Package 'misclassifyr'

August 27, 2024

Title Estimation and inference for misclassification models.

Version 0.0.0.9000

Description This package provides tools for estimation and inference of simple misclassification models, as described in Mattheis (2024).

License `use_mit_license()`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

LinkingTo Rcpp

Imports Rcpp

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

`ancienregime` *ancienregime*

Description

This is a synthetic dataset created to demonstrate the use of the `misclassifyr` package. The dataset contains observations of father-son pairs in the Third-Estate in Ancien Regime France, linked across three censuses (that didn't actually happen) in 1750, 1770, and 1780.

Usage

```
ancienregime
```

Format

A data frame with 50000 rows and 9 variables:

birthplace The region of birth for the son.

birthyear The year of birth for the son.

linked_weight Weights intended to correct for selection into the linked sample based on birthplace and birthyear.

father_occupation_1750 The occupation of the father observed in the (fictional) 1750 census.

father_income_1750 The income 'score' of the father's occupation in 1750, where the score varies across birthyear and birthplace.

son_occupation_1770 The occupation of the son observed in the (fictional) 1770 census.

son_income_1770 The income 'score' of the son's occupation in 1770, where the score varies across birthyear and birthplace.

son_occupation_1780 The occupation of the son observed in the (fictional) 1780 census.

son_income_1780 The income 'score' of the son's occupation in 1780, where the score varies across birthyear and birthplace.

Source

Synthetic data generated for the package.

`loglikelihood` *Returns the log likelihood of the data.*

Description

Returns the log likelihood of the data.

Usage

```
loglikelihood(theta)
```

Arguments

theta	A numeric vector of length Jx(J^2+K) describing the joint distribution of the data.

Value

the log likelihood of the data given theta, i.e. Pi and Delta.

`misclassifyr` *misclassifyr*

Description

This function provides a menu of options for estimation and inference of misclassification models in which the analyst has access to two noisy measures, `Y1` and `Y2` of a latent outcome `Y*`, a correctly measured covariate `X`, and discrete controls `W`.

Usage

```
misclassifyr(
  tab,
  J,
  K,
  model_to_Pi = "model_to_Pi_NP",
  model_to_Delta = "model_to_Delta_NP_ind",
  phi_0 = NA,
  psi_0 = NA,
  X_names = NA,
  Y_names = NA,
  W_names = NA,
  estimate_beta = F,
  estimate_betas = F,
  X_vals = NA,
  Y_vals = NA,
  lambda_pos = NA,
  lambda_dd = NA,
  optim_maxit = 10000,
  optim_tol = 1e-09,
  optim_stepsize = NA,
  check_stability = F,
  stability_sd = 0.1,
  cores = 1
)
```

Arguments

tab	A dataframe or a list of dataframes containing tabulated data or a list of tabulated data split by controls. The columns should be numeric with names `Y1`, `Y2`, `X`, and `n` where `Y1` and `Y2` take each value between `1` and `J`, `X` takes each value between `1` and `K`, and
J	An integer or list corresponding to the number of unique values of `Y1` and `Y2`.

K	An integer or list corresponding to the number of unique values of X.
model_to_Pi	A function or list of functions mapping the parameters of a model for the joint distribution to the joint distribution \Pi
model_to_Delta	A function or list of functions mapping the parameters of a model to the conditional distribution Y1, Y2 | Y*, \Delta
phi_0	A numeric vector or list of numeric vectors providing the starting location for optimization for the argument to model_to_Pi.
psi_0	A numeric vector or list of numeric vectors providing the starting location for optimization for the argument to model_to_Delta.
X_names	A character vector or list corresponding to the values of the regressor X.
Y_names	A character vector or list corresponding to the values of the outcome Y.
W_names	A character vector corresponding to the values of the control W in each cell.
estimate_beta	A logical value indicating whether to regress Y on X.
X_vals	A numeric vector or list of numeric vectors providing the values of X associated with the columns of Pi.
Y_vals	A numeric vector or list of numeric vectors providing the values of Y associated with the rows of Pi.
lambda_pos	scales the penalty for violations of positivity (i.e. all probabilities should be positive).
lambda_dd	scales the penalty for violations of diagonal dominance.
optim_maxit	An integer for the maximum number of iterations in numerical optimization, passed to optim()
optim_tol	A positive number defining convergence in numerical optimization, passed to optim()
optim_stepsize	A positive number for the step size in the numerical gradient, passed to optim()
check_stability	A logical value indicating whether to perform a stability test for the numerical optimizer.
cores	An integer for the number of CPUs available for parallel processing.
split_eta	An integer or list indicating where to split the vector eta in phi and psi, the arguments to model_to_Pi and model_to_Delta respectively.

Value

An object that includes estimates and information from the estimation process

---

`model_to_Delta_NP` *Maps model parameters, psi, to the conditional distribution Y1, Y2 | Y*, Delta.*

Description

Maps model parameters, psi, to the conditional distribution Y1, Y2 | Y*, Delta.

Usage

```
model_to_Delta_NP(psi)
```

Arguments

psi	A numeric vector containing Pi, Delta^(1), and Delta^(2).

Value

A numeric vector corresponding to the (JxJ)xJ matrix Delta.

---

model_to_Delta_NP_ind *Maps model parameters, psi, to the conditional distribution Y1, Y2 | Y*, Delta.*

Description

Maps model parameters, psi, to the conditional distribution Y1, Y2 | Y*, Delta.

Usage

```
model_to_Delta_NP_ind(psi)
```

Arguments

psi	A numeric vector of length 2xJx(J-1) containing Delta^(1) and Delta^(2).

