Package ‘Compositional’

January 19, 2021

Type  Package
Title  Compositional Data Analysis
Version  4.3
URL
Date  2021-01-19
Author  Michail Tsagris [aut, cre],
        Giorgos Athineou [aut],
        Abdulaziz Alenazi [ctb]
Maintainer  Michail Tsagris <mtsagris@uoc.gr>
Depends  R (>= 4.0)
Imports  codalm, Directional, doParallel, emplik, FlexDir, foreach,
         MASS, mda, mixture, NlcOptim, nnet, parallel, RANN, Rfast,
         Rfast2, sn, stats
R topics documented:

Compositional-package .................................................. 5
Aithison's simple zero replacement strategy .......................... 6
All pairwise additive log-ratio transformations ...................... 7
Alpha generalised correlations between two compositional datasets .. 8
Beta regression ............................................................. 9
Choose the number of principal components via reconstruction error .... 10
Constrained linear least squares for compositional responses and predictors 11
Contour plot of a Dirichlet distribution in $S^2$ ....................... 12
Contour plot of a Flexible Dirichlet distribution in $S^2$ .............. 13
Contour plot of a Gaussian mixture model in $S^2$ ................... 14
Contour plot of a Kent distribution in $S^2$ ............................ 16
Contour plot of the folded model in $S^2$ ............................. 17
Contour plot of the kernel density estimate in $S^2$ ................. 18
Contour plot of the normal distribution in $S^2$ ........................ 19
Contour plot of the skew skew-normal distribution in $S^2$ ........... 20
Contour plot of the t distribution in $S^2$ ............................. 21
Cross validation for some compositional regression models ........... 22
Cross validation for the alpha-k-NN regression for compositional response data .... 24
Cross validation for the alpha-k-NN regression with compositional predictor variables . 26
Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation 27
Cross validation for the ridge regression ............................... 29
Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation 31
Cross validation for the transformation-free linear regression for compositional responses and predictors .......................... 32
Cross-validation for the constrained linear least squares for compositional responses and predictors .................................. 34
Density of the Dirichlet distribution .................................... 35
Density of the Flexible Dirichlet distribution ......................... 36
Density values of a Dirichlet distribution ................................ 37
Dirichlet random values simulation ................................. 38
Dirichlet regression ...................................................... 39
Distance based regression models for proportions .................. 41
topics documented:

