name of the variable containing information for the treatment
`model.weights`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted for the propensity score. Must model the probability of being observed.
`model.augmentation.trt`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted for the ouctome model for the treated group (A=1).
`model.augmentation.ctrl`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted for the ouctome model for the control group (A=0).
`stepwise.augmentation`	if set to TRUE, a stepwise for the augmentation model is performed during the fit of the augmentation model for the OM
`stepwise.weights`	if set to TRUE, a stepwise for the propensity score is performed during the fit of the augmentation model for the OM
`nameMISS`	Name of the variable containing information for the Missing indicator
`nameY`	Name of the variable containing information for the outcome
`sandwich.nuisance`	if set to TRUE, the nuisance adjusted sandwich variance is provided.
`fay.adjustment`	if set to TRUE, the small-sample nuisance adjusted sandwich variance with Fay's adjustement is provided.
`fay.bound`	if set to 0.75 by default, bound value used for Fay's adjustement.

Details

The estimator is founds by solving:

$$0 = \sum_{i=1}^{M} \Bigg[\boldsymbol{D}_i^T \boldsymbol{V}_i^{-1} \boldsymbol{W}_i(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_W) \left(\boldsymbol{Y}_i - \boldsymbol{B}(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_B)\right)$$

$$+ \sum_{a=0,1} p^a (1-p)^{1-a} \boldsymbol{D}_i^T \boldsymbol{V}_i^{-1} \Big(\boldsymbol{B}(\boldsymbol{X}_i, A_i = a, \boldsymbol{\eta}_B) - \boldsymbol{\mu}_i(\boldsymbol{\beta}, A_i = a) \Big) \Bigg]$$

where $\boldsymbol{D}_i = \frac{\partial \boldsymbol{\mu}_i(\boldsymbol{\beta}, A_i)}{\partial \boldsymbol{\beta}^T}$ is the design matrix, $\boldsymbol{V}_i$ is the covariance matrix equal to $\boldsymbol{U}_i^{1/2} \boldsymbol{C}(\boldsymbol{\alpha}) \boldsymbol{U}_i^{1/2}$ with $\boldsymbol{U}_i$ a diagonal matrix with elements $\mathrm{var}(y_{ij})$ and $\boldsymbol{C}(\boldsymbol{\alpha})$ is the working correlation structure with non-diagonal terms $\boldsymbol{\alpha}$. Parameters $\boldsymbol{\alpha}$ are estimated using simple moment estimators from the Pearson residuals. The matrix of weights $\boldsymbol{W}_i(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_W) = diag\left[R_{ij}/\pi_{ij}(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_W)\right]_{j=1,\ldots,n_i}$, where $\pi_{ij}(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_W) = P(R_{ij}|\boldsymbol{X}_i, A_i)$ is the Propensity score (PS). The function $\boldsymbol{B}(\boldsymbol{X}_i, A_i = a, \boldsymbol{\eta}_B)$, which is called the Outcome Model (OM), is a function linking Y_{ij} with $\boldsymbol{X}_i$ and A_i. The $\boldsymbol{\eta}_B$ are nuisance parameters that are estimated. The estimator is most efficient if the OM is equal to $E(\boldsymbol{Y}_i|\boldsymbol{X}_i, A_i = a)$ The estimator denoted $\hat{\beta}_{aug}$ is found by solving the estimating equation. Although analytic solutions sometimes exist, coefficient estimates are generally obtained using an iterative procedure such as the Newton-Raphson method. Automatic implementation is such that, $\hat{\boldsymbol{\eta}}_W$ in $\boldsymbol{W}_i(\boldsymbol{X}_i, A_i, \hat{\boldsymbol{\eta}}_W)$ are obtained using a logistic regression and $\hat{\boldsymbol{\eta}}_B$ in $\boldsymbol{B}(\boldsymbol{X}_i, A_i, \hat{\boldsymbol{\eta}}_B)$ are obtained using a linear regression.

