# Package ‘dbacf’

June 29, 2023

**Version** 0.2.8

**Date** 2023-06-27

**Title** Autocovariance Estimation via Difference-Based Methods

**Author** Inder Tecuapetla-Gómez [aut, cre]

**Maintainer** Inder Tecuapetla-Gómez
<itecuapetla@conabio.gob.mx>

**Description** Provides methods for (auto)covariance/correlation function estimation in change point regression with stationary errors circumventing the pre-estimation of the underlying signal of the observations. Generic, first-order, (m+1)-gapped, difference-based autocovariance function estimator is based on M. Levine and I. Tecuapetla-Gómez (2023) <doi:10.48550/arXiv.1905.04578>. Bias-reducing, second-order, (m+1)-gapped, difference-based estimator is based on I. Tecuapetla-Gómez and A. Munk (2017) <doi:10.1111/sjos.12256>. Robust autocovariance estimator for change point regression with autoregressive errors is based on S. Chakar et al. (2017) <doi:10.3150/15-BEJ782>. It also includes a general projection-based method for covariance matrix estimation.

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** Matrix

**Depends** R (>= 2.15.3)

**NeedsCompilation** no

**RoxygenNote** 7.2.0

**Repository** CRAN

**Date/Publication** 2023-06-29 14:30:16 UTC

## R topics documented:

dbacf-package . . . . . . . . . . 2
dbacf . . . . . . . . . . 3
dbacf_AR1 . . . . . . . . . . 5
nearPDToeplitz . . . . . . . . . . 6
plot.dbacf . . . . . . . . . . 8
projectToeplitz . . . . . . . . . . 9
symBandedToeplitz . . . . . . . . . . 10

**Index** **11**

---

`dbacf-package` *Autocovariance function estimation via difference-based methods*

---

### Description

Difference-based (auto)covariance/correlation estimation in change point regression with stationary errors.

Provides bias-reducing methods for (auto)covariance-correlation estimation in change point regression with stationary $m$-dependent errors without having to pre-estimate the underlying signal of the observations. In the same spirit, provides a robust estimator of the autorregressive coefficient in change point regression with stationary, $AR(1)$ errors. It also includes a general projection-based method for covariance matrix estimation.

### Autocovariance Estimation

`dbacf` returns *and plots by default* (auto)covariance/correlation estimates without pre-estimating the underlying *not necessarily smooth* signal of observations with *stationary* $m$-dependent errors. The corresponding plot method `plot.dbacf` allows for adjusting graphical parameters to users' liking. This method is based on `plot.acf`.

`dbacf_AR1` returns (auto)covariance/correlation estimates while circumventing the difficult estimation of the underlying change point regression function from observations with stationary $AR(1)$ errors.

### Covariance Matrix Estimation

Given a matrix estimate, *not necessarily positive definite*, of the covariance matrix of a stationary process, `nearPDToeplitz` returns the nearest, *in the Frobenius norm*, covariance matrix to the initial estimate. See `projectToeplitz` for the projection of a given symmetric matrix onto the space of Toeplitz matrices. See also `symBandedToeplitz` for creating a (stationary process' large covariance) matrix by specifying its dimension and values of its autocovariance function.

### Author(s)

Tecuapetla-Gómez, I. <itecuap@conabio.gob.mx>

### References

Grigoriadis, K.M., Frazho, A., Skelton, R. (1994). *Application of alternating convex projection methods for computation of positive Toeplitz matrices*, IEEE Transactions on signal processing **42(7)**, 1873–1875.

N. Higham (2002). *Computing the nearest correlation matrix - a problem from finance*, Journal of Numerical Analysis **22**, 329–343.

Tecuapetla-Gómez, I and Munk, A. (2017). *Autocovariance estimation in regression with a discontinuous signal and $m$-dependent errors: A difference-based approach*. Scandinavian Journal of Statistics, **44(2)**, 346–368.

Levine, M. and Tecuapetla-Gómez, I. (2023). *Autocovariance function estimation via difference schemes for a semiparametric change point model with $m$-dependent errors*. Submitted.

