Package 'COMMA'

October 30, 2024

Title Correcting Misclassified Mediation Analysis

Version 1.1.0

Author Kimberly Webb [aut, cre]

Maintainer Kimberly Webb <kah343@cornell.edu>

Description Use three methods to estimate parameters from a mediation analysis with a binary misclassified mediator. These methods correct for the problem of ``label switching'' using Youden's J criteria. A detailed description of the analysis methods is available in Webb and Wells (2024), ``Effect estimation in the presence of a misclassified binary mediator'' <doi:10.48550/arXiv.2407.06970>.

Depends R (>= 4.2.0)

Imports Matrix (> 1.4-1), turboEM (>= 2021.1), dplyr (>= 1.1.4), foreach (>= 1.5.2), parallel (>= 4.3.1), doParallel (>= 1.0.17)

Suggests knitr (>= 1.40), kableExtra (>= 1.3.4), ggplot2 (>= 3.5.0), markdown (>= 1.13), stats (>= 4.3.1), svglite (>= 2.1.3)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

Collate 'sum_every_n1.R' 'sum_every_n.R' 'pistar_compute.R' 'pi_compute.R' 'COMBO_weight.R' 'COMBO_EM_function.R' 'COMBO_EM_algorithm.R' 'COMMA_data.R' 'w_m_normalY.R' 'w_m_binaryY.R' 'EM_function_normalY_XM.R' 'EM_function_normalY.R' 'EM_function_bernoulliY_XM.R' 'EM_function_bernoulliY.R' 'COMMA_EM.R' 'COMMA_boot_sample.R' 'COMMA_EM_bootstrap_SE.R' 'COMMA_OLS.R' 'COMMA_OLS_bootstrap_SE.R' 'COMMA_PVW.R' 'COMMA_PVW_bootstrap_SE.R' 'EM_function_poissonY.R' 'EM_function_poissonY_XM.R' 'NCHS2022_sample.R' 'misclassification_prob.R' 'theta_optim.R' 'theta_optim_XM.R' 'true_classification_prob.R' 'w_m_poissonY.R'

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2024-10-30 19:30:06 UTC

Contents

`COMBO_EM_algorithm` *EM-Algorithm Estimation of the Binary Outcome Misclassification Model*

Description

Jointly estimate β and γ parameters from the true outcome and observation mechanisms, respectively, in a binary outcome misclassification model.

Usage

```
COMBO_EM_algorithm(
  Ystar,
  x_matrix,
  z_matrix,
  beta_start,
  gamma_start,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

Ystar
: A numeric vector of indicator variables (1, 2) for the observed outcome `Y*`. There should be no `NA` terms. The reference category is 2.

x_matrix
: A numeric matrix of covariates in the true outcome mechanism. `x_matrix` should not contain an intercept and no values should be `NA`.

z_matrix
: A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept and no values should be `NA`.

beta_start
: A numeric vector or column matrix of starting values for the β parameters in the true outcome mechanism. The number of elements in `beta_start` should be equal to the number of columns of `x_matrix` plus 1.

gamma_start
: A numeric vector or matrix of starting values for the γ parameters in the observation mechanism. In matrix form, the `gamma_start` matrix rows correspond to parameters for the `Y* = 1` observed outcome, with the dimensions of `z_matrix` plus 1, and the gamma parameter matrix columns correspond to the true outcome categories $M \in \{1, 2\}$. A numeric vector for `gamma_start` is obtained by concatenating the gamma matrix, i.e. `gamma_start <- c(gamma_matrix)`.

tolerance
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

max_em_iterations
: An integer specifying the maximum number of iterations of the EM algorithm. The default is `1500`.

em_method
: A character string specifying which EM algorithm will be applied. Options are `"em"`, `"squarem"`, or `"pem"`. The default and recommended option is `"squarem"`.

Value

`COMBO_EM_algorithm` returns a data frame containing four columns. The first column, `Parameter`, represents a unique parameter value for each row. The next column contains the parameter `Estimates`, followed by the standard error estimates, `SE`. The final column, `Convergence`, reports whether or not the algorithm converged for a given parameter estimate.

COMBO_EM_function *EM-Algorithm Function for Estimation of the Misclassification Model*

Description

EM-Algorithm Function for Estimation of the Misclassification Model

Usage

```
COMBO_EM_function(param_current, obs_Y_matrix, X, Z, sample_size, n_cat)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the `Y* = 1` observed outcome, with the dimensions of `Z`. In matrix form, the gamma parameter matrix columns correspond to the true outcome categories $j = 1, \dots,$ `n_cat`. The numeric vector `gamma_v` is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`.

`obs_Y_matrix` A numeric matrix of indicator variables (0, 1) for the observed outcome `Y*`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`X` A numeric design matrix for the true outcome mechanism.

`Z` A numeric design matrix for the observation mechanism.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.

`n_cat` The number of categorical values that the true outcome, `Y`, and the observed outcome, `Y*` can take.

Value

`COMBO_EM_function` returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

COMBO_weight *Compute E-step for Binary Outcome Misclassification Model Estimated With the EM-Algorithm*

Description

Compute E-step for Binary Outcome Misclassification Model Estimated With the EM-Algorithm

Usage

```
COMBO_weight(ystar_matrix, pistar_matrix, pi_matrix, sample_size, n_cat)
```

Arguments

`ystar_matrix` A numeric matrix of indicator variables (0, 1) for the observed outcome `Y*`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`pistar_matrix` A numeric matrix of conditional probabilities obtained from the internal function `pistar_compute`. Rows of the matrix correspond to each subject and to each observed outcome category. Columns of the matrix correspond to each true, latent outcome category.

`pi_matrix` A numeric matrix of probabilities obtained from the internal function `pi_compute`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each true, latent outcome category.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the observed outcome matrix, `ystar_matrix`.

`n_cat` The number of categorical values that the true outcome, `Y`, and the observed outcome, `Y*`, can take.

Value

`COMBO_weight` returns a matrix of E-step weights for the EM-algorithm, computed as follows: $\sum_{k=1}^{2} \frac{y^*_{ik} \pi^*_{ikj} \pi_{ij}}{\sum_{\ell=1}^{2} \pi^*_{ik\ell} \pi_{i\ell}}$. Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true outcome categories $j = 1, \dots,$ `n_cat`.

