Package: CausalMetaR (via r-universe)

September 1, 2024

Type Package

Title Causally Interpretable Meta-Analysis

Version 0.1.2

Description Provides robust and efficient methods for estimating causal effects in a target population using a multi-source dataset, including those of Dahabreh et al. (2019) <doi:10.1111/biom.13716>, Robertson et al. (2021) <doi:10.48550/arXiv.2104.05905>, and Wang et al. (2024) <doi:10.48550/arXiv.2402.02684>. The multi-source data can be a collection of trials, observational studies, or a combination of both, which have the same data structure (outcome, treatment, and covariates). The target population can be based on an internal dataset or an external dataset where only covariate information is available. The causal estimands available are average treatment effects and subgroup treatment effects. See Wang et al. (2024) <doi:10.48550/arXiv.2402.04341> for a detailed guide on using the package.

License GPL (>=3)

Encoding UTF-8

LazyData true

Imports glmnet, metafor, nnet, progress, SuperLearner

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

URL https://github.com/ly129/CausalMetaR, https://arxiv.org/abs/2402.04341

BugReports https://github.com/ly129/CausalMetaR/issues

Depends R (>= 2.10)

Repository https://ly129.r-universe.dev

RemoteUrl https://github.com/ly129/causalmetar

RemoteRef HEAD

RemoteSha fcfb16bf7f3968b3441609a7cec304e59eb1f086

Contents

`ATE_external` *Estimating the Average Treatment Effect (ATE) in an external target population using multi-source data*

Description

Doubly-robust and efficient estimator for the ATE in an external target population using multi-source data.

Usage

```
ATE_external(
  X,
  X_external,
  Y,
  S,
  A,
  cross_fitting = FALSE,
  replications = 10L,
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(),
  external_model_args = list(),
  outcome_model_args = list(),
  show_progress = TRUE
)
```

Arguments

`X`	Data frame (or matrix) containing the covariate data in the multi-source data. It should have n rows and p columns. Character variables will be converted to factors.

`X_external` Data frame (or matrix) containing the covariate data in the external target population. It should have n_0 rows and p columns. This is the external data counterpart to the `X` argument.

`Y` Vector of length n containing the outcome.

`S` Vector of length n containing the source indicator. If `S` is a factor, it will maintain its level order; otherwise it will be converted to a factor with the default level order. The order will be carried over to the outputs and plots.

`A` Vector of length n containing the binary treatment (1 for treated and 0 for untreated).

`cross_fitting` Logical specifying whether sample splitting and cross fitting should be used.

`replications` Integer specifying the number of sample splitting and cross fitting replications to perform, if `cross_fitting = TRUE`. The default is `10L`.

`source_model` Character string specifying the (penalized) multinomial logistic regression for estimating the source model. It has two options: `"MN.glmnet"` (default) and `"MN.nnet"`, which use **glmnet** and **nnet** respectively.

`source_model_args`
List specifying the arguments for the source model (in **glmnet** or **nnet**).

`treatment_model_type`
Character string specifying how the treatment model is estimated. Options include `"separate"` (default) and `"joint"`. If `"separate"`, the treatment model (i.e., $P(A = 1|X, S = s)$) is estimated by regressing A on X within each specific internal population $S = s$. If `"joint"`, the treatment model is estimated by regressing A on X and S using the multi-source population.

`treatment_model_args`
List specifying the arguments for the treatment model (in **SuperLearner**).

`external_model_args`
List specifying the arguments for the external model (in **SuperLearner**).

`outcome_model_args`
List specifying the arguments for the outcome model (in **SuperLearner**).

`show_progress` Logical specifying whether to print a progress bar for the cross-fit replicates completed, if `cross_fitting = TRUE`.

Details

Data structure:

The multi-source dataset consists the outcome `Y`, source `S`, treatment `A`, and covariates `X` ($n \times p$) in the internal populations. The data sources can be trials, observational studies, or a combination of both.

The external dataset contains only covariates `X_external` ($n_0 \times p$).