- Divergence based regression for compositional data .................................................. 42
- Divergence based regression for compositional data with compositional data in the co-
  variates side using the alpha-transformation ......................................................... 44
- Divergence matrix of compositional data .................................................................. 46
- Empirical likelihood for a one sample mean vector hypothesis testing ...................... 47
- Empirical likelihood hypothesis testing for two mean vectors ..................................... 48
- Estimating location and scatter parameters for compositional data ......................... 50
- Estimation of the probability left outside the simplex when using the alpha-transformation 51
- Estimation of the value of alpha in the folded model .................................................. 52
- Exponential empirical likelihood for a one sample mean vector hypothesis testing ...... 55
- Exponential empirical likelihood hypothesis testing for two mean vectors .................. 56
- Fast estimation of the value of alpha ......................................................................... 58
- Fitting a Dirichlet distribution .................................................................................... 59
- Fitting a Dirichlet distribution via Newton-Rapshon .................................................. 60
- Gaussian mixture models for compositional data ......................................................... 63
- Generate random folds for cross-validation ................................................................ 65
- Greenacre’s power transformation ............................................................................. 66
- Helper Frechet mean for compositional data ............................................................... 67
- Helper functions for the Kullback-Leibler regression .................................................. 68
- Hotelling’s multivariate version of the 1 sample t-test for Euclidean data ..................... 70
- Hotelling’s multivariate version of the 2 sample t-test for Euclidean data ..................... 71
- Hypothesis testing for two or more compositional mean vectors ............................... 72
- Inverse of the alpha-transformation .......................................................................... 74
- James multivariate version of the t-test ...................................................................... 75
- Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distri-
  butions ......................................................................................................................... 77
- Log-likelihood ratio test for a Dirichlet mean vector ................................................... 78
- Log-likelihood ratio test for a symmetric Dirichlet distribution ................................... 79
- Mixture model selection via BIC .................................................................................. 80
- MLE for the multivariate t distribution ...................................................................... 82
- MLE of distributions defined in the (0, 1) interval ....................................................... 83
- MLE of the folded model for a given value of alpha .................................................... 84
- Multivariate analysis of variance .............................................................................. 86
- Multivariate analysis of variance (James test) ............................................................. 87
- Multivariate kernel density estimation ........................................................................ 88
- Multivariate linear regression ...................................................................................... 89
- Multivariate normal random values simulation on the simplex .................................. 90
- Multivariate or univariate regression with compositional data in the covariates side using
  the alpha-transformation ......................................................................................... 91
- Multivariate regression with compositional data ....................................................... 93
- Multivariate skew normal random values simulation on the simplex ....................... 94
- Multivariate t random values simulation on the simplex .......................................... 95
- Non linear least squares regression for compositional data ........................................ 97
- Perturbation operation .............................................................................................. 98
- Power operation ........................................................................................................ 99
- Principal component generalised linear models ....................................................... 100
Projection pursuit regression for compositional data ........................................... 102
Quasi binomial regression for proportions .......................................................... 103
Regression with compositional data using the alpha-transformation .................. 105
Regularised and flexible discriminant analysis for compositional data using the alpha-
transformation ........................................................................................................ 106
Regularised discriminant analysis for Euclidean data ........................................... 108
Ridge regression ....................................................................................................... 110
Ridge regression plot ............................................................................................... 111
Ridge regression with compositional data in the covariates side using the alpha-transformation ................................................................. 112
Ridge regression with the alpha-transformation plot ............................................. 113
Simulation of compositional data from Gaussian mixture models ......................... 115
Simulation of compositional data from the Flexible Dirichlet distribution ............. 116
Simulation of compositional data from the folded normal distribution .................. 117
Spatial median regression ......................................................................................... 118
Ternary diagram ....................................................................................................... 120
Ternary diagram of regression models .................................................................... 121
The additive log-ratio transformation and its inverse ............................................ 122
The alpha-distance ................................................................................................. 123
The alpha-k-NN regression for compositional response data ............................... 124
The alpha-k-NN regression with compositional predictor variables ...................... 125
The alpha-transformation ......................................................................................... 127
The ESOV-distance ................................................................................................. 128
The Frechet mean for compositional data ............................................................... 129
The Helmert sub-matrix ......................................................................................... 131
The k-nearest neighbours using the alpha-distance .............................................. 132
The k-NN algorithm for compositional data .......................................................... 133
Total variability ....................................................................................................... 135
Transformation-free linear regression for compositional responses and predictors ................................................................. 136
Tuning of the alpha generalised correlations between two compositional datasets .... 137
Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation ................................................................. 138
Tuning of the divergence based regression for compositional data with compositional
data in the covariates side using the alpha-transformation ..................................... 139
Tuning of the k-NN algorithm for compositional data ............................................ 141
Tuning of the projection pursuit regression for compositional data ....................... 143
Tuning the number of PCs in the PCR with compositional data using the alpha-transformation ................................................................. 145
Tuning the parameters of the regularised discriminant analysis ............................ 146
Tuning the principal components with GLMs ....................................................... 148
Tuning the value of alpha in the alpha-regression ............................................... 150
Unit-Weibull regression models for proportions ................................................... 152
Zero adjusted Dirichlet regression ........................................................................... 153
Compositional-package

Compositional-package  Compositional Data Analysis

Description

A collection of functions for compositional data analysis.

