Package: COMBO (via r-universe)

October 14, 2026

Title Correcting Misclassified Binary Outcomes in Association Studies

Version 1.2.0

Author Kimberly Hochstedler Webb [aut, cre]

Maintainer Kimberly Hochstedler Webb <kah343@cornell.edu>

Description Use frequentist and Bayesian methods to estimate parameters from a binary outcome misclassification model. These methods correct for the problem of ``label switching'' by assuming that the sum of outcome sensitivity and specificity is at least 1. A description of the analysis methods is available in Hochstedler and Wells (2023) <doi:10.48550/arXiv.2303.10215>.

Depends R (>= 4.2.0)

Imports dplyr (>= 1.0.10), tidyr (>= 1.2.1), Matrix (> 1.4-1), rjags (>= 4-13), turboEM (>= 2021.1), SAMBA (>= 0.9.0), utils (>= 4.2.0)

Suggests knitr (>= 1.40), testthat (>= 3.0.0), devtools (>= 2.4.5), xtable (>= 1.8.0)

SystemRequirements JAGS (http://mcmc-jags.sourceforge.net)

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_data.R' 'expit.R' 'perfect_sensitivity_EM.R' 'w_j.R' 'q_gamma_f.R' 'q_beta_f.R' 'em_function.R' 'loglik.R' 'COMBO_EM.R' 'pitilde_compute.R' 'COMBO_data_2stage.R' 'w_j_2stage.R' 'q_delta_f.R' 'em_function_2stage.R' 'loglik_2stage.R' 'COMBO_EM_2stage.R' 'COMBO_EM_data.R' 'label_switch.R' 'check_and_fix_chains.R' 'mean_pistarjj_compute.R' 'pistar_compute_for_chains.R' 'pistar_by_chain.R' 'naive_jags_picker.R' 'naive_model_picker.R' 'jags_picker.R' 'model_picker.R'

'COMBO_MCMC.R' 'model_picker_2stage.R' 'COMBO_MCMC_2stage.R'
'LSAC_data.R' 'VPRAI_synthetic_data.R'
'check_and_fix_chains_2stage.R' 'jags_picker_2stage.R'
'label_switch_2stage.R' 'misclassification_prob.R'
'misclassification_prob2.R' 'naive_jags_picker_2stage.R'
'naive_loglik_2stage.R' 'naive_model_picker_2stage.R'
'pistar_by_chain_2stage.R' 'pistar_compute_for_chains_2stage.R'
'pitilde_by_chain.R' 'pitilde_compute_for_chains.R'
'true_classification_prob.R'

LazyData true

Config/testthat/edition 3

NeedsCompilation no

Config/pak/sysreqs jags

Repository https://cran.r-universe.dev

Date/Publication 2024-10-30 15:50:02 UTC

RemoteUrl https://github.com/cran/COMBO

RemoteRef HEAD

RemoteSha eb3515bd0357257cd3a1e4677adfdf48e9a7552e

Contents

check_and_fix_chains . . . 3
check_and_fix_chains_2stage . . . 4
COMBO_data . . . 5
COMBO_data_2stage . . . 7
COMBO_EM . . . 9
COMBO_EM_2stage . . . 11
COMBO_EM_data . . . 13
COMBO_MCMC . . . 14
COMBO_MCMC_2stage . . . 17
em_function . . . 23
em_function_2stage . . . 24
expit . . . 25
jags_picker . . . 25
jags_picker_2stage . . . 27
label_switch . . . 29
label_switch_2stage . . . 30
loglik . . . 31
loglik_2stage . . . 32
LSAC_data . . . 33
mean_pistarjj_compute . . . 34
misclassification_prob . . . 34
misclassification_prob2 . . . 35
model_picker . . . 36
model_picker_2stage . . . 37

naive_jags_picker 37
naive_jags_picker_2stage 39
naive_loglik_2stage 41
naive_model_picker 42
naive_model_picker_2stage 42
perfect_sensitivity_EM 43
pi_compute 44
pistar_by_chain 45
pistar_by_chain_2stage 45
pistar_compute 46
pistar_compute_for_chains 47
pistar_compute_for_chains_2stage 48
pitilde_by_chain 48
pitilde_compute 49
pitilde_compute_for_chains 50
q_beta_f 51
q_delta_f 51
q_gamma_f 52
sum_every_n 53
sum_every_n1 54
true_classification_prob 54
VPRAI_synthetic_data 55
w_j 56
w_j_2stage 57

Index **59**

`check_and_fix_chains` *Check Assumption and Fix Label Switching if Assumption is Broken for a List of MCMC Samples*

Description

Check Assumption and Fix Label Switching if Assumption is Broken for a List of MCMC Samples

Usage

```
check_and_fix_chains(
  n_chains,
  chains_list,
  pistarjj_matrix,
  dim_x,
  dim_z,
  n_cat
)
```

Arguments

n_chains	An integer specifying the number of MCMC chains to compute over.
chains_list	A numeric list containing the samples from n_chains MCMC chains.
pistarjj_matrix	A numeric matrix of the average conditional probability $P(Y^* = j \mid Y = j, Z)$ across all subjects for each MCMC chain, obtained from the pistar_by_chain function.
dim_x	The number of columns of the design matrix of the true outcome mechanism, X.
dim_z	The number of columns of the design matrix of the observation mechanism, Z.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.

Value

check_and_fix_chains returns a numeric list of the samples from n_chains MCMC chains which have been corrected for label switching if the following assumption is not met: $P(Y^* = j \mid Y = j, Z) > 0.50 \forall j$.

---

check_and_fix_chains_2stage

Check Assumption and Fix Label Switching if Assumption is Broken for a List of MCMC Samples

---

Description

Check Assumption and Fix Label Switching if Assumption is Broken for a List of MCMC Samples

Usage

```
check_and_fix_chains_2stage(
  n_chains,
  chains_list,
  pistarjj_matrix,
  pitildejjj_matrix,
  dim_x,
  dim_z,
  dim_v,
  n_cat
)
```

Arguments

n_chains	An integer specifying the number of MCMC chains to compute over.
chains_list	A numeric list containing the samples from n_chains MCMC chains.
pistarjj_matrix	A numeric matrix of the average conditional probability $P(Y^* = j \mid Y = j, Z)$ across all subjects for each MCMC chain, obtained from the pistar_by_chain function.
pitildejjj_matrix	A numeric matrix of the average conditional probability $P(\tilde{Y} = j \mid Y^* = j, Y = j, V)$ across all subjects for each MCMC chain. Rows of the matrix correspond to MCMC chains, up to n_chains. Obtained from the pitilde_by_chain function.
dim_x	The number of columns of the design matrix of the true outcome mechanism, X.
dim_z	The number of columns of the design matrix of the first-stage observation mechanism, Z.
dim_v	The number of columns of the design matrix of the second-stage observation mechanism, V.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.

Value

check_and_fix_chains returns a numeric list of the samples from n_chains MCMC chains which have been corrected for label switching if the following assumption is not met: $P(Y^* = j \mid Y = j, Z) > 0.50 \forall j$.

COMBO_data *Generate Data to use in COMBO Functions*

Description

Generate Data to use in COMBO Functions

Usage

```
COMBO_data(sample_size, x_mu, x_sigma, z_shape, beta, gamma)
```

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.
beta	A column matrix of β parameter values (intercept, slope) to generate data under in the true outcome mechanism.
gamma	A numeric matrix of γ parameters to generate data under in the observation mechanism. 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 outcome categories $Y \in \{1, 2\}$.

Value

COMBO_data returns a list of generated data elements:

obs_Y	A vector of observed outcomes.
true_Y	A vector of true outcomes.
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 contains exactly one 0 entry and exactly one 1 entry.
x	A vector of generated predictor values in the true outcome mechanism, from the Normal distribution.
z	A vector of generated predictor values in the observation mechanism from the Gamma distribution.
x_design_matrix	The design matrix for the x predictor.
z_design_matrix	The design matrix for the z predictor.

Examples

```
set.seed(123)
n <- 500
x_mu <- 0
x_sigma <- 1
z_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)

my_data <- COMBO_data(sample_size = n,
                      x_mu = x_mu, x_sigma = x_sigma,
                      z_shape = z_shape,
                      beta = true_beta, gamma = true_gamma)
table(my_data[["obs_Y"]], my_data[["true_Y"]])
```

`COMBO_data_2stage` *Generate data to use in two-stage COMBO Functions*

Description

Generate data to use in two-stage COMBO Functions

Usage

```
COMBO_data_2stage(
  sample_size,
  x_mu,
  x_sigma,
  z1_shape,
  z2_shape,
  beta,
  gamma1,
  gamma2
)
```

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.
`z1_shape`	A positive numeric value specifying the shape parameter of `z1` predictors generated from a Gamma distribution.
`z2_shape`	A positive numeric value specifying the shape parameter of `z2` predictors generated from a Gamma distribution.
`beta`	A column matrix of β parameter values (intercept, slope) to generate data under in the true outcome mechanism.
`gamma1`	A numeric matrix of $\gamma^{(1)}$ parameters to generate data under in the first-stage observation mechanism. In matrix form, the `gamma1` matrix rows correspond to intercept (row 1) and slope (row 2) terms. The `gamma1` parameter matrix columns correspond to the true outcome categories $Y \in \{1, 2\}$.
`gamma2`	A numeric array of $\gamma^{(2)}$ parameters to generate data under the second-stage observation mechanism. In array form, the `gamma2` matrix rows correspond to intercept (row 1) and slope (row 2) terms. The matrix columns correspond to first-stage observed outcome categories. The third dimension of the `gamma2` array is indexed by the true outcome categories.