Estimation of nuissance parameters:

The following models are fit:

- External model: $q(X) = P(R = 1|X)$, where R takes value 1 if the subject belongs to any of the internal dataset and 0 if the subject belongs to the external dataset

- Propensity score model: $\eta_a(X) = P(A = a|X)$. We perform the decomposition $P(A = a|X) = \sum_s P(A = a|X, S = s)P(S = s|X)$ and estimate $P(A = 1|X, S = s)$ (i.e., the treatment model) and $P(S = s|X)$ (i.e., the source model).
- Outcome model: $\mu_a(X) = E(Y|X, A = a)$

The models are estimated by **SuperLearner** with the exception of the source model which is estimated by **glmnet** or **nnet**.

ATE estimation:

The ATE estimator is

$$\frac{\widehat{\kappa}}{N}\sum_{i=1}^{N}\left[I(R_i = 0)\widehat{\mu}_a(X_i) + I(A_i = a, R_i = 1)\frac{1-\widehat{q}(X_i)}{\widehat{\eta}_a(X_i)\widehat{q}(X_i)}\Big\{Y_i - \widehat{\mu}_a(X_i)\Big\}\right],$$

where $N = n + n_0$, and $\widehat{\kappa} = \{N^{-1}\sum_{i=1}^{N} I(R_i = 0)\}^{-1}$.

This estimator is doubly robust and non-parametrically efficient. To achieve non-parametric efficiency and asymptotic normality, it requires that $||\widehat{\mu}_a(X) - \mu_a(X)||\big\{||\widehat{\eta}_a(X) - \eta_a(X)|| + ||\widehat{q}(X) - q(X)||\big\} = o_p(n^{-1/2})$. In addition, sample splitting and cross-fitting can be performed to avoid the Donsker class assumption.

When a data source is a randomized trial, it is still recommended to estimate the propensity score for optimal efficiency.

Value

An object of class "ATE_external". This object is a list with the following elements:

`df_dif`	A data frame containing the treatment effect (mean difference) estimates for the extenal data.
`df_A0`	A data frame containing the potential outcome mean estimates under A = 0 for the extenal data.
`df_A1`	A data frame containing the potential outcome mean estimates under A = 1 for the extenal data.
`fit_outcome`	Fitted outcome model.
`fit_source`	Fitted source model.
`fit_treatment`	Fitted treatment model(s).
`fit_external`	Fitted external model.

References

Dahabreh, I.J., Robertson, S.E., Petito, L.C., Hernán, M.A. and Steingrimsson, J.A. (2019) *Efficient and robust methods for causally interpretable meta-analysis: Transporting inferences from multiple randomized trials to a target population*, Biometrics.

Wang, G., McGrath, S., Lian, Y. and Dahabreh, I. (2024) *CausalMetaR: An R package for performing causally interpretable meta-analyses*, arXiv preprint arXiv:2402.04341.

Examples

```
ae <- ATE_external(
  X = dat_multisource[, 1:10],
  Y = dat_multisource$Y,
  S = dat_multisource$S,
  A = dat_multisource$A,
  X_external = dat_external[, 1:10],
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  external_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

ATE_internal *Estimating the Average Treatment Effect (ATE) in an internal target population using multi-source data*

Description

Doubly-robust and efficient estimator for the ATE in each internal target population using multi-source data.

Usage

```
ATE_internal(
  X,
  Y,
  S,
  A,
  cross_fitting = FALSE,
  replications = 10L,
  source_model = "MN.glmnet",
  source_model_args = list(),
```