`COMMA_boot_sample` *Generate Bootstrap Samples for Estimating Standard Errors*

Description

Generate Bootstrap Samples for Estimating Standard Errors

Usage

```
COMMA_boot_sample(
  parameter_estimates,
  sigma_estimate = 1,
  outcome_distribution,
  interaction_indicator,
  x_matrix,
  z_matrix,
  c_matrix
)
```

Arguments

`parameter_estimates`
A column matrix of β, γ, and θ parameter values obtained from a COMMA analysis function. Parameter estimates should be supplied in the following order: 1) β (intercept, slope), 2) γ (intercept and slope from the M = 1 mechanism, intercept and slope from the M = 2 mechanism), and 3) θ (intercept, slope, coefficient for `x`, slope coefficient for `m`, slope coefficient for `c`, and, optionally, slope coefficient for `xm` if using).

`sigma_estimate` A numeric value specifying the estimated standard deviation. This value is only required if `outcome_distribution` is `"Normal"`. Default is 1. For non-Normal outcome distributions, the value should be `NULL`.

`outcome_distribution`
A character string specifying the distribution of the outcome variable. Options are `"Bernoulli"`, `"Normal"`, or `"Poisson"`.

`interaction_indicator`
A logical value indicating if an interaction between `x` and `m` should be used to generate the outcome variable, `y`.

`x_matrix` A numeric matrix of predictors in the true mediator and outcome mechanisms. `x_matrix` should not contain an intercept and no values should be `NA`.

`z_matrix` A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept and no values should be `NA`.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

Value

`COMMA_boot_sample` returns a list with the bootstrap sample data:

`obs_mediator` A vector of observed mediator values.

`true_mediator` A vector of true mediator values.

`outcome` A vector of outcome values.

`x_matrix` A matrix of predictor values in the true mediator mechanism. Identical to that supplied by the user.

`z_matrix` A matrix of predictor values in the observed mediator mechanism. Identical to that supplied by the user.

`c_matrix` A matrix of covariates. Identical to that supplied by the user.

`COMMA_data` *Generate Data to use in COMMA Functions*

Description

Generate Data to use in COMMA Functions

Usage

```
COMMA_data(
  sample_size,
  x_mu,
  x_sigma,
  z_shape,
  c_shape,
  interaction_indicator,
  outcome_distribution,
  true_beta,
  true_gamma,
  true_theta
)
```

Arguments

`sample_size` An integer specifying the sample size of the generated data set.

`x_mu` A numeric value specifying the mean of `x` predictors generated from a Normal distribution.

`x_sigma` A positive numeric value specifying the standard deviation of `x` predictors generated from a Normal distribution.

`z_shape` A positive numeric value specifying the shape parameter of `z` predictors generated from a Gamma distribution.

`c_shape` A positive numeric value specifying the shape parameter of `c` covariates generated from a Gamma distribution.

`interaction_indicator` A logical value indicating if an interaction between `x` and `m` should be used to generate the outcome variable, `y`.

`outcome_distribution` A character string specifying the distribution of the outcome variable. Options are `"Bernoulli"`, `"Normal"`, or `"Poisson"`.

`true_beta` A column matrix of β parameter values (intercept, slope) to generate data under in the true mediator mechanism.

`true_gamma` A numeric matrix of γ parameters to generate data in the observed mediator mechanisms. In matrix form, the `gamma` matrix rows correspond to intercept (row 1) and slope (row 2) terms. The gamma parameter matrix columns correspond to the true mediator categories $M \in \{1, 2\}$.

`true_theta` A column matrix of θ parameter values (intercept, slope coefficient for `x`, slope coefficient for `m`, slope coefficient for `c`, and, optionally, slope coefficient for `xm` if using) to generate data in the outcome mechanism.

Value

`COMMA_data` returns a list of generated data elements:

`obs_mediator` A vector of observed mediator values.

`true_mediator` A vector of true mediator values.

`outcome` A vector of outcome values.

`x` A vector of generated predictor values in the true mediator mechanism, from the Normal distribution.

`z` A vector of generated predictor values in the observed mediator mechanism from the Gamma distribution.

`c` A vector of generated covariates.

`x_design_matrix` The design matrix for the `x` predictor.

`z_design_matrix` The design matrix for the `z` predictor.

`c_design_matrix` The design matrix for the `c` predictor.

Examples

```
set.seed(20240709)
sample_size <- 10000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, -.2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Bernoulli",
                           true_beta, true_gamma, true_theta)

head(example_data$obs_mediator)
head(example_data$true_mediator)
```

`COMMA_EM` *EM Algorithm Estimation of the Binary Mediator Misclassification Model*

Description

Jointly estimate β, γ, and θ parameters from the true mediator, observed mediator, and outcome mechanisms, respectively, in a binary mediator misclassification model.

Usage

```
COMMA_EM(
  Mstar,
  outcome,
  outcome_distribution,
  interaction_indicator,
  x_matrix,
  z_matrix,
  c_matrix,
  beta_start,
  gamma_start,
  theta_start,
  sigma_start = NULL,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

Mstar — A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

outcome — A vector containing the outcome variables of interest. There should be no NA terms.

outcome_distribution — A character string specifying the distribution of the outcome variable. Options are "Bernoulli", "Normal", or "Poisson".

interaction_indicator — A logical value indicating if an interaction between x and m should be used to generate the outcome variable, y.

x_matrix — A numeric matrix of predictors in the true mediator and outcome mechanisms. x_matrix should not contain an intercept and no values should be NA.

z_matrix — A numeric matrix of covariates in the observation mechanism. z_matrix should not contain an intercept and no values should be NA.

c_matrix — A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

beta_start — A numeric vector or column matrix of starting values for the β parameters in the true mediator mechanism. The number of elements in beta_start should be equal to the number of columns of x_matrix and c_matrix plus 1. Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix.

gamma_start — A numeric vector or matrix of starting values for the γ parameters in the observation mechanism. In matrix form, the gamma_start matrix rows correspond to parameters for the M* = 1 observed mediator, with the dimensions of z_matrix plus 1, and the gamma parameter matrix columns correspond to the true mediator categories $M \in \{1, 2\}$. A numeric vector for gamma_start is obtained by concatenating the gamma matrix, i.e. gamma_start <- c(gamma_matrix). Starting values should be provided in the following order within each column: intercept, slope coefficient for first column of the z_matrix, ..., slope coefficient for the final column of the z_matrix.

theta_start
: A numeric vector or column matrix of starting values for the θ parameters in the outcome mechanism. The number of elements in theta_start should be equal to the number of columns of x_matrix and c_matrix plus 2 (if interaction_indicator is FALSE) or 3 (if interaction_indicator is TRUE). Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for the mediator m term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix, and, optionally, slope coefficient for xm).

sigma_start
: A numeric value specifying the starting value for the standard deviation. This value is only required if outcome_distribution is "Normal". Otherwise, this value is set to NULL.

tolerance
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

max_em_iterations
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

em_method
: A character string specifying which EM algorithm will be applied. Options are "em", "squarem", or "pem". The default and recommended option is "squarem".

Value

COMMA_EM returns a data frame containing four columns. The first column, Parameter, represents a unique parameter value for each row. The next column contains the parameter Estimates, followed by the standard error estimates, SE. The final column, Convergence, reports whether or not the algorithm converged for a given parameter estimate.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, -.2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Bernoulli",
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
theta_start <- matrix(rep(1, 4), ncol = 1)

Mstar = example_data[["obs_mediator"]]
outcome = example_data[["outcome"]]
x_matrix = example_data[["x"]]
z_matrix = example_data[["z"]]
c_matrix = example_data[["c"]]

EM_results <- COMMA_EM(Mstar, outcome, "Bernoulli", FALSE,
                       x_matrix, z_matrix, c_matrix,
                       beta_start, gamma_start, theta_start)

EM_results
```

