Package ‘BCEE’

August 10, 2017

Type Package

Title The Bayesian Causal Effect Estimation Algorithm

Version 1.2

Date 2017-08-10

Author Denis Talbot, Geneviève Lefebvre, Juli Atherton, Yohann Chiu.

Maintainer Denis Talbot <denis.talbot@fmed.ulaval.ca>

Description Implementation of the Bayesian Causal Effect Estimation algorithm, a data-driven method for the estimation of the causal effect of an exposure on a continuous outcome. For more details, see Talbot et al. (2015) DOI:10.1515/jci-2014-0035.

License GPL (>= 2)

Imports Rcpp (>= 0.12.12)

LinkingTo Rcpp, RcppArmadillo

Depends BMA, leaps

Encoding latin1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-08-10 20:08:07 UTC

R topics documented:

BCEE-package *The Bayesian Causal Effect Estimation (BCEE) Algorithm*

Description

Contains two functions that correspond to the approximate BCEE (A-BCEE) and the naive BCEE (N-BCEE) implementations of the BCEE algorithm described in Talbot et al. (2015).

Details

Package:	BCEE
Type:	Package
Version:	1.2
Date:	2017-08-10
License:	GPL (>=2)

ABCEE(X, Y, U, omega),
NBCEE(X, Y, U, omega).

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton, Yohann Chiu.

Maintainer: Denis Talbot <denis.talbot@fmed.ulaval.ca>

References

Talbot, D., Lefebvre, G., Atherton, J. (2015) *The Bayesian causal effect estimation algorithm*, Journal of Causal Inference, 3(2), 207-236.

See Also

ABCEE, NBCEE.

ABCEE *Approximate BCEE Implementation*

Description

A-BCEE implementation of the BCEE algorithm.

Usage

```
ABCEE(X, Y, U, omega, forX = NA, niter = 5000, nburn = 500, nthin = 10,
 maxmodelY = NA, OR = 20, family.X = "gaussian")
```

Arguments

- `X` A vector of observed values for the exposure.
- `Y` A vector of observed values for the continuous outcome.
- `U` A matrix of observed values for the `M` potential confounding covariates, where each column contains observed values for a potential confounding factor. A recommended implementation is to only consider pre-exposure covariates.
- `omega` The value of the hyperparameter omega in the BCEE's outcome model prior distribution. A recommended implementation is to take `omega = sqrt(n)*c`, where `n` is the sample size and `c` is a user-supplied constant value. Simulation studies suggest that values of `c` between 100 and 1000 yield good results.
- `forX` A Boolean vector of size `M`, where the `mth` element indicates whether or not the `mth` potential confounding covariate should be considered in the exposure modeling step of the BCEE algorithm. The default for `forX` is `NA`, which indicates that all potential confounding covariates should be considered in the exposure modeling step.
- `niter` The number of post burn-in iterations in the Markov chain Monte Carlo model composition (MC^3) algorithm (Madigan et al. 1995), prior to applying thinning. The default is 5000, but users should ensure that the value is large enough so that the number of retained samples can provide good inferences.
- `nburn` The number of burn-in iterations (prior to applying thinning). The default is 500, but users should ensure that the value is large enough so that convergence of the chain is reached. An example of diagnostics of convergence of the chain is provided below.
- `nthin` The thinning of the chain. The default is 10, but users should ensure that the value is large enough so that there is no auto-correlation between sampled values. An example of diagnostics of absence of auto-correlation is provided below.
- `maxmodelY` The maximum number of distinct outcome models that the algorithm can explore. Choosing a smaller value can shorten computing time. However, choosing a value that is too small will cause the algorithm to crash and return an error message. The default is `NA`; the maximum number of outcome models that can be explored is then set to the minimum of `niter + nburn` and `2^M`.
- `OR` A number specifying the maximum ratio for excluding models in Occam's window for the exposure modeling step (see the `bic.glm` help file, and Madigan & Raftery, 1994). The default is 20.
- `family.X` A description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See `family` for details of family functions.) The default is `"gaussian"`

Details

The `ABCEE` function first computes the exposure model's posterior distribution using the `bic.glm` function if the number of covariates is smaller than 50. Otherwise, the exact procedures depends on the value of `family.X`. The outcome model's posterior distribution is then computed using MC^3 (Madigan et al., 1995) as described in Talbot et al. (2015).

