Package ‘COMBO’

April 19, 2023

Title  Correcting Misclassified Binary Outcomes in Association Studies

Version  1.0.0

Author  Kimberly Hochstedler [aut, cre]

Maintainer  Kimberly Hochstedler <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 correct outcome classification occurs in at least 50% of observations. A description of the analysis methods is available in Hochstedler and Wells (2023) <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), kableExtra (>= 1.3.4)

License  MIT + file LICENSE

Encoding  UTF-8

RoxygenNote  7.2.1

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' 'pitiilde_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'
         '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'
         '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'
R topics documented:

'naive_loglik_2stage.R' 'naive_model_picker_2stage.R'
'pitilde_by_chain.R' 'pitilde_compute_for_chains.R'
'true_classification_prob.R'

NeedsCompilation no
Repository CRAN
Date/Publication 2023-04-19 13:50:06 UTC

R topics documented:

check_and_fix_chains ................................................. 3
check_and_fix_chains_2stage ......................................... 4
COMBO_data .......................................................... 5
COMBO_data_2stage .................................................... 6
COMBO_EM .............................................................. 8
COMBO_EM_2stage ...................................................... 10
COMBO_MCMC .......................................................... 13
COMBO_MCMC_2stage .................................................... 16
em_function ............................................................ 21
em_function_2stage .................................................... 22
expit ................................................................. 23
jags_picker ............................................................ 24
jags_picker_2stage ..................................................... 25
label_switch ........................................................... 28
label_switch_2stage .................................................... 28
loglik ................................................................. 29
loglik_2stage ........................................................... 30
mean_pistarjj_compute ............................................... 31
misclassification_prob ................................................ 32
misclassification_prob2 ............................................... 33
model_picker ........................................................... 34
model_picker_2stage ................................................... 35
naive_jags_picker ..................................................... 35
naive_jags_picker_2stage ............................................... 36
naive_loglik ............................................................ 38
naive_loglik_2stage .................................................... 38
naive_model_picker .................................................... 39
naive_model_picker_2stage ............................................ 40
perfect_sensitivity_EM ............................................... 40
pistar_by_chain ....................................................... 42
pistar_compute ......................................................... 42
pistar_compute_for_chains ........................................... 43
pitilde_by_chain ....................................................... 44
pitilde_compute ........................................................ 45
pitilde_compute_for_chains .......................................... 46
pi_compute ............................................................ 46
q_beta_f ............................................................... 47
q_delta_f ............................................................... 48
q_gamma_f .............................................................. 49
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 \( P(Y^* = j|Y = j, Z) \) 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: \( P(Y^* = j|Y = j, Z) > 0.50 \forall j \).
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,
  pistarjjj_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.
pistarjjj_matrix A numeric matrix of the average conditional probability \( P(Y^* = j | Y = j, Z) \) across all subjects for each MCMC chain, obtained from the pistar_by_chain function.
pitildejjj_matrix A numeric matrix of the average conditional probability \( P(\tilde{Y} = j | Y^* = j, Y = j, V) \) 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: \( P(Y^* = j | Y = j, Z) > 0.50 \forall j \).

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 \( \beta \) parameter values (intercept, slope) to generate data under in the true outcome mechanism.
- **gamma**: A numeric matrix of \( \gamma \) 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.
\texttt{COMBO\_data\_2stage}  

\begin{verbatim}
   z            A vector of generated predictor values in the observation mechanism from the Gamma distribution.
\end{verbatim}

\begin{verbatim}
\texttt{x\_design\_matrix}
   The design matrix for the x predictor.
\end{verbatim}

\begin{verbatim}
\texttt{z\_design\_matrix}
   The design matrix for the z predictor.
\end{verbatim}

\noindent \textbf{Examples}

\begin{verbatim}
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"])
\end{verbatim}

\section*{COMBO\_data\_2stage  \hspace{1cm} \textit{Generate Data to use in two-stage COMBO Functions}}

