

# Package: cml (via r-universe)

September 10, 2024

**Type** Package

**Title** Conditional Manifold Learning

**Version** 0.2.2

**Author** Anh Tuan Bui [aut, cre]

**Maintainer** Anh Tuan Bui <atbui@u.northwestern.edu>

**Imports** vegan

**Description** Finds a low-dimensional embedding of high-dimensional data, conditioning on available manifold information. The current version supports conditional MDS (based on either conditional SMACOF in Bui (2021) <[arXiv:2111.13646](https://arxiv.org/abs/2111.13646)> or closed-form solution in Bui (2022) <[doi:10.1016/j.patrec.2022.11.007](https://doi.org/10.1016/j.patrec.2022.11.007)>) and conditional ISOMAP in Bui (2021) <[arXiv:2111.13646](https://arxiv.org/abs/2111.13646)>.

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-04-24 07:40:05 UTC

## Contents

cml-package . . . . .	2
ccor . . . . .	3
condDist . . . . .	4
condIsomap . . . . .	5
condMDS . . . . .	6
condMDSeigen . . . . .	8
condSmacof . . . . .	9
cz . . . . .	10
mpinv . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

## Description

Finds a low-dimensional embedding of high-dimensional data, conditioning on available manifold information. The current version supports conditional MDS (based on either conditional SMACOF or closed-form solution) and conditional ISOMAP.

Please cite this package as follows:

Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>

Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. Pattern Recognition Letters 164, 148-152. <https://doi.org/10.1016/j.patrec.2022.11.007>

## Details

Brief descriptions of the main functions of the package are provided below:

`condMDS()`: is the conditional MDS method, which uses conditional SMACOF to optimize its conditional stress objective function.

`condMDSseigen()`: is the conditional MDS method, which uses a closed-form solution based on multiple linear regression and eigendecomposition.

`condIsomap()`: is the conditional ISOMAP method, which is basically conditional MDS applying to graph distances (i.e., estimated geodesic distances) of the given distances/dissimilarities.

## Author(s)

Anh Tuan Bui

Maintainer: Anh Tuan Bui <[atbui@u.northwestern.edu](mailto:atbui@u.northwestern.edu)>

## References

Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.

Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. Pattern Recognition Letters 164, 148-152. <https://doi.org/10.1016/j.patrec.2022.11.007>

## Examples

```
## Generate car-brand perception data
factor.weights <- c(90, 88, 83, 82, 81, 70, 68)/562
N <- 100
set.seed(1)
data <- matrix(runif(N*7), N, 7)
colnames(data) <- c('Quality', 'Safety', 'Value', 'Performance', 'Eco', 'Design', 'Tech')
rownames(data) <- paste('Brand', 1:N)
```

```

data.hat <- data + matrix(rnorm(N*7), N, 7)*data*.05
data.weighted <- t(apply(data, 1, function(x) x*factor.weights))
d <- dist(data.weighted)
d.hat <- d + rnorm(length(d))*d*.05

## The following examples use the first 4 factors as known features
# Conditional MDS based on conditional SMACOF
u.cmds = condMDS(d.hat, data.hat[,1:4], 3, init='none')
u.cmds$B # compare with diag(factor.weights[1:4])
ccor(data.hat[,5:7], u.cmds$U)$cancor # canonical correlations
vegan::procrustes(data.hat[,5:7], u.cmds$U, symmetric = TRUE)$ss # Procrustes statistic

# Conditional MDS based on the closed-form solution
u.cmds = condMDSeigen(d.hat, data.hat[,1:4], 3)
u.cmds$B # compare with diag(factor.weights[1:4])
ccor(data.hat[,5:7], u.cmds$U)$cancor # canonical correlations
vegan::procrustes(data.hat[,5:7], u.cmds$U, symmetric = TRUE)$ss # Procrustes statistic

# Conditional MDS based on conditional SMACOF,
# initialized by the closed-form solution
u.cmds = condMDS(d.hat, data.hat[,1:4], 3, init='eigen')
u.cmds$B # compare with diag(factor.weights[1:4])
ccor(data.hat[,5:7], u.cmds$U)$cancor # canonical correlations
vegan::procrustes(data.hat[,5:7], u.cmds$U, symmetric = TRUE)$ss # Procrustes statistic

# Conditional ISOMAP
u.cisomap = condIsomap(d.hat, data.hat[,1:4], 3, k = 20, init='eigen')
u.cisomap$B # compare with diag(factor.weights[1:4])
ccor(data.hat[,5:7], u.cisomap$U)$cancor
vegan::procrustes(data.hat[,5:7], u.cisomap$U, symmetric = TRUE)$ss