ABCEE assumes there are no missing values in the objects X, Y and U. The na.omit function which removes cases with missing data or an imputation package might be helpful.

Value

betas	A vector containing the sampled values for the exposure effect.
models.X	A matrix giving the posterior distribution of the exposure model. Each row corresponds to an exposure model. Within each row, the first M elements are Booleans indicating the inclusion (1) or the exclusion (0) of each potential confounding factor. The last element gives the posterior probability of the exposure model.
models.Y	A Boolean matrix identifying the sampled outcome models. Each row corresponds to a sampled outcome model. Within each row, the mth element equals 1 if and only if the mth potential confounding covariate is included in the sampled outcome model (and 0 otherwise).

Author(s)

Denis Talbot, Yohann Chiu, Genevieve Lefebvre, Juli Atherton.

References

Madigan, D., York, J., Allard, D. (1995) *Bayesian graphical models for discrete data*, International Statistical Review, 63, 215-232.

Madigan, D., Raftery, A. E. (1994) *Model selection and accounting for model uncertainty in graphical models using Occam's window*, Journal of the American Statistical Association, 89 (428), 1535-1546.

Talbot, D., Lefebvre, G., Atherton, J. (2015) *The Bayesian causal effect estimation algorithm*, Journal of Causal Inference, 3(2), 207-236.

See Also

bic.glm, na.omit, NBCEE.

Examples

```
#Example:
#In this example, both U1 and U2 are potential confounding covariates.
#Both are generated as independent N(0,1).
#X is generated as a function of both U1 and U2 with a N(0,1) error.
#Y is generated as a function of X and U1 with a N(0,1) error.
#Thus, only U1 is a confounder.
#The causal effect of X on Y equals 1.
#The parameter beta associated to exposure in the outcome model
#that includes U1 and the one from the full outcome model is an
#unbiased estimator of the effect of X on Y.

#Generating the data
```

```
set.seed(418949);
U1 = rnorm(200);
U2 = rnorm(200);
X = 0.5*U1 + 1*U2 + rnorm(200);
Y = 1*X + 0.5*U1 + rnorm(200);

#Using ABCEE to estimate the causal exposure effect
results = ABCEE(X,Y,cbind(U1,U2), omega = 500*sqrt(200), niter = 10000, nthin = 5, nburn = 500);

##Diagnostics of convergence of the chain:
plot(results$betas, type = "l");
lines(smooth.spline(1:length(results$beta), results$beta), col = "blue", lwd = 2);
#The plot shows no apparent trend.
#The smoothing curve confirms that there is little or no trend,
#suggesting the chain has indeed converged before burn-in iterations ended.
#Otherwise, the value of nburn should be increased.

##Diagnostics of absence of auto-correlation
acf(results$betas, main = "ACF plot");
#Most lines are within the confidence intervals' limits, which suggests
#that there is no residual auto-correlation. If there were, the value
#of nthin should be increased.

##The number of sampled values is niter/nthin = 2000, which should be
##large enough to provide good inferences for 95% confidence intervals.

#The posterior mean of the exposure effect:
mean(results$betas);
#The posterior standard deviation of the exposure effect:
sd(results$betas);
#The posterior inclusion probability of each covariate:
colMeans(results$models.Y);
#The posterior distribution of the outcome model:
table(apply(results$models.Y, 1, paste0, collapse = ""));
```

NBCEE *Naive BCEE Implementation*

Description

N-BCEE implementation of the BCEE algorithm.

Usage

```
NBCEE(X, Y, U, omega, niter = 5000, nburn = 500, nthin = 10,
 maxmodelX = NA, maxmodelY = NA, family.X = "gaussian")
```

Arguments

X
: A vector of observed values for the exposure.

Y
: A vector of observed values for the continuous outcome.