\noindent \textbf{Description}

Generate Data to use in two-stage COMBO Functions

\noindent \textbf{Usage}

\begin{verbatim}
COMBO\_data\_2stage(
   sample\_size,
   x\_mu,
   x\_sigma,
   z\_shape,
   v\_shape,
   beta,
   gamma,
   delta
)
\end{verbatim}
COMBO_data_2stage

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.
- **v_shape**: A positive numeric value specifying the shape parameter of $v$ predictors generated from a Gamma distribution.
- **beta**: A column matrix of $\beta$ parameter values (intercept, slope) to generate data under in the true outcome mechanism.
- **gamma**: A numeric matrix of $\gamma$ parameters to generate data under in the first-stage 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 $Y \in \{1, 2\}$.
- **delta**: A numeric array of $\delta$ parameters to generate data under the second-stage observation mechanism. In array form, the delta 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 delta array is indexed by the true outcome categories.

Value

COMBO_data_2stage returns a list of generated data elements:

- **obs_Ystar**: A vector of first-stage observed outcomes.
- **obs_Ytilde**: A vector of second-stage observed outcomes.
- **true_Y**: A vector of true outcomes.
- **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 contains 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 $Y^*$. 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 first-stage observation mechanism from the Gamma distribution.
- **v**: 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.

z_design_matrix
The design matrix for the z predictor.

v_design_matrix
The design matrix for the v predictor.

Examples

```r
set.seed(123)
n <- 500
x_mu <- 0
x_sigma <- 1
z_shape <- 1
v_shape <- 1
t_true_beta <- matrix(c(1, -2), ncol = 1)
t_true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
t_true_delta <- 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,
    z_shape = z_shape, v_shape = v_shape,
    beta = true_beta, gamma = true_gamma, delta = true_delta)
table(my_data[["obs_Ytilde"]], my_data[["obs_Ystar"]], 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**

```r
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 \( \beta \) 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 \( \gamma \) parameters in the obser-
vation 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 out-
come 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 sensitivity. The "Naive" parameter estimates are from a simple logistic regression Y* ~ X.

References


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
**Description**

Jointly estimate $\beta$, $\gamma$, $\delta$ parameters from the true outcome, first-stage observation, and second-stage observation mechanisms, respectively, in a two-stage binary outcome misclassification model.

**Usage**

```r
COMBO_EM_2stage(
  Ystar,  # A numeric vector of indicator variables (1, 2) for the first-stage observed outcome $Y^*$. There should be no NA terms. The reference category is 2.
  Ytilde,  # A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $\tilde{Y}$. 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 first-stage observation mechanism. z_matrix should not contain an intercept and no values should be NA.
  v_matrix,  # A numeric matrix of covariates in the second-stage observation mechanism. v_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 $\beta$ 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 $\gamma$ parameters in the first-stage observation mechanism. In matrix form, the gamma_start matrix rows correspond to parameters for the $Y^* = 1$ first-stage observed outcome, with the dimensions of z_matrix plus 1, and the gamma parameter matrix columns correspond to the true outcome categories $\tilde{Y} \in \{1, 2\}$. A numeric vector for gamma_start is obtained by concatenating the gamma matrix, i.e. gamma_start <- c(gamma_matrix).
  delta_start,  # Additional arguments.
  tolerance = 1e-07,  # Additional arguments.
  max_em_iterations = 1500,  # Additional arguments.
  em_method = "squarem"  # Additional arguments.
)
```

**Arguments**

- **Ystar**: A numeric vector of indicator variables (1, 2) for the first-stage observed outcome $Y^*$. There should be no NA terms. The reference category is 2.
- **Ytilde**: A numeric vector of indicator variables (1, 2) for the second-stage observed outcome $\tilde{Y}$. 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 first-stage observation mechanism. z_matrix should not contain an intercept and no values should be NA.
- **v_matrix**: A numeric matrix of covariates in the second-stage observation mechanism. v_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 $\beta$ 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 $\gamma$ parameters in the first-stage observation mechanism. In matrix form, the gamma_start matrix rows correspond to parameters for the $Y^* = 1$ first-stage observed outcome, with the dimensions of z_matrix plus 1, and the gamma parameter matrix columns correspond to the true outcome categories $\tilde{Y} \in \{1, 2\}$. A numeric vector for gamma_start is obtained by concatenating the gamma matrix, i.e. gamma_start <- c(gamma_matrix).
COMBO_EM_2stage

