Package ‘cml’

May 8, 2026

Type Package

Title Conditional Manifold Learning

Version 0.3.1

Imports vegan

Description Finds a low-dimensional embedding of high-dimensional data, conditioning on available manifold information.

License GPL-3

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Author Anh Tuan Bui [aut, cre]

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

Repository CRAN

Date/Publication 2026-02-12 00:30:02 UTC

Contents

cml-package *Conditional Manifold Learning*

Description

Find a low-dimensional embedding of high-dimensional data, conditioning on auxiliary manifold information. The current version supports conditional MDS and conditional ISOMAP.

Please cite this package as follows: Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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 function.

`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>

References

Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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

Bui, A.T. (2026). Conditional Multidimensional Scaling for Incomplete Conditioning Data. Journal of Multivariate Analysis. Accepted.

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
```

```
# Conditional MDS, using the first 4 factors as known features
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, using the first 4 factors as known features
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

# with missing values in V
V = data.hat[,1:4]
V[1, 1] = NA
u.cmds2 = condMDS(d.hat, V, 3, init='eigen')
u.cmds2$B # compare with diag(factor.weights[1:4])
ccor(data.hat[,5:7], u.cmds2$U)$cancor # canonical correlations
vegan::procrustes(data.hat[,5:7], u.cmds$U, symmetric = TRUE)$ss # Procrustes statistic
u.cmds2$V.hat[1,1] # imputed value for V[1, 1]; the ground truth is data[1,1]
```

`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:

`cancor`	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 *Conditional Euclidean distance*

Description

Internal functions.

Usage

```
condDist(U, V.tilde, one_n_t=t(rep(1,nrow(U))))
condDist2(U, V.tilde2, one_n_t=t(rep(1,nrow(U))))
```

Arguments

U	the embedding U
V.tilde	= V %*% B
V.tilde2	= 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. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

`condIsomap` *Conditional ISOMAP*

Description

Finds a low-dimensional manifold embedding of a given distance/dissimilarity matrix, conditioning on auxiliary manifold parameters. 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', 'condSmacof', 'eigen', 'user'),
                        U.start, B.start,
                        V.tilde.start = NULL, ...)
```

Arguments

`d`	a distance/dissimilarity matrix of N entities (or a `dist` object).
`V`	an Nxq matrix of q manifold auxiliary parameter values of the N entities.
`u.dim`	the embedding dimension.
`epsilon`	shortest dissimilarity retained.
`k`	Number of shortest dissimilarities retained for a point. If both `epsilon` and `k` are given, `epsilon` will be used.
`W`	an NxN symmetric weight matrix. If not given, a matrix of ones will be used.
`method`	if `matrix`, there are no restrictions for the B matrix . If `vector`, the B matrix is restricted to be diagonal. The latter is more efficient for large q.
`exact`	If `FALSE`, use the large-N approximation formula to update the embedding.
`it.max`	the max number of conditional SMACOF iterations.
`gamma`	conditional SMACOF stops early if the reduction of normalized conditional stress is less than `gamma`
`init`	initialization method.
`U.start`	user-defined starting values for the embedding (when `init = 'user'`)
`B.start`	starting `B` matrix.
`V.tilde.start`	starting `V.tilde` matrix.
`...`	other arguments for the `isomap{vegan}` 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.
`V.tilde.start`	starting values for the `V.tilde` matrix.
`V.hat`	imputed `V` matrix, if `V` contains rows with missing values.
`method`	the value of the `method` argument.
`exact`	the value of the `exact` argument.

Author(s)

Anh Tuan Bui

References

Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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

Bui, A.T. (2026). Conditional Multidimensional Scaling for Incomplete Conditioning Data. Journal of Multivariate Analysis. Accepted.

See Also

`condMDS`

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 auxiliary manifold parameters.

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, V.tilde.start = NULL)
```

Arguments

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

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.

`V.tilde.start`	starting values for the `V.tilde` matrix.
`V.hat`	imputed `V` matrix, if `V` contains rows with missing values.
`method`	the value of the `method` argument.
`exact`	the value of the `exact` argument.

Author(s)

Anh Tuan Bui

References

Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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

Bui, A.T. (2026). Conditional Multidimensional Scaling for Incomplete Conditioning Data. Journal of Multivariate Analysis. Accepted.

See Also

`condSmacof`, `condSmacof_mer`, `condMDSeigen`, `condIsomap`

Examples

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

`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

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

Value

`U`	the embedding result.
`B`	the estimated `B` matrix.
`eig`	the computed eigenvalues.
`stress`	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(cml)
```

---

condSmacof *Conditional SMACOF*

Description

The conditional SMACOF algorithm. 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

`d`	a `dist` object of N entities.
`V`	an Nxq matrix of q manifold auxiliary parameter values of the N entities.
`u.dim`	the embedding dimension.
`W`	an NxN symmetric weight matrix. If not given, a matrix of ones will be used.
`method`	if `matrix`, there are no restrictions for the B matrix . If `vector`, the B matrix is restricted to be diagonal. The latter is more efficient for large q.

exact	If FALSE, use the large-N approximation formula to update the embedding.
it.max	the max number of iterations.
gamma	stops early if the reduction of normalized conditional stress is less than gamma
init	initialization method.
U.start	user-defined starting values for the embedding (when init = 'user').
B.start	user-defined starting values for the B matrix (when init = 'user').

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.

Author(s)

Anh Tuan Bui

References

Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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

---

condSmacof_mer *Conditional SMACOF with incomplete conditioning data (missing entire row)*

Description

Conditional SMACOF with missing known feature values. If a row has a missing value in a known feature, the algorithm treats all other known feature values of that row as missing. The algorithm also imputes the actual missing value(s). Intended for internal usage.

Usage

```
condSmacof_mer(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, V.tilde.start)
```

Arguments

d	a dist object of N entities.
V	an Nxq matrix of q manifold auxiliary parameter values of the N entities.
u.dim	the embedding dimension.
W	an NxN symmetric weight matrix. If not given, a matrix of ones will be used.
method	if matrix, there are no restrictions for the B matrix . If vector, the B matrix is restricted to be diagonal. The latter is more efficient for large q. method = 'vector' is not available in this version.
exact	If FALSE, use the large-N approximation formula to update the embedding.
it.max	the max number of iterations.
gamma	stops early if the reduction of normalized conditional stress is less than gamma
init	initialization method. Except for init = 'user', in which case the user directly provides starting values, the other cases apply condMDS with init = 'none') (if init = 'none') or condMDSeigen (if init = 'eigen') to the complete data to initialize B and part of the embedding.
U.start	user-defined starting values for the embedding (when init = 'user').
B.start	user-defined starting values for the B matrix (when init = 'user').
V.tilde.start	user-defined starting values for the V.tilde matrix (when init = 'user').

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	the starting values for the B matrix.
V.tilde.start	the starting values for V.tilde.start.
V.hat	imputed V matrix, if V contains rows with missing values.
method	the value of the method argument.

Author(s)

Anh Tuan Bui

References

Bui, A.T. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

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

Bui, A.T. (2026). Conditional Multidimensional Scaling for Incomplete Conditioning Data. Journal of Multivariate Analysis. Accepted.

`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. (2024). Dimension Reduction with Prior Information for Knowledge Discovery. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46, 3625-3636. https://doi.org/10.1109/TPAMI.2023.3346212

`mpinv` *Moore-Penrose Inverse*

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