COMMA_EM_bootstrap_SE *Estimate Bootstrap Standard Errors using EM*

Description

Estimate Bootstrap Standard Errors using EM

Usage

```
COMMA_EM_bootstrap_SE(
  parameter_estimates,
  sigma_estimate = 1,
  n_bootstrap,
  n_parallel,
  outcome_distribution,
  interaction_indicator,
  x_matrix,
  z_matrix,
  c_matrix,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

`parameter_estimates`
: A column matrix of β, γ, and θ parameter values obtained from a COMMA analysis function. Parameter estimates should be supplied in the following order: 1) β (intercept, slope), 2) γ (intercept and slope from the M = 1 mechanism, intercept and slope from the M = 2 mechanism), and 3) θ (intercept, slope, coefficient for `x`, slope coefficient for `m`, slope coefficient for `c`, and, optionally, slope coefficient for `xm` if using).

`sigma_estimate` A numeric value specifying the estimated standard deviation. This value is only required if `outcome_distribution` is `"Normal"`. Default is 1. For non-Normal outcome distributions, the value should be `NULL`.

`n_bootstrap` A numeric value specifying the number of bootstrap samples to draw.

`n_parallel` A numeric value specifying the number of parallel cores to run the computation on.

`outcome_distribution`
: A character string specifying the distribution of the outcome variable. Options are `"Bernoulli"`, `"Normal"`, or `"Poisson"`.

`interaction_indicator`
: A logical value indicating if an interaction between `x` and `m` should be used to generate the outcome variable, `y`.

`x_matrix` A numeric matrix of predictors in the true mediator and outcome mechanisms. `x_matrix` should not contain an intercept and no values should be `NA`.

`z_matrix` A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept and no values should be `NA`.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`tolerance` A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`max_em_iterations`
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`em_method` A character string specifying which EM algorithm will be applied. Options are `"em"`, `"squarem"`, or `"pem"`. The default and recommended option is `"squarem"`.

Value

`COMMA_EM_bootstrap_SE` returns a list with two elements: 1) `bootstrap_df` and 2) `bootstrap_SE`. `bootstrap_df` is a data frame containing `COMMA_EM` output for each bootstrap sample. `bootstrap_SE` is a data frame containing bootstrap standard error estimates for each parameter.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator
```

```
# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, -.2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = ”Bernoulli”,
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
theta_start <- matrix(rep(1, 4), ncol = 1)

Mstar = example_data[[”obs_mediator”]]
outcome = example_data[[”outcome”]]
x_matrix = example_data[[”x”]]
z_matrix = example_data[[”z”]]
c_matrix = example_data[[”c”]]

EM_results <- COMMA_EM(Mstar, outcome, ”Bernoulli”, FALSE,
                       x_matrix, z_matrix, c_matrix,
                       beta_start, gamma_start, theta_start)

EM_results

EM_SEs <- COMMA_EM_bootstrap_SE(EM_results$Estimates, sigma_estimate = NULL,
                                n_bootstrap = 3,
                                n_parallel = 1,
                                outcome_distribution = ”Bernoulli”,
                                interaction_indicator = FALSE,
                                x_matrix, z_matrix, c_matrix)

EM_SEs$bootstrap_SE
```

COMMA_OLS *Ordinary Least Squares Estimation of the Binary Mediator Misclassification Model*

Description

Estimate β, γ, and θ parameters from the true mediator, observed mediator, and outcome mechanisms, respectively, in a binary mediator misclassification model using an ordinary least squares correction.

Usage

```
COMMA_OLS(
  Mstar,
  outcome,
  x_matrix,
  z_matrix,
  c_matrix,
  beta_start,
  gamma_start,
  theta_start,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

Mstar A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

outcome A vector containing the outcome variables of interest. There should be no NA terms.

x_matrix A numeric matrix of predictors in the true mediator and outcome mechanisms. x_matrix should not contain an intercept and no values should be NA.

z_matrix A numeric matrix of covariates in the observation mechanism. z_matrix should not contain an intercept and no values should be NA.

c_matrix A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

beta_start A numeric vector or column matrix of starting values for the β parameters in the true mediator mechanism. The number of elements in beta_start should be equal to the number of columns of x_matrix and c_matrix plus 1. Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix.

gamma_start A numeric vector or matrix of starting values for the γ parameters in the observation mechanism. In matrix form, the gamma_start matrix rows correspond to parameters for the M* = 1 observed mediator, with the dimensions of z_matrix plus 1, and the gamma parameter matrix columns correspond to the true mediator categories $M \in \{1, 2\}$. A numeric vector for gamma_start is obtained by concatenating the gamma matrix, i.e. gamma_start <- c(gamma_matrix). Starting values should be provided in the following order within each column: intercept, slope coefficient for first column of the z_matrix, ..., slope coefficient for the final column of the z_matrix.

theta_start A numeric vector or column matrix of starting values for the θ parameters in the outcome mechanism. The number of elements in theta_start should be equal to the number of columns of x_matrix and c_matrix plus 2. Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for the mediator m term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix.

tolerance A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

max_em_iterations

A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

em_method A character string specifying which EM algorithm will be applied. Options are "em", "squarem", or "pem". The default and recommended option is "squarem".

Details

Note that this method can only be used for Normal outcome models, and interaction terms (between x and m) are not supported.

Value

COMMA_PVW returns a data frame containing four columns. The first column, Parameter, represents a unique parameter value for each row. The next column contains the parameter Estimates. The third column, Convergence, reports whether or not the algorithm converged for a given parameter estimate. The final column, Method, reports that the estimates are obtained from the "PVW" procedure.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, 2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Normal",
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
```

```
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
theta_start <- matrix(rep(1, 4), ncol = 1)

Mstar = example_data[["obs_mediator"]]
outcome = example_data[["outcome"]]
x_matrix = example_data[["x"]]
z_matrix = example_data[["z"]]
c_matrix = example_data[["c"]]

OLS_results <- COMMA_OLS(Mstar, outcome,
                         x_matrix, z_matrix, c_matrix,
                         beta_start, gamma_start, theta_start)

OLS_results
```