U
: A matrix of observed values for the M potential confounding covariates, where each column contains observed values for a potential confounding factor. A recommended implementation is to only consider pre-exposure covariates.

omega
: The value of the hyperparameter omega in the BCEE's outcome model prior distribution. A recommended implementation is to take `omega = sqrt(n)*c`, where `n` is the sample size and `c` is a user-supplied constant value. Simulation studies suggest that values of `c` between 100 and 1000 yield good results.

niter
: The number of post burn-in iterations in the Markov chain Monte Carlo model composition (MC^3) algorithm (Madigan et al. 1995), prior to applying thinning. The default is 5000.

nburn
: The number of burn-in iterations (prior to applying thinning). The default is 500.

nthin
: The thinning of the chain. The default is 10.

maxmodelX
: The maximum number of exposure models the algorithm can explore. See `maxmodelY` and the note below.

maxmodelY
: The maximum number of distinct outcome models that the algorithm can explore. Choosing a smaller value can shorten computing time. However, choosing a value that is too small will cause the algorithm to crash. The default is `NA`; the maximum number of outcome models that can be explored is then set to the minimum of `niter + nburn` and `2^M`.

family.X
: A description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See `family` for details of family functions.) The default is `"gaussian"`

Details

`NBCEE` assumes there are no missing values in the objects `X`, `Y` and `U`. The `na.omit` function which removes cases with missing data or an imputation package might be helpful.

Value

betas
: A vector containing the sampled values for the exposure effect.

models.X
: A Boolean matrix identifying the sampled exposure models. See `models.Y`.

models.Y
: A Boolean matrix identifying the sampled outcome models. Each row corresponds to a sampled outcome model. Within each row, the `m`th element equals 1 if and only if the `m`th potential confounding covariate is included in the sampled outcome model (and 0 otherwise).

Note

Variability of the exposure effect estimator is generally underestimated by the N-BCEE implementation of BCEE. The A-BCEE, which also happens to be faster, is thus preferred. Another option is to use N-BCEE with nonparametric bootstrap (B-BCEE) to correctly estimate variability.

The difference in computing time between A-BCEE and N-BCEE is mostly explainable by the method used to compute the posterior distribution of the exposure model. In A-BCEE, this posterior distribution is calculated as a first step using `bic.glm`. In N-BCEE, the posterior distribution of the exposure model is computed inside the MC^3 algorithm.

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton.

References

Madigan, D., York, J., Allard, D. (1995) *Bayesian graphical models for discrete data*, International Statistical Review, 63, 215-232.

Talbot, D., Lefebvre, G., Atherton, J. (2015) *The Bayesian causal effect estimation algorithm*, Journal of Causal Inference, 3(2), 207-236.

See Also

`na.omit`, `ABCEE`.

Examples

```
# In this example, U1 and U2 are potential confounding covariates
# generated as independent N(0,1).
# X is generated as a function of both U1 and U2 with a N(0,1) error.
# Y is generated as a function of X and U1 with a N(0,1) error.
# Variable U1 is the only confounder.
# The causal effect of X on Y equals 1.
# The exposure effect estimator (beta hat) in the outcome model
# including U1 and U2 or including U1 only is unbiased.
# The sample size is n = 200.

# Generating the data
set.seed(418949);
U1 = rnorm(200);
U2 = rnorm(200);
X = 0.5*U1 + 1*U2 + rnorm(200);
Y = 1*X + 0.5*U1 + rnorm(200);

# Using NBCEE to estimate the causal exposure effect
n = 200;
omega.c = 500;
results = NBCEE(X,Y,cbind(U1,U2), omega = omega.c*sqrt(n),
 niter = 1000, nthin = 5, nburn = 20);

# The posterior mean of the exposure effect:
```

```
mean(results$betas);
# The posterior standard deviation of the exposure effect:
sd(results$betas);
# The posterior probability of inclusion of each covariate in the exposure model:
colMeans(results$models.X);
# The posterior distribution of the exposure model:
table(apply(results$models.X, 1, paste0, collapse = ""));
# The posterior probability of inclusion of each covariate in the outcome model:
colMeans(results$models.Y);
# The posterior distribution of the outcome model:
table(apply(results$models.Y, 1, paste0, collapse = ""));
```

Index