

Package ‘BACprior’

August 28, 2025

Type Package

Title Choice of Omega in the BAC Algorithm

Version 2.1.2

Date 2025-08-28

Author Denis Talbot [aut, cre] (main author of original code and maintainer),
Genevieve Lefebvre [aut] (revision of original code),
Juli Atherton [aut] (revision of original code)

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

Description The Bayesian Adjustment for Confounding (BAC) algorithm (Wang et al., 2012) can be used to estimate the causal effect of a continuous exposure on a continuous outcome. This package provides an approximate sensitivity analysis of BAC with regards to the hyperparameter omega. BACprior also provides functions to guide the user in their choice of an appropriate omega value. The method is based on Lefebvre, Atherton and Talbot (2014).

License GPL (>= 2)

Depends mvtnorm, leaps, boot

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2025-08-28 21:30:02 UTC

Contents

BACprior-package	2
BACprior.boot	3
BACprior.CV	5
BACprior.lm	7
Index	11

BACprior-package	<i>Choice of the Hyperparameter Omega in the Bayesian Adjustment for Confounding (BAC) Algorithm</i>
------------------	--

Description

The BACprior package contains functions to help the user select the omega value appearing in the BAC prior distribution of the covariate inclusion indicators of the outcome and exposure models.

Details

Package: BACprior
 Type: Package
 Version: 2.1.2
 Date: 2025-08-28
 License: GPL (>=2)

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton.

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

References

- Brookhart, M.A., van der Lan, M.J. (2006). *A semiparametric model selection criterion with applications to the marginal structural model*, Computational Statistics & Data Analysis, 50, 475-498.
- Hoeting, J.A., Madigan D., Raftery, A.E., Volinsky C.T. (1999). *Bayesian model averaging : A tutorial*, Statistical Science, 16, 382-417.
- Lefebvre, G., Atherton, J., Talbot, D. (2014). *The effect of the prior distribution in the Bayesian Adjustment for Confounding algorithm*, Computational Statistics & Data Analysis, 70, 227-240.
- Wang, C., Parmigiani, G., Dominici, F. (2012). *Bayesian effect estimation accounting for adjustment uncertainty*, Biometrics, 68 (3), 661-671.

BACprior.boot	<i>A Bootstrap Procedure to Guide the Choice of Omega in the Bayesian Adjustment for Confounding Algorithm.</i>
---------------	---

Description

The BACprior.boot function proposes a bootstrap procedure to select BAC's omega value in an attempt to minimize the mean squared error (MSE) of the exposure effect estimator. A number B of bootstrap samples are taken from the original sample. Then, the MSE is estimated for each selected omega value, considering the exposure estimate with omega = infinity from the original sample as the true value. The BACprior.boot function uses the BACprior.lm function to estimate the exposure effect.

Usage

```
BACprior.boot(Y, X, U,
  omega = c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf),
  maxmodels = 150, cutoff = 0.0001, B = 100)
```

Arguments

Y	A vector of observed values for the continuous outcome.
X	A vector of observed values for the continuous exposure.
U	A matrix of observed values for the potential confounders, where each column contains observed values for a potential confounder. A recommended implementation is to only consider pre-exposure covariates.
omega	A vector of omega values for which the bootstrap procedure is performed. The default is c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf).
maxmodels	The maximum number of outcome and exposure models of each size to be considered. Larger numbers improves the approximation, but can greatly increase the computational burden. The default is 150.
cutoff	Minimum posterior probability needed for an outcome model to be considered in the weighted average of the posterior mean and standard deviation of the exposure effect. Smaller values of cutoff improves the approximation, but add computational complexity. The default is 0.0001.
B	The number of bootstrap samples to be taken. Larger numbers reduce Monte Carlo error, but require more computation time.

Details

Since BACprior.boot uses the BACprior.lm function to estimate the exposure effect, users should refer to the BACprior.lm documentation for details of implementation.

BACprior.boot assumes there are no missing values. The objects X, Y and U should be processed beforehand so that every case is complete. The na.omit function which removes cases with missing data or an imputation package might be helpful.

Value

Best The omega value, among the omega values given in input, which minimizes the estimated MSE.

MSE The estimated MSE for each of the selected omega values.

BACprior.boot also returns a plot of the estimated MSEs according to the selected omega values.

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton.

