Package ‘affinitymatrix’

July 22, 2025

Title Estimation of Affinity Matrix

Version 0.1.0

Description Tools to study sorting patterns in matching markets and to estimate the affinity matrix of both the bipartite one-to-one matching model without frictions and with Transferable Utility by 'Dupuy' and 'Galichon' (2014) <doi:10.1086/677191> and its 'unipartite' variant by 'Ciscato', 'Galichon' and 'Gousse' (2020) <doi:10.1086/704611>. It also contains all the necessary tools to implement the 'saliency' analysis, to run rank tests of the affinity matrix and to build tables and plots summarizing the findings.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports stats, utils, grid, expm, MASS, Hmisc, ggplot2, ggrepel

Suggests testthat

NeedsCompilation no

Author Edoardo Ciscato [aut, cre] (ORCID: <https://orcid.org/0000-0002-9894-1778>)

Maintainer Edoardo Ciscato <edoardo.ciscato@kuleuven.be>

Repository CRAN

Date/Publication 2020-11-26 17:00:02 UTC

Contents

`estimate.affinity.matrix` *Estimate Dupuy and Galichon's model*

Description

This function estimates the affinity matrix of the matching model of Dupuy and Galichon (2014), performs the saliency analysis and the rank tests. The user must supply a *matched sample* that is treated as the equilibrium matching of a bipartite one-to-one matching model without frictions and with Transferable Utility. For the sake of clarity, in the documentation we take the example of the marriage market and refer to "men" as the observations on one side of the market and to "women" as the observations on the other side. Other applications may include matching between CEOs and firms, firms and workers, buyers and sellers, etc.

Usage

```
estimate.affinity.matrix(
  X,
  Y,
  w = rep(1, N),
  A0 = matrix(0, nrow = Kx, ncol = Ky),
  lb = matrix(-Inf, nrow = Kx, ncol = Ky),
  ub = matrix(Inf, nrow = Kx, ncol = Ky),
  pr = 0.05,
  max_iter = 10000,
  tol_level = 1e-06,
  scale = 1,
  nB = 2000,
  verbose = TRUE
)
```

Arguments

`X` The matrix of men's traits. Its rows must be ordered so that the i-th man is matched with the i-th woman: this means that `nrow(X)` must be equal to `nrow(Y)`. Its columns correspond to the different matching variables: `ncol(X)` can be different from `ncol(Y)`. For the sake of clarity of exposition when using descriptive tools such as `show.correlations`, it is recommended assigning the same matching variable to the k-th column of `X` and to the k-th column of `Y`, whenever possible. If `X` has more matching variables than `Y`, then those variables that appear in `X` but no in Y should be found in the last columns of `X` (and vice versa). The matrix is demeaned and rescaled before the start of the estimation algorithm.

`Y` The matrix of women's traits. Its rows must be ordered so that the i-th woman is matched with the i-th man: this means that `nrow(Y)` must be equal to `nrow(X)`. Its columns correspond to the different matching variables: `ncol(Y)` can be different from `ncol(X)`. The matrix is demeaned and rescaled before the start of the estimation algorithm.

`w` A vector of sample weights with length `nrow(X)`. Defaults to uniform weights.

`A0` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the initial values of the affinity matrix to be fed to the estimation algorithm. Optional. Defaults to matrix of zeros.

`lb` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the lower bounds of the elements of the affinity matrix. Defaults to `-Inf` for all parameters.

`ub` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the upper bounds of the elements of the affinity matrix. Defaults to `Inf` for all parameters.

`pr` A probability indicating the significance level used to compute bootstrap two-sided confidence intervals for `U`, `V` and `lambda`. Defaults to 0.05.

`max_iter` An integer indicating the maximum number of iterations in the Maximum Likelihood Estimation. See `optim` for the `"L-BFGS-B"` method. Defaults to 10000.

`tol_level` A positive real number indicating the tolerance level in the Maximum Likelihood Estimation. See `optim` for the `"L-BFGS-B"` method. Defaults to 1e-6.

`scale` A positive real number indicating the scale of the model. Defaults to 1.