Details

Package: Compositional
Type: Package
Version: 4.3
Date: 2021-01-19
License: GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>

Note

Acknowledgments:
Michail Tsagris would like to express his acknowledgments to Professor Andy Wood and Dr Simon Preston from the university of Nottingham for being his supervisors during his PhD in compositional data analysis. We would also like to express our acknowledgments to Profesor Kurt Hornik (and also the rest of the R core team) for his help with this package. Manos Papadakis, undergraduate student in the department of computer science, university of Crete, is also acknowledged for his programming tips. Ermanno Affuso from the university of South Alabama suggested that I have a default value in the "mkde" function. Van Thang Hoang from Hasselt university spotted a bug in the "js.compreg" function and is greatly acknowledged for that. Claudia Wehrhahn Cortes spotted a bug in the "diri.reg" function and she is greatly acknowledged for that. Philipp Kynast from Bruker Daltonik GmbH found a mistake in the function "mkde" which is now fixed. Jasmine Heyse from the university of Ghent spotted a bug in the function "kl.compreg" which is now fixed.

Author(s)

Michail Tsagris <mtsagris@uoc.gr>, Giorgos Athineou <gioathineou@gmail.com> and Abdulaziz Alenazi <a.alenazi@nbu.edu.sa>.

References

Aithison's simple zero replacement strategy

Description

Aithison’s simple zero replacement strategy.

Usage

zeroreplace(x, a = 2/3)

Arguments

x A matrix with the compositional data.

a The replacement value will be "a" times the minimum value observed in the compositional data.

Details

This is the simple zero replacement strategy suggested in Aitchison (1986, pg. 269).

Value

A matrix with the zero replaced compositional data.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

perturbation, alfa

Examples

x <- as.matrix(iris[1:20, 1:4])
x <- x / rowSums(x)
x[sample(1:20, 4), sample(1:4, 1)] <- 0
zeroreplace(x)
All pairwise additive log-ratio transformations

Description

All pairwise additive log-ratio transformations.

Usage

alr.all(x)

Arguments

x A numerical matrix with the compositional data.

Details

The additive log-ratio transformation with the first component being the common divisor is applied. Then all the other pairwise log-ratios are computed and added next to each column. For example, divide by the first component, then divide by the second component and so on.

Value

A matrix with all pairwise alr transformed data.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

alr, link{alfa}

Examples

x <- as.matrix(iris[, 2:4])
x <- x / rowSums(x)
y <- alr.all(x)
Alpha generalised correlations between two compositional datasets

Description
Alpha generalised correlations between two compositional datasets.

Usage

acor(y, x, a, type = "dcor")

Arguments

y
A matrix with the compositional data.

x
A matrix with the compositional data.

a
The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied. If more than one values are supplied the distance or canonical correlation are computed for all values.

type
the type of correlation to compute, the distance correlation ("edist"), the canonical correlation ("cancor") or "both".

Details
The $\alpha$-transformation is applied to each composition and then the distance correlation or the canonical correlation is computed. If one value of $\alpha$ is supplied the type="cancor" will return all eigenvalues. If more than one values of $\alpha$ are provided then the first eigenvalue only will be returned.

Value
A vector or a matrix depending on the length of the values of $\alpha$ and the type of the correlation to be computed.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

acor.tune, alfa.profile, alfa, alfainv

Examples

y <- rdiri(30, runif(3) )
x <- rdiri(30, runif(4) )
acor(y, x, a = 0.4)
Beta regression

Description
Beta regression.

Usage
beta.reg(y, x, xnew = NULL)

Arguments
- **y**: The response variable. It must be a numerical vector with proportions excluding 0 and 1.
- **x**: The independent variable(s). It can be a vector, a matrix or a dataframe with continuous only variables, a data frame with mixed or only categorical variables.
- **xnew**: If you have new values for the predictor variables (dataset) whose response values you want to predict insert them here.

Details
Beta regression is fitted.

Value
A list including:
- **phi**: The estimated precision parameter.
- **info**: A matrix with the estimated regression parameters, their standard errors, Wald statistics and associated p-values.
- **loglik**: The log-likelihood of the regression model.
- **est**: The estimated values if xnew is not NULL.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References

See Also
beta.est, prop.reg, diri.reg
Choose the number of principal components via reconstruction error

Examples

```r
y <- rbeta(300, 3, 5)
x <- matrix( rnorm(300 * 2), ncol = 2)
beta.reg(y, x)
```

Description

Choose the number of principal components via reconstruction error.

