Package 'inferference'

June 8, 2015

Type Package

Title Methods for Causal Inference with Interference

Version 0.4.62

Date 2015-06-06

Author Bradley Saul

Maintainer Bradley Saul <saulb@live.unc.edu>

Description Provides methods for estimating causal effects in the presence of interference. Currently it implements the IPW estimators proposed by E.J. Tchetgen Tchetgen and T.J. Vanderweele in ``On causal inference in the presence of interference'' (Statistical Methods in Medical Research, 21(1) 55-75).

Depends R (>= 3.1)

Imports numDeriv (>= 2012.9-1), lme4 (>= 1.1-6), Formula (>= 1.1-2)

License GPL (>= 2)

Suggests testthat, knitr

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-08 08:39:23

R topics documented:

`direct_effect` *Retreive Direct Effect estimates*

Description

Retrieves the population average direct causal effect for a specified allocation: $\hat{Y}(0, alpha) - \hat{Y}(1, alpha)$.

Usage

```
direct_effect(object, allocation = NULL, trt.lvl1 = 0)
```

Arguments

`object`	an object of class `interference`
`allocation`	the allocation scheme for which to estimate direct effects. If NULL, then returns all direct effects.
`trt.lvl1`	Defaults to 0.

Value

a data.frame with requested values

`indirect_effect` *Retreive Indirect Effect estimates*

Description

Retrieves the population average indirect causal effect for specified allocations: $\hat{Y}(0, alpha1) - \hat{Y}(0, alpha2)$. This is the effect due to the coverage (allocation) levels.

Retreive Indirect Effect estimates

Usage

```
indirect_effect(object, allocation1, allocation2 = NULL, trt.lvl = 0)

ie(object, allocation1, allocation2 = NULL, trt.lvl = 0)
```

Arguments

`object`	an object of class `interference`
`allocation1`	the allocation scheme for which to estimate indirect effects
`allocation2`	the allocation scheme for which to estimate indirect effects. If NULL, then returns all indirect effects compared to allocation1.
`trt.lvl`	Defaults to 0.

Value

a data.frame with requested values

`inferference` *Methods for causal inference with interference*

Description

Interference occurs when the treatment of one unit affects outcomes of other units. This package provides methods for estimating causal effects in the presence of interference. Currently it implements the IPW estimators proposed by Tchetgen Tchetgen and Vanderweele (2012). and developed further in Heydrich-Perez et al. (2014).

`interference` *Estimate Causal Effects in presence of interference*

Description

Estimate Causal Effects in presence of interference

Usage

```
interference(formula, propensity_integrand = "logit_integrand",
  loglihood_integrand = propensity_integrand, allocations, data,
  model_method = "glmer", model_options = list(family = binomial(link =
  "logit")), causal_estimation_method = "ipw",
  causal_estimation_options = list(set_NA_to_0 = TRUE, variance_estimation =
  "robust"), conf.level = 0.95, rescale.factor = 1, ...)
```

Arguments

`formula` The formula used to define the causal model. Has a minimum of 4 parts, separated by | and ~ in a specific structure: `outcome | exposure ~ propensity covariates | group`. The order matters, and the pipes split the data frame into corresponding pieces. The part separated by ~ is passed to the chosen `model_method` used to estimate or fix propensity parameters.

`propensity_integrand`
A function, which may be created by the user, used to compute the IP weights. This defaults to `logit_integrand`, which calculates the product of inverse logits for individuals in a group: $\prod_{j=1}^{n_i}\{r\times h_{ij}(b_i)^{A_{ij}}\}\{1-r\times h_{ij}(b_i)^{1-A_{ij}}\}f_b(b_i;\theta_s)$ where

$$h_{ij}(b_i) = logit^{-1}(\mathbf{X}_{ij}\theta_a + b_i)$$

and b_i is a group-level random effect, f_b is a $N(0,\theta_s)$ density, and r is a known randomization probability which may be useful if a participation vector is included in the `formula`. If no random effect was included in the `formula`,

logit_integrand essentially ignores the random effect and $f_b(b_i, \theta_s)$ integrates to 1. See details for arguments that can be passed to logit_integrand

loglihood_integrand
: A function, which may be created by the user, that defines the log likelihood of the logit model used for robust variance estimation. Generally, this will be the same function as propensity_integrand. Indeed, this is the default.

allocations
: a vector of values in [0, 1]. Increasing the number of elements of the allocation vector greatly increases computation time; however, a larger number of allocations will make plots look nicer. A minimum of two allocations is required.

data
: the analysis data frame. This must include all the variables defined in the formula.

model_method
: the method used to estimate or set the propensity model parameters. Must be one of 'glm', 'glmer', or 'oracle'. Defaults to 'glmer'. For a fixed effects only model use 'glm', and to include random effects use 'glmer'. logit_integrand only supports a single random effect for the grouping variable, so if more random effects are included in the model, different propensity_integrand and loglihood_integrand functions should be defined. When the propensity parameters are known (as in simulations) or if estimating parameters by other methods, use the 'oracle' option. See model_options for details on how to pass the oracle parameters.

model_options
: a list of options passed to the function in model_method. Defaults to list(family = binomial(link = '
When model_method = 'oracle', the list must have two elements fixed.effects and random.effects. If the model did not include random effects, set random.effects = NULL.

causal_estimation_method
: currently only supports 'ipw'.

causal_estimation_options
: A list with two slots. (1) variance_estimation is either 'naive' or 'robust'. See details. Defaults to 'robust'. (2) is set_NA_to_0. Defaults to TRUE. When, for example, group sizes reach over 1000, the product terms of the propensity diminish to zero. This may result in NaN values for the weights or loglihood. This option sets such cases to zero.

conf.level
: level for confidence intervals. Defaults to 0.95.

rescale.factor
: a scalar multiplication factor by which to rescale outcomes and effects. Defaults to 1.