---

`dbacf` *Difference-based (auto)covariance/correlation function estimation*

---

## Description

Computes *and by default plots* the (auto)covariance/correlation function estimate without pre-estimating the underlying *piecewise constant signal* of the observations. To that end, a class of second-order *difference-based estimators* is implemented according to Eqs.(2.5)-(2.6) of *Tecuapetla-Gómez and Munk (2017)*. By default, this function computes a subclass of estimates with minimal bias according to Eqs.(2.12)-(2.14) of the aforementioned paper.

## Usage

```
dbacf(
  data,
  m,
  d,
  type = c("covariance", "correlation"),
  order = c("second", "first"),
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `data` | numeric vector or a univariate object of class `ts` of length at least `2(m + 1)`. |
| `m` | integer scalar giving the underlying level of dependency. |
| `d` | numeric vector giving the weights used in difference-based estimation method. Only pertinent when `order=second`. If missing, the weights `d` are calculated according to Eqs.(2.12)-(2.14) of *Tecuapetla-Gómez and Munk (2017)*. When a single value $d^*$ is specified, `d = rep(`$d^*$`, m + 1)`. |
| `type` | character string specifying whether covariance (default) or correlation must be computed. |
| `order` | character specifying whether a `first` (default) or a `second` difference-based estimate should be employed. |
| `plot` | logical. If `TRUE` (default) the acf is plotted. |
| `...` | further arguments passed to `plot.dbacf`. |

## Value

An object of class "dbacf" containing:

| | |
|---|---|
| `acf` | numeric vector of length `m + 1` giving estimated (auto)covariance-correlation. |
| `m` | integer giving underlying level of dependency. |
| `d` | numeric vector containing the weights used to estimate acf. |
| `acfType` | string indicating whether `covariance` or `correlation` has been computed. |
| `n` | integer giving `length(data)`. |
| `series` | string with name of variable `data`. |

## Note

Although the theoretical properties of the methods implemented in this function were derived for change point regression with stationary *Gaussian* $m$-dependent errors, these methods have proven robust against non-normality of the errors and as efficient as other methods in which pre-estimation of an underlying smooth signal is required. For further details see Section 6 of *Tecuapetla-Gómez and Munk (2017)*.

The first-order difference-based estimator was implemented following Eqs.(4)-(5) of *Levine and Tecuapetla-Gómez (2023)*. For the robustness of this estimator see Section 4 of the just mentioned paper.

## References

Tecuapetla-Gómez, I and Munk, A. (2017). *Autocovariance estimation in regression with a discontinuous signal and $m$-dependent errors: A difference-based approach*. Scandinavian Journal of Statistics, **44(2)**, 346–368.

Levine, M. and Tecuapetla-Gómez, I. (2023). *Autocovariance function estimation via difference schemes for a semiparametric change point model with $m$-dependent errors*. Submitted.

## See Also

`acf`, `plot.dbacf`

## Examples

```
ma2 <- arima.sim(n = 50, model = list(ma = c(0.4, -0.4), order = c(0, 0, 2)),
                 sd = 0.25)
dbacf(data=ma2, m = 2)
dbacf(data=ma2, m = 2, order="first")
```

---

`dbacf_AR1` *Robust dbacf in change point regression with AR(1) errors*

---

## Description

In the context of change point regression with a stationary AR(1) error process, this function estimates the autoregressive coefficient along with the autocovariance/correlation function as a function of given lags.

## Usage

```
dbacf_AR1(data, type = c("covariance", "correlation"), lags)
```

## Arguments

| | |
|---|---|
| `data` | numeric vector or a univariate object of class `ts`. |
| `type` | character string specifying whether covariance (default) or correlation must be computed. |
| `lags` | numeric giving the number of lags to compute. |

## Value

An object of class "dbacf" containing:

- `acf` numeric vector of length `lags + 1` giving estimated (auto)covariance/correlation function
- `rho` numeric, estimate of autoregressive coefficient
- `acfType` string indicating whether `covariance` or `correlation` has been computed
- `n` integer giving `length(data)`

## References

Chakar, S. and Lebarbier, E. and Lévy-Leduc, C. and Robin, S. (2017). *A robust approach for estimating change-points in the mean of an AR(1) process*, Bernoulli, **23(2)**, 1408-1447

## Examples

