Package 'SAMBA'

July 21, 2025

Title Selection and Misclassification Bias Adjustment for Logistic Regression Models

Version 0.9.0

Description Health research using data from electronic health records (EHR) has gained popularity, but misclassification of EHR-derived disease status and lack of representativeness of the study sample can result in substantial bias in effect estimates and can impact power and type I error for association tests. Here, the assumed target of inference is the relationship between binary disease status and predictors modeled using a logistic regression model. 'SAMBA' implements several methods for obtaining bias-corrected point estimates along with valid standard errors as proposed in Beesley and Mukherjee (2020) <doi:10.1101/2019.12.26.19015859>, currently under review.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports stats, optimx, survey

Suggests knitr, rmarkdown, ggplot2, scales, MASS

VignetteBuilder knitr

NeedsCompilation no

Author Alexander Rix [cre], Lauren Beesley [aut]

Maintainer Alexander Rix <alexrix@umich.edu>

Repository CRAN

Date/Publication 2020-02-20 07:50:07 UTC

Contents

`approxdist` *Estimate parameters in the disease model approximating the observed data distribution*

Description

`approxdist` estimates parameters in the disease model given a previously-estimated marginal sensitivity. This estimation is based on approximating the distribution of D* given Z.

Usage

```
approxdist(Dstar, Z, c_marg, weights = NULL)
```

Arguments

`Dstar`	Numeric vector containing observed disease status. Should be coded as 0/1
`Z`	Numeric matrix of covariates in disease model
`c_marg`	marginal sensitivity, P(D* = 1 | D = 1, S = 1)
`weights`	Optional numeric vector of patient-specific weights used for selection bias adjustment. Default is NULL

Details

We are interested in modeling the relationship between binary disease status and covariates Z using a logistic regression model. However, D may be misclassified, and our observed data may not well-represent the population of interest. In this setting, we estimate parameters from the disease model using the following modeling framework.

Notation:

D Binary disease status of interest.

D* Observed binary disease status. Potentially a misclassified version of D. We assume D = 0 implies D* = 0.

S Indicator for whether patient from population of interest is included in the analytical dataset.

Z Covariates in disease model of interest.

W Covariates in model for patient inclusion in analytical dataset (selection model).

X Covariates in model for probability of observing disease given patient has disease (sensitivity model).

Model Structure:

Disease Model

$$logit(P(D = 1|X)) = theta_0 + theta_Z Z$$

Selection Model

$$P(S = 1|W, D)$$

Sensitivity Model

$$logit(P(D* = 1|D = 1, S = 1, X)) = beta_0 + beta_X X$$

Value

a list with two elements: (1) 'param', a vector with parameter estimates for disease model (logOR of Z), and (2) 'variance', a vector of variance estimates for disease model parameters. Results do not include intercept.

References

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification Lauren J Beesley and Bhramar Mukherjee medRxiv 2019.12.26.19015859

Examples

```
library(SAMBA)
# These examples are generated from the vignette. See it for more details.

# Generate IPW weights from the true model
expit <- function(x) exp(x) / (1 + exp(x))
prob.WD <- expit(-0.6 + 1 * samba.df$D + 0.5 * samba.df$W)
weights <- nrow(samba.df) * (1  / prob.WD) / (sum(1 / prob.WD))

# Estimate sensitivity by using inverse probability of selection weights
# and P(D=1)
sens <- sensitivity(samba.df$Dstar, samba.df$X, prev = mean(samba.df$D),
                    weights = weights)

approx1 <- approxdist(samba.df$Dstar, samba.df$Z, sens$c_marg,
                      weights = weights)
```

`nonlogistic` *Estimate parameters in the disease model given sensitivity as a function of covariates.*

Description

non-logistic link function for D* given Z and sensitivity. This function assumes that sensitivity as a function of X is known or has been estimated

Usage

```
nonlogistic(Dstar, Z, c_X, weights = NULL)
```

Arguments

`Dstar`	Numeric vector containing observed disease status. Should be coded as 0/1
`Z`	numeric matrix of covariates in disease model
`c_X`	sensitivity as a function of X, P(D* = 1| D = 1, S = 1, X)
`weights`	Optional numeric vector of patient-specific weights used for selection bias adjustment. Default is NULL

Details

We are interested in modeling the relationship between binary disease status and covariates Z using a logistic regression model. However, D may be misclassified, and our observed data may not well-represent the population of interest. In this setting, we estimate parameters from the disease model using the following modeling framework.