`nB` An integer indicating the number of bootstrap replications used to compute the confidence intervals of `U`, `V` and `lambda`. Defaults to 2000.

`verbose` If `TRUE`, the function displays messages to keep track of its progress. Defaults to `TRUE`.

Value

The function returns a list with elements: `X`, the demeaned and rescaled matrix of men's traits; `Y`, the demeaned and rescaled matrix of men's traits; `fx`, the empirical marginal distribution of men; `fy`, the empirical marginal distribution of women; `Aopt`, the estimated affinity matrix; `sdA`, the standard errors of `Aopt`; `tA`, the Z-test statistics of `Aopt`; `VarCovA`, the full variance-covariance matrix of `Aopt`; `rank.tests`, a list with all the summaries of the rank tests on `Aopt`; `U`, whose columns are the left-singular vectors of `Aopt`; `V`, whose columns are the right-singular vectors of `Aopt`; `lambda`, whose elements are the singular values of `Aopt`; `UCI`, whose columns are the lower and the upper bounds of the confidence intervals of `U`; `VCI`, whose columns are the lower and the upper bounds of the confidence intervals of `V`; `lambdaCI`, whose columns are the lower and the upper bounds of the confidence intervals of `lambda`; `df.bootstrap`, a data frame resulting from the `nB` bootstrap replications and used to infer the empirical distribution of the estimated objects.

See Also

Dupuy, Arnaud, and Alfred Galichon. "Personality traits and the marriage market." *Journal of Political Economy* 122, no. 6 (2014): 1271-1319.

Examples

```
# Parameters
Kx = 4; Ky = 4; # number of matching variables on both sides of the market
```

```
N = 200 # sample size
mu = rep(0, Kx+Ky) # means of the data generating process
Sigma = matrix(c(1, 0.326, 0.1446, -0.0668, 0.5712, 0.4277, 0.1847, -0.2883,
                 0.326, 1, -0.0372, 0.0215, 0.2795, 0.8471, 0.1211, -0.0902,
                 0.1446, -0.0372, 1, -0.0244, 0.2186, 0.0636, 0.1489,
                 -0.1301, -0.0668, 0.0215, -0.0244, 1, 0.0192, 0.0452,
                 -0.0553, 0.2717, 0.5712, 0.2795, 0.2186, 0.0192, 1, 0.3309,
                 0.1324, -0.1896, 0.4277, 0.8471, 0.0636, 0.0452, 0.3309, 1,
                 0.0915, -0.1299, 0.1847, 0.1211, 0.1489, -0.0553, 0.1324,
                 0.0915, 1, -0.1959, -0.2883, -0.0902, -0.1301, 0.2717,
                 -0.1896, -0.1299, -0.1959, 1),
               nrow=Kx+Ky) # (normalized) variance-covariance matrix of the
               # data generating process
labels_x = c("Educ.", "Age", "Height", "BMI") # labels for men's matching variables
labels_y = c("Educ.", "Age", "Height", "BMI") # labels for women's matching variables

# Sample
data = MASS::mvrnorm(N, mu, Sigma) # generating sample
X = data[,1:Kx]; Y = data[,Kx+1:Ky] # men's and women's sample data
w = sort(runif(N-1)); w = c(w,1) - c(0,w) # sample weights

# Main estimation
res = estimate.affinity.matrix(X, Y, w = w, nB = 500)

# Summarize results
show.affinity.matrix(res, labels_x = labels_x, labels_y = labels_y)
show.diagonal(res, labels = labels_x)
show.test(res)
show.saliency(res, labels_x = labels_x, labels_y = labels_y,
              ncol_x = 2, ncol_y = 2)
show.correlations(res, labels_x = labels_x, labels_y = labels_y,
                  label_x_axis = "Husband", label_y_axis = "Wife", ndims = 2)
```