delta_start A numeric array of starting values for the $\delta$ parameters in the second-stage observation mechanism. The first dimension (matrix rows) of delta_start correspond to parameters for the $Y = 1$ second-stage observed outcome, with the dimensions of the $v_{\text{matrix}}$ plus 1. 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 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

```r
set.seed(123)
n <- 1000
x_mu <- 0
x_sigma <- 1
z_shape <- 1
v_shape <- 1
true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
true_delta <- 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,
                           z_shape = z_shape, v_shape = v_shape,
                           beta = true_beta, gamma = true_gamma, delta = true_delta)
table(my_data[["obs_Ytilde"]], my_data[["obs_Ystar"]], my_data[["true_Y"]])

beta_start <- rnorm(length(c(true_beta)))
gamma_start <- rnorm(length(c(true_gamma)))
delta_start <- rnorm(length(c(true_delta)))

EM_results <- COMBO_EM_2stage(Ystar = my_data[["obs_Ystar"]],
                            Ytilde = my_data[["obs_Ytilde"]],
                            x_matrix = my_data[["x"]])
```
z_matrix = my_data["z"],
v_matrix = my_data["v"],
beta_start = beta_start,
gamma_start = gamma_start,
delta_start = delta_start)

EM_results

---

**COMBO_MCMC**  
*MCMC Estimation of the Binary Outcome Misclassification Model*

---

**Description**

Jointly estimate $\beta$ and $\gamma$ parameters from the true outcome and observation mechanisms, respectively, in a binary outcome misclassification model.

**Usage**

```r
COMBO_MCMC(
    Ystar,
    x,
    z,
    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`  
  A numeric matrix of covariates in the true outcome mechanism. $x$ should not contain an intercept.

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

- `prior`  
  A character string specifying the prior distribution for the $\beta$ and $\gamma$ parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).

- `beta_prior_parameters`  
  A numeric list of prior distribution parameters for the $\beta$ 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 \( \beta \) 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 \( \beta \) terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for \( \beta \) terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions \( n_{\text{cat}} \times \text{dim}_x \), and all elements in the \( n_{\text{cat}} \) row should be set to NA.

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 \( \gamma \) 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 \( \gamma \) terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for \( \gamma \) terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_z \), and all elements in the \( n_{\text{cat}} \) row should be set to NA.

gamma_prior_parameters
A numeric list of prior distribution parameters for the \( \gamma \) 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 \( \gamma \) 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 \( \gamma \) terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for \( \gamma \) terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_z \), and all elements in the \( n_{\text{cat}} \) 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. \( \beta \) terms have dimensions \( \text{dim}_x \times n_{\text{cat}} \). The \( \gamma \) terms have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_z \), 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 $\beta$ 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