Usage

```r
choose.pc(x, plot = TRUE)
```

Arguments

- `x` - A numerical matrix with more rows than columns.
- `plot` - Should the plot of the PRESS values appear? Default value is TRUE.

Details

The function allows for selecting the number of eigenvectors via the reconstruction error which is computed for all eigenvectors based on SVD.

Value

A list including:

- `values` - The eigenvalues of the covariance matrix.
- `cumprop` - The cumulative proportion of the eigenvalues of the covariance matrix.
- `per` - The differences in the cumulative proportion of the eigenvalues of the covariance matrix.
- `press` - The reconstruction error $\sqrt{\sum_{ij} (x_{ij} - \hat{x}_{ij})^2}$ for each number of eigenvectors.
- `runtime` - The runtime of the algorithm.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also

`pcr, alfa.pcr, alfapcr.tune`

Examples

```r
library(MASS)
x <- as.matrix(fgl[, 2:9])
a <- choose.pc(x, plot = FALSE)
```

Description

Constrained linear least squares for compositional responses and predictors.

Usage

```r
ols.compcomp(y, x, rs = 5, tol = 1e-4, xnew = NULL)
```

Arguments

- `y` A matrix with the compositional data (dependent variable). Zero values are allowed.
- `x` A matrix with the compositional predictors. Zero values are allowed.
- `rs` The number of times to run the constrained optimisation using different random starting values each time.
- `tol` The threshold upon which to stop the iterations of the constrained optimisation.
- `xnew` If you have new data use it, otherwise leave it NULL.

Details

The function performs least squares regression where the beta coefficients are constrained to be positive and sum to 1. We were inspired by the transformation-free linear regression for compositional responses and predictors of Fiksel, Zeger and Datta (2020).

Value

A list including:

- `runtime` The time required by the regression.
- `mse` The mean squared errors.
- `be` The beta coefficients.
- `est` The fitted of xnew if xnew is not NULL.
Contour plot of a Dirichlet distribution in $S^2$

Author(s)
Michail Tsagris.
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
cv.olscompcomp, tflr, kl.alfapcr

Examples

```r
library(MASS)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- ols.compcomp(y, x)
mod
```

Description
Contour plot of a Dirichlet distribution in $S^2$.

Usage
`diri.contour(a, n = 100, x = NULL)`

Arguments
- `a`: A vector with three elements corresponding to the 3 (estimated) parameters.
- `n`: The number of grid points to consider over which the density is calculated.
- `x`: This is either NULL (no data) or contains a 3 column matrix with compositional data.

Details
The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should he/she wish to.
Contour plot of a Flexible Dirichlet distribution in $S^2$

Value

A ternary diagram with the points and the Dirichlet contour lines.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

norm.contour, bivt.contour, comp.kerncontour, mixnorm.contour

Examples

x <- as.matrix( iris[, 1:3] )
x <- x / rowSums(x)
diri.contour( a = c(3, 4, 2) )

Description

Contour plot of a Flexible Dirichlet distribution in $S^2$.

Usage

fd.contour(alpha, prob, tau, n = 100, x = NULL)

Arguments

alpha A vector of the non-negative alpha parameters.
prob A vector of the clusters’ probabilities. It must sum to one.
tau The non-negative scalar tau parameter.
n The number of grid points to consider over which the density is calculated.
x This is either NULL (no data) or contains a 3 column matrix with compositional data.
Details

The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should they wish to.

Value

A ternary diagram with the points and the Flexible Dirichlet contour lines.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

norm.contour, bivt.contour, comp.kerncontour, mixnorm.contour

Examples

```r
## Not run:
x <- rfd(n = 50, a = c(12, 11, 10), p = c(0.25, 0.25, 0.5), tau = 4)
mod <- fd.est(x, ini.iter = 10, final.iter = 20)
fd.contour(mod$alpha, mod$prob, mod$tau, x = x)
## End(Not run)
```

Description

Contour plot of a Gaussian mixture model in $S^2$.