COMMA_OLS_bootstrap_SE

Estimate Bootstrap Standard Errors using OLS

Description

Estimate Bootstrap Standard Errors using OLS

Usage

```
COMMA_OLS_bootstrap_SE(
  parameter_estimates,
  sigma_estimate = 1,
  n_bootstrap,
  n_parallel,
  x_matrix,
  z_matrix,
  c_matrix,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

`parameter_estimates`

A column matrix of β, γ, and θ parameter values obtained from a COMMA analysis function. Parameter estimates should be supplied in the following order: 1) β (intercept, slope), 2) γ (intercept and slope from the M = 1 mechanism, intercept and slope from the M = 2 mechanism), and 3) θ (intercept, slope, coefficient for `x`, slope coefficient for `m`, slope coefficient for `c`, and, optionally, slope coefficient for `xm` if using).

`sigma_estimate` A numeric value specifying the estimated standard deviation. Default is 1.

`n_bootstrap` A numeric value specifying the number of bootstrap samples to draw.

`n_parallel` A numeric value specifying the number of parallel cores to run the computation on.

`x_matrix` A numeric matrix of predictors in the true mediator and outcome mechanisms. `x_matrix` should not contain an intercept and no values should be `NA`.

`z_matrix` A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept and no values should be `NA`.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`tolerance` A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`max_em_iterations`
A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`em_method` A character string specifying which EM algorithm will be applied. Options are `"em"`, `"squarem"`, or `"pem"`. The default and recommended option is `"squarem"`.

Value

`COMMA_OLS_bootstrap_SE` returns a list with two elements: 1) `bootstrap_df` and 2) `bootstrap_SE`. `bootstrap_df` is a data frame containing `COMMA_OLS` output for each bootstrap sample. `bootstrap_SE` is a data frame containing bootstrap standard error estimates for each parameter.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, 2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Normal",
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
```

```
theta_start <- matrix(rep(1, 4), ncol = 1)

Mstar = example_data[["obs_mediator"]]
outcome = example_data[["outcome"]]
x_matrix = example_data[["x"]]
z_matrix = example_data[["z"]]
c_matrix = example_data[["c"]]

OLS_results <- COMMA_OLS(Mstar, outcome,
                         x_matrix, z_matrix, c_matrix,
                         beta_start, gamma_start, theta_start)

OLS_results

OLS_SEs <- COMMA_OLS_bootstrap_SE(OLS_results$Estimates, sigma_estimate = 1,
                                  n_bootstrap = 3,
                                  n_parallel = 1,
                                  x_matrix, z_matrix, c_matrix)

OLS_SEs$bootstrap_SE
```

COMMA_PVW *Predictive Value Weighting Estimation of the Binary Mediator Misclassification Model*

Description

Estimate β, γ, and θ parameters from the true mediator, observed mediator, and outcome mechanisms, respectively, in a binary mediator misclassification model using a predictive value weighting approach.

Usage

```
COMMA_PVW(
  Mstar,
  outcome,
  outcome_distribution,
  interaction_indicator,
  x_matrix,
  z_matrix,
  c_matrix,
  beta_start,
  gamma_start,
  theta_start,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

Mstar
: A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

outcome
: A vector containing the outcome variables of interest. There should be no NA terms.

outcome_distribution
: A character string specifying the distribution of the outcome variable. Options are "Bernoulli", "Poisson", or "Normal".

interaction_indicator
: A logical value indicating if an interaction between x and m should be used to generate the outcome variable, y.

x_matrix
: A numeric matrix of predictors in the true mediator and outcome mechanisms. x_matrix should not contain an intercept and no values should be NA.

z_matrix
: A numeric matrix of covariates in the observation mechanism. z_matrix should not contain an intercept and no values should be NA.

c_matrix
: A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

beta_start
: A numeric vector or column matrix of starting values for the β parameters in the true mediator mechanism. The number of elements in beta_start should be equal to the number of columns of x_matrix and c_matrix plus 1. Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix.

gamma_start
: A numeric vector or matrix of starting values for the γ parameters in the observation mechanism. In matrix form, the gamma_start matrix rows correspond to parameters for the M* = 1 observed mediator, with the dimensions of z_matrix plus 1, and the gamma parameter matrix columns correspond to the true mediator categories $M \in \{1, 2\}$. A numeric vector for gamma_start is obtained by concatenating the gamma matrix, i.e. gamma_start <- c(gamma_matrix). Starting values should be provided in the following order within each column: intercept, slope coefficient for first column of the z_matrix, ..., slope coefficient for the final column of the z_matrix.

theta_start
: A numeric vector or column matrix of starting values for the θ parameters in the outcome mechanism. The number of elements in theta_start should be equal to the number of columns of x_matrix and c_matrix plus 2 (if interaction_indicator is FALSE) or 3 (if interaction_indicator is TRUE). Starting values should be provided in the following order: intercept, slope coefficient for the x_matrix term, slope coefficient for the mediator m term, slope coefficient for first column of the c_matrix, ..., slope coefficient for the final column of the c_matrix, and, optionally, slope coefficient for xm).

tolerance
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

max_em_iterations
: A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is 1e-7.

`em_method` A character string specifying which EM algorithm will be applied. Options are `"em"`, `"squarem"`, or `"pem"`. The default and recommended option is `"squarem"`.

Details

Note that this method can only be used for binary outcome models.

Value

`COMMA_PVW` returns a data frame containing four columns. The first column, `Parameter`, represents a unique parameter value for each row. The next column contains the parameter `Estimates`. The third column, `Convergence`, reports whether or not the algorithm converged for a given parameter estimate. The final column, `Method`, reports that the estimates are obtained from the "PVW" procedure.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, -.2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Bernoulli",
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
theta_start <- matrix(rep(1, 4), ncol = 1)

Mstar = example_data[["obs_mediator"]]
outcome = example_data[["outcome"]]
x_matrix = example_data[["x"]]
z_matrix = example_data[["z"]]
c_matrix = example_data[["c"]]

PVW_results <- COMMA_PVW(Mstar, outcome, outcome_distribution = "Bernoulli",
                         interaction_indicator = FALSE,
                         x_matrix, z_matrix, c_matrix,
                         beta_start, gamma_start, theta_start)
```

```
PVW_results
```

