

Package: mbend (via r-universe)

October 17, 2024

Title Matrix Bending

Version 1.3.1

Description Bending non-positive-definite (symmetric) matrices to positive-definite, using weighted and unweighted methods.
Jorjani, H., et al. (2003)
<[doi:10.3168/jds.S0022-0302\(03\)73646-7](https://doi.org/10.3168/jds.S0022-0302(03)73646-7)>. Schaeffer, L. R. (2014)
<<http://animalbiosciences.uoguelph.ca/~lrs/ELARES/PDforce.pdf>>.

License GPL-3

LazyData true

URL <https://github.com/nilforooshan/mbend>

BugReports <https://github.com/nilforooshan/mbend/issues>

RoxygenNote 7.1.2

Encoding UTF-8

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Mohammad Ali Nilforooshan [aut, cre] (0000-0003-0339-5442)

Maintainer Mohammad Ali Nilforooshan <m.a.nilforooshan@gmail.com>

Repository <https://nilforooshan.r-universe.dev>

RemoteUrl <https://github.com/nilforooshan/mbend>

RemoteRef HEAD

RemoteSha 5499efa3b469f825bf7a8418dd132ddeacd54ef9

Contents

mbend-package	2
bend	3

Index	6
--------------	----------

Description

(Co)variance or correlation matrices are required in multivariate mixed models. For example, in multi-trait animal models, genetic and residual (co)variance matrices are required, involving the traits of interest. These matrices need to be positive definite (PD) and invertible. Variance component estimation is computationally expensive, especially for big data and many variables. As a result, the full (co)variance matrix may be assembled by combining smaller (co)variance matrices from variance component estimation analyses on subsets of variables. Possible missing covariances are filled with values from the literature or the best possible guess. Consequently, the assembled matrix may not be PD, and it needs to be bent to a PD matrix before being used in the model.

Details

A method for weighted bending of (co)variance matrices was developed by Jorjani et al. (2003), in which the matrix of interest is decomposed to matrices of eigenvectors and eigenvalues. Iteratively, eigenvalues smaller than a small positive value (close to 0) are replaced with that small positive value and the matrix is rebuilt, until the convergence is met (i.e., all eigenvalues being positive). Because there are different amount of data and certainty associated with different elements of the matrix, weighting factors should be involved, which are introduced through a symmetric matrix. Certainty associated with the elements of the matrix and the corresponding weights are inversely related. For example, the reciprocal of the number of common data points (i.e., data points in common between pairs of variables) can be used as weighting factors. Alternatively, number of data points can be used directly by setting the argument `reciprocal = TRUE`. To keep specific elements of the matrix unchanged during the bending process, set corresponding weights to 0. Providing no weight matrix is equivalent to unweighted bending. Another method implemented in this package is from Schaeffer (2014), which can be defined by using the argument `method = "lrs"`. In this method, negative eigenvalues are replaced with positive values in a descending order. If no method is defined, the default `method = "hj"` (Jorjani et al., 2003) is used. As a development to the method of Schaeffer (2014), a weight matrix can be used for weighted bending. Any (co)variance matrix with all diagonal elements equal to 1 is considered as a correlation matrix by the program.

Author(s)

Mohammad Ali Nilforooshan <m.a.nilforooshan@gmail.com>

References

- Jorjani, H., et al. (2003). A Simple Method for Weighted Bending of Genetic (Co)variance Matrices. *J. Dairy Sci.*, 86:677-679. <doi:10.3168/jds.S0022-0302(03)73646-7>
- Schaeffer, L. R. (2014). Making covariance matrices positive definite. Available at: [Link](#)

bend	<i>Matrix bending to positive-definite</i>
------	--

Description

Bending a symmetric non-positive-definite matrix to positive-definite, using weighted or unweighted methods.

Usage

```
bend(  
  inmat,  
  wtmat,  
  reciprocal = FALSE,  
  max.iter = 10000,  
  small.positive = 1e-04,  
  method = "hj"  
)
```

Arguments

`inmat` : The matrix to be bent.

`wtmat` : The weight matrix for weighted bending. If no input is provided, the unweighted method (default) is used.

`reciprocal` : If TRUE, reciprocal of the weighting factors are used. If no input is provided, default = FALSE.

`max.iter` : Maximum number of iterations. If no input is provided, default = 10000.

`small.positive` : Eigenvalues smaller than this value are replaced with this value. If no input is provided, default = 0.0001.

`method` : "hj" (Jorjani et al., 2003) or "lrs" (Schaeffer, 2014), default = "hj"

Value

`bent` : The bent matrix.

`init.ev` : Eigenvalues of the initial (`inmat`) matrix.

`final.ev` : Eigenvalues of the bent matrix.

`min.dev` : $\min(\text{bent} - \text{inmat})$.

`max.dev` : $\max(\text{bent} - \text{inmat})$.

`loc.min.dev` : Location (indices) of `min.dev` element.

`loc.max.dev` : Location (indices) of `max.dev` element.

`ave.dev` : Average deviation ($\text{bent} - \text{inmat}$) of the upper triangle elements (excluding diagonal elements for correlation matrices).

AAD : Average absolute deviation of the upper triangle elements (excluding diagonal elements for correlation matrices) of bent and inmat.

Cor : Correlation between the upper triangle elements (excluding diagonal elements for correlation matrices) of bent and inmat.

RMSD : Root of mean squared deviation of the upper triangle elements (excluding diagonal elements for correlation matrices) of bent and inmat.

w_gt_0 : Number of weight elements greater than 0, in the upper triangle of wtmat (for weighted bending).

wAAD : Weighted AAD (for weighted bending).

wCor : Weighted Cor (for weighted bending).

wRMSD : Weighted RMSD (for weighted bending).

Examples

```
# Test data
V = matrix(nrow=5, ncol=5, c( # matrix to be bent
100, 95, 80, 40, 40,
95, 100, 95, 80, 40,
80, 95, 100, 95, 80,
40, 80, 95, 100, 95,
40, 40, 80, 95, 100))
W = matrix(nrow=5, ncol=5, c( # matrix of weights
1000, 500, 20, 50, 200,
500, 1000, 500, 5, 50,
20, 500, 1000, 20, 20,
50, 5, 20, 1000, 200,
200, 50, 20, 200, 1000))

# Example 1: Unweighted bending
bend(V)
## The default method (Jojani et al. 2003) is used.

# Example 2: Weighted bending using reciprocal of the weighting factors
bend(inmat=V, wtmat=W, reciprocal=TRUE)

# Example 3: Bending with fixed elements
## Assume we want to keep V[1:2, 1:2] constant.
W2 = W; W2[1:2, 1:2] = 0
bend(inmat=V, wtmat=W2, reciprocal=TRUE)

# Example 4: Bending a correlation matrix
V2 = cov2cor(V)
bend(V2, W, reciprocal=TRUE)

# Example 5: Bending using the method of Schaeffer (2014)
bend(inmat=V, method="lrs")

# Example 6: Bending a correlation matrix using the method of Schaeffer (2014)
bend(V2, method="lrs")
```

```
# Example 7: Bending the same correlation matrix using a weighted development of Schaeffer (2014)
bend(V2, W, reciprocal=TRUE, method="lrs")
```

```
# Example 8: Bending a covariance matrix using a weighted development of Schaeffer (2014)
bend(V, W, reciprocal=TRUE, method="lrs")
```

Index

[bend](#), [3](#)

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