Package ‘bacr’

October 12, 2022

Type  Package
Title  Bayesian Adjustment for Confounding
Version  1.0.1
Depends  R(>= 2.13.0), graphics, stats, MCMCpack
Date  2016-10-23
Author  Chi Wang <chi.wang@uky.edu>
Maintainer  Chi Wang <chi.wang@uky.edu>
Description  Estimating the average causal effect based on the Bayesian Adjustment for Confounding (BAC) algorithm.
License  GPL-2
NeedsCompilation  no
Repository  CRAN
Date/Publication  2016-10-23 22:58:52

R topics documented:

bacr-package ......................................................... 1
bac ................................................................. 3
plot.bacr .......................................................... 5
summary.bacr ..................................................... 5

Description

This package implements the Bayesian Adjustment for Confounding (BAC) algorithm for estimating the Average Causal Effect (ACE) in Generalized Linear Models. It accounts for uncertainty in confounder and effect modifier selection and allows estimation of the ACE for the whole population or for a specific subpopulation.
Details

Package: bacr
Type: Package
Version: 1.0.1
Depends: R(>= 2.13.0), graphics, stats, MCMCpack
Date: 2016-10-23
License: GPL-2

Author(s)
Chi Wang
Maintainer: Chi Wang <chi.wang@uky.edu>

References

Examples

######### Note that the example below is for illustration purpose only. ######
########## In practice, larger number of iterations will be needed.###########

##### simulate data ############

n = 200; m = 4
V = matrix(rnorm(n*m),ncol=m)
X = rbinom(n, size=1, prob=exp(V[,1])/(1+exp(V[,1])))
beta = c(1,1,1,0.5)
temp0 = cbind(rep(0,n), V[,1:3])
temp1 = cbind(rep(1,n), V[,1:3])
Y0 = rbinom(n, size=1, prob=exp(temp0)/(1+exp(temp0)))
Y1 = rbinom(n, size=1, prob=exp(temp1)/(1+exp(temp1)))
Y = Y0
Y[X==1] = Y1[X==1]
Z = as.data.frame(cbind(Y, X, V))

names(Z) = c("Y", "X", paste("V", 1:m, sep=""))

##### run BAC ###############

result = bac(data=Z, exposure="X", outcome="Y", confounders=paste("V", 1:m, sep=""),
interactors=NULL, familyX="binomial", familyY="binomial", omega=Inf,
num_its=5, burnM=1, burnB=1, thin=1)

##### summarize results ######

summary(result)
plot(result)

##### Adding interaction terms ###############

beta = c(1,1,1,1,1)
temp0 = cbind(rep(0,n), V[,1:3], rep(0,n)*V[,3])
temp1 = cbind(rep(1,n), V[,1:3], rep(1,n)*V[,3])
Y0 = rbinom(n, size=1, prob=exp(temp0)/(1+exp(temp0)))
Y1 = rbinom(n, size=1, prob=exp(temp1)/(1+exp(temp1)))
Y = Y0
Y[X==1] = Y1[X==1]
Z = as.data.frame(cbind(Y, X, V))
names(Z) = c("Y", "X", paste("V", 1:m, sep=""))
result = bac(data=Z, exposure="X", outcome="Y", confounders=paste("V", 1:m, sep=""),
            interactors=paste("V", 1:m, sep=""), familyX="binomial", familyY="binomial",
            omega=Inf, num_its=5, burnM=1, burnB=1, thin=1)
summary(result)
plot(result)

##### Estimate the ACE in the exposed subgroup ############
result = bac(data=Z, exposure="X", outcome="Y", confounders=paste("V", 1:m, sep=""),
            interactors=paste("V", 1:m, sep=""), familyX="binomial", familyY="binomial",
            omega=Inf, num_its=5, burnM=1, burnB=1, thin=1, population=(X==1))
summary(result)
plot(result)

bac  

Bayesian Adjustment for Confounding (BAC)

Description
Estimating the Average Causal Effect (ACE) based on the BAC algorithm

Usage
bac(data, exposure, outcome, confounders, interactors,
    familyX, familyY, omega = Inf, num_its, burnM, burnB,
    thin, population = NULL)

Arguments

- data: a data from containing the input data.
- exposure: the exposure variable
- outcome: the outcome variable
- confounders: a vector of potential confounder variable names
- interactors: a vector of the names of potential confounders that may interact with the exposure
- familyX: the family of the exposure model. Currently, it allows guassian, binomial, and poisson.
- familyY: the family of the outcome model. Currently, it allows guassian, binomial, and poisson.
- omega: a dependent parameter, which is the prior odds of including a predictor in the outcome model, given that the same predictor is in the exposure model. The default value if Inf, which forces predictors in the exposure model to be included in the outcome model.
num_its number of MCMC iterations excluding the burn-in iterations.
burnM number of burn-in iterations when sampling the exposure and outcome models.
burnB number of burn-in iterations when sampling model coefficients based on a given outcome model.
thin the thinning parameter when sampling model coefficients based on a given outcome model.
population the population for which the ACE is based on. It can be either unspecified or a vector of TRUE and FALSE. If unspecified, the function will estimate the ACE for the whole population. If specified, the function will estimate the ACE for the subpopulation defined by the individuals indicated by TRUE.

Details

The function may run slowly for data with large sample size or many potential confounders. The users are suggested to choose small number of iterations first, evaluate the computational speed, then increase the number of iterations. Note that this function assumes a non-informative prior for outcome model coefficients and does not handle informative priors.

Value

a list variable, which contains

data a data frame containing the input data.
MX a matrix of MCMC samples of exposure models.
MY a matrix of MCMC samples of outcome models.
models a list variable containing unique outcome models in the MCMC sample and the appearance frequencies.
exposure the exposure variable.
outcome the outcome variable.
confounders a vector of potential confounder variable names.
interactors a vector of the names of potential confounders that may interact with the exposure.
predictorsY All the possible predictors of the outcome.
ACE a vector of MCMC samples of the ACE.
para a list of model parameters

Author(s)

Chi Wang

References

plot.bacr

See Also

plot.bacr, summary.bacr

---

**plot.bacr**

*Plot Method for Class ‘bacr’*

**Description**

Plot the posterior inclusion probabilities of potential confounders for a bacr object.

**Usage**

```r
## S3 method for class 'bacr'
plot(x, ...)  
```

**Arguments**

- `x`: An object returned by the bac function.
- `...`: Arguments passed to or from other methods.

**Author(s)**

Chi Wang

**See Also**

bac, summary.bacr

---

**summary.bacr**

*Summary Method for Class ‘bacr’*

**Description**

Summarize results from running the bac function.

**Usage**

```r
## S3 method for class 'bacr'
summary(object, ...)  
```

**Arguments**

- `object`: A bacr object.
- `...`: Arguments passed to or from other methods.
Value

a list variable, which contains

posterior.mean  Posterior mean of the ACE
CI             95% posterior interval of the ACE
PIP           A vector of posterior inclusion probabilities for potential confounders

Author(s)

Chi Wang

See Also

bac, plot.bacr
Index

*BAC*
  bacr-package, 1

bac, 3, 5, 6
bacr (bacr-package), 1
bacr-package, 1

plot.bacr, 5, 5, 6

summary.bacr, 5, 5