The variance of $\hat{\boldsymbol{\beta}}_{aug}$ is estimated by the sandwich variance estimator. There are two external sources of variability that need to be accounted for: estimation of $\boldsymbol{\eta}_W$ for the PS and of $\boldsymbol{\eta}_B$ for the OM. We denote $\boldsymbol{\Omega} = (\boldsymbol{\beta}, \boldsymbol{\eta}_W, \boldsymbol{\eta}_B)$ the estimated parameters of interest and nuisance parameters. We can stack estimating functions and score functions for $\boldsymbol{\Omega}$:

$$\boldsymbol{U}_i(\boldsymbol{\Omega}) = \begin{pmatrix} \boldsymbol{\Phi}_i(\boldsymbol{Y}_i, \boldsymbol{X}_i, A_i, \boldsymbol{\beta}, \boldsymbol{\eta}_W, \boldsymbol{\eta}_B) \\ \boldsymbol{S}_i^W(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_W) \\ \boldsymbol{S}_i^B(\boldsymbol{X}_i, A_i, \boldsymbol{\eta}_B) \end{pmatrix}$$

where $\boldsymbol{S}_i^W$ and $\boldsymbol{S}_i^B$ represent the score equations for patients in cluster i for the estimation of $\boldsymbol{\eta}_W$ and $\boldsymbol{\eta}_B$ in the PS and the OM. A standard Taylor expansion paired with Slutzky's theorem and the central limit theorem give the sandwich estimator adjusted for nuisance parameters estimation in the OM and PS:

$$Var(\boldsymbol{\Omega}) = E\left[\frac{\partial \boldsymbol{U}_i(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}}\right]^{-1^T} \underbrace{E\left[\boldsymbol{U}_i(\boldsymbol{\Omega})\boldsymbol{U}_i^T(\boldsymbol{\Omega})\right]}_{\boldsymbol{\Delta}_{adj}} \underbrace{E\left[\frac{\partial \boldsymbol{U}_i(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}}\right]^{-1}}_{\boldsymbol{\Gamma}_{adj}^{-1}}.$$

Value

An object of type 'CRTgeeDR'

$beta Final values for regressors estimates

- $phi scale parameter estimate
- $alpha Final values for association parameters in the working correlation structure when exchangeable
- $coefnames Name of the regressors in the main regression
- $niter Number of iteration done by the algorithm before convergence
- $converged convergence status
- $var.naiv Variance of the estimates model based (naive)
- $var Variance of the estimates sandwich
- $var.nuisance Variance of the estimates nuisance adjusted sandwich
- $var.fay Variance of the estimates nuisance adjusted sandwich with Fay correction for small samples
- $call Call function

- $corr Correlation structure used
- $clusz Number of unit in each cluster
- $FunList List of function associated with the family
- $X design matrix for the main regression
- $offset Offset specified in the regression
- $eta predicted values
- $weights Weights vector used in the diagonal term for the IPW
- $ps.model Summary of the regression fitted for the PS if computed internally
- $om.model.trt Summary of the regression fitted for the OM for treated if computed internally
- $om.model.ctrl Summary of the regression fitted for the OM for control if computed internally

Author(s)

Melanie Prague [based on R packages 'geeM' L. S. McDaniel, N. C. Henderson, and P. J. Rathouz. Fast Pure R Implementation of GEE: Application of the Matrix Package. The R Journal, 5(1):181-188, June 2013.]

References

Details regarding implementation can be found in

- 'Augmented GEE for improving efficiency and validity of estimation in cluster randomized trials by leveraging cluster-and individual-level covariates' - 2012 - Stephens A., Tchetgen Tchetgen E. and De Gruttola V. : Stat Med 31(10) - 915-930.
- 'Accounting for interactions and complex inter-subject dependency for estimating treatment effect in cluster randomized trials with missing at random outcomes' - 2015 - Prague M., Wang R., Stephens A., Tchetgen Tchetgen E. and De Gruttola V. : in revision.
- 'Fast Pure R Implementation of GEE: Application of the Matrix Package' - 2013 - McDaniel, Lee S and Henderson, Nicholas C and Rathouz, Paul J : The R Journal 5(1) - 181-197.
- 'Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators' - 2001 - Fay, Michael P and Graubard, Barry I : Biometrics 57(4) - 1198-1206.

Examples

```
data(data.sim)
 ## Not run:
 #### STANDARD GEE
 geeresults<-geeDREstimation(formula=OUTCOME~TRT,
                              id="CLUSTER" , data = data.sim,
                              family = "binomial", corstr = "independence")
 summary(geeresults)
 #### IPW GEE
 ipwresults<-geeDREstimation(formula=OUTCOME~TRT,
                              id="CLUSTER" , data = data.sim,
                              family = "binomial", corstr = "independence",
                              model.weights=I(MISSING==0)~TRT*AGE)
 summary(ipwresults)
```

```
 #### AUGMENTED GEE
 augresults<-geeDREstimation(formula=OUTCOME~TRT,
                                 id="CLUSTER" , data = data.sim,
                                 family = "binomial", corstr = "independence",
                                 model.augmentation.trt=OUTCOME~AGE,
                         model.augmentation.ctrl=OUTCOME~AGE, stepwise.augmentation=FALSE)
 summary(augresults)

## End(Not run)
 #### DOUBLY ROBUST
 drresults<-geeDREstimation(formula=OUTCOME~TRT,
                                 id="CLUSTER" , data = data.sim,
                                 family = "binomial", corstr = "independence",
                                 model.weights=I(MISSING==0)~TRT*AGE,
                                 model.augmentation.trt=OUTCOME~AGE,
                         model.augmentation.ctrl=OUTCOME~AGE, stepwise.augmentation=FALSE)
 summary(drresults)
```