```

---

ccor

*Canonical Correlations*


---

## Description

Computes canonical correlations for two sets of multivariate data  $x$  and  $y$ .

## Usage

```
ccor(x, y)
```

## Arguments

$x$	the first multivariate dataset.
$y$	the second multivariate dataset.

**Value**

a list of the following components:

canor	a vector of canonical correlations.
xcoef	a matrix, each column of which is the vector of coefficients of x to produce the corresponding canonical covariate.
ycoef	a matrix, each column of which is the vector of coefficients of y to produce the corresponding canonical covariate.

**Author(s)**

Anh Tuan Bui

**Examples**

```
ccor(iris[,1:2], iris[,3:4])
```

---

condDist	<i>Conditional Euclidean distance</i>
----------	---------------------------------------

---

**Description**

Internal functions.

**Usage**

```
condDist(U, V.tilda, one_n_t=t(rep(1,nrow(U))))
condDist2(U, V.tilda2, one_n_t=t(rep(1,nrow(U))))
```

**Arguments**

U	the embedding U
V.tilda	= V %*% B
V.tilda2	= V %*% b^2*t(V)
one_n_t	= t(rep(1,nrow(U)))

**Value**

a dist object.

**Author(s)**

Anh Tuan Bui

**References**

Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.

---

condIsomap

*Conditional ISOMAP*


---

## Description

Finds a low-dimensional manifold embedding of a given distance/dissimilarity matrix, conditioning on available manifold information. The method applies conditional MDS (see [condMDS](#)) to a graph distance matrix computed for the given distances/dissimilarities, using the `isomap{vegan}` function.

## Usage

```
condIsomap(d, V, u.dim, epsilon = NULL, k, W,
           method = c('matrix', 'vector'), exact = TRUE,
           it.max = 1000, gamma = 1e-05,
           init = c('none', 'eigen', 'user'),
           U.start, B.start, ...)
```

## Arguments

<code>d</code>	a distance/dissimilarity matrix of $N$ entities (or a <code>dist</code> object).
<code>V</code>	an $N \times q$ matrix of $q$ manifold auxiliary parameter values of the $N$ entities.
<code>u.dim</code>	the embedding dimension.
<code>epsilon</code>	shortest dissimilarity retained.
<code>k</code>	Number of shortest dissimilarities retained for a point. If both <code>epsilon</code> and <code>k</code> are given, <code>epsilon</code> will be used.
<code>W</code>	an $N \times N$ symmetric weight matrix. If not given, a matrix of ones will be used.
<code>method</code>	if <code>matrix</code> , there are no restrictions for the <code>B</code> matrix. If <code>vector</code> , the <code>B</code> matrix is restricted to be diagonal. The latter is more efficient for large $q$ .
<code>exact</code>	only relevant if <code>W</code> is not given. In this case, if <code>exact == FALSE</code> , <code>U</code> is updated by the large- $N$ approximation formula.
<code>it.max</code>	the max number of conditional SMACOF iterations.
<code>gamma</code>	conditional SMACOF stops early if the reduction of normalized conditional stress is less than <code>gamma</code>
<code>init</code>	initialization method.
<code>U.start</code>	user-defined starting values for the embedding (when <code>init = 'user'</code> )
<code>B.start</code>	starting <code>B</code> matrix.
<code>...</code>	other arguments for the <code>isomap{vegan}</code> function.