```
ar1 <- arima.sim(n = 50, model = list(ar = c(0.5), order = c(1, 0, 0)),
                 sd = 0.25)
dbacf_AR1(ar1, type="correlation", lags=10)
```

nearPDToeplitz *Computes the nearest positive definite Toeplitz matrix*

## Description

Computes the nearest positive definite Toeplitz matrix to an initial approximation, typically a covariance (correlation) matrix of a stationary process. This function implements an *alternating projection algorithm* that combines Grigoriadis et al. (1994) and Higham (2002). For further details see Section 5 of Tecuapetla-Gómez and Munk (2017).

## Usage

```
nearPDToeplitz(
  matrix,
  type = c("covariance", "correlation"),
  toleranceEigen = 1e-06,
  toleranceConvergence = 1e-06,
  tolerancePosDef = 1e-06,
  maxIterations = 100,
  doEigen = TRUE
)
```

## Arguments

`matrix` a symmetric matrix.

`type` string indicating whether the elements of the main diagonal must be all equal to 1 ("`correlation`") or not ("`covariance`").

`toleranceEigen` defines relative positiveness of eigenvalues compared to largest one.

`toleranceConvergence` numeric indicating convergence tolerance for alternating projection algorithm.

`tolerancePosDef` tolerance for forcing positive definiteness (in the final step) of alternating projection algorithm.

`maxIterations` integer giving maximum number of iterations allowed in alternating projection algorithm; when this number is exceeded without convergence a warning is displayed and matrix computed in step `maxIterations` of algorithm is returned.

`doEigen` logical indicating whether finding the closest positive definite matrix -through a eigen step- should be applied to the result of the alternating projection algorithm.

## Details

This function is based on an alternating projection algorithm which involves projecting sequentially and iteratively the initial matrix into the set of symmetric positive definite and into the space of Toeplitz matrices, respectively. The iteration process will stop because either a criterion of convergence is met or `maxIterations` has been exceeded (without convergence). Criterion of convergence: if the Frobenius norm of the difference of the projection matrices computed in the last two iterations of the algorithm is smaller than toleranceConvergence, then the algorithm stops returning the projection matrix computed in the last iteration.

When projecting onto the set of symmetric positive definite matrices, toleranceEigen controls the relative magnitude of any eigenvalue $\lambda_k$ with respect to the largest one $\lambda_1$ and all eigenvalues $\lambda_k$ are treated as zero if $\lambda_k/\lambda_1 \leq$ toleranceEigen.

## Value

A list containing:

- `projection` a matrix, the computed symmetric positive definite Toeplitz matrix.
- `normF` Frobenius norm of the difference between original and projection matrix.
- `iterations` number of iterations used for alternating projection algorithm.
- `relativeTolerance` numeric giving relative error (in Frobenius norm) of final approximation with respect to original matrix.
- `converged` logical indicating if alternating projection algorithm converged.

## References

Grigoriadis, K.M., Frazho, A., Skelton, R. (1994). *Application of alternating convex projection methods for computation of positive Toeplitz matrices*, IEEE Transactions on signal processing **42(7)**, 1873–1875.

N. Higham (2002). *Computing the nearest correlation matrix - a problem from finance*, Journal of Numerical Analysis **22**, 329–343.

Tecuapetla-Gómez, I and Munk, A. (2017). *Autocovariance estimation in regression with a discontinuous signal and $m$-dependent errors: A difference-based approach*. Scandinavian Journal of Statistics, **44(2)**, 346–368.

## See Also

nearPD, projectToeplitz, symBandedToeplitz, posdefify

## Examples

```
# Higham (2002), p. 334
(mat <- matrix(c(1, 1, 0, 1, 1, 1, 0, 1, 1), byrow = TRUE, ncol = 3))
matProj <- matrix(c(1, 0.7607, 0.1573, 0.7607, 1, 0.7607, 0.1573, 0.7607, 1),
                  byrow = TRUE, ncol = 3)
nrPDT.mat <- nearPDToeplitz(mat, type = "correlation")
stopifnot( identical(unname(matProj), unname(round(as.matrix(nrPDT.mat$projection),
                     digits=4) ) ))
eigen(nrPDT.mat$projection)$values

# Toeplitz banded matrix near to the covariance matrix of 100 realizations
# of an MA(5) with following parameters:

n <- 1e2
alphas <- c(-2, 0.5, -4, 0, 0.75)
```

```
(true.acf <- ARMAacf(ma = alphas))
alphasMat <- symBandedToeplitz(true.acf, n = n)
stopifnot( min(eigen(alphasMat)$values) > 0 ) # alphasMat is a positive definite matrix