estimate.affinity.matrix.lowrank

Estimate Dupuy and Galichon's model

Description

This function estimates the affinity matrix of the matching model of Dupuy and Galichon (2014) under a rank restriction on the affinity matrix, as suggested by Dupuy, Galichon and Sun (2019). In their own words, "to accommodate high dimensionality of the data, they propose a novel method that incorporates a nuclear norm regularization which effectively enforces a rank constraint on the affinity matrix." This function also performs the saliency analysis and the rank tests. The user must supply a *matched sample* that is treated as the equilibrium matching of a bipartite one-to-one matching model without frictions and with Transferable Utility. For the sake of clarity, in the documentation we take the example of the marriage market and refer to "men" as the observations on one side of the market and to "women" as the observations on the other side. Other applications may include matching between CEOs and firms, firms and workers, buyers and sellers, etc.

Usage

```
estimate.affinity.matrix.lowrank(
  X,
  Y,
  w = rep(1, N),
  A0 = matrix(0, nrow = Kx, ncol = Ky),
  lb = matrix(-Inf, nrow = Kx, ncol = Ky),
  ub = matrix(Inf, nrow = Kx, ncol = Ky),
  pr = 0.05,
  max_iter = 10000,
  tol_level = 1e-08,
  tau = 1,
  scale = 1,
  cross_validation = TRUE,
  manual_lambda = 0,
  lambda_min = 0,
  Nfolds = 5,
  nB = 2000,
  verbose = TRUE
)
```

Arguments

`X` The matrix of men's traits. Its rows must be ordered so that the i-th man is matched with the i-th woman: this means that `nrow(X)` must be equal to `nrow(Y)`. Its columns correspond to the different matching variables: `ncol(X)` can be different from `ncol(Y)`. For the sake of clarity of exposition when using descriptive tools such as `show.correlations`, it is recommended assigning the same matching variable to the k-th column of `X` and to the k-th column of `Y`, whenever possible. If `X` has more matching variables than `Y`, then those variables that appear in `X` but no in Y should be found in the last columns of `X` (and vice versa). The matrix is demeaned and rescaled before the start of the estimation algorithm.

`Y` The matrix of women's traits. Its rows must be ordered so that the i-th woman is matched with the i-th man: this means that `nrow(Y)` must be equal to `nrow(X)`. Its columns correspond to the different matching variables: `ncol(Y)` can be different from `ncol(X)`. The matrix is demeaned and rescaled before the start of the estimation algorithm.

`w` A vector of sample weights with length `nrow(X)`. Defaults to uniform weights.

`A0` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the initial values of the affinity matrix to be fed to the estimation algorithm. Optional. Defaults to matrix of zeros.

`lb` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the lower bounds of the elements of the affinity matrix. Defaults to `-Inf` for all parameters.

`ub` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the upper bounds of the elements of the affinity matrix. Defaults to `Inf` for all parameters.

`pr` A probability indicating the significance level used to compute bootstrap two-sided confidence intervals for `U`, `V` and `lambda`. Defaults to 0.05.

`max_iter` An integer indicating the maximum number of iterations in the proximal gradient descent algorithm. Defaults to 10000.

`tol_level` A positive real number indicating the tolerance level in the proximal gradient descent algorithm. Defaults to 1e-8.

`tau` A positive real number indicating a sensitivity parameter in the proximal gradient descent algorithm. Defaults to 1 and should not be changed unless computational problems arise.

`scale` A positive real number indicating the scale of the model. Defaults to 1.

`cross_validation`
If `TRUE`, the function looks for a rank restriction through cross validation. The cross validation exercise aims to minimize the covariance mismatch: in other words, it avoids overfitting without excessively reducing the number of free parameters. Defaults to `TRUE`.

`manual_lambda` A positive real number indicating the user-supply `lambda` when `cross_validation==FALSE`. The higher `lambda`, the tighter the rank restriction. Defaults to 0.

`lambda_min` A positive real number indicating minimum value for `lambda` considered during the cross validation. We recommend using 0, but with a high number of matching variables relatively to the sample size it is reasonable to set `lambda_min` to a higher value. Defaults to 0.

`Nfolds` An integer indicating the number of folds in the cross validation. Defaults to 5 and can be increased with a large sample size.

