Package ‘OptimalSurrogate’

May 7, 2026

Title Model Free Approach to Quantifying Surrogacy

Version 1.0

Description Identifies an optimal transformation of a surrogate marker such that the proportion of treatment effect explained can be inferred based on the transformation of the surrogate and nonparametrically estimates two model-free quantities of this proportion. Details are described in Wang et al (2020) <doi:10.1093/biomet/asz065>.

License GPL-3

Depends R (>= 3.0)

Imports splines, MASS, stats

NeedsCompilation no

Author Xuan Wang [aut],
Layla Parast [aut, cre],
Ming Yang [aut],
Lu Tian [aut],
Tianxi Cai [aut]

Maintainer Layla Parast <`parast@austin.utexas.edu`>

Repository CRAN

Date/Publication 2022-09-27 10:20:07 UTC

Contents

`marker_cont` *Simulated data with continuous surrogate marker*

Description

Simulated data with continuous surrogate marker

`marker_disc` *Simulated data with discrete surrogate marker*

Description

Simulated data with discrete surrogate marker

`pte_cont` *PTE estimation with a continuous surrogate marker*

Description

PTE estimation with continuous surrogate marker

Usage

```
pte_cont(sob, yob, aob, var = TRUE, conf.int = TRUE, rep = 500)
```

Arguments

`sob`	CONTINUOUS surrogate marker
`yob`	outcome of interest (continuous or binary)
`aob`	treatment assignment (1: treatment; 0: control)
`var`	whether variance should be calculated (TRUE/FALSE)
`conf.int`	whether 95% confidence intervals should be calculated (TRUE/FALSE)
`rep`	number of resampling replications (default is 500)

Value

`Estimates`	Estimates of the treatment effect on the priamry outcome, delta, the treatment effect on the transformation of the surrogate, delta.gs, two versions of the proportion of treatment effect explained by the surrogate, pte1 and pte2; if var = TRUE, standard error estimates are also provided (se); if conf.int = TRUE, 95% confidence intervals are also provided
`Transformed.S`	the transformed surrogate, g(s), for each value of the surrogate, s; if var = TRUE, standard error estimates are also provided (se); if conf.int = TRUE, 95% confidence intervals are also provided

Examples

```
data(marker_cont)
out <- pte_cont(
  sob = marker_cont$sob,
  yob = marker_cont$yob,
  aob = marker_cont$aob, var = FALSE, conf.int = FALSE)
out

x <- as.numeric(names(out$Transformed.S))
plot(x, out$Transformed.S, ylim = range(out$Transformed.S), type = "l",
     las = 1, xlab = "Surrogate Marker", ylab = "Optimal Transformation")


out <- pte_cont(
  sob = marker_cont$sob,
  yob = marker_cont$yob,
  aob = marker_cont$aob,
  var = TRUE, conf.int = TRUE, rep = 1000)
out$Estimates

x <- as.numeric(rownames(out$Transformed.S))
plot(x, out$Transformed.S[, "est"], ylim = range(out$Transformed.S[, -2]), type = "l",
     las = 1, xlab = "Surrogate Marker", ylab = "Optimal Transformation")
  lines(x, out$Transformed.S[, "lower"], lty = 2)
  lines(x, out$Transformed.S[, "upper"], lty = 2)
```

`pte_disc` *PTE estimation with a discrete surrogate marker*

Description

PTE estimation with discrete surrogate marker

Usage

```
pte_disc(sob, yob, aob, var = TRUE, conf.int = TRUE, rep = 500)
```

Arguments

`sob`	DISCRETE surrogate marker
`yob`	outcome of interest (continuous or binary)
`aob`	treatment assignment (1: treatment; 0: control)
`var`	whether variance should be calculated (TRUE/FALSE)
`conf.int`	whether 95% confidence intervals should be calculated (TRUE/FALSE)
`rep`	number of resampling replications (default is 500)

Value

`Estimates`	Estimates of the treatment effect on the priamry outcome, delta, the treatment effect on the transformation of the surrogate, delta.gs, two versions of the proportion of treatment effect explained by the surrogate, pte1 and pte2; if var = TRUE, standard error estimates are also provided (se); if conf.int = TRUE, 95% confidence intervals are also provided
`Transformed.S`	the transformed surrogate, g(s), for each value of the surrogate, s; if var = TRUE, standard error estimates are also provided (se); if conf.int = TRUE, 95% confidence intervals are also provided

Examples

```
data(marker_disc)
out <- pte_disc(
  sob = marker_disc$sob,
  yob = marker_disc$yob,
  aob = marker_disc$aob, var = FALSE, conf.int = FALSE)
out

out <- pte_disc(
  sob = marker_disc$sob,
  yob = marker_disc$yob,
  aob = marker_disc$aob,
  var = TRUE, conf.int = TRUE, rep = 1000)
out
```

Index