Value

COMBO_data_2stage returns a list of generated data elements:

obs_Ystar1 A vector of first-stage observed outcomes.

obs_Ystar2 A vector of second-stage observed outcomes.

true_Y A vector of true outcomes.

obs_Ystar1_matrix
A numeric matrix of indicator variables (0, 1) for the first-stage observed outcome $Y^{*(1)}$. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row contains exactly one 0 entry and exactly one 1 entry.

obs_Ystar2_matrix
A numeric matrix of indicator variables (0, 1) for the second-stage observed outcome $Y^{*(2)}$. Rows of the matrix correspond to each subject. Columns of the matrix correspond to each observed outcome category. Each row contains exactly one 0 entry and exactly one 1 entry.

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

z1 A vector of generated predictor values in the first-stage observation mechanism from the Gamma distribution.

z2 A vector of generated predictor values in the second-stage observation mechanism from the Gamma distribution.

x_design_matrix
The design matrix for the x predictor.

z1_design_matrix
The design matrix for the z1 predictor.

z2_design_matrix
The design matrix for the z2 predictor.

Examples

```
set.seed(123)
n <- 1000
x_mu <- 0
x_sigma <- 1
z1_shape <- 1
z2_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma1 <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
true_gamma2 <- array(c(1.5, 1, .5, .5, -.5, 0, -1, -1), dim = c(2, 2, 2))

my_data <- COMBO_data_2stage(sample_size = n,
                             x_mu = x_mu, x_sigma = x_sigma,
                             z1_shape = z1_shape, z2_shape = z2_shape,
                            beta = true_beta, gamma1 = true_gamma1, gamma2 = true_gamma2)
table(my_data[["obs_Ystar2"]], my_data[["obs_Ystar1"]], my_data[["true_Y"]])
```

COMBO_EM *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(
  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 $Y \in \{1, 2\}$. 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 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.

Estimates are provided for the binary misclassification model, as well as two additional cases. The "SAMBA" parameter estimates are from the R Package, SAMBA, which uses the EM algorithm to estimate a binary outcome misclassification model that assumes there is perfect specificity. The "PSens" parameter estimates are estimated using the EM algorithm for the binary outcome misclassification model that assumes there is perfect sensitivitiy. The "Naive" parameter estimates are from a simple logistic regression Y* ~ X.

References

Beesley, L. and Mukherjee, B. (2020). Statistical inference for association studies using electronic health records: Handling both selection bias and outcome misclassification. Biometrics, 78, 214-226.

Examples

```
set.seed(123)
n <- 1000
x_mu <- 0
x_sigma <- 1
z_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)

x_matrix = matrix(rnorm(n, x_mu, x_sigma), ncol = 1)
X = matrix(c(rep(1, n), x_matrix[,1]), ncol = 2, byrow = FALSE)
z_matrix = matrix(rgamma(n, z_shape), ncol = 1)
Z = matrix(c(rep(1, n), z_matrix[,1]), ncol = 2, byrow = FALSE)

exp_xb = exp(X %*% true_beta)
pi_result = exp_xb[,1] / (exp_xb[,1] + 1)
pi_matrix = matrix(c(pi_result, 1 - pi_result), ncol = 2, byrow = FALSE)

true_Y <- rep(NA, n)
for(i in 1:n){
    true_Y[i] = which(stats::rmultinom(1, 1, pi_matrix[i,]) == 1)
}

exp_zg = exp(Z %*% true_gamma)
pistar_denominator = matrix(c(1 + exp_zg[,1], 1 + exp_zg[,2]), ncol = 2, byrow = FALSE)
pistar_result = exp_zg / pistar_denominator

pistar_matrix = matrix(c(pistar_result[,1], 1 - pistar_result[,1],
                         pistar_result[,2], 1 - pistar_result[,2]),
                       ncol = 2, byrow = FALSE)
```

```
obs_Y <- rep(NA, n)
for(i in 1:n){
    true_j = true_Y[i]
    obs_Y[i] = which(rmultinom(1, 1,
                    pistar_matrix[c(i, n + i),
                                  true_j]) == 1)
 }

Ystar <- obs_Y

starting_values <- rep(1,6)
beta_start <- matrix(starting_values[1:2], ncol = 1)
gamma_start <- matrix(starting_values[3:6], ncol = 2, nrow = 2, byrow = FALSE)

EM_results <- COMBO_EM(Ystar, x_matrix = x_matrix, z_matrix = z_matrix,
                       beta_start = beta_start, gamma_start = gamma_start)

EM_results
```

COMBO_EM_2stage *EM-Algorithm Estimation of the Two-Stage Binary Outcome Misclassification Model*

Description

Jointly estimate β, $\gamma^{(1)}$, $\gamma^{(2)}$ parameters from the true outcome, first-stage observation, and second-stage observation mechanisms, respectively, in a two-stage binary outcome misclassification model.

Usage

```
COMBO_EM_2stage(
  Ystar1,
  Ystar2,
  x_matrix,
  z1_matrix,
  z2_matrix,
  beta_start,
  gamma1_start,
  gamma2_start,
  tolerance = 1e-07,
  max_em_iterations = 1500,
  em_method = "squarem"
)
```

Arguments

Ystar1	A numeric vector of indicator variables (1, 2) for the first-stage observed outcome $Y^{*(1)}$. There should be no NA terms. The reference category is 2.

`Ystar2` A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $Y^{*(2)}$. 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`.

`z1_matrix` A numeric matrix of covariates in the first-stage observation mechanism. `z1_matrix` should not contain an intercept and no values should be `NA`.

`z2_matrix` A numeric matrix of covariates in the second-stage observation mechanism. `z2_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.

`gamma1_start` A numeric vector or matrix of starting values for the $\gamma^{(1)}$ parameters in the first-stage observation mechanism. In matrix form, the `gamma1_start` matrix rows correspond to parameters for the $Y^{*(1)} = 1$ first-stage observed outcome, with the dimensions of `z1_matrix` plus 1, and the parameter matrix columns correspond to the true outcome categories $Y \in \{1, 2\}$. A numeric vector for `gamma1_start` is obtained by concatenating the matrix, i.e. `gamma1_start <- c(gamma1_matrix)`.

`gamma2_start` A numeric array of starting values for the $\gamma^{(2)}$ parameters in the second-stage observation mechanism. The first dimension (matrix rows) of `gamma2_start` correspond to parameters for the $Y^{*(2)} = 1$ second-stage observed outcome, with the dimensions of the `z2_matrix` plus 1. The second dimension (matrix columns) correspond to the first-stage observed outcome categories $Y^{*(1)} \in \{1, 2\}$. The third dimension of `gamma2_start` corresponds to to the true outcome categories $Y \in \{1, 2\}$.

`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_2stage` 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.

Estimates are provided for the two-stage binary misclassification model.

Examples

```
set.seed(123)
n <- 1000
x_mu <- 0
```

```
x_sigma <- 1
z1_shape <- 1
z2_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma1 <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
true_gamma2 <- array(c(1.5, 1, .5, .5, -.5, 0, -1, -1), dim = c(2, 2, 2))

my_data <- COMBO_data_2stage(sample_size = n,
                             x_mu = x_mu, x_sigma = x_sigma,
                             z1_shape = z1_shape, z2_shape = z2_shape,
                            beta = true_beta, gamma1 = true_gamma1, gamma2 = true_gamma2)
table(my_data[["obs_Ystar2"]], my_data[["obs_Ystar1"]], my_data[["true_Y"]])

beta_start <- rnorm(length(c(true_beta)))
gamma1_start <- rnorm(length(c(true_gamma1)))
gamma2_start <- rnorm(length(c(true_gamma2)))

EM_results <- COMBO_EM_2stage(Ystar1 = my_data[["obs_Ystar1"]],
                              Ystar2 = my_data[["obs_Ystar2"]],
                              x_matrix = my_data[["x"]],
                              z1_matrix = my_data[["z1"]],
                              z2_matrix = my_data[["z2"]],
                              beta_start = beta_start,
                              gamma1_start = gamma1_start,
                              gamma2_start = gamma2_start)

EM_results
```

`COMBO_EM_data` *Test data for the COMBO_EM function*

Description

A dataset for testing the `COMBO_EM` function, generated from the `COMBO_data` function.

Usage

```
COMBO_EM_data
```

Format

A list containing 6 variables for 1000 observations:

Y The true outcome variable

Ystar The observed outcome variable

x_matrix A matrix of predictor values in the true outcome mechanism

z_matrix A matrix of predictor values in the observed outcome mechanism

true_beta Beta parameter values used for data generation in the true outcome mechanism

true_gamma Gamma parameter values used for data generation in the observed outcome mechanism

Examples

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

## End(Not run)
```

COMBO_MCMC *MCMC 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_MCMC(
  Ystar,
  x_matrix,
  z_matrix,
  prior,
  beta_prior_parameters,
  gamma_prior_parameters,
  number_MCMC_chains = 4,
  MCMC_sample = 2000,
  burn_in = 1000,
  display_progress = TRUE
)
```

Arguments

Ystar	A numeric vector of indicator variables (1, 2) for the observed outcome Y*. The reference category is 2.
x_matrix	A numeric matrix of covariates in the true outcome mechanism. x_matrix should not contain an intercept.
z_matrix	A numeric matrix of covariates in the observation mechanism. z_matrix should not contain an intercept.
prior	A character string specifying the prior distribution for the β and γ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

beta_prior_parameters
: A numeric list of prior distribution parameters for the β terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions n_cat X dim_x, and all elements in the n_cat row should be set to NA.

gamma_prior_parameters
: A numeric list of prior distribution parameters for the γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for γ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for γ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X dim_z, and all elements in the n_cat row should be set to NA.

number_MCMC_chains
: An integer specifying the number of MCMC chains to compute. The default is 4.