`nB` An integer indicating the number of bootstrap replications used to compute the confidence intervals of `Aopt`, `U`, `V` and `lambda`. Defaults to 2000.

`verbose` If `TRUE`, the function displays messages to keep track of its progress. Defaults to `TRUE`.

Value

The function returns a list with elements: `X`, the demeaned and rescaled matrix of men's traits; `Y`, the demeaned and rescaled matrix of men's traits; `fx`, the empirical marginal distribution of men; `fy`, the empirical marginal distribution of women; `Aopt`, the estimated affinity matrix; `sdA`, the standard errors of `Aopt`; `tA`, the Z-test statistics of `Aopt`; `VarCovA`, the full variance-covariance matrix of `Aopt`; `rank.tests`, a list with all the summaries of the rank tests on `Aopt`; `U`, whose columns are the left-singular vectors of `Aopt`; `V`, whose columns are the right-singular vectors of `Aopt`; `lambda`, whose elements are the singular values of `Aopt`; `UCI`, whose columns are the lower and the upper bounds of the confidence intervals of `U`; `VCI`, whose columns are the lower and the upper bounds of the confidence intervals of `V`; `lambdaCI`, whose columns are the lower and the upper bounds of the confidence intervals of `lambda`; `df.bootstrap`, a data frame resulting from the `nB` bootstrap replications and used to infer the empirical distribution of the estimated objects; `lambda.rank.restriction`, a positive real number indicating the value of the Lagrange multiplier of the nuclear norm constraint of the affinity matrix, either chosen by the user or through Cross Validation; `df.cross.validation`, a data frame containing the detailed results of the cross validation exercise.

See Also

Dupuy, Arnaud, Alfred Galichon, and Yifei Sun. "Estimating matching affinity matrices under low-rank constraints." *Information and Inference: A Journal of the IMA* 8, no. 4 (2019): 677-689. **Dupuy, Arnaud, and Alfred Galichon**. "Personality traits and the marriage market." *Journal of Political Economy* 122, no. 6 (2014): 1271-1319.

Examples

```
# Parameters
Kx = 2; Ky = 2; # number of matching variables on both sides of the market
N = 100 # sample size
mu = rep(0, Kx+Ky) # means of the data generating process
Sigma = matrix(c(1, -0.0244, 0.1489, -0.1301, -0.0244, 1, -0.0553, 0.2717,
                 0.1489, -0.0553, 1, -0.1959, -0.1301, 0.2717, -0.1959, 1),
                 nrow=Kx+Ky)
    # (normalized) variance-covariance matrix of the data generating process
labels_x = c("Height", "BMI") # labels for men's matching variables
labels_y = c("Height", "BMI") # labels for women's matching variables

# Sample
data = MASS::mvrnorm(N, mu, Sigma) # generating sample
X = data[,1:Kx]; Y = data[,Kx+1:Ky] # men's and women's sample data
w = sort(runif(N-1)); w = c(w,1) - c(0,w) # sample weights

# Main estimation
res = estimate.affinity.matrix.lowrank(X, Y, w = w, tol_level = 1e-03,
                                       nB = 50, Nfolds = 2)

# Summarize results
show.affinity.matrix(res, labels_x = labels_x, labels_y = labels_y)
show.diagonal(res, labels = labels_x)
show.test(res)
show.saliency(res, labels_x = labels_x, labels_y = labels_y,
              ncol_x = 2, ncol_y = 2)
show.cross.validation(res)
show.correlations(res, labels_x = labels_x, labels_y = labels_y,
                  label_x_axis = "Husband", label_y_axis = "Wife", ndims = 2)
```

---

estimate.affinity.matrix.unipartite

Estimate Ciscato, Galichon and Gousse's model

Description

This function estimates the affinity matrix of the matching model of Ciscato Gousse and Galichon (2020), performs the saliency analysis and the rank tests. The user must supply a *matched sample* that is treated as the equilibrium matching of a bipartite one-to-one matching model without frictions and with Transferable Utility. The model differs from the original Dupuy and Galichon (2014) since all agents are pooled in one group and can match within the group. For the sake of clarity, in the documentation we take the example of the same-sex marriage market and refer to "first partner" and "second partner" in order to distinguish between the arbitrary partner order in a database (e.g., survey respondent and partner of the respondent). Note that in this case the variable "sex" is treated as a matching variable rather than a criterion to assign partners to one side of the market as in the bipartite case. Other applications may include matching between coworkers, roommates or teammates.