```r
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)
ipi_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)
```

COMBO_MCMC_2stage

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 $\beta$, $\gamma$, and $\delta$ 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(
  Ystar,
  Ytilde,
  x,
  z,
  v,
  prior,
  beta_prior_parameters,
  gamma_prior_parameters,
  delta_prior_parameters,
  naive_delta_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.

Ytilde
A numeric vector of indicator variables (1, 2) for the second-stage observed outcome \( \tilde{Y} \). There should be no NA terms. The reference category is 2.

x
A numeric matrix of covariates in the true outcome mechanism. \( x \) should not contain an intercept.

z
A numeric matrix of covariates in the observation mechanism. \( z \) should not contain an intercept.

v
A numeric matrix of covariates in the second-stage observation mechanism. \( v \) should not contain an intercept and no values should be NA.

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

beta_prior_parameters
A numeric list of prior distribution parameters for the \( \beta \) 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 \( \beta \) 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 \( \beta \) terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for \( \beta \) terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions \( n_{\text{cat}} \times \text{dim}_x \), and all elements in the \( n_{\text{cat}} \) row should be set to NA.

gamma_prior_parameters
A numeric list of prior distribution parameters for the \( \gamma \) 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 \( \gamma \) 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 \( \gamma \) terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for \( \gamma \) terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_z \), and all elements in the \( n_{\text{cat}} \) row should be set to NA.

delta_prior_parameters
A numeric list of prior distribution parameters for the \( \delta \) 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 \( \delta \) 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 \( \delta \) terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for \( \delta \) terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_v \), and all elements in the \( n_{\text{cat}} \) row should be set to NA.
naive_delta_prior_parameters
A numeric list of prior distribution parameters for the naive model 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 naive 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 naive δ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for naive 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. Note that prior distributions for the naive β terms are inherited 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 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

# 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
z_shape <- 1
v_shape <- 1
true_beta <- matrix(c(1, -2), ncol = 1)
true_gamma <- matrix(c(.5, 1, -.5, -1), nrow = 2, byrow = FALSE)
true_delta <- 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)
z_matrix = matrix(rgamma(n, z_shape), ncol = 1)
Z = matrix(c(rep(1, n), z_matrix[,1]), ncol = 2, byrow = FALSE)
v_matrix = matrix(rgamma(n, v_shape), ncol = 1)
V = matrix(c(rep(1, n), v_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

exp_vd1 = exp(V %*% true_delta[,1])
exp_vd2 = exp(V %*% true_delta[,2])

pi_denominator1 = apply(exp_vd1, FUN = sum_every_n1, n, MARGIN = 2)
pi_result1 = exp_vd1 / rbind(pi_denominator1)

pi_denominator2 = apply(exp_vd2, FUN = sum_every_n1, n, MARGIN = 2)
pi_result2 = exp_vd2 / rbind(pi_denominator2)

pitilde_matrix1 = rbind(pi_result1,
                         1 - apply(pi_result1, FUN = sum_every_n, n = n, MARGIN = 2))

pitilde_matrix2 = rbind(pi_result2,
                         1 - apply(pi_result2, FUN = sum_every_n, n = n, MARGIN = 2))

pitilde_array = array(c(pitilde_matrix1, pitilde_matrix2),
                       dim = c(dim(pitilde_matrix1), 2))

obs_Ytilde <- rep(NA, n)
for(i in 1:n){
  true_j = true_Y[i]
  obs_k = Ystar[i]
  obs_Ytilde[i] = which(rmultinom(1, 1, pitilde_array[c(i, n + i),
                                                       obs_k, true_j]) == 1)
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 $\beta, \gamma$. The $\gamma$ 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 $j = 1, \ldots, n_{\text{cat}}$. 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.

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 $\beta, \gamma, \delta$. The $\gamma$ 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 $j = 1, \ldots, n_{\text{cat}}$. The numeric vector $\gamma$ is obtained by concatenating the gamma matrix, i.e. $\text{gamma}_v <- \text{c(gamma_matrix)}$. The $\delta$ vector is obtained from the array form. In array form, the first dimension (matrix rows) of delta corresponds to parameters for the $\hat{Y} = 1$ second-stage observed outcome, with the dimensions of the $\delta$. The second dimension (matrix columns) correspond to the first-stage observed outcome categories $Y^* \in \{1, 2\}$. The third dimension of $\delta$ corresponds to the true outcome categories $Y \in \{1, 2\}$. The numeric vector $\delta$ is obtained by concatenating the delta array, i.e. $\text{delta_vector} <- \text{c(delta_array)}$.

- **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 $\hat{Y}$. 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 $\hat{Y}$, 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

\texttt{expit(x)}

Arguments

\texttt{x} A numeric value or vector to compute the expit function on.

Value

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

\begin{verbatim}
  jags_picker Set up a Binary Outcome Misclassification jags.model Object for a
  Given Prior
\end{verbatim}

Description

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

Usage

\begin{verbatim}
  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
  )
\end{verbatim}

Arguments

\begin{itemize}
  \item \texttt{prior} A character string specifying the prior distribution for the \( \beta \) and \( \gamma \) parameters. Options are "t", "uniform", "normal", or "dexp" (double Exponential, or Weibull).
  \item \texttt{sample_size} An integer value specifying the number of observations in the sample.
  \item \texttt{dim_x} An integer specifying the number of columns of the design matrix of the true outcome mechanism, \( X \).
  \item \texttt{dim_z} An integer specifying the number of columns of the design matrix of the observation mechanism, \( Z \).
\end{itemize}
**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 $\beta$ 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 $\beta$ 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 $\beta$ terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for $\beta$ terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions $\dim_x \times n_{cat}$, 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 $\gamma$ 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 $\gamma$ 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 $\gamma$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for $\gamma$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions $n_{cat} \times n_{cat} \times \dim_z$, 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 $\beta$, $\gamma$, and $\delta$ 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.
A numeric design matrix for the second-stage observation mechanism.

beta_prior_parameters
A numeric list of prior distribution parameters for the $\beta$ 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 $\beta$ 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 $\beta$ terms. For prior distribution "t", the third element of the list should contain a matrix of the degrees of freedom for $\beta$ terms. The third list element should be empty for all other prior distributions. All matrices in the list should have dimensions $\dim_x \times n_cat$, 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 $\gamma$ 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 $\gamma$ 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 $\gamma$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for $\gamma$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions $n_cat \times n_cat \times \dim_z$, 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 $\delta$ 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 $\delta$ 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 $\delta$ terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for $\delta$ terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions $n_cat \times n_cat \times n_cat \times \dim_v$, 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**