```
  treatment_model_type = "separate",
  treatment_model_args = list(),
  outcome_model_args = list(),
  show_progress = TRUE
)
```

Arguments

X	Data frame (or matrix) containing the covariate data in the multi-source data. It should have n rows and p columns. Character variables will be converted to factors.
Y	Vector of length n containing the outcome.
S	Vector of length n containing the source indicator. If S is a factor, it will maintain its level order; otherwise it will be converted to a factor with the default level order. The order will be carried over to the outputs and plots.
A	Vector of length n containing the binary treatment (1 for treated and 0 for untreated).
cross_fitting	Logical specifying whether sample splitting and cross fitting should be used.
replications	Integer specifying the number of sample splitting and cross fitting replications to perform, if cross_fitting = TRUE. The default is 10L.
source_model	Character string specifying the (penalized) multinomial logistic regression for estimating the source model. It has two options: "MN.glmnet" (default) and "MN.nnet", which use **glmnet** and **nnet** respectively.
source_model_args	List specifying the arguments for the source model (in **glmnet** or **nnet**).
treatment_model_type	Character string specifying how the treatment model is estimated. Options include "separate" (default) and "joint". If "separate", the treatment model (i.e., $P(A = 1\|X, S = s)$) is estimated by regressing A on X within each specific internal population $S = s$. If "joint", the treatment model is estimated by regressing A on X and S using the multi-source population.
treatment_model_args	List specifying the arguments for the treatment model (in **SuperLearner**).
outcome_model_args	List specifying the arguments for the outcome model (in **SuperLearner**).
show_progress	Logical specifying whether to print a progress bar for the cross-fit replicates completed, if cross_fitting = TRUE.

Details

Data structure:

The multi-source dataset consists the outcome Y, source S, treatment A, and covariates X ($n \times p$) in the internal populations. The data sources can be trials, observational studies, or a combination of both.

Estimation of nuissance parameters:

The following models are fit:

- Propensity score model: $\eta_a(X) = P(A = a|X)$. We perform the decomposition $P(A = a|X) = \sum_s P(A = a|X, S = s)P(S = s|X)$ and estimate $P(A = 1|X, S = s)$ (i.e., the treatment model) and $q_s(X) = P(S = s|X)$ (i.e., the source model).
- Outcome model: $\mu_a(X) = E(Y|X, A = a)$

The models are estimated by **SuperLearner** with the exception of the source model which is estimated by **glmnet** or **nnet**.

ATE estimation:

The ATE estimator is

$$\frac{\widehat{\kappa}}{n}\sum_{i=1}^{n}\left[I(S_i = s)\widehat{\mu}_a(X_i) + I(A_i = a)\frac{\widehat{q}_s(X_i)}{\widehat{\eta}_a(X_i)}\Big\{Y_i - \widehat{\mu}_a(X_i)\Big\}\right],$$

where $\widehat{\kappa} = \{n^{-1}\sum_{i=1}^{n} I(S_i = s)\}^{-1}$. The estimator is doubly robust and non-parametrically efficient.

To achieve non-parametric efficiency and asymptotic normality, it requires that $||\widehat{\mu}_a(X) - \mu_a(X)||\big\{||\widehat{\eta}_a(X) - \eta_a(X)|| + ||\widehat{q}_s(X) - q_s(X)||\big\} = o_p(n^{-1/2})$. In addition, sample splitting and cross-fitting can be performed to avoid the Donsker class assumption.

When a data source is a randomized trial, it is still recommended to estimate the propensity score for optimal efficiency.

Value

An object of class "ATE_internal". This object is a list with the following elements:

`df_dif`	A data frame containing the treatment effect (mean difference) estimates for the internal populations.
`df_A0`	A data frame containing the potential outcome mean estimates under A = 0 for the internal populations.
`df_A1`	A data frame containing the potential outcome mean estimates under A = 1 for the internal populations.
`fit_outcome`	Fitted outcome model.
`fit_source`	Fitted source model.
`fit_treatment`	Fitted treatment model(s).

References

Robertson, S.E., Steingrimsson, J.A., Joyce, N.R., Stuart, E.A. and Dahabreh, I.J. (2021). *Center-specific causal inference with multicenter trials: Reinterpreting trial evidence in the context of each participating center*. arXiv preprint arXiv:2104.05905.

Wang, G., McGrath, S., Lian, Y. and Dahabreh, I. (2024) *CausalMetaR: An R package for performing causally interpretable meta-analyses*, arXiv preprint arXiv:2402.04341.

Examples

```
ai <- ATE_internal(
  X = dat_multisource[, 1:10],
  Y = dat_multisource$Y,
  S = dat_multisource$S,
  A = dat_multisource$A,
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

---

dat_external *External dataset*

Description

Simulated external dataset.

Usage

```
dat_external
```

Format

A data frame with 10,083 rows and 13 columns. The columns are:

EM	Effect modifier.
X2, ..., X10	Covariates.
S	Source indicator.
A	Treatment (1 for treated and 0 for untreated).
Y	Outcome.

dat_multisource *Multi-source dataset*

Description

Simulated multi-source dataset consisting of 3 sources.