(l <- length(true.acf))
(acf.modified <- c(true.acf[-c(l - 1, l)], 0.25)) # modifying original acf
x <- acf.modified
acfMat <- symBandedToeplitz(x, n = n)

# no. of non positive eigenvalues of acfMat (6)
length( eigen(acfMat)$values[eigen(acfMat)$values < 0 ] )
# acfMat is a 100 x 100 symmetric banded Toeplitz matrix
acfMat[1:15, 1:30]

system.time(nrPDT.acfMat <- nearPDToeplitz(acfMat, type = "correlation"))
y <- eigen(nrPDT.acfMat$projection)$values
# no. of non positive eigenvalues of nrPDT.acfMat
length( y[ y < 0 ] ) # none!
```

---

`plot.dbacf` *Plot autocovariance and autocorrelation functions*

---

### Description

This function returns the plot method for objects of class "dbacf".

### Usage

```
## S3 method for class 'dbacf'
plot(
  x,
  type = "h",
  xlab = "Lag",
  ylab = paste("ACF", ifelse(x$acfType == "covariance", "(cov)", " ")),
  xlim = c(0, x$m + 1),
  main = paste("Series", x$series),
  ltyZeroLine = 3,
  colZeroLine = "blue",
  ...
)
```

### Arguments

| | |
|---|---|
| `x` | an object of class "dbacf". |
| `type` | what type of plot should be drawn. For possible types see `plot`. |
| `xlab` | the x label of the plot. |
| `ylab` | the y label of the plot. |

| | |
|---|---|
| `xlim` | numeric vector of length 2 giving the x coordinates range. |
| `main` | an overall title for the plot. |
| `ltyZeroLine` | type of line used to draw horizontal line passing at 0. |
| `colZeroLine` | string indicating color of horizontal line passing at 0. |
| `...` | extra arguments to be passed to plot. |

## Value

No return value

## Note

`dbacf` documents the structure of objects of class "dbacf".

## See Also

`acf`, `dbacf`.

---

`projectToeplitz` *Projection onto the set of symmetric Toeplitz matrices*

---

## Description

Computes the orthogonal projection onto the space of symmetric Toeplitz matrices as given in *Grigoriadis et al. (1994).*

## Usage

```
projectToeplitz(matrix)
```

## Arguments

| | |
|---|---|
| `matrix` | a symmetric matrix. |

## Value

The computed projection matrix.

## References

Grigoriadis, K.M., Frazho, A., Skelton, R. (1994). *Application of alternating convex projection methods for computation of positive Toeplitz matrices*, IEEE Transactions on signal processing **42(7)**, 1873–1875

## See Also

`nearPDToeplitz`

## Examples

```
A <- matrix(c(2, 1, 1, 1, 2, 0, 1, 0, 0), byrow = 3, nrow = 3)
projectToeplitz(A)
```

---

`symBandedToeplitz` *Creates a symmetric banded Toeplitz matrix*

---

## Description

Creates a symmetric banded Toeplitz matrix

## Usage

```
symBandedToeplitz(x, n)
```

## Arguments

| | |
|---|---|
| x | numeric vector or an object of class `dbacf`. |
| n | integer specifying number of columns (rows) of banded matrix. |

## Value

An $n \times n$ symmetric banded Toeplitz matrix whose entries in main band are given by object `x`.

## See Also

`nearPDToeplitz`, `bandSparse`

## Examples

```
alphas <- c(-2, 0.5, -4)
(true.acf <- ARMAacf(ma = alphas))
symBandedToeplitz(true.acf, n = 10)
```

# Index

∗ **package**
    dbacf-package, 2

acf, *4*, *9*

bandSparse, *10*

dbacf, *2*, 3, *9, 10*
dbacf-package, 2
dbacf_AR1, *2*, 5

nearPD, *7*
nearPDToeplitz, *2*, 6, *9, 10*

plot, *8*
plot.acf, *2*
plot.dbacf, *2–4*, 8
posdefify, *7*
projectToeplitz, *2*, *7*, 9

symBandedToeplitz, *2*, *7*, 10

ts, *3*, *5*