```r
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 $\beta$ terms are multiplied by -1, all $\gamma$ 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**

```r
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 \( \beta \) terms are multiplied by -1, all \( \gamma \) and \( \delta \) 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 \( \beta, \gamma \). The \( \gamma \) 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. \( \text{gamma_v} \leftarrow c(\text{gamma_matrix}) \).

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.
loglik_2stage

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, $Q = \sum_{i=1}^{N} \left[ \sum_{j=1}^{2} w_{ij} \log\{\pi_{ij}\} + \sum_{j=1}^{2} \sum_{k=1}^{2} w_{ij} y_{ik}^{*} \log\{\pi_{ikj}^{*}\} \right]$, 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 $\beta, \gamma, \delta$. The $\gamma$ 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 $j = 1, \ldots, n_{\text{cat}}$. The numeric vector $\gamma$ is obtained by concatenating the gamma matrix, i.e. `gamma_v <- c(gamma_matrix)`. The $\delta$ vector is obtained from the array form. In array form, the first dimension (matrix rows) of delta corresponds to parameters for the $Y^* = 1$ 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 \( \delta \) is obtained by concatenating the delta array, i.e. \( \text{delta_vector} <- \text{c(delta_array)} \).

**obs_Ystar_matrix**

A numeric matrix of indicator variables (0, 1) for the first-stage observed outcome \( Y^\ast \). 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, \( Y^\ast \) and \( \tilde{Y} \), can take.

**Value**

\[
\text{loglik}_2\text{stage} \text{ returns the negative value of the expected log-likelihood function, } Q = \sum_{i=1}^{N} \left[ \sum_{j=1}^{2} w_{ij} \log \{ \pi_{ij} \} + \sum_{j=1}^{2} \sum_{k=1}^{2} w_{ij} y_{ik}^\ast \log \{ \pi_{ikj}^\ast \} + \sum_{j=1}^{2} \sum_{k=1}^{2} \sum_{l=1}^{2} w_{ij} y_{ik}^\ast \tilde{y}_{il} \log \{ \tilde{\pi}_{iklj} \} \right], \text{ at the provided inputs.}
\]

---

**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**

\[
\text{mean_pistarjj_compute(pistar_matrix, j, sample_size)}
\]

**Arguments**

**pistar_matrix**  
A numeric matrix of conditional probabilities obtained from the internal function \( \text{pistar_compute_for_chains} \). 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.
misclassification_prob

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, $Y^* \mid Y$ (observed outcome, given the true outcome) $\sim Z$ (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, $Y^{\star}$, 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

```r
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 $\tilde{Y} \in \{1, 2\}$ given the latent true outcome $Y \in \{1, 2\}$ and the first-stage outcome $Y^* \in \{1, 2\}$ as $\frac{\exp\{\delta_{kj} + \delta_{kl} \cdot V_i\}}{1 + \exp\{\delta_{kj} + \delta_{kl} \cdot V_i\}}$ for each of the $i = 1, \ldots, n$ subjects.