COMMA_PVW_bootstrap_SE

Estimate Bootstrap Standard Errors using PVW

Description

Estimate Bootstrap Standard Errors using PVW

Usage

```
COMMA_PVW_bootstrap_SE(
  parameter_estimates,
  sigma_estimate,
  n_bootstrap,
  n_parallel,
  outcome_distribution,
  interaction_indicator,
  x_matrix,
  z_matrix,
  c_matrix,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

parameter_estimates
: A column matrix of β, γ, and θ parameter values obtained from a COMMA analysis function. Parameter estimates should be supplied in the following order: 1) β (intercept, slope), 2) γ (intercept and slope from the M = 1 mechanism, intercept and slope from the M = 2 mechanism), and 3) θ (intercept, slope, coefficient for x, slope coefficient for m, slope coefficient for c, and, optionally, slope coefficient for xm if using).

sigma_estimate
: A numeric value specifying the estimated standard deviation. This value is only required if outcome_distribution is "Normal". Default is 1. For non-Normal outcome distributions, the value should be NULL.

n_bootstrap
: A numeric value specifying the number of bootstrap samples to draw.

n_parallel
: A numeric value specifying the number of parallel cores to run the computation on.

outcome_distribution
: A character string specifying the distribution of the outcome variable. Options are "Bernoulli", "Normal", or "Poisson".

`interaction_indicator`
A logical value indicating if an interaction between `x` and `m` should be used to generate the outcome variable, `y`.

`x_matrix` A numeric matrix of predictors in the true mediator and outcome mechanisms. `x_matrix` should not contain an intercept and no values should be `NA`.

`z_matrix` A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept and no values should be `NA`.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`tolerance` A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`max_em_iterations`
A numeric value specifying when to stop estimation, based on the difference of subsequent log-likelihood estimates. The default is `1e-7`.

`em_method` A character string specifying which EM algorithm will be applied. Options are `"em"`, `"squarem"`, or `"pem"`. The default and recommended option is `"squarem"`.

Value

`COMMA_PVW_bootstrap_SE` returns a list with two elements: 1) `bootstrap_df` and 2) `bootstrap_SE`. `bootstrap_df` is a data frame containing `COMMA_PVW` output for each bootstrap sample. `bootstrap_SE` is a data frame containing bootstrap standard error estimates for each parameter.

Examples

```
set.seed(20240709)
sample_size <- 2000

n_cat <- 2 # Number of categories in the binary mediator

# Data generation settings
x_mu <- 0
x_sigma <- 1
z_shape <- 1
c_shape <- 1

# True parameter values (gamma terms set the misclassification rate)
true_beta <- matrix(c(1, -2, .5), ncol = 1)
true_gamma <- matrix(c(1, 1, -.5, -1.5), nrow = 2, byrow = FALSE)
true_theta <- matrix(c(1, 1.5, -2, -.2), ncol = 1)

example_data <- COMMA_data(sample_size, x_mu, x_sigma, z_shape, c_shape,
                           interaction_indicator = FALSE,
                           outcome_distribution = "Bernoulli",
                           true_beta, true_gamma, true_theta)

beta_start <- matrix(rep(1, 3), ncol = 1)
gamma_start <- matrix(rep(1, 4), nrow = 2, ncol = 2)
theta_start <- matrix(rep(1, 4), ncol = 1)
```

```
Mstar = example_data[[”obs_mediator”]]
outcome = example_data[[”outcome”]]
x_matrix = example_data[[”x”]]
z_matrix = example_data[[”z”]]
c_matrix = example_data[[”c”]]

PVW_results <- COMMA_PVW(Mstar, outcome, outcome_distribution = "Bernoulli",
                         interaction_indicator = FALSE,
                         x_matrix, z_matrix, c_matrix,
                         beta_start, gamma_start, theta_start)

PVW_results

PVW_SEs <- COMMA_PVW_bootstrap_SE(PVW_results$Estimates,
                                  sigma_estimate = NULL,
                                  n_bootstrap = 3,
                                  n_parallel = 1,
                                  outcome_distribution = "Bernoulli",
                                  interaction_indicator = FALSE,
                                  x_matrix, z_matrix, c_matrix)

PVW_SEs$bootstrap_SE
```

EM_function_bernoulliY

EM Algorithm Function for Estimation of the Misclassification Model

Description

Function is for cases with $Y \sim Bernoulli$ and with no interaction term in the outcome mechanism.

Usage

```
EM_function_bernoulliY(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the `M* = 1` observed mediator, with the dimensions of Z. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \ldots,$ n_cat. The numeric vector gamma_v is obtained by concatenating the gamma matrix, i.e. gamma_v <- c(gamma_matrix).

obs_mediator A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

obs_outcome A vector containing the outcome variables of interest. There should be no NA terms.

X A numeric design matrix for the true mediator mechanism.

Z A numeric design matrix for the observation mechanism.

c_matrix A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, X or Z.

n_cat The number of categorical values that the true outcome, M, and the observed outcome, M* can take.

Value

EM_function_bernoulliY returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

EM_function_bernoulliY_XM

EM Algorithm Function for Estimation of the Misclassification Model

Description

Function is for cases with $Y \sim Bernoulli$ and with an interaction term in the outcome mechanism.