**Value**

U	the embedding result.
B	the estimated B matrix.
stress	Normalized conditional stress value.
sigma	the conditional stress value at each iteration.
init	the value of the init argument.
U.start	the starting values for the embedding.
B.start	starting values for the B matrix.
method	the value of the method argument.
exact	the value of the exact argument.

**Author(s)**

Anh Tuan Bui

**References**

- Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.
- Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. Pattern Recognition Letters 164, 148-152. <https://doi.org/10.1016/j.patrec.2022.11.007>

**See Also**

[condMDS](#), [condMDSeigen](#)

**Examples**

```
# see help(cml)
```

---

condMDS

*Conditional Multidimensional Scaling*

---

**Description**

Wrapper of condSmacof, which finds a low-dimensional embedding of a given distance/dissimilarity matrix, conditioning on available manifold information.

**Usage**

```
condMDS(d, V, u.dim, W,
        method = c('matrix', 'vector'), exact = TRUE,
        it.max = 1000, gamma = 1e-05,
        init = c('none', 'eigen', 'user'),
        U.start, B.start)
```

**Arguments**

<code>d</code>	a distance/dissimilarity matrix of $N$ entities (or a <code>dist</code> object).
<code>V</code>	an $N \times q$ matrix of $q$ manifold auxiliary parameter values of the $N$ entities.
<code>u.dim</code>	the embedding dimension.
<code>W</code>	an $N \times N$ symmetric weight matrix. If not given, a matrix of ones will be used.
<code>method</code>	if <code>matrix</code> , there are no restrictions for the B matrix . If <code>vector</code> , the B matrix is restricted to be diagonal. The latter is more efficient for large $q$ .
<code>exact</code>	only relevant if <code>W</code> is not given. In this case, if <code>exact == FALSE</code> , <code>U</code> is updated by the large- $N$ approximation formula.
<code>it.max</code>	the max number of conditional SMACOF iterations.
<code>gamma</code>	conditional SMACOF stops early if the reduction of normalized conditional stress is less than <code>gamma</code>
<code>init</code>	initialization method.
<code>U.start</code>	user-defined starting values for the embedding (when <code>init = 'user'</code> )
<code>B.start</code>	starting B matrix.

**Value**

<code>U</code>	the embedding result.
<code>B</code>	the estimated B matrix.
<code>stress</code>	Normalized conditional stress value.
<code>sigma</code>	the conditional stress value at each iteration.
<code>init</code>	the value of the <code>init</code> argument.
<code>U.start</code>	the starting values for the embedding.
<code>B.start</code>	starting values for the B matrix.
<code>method</code>	the value of the <code>method</code> argument.
<code>exact</code>	the value of the <code>exact</code> argument.

**Author(s)**

Anh Tuan Bui

**References**

- Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.
- Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. Pattern Recognition Letters 164, 148-152. <https://doi.org/10.1016/j.patrec.2022.11.007>

**See Also**

[condSmacof](#), [condMDSeigen](#), [condIsomap](#)

**Examples**

```
# see help(cm1)
```

---

`condMDSeigen`*Conditional Multidimensional Scaling With Closed-Form Solution*

---

**Description**

Provides a closed-form solution for conditional multidimensional scaling, based on multiple linear regression and eigendecomposition.

**Usage**

```
condMDSeigen(d, V, u.dim, method = c('matrix', 'vector'))
```

**Arguments**

<code>d</code>	a <code>dist</code> object of $N$ entities.
<code>V</code>	an $N \times q$ matrix of $q$ manifold auxiliary parameter values of the $N$ entities.
<code>u.dim</code>	the embedding dimension.
<code>method</code>	if <code>matrix</code> , there are no restrictions for the B matrix . If <code>vector</code> , the B matrix is restricted to be diagonal.

**Value**

<code>U</code>	the embedding result.
<code>B</code>	the estimated B matrix.
<code>eig</code>	the computed eigenvalues.
<code>stress</code>	the corresponding normalized conditional stress value of the solution.