...
: Used to pass additional arguments to internal functions such as numDeriv::grad() or integrate(). Additionally, arguments can be passed to the propensity_integrand and loglihood_integrand functions.

Details

The following formula includes a random effect for the group: outcome | exposure ~ propensity covariates + (1|grou
In this instance, the group variable appears twice. If the study design includes a "participation" variable, this is easily added to the formula: outcome | exposure | participation ~ propensity covariates | group.

logit_integrand has two options that can be passed via the ... argument:

- `randomization`: a scalar. This is the r in the formula just above. It defaults to 1 in the case that a `participation` vector is not included. The vaccine study example demonstrates use of this argument.
- `integrate_allocation`: `TRUE/FALSE`. When group sizes grow large (over 1000), the product term of `logit_integrand` tends quickly to 0. When set to `TRUE`, the IP weights tend less quickly to 0. Defaults to `FALSE`.

If the true propensity model is known (e.g. in simulations) use `variance_estimatation = 'naive'`; otherwise, use the default `variance_estimatation = 'robust'`. Refer to the web appendix of Heydrich-Perez et al. (2014) for complete details.

Value

Returns a list of overall and group-level IPW point estimates, overall and group-level IPW point estimates (using the weight derivatives), derivatives of the loglihood, the computed weight matrix, the computed weight derivative array, and a summary.

`logit_integrand` *Default integrand for the group-level propensity score*

Description

Computes the following function:

$$\prod_{j=1}^{n}(rh_j(b))^{A_j}(1-rh_j(b))^{1-A_j}f_b(b;\theta_b)$$

where r is the randomization scheme. X is the covariate(s) vectors. $fixef$ is the vector of fixed effects. b is the random (group-level) effect. $ranef$ is the random effect variance.

Usage

```
logit_integrand(b, X, A, fixed.effects, random.effects = NULL, x = NULL,
  pos = NULL, allocation = NULL, randomization = 1,
  integrate.allocation = FALSE)
```

Arguments

`b`	vector argument of values necessary for `integrate`.
`X`	n by length(fixed.effects) matrix of covariates.
`A`	vector of observed treatments (0,1)
`fixed.effects`	vector of fixed effect parameters.
`random.effects`	OPTIONAL vector of random effect parameters. If provided, only the first element is used. If this element is <= 0, it is ignored.
`x`	Used by `grad` for taking the derivative with respect an element of params. Only used if pos is not NULL.

pos	The position of theta for which to take the derivative. Defaults to NULL.
allocation	The allocation strategy. Required if include.allocations == TRUE. Defaults to NA.
randomization	Randomization probability. Defaults to 1.
integrate.allocation	Either TRUE for including allocation in the product or FALSE does not include allocation.

Value

value of the integrand

`overall_effect` *Retrieve Overall Effect Estimates*

Description

Retrieves the population average overall causal effect: $\hat{Y}(alpha1) - \hat{Y}(lpha2)$

Usage

```
overall_effect(object, allocation1, allocation2 = NULL)

oe(object, allocation1, allocation2 = NULL)
```

Arguments

object	an object of class interference
allocation1	the allocation scheme for which to estimate overall effects
allocation2	the allocation scheme for which to estimate overall effects

Value

a data.frame with a single row with requested values

print.interference *Prints a summary of an interference object*

Description

Prints a summary of an interference object

Usage

```
## S3 method for class 'interference'
print(x, ...)
```

Arguments

x	object of class 'interference'
...	ignored

total_effect *Retrieve Total Effect estimates*

Description

Retrieves the population average total causal effect for specified allocations: $\hat{Y}(0, alpha1) - \hat{Y}(1, alpha2)$

Usage

```
total_effect(object, allocation1, allocation2 = NULL, trt.lvl1 = 0)

te(object, allocation1, allocation2 = NULL, trt.lvl1 = 0)
```

Arguments

object	an object of class interference
allocation1	the allocation scheme for which to estimate total effects
allocation2	the allocation scheme for which to estimate total effects If NULL, then returns all indirect effects compared to allocation1.
trt.lvl1	Defaults to 0.

Value

a data.frame with requested values

vaccinesim *Vaccine Study Sample Data*

Description

A sample dataset based on the simulations of a cholera vaccine trial in Heydrich-Perez et al. (2014) except with 3000 individuals in 250 groups rather than 10000 in 500.

Format

a dataset with 6 variables and 3000 rows

y the outcome (0 - no cholera; 1 - cholera)

X1 an individual's age (in decades)

X2 an individual's distance from river

A an indicator of vaccination (0 - no vaccine; 1 - vaccine)

B an indicator of participation (0 - did not participant in vaccine trial, 1 - did participate)

group group membership

voters *Voting Contagion Experiment Data*

Description

A dataset of a voting contagion experiment. For complete documentation, see the 'Minneapolis and Denver Contagion Experiment' files on David Nickerson's website. See also Nickerson 2008 for more details.

Format

a dataset with 21 variables and 7722 rows

family household ID

denver 1 = subject in Denver, 0 = Minneapolis

treatment 1 = voting encouragement, 2 = recycling message, 3 = not contacted

reached 1 = subject answered door, 0 = not

hsecontact 1 = household contacted by canvassers, 0 = not

voted02p 1 = voted in '02 primary, 0 = not

party party affiliation

age age

gender gender

Details

Just the variables used in the package vignette are documented here.

Source

http://www3.nd.edu/~dnickers/data.php

Index