Usage

```
estimate.affinity.matrix.unipartite(
  X,
  Y,
  w = rep(1, N),
  A0 = matrix(0, nrow = K, ncol = K),
  lb = matrix(-Inf, nrow = K, ncol = K),
  ub = matrix(Inf, nrow = K, ncol = K),
  pr = 0.05,
  max_iter = 10000,
  tol_level = 1e-06,
  scale = 1,
  nB = 2000,
  verbose = TRUE
)
```

Arguments

`X` The matrix of traits of the first partner. Its rows must be ordered so that the i-th individual in `X` is matched with the i-th partner in `Y`: this means that `nrow(X)` must be equal to `nrow(Y)`. Its columns correspond to the different matching variables: `ncol(X)` must be equal to `ncol(Y)` and the variables must be sorted in the same way in both matrices. The matrix is demeaned and rescaled before the start of the estimation algorithm.

`Y` The matrix of traits of the second partner. Its rows must be ordered so that the i-th individual in `Y` is matched with the i-th partner in `X`: this means that `nrow(Y)` must be equal to `nrow(X)`. Its columns correspond to the different matching variables: `ncol(Y)` must be equal to `ncol(X)` and the variables must be sorted in the same way in both matrices. The matrix is demeaned and rescaled before the start of the estimation algorithm.

`w` A vector of sample weights with length `nrow(X)`. Defaults to uniform weights.

`A0` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the initial values of the affinity matrix to be fed to the estimation algorithm. Optional. Defaults to a matrix of zeros.

`lb` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the lower bounds of the elements of the affinity matrix. Defaults to `-Inf` for all parameters.

`ub` A vector or matrix with `ncol(X)*ncol(Y)` elements corresponding to the upper bounds of the elements of the affinity matrix. Defaults to `Inf` for all parameters.

pr A probability indicating the significance level used to compute bootstrap two-sided confidence intervals for U, V and lambda. Defaults to 0.05.

max_iter An integer indicating the maximum number of iterations in the Maximum Likelihood Estimation. See optim for the "L-BFGS-B" method. Defaults to 10000.

tol_level A positive real number indicating the tolerance level in the Maximum Likelihood Estimation. See optim for the "L-BFGS-B" method. Defaults to 1e-6.

scale A positive real number indicating the scale of the model. Defaults to 1.

nB An integer indicating the number of bootstrap replications used to compute the confidence intervals of U, V and lambda. Defaults to 2000.

verbose If TRUE, the function displays messages to keep track of its progress. Defaults to TRUE.

Value

The function returns a list with elements: X, the demeaned and rescaled matrix of traits of the first partner; Y, the demeaned and rescaled matrix of traits of the second partner; fx, the empirical marginal distribution of first partners; fy, the empirical marginal distribution of second partners; Aopt, the estimated affinity matrix; sdA, the standard errors of Aopt; tA, the Z-test statistics of Aopt; VarCovA, the full variance-covariance matrix of Aopt; rank.tests, a list with all the summaries of the rank tests on Aopt; U, whose columns are the left-singular vectors of Aopt; V, whose columns are the right-singular vectors of Aopt; lambda, whose elements are the singular values of Aopt; UCI, whose columns are the lower and the upper bounds of the confidence intervals of U; VCI, whose columns are the lower and the upper bounds of the confidence intervals of V; lambdaCI, whose columns are the lower and the upper bounds of the confidence intervals of lambda; df.bootstrap, a data frame resulting from the nB bootstrap replications and used to infer the empirical distribution of the estimated objects.