**Usage**

```r
misclassification_prob2(delta_array, v_matrix)
```

**Arguments**

- **delta_array**: A numeric array of estimated regression parameters for the observation mechanism, $\tilde{Y} | Y^*, Y$ (second-stage observed outcome, given the first-stage observed outcome and the true outcome) $\sim V$ (second-stage misclassification predictor matrix). Rows of the array correspond to parameters for the $\tilde{Y} = 1$ observed outcome, with the dimensions of `v_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.

- **v_matrix**: A numeric matrix of covariates in the second-stage observation mechanism. `v_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 `v_matrix`. The second column, `Y`, represents a true, latent outcome category $Y \in \{1, 2\}$. The third column, `Ystar`, represents a first-stage observed outcome category $Y^* \in \{1, 2\}$. The fourth column, `Ytilde`, represents a second-stage observed outcome category $\tilde{Y} \in \{1, 2\}$. The last column, `Probability`, is the value of the equation $\frac{\exp\{\delta_{kj} + \delta_{kl} \cdot V_i\}}{1 + \exp\{\delta_{kj} + \delta_{kl} \cdot V_i\}}$ computed for each subject,
first-stage observed outcome category, second-stage observed outcome category, and true, latent outcome category.

Examples

```r
set.seed(123)
sample_size <- 1000
cov1 <- rnorm(sample_size)
cov2 <- rnorm(sample_size, 1, 2)
v_matrix <- matrix(c(cov1, cov2), nrow = sample_size, byrow = FALSE)
estimated_deltas <- array(c(1, -1, .5, .2, -.6, 1.5,
                           -1, .5, -1, -.5, -1, -.5), dim = c(3,2,2))
P_Ytilde_Ystar_Y <- misclassification_prob2(estimated_deltas, v_matrix)
head(P_Ytilde_Ystar_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**

```r
model_picker(prior)
```

**Arguments**

- **prior**
  
  A character string specifying the prior distribution for the \( \beta \) and \( \gamma \) 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**

```r
model_picker_2stage(prior)
```

**Arguments**

- `prior`  
  A character string specifying the prior distribution for the $\beta$, $\gamma$, and $\delta$ 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**

