Package 'prome'

March 22, 2026

Title Patient-Reported Outcome Data Analysis with Stan

Type Package

Version 4.0.0.3

Description Estimation for blinding bias in randomized controlled trials with a latent continuous outcome, a binary response depending on treatment and the latent outcome, and a noisy surrogate subject to possibly response-dependent measurement error. Implements EM estimators in R backed by compiled C routines for models with and without the restriction delta0 = 0, and Bayesian Stan wrappers for the same two models. Functions were added for latent outcome models with differential measurement error.

License Unlimited

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, utils, rstan

LinkingTo Rcpp, RcppEigen, StanHeaders

Suggests posterior

RoxygenNote 7.3.3

NeedsCompilation yes

Author Bin Wang [aut, cre]

Maintainer Bin Wang <bwang831@gmail.com>

Repository CRAN

Date/Publication 2026-03-22 19:40:02 UTC

Contents

`blindingMLE` *Unblinding bias correction*

Description

Unblinding bias correction

Usage

```
blindingMLE(
  W,
  T,
  G,
  Bayes = FALSE,
  ActiveOnly = TRUE,
  max_iter = 200,
  tol = 1e-06,
  grid_size = 81,
  grid_sd = 6,
  verbose = FALSE,
  compute_se = TRUE,
  hess_step = 1e-04,
  init = NULL,
  chains = 4,
  cores = chains,
  iter_warmup = 1000,
  iter_sampling = 1000,
  seed = 123,
  control = list(adapt_delta = 0.95, max_treedepth = 12),
  refresh = 0
)
```

Arguments

`W`	Numeric surrogate outcome.
`T`	Binary treatment indicator.
`G`	Binary response indicator.
`Bayes`	choose from Bayesian (stan models) or EM approaches.
`ActiveOnly`	pose restriction delta0 = 0.
`max_iter`	Maximum EM iterations.
`tol`	Convergence tolerance.

grid_size	Number of E-step grid points.
grid_sd	Grid half-width in posterior SD units.
verbose	Print iteration trace.
compute_se	Compute approximate SEs from a numerical observed Hessian.
hess_step	Finite-difference step for Hessian approximation.
init	Optional named list of initial values.
chains	Number of chains.
cores	Number of CPU cores for parallel sampling. Defaults to ‘chains‘.
iter_warmup	Warmup iterations.
iter_sampling	Sampling iterations.
seed	RNG seed.
control	Optional named list of ‘rstan‘ control arguments such as ‘list(adapt_delta = 0.95, max_treedepth = 12)‘.
refresh	Stan refresh interval.

Value

An object of class ‘rctme_fit‘.

Examples

```
## Not run:
if (requireNamespace("rstan", quietly = TRUE)) {
sigma = 1.2
sig.theta = 1.0
beta0 = 0
beta1 = 1
beta2 = 2
ntreat = nsham = 100
Tind  = c(rep(1, ntreat), rep(0,nsham))  #treatment group indicator
u1v   = rep(u1,ntreat)
u2v   = rep(u2,nsham)
uv    = c(u1v,u2v)
tauv  = uv - rep(u2, ntreat+nsham)
r = rnorm(ntreat + nsham, mean = 0, sd = sigma)
x = uv + r #actual endpoint outcome
q = 1/(1 + exp(-(beta0 + beta1*Tind + beta2*(tauv+r))))
bernGen = function(qq){rbinom(1,1,qq)}
I = sapply(q,bernGen)
rsham = rnorm(ntreat + nsham, mean = 0, sd = sig.theta)
w = x + (theta + rsham)*I
lm0 = lm(w~Tind)
tau1 = lm0$coef[2]; tau1
u12 = tapply(w,Tind,FUN=mean,na.rm=TRUE)
lm1 = lm(w~Tind+I)
tau2 = lm1$coef[2]; tau2
mydata <- data.frame(y=w,group=Tind,guess=I)
```

```
out5 <- blindingMLE(W = mydata$y, T = mydata$group, G = mydata$guess,
                    Bayes=FALSE,ActiveOnly=TRUE)
out5
}

## End(Not run)
```

`fit_rctme_em` *EM fit with free 'delta0' and 'delta1'*

Description

Fits the latent-outcome model

$$X|T \sim N(\mu_0 + \tau T, \sigma_X^2),$$
$$G|X,T \sim \text{Bernoulli}(\text{logit}^{-1}(\alpha_0 + \alpha_T T + \alpha_X (X - \mu_0))),$$
$$W|X, G = g \sim N(X + \delta_g, \sigma_{Wg}^2).$$

Usage

```
fit_rctme_em(
  W,
  T,
  G,
  max_iter = 200,
  tol = 1e-06,
  grid_size = 81,
  grid_sd = 6,
  verbose = FALSE,
  compute_se = TRUE,
  hess_step = 1e-04,
  init = NULL
)
```

Arguments

`W`	Numeric surrogate outcome.
`T`	Binary treatment indicator.
`G`	Binary response indicator.
`max_iter`	Maximum EM iterations.
`tol`	Convergence tolerance.
`grid_size`	Number of E-step grid points.
`grid_sd`	Grid half-width in posterior SD units.
`verbose`	Print iteration trace.
`compute_se`	Compute approximate SEs from a numerical observed Hessian.
`hess_step`	Finite-difference step for Hessian approximation.
`init`	Optional named list of initial values.

Details

The E-step and observed-data log-likelihood are evaluated in compiled C code for speed.