MCMC_sample
: An integer specifying the number of MCMC samples to draw. The default is 2000.

burn_in
: An integer specifying the number of MCMC samples to discard for the burn-in period. The default is 1000.

display_progress
: A logical value specifying whether messages should be displayed during model compilation. The default is TRUE.

Value

COMBO_MCMC returns a list of the posterior samples and posterior means for both the binary outcome misclassification model and a naive logistic regression of the observed outcome, Y*, predicted by the matrix x. The list contains the following components:

posterior_sample_df
: A data frame containing three columns. The first column indicates the chain from which a sample is taken, from 1 to number_MCMC_chains. The second column specifies the parameter associated with a given row. β terms have dimensions dim_x X n_cat. The γ terms have dimensions n_cat X n_cat X dim_z, where the first index specifies the observed outcome category and the second index specifies the true outcome category. The final column provides the MCMC sample.

posterior_means_df
: A data frame containing three columns. The first column specifies the parameter associated with a given row. Parameters are indexed as in the posterior_sample_df. The second column provides the posterior mean computed across all chains and all samples. The final column provides the posterior median computed across all chains and all samples.

naive_posterior_sample_df
: A data frame containing three columns. The first column indicates the chain from which a sample is taken, from 1 to number_MCMC_chains. The second column specifies the parameter associated with a given row. Naive β terms have dimensions dim_x X n_cat. The final column provides the MCMC sample.

naive_posterior_means_df
: A data frame containing three columns. The first column specifies the naive parameter associated with a given row. Parameters are indexed as in the naive_posterior_sample_df. The second column provides the posterior mean computed across all chains and all samples. The final column provides the posterior median computed across all chains and all samples.

Examples

```
set.seed(123)
n <- 1000
x_mu <- 0
x_sigma <- 1
z_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)

x_matrix = matrix(rnorm(n, x_mu, x_sigma), ncol = 1)
X = matrix(c(rep(1, n), x_matrix[,1]), ncol = 2, byrow = FALSE)
z_matrix = matrix(rgamma(n, z_shape), ncol = 1)
Z = matrix(c(rep(1, n), z_matrix[,1]), ncol = 2, byrow = FALSE)

exp_xb = exp(X %*% true_beta)
pi_result = exp_xb[,1] / (exp_xb[,1] + 1)
pi_matrix = matrix(c(pi_result, 1 - pi_result), ncol = 2, byrow = FALSE)

true_Y <- rep(NA, n)
for(i in 1:n){
    true_Y[i] = which(stats::rmultinom(1, 1, pi_matrix[i,]) == 1)
}

exp_zg = exp(Z %*% true_gamma)
pistar_denominator = matrix(c(1 + exp_zg[,1], 1 + exp_zg[,2]), ncol = 2, byrow = FALSE)
pistar_result = exp_zg / pistar_denominator

pistar_matrix = matrix(c(pistar_result[,1], 1 - pistar_result[,1],
                         pistar_result[,2], 1 - pistar_result[,2]),
                       ncol = 2, byrow = FALSE)
```

```
obs_Y <- rep(NA, n)
for(i in 1:n){
    true_j = true_Y[i]
    obs_Y[i] = which(rmultinom(1, 1,
                    pistar_matrix[c(i, n + i),
                                  true_j]) == 1)
 }

Ystar <- obs_Y

unif_lower_beta <- matrix(c(-5, -5, NA, NA), nrow = 2, byrow = TRUE)
unif_upper_beta <- matrix(c(5, 5, NA, NA), nrow = 2, byrow = TRUE)

unif_lower_gamma <- array(data = c(-5, NA, -5, NA, -5, NA, -5, NA),
                          dim = c(2,2,2))
unif_upper_gamma <- array(data = c(5, NA, 5, NA, 5, NA, 5, NA),
                          dim = c(2,2,2))

beta_prior_parameters <- list(lower = unif_lower_beta, upper = unif_upper_beta)
gamma_prior_parameters <- list(lower = unif_lower_gamma, upper = unif_upper_gamma)

MCMC_results <- COMBO_MCMC(Ystar, x = x_matrix, z = z_matrix,
                           prior = "uniform",
                           beta_prior_parameters = beta_prior_parameters,
                           gamma_prior_parameters = gamma_prior_parameters,
                           number_MCMC_chains = 2,
                           MCMC_sample = 200, burn_in = 100)
MCMC_results$posterior_means_df
```

COMBO_MCMC_2stage *MCMC Estimation of the Two-Stage Binary Outcome Misclassification Model*

Description

Jointly estimate β, $\gamma^{(1)}$, and $\gamma^{(2)}$ parameters from the true outcome first-stage observation, and second-stage observation mechanisms, respectively, in a two-stage binary outcome misclassification model.

Usage

```
COMBO_MCMC_2stage(
  Ystar1,
  Ystar2,
  x_matrix,
  z1_matrix,
  z2_matrix,
  prior,
  beta_prior_parameters,
```

```
  gamma1_prior_parameters,
  gamma2_prior_parameters,
  naive_gamma2_prior_parameters,
  number_MCMC_chains = 4,
  MCMC_sample = 2000,
  burn_in = 1000,
  display_progress = TRUE
)
```

Arguments

Ystar1	A numeric vector of indicator variables (1, 2) for the observed outcome $Y^{*(1)}$. The reference category is 2.
Ystar2	A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $Y^{*(2)}$. 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.
z1_matrix	A numeric matrix of covariates in the observation mechanism. z1_matrix should not contain an intercept.
z2_matrix	A numeric matrix of covariates in the second-stage observation mechanism. z2_matrix should not contain an intercept and no values should be NA.
prior	A character string specifying the prior distribution for the β, $\gamma^{(1)}$, and $\gamma^{(2)}$ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
beta_prior_parameters	A numeric list of prior distribution parameters for the β terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions n_cat X dim_x, and all elements in the n_cat row should be set to NA.
gamma1_prior_parameters	A numeric list of prior distribution parameters for the $\gamma^{(1)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for $\gamma^{(1)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for $\gamma^{(1)}$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for $\gamma^{(1)}$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X dim_z1, and all elements in the n_cat row should be set to NA.

gamma2_prior_parameters
: A numeric list of prior distribution parameters for the $\gamma^{(2)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for $\gamma^{(2)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for $\gamma^{(2)}$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for $\gamma^{(2)}$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X n_cat X dim_z2, and all elements in the n_cat row should be set to NA.

naive_gamma2_prior_parameters
: A numeric list of prior distribution parameters for the naive model $\gamma^{(2)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for naive $\gamma^{(2)}$ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for naive $\gamma^{(2)}$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for naive $\gamma^{(2)}$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X dim_z2, and all elements in the n_cat row should be set to NA. Note that prior distributions for the naive β terms are inherted from the beta_prior_parameters argument.

number_MCMC_chains
: An integer specifying the number of MCMC chains to compute. The default is 4.

MCMC_sample
: An integer specifying the number of MCMC samples to draw. The default is 2000.

burn_in
: An integer specifying the number of MCMC samples to discard for the burn-in period. The default is 1000.

display_progress
: A logical value specifying whether messages should be displayed during model compilation. The default is TRUE.

Value

COMBO_MCMC_2stage returns a list of the posterior samples and posterior means for both the binary outcome misclassification model and a naive logistic regression of the observed outcome, Y*, predicted by the matrix x. The list contains the following components:

posterior_sample_df
: A data frame containing three columns. The first column indicates the chain from which a sample is taken, from 1 to number_MCMC_chains. The second column specifies the parameter associated with a given row. β terms have dimensions dim_x X n_cat. The $\gamma^{(1)}$ terms have dimensions n_cat X n_cat X dim_z1, where the first index specifies the first-stage observed outcome category

and the second index specifies the true outcome category. The $\gamma^{(2)}$ terms have dimensions n_cat X n_cat X n_cat X dim_z2, where the first index specifies the second-stage observed outcome category, the second index specifies the first-stage observed outcome category, and the third index specifies the true outcome category. The final column provides the MCMC sample.

posterior_means_df
: A data frame containing three columns. The first column specifies the parameter associated with a given row. Parameters are indexed as in the posterior_sample_df. The second column provides the posterior mean computed across all chains and all samples. The final column provides the posterior median computed across all chains and all samples.

naive_posterior_sample_df
: A data frame containing three columns. The first column indicates the chain from which a sample is taken, from 1 to number_MCMC_chains. The second column specifies the parameter associated with a given row. Naive β terms have dimensions dim_x X n_cat. The final column provides the MCMC sample.

naive_posterior_means_df
: A data frame containing three columns. The first column specifies the naive parameter associated with a given row. Parameters are indexed as in the naive_posterior_sample_df. The second column provides the posterior mean computed across all chains and all samples. The final column provides the posterior median computed across all chains and all samples.