```r
naive_jags_picker(
  prior,
  sample_size,
  dim_x,
  n_cat,
  Ystar,
  X,
  beta_prior_parameters,
  number_MCMC_chains,
  naive_model_file,
  display_progress = TRUE
)
```
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 \( \text{dim}_x \times n_{\text{cat}} \), and all elements in the \( n_{\text{cat}} \) column should be set to \( \text{NA} \).

**delta_prior_parameters**

A numeric list of prior distribution parameters for the naive \( \delta \) 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 \( \delta \) 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 \( \delta \) terms. For prior distribution "t", the third element of the list should contain an array of the degrees of freedom for \( \delta \) terms. The third list element should be empty for all other prior distributions. All arrays in the list should have dimensions \( n_{\text{cat}} \times n_{\text{cat}} \times \text{dim}_v \), and all elements in the \( n_{\text{cat}} \) row should be set to \( \text{NA} \).

**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 \texttt{rjags}.

**display_progress**

A logical value specifying whether messages should be displayed during model compilation. The default is \texttt{TRUE}.

**Value**

\texttt{naive_jags_picker_2stage} returns a \texttt{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**

\begin{verbatim}
naive_loglik_2stage(
  param_current, 
  X, 
  V, 
  obs_Ystar_matrix, 
  obs_Ytilda_matrix, 
  sample_size, 
  n_cat
)
\end{verbatim}
Arguments

param_current  A numeric vector of regression parameters, in the order $\beta, \delta$. The $\delta$ 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. $\text{delta_v} \leftarrow \text{c(delta_matrix)}$.

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 $Y^\ast$. 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, $Y^\ast$ and $\tilde{Y}$, can take.

Value

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

at the provided inputs.

Description

Select a Logistic Regression Model for a Given Prior

Usage

naive_model_picker(prior)

Arguments

prior  A character string specifying the prior distribution for the naive $\beta$ 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

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,
perfect_sensitivity_EM

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 ($\theta$) and the observation mechanism ($\beta$), 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

pistar_by_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_compute

**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, $Y^* \mid Y$ (observed outcome, given the true outcome) $\sim Z$ (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, $P(Y_i^* = k \mid Y_i = j, Z_i) = \frac{\exp(\gamma_{jk0} + \gamma_{kj} Z_{ij})}{1 + \exp(\gamma_{jk0} + \gamma_{kj} Z_{ij})}$ 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

\texttt{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 \( j = 1, \ldots, n_{\text{cat}} \).

\begin{verbatim}
pitilde_by_chain(pitilde_by_chain(n_chains, chains_list, V, n, n_cat))
\end{verbatim}

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

\texttt{pitilde_by_chain(n_chains, chains_list, V, n, n_cat)}

Arguments

\begin{itemize}
\item \texttt{n_chains} An integer specifying the number of MCMC chains to compute over.
\item \texttt{chains_list} A numeric list containing the samples from \texttt{n_chains} MCMC chains.
\item \texttt{V} A numeric design matrix.
\item \texttt{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, \texttt{V}.
\item \texttt{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.
\end{itemize}

Value

\texttt{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 \texttt{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 \( P(\tilde{Y} = 2 | Y^* = 2, Y = 2, V) \).
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 $Y = 1$ 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, $P(\tilde{Y}_i = \ell | Y^*_i = k, Y_i = j, V_i) = \frac{\exp\{\delta_{\ell k j0} + \delta_{\ell kj V_i}\}}{1 + \exp\{\delta_{\ell k0j} + \delta_{\ell j V_i}\}}$ 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, $j = 1, \ldots, n_{cat}$.
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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>beta</code></td>
<td>A numeric column matrix of regression parameters for the $Y$ (true outcome) $\sim X$ (predictor matrix of interest).</td>
</tr>
<tr>
<td><code>X</code></td>
<td>A numeric design matrix.</td>
</tr>
<tr>
<td><code>n</code></td>
<td>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, $X$.</td>
</tr>
<tr>
<td><code>n_cat</code></td>
<td>The number of categorical values that the true outcome, $Y$, can take.</td>
</tr>
</tbody>
</table>

Value

The function `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 $j = 1, \ldots, n_{\text{cat}}$.

---

**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 $\beta$ parameters.

**Usage**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>beta</code></td>
<td>A numeric vector of regression parameters for the $Y$ (true outcome) $\sim X$ (predictor matrix of interest).</td>
</tr>
<tr>
<td><code>X</code></td>
<td>A numeric design matrix.</td>
</tr>
<tr>
<td><code>w_mat</code></td>
<td>Matrix of E-step weights obtained from $w_{j}$.</td>
</tr>
<tr>
<td><code>sample_size</code></td>
<td>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, $X$.</td>
</tr>
<tr>
<td><code>n_cat</code></td>
<td>The number of categorical values that the true outcome, $Y$, can take.</td>
</tr>
</tbody>
</table>

**Value**

$q_{\text{beta}}$ 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 \( \delta \) parameters.

**Usage**

```r
q_delta_f(
  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 \( \delta \) 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 \( \delta \) corresponds to the true outcome categories \( Y \in \{1, 2\} \). The numeric vector \( \delta \) is obtained by concatenating the delta array, i.e. \( \text{delta_v} <- c(\text{delta_array}) \).
  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 \( w_{j2}\).
  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, \( V \).  # 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 \( \delta \) 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 \( \delta \) corresponds to the true outcome categories \( Y \in \{1, 2\} \). The numeric vector \( \delta \) is obtained by concatenating the delta array, i.e. \( \text{delta_v} <- c(\text{delta_array}) \).
- **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 \( w_{j2}\).
- **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, \( V \).
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, $Q_{\delta} = \sum_{i=1}^{N} \left[ \sum_{j=1}^{2} \sum_{k=1}^{2} \sum_{\ell=1}^{2} w_{ij} y_{ik}^* \tilde{y}_{i\ell} \log \{ \pi_{ikj} \} \right]$, at the provided inputs.