References

Brookhart, M.A., van der Laan, M.J. (2006). *A semiparametric model selection criterion with applications to the marginal structural model*, Computational Statistics & Data Analysis, 50, 475-498.

Lefebvre, G., Atherton, J., Talbot, D. (2014). *The effect of the prior distribution in the Bayesian Adjustment for Confounding algorithm*, Computational Statistics & Data Analysis, 70, 227-240.

See Also

[BACprior.lm](#), [BACprior.CV](#), [na.omit](#)

Examples

```
# Required package to simulate from a multivariate normal distribution.
require(mvtnorm);

# Simulate data
# n = 500 observations with 5 covariates.
# (U1, U2, U4) is multivariate normal with mean vector 0,
# variances of 1 and 0 pairwise correlations.
# U3 and U5 are causal effects of U2 and U4, respectively.
# X is a causal effect of U1, U2 and U4.
# Y is a causal effect of U3, U4, U5 and X.

set.seed(3417817);
n = 500;
U = rmvnorm(n = n, mean = rep(0, 5), sigma = diag(1, nrow = 5) + matrix(0, nrow = 5, ncol = 5));
U[,3] = U[,2] + rnorm(n);
U[,5] = U[,4] + rnorm(n);
X = U[,1] + U[,2] + U[,4] + rnorm(n);
Y = U[,3] + 0.1*U[,4] + U[,5] + 0.1*X + rnorm(n);

# Remove ``#' to run example
# BACprior.boot(Y, X, U, maxmodels = 150);
# $best
# [1] 1
# $MSE
# [1] 0.001467631 0.001480494 0.001505006 0.001539194 0.001580756
```

```
# 0.001803000 0.002017034 0.002375198 0.002516998 0.002662188 0.002865611
# Best omega value would be 1
BACprior.lm(Y, X, U);
# $results
#      omega Posterior mean Standard deviation
# [1,]  1.0      0.1089228      0.02951582
# [2,]  1.1      0.1087689      0.02971457
# [3,]  1.3      0.1084802      0.03008991
# [4,]  1.6      0.1080900      0.03060449
# [5,]  2.0      0.1076376      0.03121568
# [6,]  5.0      0.1057020      0.03426854
# [7,] 10.0      0.1046804      0.03696670
# [8,] 30.0      0.1044711      0.04124805
# [9,] 50.0      0.1047315      0.04291842
# [10,] 100.0     0.1051211      0.04462874
# [11,] Inf       0.1058021      0.04703111

# Posterior mean doesn't change much with omega,
# but posterior standard deviation greatly increases.
# This supports the choice of omega = 1.
```

BACprior.CV

A Cross-Validation Procedure for the Choice of the Omega Value in the Bayesian Adjustment for Confounding Algorithm.

Description

The BACprior.CV function proposes two cross-validation procedures to select BAC's omega value in an attempt to minimize the mean squared error (MSE) of the exposure effect estimator. The data are split into half V times. Each time, the exposure effect is estimated with $\omega = \text{infinity}$ on one half of the data. On the other half of the data, the exposure effect is estimated for each of the selected omega values. A criterion to be minimized is then computed. Details can be found in Lefebvre et al. (2014), which is inspired by a procedure proposed by Brookhart and van der Laan (2006). The BACprior.CV function uses the BACprior.lm function to estimate the exposure effect.

Usage

```
BACprior.CV(Y, X, U,
  omega = c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf),
  maxmodels = 150, cutoff = 0.0001, V = 100, criterion = "CVm")
```

Arguments

Y	A vector of observed values for the continuous outcome.
X	A vector of observed values for the continuous exposure.
U	A matrix of observed values for the potential confounders, where each column contains observed values for a potential confounder. A recommended implementation is to only consider pre-exposure covariates.

omega	A vector of omega values for which the cross-validation procedure is performed. The default is <code>c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf)</code> .
maxmodels	The maximum number of outcome and exposure models of each size to be considered. Larger numbers improves the approximation, but can greatly increase the computational burden. The default is 150.
cutoff	Minimum posterior probability needed for an outcome model to be considered in the weighted average of the posterior mean and standard deviation of the exposure effect. Smaller values of cutoff improves the approximation, but add computational complexity. The default is 0.0001.
V	The number of times the data are split into half. Larger numbers reduce Monte Carlo error, but require more computation time.
criterion	The criterion based on MSE to be computed and minimized. The possible values are “CVm” and “CV” and the default is “CVm”. Both criteria are detailed in Lefebvre et al. (2014).