Notation:

D Binary disease status of interest.

D* Observed binary disease status. Potentially a misclassified version of D. We assume D = 0 implies D* = 0.

S Indicator for whether patient from population of interest is included in the analytical dataset.

Z Covariates in disease model of interest.

W Covariates in model for patient inclusion in analytical dataset (selection model).

X Covariates in model for probability of observing disease given patient has disease (sensitivity model).

Model Structure:

Disease Model

$$logit(P(D = 1|X)) = theta_0 + theta_Z Z$$

Selection Model

$$P(S = 1|W, D)$$

Sensitivity Model

$$logit(P(D* = 1|D = 1, S = 1, X)) = beta_0 + beta_X X$$

Value

a list with two elements: (1) 'param', a vector with parameter estimates for disease model (logOR of Z), and (2) 'variance', a vector of variance estimates for disease model parameters. Results do not include intercept.

References

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification Lauren J Beesley and Bhramar Mukherjee medRxiv 2019.12.26.19015859

Examples

```
library(SAMBA)
# These examples are generated from the vignette. See it for more details.

# Generate IPW weights from the true model
expit <- function(x) exp(x) / (1 + exp(x))
prob.WD <- expit(-0.6 + 1 * samba.df$D + 0.5 * samba.df$W)
weights <- nrow(samba.df) * (1  / prob.WD) / (sum(1 / prob.WD))

# Estimate sensitivity by using inverse probability of selection weights
# and P(D=1)
sens <- sensitivity(samba.df$Dstar, samba.df$X, prev = mean(samba.df$D),
                    weights = weights)

nonlog1 <- nonlogistic(samba.df$Dstar, samba.df$Z, c_X = sens$c_X,
                       weights = weights)
```

`obsloglik` *Estimate parameters in the disease model using observed data log-likelihood using direct maximization.*

Description

`obsloglik` jointly estimates the disease model and sensitivity model parameters using profile likelihood methods. Estimation involves direct maximization of the observed data log-likelihood.

Usage

```
obsloglik(Dstar, Z, X, start, beta0_fixed = NULL, weights = NULL,
  expected = TRUE, itnmax = 5000)
```

Arguments

`Dstar`	Numeric vector containing observed disease status. Should be coded as 0/1
`Z`	Numeric matrix of covariates in disease model. 'Z' should not contain an intercept
`X`	Numeric matrix of covariates in sensitivity model. Set to NULL to fit model with no covariates in sensitivity model. 'X' should not contain an intercept
`start`	Numeric vector of starting values for theta and beta (theta, beta). Theta is the parameter of the disease model, and beta is the parameter of the sensitivity model
`beta0_fixed`	Optional numeric vector of values of sensitivity model intercept to profile over. If a single value, corresponds to fixing intercept at specified value. Default is NULL
`weights`	Optional vector of patient-specific weights used for selection bias adjustment. Default is NULL
`expected`	Whether or not to calculate the covariance matrix via the expected fisher information matrix. Default is TRUE
`itnmax`	Maximum number of iterations to run `optimx`

Details

We are interested in modeling the relationship between binary disease status and covariates Z using a logistic regression model. However, D may be misclassified, and our observed data may not well-represent the population of interest. In this setting, we estimate parameters from the disease model using the following modeling framework. Notation:

D Binary disease status of interest.

D* Observed binary disease status. Potentially a misclassified version of D. We assume D = 0 implies D* = 0.

S Indicator for whether patient from population of interest is included in the analytical dataset.

Z Covariates in disease model of interest.

W Covariates in model for patient inclusion in analytical dataset (selection model).

X Covariates in model for probability of observing disease given patient has disease (sensitivity model).

Model Structure:

Disease Model

$$logit(P(D = 1|X)) = theta_0 + theta_Z Z$$

Selection Model

$$P(S = 1|W, D)$$

Sensitivity Model

$$logit(P(D* = 1|D = 1, S = 1, X)) = beta_0 + beta_X X$$

Value

A "SAMBA.fit" object with nine elements: 'param', the maximum likelihood estimate of the coeficients, 'variance', the covariance matrix of the final estimate, param.seq', the sequence of estimates at each value of beta0, and 'loglik.seq', the log likelihood at each value. The rest of the elements are Dstar', 'X', 'Z', and 'weights'.