Value

An object of class 'rctme_fit'.

`fit_rctme_em_delta0_fixed`
EM fit with 'delta0' fixed at 0

Description

Same model as [fit_rctme_em()] but with the measurement-error mean shift in the 'G = 0' group fixed at zero.

Usage

```
fit_rctme_em_delta0_fixed(
  W,
  T,
  G,
  max_iter = 200,
  tol = 1e-06,
  grid_size = 81,
  grid_sd = 6,
  verbose = FALSE,
  compute_se = TRUE,
  hess_step = 1e-04,
  init = NULL
)
```

Arguments

`W`	Numeric surrogate outcome.
`T`	Binary treatment indicator.
`G`	Binary response indicator.
`max_iter`	Maximum EM iterations.
`tol`	Convergence tolerance.
`grid_size`	Number of E-step grid points.
`grid_sd`	Grid half-width in posterior SD units.
`verbose`	Print iteration trace.
`compute_se`	Compute approximate SEs from a numerical observed Hessian.
`hess_step`	Finite-difference step for Hessian approximation.
`init`	Optional named list of initial values.

Value

An object of class ‘rctme_fit‘.

`fit_rctme_stan` *Bayesian Stan fit with free ‘delta0‘ and ‘delta1‘*

Description

Fits the unrestricted Bayesian latent-outcome model using ‘rstan‘ and the Stan program bundled with the package.

Usage

```
fit_rctme_stan(
  W,
  T,
  G,
  chains = 4,
  cores = chains,
  iter_warmup = 1000,
  iter_sampling = 1000,
  seed = 123,
  control = list(adapt_delta = 0.95, max_treedepth = 12),
  refresh = 200
)
```

Arguments

`W`	Numeric surrogate outcome.
`T`	Binary treatment indicator.
`G`	Binary response indicator.
`chains`	Number of chains.
`cores`	Number of CPU cores for parallel sampling. Defaults to ‘chains‘.
`iter_warmup`	Warmup iterations.
`iter_sampling`	Sampling iterations.
`seed`	RNG seed.
`control`	Optional named list of ‘rstan‘ control arguments such as ‘list(adapt_delta = 0.95, max_treedepth = 12)‘.
`refresh`	Stan refresh interval.

Value

A list with components ‘fit‘, ‘summary‘, and ‘stan_file‘.

`fit_rctme_stan_delta0_fixed` *Bayesian Stan fit with ‘delta0‘ fixed at 0*

Description

Fits the restricted Bayesian latent-outcome model using ‘rstan‘ and the Stan program bundled with the package.

Usage

```
fit_rctme_stan_delta0_fixed(
  W,
  T,
  G,
  chains = 4,
  cores = chains,
  iter_warmup = 1000,
  iter_sampling = 1000,
  seed = 123,
  control = list(adapt_delta = 0.95, max_treedepth = 12),
  refresh = 200
)
```

Arguments

`W`	Numeric surrogate outcome.
`T`	Binary treatment indicator.
`G`	Binary response indicator.
`chains`	Number of chains.
`cores`	Number of CPU cores for parallel sampling. Defaults to ‘chains‘.
`iter_warmup`	Warmup iterations.
`iter_sampling`	Sampling iterations.
`seed`	RNG seed.
`control`	Optional named list of ‘rstan‘ control arguments such as ‘list(adapt_delta = 0.95, max_treedepth = 12)‘.
`refresh`	Stan refresh interval.

Value

A list with components ‘fit‘, ‘summary‘, and ‘stan_file‘.

`prome` *rctme: Latent-outcome models for RCTs with differential measurement error*

Description

Tools for fitting latent continuous-outcome models in randomized trials with a binary response process and a surrogate measurement subject to response-dependent error. The package provides EM estimators implemented with compiled C code for fast E-steps and observed-data log-likelihood evaluation, plus Bayesian Stan wrappers for corresponding models.

Author(s)

Maintainer: Bin Wang <bwang831@gmail.com>

`rctme_stan_file` *Locate a bundled Stan file*

Description

Locate a bundled Stan file

Usage

```
rctme_stan_file(model = c("delta0_fixed", "delta_free"))
```

Arguments

`model`	One of ‘"delta0_fixed"‘ or ‘"delta_free"‘.

Value

Path to the ‘.stan‘ file shipped with the package.

`simulate_rctme_data` *Simulate a small RCT measurement-error example dataset*

Description

Generates a toy dataset for examples and testing under the latent outcome model implemented in this package.

Usage

```
simulate_rctme_data(
  n = 200,
  mu0 = 0.5,
  tau = 1,
  sigmaX = 1,
  alpha0 = -0.5,
  alphaT = 0.4,
  alphaX = 1,
  delta0 = 0,
  delta1 = -0.4,
  sigmaW0 = 0.8,
  sigmaW1 = 1,
  seed = NULL
)
```

Arguments

`n`	Sample size.
`mu0`	Mean of latent outcome in controls.
`tau`	Treatment effect on latent outcome.
`sigmaX`	SD of latent outcome noise.
`alpha0`, `alphaT`, `alphaX`	Logistic-response coefficients.
`delta0`, `delta1`	Mean measurement error shifts by response group.
`sigmaW0`, `sigmaW1`	Measurement-error SDs by response group.
`seed`	Optional RNG seed.

Value

A data frame with columns `W`, `T`, `G`, and latent `X`.

Index