Usage

mixnorm.contour(x, mod)
Contour plot of a Gaussian mixture model in $S^2$

Arguments

- **x**: A matrix with the compositional data.
- **mod**: An object containing the output of a `mix.compnorm` model.

Details

The contour plot of a Gaussian mixture model is plotted. For this you need the data and the fitted model.

Value

A ternary plot with the data and the contour lines of the fitted Gaussian mixture model.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

`mix.compnorm, bic.mixcompnorm, diri.contour`

Examples

```r
## Not run:
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
mod <- mix.compnorm(x, 3, model = "EII")
mixnorm.contour(x, mod)
## End(Not run)
```
Contour plot of a Kent distribution in $S^2$

Description
Contour plot of a Kent distribution in $S^2$.

Usage
kent.contour(G, param, n = 100, x = NULL)

Arguments
- **G**: A 3 x 3 matrix whose first column is the mean direction. The second and third columns are the major and minor axes respectively.
- **param**: A vector with the concentration $kappa$ and ovalness $beta$ parameters (the $psi$ parameter has been absorbed inside the matrix G).
- **n**: The number of grid points to consider over which the density is calculated.
- **x**: This is either NULL (no data) or contains a 3 column matrix with compositional data.

Details
The user can plot only the contour lines of a Dirichlet with a given vector of parameters, or can also add the relevant data should they wish to.

Value
A ternary diagram with the points and the Dirichlet contour lines.

Author(s)
Michail Tsagris.
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
norm.contour, bivt.contour, comp.kerncontour, mixnorm.contour
Examples

```
G <- matrix( c(0.05713382, 0.96029716, 0.27306608, -0.98809661, 
                0.01525976, 0.15307588, 0.1428314, -0.2785615, 0.9497382), 
              ncol = 3 )
param <- c(2361.8401338, 1171.3808172, 0.1435577)
kent.contour(G, param)
```

Description

Contour plot of the folded model in $S^2$.

Usage

```
fold.contour(m, s, p, a, n = 100, x = NULL)
```

Arguments

- `m`: The mean vector of the folded model.
- `s`: The covariance matrix of the folded model.
- `p`: The probability inside the simplex of the folded model.
- `a`: The value of $a$ for the $\alpha$-transformation.
- `n`: The number of grid points to consider over which the density is calculated.
- `x`: This is either NULL (no data) or contains a 3 column matrix with compositional data.

Details

The $\alpha$-transformation is applied to the compositional data and then for a grid of points within the 2-dimensional simplex the folded model’s density is calculated and the contours are plotted.

Value

The contour plot of the folded model appears.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also

diri.contour, mixnorm.contour, bivt.contour, skewnorm.contour

Examples

x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
a <- a.est(x)$best
mod <- alpha.mle(x, a)
fold.contour(mod$mu, mod$su, mod$p, a)

Description

Contour plot of the kernel density estimate in $S^2$.

Usage

comp.kerncontour(x, type = "alr", n = 100)

Arguments

x A matrix with the compositional data. It has to be a 3 column matrix.
type This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
n The number of grid points to consider, over which the density is calculated.

Details

The alr or the ilr transformation are applied to the compositional data. Then, the optimal bandwidth using maximum likelihood cross-validation is chosen. The multivariate normal kernel density is calculated for a grid of points. Those points are the points on the 2-dimensional simplex. Finally the contours are plotted.

Value

A ternary diagram with the points and the kernel contour lines.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>
Contour plot of the normal distribution in $S^2$

References


See Also

diri.contour, mixnorm.contour, bivt.contour, norm.contour

Examples

```r
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
comp.kerncontour(x, type = "alr", n = 20)
comp.kerncontour(x, type = "ilr", n = 20)
```

Description

Contour plot of the normal distribution in $S^2$.