See Also

Ciscato, Edoardo, Alfred Galichon, and Marion Gousse. "Like attract like? a structural comparison of homogamy across same-sex and different-sex households." *Journal of Political Economy* 128, no. 2 (2020): 740-781. **Dupuy, Arnaud, and Alfred Galichon**. "Personality traits and the marriage market." *Journal of Political Economy* 122, no. 6 (2014): 1271-1319.

Examples

```
# Parameters
K = 4 # number of matching variables
N = 100 # sample size
mu = rep(0, 2*K) # means of the data generating process
Sigma = matrix(c(1, -0.0992, 0.0443, -0.0246, -0.8145, 0.083, -0.0438,
    0.0357, -0.0992, 1, 0.0699, -0.0043, 0.083, 0.8463, 0.0699, -0.0129, 0.0443,
    0.0699, 1, -0.0434, -0.0438, 0.0699, 0.5127, -0.0383, -0.0246, -0.0043,
    -0.0434, 1, 0.0357, -0.0129, -0.0383, 0.6259, -0.8145, 0.083, -0.0438,
    0.0357, 1, -0.0992, 0.0443, -0.0246, 0.083, 0.8463, 0.0699, -0.0129, -0.0992,
    1, 0.0699, -0.0043, -0.0438, 0.0699, 0.5127, -0.0383, 0.0443, 0.0699, 1,
    -0.0434, 0.0357, -0.0129, -0.0383, 0.6259, -0.0246, -0.0043, -0.0434, 1),
               nrow=K+K) # (normalized) variance-covariance matrix of the
               # data generating process with a block symmetric structure
```

```
labels = c("Sex", "Age", "Educ.", "Black") # labels for matching variables

# Sample
data = MASS::mvrnorm(N, mu, Sigma) # generating sample
X = data[,1:K]; Y = data[,K+1:K] # men's and women's sample data
w = sort(runif(N-1)); w = c(w,1) - c(0,w) # sample weights

# Main estimation
res = estimate.affinity.matrix.unipartite(X, Y, w = w, nB = 500)

# Summarize results
show.affinity.matrix(res, labels_x = labels, labels_y = labels)
show.diagonal(res, labels = labels)
show.test(res)
show.saliency(res, labels_x = labels, labels_y = labels,
              ncol_x = 2, ncol_y = 2)
show.correlations(res, labels_x = labels, labels_y = labels,
                  label_x_axis = "First partner",
                  label_y_axis = "Second partner", ndims = 2)
```

`export.table` *Export an affinitymatrix table*

Description

The function stores a LaTeX style table in a txt file.

Usage

```
export.table(tabular, name = "table", path = getwd())
```

Arguments

`tabular` A long string corresponding to the output of `show.affinity.matrix`, `show.diagonal` or `show.test`, or one of the two elements of `show.saliency` (`U.table` or `V.table`).

`name` A string indicating the name of the txt file. Defaults to `"affinity_matrix"`.

`path` A string indicating the path where to save the txt file. Defaults to current path.

Value

The function stores a long string in LaTeX style that can be processed in the standard LaTeX tabular environment in a txt file in located in `path`.

show.affinity.matrix *Print affinity matrix*

Description

This function prints the estimated affinity matrix in LaTeX style. Standard errors are printed below the elements of the affinity matrix. Estimates that are significant at the pr level are printed in boldface: this format feature can be avoided by setting pr to 0.

Usage

```
show.affinity.matrix(
  res,
  labels_x = paste0("Trait ", 1:Kx),
  labels_y = paste0("Trait ", 1:Ky),
  pr = 0.05
)
```

Arguments

res — A list corresponding to the output of estimate.affinity.matrix, estimate.affinity.matrix.lowra or estimate.affinity.matrix.unipartite.

labels_x — A vector of strings indicating the names of men's matching variables. Defaults to "Trait k" for every k matching variable.

labels_y — A vector of strings indicating the names of women's matching variables. Defaults to "Trait k" for every k matching variable.

pr — A probability indicating the two-tailed significance level required for an estimated parameter to be printed in boldface. Defaults to 0.05 and can be set to 0 to avoid printing any estimate in boldface.