Examples

```
# Helper functions
sum_every_n <- function(x, n){
vector_groups = split(x,
                      ceiling(seq_along(x) / n))
sum_x = Reduce(`+`, vector_groups)

return(sum_x)
}

sum_every_n1 <- function(x, n){
vector_groups = split(x,
                      ceiling(seq_along(x) / n))
sum_x = Reduce(`+`, vector_groups) + 1

return(sum_x)
}

# Example

set.seed(123)
n <- 1000
x_mu <- 0
x_sigma <- 1
z1_shape <- 1
```

```
z2_shape <- 1

true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma1 <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
true_gamma2 <- array(c(1.5, 1, .5, .5, -.5, 0, -1, -1), dim = c(2, 2, 2))

x_matrix = matrix(rnorm(n, x_mu, x_sigma), ncol = 1)
X = matrix(c(rep(1, n), x_matrix[,1]), ncol = 2, byrow = FALSE)
z1_matrix = matrix(rgamma(n, z1_shape), ncol = 1)
Z1 = matrix(c(rep(1, n), z1_matrix[,1]), ncol = 2, byrow = FALSE)
z2_matrix = matrix(rgamma(n, z2_shape), ncol = 1)
Z2 = matrix(c(rep(1, n), z2_matrix[,1]), ncol = 2, byrow = FALSE)

exp_xb = exp(X %*% true_beta)
pi_result = exp_xb[,1] / (exp_xb[,1] + 1)
pi_matrix = matrix(c(pi_result, 1 - pi_result), ncol = 2, byrow = FALSE)

true_Y <- rep(NA, n)
for(i in 1:n){
    true_Y[i] = which(stats::rmultinom(1, 1, pi_matrix[i,]) == 1)
}

exp_z1g1 = exp(Z1 %*% true_gamma1)
pistar1_denominator = matrix(c(1 + exp_z1g1[,1], 1 + exp_z1g1[,2]),
                             ncol = 2, byrow = FALSE)
pistar1_result = exp_z1g1 / pistar1_denominator

pistar1_matrix = matrix(c(pistar1_result[,1], 1 - pistar1_result[,1],
                          pistar1_result[,2], 1 - pistar1_result[,2]),
                        ncol = 2, byrow = FALSE)


obs_Y1 <- rep(NA, n)
for(i in 1:n){
    true_j = true_Y[i]
    obs_Y1[i] = which(rmultinom(1, 1,
                      pistar1_matrix[c(i, n + i),
                                     true_j]) == 1)
 }

Ystar1 <- obs_Y1

exp_z2g2_1 = exp(Z2 %*% true_gamma2[,,1])
exp_z2g2_2 = exp(Z2 %*% true_gamma2[,,2])

pi_denominator1 = apply(exp_z2g2_1, FUN = sum_every_n1, n, MARGIN = 2)
pi_result1 = exp_z2g2_1 / rbind(pi_denominator1)

pi_denominator2 = apply(exp_z2g2_2, FUN = sum_every_n1, n, MARGIN = 2)
pi_result2 = exp_z2g2_2 / rbind(pi_denominator2)

pistar2_matrix1 = rbind(pi_result1,
                        1 - apply(pi_result1,
```

```
                                FUN = sum_every_n, n = n,
                                MARGIN = 2))

pistar2_matrix2 = rbind(pi_result2,
                        1 - apply(pi_result2,
                                FUN = sum_every_n, n = n,
                                MARGIN = 2))

pistar2_array = array(c(pistar2_matrix1, pistar2_matrix2),
                  dim = c(dim(pistar2_matrix1), 2))

obs_Y2 <- rep(NA, n)
for(i in 1:n){
    true_j = true_Y[i]
    obs_k = Ystar1[i]
    obs_Y2[i] = which(rmultinom(1, 1,
                                pistar2_array[c(i,n+ i),
                                              obs_k, true_j]) == 1)
}

Ystar2 <- obs_Y2

unif_lower_beta <- matrix(c(-5, -5, NA, NA), nrow = 2, byrow = TRUE)
unif_upper_beta <- matrix(c(5, 5, NA, NA), nrow = 2, byrow = TRUE)

unif_lower_gamma1 <- array(data = c(-5, NA, -5, NA, -5, NA, -5, NA),
                           dim = c(2,2,2))
unif_upper_gamma1 <- array(data = c(5, NA, 5, NA, 5, NA, 5, NA),
                           dim = c(2,2,2))

unif_upper_gamma2 <- array(rep(c(5, NA), 8), dim = c(2,2,2,2))
unif_lower_gamma2 <- array(rep(c(-5, NA), 8), dim = c(2,2,2,2))

unif_lower_naive_gamma2 <- array(data = c(-5, NA, -5, NA, -5, NA, -5, NA),
                                 dim = c(2,2,2))
unif_upper_naive_gamma2 <- array(data = c(5, NA, 5, NA, 5, NA, 5, NA),
                                 dim = c(2,2,2))

beta_prior_parameters <- list(lower = unif_lower_beta, upper = unif_upper_beta)
gamma1_prior_parameters <- list(lower = unif_lower_gamma1, upper = unif_upper_gamma1)
gamma2_prior_parameters <- list(lower = unif_lower_gamma2, upper = unif_upper_gamma2)
naive_gamma2_prior_parameters <- list(lower = unif_lower_naive_gamma2,
                                      upper = unif_upper_naive_gamma2)

MCMC_results <- COMBO_MCMC_2stage(Ystar1, Ystar2,
                                x_matrix = x_matrix, z1_matrix = z1_matrix,
                                z2_matrix = z2_matrix,
                                prior = "uniform",
                                beta_prior_parameters = beta_prior_parameters,
                                gamma1_prior_parameters = gamma1_prior_parameters,
                                gamma2_prior_parameters = gamma2_prior_parameters,
                            naive_gamma2_prior_parameters = naive_gamma2_prior_parameters,
                                number_MCMC_chains = 2,
```

```
                                    MCMC_sample = 200, burn_in = 100)
MCMC_results$posterior_means_df
```

`em_function` *EM-Algorithm Function for Estimation of the Misclassification Model*

Description

EM-Algorithm Function for Estimation of the Misclassification Model

Usage

```
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 $j = 1, \ldots,$ `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

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

`em_function_2stage` *EM-Algorithm Function for Estimation of the Two-Stage Misclassification Model*

Description

EM-Algorithm Function for Estimation of the Two-Stage Misclassification Model

Usage

```
em_function_2stage(
  param_current,
  obs_Ystar_matrix,
  obs_Ytilde_matrix,
  X,
  Z,
  V,
  sample_size,
  n_cat
)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ, δ. 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 $j = 1, \ldots,$ `n_cat`. The numeric vector γ is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`. The δ vector is obtained from the array form. In array form, the first dimension (matrix rows) of `delta` corresponds to parameters for the $\tilde{Y} = 1$ second-stage observed outcome, with the dimensions of the `V` The second dimension (matrix columns) correspond to the first-stage observed outcome categories $Y^* \in \{1, 2\}$. The third dimension of `delta_start` corresponds to to the true outcome categories $Y \in \{1, 2\}$. The numeric vector δ is obtained by concatenating the delta array, i.e. `delta_vector <- c(delta_array)`.

`obs_Ystar_matrix`
A numeric matrix of indicator variables (0, 1) for the first-stage 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.

`obs_Ytilde_matrix`
A numeric matrix of indicator variables (0, 1) for the second-stage observed outcome $\tilde{Y}$. 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 first-stage observation mechanism.
V	A numeric design matrix for the second-stage 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 matrices, X, Z, and V.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcomes, Y* and $\tilde{Y}$, can take.

Value

em_function_2stage returns a numeric vector of updated parameter estimates from one iteration of the EM-algorithm.

---

expit *Expit function*

Description

$\frac{\exp\{x\}}{1+\exp\{x\}}$

Usage

```
expit(x)
```

Arguments

x	A numeric value or vector to compute the expit function on.

Value

expit returns the result of the function $f(x) = \frac{\exp\{x\}}{1+\exp\{x\}}$ for a given x.

---

jags_picker *Set up a Binary Outcome Misclassification* jags.model *Object for a Given Prior*

Description

Set up a Binary Outcome Misclassification jags.model Object for a Given Prior

Usage

```
jags_picker(
  prior,
  sample_size,
  dim_x,
  dim_z,
  n_cat,
  Ystar,
  X,
  Z,
  beta_prior_parameters,
  gamma_prior_parameters,
  number_MCMC_chains,
  model_file,
  display_progress = TRUE
)
```

Arguments

prior	A character string specifying the prior distribution for the β and γ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
sample_size	An integer value specifying the number of observations in the sample.
dim_x	An integer specifying the number of columns of the design matrix of the true outcome mechanism, X.
dim_z	An integer specifying the number of columns of the design matrix of the observation mechanism, Z.
n_cat	An integer specifying the number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.
Ystar	A numeric vector of indicator variables (1, 2) for the observed outcome Y*. The reference category is 2.
X	A numeric design matrix for the true outcome mechanism.
Z	A numeric design matrix for the observation mechanism.
beta_prior_parameters	A numeric list of prior distribution parameters for the β terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions dim_x X n_cat, and all elements in the n_cat column should be set to NA.

gamma_prior_parameters
: A numeric list of prior distribution parameters for the γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for γ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for γ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X dim_z, and all elements in the n_cat row should be set to NA.

number_MCMC_chains
: An integer specifying the number of MCMC chains to compute.

model_file
: A .BUG file and used for MCMC estimation with rjags.

display_progress
: A logical value specifying whether messages should be displayed during model compilation. The default is TRUE.