**Author(s)**

Anh Tuan Bui

**References**

Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. *Pattern Recognition Letters* 164, 148-152. <https://doi.org/10.1016/j.patrec.2022.11.007>

**See Also**

[condMDS](#), [condIsomap](#)

**Examples**

```
# see help(cm1)
```



---

condSmacof	<i>Conditional SMACOF</i>
------------	---------------------------

---

**Description**

Conditional SMACOF algorithms. Intended for internal usage.

**Usage**

```
condSmacof(d, V, u.dim, W,
           method = c('matrix', 'vector'), exact = TRUE,
           it.max = 1000, gamma = 1e-05,
           init = c('none', 'eigen', 'user'),
           U.start, B.start)
```

**Arguments**

<code>d</code>	a <code>dist</code> object of $N$ entities.
<code>V</code>	an $N \times q$ matrix of $q$ manifold auxiliary parameter values of the $N$ entities.
<code>u.dim</code>	the embedding dimension.
<code>W</code>	an $N \times N$ symmetric weight matrix. If not given, a matrix of ones will be used.
<code>method</code>	if <code>matrix</code> , there are no restrictions for the B matrix . If <code>vector</code> , the B matrix is restricted to be diagonal. The latter is more efficient for large $q$ .
<code>exact</code>	only relevant if <code>W</code> is not given. In this case, if <code>exact == FALSE</code> , <code>U</code> is updated by the large- $N$ approximation formula.
<code>it.max</code>	the max number of conditional SMACOF iterations.
<code>gamma</code>	conditional SMACOF stops early if the reduction of normalized conditional stress is less than <code>gamma</code>
<code>init</code>	initialization method.
<code>U.start</code>	user-defined starting values for the embedding (when <code>init = 'user'</code> )
<code>B.start</code>	starting B matrix.

**Value**

<code>U</code>	the embedding result.
<code>B</code>	the estimated B matrix.
<code>stress</code>	Normalized conditional stress value.
<code>sigma</code>	the conditional stress value at each iteration.
<code>init</code>	the value of the <code>init</code> argument.
<code>U.start</code>	the starting values for the embedding.
<code>B.start</code>	starting values for the B matrix.
<code>method</code>	the value of the <code>method</code> argument.
<code>exact</code>	the value of the <code>exact</code> argument.

**Author(s)**

Anh Tuan Bui

**References**

Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.

Bui, A. T. (2022). A Closed-Form Solution for Conditional Multidimensional Scaling. Pattern Recognition Letters. <https://doi.org/10.1016/j.patrec.2022.11.007>

---

cz

$C(Z)$

---

**Description**

Internal function.

**Usage**

`cz(w, d, dz)`

**Arguments**

w            the `dist` object of a weight matrix.  
d            the `dist` object of a distance/dissimilarity matrix.  
dz           the `dist` object of conditional distances.

**Value**

the matrix  $C(Z)$

**Author(s)**

Anh Tuan Bui

**References**

Bui, A.T. (2021). Dimension Reduction with Prior Information for Knowledge Discovery. arXiv:2111.13646. <https://arxiv.org/abs/2111.13646>.

---

mpinv	<i>Moore-Penrose Inverse</i>
-------	------------------------------

---

**Description**

Computes the Moore-Penrose inverse (a.k.a., generalized inverse or pseudoinverse) of a matrix based on singular-value decomposition (SVD).

**Usage**

```
mpinv(A, eps = sqrt(.Machine$double.eps))
```

**Arguments**

A	a matrix of real numbers.
eps	a threshold (to be multiplied with the largest singular value) for dropping SVD parts that correspond to small singular values.

**Value**

the Moore-Penrose inverse.

**Author(s)**

Anh Tuan Bui

**Examples**

```
mpinv(2*diag(4))
```

# Index

\* **generalized inverse**

mpinv, 11

\* **pseudoinverse**

mpinv, 11

ccor, 3

cml (cml-package), 2

cml-package, 2

condDist, 4

condDist2 (condDist), 4

condIsomap, 5, 7, 8

condMDS, 5, 6, 6, 8

condMDSeigen, 6, 7, 8

condSmacof, 7, 9

cz, 10

mpinv, 11