`getCI` *Get Mean, Sd and CI for estimates from CRTgeeDR object.*

Description

Get the estimates, standard deviations and confidence intervals from an CRTgeeDR object associated with a regressor given in argument.

Usage

```
getCI(object, nameTRT = "TRT", quantile = 1.96)
```

Arguments

`object`	CRTgeeDR
`nameTRT`	character including the name of the variable of interest (often the treatment)
`quantile`	value of the normal quantile for the IC. default is 1.96 for 95%CI.

`getOMPlot` *Get the observed vs fitted residuals*

Description

Get the histogram and some basic statistics for the weights used in the IPW part.

Usage

```
getOMPlot(object, save = FALSE, name = "plotOM", typeplot = 0)
```

Arguments

`object`	CRTgeeDR
`save`	logical if TRUE the plot is saved as a pdf in the current directory
`name`	name of the plot saved as pdf
`typeplot`	integer indicating which is the adequation diagnostic plot for the PS. '0', all available in plot.glm are displayed, '1' Residuals vs Fitted, '2' Normal Q-Q, '3' Scale-Location, '4' Cook's distance, '5' Residuals vs Leverage and '6' Cook's dist vs Leverage* h[ii] / (1 - h[ii])

`getPSPlot` *Get the histogram of weights for IPW and adequation for the glm weights model*

Description

Get the histogram and some basic statistics for the weights used in the IPW part.

Usage

```
getPSPlot(object, save = FALSE, name = "plotPS", typeplot = NULL)
```

Arguments

`object`	CRTgeeDR
`save`	logical if TRUE the plot is saved as a pdf in the current directory
`name`	name of the plot saved as pdf
`typeplot`	integer indicating which is the adequation diagnostic plot for the PS. Default is NULL no output. '0', all available in plot.glm are displayed, '1' Residuals vs Fitted, '2' Normal Q-Q, '3' Scale-Location, '4' Cook's distance, '5' Residuals vs Leverage and '6' Cook's dist vs Leverage* h[ii] / (1 - h[ii])

`predict.CRTgeeDR` *Predict CRTgeeDR object.*

Description

Predict CRTgeeDR object to a dataset

Usage

```
## S3 method for class 'CRTgeeDR'
predict(object, newdata = NULL, ...)
```

Arguments

`object`	CRTgeeDR object
`newdata`	dataframe, new dataset to which the CRTgeeDRneed to be used for prediction
`...`	ignored

`print.CRTgeeDR` *Prints CRTgeeDR object.*

Description

Prints CRTgeeDR object

Usage

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

Arguments

`x`	CRTgeeDR x
`...`	ignored

`print.summary.CRTgeeDR` *Print the summarizing CRTgeeDR object.*

Description

Print Summary CRTgeeDR object

Usage

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

Arguments

`x`	summary.CRTgeeDR x
`...`	ignored

`summary.CRTgeeDR` *Summarizing CRTgeeDR object.*

Description

Summary CRTgeeDR object

Usage

```
## S3 method for class 'CRTgeeDR'
summary(object, ...)
```

Arguments

`object`	CRTgeeDR object
`...`	ignored

Index

CRTgeeDR, 2
CRTgeeDR-package *(*CRTgeeDR*)*, 2

data.sim, 2

fitted *(*fitted.CRTgeeDR*)*, 3
fitted.CRTgeeDR, 3

geeDREstimation, *2*, 3
getCI, 8
getOMPlot, 8
getPSPlot, 9

predict *(*predict.CRTgeeDR*)*, 9
predict.CRTgeeDR, 9
print *(*print.CRTgeeDR*)*, 10
print.CRTgeeDR, 10
print.summary *(*print.summary.CRTgeeDR*)*, 10
print.summary.CRTgeeDR, 10

summary *(*summary.CRTgeeDR*)*, 11
summary.CRTgeeDR, 11