Value

jags_picker returns a jags.model object for a binary outcome misclassification model. The object includes the specified prior distribution, model, number of chains, and data.

---

jags_picker_2stage *Set up a Two-Stage Binary Outcome Misclassification* jags.model *Object for a Given Prior*

Description

Set up a Two-Stage Binary Outcome Misclassification jags.model Object for a Given Prior

Usage

```
jags_picker_2stage(
  prior,
  sample_size,
  dim_x,
  dim_z,
  dim_v,
  n_cat,
  Ystar,
  Ytilde,
  X,
  Z,
  V,
  beta_prior_parameters,
  gamma_prior_parameters,
```

```
  delta_prior_parameters,
  number_MCMC_chains,
  model_file,
  display_progress = TRUE
)
```

Arguments

prior	A character string specifying the prior distribution for the β, γ, and δ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
sample_size	An integer value specifying the number of observations in the sample.
dim_x	An integer specifying the number of columns of the design matrix of the true outcome mechanism, X.
dim_z	An integer specifying the number of columns of the design matrix of the first-stage observation mechanism, Z.
dim_v	An integer specifying the number of columns of the design matrix of the second-stage observation mechanism, V.
n_cat	An integer specifying the number of categorical values that the true outcome, Y, and the observed outcomes, Y^* and $\tilde{Y}$, can take.
Ystar	A numeric vector of indicator variables (1, 2) for the first-stage observed outcome Y*. The reference category is 2.
Ytilde	A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $\tilde{Y}$. The reference category is 2.
X	A numeric design matrix for the true outcome mechanism.
Z	A numeric design matrix for the first-stage observation mechanism.
V	A numeric design matrix for the second-stage observation mechanism.
beta_prior_parameters	A numeric list of prior distribution parameters for the β terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions dim_x X n_cat, and all elements in the n_cat column should be set to NA.
gamma_prior_parameters	A numeric list of prior distribution parameters for the γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for γ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for γ terms. For

prior distribution "t", the third element of the list should contain an array of the degrees of freedom for γ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X dim_z, and all elements in the n_cat row should be set to NA.

delta_prior_parameters
: A numeric list of prior distribution parameters for the δ terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for δ terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for δ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for δ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions n_cat X n_cat X n_cat X dim_v, and all elements in the n_cat row should be set to NA.

number_MCMC_chains
: An integer specifying the number of MCMC chains to compute.

model_file
: A .BUG file and used for MCMC estimation with rjags.

display_progress
: A logical value specifying whether messages should be displayed during model compilation. The default is TRUE.

Value

jags_picker returns a jags.model object for a two-stage binary outcome misclassification model. The object includes the specified prior distribution, model, number of chains, and data.

---

label_switch *Fix Label Switching in MCMC Results from a Binary Outcome Misclassification Model*

Description

Fix Label Switching in MCMC Results from a Binary Outcome Misclassification Model

Usage

```
label_switch(chain_matrix, dim_x, dim_z, n_cat)
```

Arguments

chain_matrix
: A numeric matrix containing the posterior samples for all parameters in a given MCMC chain. chain_matrix must be a named object (i.e. each parameter must be named as beta[j, p] or gamma[k,j,p]).

dim_x
: An integer specifying the number of columns of the design matrix of the true outcome mechanism, X.

dim_z	An integer specifying the number of columns of the design matrix of the observation mechanism, Z.
n_cat	An integer specifying the number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.

Value

label_switch returns a named matrix of MCMC posterior samples for all parameters after performing label switching according the following pattern: all β terms are multiplied by -1, all γ terms are "swapped" with the opposite j index.

---

label_switch_2stage *Fix Label Switching in MCMC Results from a Binary Outcome Misclassification Model*

Description

Fix Label Switching in MCMC Results from a Binary Outcome Misclassification Model

Usage

```
label_switch_2stage(chain_matrix, dim_x, dim_z, dim_v, n_cat)
```

Arguments

chain_matrix	A numeric matrix containing the posterior samples for all parameters in a given MCMC chain. chain_matrix must be a named object (i.e. each parameter must be named as beta[j, p], gamma[k,j,p], or delta[l,k,j,p]).
dim_x	An integer specifying the number of columns of the design matrix of the true outcome mechanism, X.
dim_z	An integer specifying the number of columns of the design matrix of the first-stage observation mechanism, Z.
dim_v	An integer specifying the number of columns of the design matrix of the second-stage observation mechanism, V.
n_cat	An integer specifying the number of categorical values that the true outcome, Y, the first-stage observed outcome, Y^*, and the second-stage observed outcome $\tilde{Y}$ can take.

Value

label_switch_2stage returns a named matrix of MCMC posterior samples for all parameters after performing label switching according the following pattern: all β terms are multiplied by -1, all γ and δ terms are "swapped" with the opposite j index.

loglik *Expected Complete Data Log-Likelihood Function for Estimation of the Misclassification Model*

Description

Expected Complete Data Log-Likelihood Function for Estimation of the Misclassification Model

Usage

```
loglik(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 $j = 1, \ldots,$ 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	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

loglik returns the negative value of the expected log-likelihood function, $Q = \sum_{i=1}^{N} \Big[\sum_{j=1}^{2} w_{ij} \log\{\pi_{ij}\} + \sum_{j=1}^{2} \sum_{k=1}^{2} w_{ij} y_{ik}^{*} \log\{\pi_{ikj}^{*}\} \Big]$, at the provided inputs.

`loglik_2stage` *Expected Complete Data Log-Likelihood Function for Estimation of the Two-Stage Misclassification Model*

Description

Expected Complete Data Log-Likelihood Function for Estimation of the Two-Stage Misclassification Model

Usage

```
loglik_2stage(
  param_current,
  obs_Ystar_matrix,
  obs_Ytilde_matrix,
  X,
  Z,
  V,
  sample_size,
  n_cat
)
```

Arguments

`param_current` A numeric vector of regression parameters, in the order β, γ, δ. 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 $j = 1, \ldots,$ `n_cat`. The numeric vector γ is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`. The δ vector is obtained from the array form. In array form, the first dimension (matrix rows) of `delta` corresponds to parameters for the $\tilde{Y} = 1$ second-stage observed outcome, with the dimensions of the `V` The second dimension (matrix columns) correspond to the first-stage observed outcome categories $Y^* \in \{1, 2\}$. The third dimension of `delta_start` corresponds to to the true outcome categories $Y \in \{1, 2\}$. The numeric vector δ is obtained by concatenating the delta array, i.e. `delta_vector <- c(delta_array)`.

`obs_Ystar_matrix`
A numeric matrix of indicator variables (0, 1) for the first-stage 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.

`obs_Ytilde_matrix`
A numeric matrix of indicator variables (0, 1) for the second-stage observed outcome $\tilde{Y}$. 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 first-stage observation mechanism.
V	A numeric design matrix for the second-stage 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 matrices, X, Z, and V.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcomes, Y* and $\tilde{Y}$, can take.

Value

loglik_2stage returns the negative value of the expected log-likelihood function, $Q = \sum_{i=1}^{N}\left[\sum_{j=1}^{2} w_{ij}\log\{\pi_{ij}\}+\sum_{j=1}^{2}\sum_{k=1}^{2} w_{ij}y_{ik}^{*}\log\{\pi_{ikj}^{*}\}+\sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{\ell=1}^{2} w_{ij}y_{ik}^{*}\tilde{y}_{i\ell}\log\{\tilde{\pi}_{i\ell kj}\}\right]$, at the provided inputs.

LSAC_data *Example data from The Law School Admissions Council's (LSAC) National Bar Passage Study (Linda Wightman, 1998)*

Description

Example data from The Law School Admissions Council's (LSAC) National Bar Passage Study (Linda Wightman, 1998)

Usage

```
LSAC_data
```

Format

A dataframe 39 columns, including background and demographic information, as well as if the candidates passed the bar exam to become lawyers in the USA.

Source

https://www.kaggle.com/datasets/danofer/law-school-admissions-bar-passage/data?select=bar_pass_prediction.csv

Examples

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

## End(Not run)
```

mean_pistarjj_compute *Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects*

Description

Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects

Usage

```
mean_pistarjj_compute(pistar_matrix, j, sample_size)
```

Arguments

pistar_matrix	A numeric matrix of conditional probabilities obtained from the internal function pistar_compute_for_chains. 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.
j	An integer value representing the true outcome category to compute the average conditional probability of correct classification for. j can take on values 1 and 2.
sample_size	An integer value specifying the number of observations in the sample.

Value

mean_pistarjj_compute returns a numeric value equal to the average conditional probability $P(Y^* = j|Y = j, Z)$ across all subjects.