Usage

```
EM_function_bernoulliY_XM(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the `M* = 1` observed mediator, with the dimensions of `Z`. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \dots,$ `n_cat`. The numeric vector `gamma_v` is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`.

`obs_mediator` A numeric vector of indicator variables (1, 2) for the observed mediator `M*`. There should be no `NA` terms. The reference category is 2.

`obs_outcome` A vector containing the outcome variables of interest. There should be no `NA` terms.

`X` A numeric design matrix for the true mediator mechanism.

`Z` A numeric design matrix for the observation mechanism.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.

`n_cat` The number of categorical values that the true outcome, `M`, and the observed outcome, `M*` can take.

Value

`EM_function_bernoulliY` returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

`EM_function_normalY` *EM Algorithm Function for Estimation of the Misclassification Model*

Description

Function is for cases with $Y \sim Normal$ and with no interaction term in the outcome mechanism.

Usage

```
EM_function_normalY(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the `M* = 1` observed mediator, with the dimensions of `Z`. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \ldots,$ `n_cat`. The numeric vector `gamma_v` is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`.

`obs_mediator` A numeric vector of indicator variables (1, 2) for the observed mediator `M*`. There should be no `NA` terms. The reference category is 2.

`obs_outcome` A vector containing the outcome variables of interest. There should be no `NA` terms.

`X` A numeric design matrix for the true mediator mechanism.

`Z` A numeric design matrix for the observation mechanism.

`c_matrix` A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.

`n_cat` The number of categorical values that the true outcome, `M`, and the observed outcome, `M*` can take.

Value

`EM_function_bernoulliY` returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

EM_function_normalY_XM

EM Algorithm Function for Estimation of the Misclassification Model

Description

Function is for cases with $Y \sim Normal$ and with an interaction term in the outcome mechanism.

Usage

```
EM_function_normalY_XM(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

param_current A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the M* = 1 observed mediator, with the dimensions of Z. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \dots,$ n_cat. The numeric vector gamma_v is obtained by concatenating the gamma matrix, i.e. gamma_v <- c(gamma_matrix).

obs_mediator A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

obs_outcome A vector containing the outcome variables of interest. There should be no NA terms.

X A numeric design matrix for the true mediator mechanism.

Z A numeric design matrix for the observation mechanism.

c_matrix A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, X or Z.

n_cat The number of categorical values that the true outcome, M, and the observed outcome, M* can take.

Value

EM_function_bernoulliY returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

EM_function_poissonY *EM Algorithm Function for Estimation of the Misclassification Model*

Description

Function is for cases with $Y \sim Poisson$ and without an interaction term in the outcome mechanism.

Usage

```
EM_function_poissonY(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

param_current A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the M* = 1 observed mediator, with the dimensions of Z. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \ldots,$ n_cat. The numeric vector gamma_v is obtained by concatenating the gamma matrix, i.e. gamma_v <- c(gamma_matrix).

obs_mediator A numeric vector of indicator variables (1, 2) for the observed mediator M*. There should be no NA terms. The reference category is 2.

obs_outcome A vector containing the outcome variables of interest. There should be no NA terms.

X A numeric design matrix for the true mediator mechanism.

Z A numeric design matrix for the observation mechanism.

c_matrix A numeric matrix of covariates in the true mediator and outcome mechanisms. c_matrix should not contain an intercept and no values should be NA.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, X or Z.

n_cat The number of categorical values that the true outcome, M, and the observed outcome, M* can take.

Value

EM_function_bernoulliY returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

EM_function_poissonY_XM

EM Algorithm Function for Estimation of the Misclassification Model

---

Description

Function is for cases with $Y \sim Poisson$ and with an interaction term in the outcome mechanism.

Usage

```
EM_function_poissonY_XM(
  param_current,
  obs_mediator,
  obs_outcome,
  X,
  Z,
  c_matrix,
  sample_size,
  n_cat
)
```

Arguments

param_current A numeric vector of regression parameters, in the order β, γ, θ. The γ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the `M* = 1` observed mediator, with the dimensions of `Z`. In matrix form, the gamma parameter matrix columns correspond to the true mediator categories $j = 1, \dots,$ `n_cat`. The numeric vector `gamma_v` is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`.

obs_mediator A numeric vector of indicator variables (1, 2) for the observed mediator `M*`. There should be no `NA` terms. The reference category is 2.

obs_outcome A vector containing the outcome variables of interest. There should be no `NA` terms.

X A numeric design matrix for the true mediator mechanism.

Z A numeric design matrix for the observation mechanism.

c_matrix A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.

n_cat The number of categorical values that the true outcome, `M`, and the observed outcome, `M*` can take.

Value

`EM_function_bernoulliY` returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

---

misclassification_prob

Compute Conditional Probability of Observed Mediator Given True Mediator, for Every Subject

---

Description

Compute the conditional probability of observing mediator $M^* \in \{1, 2\}$ given the latent true mediator $M \in \{1, 2\}$ as $\frac{\exp\{\gamma_{kj0} + \gamma_{kjZ} Z_i\}}{1 + \exp\{\gamma_{kj0} + \gamma_{kjZ} Z_i\}}$ for each of the $i = 1, \dots,$ `n` subjects.

Usage

```
misclassification_prob(gamma_matrix, z_matrix)
```

Arguments

`gamma_matrix` A numeric matrix of estimated regression parameters for the observation mechanism, `M* | M` (observed mediator, given the true mediator) ~ `Z` (misclassification predictor matrix). Rows of the matrix correspond to parameters for the `M* = 1` observed mediator, with the dimensions of `z_matrix`. Columns of the matrix correspond to the true mediator categories $j = 1, \dots,$ `n_cat`. The matrix should be obtained by `COMMA_EM`, `COMMA_PVW`, or `COMMA_OLS`.

`z_matrix` A numeric matrix of covariates in the observation mechanism. `z_matrix` should not contain an intercept.

Value

`misclassification_prob` returns a dataframe containing four columns. The first column, `Subject`, represents the subject ID, from 1 to `n`, where `n` is the sample size, or equivalently, the number of rows in `z_matrix`. The second column, `M`, represents a true, latent mediator category $M \in \{1, 2\}$. The third column, `Mstar`, represents an observed outcome category $M^* \in \{1, 2\}$. The last column, `Probability`, is the value of the equation $\frac{\exp\{\gamma_{kj0}+\gamma_{kjZ}Z_i\}}{1+\exp\{\gamma_{kj0}+\gamma_{kjZ}Z_i\}}$ computed for each subject, observed mediator category, and true, latent mediator category.

