Package 'obshetsurr'

February 21, 2025

Type Package

Title Assessing Surrogate Heterogeneity in Observational Data

Version 1.0

Date 2025-02-20

Author Rebecca Knowlton

Maintainer Rebecca Knowlton <rknowlton@utexas.edu>

Description Provides functions to estimate the proportion of the treatment effect explained (PTE) for a surrogate as a function of multiple baseline covariates in an observational setting, with options for different base learners. More details will be available in the future in: Knowlton, R., Parast, L. (2025) ``Assessing Surrogate Heterogeneity in Real World Data Using Meta-Learners."

License GPL

Imports stats, matrixStats, mgcv, grf

NeedsCompilation no

Depends R (>= 3.5.0)

R topics documented:

`boot.var` *Calculate bootstrapped variance estimates.*

Description

Calculates bootstrapped variance estimates of delta, delta.s, and R.s, and optionally calculates p-values for identifying strong surrogates.

Usage

```
boot.var(df.train, df.test, type, numeric_predictors, categorical_predictors,
  threshold, use.actual.control.S)
```

Arguments

df.train	A dataframe containing training data.
df.test	A dataframe containing testing data.
type	Options are "linear", "gam", "trees", or "all"; type of base learners to use.
numeric_predictors	The column names in the dataframes that represent numeric baseline covariates.
categorical_predictors	The column names in the dataframes that represent categorical baseline covariates.
threshold	An optional threshold to test individuals for the null hypothesis that PTE > threshold.
use.actual.control.S	TRUE or FALSE, if user prefers to use the actual observed values for the surrogate in the control group instead of predicting values from the base learners.

Value

A dataframe is returned, which is the df.test argument with new columns appended for the estimated variances of delta, delta.s, and R.s, as well as p-values if a threshold is provided.

---

estimate.PTE *Estimate the proportion of the treatment effect explained*

Description

Fits base learners using the specified type of model on the training data, and uses those models to calculate delta, delta.s, and R.s on the testing set.

Usage

```
estimate.PTE(df.train, df.test, type, numeric_predictors, categorical_predictors,
  use.actual.control.S)
```

Arguments

df.train	A dataframe containing training data.
df.test	A dataframe containing testing data.
type	Options are "linear", "gam", "trees", or "all"; type of base learners to use.
numeric_predictors	The column names in the dataframes that represent numeric baseline covariates.
categorical_predictors	The column names in the dataframes that represent categorical baseline covariates.
use.actual.control.S	TRUE or FALSE, if user prefers to use the actual observed values for the surrogate in the control group instead of predicting values from the base learners.

Value

A dataframe is returned, which is the df.test argument with new columns appended for the estimates of delta, delta.s, and R.s.

`exampledata_test` *Example testing data*

Description

Example testing data

Usage

```
data("exampledata_test")
```

Format

A data frame with 200 observations on the following 9 variables.

`X1` a numeric baseline covariate of interest

`X2` a numeric baseline covariate of interest

`X3` a numeric baseline covariate of interest

`X4` a numeric baseline covariate of interest

`X5` a numeric baseline covariate of interest

`X6` a numeric baseline covariate of interest

`G` the treatment assignment, where 1 indicates treated and 0 indicates control

`S` the surrogate marker

`Y` the primary outcome

Examples

```
data(exampledata_test)
names(exampledata_test)
```

`exampledata_train` *Example training data*

Description

Example training data

Usage

```
data("exampledata_train")
```

Format

A data frame with 1800 observations on the following 9 variables.

`X1` a numeric baseline covariate of interest

`X2` a numeric baseline covariate of interest

`X3` a numeric baseline covariate of interest

`X4` a numeric baseline covariate of interest

`X5` a numeric baseline covariate of interest

`X6` a numeric baseline covariate of interest

`G` the treatment assignment, where 1 indicates treated and 0 indicates control

`S` the surrogate marker

`Y` the primary outcome

Examples

```
data(exampledata_train)
names(exampledata_train)
```

`obs.het.surr` *Estimate the proportion of the treatment effect explained by the surrogate marker as a function of multiple baseline covariates in an observational setting*

Description

Assesses surrogate heterogeneity in real world data by estimating the proportion of the treatment effect explained as a function of baseline covariates. Optionally tests individuals for strong surrogacy based on a threshold.

Usage

```
obs.het.surr(df.train, df.test, type, var.want = FALSE, threshold = NULL,
  use.actual.control.S = FALSE)
```

Arguments

`df.train`	A dataframe containing training data. Must have columns G (treatment assignment), S (surrogate marker), and Y (primary outcome), in addition to the baseline covariates of interest.
`df.test`	A dataframe containing testing data. Must contain the same baseline covariate columns as the training data.
`type`	Options are "linear", "gam", "trees", or "all"; type of base learners to use.
`var.want`	TRUE or FALSE, if variance estimates are wanted.
`threshold`	An optional threshold to test individuals for the null hypothesis that PTE > threshold; must have var.want = TRUE to return p-values.
`use.actual.control.S`	TRUE or FALSE, if user prefers to use the actual observed values for the surrogate in the control group instead of predicting values from the base learners.

Value

A dataframe is returned, which is the df.test argument with new columns appended for the estimates and corresponding variances of delta, delta.s, and R.s. If a threshold is specified, returns a p-value for the null hypothesis that PTE > threshold.

Author(s)

Rebecca Knowlton

References

Knowlton, R. and Parast, L. (2025) “Assessing Surrogate Heterogeneity in Real World Data Using Meta-Learners." Under Review.

Examples

```
data(exampledata_train)
data(exampledata_test)
obs.het.surr(df.train = exampledata_train, df.test = exampledata_test, type = "all", var.want = FALSE)
```

Index