Usage

```
dat_multisource
```

Format

A data frame with 3,917 rows and 13 columns. The columns are:

EM	Effect modifier.
X2, ..., X10	Covariates.
S	Source indicator.
A	Treatment (1 for treated and 0 for untreated).
Y	Outcome.

plot.ATE_internal *Plot method for objects of class "ATE_internal"*

Description

This function creates forest plots of objects of class "ATE_internal".

Usage

```
## S3 method for class 'ATE_internal'
plot(x, source_names, ...)
```

Arguments

x	Object of class "ATE_internal".
source_names	optional, vector of character strings specifying the names of the sources. Defaults are the values in S provided by the user to ATE_internal.
...	Other arguments, which are passed to forest.rma.

Value

No value is returned.

See Also

ATE_internal

Examples

```
ai <- ATE_internal(
  X = dat_multisource[, 1:10],
  Y = dat_multisource$Y,
  S = dat_multisource$S,
  A = dat_multisource$A,
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
plot(ai)
```

---

plot.STE_internal *Plot method for objects of class "STE_internal"*

Description

This function creates forest plots of objects of class "STE_internal".

Usage

```
## S3 method for class 'STE_internal'
plot(
  x,
  use_scb = FALSE,
  header = c("Source", "Subgroup", ifelse(use_scb, "Estimate [95% SCB]",
    "Estimate [95% CI]")),
  source_names,
  subgroup_names,
  ...
)
```

Arguments

x	Object of class "STE_internal".
use_scb	logical scalar specifying whether the intervals in the forest plot should be simultaneous confidence bands (rather than confidence intervals). The default is FALSE.
header	optional, vector of character strings of length 3, headers for the source, effect modifier subgroup and the estimates in the forest plot.
source_names	optional, vector of character strings specifying the names of the sources. Defaults are the values in S provided by the user to STE_internal.
subgroup_names	optional, vector of character strings specifying the names of the effect modifier subgroups. Defaults are the values in EM provided by the user to STE_internal.
...	Other arguments, which are passed to forest.rma.

Details

Note that users may need to custom set the argument ilab.xpos which specifies the position (along the x-axis) of the effect modifier header and subgroup labels. See forest.rma for further details.

Value

No value is returned.

See Also

STE_internal

Examples

```
si <- STE_internal(
  X = dat_multisource[, 2:10],
  Y = dat_multisource$Y,
  EM = dat_multisource$EM,
  S = dat_multisource$S,
  A = dat_multisource$A,
  cross_fitting = FALSE,
  source_model = "MN.nnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

```
plot(si)
```

`print.STE_internal` *Print method for objects of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external"*

Description

Print method for objects of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external"

Usage

```
## S3 method for class 'STE_internal'
print(x, digits = 4, ...)

## S3 method for class 'ATE_internal'
print(x, digits = 4, ...)

## S3 method for class 'STE_external'
print(x, digits = 4, ...)

## S3 method for class 'ATE_external'
print(x, digits = 4, ...)
```

Arguments

`x`	Object of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external".
`digits`	Integer specifying the number of decimal places to display.
`...`	Other arguments (ignored).

Value

No value is returned.

See Also

`ATE_internal`, `ATE_external`, `STE_internal`, `STE_external`

Examples

```
si <- STE_internal(
  X = dat_multisource[, 2:10],
  Y = dat_multisource$Y,
  EM = dat_multisource$EM,
  S = dat_multisource$S,
  A = dat_multisource$A,
```

```
  cross_fitting = FALSE,
  source_model = "MN.nnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
print(si)
```

STE_external *Estimating the Subgroup Treatment Effect (STE) in an external target population using multi-source data*

Description

Doubly-robust and efficient estimator for the STE in an external target population using multi-source data.

Usage