Details

Since `BACprior.CV` uses the `BACprior.lm` function to estimate the exposure effect, users should refer to the `BACprior.lm` documentation for details of implementation.

`BACprior.CV` assumes there are no missing values. The objects `X`, `Y` and `U` should be processed beforehand so that every case is complete. The `na.omit` function which removes cases with missing data or an imputation package might be helpful.

Value

`Best` The omega value, among the omega values given in input, which minimizes the selected criterion.

`Criterion.Value` The criterion values for the selected omega values.

`BACprior.CV` also returns a plot of the criterion values according to the selected omega values.

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton.

References

Brookhart, M.A., van der Laan, M.J. (2006). *A semiparametric model selection criterion with applications to the marginal structural model*, Computational Statistics & Data Analysis, 50, 475-498.

Lefebvre, G., Atherton, J., Talbot, D. (2014). *The effect of the prior distribution in the Bayesian Adjustment for Confounding algorithm*, Computational Statistics & Data Analysis, 70, 227-240.

See Also

[BACprior.lm](#), [BACprior.boot](#), [na.omit](#)

Examples

```
# Required package to simulate from a multivariate normal distribution.
require(mvtnorm);

# Simulate data
# n = 500 observations with 5 covariates.
# (U1, U2, U4) is multivariate normal with mean vector 0,
# variances of 1 and 0 pairwise correlations.
# U3 and U5 are causal effects of U2 and U4, respectively.
# X is a causal effect of U1, U2 and U4.
# Y is a causal effect of U3, U4, U5 and X.

set.seed(3417817);
n = 500;
U = rmvnorm(n = n, mean = rep(0, 5), sigma = diag(1, nrow = 5) + matrix(0, nrow = 5, ncol = 5));
U[,3] = U[,2] + rnorm(n);
U[,5] = U[,4] + rnorm(n);
X = U[,1] + U[,2] + U[,4] + rnorm(n);
Y = U[,3] + 0.1*U[,4] + U[,5] + 0.1*X + rnorm(n);

# Remove ``#' to run example
# BACprior.CV(Y, X, U, maxmodels = 150, criterion = "CVm");

# $best
# [1] 1
# $Criterion
# [1] 0.0008764926 0.0008817157 0.0008916412 0.0009056528 0.0009233560
# 0.0010425601 0.0012070083 0.0015884799 0.0017678894 0.0019616864 0.0022413220
# Best omega value would be 1

BACprior.lm(Y, X, U);
# $results
#      omega Posterior mean Standard deviation
# [1,] 1.0      0.1089228      0.02951582
# [2,] 1.1      0.1087689      0.02971457
# [3,] 1.3      0.1084802      0.03008991
# [4,] 1.6      0.1080900      0.03060449
# [5,] 2.0      0.1076376      0.03121568
# [6,] 5.0      0.1057020      0.03426854
# [7,] 10.0     0.1046804      0.03696670
# [8,] 30.0     0.1044711      0.04124805
# [9,] 50.0     0.1047315      0.04291842
# [10,] 100.0   0.1051211      0.04462874
# [11,] Inf     0.1058021      0.04703111
# Posterior mean doesn't change much with omega,
# but posterior standard deviation greatly increases.
# This supports the choice of omega = 1.
```

BACprior.lm

Sensitivity of the Bayesian Adjustment for Confounding Algorithm to the Value of the Hyperparameter Omega When the Outcome and the Exposure Are Continuous Variables.

Description

The BACprior.lm function provides estimates of the posterior mean and posterior standard deviation of the exposure effect for selected values of omega in the covariate inclusion indicators' prior distribution. The output allows the user to evaluate how sensitive the BAC procedure is to the choice of the hyperparameter omega value.

Usage

```
BACprior.lm(Y, X, U,
  omega = c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf),
  maxmodels = 150, cutoff = 0.0001, return.best = FALSE)
```

Arguments

Y	A vector of observed values for the continuous outcome.
X	A vector of observed values for the continuous exposure.
U	A matrix of observed values for the potential confounders, where each column contains observed values for a potential confounder. A recommended implementation is to only consider pre-exposure covariates.
omega	A vector of omega values for which the sensitivity analysis is performed. The default is c(1, 1.1, 1.3, 1.6, 2, 5, 10, 30, 50, 100, Inf).
maxmodels	The maximum number of outcome and exposure models of each size to be considered. Larger numbers improves the approximation, but can greatly increase the computational burden. The default is 150.
cutoff	Minimum posterior probability needed for an outcome model to be considered in the weighted average of the posterior mean and standard deviation of the exposure effect. Smaller values of cutoff improves the approximation, but add computational complexity. The default is 0.0001.
return.best	If TRUE, BACprior.lm returns outcome models meeting the cutoff criterion and their posterior probabilities. The default if FALSE.