---

$q_{\gamma_f}$

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

**Description**

Objective function of the form: $Q_{\gamma} = \sum_{i=1}^{N} \left[ \sum_{j=1}^{2} \sum_{k=1}^{2} \sum_{\ell=1}^{2} w_{ij} y_{ik}^* \log \{ \pi_{ikj} \} \right]$. Used to obtain estimates of $\gamma$ parameters.

**Usage**

$q_{\gamma_f}(\text{gamma}_v, Z, \text{obs}_Y\_\text{matrix}, \text{w}_\text{mat}, \text{sample}\_\text{size}, n\_\text{cat})$

**Arguments**

- **gamma_v**: A numeric vector of regression parameters for the observed outcome mechanism, $Y^* | Y$ (observed outcome, given the true outcome) $\sim Z$ (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, $Q_{\gamma} = \sum_{i=1}^{N} \left[ \sum_{j=1}^{2} \sum_{k=1}^{2} \sum_{\ell=1}^{2} w_{ij} y_{ik}^* \log \{ \pi_{ikj}^* \} \right]$, 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 n\(^{th}\) element of the vector x.

sum_every_n1

Sum Every "n"th Element, then add 1

Description

Sum Every "n"th Element, then add 1

Usage

sum_every_n1(x, n)

Arguments

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

Value

sum_every_n1 returns a vector of sums of every n\(^{th}\) element of the vector x, plus 1.
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) \( \sim X \) (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

```r
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)
```
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:

\[ \sum_{k=1}^{2} \frac{y_{ik}^* \pi_{ikj} \pi_{ij}}{\sum_{j=1}^{2} \pi_{ikj} \pi_{ij}} \]

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.
\( n_{\text{cat}} \) 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:

\[
\sum_{k=1}^{2} \sum_{\ell=1}^{2} \frac{y_{i\ell} \tilde{y}_{ik} \pi_{ikj} \pi_{ij}}{\sum_{h=1}^{2} \tilde{\pi}_{i\ellkh} \pi_{ikh} \pi_{ihk}}.
\]

Rows of the matrix correspond to each subject. Columns of the matrix correspond to the true outcome categories \( j = 1, \ldots, n_{\text{cat}} \).
Index

check_and_fix_chains, 3
check_and_fix_chains_2stage, 4
COMBO_data, 5
COMBO_data_2stage, 6
COMBO_EM, 8
COMBO_EM_2stage, 10
COMBO_MCMC, 13
COMBO_MCMC_2stage, 16

em_function, 21
em_function_2stage, 22
expit, 23

jags_picker, 24
jags_picker_2stage, 25

label_switch, 28
label_switch_2stage, 28
loglik, 29
loglik_2stage, 30

mean_pistarjj_compute, 31
misclassification_prob, 32
misclassification_prob2, 33
model_picker, 34
model_picker_2stage, 35

naive_jags_picker, 35
naive_jags_picker_2stage, 36
naive_loglik_2stage, 38
naive_model_picker, 39
naive_model_picker_2stage, 40

perfect_sensitivity_EM, 40
pi_compute, 46
pitard_by_chain, 42
pitarc_compute, 42
pistar_compute_for_chains, 43
pitilde_by_chain, 44
pitilde_compute, 45
pitilde_compute_for_chains, 46

q_beta_f, 47
q_delta_f, 48
q_gamma_f, 49

sum_every_n, 50
sum_every_n1, 50

true_classification_prob, 51

w_j, 52
w_j_2stage, 53