Value

The function returns a long string in LaTeX style that can be processed in the standard LaTeX tabular environment in order to display the estimates of the affinity matrix Aopt.

show.correlations *Print correlations of matching factors with matching and outcome variables*

Description

This function returns a list of plots, one for each of the first ndims orthogonal sorting dimension. In the k-th plot, the correlation between a man's observed matching variable and the man's k-th matching factor is plotted on the x-axis; the correlation between a woman's observed matching variable and the woman's k-th matching factor is plotted on the y-axis. In addition, the user can supply additional variables stored in the matrix Z that were not previously used in the estimation ("outcome variables"). The function prints the correlation between the outcome variable and the man's k-th matching factor on the x-axis, while the correlation between the outcome variable and the woman's k-th matching factor is on the y-axis.

Usage

```
show.correlations(
  res,
  Z = matrix(0, nrow = N, ncol = 0),
  labels_x = paste0("Trait ", 1:Kx),
  labels_y = paste0("Trait ", 1:Ky),
  labels_z = if (Kz > 0) paste0("Outcome ", 1:Kz) else c(),
  ndims = min(Kx, Ky, 10),
  pr = 0.02,
  color_arrows = c("black", "red"),
  size_arrows = 0.5,
  font_labels = c("bold", "italic"),
  label_x_axis = "First partner",
  label_y_axis = "Second partner"
)
```

Arguments

res — A list corresponding to the output of estimate.affinity.matrix, estimate.affinity.matrix.lowra or estimate.affinity.matrix.unipartite.

Z — A matrix Z with additional variables that were not previously used in the estimation. The i-th row of Z must contain information on the couple formed by the i-th row of X and the i-th row of Y, so that nrow(Z)=nrow(X). Defaults to an empty matrix: Z is optional.

labels_x — A vector of strings indicating the names of men's matching variables. Defaults to "Trait k" for every k matching variable.

labels_y — A vector of strings indicating the names of women's matching variables. Defaults to "Trait k" for every k matching variable.

labels_z — A vector of strings indicating the names of the outcome variables. Defaults to "Outcome k" for every k outcome variable.

ndims — An integer indicating the number of orthogonal matching dimensions that will be plotted. The function plots the first \ codendims dimensions. Defaults to all dimensions unless the latter are more than 10, in which case only the first 10 are plotted.

pr — A probability indicating the two-tailed significance level required for a matching or outcome variable to be displayed in a plot. In order to avoid having too many variables plotted at the same time, the function only selects those whose correlation with the matching factor is significantly different from zero (in a two-tailed test) at the `pr` level. Defaults to 0.02 and can be set to 1 to print all variables.

`color_arrows` A string or a vector of strings containing color names for the arrows. All matching variables are assigned the first color given in the vector, while all outcome variables are assigned the second color. See `ggplot`. Defaults to `"black"` and `"red"` respectively.

`size_arrows` A positive real number or a vector containing the size of the arrows. All matching variables are assigned the first size given in the vector, while all outcome variables are assigned the second size. See `ggplot`. Defaults to 0.5 for both.

`font_labels` A string or a vector of strings containing font types for the labels. All matching variables are assigned the first font type given in the vector, while all outcome variables are assigned the second font type. See `ggplot`. Defaults to `"bold"` and `"italic"` respectively.

`label_x_axis` A string containing a root for all x-axis names in different plots. Defaults to `"First partner"`.

`label_y_axis` A string containing a root for all y-axis names in different plots. Defaults to `"Second partner"`.

Value

The function returns a list of `ndims` plots created with `ggplot`.

See Also

Chiappori, Pierre-Andre, Edoardo Ciscato, and Carla Guerriero. "Analyzing matching patterns in marriage: theory and application to Italian data." *HCEO Working Paper* no. 2020-080 (2020).

`show.cross.validation` *Print cross validation summary*

Description

This function returns a plot reporting the estimated covariance mismatch as a function of the rank restriction parameter lambda. This is the result of the cross validation exercise. The function is expected to be convex in lambda and the chosen lambda is the unique minimum.