Usage

```r
norm.contour(m, s, type = "alr", n = 100, x = NULL)
```

Arguments

- `m` The mean vector.
- `s` The covariance matrix.
- `type` This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
- `n` The number of grid points to consider over which the density is calculated.
- `x` This is either NULL (no data) or contains a 3 column matrix with compositional data.

Details

The alr or the ilr transformation is applied to the compositional data at first. Then for a grid of points within the 2-dimensional simplex the bivariate normal density is calculated and the contours are plotted along with the points.

Value

A ternary diagram with the points (if appear = TRUE) and the bivariate normal contour lines.
Description

Contour plot of the skew skew-normal distribution in $S^2$.

Usage

skewnorm.contour(x, type = "alr", n = 100, appear = TRUE)

Arguments

x A matrix with the compositional data. It has to be a 3 column matrix.
type This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
n The number of grid points to consider over which the density is calculated.
appear Should the available data appear on the ternary plot (TRUE) or not (FALSE)?

Details

The alr or the ilr transformation is applied to the compositional data at first. Then for a grid of points within the 2-dimensional simplex the bivariate skew skew-normal density is calculated and the contours are plotted along with the points.
Contour plot of the t distribution in $S^2$

Value

A ternary diagram with the points (if appear = TRUE) and the bivariate skew skew-normal contour lines.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

diri.contour,mixnorm.contour,bivt.contour,norm.contour

Examples

```r
x <- as.matrix(iris[51:100, 1:3])
x <- x / rowSums(x)
skewnorm.contour(x)
```

Description

Contour plot of the t distribution in $S^2$.

Usage

```r
bivt.contour(x, type = "alr", n = 100, appear = TRUE)
```

Arguments

- **x**: A matrix with compositional data. It has to be a 3 column matrix.
- **type**: This is either "alr" or "ilr", corresponding to the additive and the isometric log-ratio transformation respectively.
- **n**: The number of grid points to consider over which the density is calculated.
- **appear**: Should the available data appear on the ternary plot (TRUE) or not (FALSE)?
Cross validation for some compositional regression models

Details
The alr or the ilr transformation is applied to the compositional data at first and the location, scatter and degrees of freedom of the bivariate t distribution are computed. Then for a grid of points within the 2-dimensional simplex the bivariate t density is calculated and the contours are plotted along with the points.

Value
A ternary diagram with the points (if appear = TRUE) and the bivariate t contour lines.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
diri.contour, mixnorm.contour, norm.contour, skewnorm.contour

Examples
```r
x <- as.matrix( iris[, 1:3] )
x <- x / rowSums(x)
bivt.contour(x)
bivt.contour(x, type = "ilr")
```

Cross validation for some compositional regression models

Description
Cross validation for some compositional regression models.

Usage
```r
cv.comp.reg(y, x, type = "comp.reg", nfolds = 10, folds = NULL, seed = FALSE)
```
Cross validation for some compositional regression models

Arguments

- **y**: A matrix with compositional data. Zero values are allowed for some regression models.
- **x**: The predictor variable(s).
- **type**: This can be one of the following: "comp.reg", "robust", "kl.compreg", "js.compreg", "diri.reg" or "zadr".
- **nfolds**: The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
- **folds**: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- **seed**: If seed is TRUE the results will always be the same.

Details

A k-fold cross validation for a compositional regression model is performed.

Value

A list including:

- **runtime**: The runtime of the cross-validation procedure.
- **kl**: The Kullback-Leibler divergences for all runs.
- **js**: The Jensen-Shannon divergences for all runs.
- **perf**: The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

- comp.reg, kl.compreg, compppr.tune, aknnreg.tune

Examples

```r
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- cv.comp.reg(y, x)
```
Cross validation for the alpha-k-NN regression for compositional response data

Description

Cross validation for the $\alpha$-k-NN regression for compositional response data.