Examples

```
set.seed(123)
sample_size <- 1000
cov1 <- rnorm(sample_size)
cov2 <- rnorm(sample_size, 1, 2)
z_matrix <- matrix(c(cov1, cov2), nrow = sample_size, byrow = FALSE)
estimated_gammas <- matrix(c(1, -1, .5, .2, -.6, 1.5), ncol = 2)
P_Ystar_M <- misclassification_prob(estimated_gammas, z_matrix)
head(P_Ystar_M)
```

`NCHS2022_sample` *Example data from the National Vital Statistics System of the National Center for Health Statistics (NCHS), 2022*

Description

Example data from the National Vital Statistics System of the National Center for Health Statistics (NCHS), 2022

Usage

```
NCHS2022_sample
```

Format

A dataframe 30 columns, including demographic and birth information for a random sample of 20,000 singleton births from nulliparous mothers in the US in 2022.

Source

https://data.nber.org/nvss/natality/inputs/raw/2022/

Examples

```
## Not run:
data("NCHS2022_sample")
head(NCHS2022_sample)

## End(Not run)
```

`pistar_compute` *Compute Conditional Probability of Each Observed Outcome Given Each True Outcome, for Every Subject*

Description

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome, for Every Subject

Usage

```
pistar_compute(gamma, Z, n, n_cat)
```

Arguments

`gamma` A numeric matrix of regression parameters for the observed outcome mechanism, `Y* | Y` (observed outcome, given the true outcome) ~ `Z` (misclassification predictor matrix). Rows of the matrix correspond to parameters for the `Y* = 1` observed outcome, with the dimensions of `Z`. Columns of the matrix correspond to the true outcome categories $j = 1, \dots,$ `n_cat`.

`Z` A numeric design matrix.

`n` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `Z`.

`n_cat` The number of categorical values that the true outcome, `Y`, and the observed outcome, `Y*` can take.

Value

`pistar_compute` returns a matrix of conditional probabilities, $P(Y_i^* = k|Y_i = j, Z_i) = \frac{\exp\{\gamma_{kj0}+\gamma_{kjZ}Z_i\}}{1+\exp\{\gamma_{kj0}+\gamma_{kjZ}Z_i\}}$ for each of the $i = 1, \dots,$ `n` subjects. Rows of the matrix correspond to each subject and observed outcome. Specifically, the probability for subject i and observed category \$1\$ occurs at row i. The probability for subject i and observed category \$2\$ occurs at row $i+$ `n`. Columns of the matrix correspond to the true outcome categories $j = 1, \dots,$ `n_cat`.

pi_compute *Compute Probability of Each True Outcome, for Every Subject*

Description

Compute Probability of Each True Outcome, for Every Subject

Usage

```
pi_compute(beta, X, n, n_cat)
```

Arguments

beta	A numeric column matrix of regression parameters for the Y (true outcome) ~ X (predictor matrix of interest).
X	A numeric design matrix.
n	An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, X.
n_cat	The number of categorical values that the true outcome, Y, can take.

Value

pi_compute returns a matrix of probabilities, $P(Y_i = j|X_i) = \frac{\exp(X_i \beta)}{1+\exp(X_i \beta)}$ for each of the $i = 1, \ldots,$ n subjects. Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true outcome categories $j = 1, \ldots,$ n_cat.

sum_every_n *Sum Every "n"th Element*

Description

Sum Every "n"th Element

Usage

```
sum_every_n(x, n)
```

Arguments

x	A numeric vector to sum over
n	A numeric value specifying the distance between the reference index and the next index to be summed

Value

sum_every_n returns a vector of sums of every nth element of the vector x.

`sum_every_n1` *Sum Every "n"th Element, then add 1*

Description

Sum Every "n"th Element, then add 1

Usage

```
sum_every_n1(x, n)
```

Arguments

`x`	A numeric vector to sum over
`n`	A numeric value specifying the distance between the reference index and the next index to be summed

Value

`sum_every_n1` returns a vector of sums of every `n`th element of the vector `x`, plus 1.

`theta_optim` *Likelihood Function for Normal Outcome Mechanism with a Binary Mediator*

Description

Likelihood Function for Normal Outcome Mechanism with a Binary Mediator

Usage

```
theta_optim(param_start, m, x, c_matrix, outcome, sample_size, n_cat)
```

Arguments

`param_start`	A numeric vector or column matrix of starting values for the θ parameters in the outcome mechanism and σ parameter. The number of elements in `param_start` should be equal to the number of columns of `x_matrix` and `c_matrix` plus 2 (if `interaction_indicator` is `FALSE`) or 3 (if `interaction_indicator` is `TRUE`). Starting values should be provided in the following order: intercept, slope coefficient for the `x_matrix` term, slope coefficient for the mediator `m` term, slope coefficient for first column of the `c_matrix`, ..., slope coefficient for the final column of the `c_matrix`, and, optionally, slope coefficient for `xm`). The final entry should be the starting value for σ.
`m`	A vector or column matrix containing the true binary mediator or the E-step weight (with values between 0 and 1). There should be no `NA` terms.

`x`	A vector or column matrix of the predictor or exposure of interest. There should be no `NA` terms.
`c_matrix`	A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.
`outcome`	A vector containing the outcome variables of interest. There should be no `NA` terms.
`sample_size`	An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.
`n_cat`	The number of categorical values that the true outcome, `M`, and the observed outcome, `M*` can take.

Value

`theta_optim` returns a numeric value of the (negative) log-likelihood function.

`theta_optim_XM` *Likelihood Function for Normal Outcome Mechanism with a Binary Mediator and an Interaction Term*

Description

Likelihood Function for Normal Outcome Mechanism with a Binary Mediator and an Interaction Term

Usage

```
theta_optim_XM(param_start, m, x, c_matrix, outcome, sample_size, n_cat)
```

Arguments

`param_start`	A numeric vector or column matrix of starting values for the θ parameters in the outcome mechanism and σ parameter. The number of elements in `param_start` should be equal to the number of columns of `x_matrix` and `c_matrix` plus 2 (if `interaction_indicator` is `FALSE`) or 3 (if `interaction_indicator` is `TRUE`). Starting values should be provided in the following order: intercept, slope coefficient for the `x_matrix` term, slope coefficient for the mediator `m` term, slope coefficient for first column of the `c_matrix`, ..., slope coefficient for the final column of the `c_matrix`, and, optionally, slope coefficient for `xm`). The final entry should be the starting value for σ.
`m`	vector or column matrix containing the true binary mediator or the E-step weight (with values between 0 and 1). There should be no `NA` terms.
`x`	A vector or column matrix of the predictor or exposure of interest. There should be no `NA` terms.
`c_matrix`	A numeric matrix of covariates in the true mediator and outcome mechanisms. `c_matrix` should not contain an intercept and no values should be `NA`.