Details

Only the best maxmodels of each size are recorded for both the outcome and exposure models. The marginal likelihoods of these models are a function of the corresponding BIC values. The posterior mean of the exposure effect parameter is the weighted average of the maximum likelihood estimates, where the weights are the posterior probabilities of the outcome models. The standard deviation of the exposure effect parameter accounts for both the within- and between-model variability (Hoeting et al., 1999), where the within-model variability is given by the standard error of the maximum likelihood exposure effect estimate for the model.

BAC.prior.lm assumes there are no missing values. The objects X, Y and U should be processed beforehand so that every case is complete. The `na.omit` function which removes cases with missing data or an imputation package might be helpful.

Value

<code>results</code>	A matrix in which the first column contains the selected omega values, the second column contains the posterior mean of the exposure effect for each omega value and the third column contains the standard deviation of the exposure effect for each omega value.
<code>best.models</code>	Only returned if <code>return.best = TRUE</code> . A logical matrix whose columns represent the different potential confounders in U and whose lines indicate which covariates are included in a given outcome model. Only outcome models that have a posterior probability larger than the cutoff value for at least one omega value are returned.
<code>posterior.prob</code>	Only returned if <code>return.best = TRUE</code> . A matrix containing the posterior probability of each outcome model in the set of <code>best.models</code> for the different omega values. Each line represents a different outcome model and each column corresponds to one of the selected omega value.

Note

If either the null outcome model (intercept + exposure only) or the null exposure model (intercept only) have a nonnegligible weight, the approximate inferences given by this function might be poor. This occurs since the `regsubset` function used to find the best models of each size does not return null models.

Author(s)

Denis Talbot, Genevieve Lefebvre, Juli Atherton.

References

- Hoeting, J.A., Madigan D., Raftery, A.E., Volinsky C.T. (1999). *Bayesian model averaging : A tutorial*, Statistical Science, 16, 382-417.
- Lefebvre, G., Atherton, J., Talbot, D. (2014). *The effect of the prior distribution in the Bayesian Adjustment for Confounding algorithm*, Computational Statistics & Data Analysis, 70, 227-240.
- Wang, C., Parmigiani, G., Dominici, F. (2012). *Bayesian effect estimation accounting for adjustment uncertainty*, Biometrics, 68 (3), 661-671.

See Also

[regsubsets](#), [na.omit](#)

Examples

```
# Required package to simulate from a multivariate normal distribution.
require(mvtnorm);
```

```

# Simulate data
# n = 500 observations with 10 covariates.
# (U1, U2, U4, U6, U7, U8, U9, U10) is multivariate normal
# with mean vector 0, variances of 1 and pairwise correlations of 0.25.
# U3 and U5 are causal effects of U2 and U4, respectively.
# X is a causal effect of U1, U2 and U4.
# Y is a causal effect of X, U3, U4 and U5.
# The true exposure effect is 0.1.
ncov = 10;
n = 500;
U = rmvnorm(n = n, mean = rep(0, ncov),
  sigma = diag(0.75, nrow = ncov) + matrix(0.25, nrow = ncov, ncol = ncov));
U[,3] = U[,2] + rnorm(n);
U[,5] = U[,4] + rnorm(n);
X = 0.5*U[,1] + 0.5*U[,2] + U[,4] + rnorm(n);
Y = U[,3] + 0.3*U[,4] + U[,5] + 0.1*X + rnorm(n);

BACprior.lm(Y, X, U, omega = c(1, 1.5, 2, 5, 10, 50, 100, Inf));

```

Index

BACprior (BACprior-package), [2](#)

BACprior-package, [2](#)

BACprior.boot, [3](#), [6](#)

BACprior.CV, [4](#), [5](#)

BACprior.lm, [4](#), [6](#), [7](#)

na.omit, [4](#), [6](#), [9](#)

regsubsets, [9](#)