```
STE_external(
  X,
  X_external,
  EM,
  EM_external,
  Y,
  S,
  A,
  cross_fitting = FALSE,
  replications = 10L,
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(),
  external_model_args = list(),
  outcome_model_args = list(),
  show_progress = TRUE
)
```

Arguments

X	Data frame (or matrix) containing the covariate data in the multi-source data. It should have n rows and p columns. Character variables will be converted to factors.
X_external	Data frame (or matrix) containing the covariate data in the external target population. It should have n_0 rows and p columns. This is the external data counterpart to the X argument.
EM	Vector of length n containing the effect modifier in the multi-source data. If EM is a factor, it will maintain its subgroup level order; otherwise it will be converted to a factor with default level order.
EM_external	Vector of length n_0 containing the effect modifier in the external data. This is the external data counterpart to the EM argument.
Y	Vector of length n containing the outcome.
S	Vector of length n containing the source indicator. If S is a factor, it will maintain its level order; otherwise it will be converted to a factor with the default level order. The order will be carried over to the outputs and plots.
A	Vector of length n containing the binary treatment (1 for treated and 0 for untreated).
cross_fitting	Logical specifying whether sample splitting and cross fitting should be used.
replications	Integer specifying the number of sample splitting and cross fitting replications to perform, if cross_fitting = TRUE. The default is 10L.
source_model	Character string specifying the (penalized) multinomial logistic regression for estimating the source model. It has two options: "MN.glmnet" (default) and "MN.nnet", which use **glmnet** and **nnet** respectively.
source_model_args	List specifying the arguments for the source model (in **glmnet** or **nnet**).
treatment_model_type	Character string specifying how the treatment model is estimated. Options include "separate" (default) and "joint". If "separate", the treatment model (i.e., $P(A = 1\|X, S = s)$) is estimated by regressing A on X within each specific internal population $S = s$. If "joint", the treatment model is estimated by regressing A on X and S using the multi-source population.
treatment_model_args	List specifying the arguments for the treatment model (in **SuperLearner**).
external_model_args	List specifying the arguments for the external model (in **SuperLearner**).
outcome_model_args	List specifying the arguments for the outcome model (in **SuperLearner**).
show_progress	Logical specifying whether to print a progress bar for the cross-fit replicates completed, if cross_fitting = TRUE.

Details

Data structure:

The multi-source dataset consists the outcome `Y`, source `S`, treatment `A`, covariates `X` ($n \times p$), and effect modifier `EM` in the internal populations. The data sources can be trials, observational studies, or a combination of both.

The external dataset contains only covariates `X_external` ($n_0 \times p$) and the effect modifier `EM_external`.

Estimation of nuissance parameters:

The following models are fit:

- External model: $q(X) = P(R = 1|X)$, where R takes value 1 if the subject belongs to any of the internal dataset and 0 if the subject belongs to the external dataset
- Propensity score model: $\eta_a(X) = P(A = a|X)$. We perform the decomposition $P(A = a|X) = \sum_s P(A = a|X, S = s)P(S = s|X)$ and estimate $P(A = 1|X, S = s)$ (i.e., the treatment model) and $P(S = s|X)$ (i.e., the source model).
- Outcome model: $\mu_a(X) = E(Y|X, A = a)$

The models are estimated by **SuperLearner** with the exception of the source model which is estimated by **glmnet** or **nnet**.

STE estimation:

The estimator is

$$\frac{\widehat{\kappa}}{N}\sum_{i=1}^{N}\left[I(R_i = 0)\widehat{\mu}_a(X_i) + I(A_i = a, R_i = 1)\frac{1 - \widehat{q}(X_i)}{\widehat{\eta}_a(X_i)\widehat{q}(X_i)}\Big\{Y_i - \widehat{\mu}_a(X_i)\Big\}\right],$$

where $N = n + n_0$, $\widehat{\kappa} = \{N^{-1}\sum_{i=1}^{N} I(R_i = 0)\}^{-1}$, and and $\widetilde{X}$ denotes the effect modifier.

The estimator is doubly robust and non-parametrically efficient. To achieve non-parametric efficiency and asymptotic normality, it requires that $||\widehat{\mu}_a(X) - \mu_a(X)||\big\{||\widehat{\eta}_a(X) - \eta_a(X)|| + ||\widehat{q}(X) - q(X)||\big\} = o_p(n^{-1/2})$. In addition, sample splitting and cross-fitting can be performed to avoid the Donsker class assumption.

When a data source is a randomized trial, it is still recommended to estimate the propensity score for optimal efficiency.

Value

An object of class "STE_external". This object is a list with the following elements:

`df_dif`	A data frame containing the subgroup treatment effect (mean difference) estimates for the extenal data.
`df_A0`	A data frame containing the subgroup potential outcome mean estimates under A = 0 for the extenal data.
`df_A1`	A data frame containing the subgroup potential outcome mean estimates under A = 1 for the extenal data.
`fit_outcome`	Fitted outcome model.
`fit_source`	Fitted source model.
`fit_treatment`	Fitted treatment model(s).
`fit_external`	Fitted external model.

References

Wang, G., Levis, A., Steingrimsson, J. and Dahabreh, I. (2024) *Efficient estimation of subgroup treatment effects using multi-source data*, arXiv preprint arXiv:2402.02684.

Wang, G., McGrath, S., Lian, Y. and Dahabreh, I. (2024) *CausalMetaR: An R package for performing causally interpretable meta-analyses*, arXiv preprint arXiv:2402.04341.

Examples