`outcome` A vector containing the outcome variables of interest. There should be no `NA` terms.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the design matrix, `X` or `Z`.

`n_cat` The number of categorical values that the true outcome, `M`, and the observed outcome, `M*` can take.

Value

`theta_optim_XM` returns a numeric value of the (negative) log-likelihood function.

`true_classification_prob` *Compute Probability of Each True Mediator, for Every Subject*

Description

Compute the probability of the latent true mediator $M \in \{1, 2\}$ as $P(M_i = j|X_i) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$ for each of the $i = 1, \dots,$ `n` subjects.

Usage

```
true_classification_prob(beta_matrix, x_matrix)
```

Arguments

`beta_matrix` A numeric column matrix of estimated regression parameters for the true mediator mechanism, `M` (true mediator) ~ `X` (predictor matrix of interest), obtained from `COMMA_EM`, `COMMA_PVW`, or `COMMA_OLS`.

`x_matrix` A numeric matrix of covariates in the true mediator mechanism. `x_matrix` should not contain an intercept.

Value

`true_classification_prob` returns a dataframe containing three columns. The first column, `Subject`, represents the subject ID, from 1 to `n`, where `n` is the sample size, or equivalently, the number of rows in `x_matrix`. The second column, `M`, represents a true, latent mediator category $M \in \{1, 2\}$. The last column, `Probability`, is the value of the equation $P(M_i = j|X_i) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$ computed for each subject and true, latent mediator category.

Examples

```
set.seed(123)
sample_size <- 1000
cov1 <- rnorm(sample_size)
cov2 <- rnorm(sample_size, 1, 2)
x_matrix <- matrix(c(cov1, cov2), nrow = sample_size, byrow = FALSE)
estimated_betas <- matrix(c(1, -1, .5), ncol = 1)
P_M <- true_classification_prob(estimated_betas, x_matrix)
head(P_M)
```

---

w_m_binaryY *Compute E-step for Binary Mediator Misclassification Model Estimated With the EM Algorithm*

Description

Note that this function should only be used for Binary outcome models.

Usage

```
w_m_binaryY(
  mstar_matrix,
  outcome_matrix,
  pistar_matrix,
  pi_matrix,
  p_yi_m0,
  p_yi_m1,
  sample_size,
  n_cat
)
```

Arguments

`mstar_matrix` A numeric matrix of indicator variables (0, 1) for the observed mediator `M*`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed mediator category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`outcome_matrix` A numeric matrix of indicator variables (0, 1) for the observed outcome `Y`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`pistar_matrix` A numeric matrix of conditional probabilities obtained from the internal function `pistar_compute`. Rows of the matrix correspond to each subject and to each observed mediator category. Columns of the matrix correspond to each true, latent mediator category.

pi_matrix A numeric matrix of probabilities obtained from the internal function pi_compute. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each true, latent mediator category.

p_yi_m0 A numeric vector of outcome probabilities computed assuming a true mediator value of 0.

p_yi_m1 A numeric vector of outcome probabilities computed assuming a true mediator value of 1.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the observed mediator matrix, mstar_matrix.

n_cat The number of categorical values that the true outcome, M, and the observed outcome, M*, can take.

Value

w_m_binaryY returns a matrix of E-step weights for the EM-algorithm. Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true mediator categories $j = 1, \dots,$ n_cat.

w_m_normalY *Compute E-step for Binary Mediator Misclassification Model Estimated With the EM Algorithm*

Description

Note that this function should only be used for Normal outcome models.

Usage

```
w_m_normalY(
  mstar_matrix,
  pistar_matrix,
  pi_matrix,
  p_yi_m0,
  p_yi_m1,
  sample_size,
  n_cat
)
```

Arguments

mstar_matrix A numeric matrix of indicator variables (0, 1) for the observed mediator M*. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed mediator category. Each row should contain exactly one 0 entry and exactly one 1 entry.

pistar_matrix A numeric matrix of conditional probabilities obtained from the internal function pistar_compute. Rows of the matrix correspond to each subject and to each observed mediator category. Columns of the matrix correspond to each true, latent mediator category.

pi_matrix A numeric matrix of probabilities obtained from the internal function pi_compute. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each true, latent mediator category.

p_yi_m0 A numeric vector of Normal outcome likelihoods computed assuming a true mediator value of 0.

p_yi_m1 A numeric vector of Normal outcome likelihoods computed assuming a true mediator value of 1.

sample_size An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the observed mediator matrix, mstar_matrix.

n_cat The number of categorical values that the true outcome, M, and the observed outcome, M*, can take.

Value

w_m_normalY returns a matrix of E-step weights for the EM-algorithm. Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true mediator categories $j = 1, \ldots,$ n_cat.

---

w_m_poissonY *Compute E-step for Binary Mediator Misclassification Model Estimated With the EM Algorithm*

Description

Note that this function should only be used for Poisson outcome models.

Usage

```
w_m_poissonY(
  mstar_matrix,
  outcome_matrix,
  pistar_matrix,
  pi_matrix,
  p_yi_m0,
  p_yi_m1,
  sample_size,
  n_cat
)
```

Arguments

`mstar_matrix` A numeric matrix of indicator variables (0, 1) for the observed mediator `M*`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed mediator category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`outcome_matrix` A numeric matrix of indicator variables (0, 1) for the observed outcome `Y`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row should contain exactly one 0 entry and exactly one 1 entry.

`pistar_matrix` A numeric matrix of conditional probabilities obtained from the internal function `pistar_compute`. Rows of the matrix correspond to each subject and to each observed mediator category. Columns of the matrix correspond to each true, latent mediator category.

`pi_matrix` A numeric matrix of probabilities obtained from the internal function `pi_compute`. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each true, latent mediator category.

`p_yi_m0` A numeric vector of outcome probabilities computed assuming a true mediator value of 0.

`p_yi_m1` A numeric vector of outcome probabilities computed assuming a true mediator value of 1.

`sample_size` An integer value specifying the number of observations in the sample. This value should be equal to the number of rows of the observed mediator matrix, `mstar_matrix`.

`n_cat` The number of categorical values that the true outcome, `M`, and the observed outcome, `M*`, can take.

Value

`w_m_poissonY` returns a matrix of E-step weights for the EM-algorithm. Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true mediator categories $j = 1, \dots,$ `n_cat`.

Index