Package ‘SurrogateParadoxTest’

July 21, 2025

Type Package

Title Empirical Testing of Surrogate Paradox Assumptions

Version 2.0

Date 2025-01-20

Description Provides functions to nonparametrically assess assumptions necessary to prevent the surrogate paradox through hypothesis tests of stochastic dominance, monotonicity of regression functions, and non-negative residual treatment effects. More details are available in Hsiao et al 2025 (under review). A tutorial for this package can be found at <`https://laylaparast.com/home/SurrogateParadoxTest.html`>.

License GPL

Imports stats, MonotonicityTest

NeedsCompilation no

Author Emily Hsiao [aut, cre]

Maintainer Emily Hsiao <`ehsiao@utexas.edu`>

Repository CRAN

Date/Publication 2025-01-20 23:30:02 UTC

Contents

`test_assumptions` *Test assumptions to prevent surrogate paradox*

Description

Tests the assumptions necessary to prevent the surrogate paradox: stochastic dominance of surrogate values in the treatment group over control group, monotonicity of the relationship between surrogate and primary endpoint in both treatment and control group, and non-negative residual treatment effect of the treatment group over the control group. For computational efficiency, Version 2.0 of this package uses the monotonicity_test function from the MonotonicityTest package.

Usage

```
test_assumptions(s0 = NULL, y0 = NULL, s1 = NULL, y1 = NULL, trim = 0.95,
alpha = 0.05, type = "all", all_results = TRUE, direction = "positive",
monotonicity_bootstrap_n = 100, nnr_bootstrap_n = 200)
```

Arguments

`s0`	Vector of surrogate values in control group.
`y0`	Vector of primary endpoint values in control group.
`s1`	Vector of surrogate values in treatment group.
`y1`	Vector of primary endpoint values in treatment group.
`trim`	Proportion of data to keep after trimming the outliers. Defaults to 95%. Trims data by sorting by surrogate value and removing (1 - trim)/2 % of the lowest and highest surrogate values with their corresponding primary endpoint values.
`alpha`	Desired alpha level of tests.
`type`	Type of test to run. Defaults to "all"; possible inputs are "sd" (stochastic dominance), "monotonicity" (monotonicity), and "nnr" (non-negative residual treatment effect).
`all_results`	TRUE or FALSE; return all outputs from hypothesis tests. Defaults to TRUE.
`direction`	Direction of the test. Defaults to "positive", which tests that the treatment group stochastically dominates the control group, that μ_0 and μ_1 are monotonically increasing, and that $\mu_0 \leq \mu_1 \forall s$. Parameter "negative" tests that the control group stochastically dominates the treatment group, that μ_0 and μ_1 are monotonically decreasing, and that $\mu_1 \leq \mu_0 \forall s$.
`monotonicity_bootstrap_n`	Number of bootstrap samples for monotonicity test.
`nnr_bootstrap_n`	Number of bootstrap samples for nnr test.

Value

`result`	Table or string of results of the tests
`sd_result`	Detailed results of stochastic dominance test; only returned if all_results is TRUE
`monotonicity0_result`	Detailed results of monotonicity test in control group; only returned if all_results is TRUE
`monotonicity1_result`	Detailed results of monotonicity test in treatment group; only returned if all_results is TRUE
`nnr_result`	Detailed results of nnr test; only returned if all_results is TRUE

Author(s)

Emily Hsiao

References

Barrett, Garry F., and Stephen G. Donald. "Consistent tests for stochastic dominance." Econometrica 71.1 (2003): 71-104.

Hall, Peter, and Nancy E. Heckman. "Testing for monotonicity of a regression mean by calibrating for linear functions." Annals of Statistics (2000): 20-39.

Hsiao, Tian, Parast. "Avoiding the Surrogate Paradox: An Empirical Framework for Assessing Assumptions." 2025 (Under Review)

Examples

```
m_c <- function(s) 1 + 2 * s
m_t <- function(s) 1 + 2 * s

s_c <- rnorm(100, 3, 1)
y_c <- sapply(s_c, function(s) rnorm(1, m_c(s), 1))
s_t <- rnorm(100, 3, 1)
y_t <- sapply(s_t, function(s) rnorm(1, m_t(s), 1))

test_assumptions(
s0 = s_c, y0 = y_t, s1 = s_t, y1 = y_t, type = "sd"
)


test_assumptions(
s0 = s_c, y0 = y_t, s1 = s_t, y1 = y_t, type = "all")
```

Index