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

Description

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

Usage

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

Arguments

gamma_matrix	A numeric matrix of estimated regression parameters for the observation 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_matrix. Columns of the matrix correspond to the true outcome categories $j = 1, \ldots,$ n_cat. The matrix should be obtained by COMBO_EM or COMBO_MCMC.
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, Y, represents a true, latent outcome category $Y \in \{1, 2\}$. The third column, Ystar, represents an observed outcome category $Y^* \in \{1, 2\}$. 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 outcome category, and true, latent outcome 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_Y <- misclassification_prob(estimated_gammas, z_matrix)
head(P_Ystar_Y)
```

---

misclassification_prob2

Compute Conditional Probability of Each Second-Stage Observed Outcome Given Each True Outcome and First-Stage Observed Outcome, for Every Subject

---

Description

Compute the conditional probability of observing second-stage outcome $Y^{*(2)} \in \{1, 2\}$ given the latent true outcome $Y \in \{1, 2\}$ and the first-stage outcome $Y^{*(1)} \in \{1, 2\}$ as $\frac{\exp\{\gamma^{(2)}_{\ell kj0}+\gamma^{(2)}_{\ell kjZ^{(2)}}Z^{(2)}\}}{1+\exp\{\gamma^{(2)}_{\ell kj0}+\gamma^{(2)}_{\ell kjZ^{(2)}}Z^{(2)}_i\}}$ for each of the $i = 1, \ldots, n$ subjects.

Usage

```
misclassification_prob2(gamma2_array, z2_matrix)
```

Arguments

gamma2_array	A numeric array of estimated regression parameters for the observation mechanism, $Y^{*(2)}\|Y^{*(1)}, Y$ (second-stage observed outcome, given the first-stage observed outcome and the true outcome) ~ $Z^{(2)}$ (second-stage misclassification predictor matrix). Rows of the array correspond to parameters for the $Y^{*(2)} = 1$ observed outcome, with the dimensions of z2_matrix. Columns of the array correspond to the first-stage outcome categories $k = 1, \ldots,$ n_cat. The third stage of the array corresponds to the true outcome categories $j = 1, \ldots,$ n_cat. The array should be obtained by COMBO_EM or COMBO_MCMC.
z2_matrix	A numeric matrix of covariates in the second-stage observation mechanism. z2_matrix should not contain an intercept.

Value

misclassification_prob2 returns a dataframe containing five 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 z2_matrix. The second column, Y, represents a true, latent outcome category $Y \in \{1, 2\}$. The third column, Ystar1, represents a first-stage observed outcome category $Y^{*(1)} \in \{1, 2\}$. The fourth column, Ystar2, represents a second-stage observed outcome category $Y^{*(2)} \in \{1, 2\}$. The last column, Probability, is the value of the equation $\frac{\exp\{\gamma^{(2)}_{\ell kj0} + \gamma^{(2)}_{\ell kj Z^{(2)}} Z^{(2)}\}}{1+\exp\{\gamma^{(2)}_{\ell kj0} + \gamma^{(2)}_{\ell kj Z^{(2)}} Z^{(2)}_i\}}$ computed for each subject, first-stage observed outcome category, second-stage observed outcome category, and true, latent outcome category.

Examples

```
set.seed(123)
sample_size <- 1000
cov1 <- rnorm(sample_size)
cov2 <- rnorm(sample_size, 1, 2)
z2_matrix <- matrix(c(cov1, cov2), nrow = sample_size, byrow = FALSE)
estimated_gamma2 <- array(c(1, -1, .5, .2, -.6, 1.5,
                            -1, .5, -1, -.5, -1, -.5), dim = c(3,2,2))
P_Ystar2_Ystar1_Y <- misclassification_prob2(estimated_gamma2, z2_matrix)
head(P_Ystar2_Ystar1_Y)
```

---

model_picker *Select a Binary Outcome Misclassification Model for a Given Prior*

Description

Select a Binary Outcome Misclassification Model for a Given Prior

Usage

```
model_picker(prior)
```

Arguments

prior	A character string specifying the prior distribution for the β and γ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

Value

model_picker returns a character string specifying the binary outcome misclassification model to be turned into a .BUG file and used for MCMC estimation with rjags.

---

model_picker_2stage *Select a Two-Stage Binary Outcome Misclassification Model for a Given Prior*

Description

Select a Two-Stage Binary Outcome Misclassification Model for a Given Prior

Usage

```
model_picker_2stage(prior)
```

Arguments

prior	A character string specifying the prior distribution for the β, γ, and δ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

Value

model_picker returns a character string specifying the two-stage binary outcome misclassification model to be turned into a .BUG file and used for MCMC estimation with rjags.

---

naive_jags_picker *Set up a Naive Logistic Regression* jags.model *Object for a Given Prior*

Description

Set up a Naive Logistic Regression jags.model Object for a Given Prior

Usage

```
naive_jags_picker(
  prior,
  sample_size,
  dim_x,
  n_cat,
  Ystar,
  X,
  beta_prior_parameters,
  number_MCMC_chains,
  naive_model_file,
  display_progress = TRUE
)
```

Arguments

prior	character string specifying the prior distribution for the naive β parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
sample_size	An integer value specifying the number of observations in the sample.
dim_x	An integer specifying the number of columns of the design matrix of the true outcome mechanism, X.
n_cat	An integer specifying the number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.
Ystar	A numeric vector of indicator variables (1, 2) for the observed outcome Y*. The reference category is 2.
X	A numeric design matrix for the true outcome mechanism.
beta_prior_parameters	A numeric list of prior distribution parameters for the β terms. For prior distributions "t", "uniform", "normal", or "dexp", the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions "t", "uniform", "normal", or "dexp", the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions dim_x X n_cat, and all elements in the n_cat column should be set to NA.
number_MCMC_chains	An integer specifying the number of MCMC chains to compute.
naive_model_file	A .BUG file and used for MCMC estimation with rjags.
display_progress	A logical value specifying whether messages should be displayed during model compilation. The default is TRUE.

Value

naive_jags_picker returns a jags.model object for a naive logistic regression model predicting the potentially misclassified Y* from the predictor matrix x. The object includes the specified prior distribution, model, number of chains, and data.

---

naive_jags_picker_2stage *Set up a Naive Two-Stage Regression* jags.model *Object for a Given Prior*

Description

Set up a Naive Two-Stage Regression jags.model Object for a Given Prior

Usage

```
naive_jags_picker_2stage(
  prior,
  sample_size,
  dim_x,
  dim_v,
  n_cat,
  Ystar,
  Ytilde,
  X,
  V,
  beta_prior_parameters,
  delta_prior_parameters,
  number_MCMC_chains,
  naive_model_file,
  display_progress = TRUE
)
```

Arguments

prior	character string specifying the prior distribution for the naive β parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
sample_size	An integer value specifying the number of observations in the sample.
dim_x	An integer specifying the number of columns of the design matrix of the first-stage outcome mechanism, X.
dim_v	An integer specifying the number of columns of the design matrix of the second-stage outcome mechanism, V.
n_cat	An integer specifying the number of categorical values that the observed outcomes can take.

`Ystar` A numeric vector of indicator variables (1, 2) for the first-stage observed outcome `Y*`. The reference category is 2.

`Ytilde` A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $\tilde{Y}$. The reference category is 2.

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

`V` A numeric design matrix for the second-stage outcome mechanism.

`beta_prior_parameters`
A numeric list of prior distribution parameters for the β terms. For prior distributions `"t"`, `"uniform"`, `"normal"`, or `"dexp"`, the first element of the list should contain a matrix of location, lower bound, mean, or shape parameters, respectively, for β terms. For prior distributions `"t"`, `"uniform"`, `"normal"`, or `"dexp"`, the second element of the list should contain a matrix of shape, upper bound, standard deviation, or scale parameters, respectively, for β terms. For prior distribution `"t"`, the third element of the list should contain a matrix of the degrees of freedom for β terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions `dim_x` X `n_cat`, and all elements in the `n_cat` column should be set to `NA`.

`delta_prior_parameters`
A numeric list of prior distribution parameters for the naive δ terms. For prior distributions `"t"`, `"uniform"`, `"normal"`, or `"dexp"`, the first element of the list should contain an array of location, lower bound, mean, or shape parameters, respectively, for δ terms. For prior distributions `"t"`, `"uniform"`, `"normal"`, or `"dexp"`, the second element of the list should contain an array of shape, upper bound, standard deviation, or scale parameters, respectively, for δ terms. For prior distribution `"t"`, the third element of the list should contain an array of the degrees of freedom for δ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions `n_cat` X `n_cat` X `dim_v`, and all elements in the `n_cat` row should be set to `NA`.

`number_MCMC_chains`
An integer specifying the number of MCMC chains to compute.

`naive_model_file`
A .BUG file and used for MCMC estimation with `rjags`.

`display_progress`
A logical value specifying whether messages should be displayed during model compilation. The default is `TRUE`.

Value

`naive_jags_picker_2stage` returns a `jags.model` object for a naive two-stage regression model predicting the potentially misclassified `Y*` from the predictor matrix `x` and the potentially misclassified $\tilde{Y}|Y^*$ from the predictor matrix `v`. The object includes the specified prior distribution, model, number of chains, and data.

`naive_loglik_2stage` *Observed Data Log-Likelihood Function for Estimation of the Naive Two-Stage Misclassification Model*

Description

Observed Data Log-Likelihood Function for Estimation of the Naive Two-Stage Misclassification Model

Usage