```
se <- STE_external(
  X = dat_multisource[, 2:10],
  Y = dat_multisource$Y,
  EM = dat_multisource$EM,
  S = dat_multisource$S,
  A = dat_multisource$A,
  X_external = dat_external[, 2:10],
  EM_external = dat_external$EM,
  cross_fitting = FALSE,
  source_model = "MN.nnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  external_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

STE_internal *Estimating the Subgroup Treatment Effect (STE) in an internal target population using multi-source data*

Description

Doubly-robust and efficient estimator for the STE in each internal target population using multi-source data.

Usage

```
STE_internal(
  X,
  Y,
  EM,
  S,
  A,
  cross_fitting = FALSE,
  replications = 10L,
  source_model = "MN.glmnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(),
  outcome_model_args = list(),
  show_progress = TRUE
)
```

Arguments

X	Data frame (or matrix) containing the covariate data in the multi-source data. It should have n rows and p columns. Character variables will be converted to factors.
Y	Vector of length n containing the outcome.
EM	Vector of length n containing the effect modifier in the multi-source data. If EM is a factor, it will maintain its subgroup level order; otherwise it will be converted to a factor with default level order.
S	Vector of length n containing the source indicator. If S is a factor, it will maintain its level order; otherwise it will be converted to a factor with the default level order. The order will be carried over to the outputs and plots.
A	Vector of length n containing the binary treatment (1 for treated and 0 for untreated).
cross_fitting	Logical specifying whether sample splitting and cross fitting should be used.
replications	Integer specifying the number of sample splitting and cross fitting replications to perform, if cross_fitting = TRUE. The default is 10L.
source_model	Character string specifying the (penalized) multinomial logistic regression for estimating the source model. It has two options: "MN.glmnet" (default) and "MN.nnet", which use **glmnet** and **nnet** respectively.
source_model_args	List specifying the arguments for the source model (in **glmnet** or **nnet**).
treatment_model_type	Character string specifying how the treatment model is estimated. Options include "separate" (default) and "joint". If "separate", the treatment model (i.e., $P(A = 1\|X, S = s)$) is estimated by regressing A on X within each specific internal population $S = s$. If "joint", the treatment model is estimated by regressing A on X and S using the multi-source population.

`treatment_model_args`
: List specifying the arguments for the treatment model (in **SuperLearner**).

`outcome_model_args`
: List specifying the arguments for the outcome model (in **SuperLearner**).

`show_progress`
: Logical specifying whether to print a progress bar for the cross-fit replicates completed, if `cross_fitting = TRUE`.

Details

Data structure:

The multi-source dataset consists the outcome `Y`, source `S`, treatment `A`, covariates `X` ($n \times p$), and effect modifier `EM` in the internal populations. The data sources can be trials, observational studies, or a combination of both.