Usage

```
show.cross.validation(res)
```

Arguments

`res` A list corresponding to the output of `estimate.affinity.matrix`, `estimate.affinity.matrix.lowra` or `estimate.affinity.matrix.unipartite`.

Value

The function returns a plot created with `ggplot`.

show.diagonal *Print the diagonal of the affinity matrix*

Description

This function prints the estimates of the diagonal of the affinity matrix in LaTeX style. Standard errors are printed below the elements of the affinity matrix. Estimates that are significant at the `pr` level are printed in boldface: this format feature can be avoided by setting `pr` to 0.

Usage

```
show.diagonal(res, labels = paste0("Trait ", 1:K), pr = 0.05)
```

Arguments

`res` A list corresponding to the output of `estimate.affinity.matrix`, `estimate.affinity.matrix.lowra` or `estimate.affinity.matrix.unipartite`.

`labels` A vector of strings indicating the names of the matching variables. Defaults to `"Trait k"` for every `k` matching variable.

`pr` A probability indicating the two-tailed significance level required for an estimated parameter to be printed in boldface. Defaults to 0.05 and can be set to 0 to avoid printing any estimate in boldface.

Value

The function returns a long string in LaTeX style that can be processed in the standard LaTeX tabular environment in order to display the estimates of diagonal of the affinity matrix `Aopt`.

show.saliency *Print summary of saliency analysis*

Description

This function prints the results from the saliency analysis in LaTeX style. The function returns a list of two elements: `U.table` contains the first `ncol_x` vectors of loadings that map men's `Kx` observed traits into the first `ncol_x` matching factors; `V.table` contains the first `ncol_y` vectors of loadings that map women's `Ky` observed traits into the first `ncol_y` matching factors. In both tables, the last line reports the normalized singular values of the affinity matrix in descending order.

Usage

```
show.saliency(
  res,
  ncol_x = Kx,
  ncol_y = Ky,
  labels_x = paste0("Trait ", 1:Kx),
  labels_y = paste0("Trait ", 1:Ky),
  pr = 0.05
)
```

Arguments

`res` A list corresponding to the output of `estimate.affinity.matrix`, `estimate.affinity.matrix.lowra` or `estimate.affinity.matrix.unipartite`.

`ncol_x` An integer indicating the number of singular vector to print for men. The function prints the first `ncol_x` singular vectors. Defaults to `ncol(U)`.

`ncol_y` An integer indicating the number of singular vector to print for women. The function prints the first `ncol_y` singular vectors. Defaults to `ncol(V)`.

`labels_x` A vector of strings indicating the names of men's matching variables. Defaults to `"Trait k"` for every `k` matching variable.

`labels_y` A vector of strings indicating the names of women's matching variables. Defaults to `"Trait k"` for every `k` matching variable.

`pr` A probability indicating the two-tailed significance level required for an estimated parameter to be printed in boldface. Defaults to 0.05 and can be set to 0 to avoid printing any estimate in boldface.

Value

The function returns a long string in LaTeX style that can be processed in the standard LaTeX tabular environment in order to display the estimates of the vectors of loadings for the first `ncol_x` men's matching factors and the first `ncol_y` women's matching factors.

`show.test` *Print summaries of rank tests*

Description

This function prints the summaries of the first `n_tests` rank tests in in LaTeX. The first row specifies the null hypothesis, the second row gives the test statistic, the third the degrees of freedom and the fourth says whether the null hypothesis passes the test at the `pr` level.

Usage

```
show.test(res, pr = 0.05, n_tests = K - 1)
```

Arguments

`res` A list corresponding to the output of `estimate.affinity.matrix`, `estimate.affinity.matrix.lowra` or `estimate.affinity.matrix.unipartite`.

`pr` A probability indicating the significance level required to pass a rank test. Defaults to 0.05.

`n_tests` An integer indicating the number of tests to show. The function prints the first `n_tests` rank tests. Defaults to `min(nrow(Y),nrow(X))-1`.

Value

The function returns a long string in LaTeX style that can be processed in the standard LaTeX tabular environment in order to display the results from the first `n_tests` rank tests of the affinity matrix.

Index