```
naive_loglik_2stage(
  param_current,
  X,
  V,
  obs_Ystar_matrix,
  obs_Ytilde_matrix,
  sample_size,
  n_cat
)
```

Arguments

`param_current`	A numeric vector of regression parameters, in the order β, δ. The δ vector is obtained from the matrix form. In matrix form, the gamma parameter matrix rows correspond to parameters for the $\tilde{Y} = 1$ observed outcome, with the dimensions of `V`. In matrix form, the gamma parameter matrix columns correspond to the true outcome categories $j = 1, \ldots,$ `n_cat`. The numeric vector `delta_v` is obtained by concatenating the delta matrix, i.e. `delta_v <- c(delta_matrix)`.
`X`	A numeric design matrix for the first-stage observed mechanism.
`V`	A numeric design matrix for the second-stage observed mechanism.
`obs_Ystar_matrix`	A numeric matrix of indicator variables (0, 1) for the first-stage 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.
`obs_Ytilde_matrix`	A numeric matrix of indicator variables (0, 1) for the second-stage observed outcome $\tilde{Y}$. 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.
`sample_size`	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 `V`.
`n_cat`	The number of categorical values that the first- and second-stage outcomes, Y^* and $\tilde{Y}$, can take.

Value

naive_loglik_2stage returns the negative value of the observed data log-likelihood function, $\sum_{i=1}^{N}\left[\sum_{k=1}^{2}\sum_{k=1}^{2}\sum_{\ell=1}^{2} y_{ik}^{*}\tilde{y}_{i}\log\{P(\tilde{Y}_i=\ell, Y_i^*=k|x_i,v_i)\}\right]$, at the provided inputs.

---

naive_model_picker *Select a Logisitic Regression Model for a Given Prior*

Description

Select a Logisitic Regression Model for a Given Prior

Usage

```
naive_model_picker(prior)
```

Arguments

prior	A character string specifying the prior distribution for the naive β parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

Value

naive_model_picker returns a character string specifying the logistic regression model to be turned into a .BUG file and used for MCMC estimation with rjags.

---

naive_model_picker_2stage *Select a Naive Two-Stage Regression Model for a Given Prior*

Description

Select a Naive Two-Stage Regression Model for a Given Prior

Usage

```
naive_model_picker_2stage(prior)
```

Arguments

prior	A character string specifying the prior distribution for the naive β parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

Value

naive_model_picker_2stage returns a character string specifying the logistic regression model to be turned into a .BUG file and used for MCMC estimation with rjags.

perfect_sensitivity_EM *EM-Algorithm Estimation of the Binary Outcome Misclassification Model while Assuming Perfect Sensitivity*

Description

Code is adapted by the SAMBA R package from Lauren Beesley and Bhramar Mukherjee.

Usage

```
perfect_sensitivity_EM(
  Ystar,
  Z,
  X,
  start,
  beta0_fixed = NULL,
  weights = NULL,
  expected = TRUE,
  tolerance = 1e-07,
  max_em_iterations = 1500
)
```

Arguments

Ystar	A numeric vector of indicator variables (1, 0) for the observed outcome Y*. The reference category is 0.
Z	A numeric matrix of covariates in the true outcome mechanism. Z should not contain an intercept.
X	A numeric matrix of covariates in the observation mechanism. X should not contain an intercept.
start	Numeric vector of starting values for parameters in the true outcome mechanism (θ) and the observation mechanism (β), respectively.
beta0_fixed	Optional numeric vector of values of the observation mechanism intercept to profile over. If a single value is entered, this corresponds to fixing the intercept at the specified value. The default is NULL.
weights	Optional vector of row-specific weights used for selection bias adjustment. The default is NULL.
expected	A logical value indicating whether or not to calculate the covariance matrix via the expected Fisher information matrix. The default is TRUE.
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.

Value

perfect_sensitivity_EM returns a list containing nine elements. The elements are detailed in ?SAMBA::obsloglikEM documentation. Code is adapted from the SAMBA::obsloglikEM function.

References

Beesley, L. and Mukherjee, B. (2020). Statistical inference for association studies using electronic health records: Handling both selection bias and outcome misclassification. Biometrics, 78, 214-226.

---

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.

pistar_by_chain *Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects for each MCMC Chain*

Description

Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects for each MCMC Chain

Usage

```
pistar_by_chain(n_chains, chains_list, Z, n, n_cat)
```

Arguments

n_chains	An integer specifying the number of MCMC chains to compute over.
chains_list	A numeric list containing the samples from n_chains MCMC chains.
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_by_chain returns a numeric matrix of the average conditional probability $P(Y^* = j|Y = j, Z)$ across all subjects for each MCMC chain. Rows of the matrix correspond to MCMC chains, up to n_chains. The first column contains the conditional probability $P(Y^* = 1|Y = 1, Z)$. The second column contains the conditional probability $P(Y^* = 2|Y = 2, Z)$.

pistar_by_chain_2stage *Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects for each MCMC Chain for a 2-stage model*

Description

Compute the Mean Conditional Probability of Correct Classification, by True Outcome Across all Subjects for each MCMC Chain for a 2-stage model

Usage

```
pistar_by_chain_2stage(n_chains, chains_list, Z, n, n_cat)
```

Arguments

n_chains	An integer specifying the number of MCMC chains to compute over.
chains_list	A numeric list containing the samples from n_chains MCMC chains.
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_by_chain returns a numeric matrix of the average conditional probability $P(Y^* = j|Y = j, Z)$ across all subjects for each MCMC chain. Rows of the matrix correspond to MCMC chains, up to n_chains. The first column contains the conditional probability $P(Y^* = 1|Y = 1, Z)$. The second column contains the conditional probability $P(Y^* = 2|Y = 2, Z)$.

---

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 $j = 1, \ldots,$ 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 $i = 1, \ldots,$ 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 $j = 1, \ldots,$ n_cat.

---

pistar_compute_for_chains

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject

---

Description

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject

Usage

```
pistar_compute_for_chains(chain_colMeans, Z, n, n_cat)
```

Arguments

chain_colMeans	A numeric vector containing the posterior means for all sampled parameters in a given MCMC chain. chain_colMeans must be a named object (i.e. each parameter must be named as gamma[k,j,p]).
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_for_chains 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 $i = 1, \ldots,$ n subjects. Rows of the matrix correspond to each subject and observed outcome. Specifically, the probability for subject i and observed category \$0\$ occurs at row i. The probability for subject i and observed category \$1\$ occurs at row $i+$ n. Columns of the matrix correspond to the true outcome categories $j = 1, \ldots,$ n_cat.

pistar_compute_for_chains_2stage *Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject for 2-stage models*

Description

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject for 2-stage models

Usage

```
pistar_compute_for_chains_2stage(chain_colMeans, Z, n, n_cat)
```

Arguments

chain_colMeans	A numeric vector containing the posterior means for all sampled parameters in a given MCMC chain. chain_colMeans must be a named object (i.e. each parameter must be named as gamma[k,j,p]).
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_for_chains 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 $i = 1, \ldots,$ n subjects. Rows of the matrix correspond to each subject and observed outcome. Specifically, the probability for subject i and observed category \$0\$ occurs at row i. The probability for subject i and observed category \$1\$ occurs at row $i+$ n. Columns of the matrix correspond to the true outcome categories $j = 1, \ldots,$ n_cat.

pitilde_by_chain *Compute the Mean Conditional Probability of Second-Stage Correct Classification, by First-Stage and True Outcome Across all Subjects for each MCMC Chain*

Description

Compute the Mean Conditional Probability of Second-Stage Correct Classification, by First-Stage and True Outcome Across all Subjects for each MCMC Chain

Usage

```
pitilde_by_chain(n_chains, chains_list, V, n, n_cat)
```

Arguments

n_chains	An integer specifying the number of MCMC chains to compute over.
chains_list	A numeric list containing the samples from n_chains MCMC chains.
V	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, V.
n_cat	The number of categorical values that the true outcome, Y, the first-stage observed outcome, $Y*$, and the second-stage observed outcome, $\tilde{Y}$, can take.

Value

pitilde_by_chain returns a numeric matrix of the average conditional probability $P(\tilde{Y} = j|Y^* = j, Y = j, V)$ across all subjects for each MCMC chain. Rows of the matrix correspond to MCMC chains, up to n_chains. The first column contains the conditional probability $P(\tilde{Y} = 1|Y^* = 1, Y = 1, V)$. The second column contains the conditional probability $P(\tilde{Y} = 2|Y^* = 2, Y = 2, V)$.

---

pitilde_compute *Compute Conditional Probability of Each Second-Stage Observed Outcome Given Each True Outcome and First-Stage Observed Outcome, for Every Subject*

Description

Compute Conditional Probability of Each Second-Stage Observed Outcome Given Each True Outcome and First-Stage Observed Outcome, for Every Subject

Usage

```
pitilde_compute(delta, V, n, n_cat)
```

Arguments

delta	A numeric array of regression parameters for the second-stage observed outcome mechanism, $\tilde{Y}|Y^*, Y$ (second-stage observed outcome, given the first-stage observed outcome and the true outcome) ~ V (misclassification predictor matrix). Rows of the matrix correspond to parameters for the $\tilde{Y} = 1$ observed outcome, with the dimensions of V. Columns of the matrix correspond to the first-stage observed outcome categories $k = 1, \ldots,$ n_cat. The third dimension of the array corresponds to the true outcome categories $j = 1, \ldots,$ n_cat
V	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, V.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcomes can take.

Value

pitilde_compute returns an array of conditional probabilities, $P(\tilde{Y}_i = \ell | Y_i^* = k, Y_i = j, V_i) = \frac{\exp\{\delta_{\ell kj0} + \delta_{\ell kjV} V_i\}}{1+\exp\{\delta_{\ell kj0} + \delta_{\ell kjV} V_i\}}$ for each of the $i = 1, \ldots,$ n subjects. Rows of the matrix correspond to each subject and second-stage 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 first-stage outcome categories, $k = 1, \ldots,$ n_cat. The third dimension of the array corresponds to the true outcome categories, $j = 1, \ldots,$ n_cat.

---

pitilde_compute_for_chains

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject

---

Description

Compute Conditional Probability of Each Observed Outcome Given Each True Outcome for a given MCMC Chain, for Every Subject

Usage