References

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification Lauren J Beesley and Bhramar Mukherjee medRxiv 2019.12.26.19015859

Examples

```
library(SAMBA)
# These examples are generated from the vignette. See it for more details.

# Generate IPW weights from the true model
expit <- function(x) exp(x) / (1 + exp(x))
prob.WD <- expit(-0.6 + 1 * samba.df$D + 0.5 * samba.df$W)
weights <- nrow(samba.df) * (1  / prob.WD) / (sum(1 / prob.WD))

# Get initial parameter estimates
```

```
logit <- function(x) log(x / (1 - x))
fitBeta  <- glm(Dstar ~ X, binomial(), data = samba.df)
fitTheta <- glm(Dstar ~ Z, binomial(), data = samba.df)

sens <- sensitivity(samba.df$Dstar, samba.df$X, mean(samba.df$D), r = 2)
start <- c(coef(fitTheta), logit(sens$c_marg), coef(fitBeta)[2])

# Direct observed data likelihood maximization without fixed intercept

fit1 <- obsloglik(samba.df$Dstar, samba.df$Z, samba.df$X, start = start,
                  weights = weights)
obsloglik1 <- list(param = fit1$param, variance = diag(fit1$variance))

# Direct observed data likelihood maximization with fixed intercept

fit2   <- obsloglik(samba.df$Dstar, samba.df$Z, samba.df$X, start = start,
                  beta0_fixed = logit(sens$c_marg), weights = weights)

# since beta0 is fixed, its variance is NA
obsloglik1 <- list(param = fit2$param, variance = diag(fit2$variance))
```

`obsloglikEM` *Estimate parameters in the disease model using observed data log-likelihood using the expectation-maximization algorithm*

Description

`obsloglikEM` jointly estimates the disease model and sensitivity model parameters using profile likelihood methods. Estimation involves an expectation-maximization algorithm.

Usage

```
obsloglikEM(Dstar, Z, X, start, beta0_fixed = NULL, weights = NULL,
  expected = TRUE, tol = 1e-06, maxit = 50)
```

Arguments

`Dstar`	Numeric vector containing observed disease status. Should be coded as 0/1
`Z`	Numeric matrix of covariates in disease model. 'Z' should not contain an intercept
`X`	Numeric matrix of covariates in sensitivity model. Set to NULL to fit model with no covariates in sensitivity model. 'X' should not contain an intercept
`start`	Numeric vector of starting values for theta and beta (theta, beta). Theta is the parameter of the disease model, and beta is the parameter of the sensitivity model
`beta0_fixed`	Optional numeric vector of values of sensitivity model intercept to profile over. If a single value, corresponds to fixing intercept at specified value. Default is NULL

`weights`	Optional vector of patient-specific weights used for selection bias adjustment. Default is NULL
`expected`	Whether or not to calculate the covariance matrix via the expected fisher information matrix. Default is TRUE
`tol`	stop estimation when subsequent log-likelihood estimates are within this value
`maxit`	Maximum number of iterations of the estimation algorithm

Details

We are interested in modeling the relationship between binary disease status and covariates Z using a logistic regression model. However, D may be misclassified, and our observed data may not well-represent the population of interest. In this setting, we estimate parameters from the disease model using the following modeling framework. Notation:

D Binary disease status of interest.

D* Observed binary disease status. Potentially a misclassified version of D. We assume D = 0 implies D* = 0.

S Indicator for whether patient from population of interest is included in the analytical dataset.

Z Covariates in disease model of interest.

W Covariates in model for patient inclusion in analytical dataset (selection model).

X Covariates in model for probability of observing disease given patient has disease (sensitivity model).

Model Structure:

Disease Model

$$logit(P(D = 1|X)) = theta_0 + theta_Z Z$$

Selection Model

$$P(S = 1|W, D)$$

Sensitivity Model

$$logit(P(D* = 1|D = 1, S = 1, X)) = beta_0 + beta_X X$$

Value

A "SAMBA.fit" object with nine elements: 'param', the final estimate of the coeficients organized as (theta, beta), 'variance', the covariance matrix of the final estimate, param.seq', the sequence of estimates at each step of the EM algorithm, and 'loglik.seq', the log likelihood at each step. The rest of the elements are Dstar', 'X', 'Z', and 'weights'.