Usage

`aknnreg.tune(y, x, a = seq(0.1, 1, by = 0.1), k = 2:10, apostasi = "euclidean", nfolds = 10, folds = NULL, seed = FALSE, B = 1, rann = FALSE)`

Arguments

- `y`: A matrix with the compositional response data. Zeros are allowed.
- `x`: A matrix with the available predictor variables.
- `a`: A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
- `k`: The number of nearest neighbours to consider. It can be a single number or a vector.
- `apostasi`: The type of distance to use, either "euclidean" or "manhattan".
- `nfolds`: The number of folds. Set to 10 by default.
- `folds`: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `seed`: If seed is TRUE the results will always be the same.
- `B`: If you want to correct for the optimistic bias set this to more than 1, otherwise no bootstrap bias correction takes place. If you have large sample sizes, say 1000 or more, bootstrap bias correction may not be really necessary.
- `rann`: If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

A k-fold cross validation for the $\alpha$-k-NN regression for compositional response data is performed.
Cross validation for the alpha-k-NN regression for compositional response data

Value

A list including:

- \( \text{k} \) The Kullback-Leibler divergence for all combinations of \( \alpha \) and \( k \).
- \( \text{js} \) The Jensen-Shannon divergence for all combinations of \( \alpha \) and \( k \).
- \( \text{klmin} \) The minimum Kullback-Leibler divergence.
- \( \text{jsmin} \) The minimum Jensen-Shannon divergence.
- \( \text{bc.kl} \) The bootstrap bias corrected minimum Kullback-Leibler divergence.
- \( \text{bs.js} \) The bootstrap bias corrected minimum Jensen-Shannon divergence.
- \( \text{kl.alpha} \) The optimim \( \alpha \) that leads to the minimum Kullback-Leibler divergence.
- \( \text{kl.k} \) The optimim \( k \) that leads to the minimum Kullback-Leibler divergence.
- \( \text{js.alpha} \) The optimim \( \alpha \) that leads to the minimum Jensen-Shannon divergence.
- \( \text{js.k} \) The optimim \( k \) that leads to the minimum Jensen-Shannon divergence.
- \( \text{runtime} \) The runtime of the cross-validation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

\texttt{alfa.rda, alfa.fda, rda.tune}

Examples

```r
y <- as.matrix( iris[, 1:3] )
y <- y / rowSums(y)
x <- iris[, 4]
mod <- aknnreg.tune(y, x, a = c(0.4, 0.6), k = 2:4, nfolds = 5)
```
Cross validation for the alpha-k-NN regression with compositional predictor variables

Description

Cross validation for the $\alpha$-k-NN regression with compositional predictor variables.

Usage

```r
alfaknnreg.tune(y, x, a = seq(-1, 1, by = 0.1), k = 2:10, nfolds = 10,
apostasi = "euclidean", method = "average", folds = NULL, seed = FALSE, graph = FALSE)
```

Arguments

- `y`: The response variable, a numerical vector.
- `x`: A matrix with the available compositional data. Zeros are allowed.
- `a`: A vector with a grid of values of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
- `k`: The number of nearest neighbours to consider. It can be a single number or a vector.
- `nfolds`: The number of folds. Set to 10 by default.
- `apostasi`: The type of distance to use, either "euclidean" or "manhattan".
- `method`: If you want to take the average of the responses of the k closest observations, type "average". For the median, type "median" and for the harmonic mean, type "harmonic".
- `folds`: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `seed`: If seed is TRUE the results will always be the same.
- `graph`: If graph is TRUE (default value) a filled contour plot will appear.

Details

A k-fold cross validation for the $\alpha$-k-NN regression for compositional response data is performed.

Value

A list including:

- `mspe`: The mean square error of prediction.
- `performance`: The minimum mean square error of prediction.
- `opt_a`: The optimal value of $\alpha$.
- `opt_k`: The optimal value of k.
- `runtime`: The runtime of the cross-validation procedure.
Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
alfa.rda, alfa.fda, rda.tune

Examples
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- fgl[, 1]
mod <- alfaknnreg.tune(y, x, a = seq(0.2, 0.4, by = 0.1), k = 2:4, nfolds = 5)

Description
Cross validation for the regularised and flexible discriminant analysis with compositional data using the α-transformation.