Estimation of nuissance parameters:

The following models are fit:

- Propensity score model: $\eta_a(X) = P(A = a|X)$. We perform the decomposition $P(A = a|X) = \sum_s P(A = a|X, S = s)P(S = s|X)$ and estimate $P(A = 1|X, S = s)$ (i.e., the treatment model) and $q_s(X) = P(S = s|X)$ (i.e., the source model).
- Outcome model: $\mu_a(X) = E(Y|X, A = a)$

The models are estimated by **SuperLearner** with the exception of the source model which is estimated by **glmnet** or **nnet**.

STE estimation:

The estimator is

$$\frac{\widehat{\kappa}}{n}\sum_{i=1}^{n}\left[I(S_i = s, \widetilde{X}_i = \widetilde{x})\widehat{\mu}_a(X_i) + I(A_i = a, \widetilde{X}_i = \widetilde{x})\frac{\widehat{q}_s(X_i)}{\widehat{\eta}_a(X_i)}\left\{Y_i - \widehat{\mu}_a(X_i)\right\}\right],$$

where $\widehat{\kappa} = \{n^{-1}\sum_{i=1}^{n} I(S_i = s, \widetilde{X}_i = \widetilde{x})\}^{-1}$ and $\widetilde{X}$ denotes the effect modifier.

The estimator is doubly robust and non-parametrically efficient. To achieve non-parametric efficiency and asymptotic normality, it requires that $||\widehat{\mu}_a(X) - \mu_a(X)||\{||\widehat{\eta}_a(X) - \eta_a(X)|| + ||\widehat{q}_s(X) - q_s(X)||\} = o_p(n^{-1/2})$. In addition, sample splitting and cross-fitting can be performed to avoid the Donsker class assumption.

When a data source is a randomized trial, it is still recommended to estimate the propensity score for optimal efficiency.

Value

An object of class "STE_internal". This object is a list with the following elements:

`df_dif`
: A data frame containing the subgroup treatment effect (mean difference) estimates for the internal populations.

`df_A0`
: A data frame containing the subgroup potential outcome mean estimates under A = 0 for the internal populations.

`df_A1`
: A data frame containing the subgroup potential outcome mean estimates under A = 1 for the internal populations.

fit_outcome	Fitted outcome model.
fit_source	Fitted source model.
fit_treatment	Fitted treatment model(s).
...	Some additional elements.

References

Wang, G., Levis, A., Steingrimsson, J. and Dahabreh, I. (2024) *Efficient estimation of subgroup treatment effects using multi-source data*, arXiv preprint arXiv:2402.02684.

Wang, G., McGrath, S., Lian, Y. and Dahabreh, I. (2024) *CausalMetaR: An R package for performing causally interpretable meta-analyses*, arXiv preprint arXiv:2402.04341.

Examples

```
si <- STE_internal(
  X = dat_multisource[, 2:10],
  Y = dat_multisource$Y,
  EM = dat_multisource$EM,
  S = dat_multisource$S,
  A = dat_multisource$A,
  cross_fitting = FALSE,
  source_model = "MN.nnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

---

summary.STE_internal *Summary method for objects of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external"*

Description

Summary method for objects of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external"

Usage

```
## S3 method for class 'STE_internal'
summary(object, digits = 4, ...)

## S3 method for class 'STE_external'
summary(object, digits = 4, ...)

## S3 method for class 'ATE_external'
summary(object, digits = 4, ...)

## S3 method for class 'ATE_internal'
summary(object, digits = 4, ...)
```

Arguments

object	Object of class "ATE_internal", "ATE_external", "STE_internal", or "STE_external".
digits	Integer specifying the number of decimal places to display.
...	Other arguments.

Value

No value is returned.

See Also

ATE_internal, ATE_external, STE_internal, STE_external

Examples

```
si <- STE_internal(
  X = dat_multisource[, 2:10],
  Y = dat_multisource$Y,
  EM = dat_multisource$EM,
  S = dat_multisource$S,
  A = dat_multisource$A,
  cross_fitting = FALSE,
  source_model = "MN.nnet",
  source_model_args = list(),
  treatment_model_type = "separate",
  treatment_model_args = list(
    family = binomial(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  ),
  outcome_model_args = list(
    family = gaussian(),
    SL.library = c("SL.glmnet", "SL.nnet", "SL.glm"),
    cvControl = list(V = 5L)
  )
)
```

```
summary(si)
```

Index