References

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification Lauren J Beesley and Bhramar Mukherjee medRxiv 2019.12.26.19015859

Examples

```
library(SAMBA)
# These examples are generated from the vignette. See it for more details.

# Generate IPW weights from the true model
expit <- function(x) exp(x) / (1 + exp(x))
prob.WD <- expit(-0.6 + 1 * samba.df$D + 0.5 * samba.df$W)
weights <- nrow(samba.df) * (1  / prob.WD) / (sum(1 / prob.WD))

# Get initial parameter estimates
logit <- function(x) log(x / (1 - x))
fitBeta  <- glm(Dstar ~ X, binomial(), data = samba.df)
fitTheta <- glm(Dstar ~ Z, binomial(), data = samba.df)

sens <- sensitivity(samba.df$Dstar, samba.df$X, mean(samba.df$D), r = 2)
start <- c(coef(fitTheta), logit(sens$c_marg), coef(fitBeta)[2])

# Direct observed data likelihood maximization without fixed intercept
fit1 <- obsloglikEM(samba.df$Dstar, samba.df$Z, samba.df$X, start = start,
                weights = weights)
obsloglik1 <- list(param = fit1$param, variance = diag(fit1$variance))

# Direct observed data likelihood maximization with fixed intercept
fit2   <- obsloglikEM(samba.df$Dstar, samba.df$Z, samba.df$X, start = start,
                beta0_fixed = logit(sens$c_marg), weights = weights)
# since beta0 is fixed, its variance is NA

list(param = fit2$param, variance = diag(fit2$variance))
```

---

samba.df *Synthetic example data for SAMBA adapted from the vignette*

Description

'samba.df' is the sampled data from the entire population

Usage

```
samba.df
```

Format

A synthetic data.frame with 4999 observations on 5 variables:

X Covariate for sensitivity model.

Z Covariate for disease model.

W Selection Covariate

D True disease status.

Dstar Observed disease status.

`sensitivity` *Estimate sensitivity*

Description

`sensitivity` estimates (1) marginal sensitivity and (2) sensitivity as a function of covariates X for a misclassified binary outcome.

Usage

```
sensitivity(Dstar, X, prev, r = NULL, weights = NULL)
```

Arguments

`Dstar`	Numeric vector containing observed disease status. Should be coded as 0/1
`X`	Numeric matrix with covariates in sensitivity model. Set to NULL to fit model with no covariates in sensitivity model. 'X' should not contain an intercept
`prev`	marginal disease prevalence $P(D = 1)$ or patient-specific $P(D = 1\|X)$ in population
`r`	(optional) marginal sampling ratio, $P(S = 1\|D = 1)/P(S = 1\|D = 0)$. Only one of 'r' and 'weights' can be specified. Default is 'NULL'
`weights`	Optional vector of patient-specific weights used for selection bias adjustment. Only one of r and weights can be specified. Default is 'NULL'

Details

We are interested in modeling the relationship between binary disease status and covariates Z using a logistic regression model. However, D may be misclassified, and our observed data may not well-represent the population of interest. In this setting, we estimate parameters from the disease model using the following modeling framework.

Notation:

D Binary disease status of interest.

D* Observed binary disease status. Potentially a misclassified version of D. We assume D = 0 implies D* = 0.

S Indicator for whether patient from population of interest is included in the analytical dataset.

Z Covariates in disease model of interest.

W Covariates in model for patient inclusion in analytical dataset (selection model).

X Covariates in model for probability of observing disease given patient has disease (sensitivity model).

Model Structure:

Disease Model

$$logit(P(D = 1|X)) = theta_0 + theta_Z Z$$

Selection Model

$$P(S = 1|W, D)$$

Sensitivity Model

$$logit(P(D* = 1|D = 1, S = 1, X)) = beta_0 + beta_X X$$

Value

a list with two elements: (1) 'c_marg', marginal sensitivity estimate $P(D* = 1|D = 1, S = 1)$, and (2) 'c_X', sensitivity as a function of X $P(D* = 1|D = 1, S = 1, X)$

References

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification Lauren J Beesley and Bhramar Mukherjee medRxiv 2019.12.26.19015859

Examples

```
library(SAMBA)
# These examples are generated from the vignette. See it for more details.

# Generate IPW weights from the true model
expit <- function(x) exp(x) / (1 + exp(x))
prob.WD <- expit(-0.6 + 1 * samba.df$D + 0.5 * samba.df$W)
weights <- nrow(samba.df) * (1  / prob.WD) / (sum(1 / prob.WD))

# Using marginal sampling ratio r ~ 2 and P(D=1)
sens <- sensitivity(samba.df$Dstar, samba.df$X, mean(samba.df$D),
                    r = 2)
# Using inverse probability of selection weights and P(D=1)
sens <- sensitivity(samba.df$Dstar, samba.df$X, prev = mean(samba.df$D),
                    weights = weights)
```

Index