```
pitilde_compute_for_chains(chain_colMeans, V, n, n_cat)
```

Arguments

chain_colMeans	A numeric vector containing the posterior means for all sampled parameters in a given MCMC chain. chain_colMeans must be a named object (i.e. each parameter must be named as delta[l,k,j,p]).
V	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, V.
n_cat	The number of categorical values that the true outcome, Y, the first-stage observed outcome, Y^*, and the second-stage observed outcome, $\tilde{Y}$,\ can take.

Value

pitilde_compute_for_chains returns a matrix of conditional probabilities, $P(\tilde{Y}_i = \ell | Y_i^* = k, Y_i = j, V_i) = \frac{\exp\{\delta_{\ell kj0} + \delta_{\ell kjV} V_i\}}{1+\exp\{\delta_{\ell kj0} + \delta_{\ell kjV} V_i\}}$ corresponding to each subject and observed outcome. Specifically, the probability for subject i and second-stage observed category \$1\$ occurs at row i. The probability for subject i and second-stage observed category \$2\$ occurs at row $i+$ n. Columns of the matrix correspond to the first-stage outcome categories $j = 1, \ldots,$ n_cat. The third dimension of the array corresponds to the true outcome categories, $j = 1, \ldots,$ n_cat.

q_beta_f *M-Step Expected Log-Likelihood with respect to Beta*

Description

Objective function of the form: $Q_\beta = \sum_{i=1}^{N}\left[\sum_{j=0}^{1} w_{ij}\log\{\pi_{ij}\}\right]$. Used to obtain estimates of β parameters.

Usage

```
q_beta_f(beta, X, w_mat, sample_size, n_cat)
```

Arguments

beta	A numeric vector of regression parameters for the Y (true outcome) ~ X (predictor matrix of interest).
X	A numeric design matrix.
w_mat	Matrix of E-step weights obtained from w_j.
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.
n_cat	The number of categorical values that the true outcome, Y, can take.

Value

q_beta_f returns the negative value of the expected log-likelihood function, $Q_\beta = \sum_{i=1}^{N}\left[\sum_{j=1}^{2} w_{ij}\log\{\pi_{ij}\}\right]$, at the provided inputs.

q_delta_f *M-Step Expected Log-Likelihood with respect to Delta*

Description

Objective function of the form: $Q_\delta = \sum_{i=1}^{N}\left[\sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{\ell=1}^{2} w_{ij}y^{*}_{ik}\tilde{y}_{i\ell}\log\{\tilde{\pi}_{i\ell kj}\}\right]$. Used to obtain estimates of δ parameters.

Usage

```
q_delta_f(
  delta_v,
  V,
  obs_Ystar_matrix,
  obs_Ytilde_matrix,
  w_mat,
  sample_size,
  n_cat
)
```

Arguments

delta_v	A numeric array of regression parameters for the second-stage observed outcome mechanism, $\tilde{Y}\|Y^*, Y$ (second-stage observed outcome, given the first-stage observed outcome and the true outcome) ~ V (misclassification predictor matrix). The δ vector is obtained from the array form. In array form, the first dimension (matrix rows) of delta corresponds to parameters for the $\tilde{Y} = 1$ second-stage observed outcome, with the dimensions of the V The second dimension (matrix columns) correspond to the first-stage observed outcome categories $Y^* \in \{1, 2\}$. The third dimension of delta_start corresponds to to the true outcome categories $Y \in \{1, 2\}$. The numeric vector δ is obtained by concatenating the delta array, i.e. delta_v <- c(delta_array).
V	A numeric design matrix.
obs_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.
obs_Ytilde_matrix	A numeric matrix of indicator variables (0, 1) for the observed outcome $\tilde{Y}$. 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.
w_mat	Matrix of E-step weights obtained from w_j_2stage.
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, V.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcomes can take.

Value

q_beta_f returns the negative value of the expected log-likelihood function, $Q_\delta = \sum_{i=1}^{N}\Big[\sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{\ell=1}^{2} w_{ij}y^*_{ik}\tilde{y}_{i\ell}\log\{\hat{\pi}$ at the provided inputs.

---

q_gamma_f *M-Step Expected Log-Likelihood with respect to Gamma*

Description

Objective function of the form: $Q_\gamma = \sum_{i=1}^{N}\Big[\sum_{j=1}^{2}\sum_{k=1}^{2} w_{ij}y^*_{ik}\log\{\pi^*_{ikj}\}\Big]$. Used to obtain estimates of γ parameters.

Usage

```
q_gamma_f(gamma_v, Z, obs_Y_matrix, w_mat, sample_size, n_cat)
```

Arguments

gamma_v	A numeric vector of regression parameters for the observed outcome mechanism, Y* | Y (observed outcome, given the true outcome) ~ Z (misclassification predictor matrix). In matrix form, the gamma parameter matrix rows correspond to parameters for the Y* = 0 observed outcome, with the dimensions of Z. In matrix form, the gamma parameter matrix columns correspond to the true outcome 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).
Z	A numeric design 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.
w_mat	Matrix of E-step weights obtained from w_j.
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, Z.
n_cat	The number of categorical values that the true outcome, Y, and the observed outcome, Y* can take.

Value

q_beta_f returns the negative value of the expected log-likelihood function, $Q_\gamma = \sum_{i=1}^{N}\left[\sum_{j=1}^{2}\sum_{k=1}^{2} w_{ij} y_{ik}^{*} \log\{\pi_{ikj}^{*}\}\right]$, at the provided inputs.

---

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 nth element of the vector x, plus 1.

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

Description

Compute the probability of the latent true outcome $Y \in \{1, 2\}$ as $P(Y_i = j|X_i) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$ for each of the $i = 1, \ldots,$ n subjects.

Usage

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

Arguments

beta_matrix	A numeric column matrix of estimated regression parameters for the true outcome mechanism, Y (true outcome) ~ X (predictor matrix of interest), obtained from COMBO_EM or COMBO_MCMC.
x_matrix	A numeric matrix of covariates in the true outcome 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, Y, represents a true, latent outcome category $Y \in \{1, 2\}$. The last column, Probability, is the value of the equation $P(Y_i = j|X_i) = \frac{\exp(X_i\beta)}{1+\exp(X_i\beta)}$ computed for each subject and true, latent outcome 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_Y <- true_classification_prob(estimated_betas, x_matrix)
head(P_Y)
```

VPRAI_synthetic_data *Synthetic example data of pretrial failure risk factors and outcomes, VPRAI recommendations, and judge decisions*

Description

Synthetic example data of pretrial failure risk factors and outcomes, VPRAI recommendations, and judge decisions

Usage

```
VPRAI_synthetic_data
```

Format

A dataframe 1990 columns, including defendant race, risk factors, VPRAI recommendations, judge decisions, and pretrial failure outcomes.

Examples

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

## End(Not run)
```

w_j *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

```
w_j(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

w_j 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 $j = 1, \ldots,$ n_cat.

w_j_2stage *Compute E-step for Two-Stage Binary Outcome Misclassification Model Estimated With the EM-Algorithm*

Description

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

Usage

```
w_j_2stage(
  ystar_matrix,
  ytilde_matrix,
  pitilde_array,
  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.
ytilde_matrix	A numeric matrix of indicator variables (0, 1) for the observed outcome $\tilde{Y}$. 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.
pitilde_array	A numeric array of conditional probabilities obtained from the internal function pitilde_compute. Rows of the matrices correspond to each subject and to each second-stage observed outcome category. Columns of the matrix correspond to each first-stage observed outcome category. The third dimension of the array corresponds to each true, latent outcome category.
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 first-stage 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 matrices, ystar_matrix and ytilde_matrix.

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

Value

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

Index

∗ **datasets**
COMBO_EM_data, 13
LSAC_data, 33
VPRAI_synthetic_data, 55

check_and_fix_chains, 3
check_and_fix_chains_2stage, 4
COMBO_data, 5
COMBO_data_2stage, 7
COMBO_EM, 9
COMBO_EM_2stage, 11
COMBO_EM_data, 13
COMBO_MCMC, 14
COMBO_MCMC_2stage, 17

em_function, 23
em_function_2stage, 24
expit, 25

jags_picker, 25
jags_picker_2stage, 27

label_switch, 29
label_switch_2stage, 30
loglik, 31
loglik_2stage, 32
LSAC_data, 33

mean_pistarjj_compute, 34
misclassification_prob, 34
misclassification_prob2, 35
model_picker, 36
model_picker_2stage, 37

naive_jags_picker, 37
naive_jags_picker_2stage, 39
naive_loglik_2stage, 41
naive_model_picker, 42
naive_model_picker_2stage, 42

perfect_sensitivity_EM, 43
pi_compute, 44
pistar_by_chain, 45
pistar_by_chain_2stage, 45
pistar_compute, 46
pistar_compute_for_chains, 47
pistar_compute_for_chains_2stage, 48
pitilde_by_chain, 48
pitilde_compute, 49
pitilde_compute_for_chains, 50

q_beta_f, 51
q_delta_f, 51
q_gamma_f, 52

sum_every_n, 53
sum_every_n1, 54

true_classification_prob, 54

VPRAI_synthetic_data, 55

w_j, 56
w_j_2stage, 57