Value

A numeric vector corresponding to the (JxJ)xJ matrix Delta. d

---

model_to_Delta_RL_ind *Maps model parameters, psi, to the joint distribution of the data, theta.*

Description

Longer description of what it does...

Usage

```
model_to_Delta_RL_ind(psi)
```

Arguments

psi	A numeric vector of length 2(J-1)+2J corresponding to the column and row scales of the record linkage.

Details

some details.

Value

something

Examples

```
## Not run:
some example code # Should return something

## End(Not run)
```

`model_to_Pi_NP` *Maps model parameters, phi, to the joint distribution of X and Y*, Pi.*

Description

Maps model parameters, phi, to the joint distribution of X and Y*, Pi.

Usage

```
model_to_Pi_NP(phi)
```

Arguments

`phi`	A numeric vector.
`J`	An integer corresponding to the dimension of Y.
`K`	An integer corresponding to the dimension of X.

Value

A numeric vector corresponding to the JxK matrix Pi

`Pi_to_beta` *Maps the joint distribution, Pi, of X and Y* to a scalar, beta*

Description

Maps the joint distribution, Pi, of X and Y* to a scalar, beta

Usage

```
Pi_to_beta(Pi, X_vals, Y_vals, W_weights)
```

Arguments

`Pi`	A numeric vector or list of numeric vectors containing the elements of Pi.
`X_vals`	A numeric vector or a list of numeric vectors representing the scalar values associated with X.
`Y_vals`	A numeric vector or a list of numeric vectors representing the scalar values associated with Y.
`W_weights`	A numeric vector representing the sample size of each control cell.

Value

A scalar equal to beta.

`Pi_to_betas` *Maps the joint distribution, Pi, of X and Y* to a vector, representing beta in each covariate cell*

Description

Longer description of what it does...

Usage

```
Pi_to_betas(Pi, X_vals, Y_vals)
```

Arguments

`Pi`	A list of numeric vectors containing the elements of Pi.
`X_vals`	A list of numeric vectors representing the scalar values associated with X.
`Y_vals`	A list of numeric vectors representing the scalar values associated with Y.

Details

some details.

Value

A scalar equal to beta.

Examples

```
## Not run:
some example code # Should return something

## End(Not run)
```

`Pi_to_beta_sim` *Maps the joint distribution, Pi, of X and Y* to a scalar, beta via simulation*

Description

Longer description of what it does...

Usage

```
Pi_to_beta_sim(Pi, X_vals, Y_vals, W_weights)
```

Arguments

Pi	A list of numeric vectors containing the elements of Pi
X_vals	A numeric vector or a list of numeric vectors representing the scalar values associated with X.
Y_vals	A numeric vector or a list of numeric vectors representing the scalar values associated with Y.
W_weights	A numeric vector representing the sample size of each control cell.

Details

some details.

Value

A scalar equal to beta.

Examples

```
## Not run:
some example code # Should return something

## End(Not run)
```

---

prep_misclassification_data

prep_misclassification_data

Description

This function tabulates data and generates metadata in a format to be used with the misclassifyr() function.

Usage

```
prep_misclassification_data(
  data,
  outcome_1,
  outcome_2,
  regressor,
  controls = NA,
  weights = NA,
  record_vals = F
)
```

Arguments

data	A data.frame containing the outcome variable,
outcome_1	A character string denoting the variable in the dataframe to be used as the first measure of an outcome, Y_1.

outcome_2	A character string denoting the variable in the dataframe to be used as the second measure of an outcome, Y_2.
regressor	A character string denoting the variable in the dataframe to be used as a regressor, X.
controls	A character string or vector of character strings denoting the variable/variables to be used as non-parametric controls, W.
weights	A character string denoting a variable containing individual level weights
record_vals	A logical value indicating whether to record the unique values of the outcomes and the regressor. If record_vals = F, you likely want to order the data by the regressor and outcomes before applying prep_misclassification_data.

Value

A list of objects including tabulated data to be used in misclassifyr()

`softlog` *Logarithm with a lower bound*

Description

If `x` is greater than `1e-20`, it returns `log(x)`. Otherwise, it returns `log(1e-20)`.

Usage

```
softlog(x)
```

Arguments

x	A numeric vector.

Details

The arguments to the `ll` function are the log value of the probabilities in Delta and Pi of the misclassification model. To prevent convergence issues, `ll` enforces a lower bound on these probabilities of `1e-20`. This function is useful for mapping probabilities that may be zero to `ll`. It will throw an error if any element of `x` is negative.

Value

A numeric vector composed of the elements of `log(x)` or `log(1e-20)` for element is less than `1e-20`.

Examples

```
## Not run:
softlog(c(0.5, 0.1, 0, -1)) # Should return the log values including log(1e-20) for 0 and -1

## End(Not run)
```

`synthetic_data` *Generates synthetic misclassification data.*

Description

Longer description of what it does...

Usage

```
synthetic_data(
  J = 5,
  K = 5,
  I = 2,
  sample_size = 1e+06,
  dgp_delta = "Nonparametric, independent, strong diagonal",
  dgp_pi = "Exponential"
)
```

Arguments

`J`	An integer indicating the dimension of Y.
`K`	An integer indicating the dimension of X.
`I`	An integer indicating the dimension of W.
`sample_size`	An integer denoting the number of synthetic observations.
`dgp_delta`	A character string indicating the data generating process for the synthetic noise
`dgp_pi`	A character string indicating the data generating process for the joint distribution of X and Y*

Details

some details...

Value

A list including tabulated data `tab` and matrices `Pi`, `Delta`

Examples

```
## Not run:
some example code # Should return something

## End(Not run)
```

Index