Usage
alfarda.tune(x, ina, a = seq(-1, 1, by = 0.1), nfolds = 10,  
gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),  
ncores = 1, folds = NULL, stratified = TRUE, seed = FALSE)

alfafda.tune(x, ina, a = seq(-1, 1, by = 0.1), nfolds = 10,  
folds = NULL, stratified = TRUE, seed = FALSE, graph = FALSE)

Arguments
x A matrix with the available compositional data. Zeros are allowed.
ina A group indicator variable for the available data.
a A vector with a grid of values of the power transformation, it has to be between 
-1 and 1. If zero values are present it has to be greater than 0. If α = 0 the
isometric log-ratio transformation is applied.
Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation

nfolds The number of folds. Set to 10 by default.
gam A vector of values between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.
del A vector of values between 0 and 1. It is the weight of the LDA and QDA.
ncores The number of cores to use. If it is more than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.
folds If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
stratified Do you want the folds to be created in a stratified way? TRUE or FALSE.
seed If seed is TRUE the results will always be the same.
graph If graph is TRUE (default value) a filled contour plot will appear.

Details
A k-fold cross validation is performed.

Value
For the alfa.rda a list including:

res The estimated optimal rate and the best values of \( \alpha \), \( \gamma \) and \( \delta \).
percent For the best value of \( \alpha \) the averaged over all folds best rates of correct classification. It is a matrix, where rows correspond to the \( \gamma \) values and columns correspond to \( \delta \) values.
se The estimated standard errors of the "percent" matrix.
runtime The runtime of the cross-validation procedure.

For the alfa.fda a list including:

per The performance of the fda in each fold for each value of \( \alpha \).
performance The average performance for each value of \( \alpha \).
opt_a The optimal value of \( \alpha \).
runtime The runtime of the cross-validation procedure.

Author(s)
Michail Tsagris

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>
References


See Also

alfa.rda, alfa.fda, rda.tune

Examples

library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
ina <- fgl[, 10]
moda <- alfa.fda.tune(x, ina, a = seq(0.7, 1, by = 0.1), nfolds = 10, 
gam = seq(0.1, 0.3, by = 0.1), del = seq(0.1, 0.3, by = 0.1))

Cross validation for the ridge regression

Description

Cross validation for the ridge regression is performed. There is an option for the GCV criterion which is automatic.

Usage

ridge.tune(y, x, nfolds = 10, lambda = seq(0, 2, by = 0.1), folds = NULL, ncores = 1, seed = FALSE, graph = FALSE)

Arguments

y A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation.
x A numeric matrix containing the variables.
nfolds The number of folds in the cross validation.
lambda A vector with the a grid of values of λ to be used.
folds If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
Cross validation for the ridge regression

ncores
The number of cores to use. If it is more than 1 parallel computing is performed.

seed
If seed is TRUE the results will always be the same.

graph
If graph is set to TRUE the performances for each fold as a function of the $\lambda$ values will appear.

Details
A k-fold cross validation is performed. This function is used by alfardige.tune.

Value
A list including:

msp
The performance of the ridge regression for every fold.

mspe
The values of the mean prediction error for each value of $\lambda$.

lambda
The value of $\lambda$ which corresponds to the minimum MSPE.

performance
The minimum MSPE.

runtime
The time required by the cross-validation procedure.

Author(s)
Michail Tsagris

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>

References


See Also

ridge.reg, alfardige.tune

Examples

y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
ridge.tune( y, x, nfolds = 10, lambda = seq(0, 2, by = 0.1), graph = TRUE )
Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation

Description

Cross validation for the ridge regression is performed. There is an option for the GCV criterion which is automatic. The predictor variables are compositional data and the $\alpha$-transformation is applied first.

Usage

```r
alfaridge.tune(y, x, nfolds = 10, a = seq(-1, 1, by = 0.1), lambda = seq(0, 2, by = 0.1), folds = NULL, ncores = 1, graph = TRUE, col.nu = 15, seed = FALSE)
```

Arguments

- `y` A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation.
- `x` A numeric matrix containing the compositional data, i.e. the predictor variables. Zero values are allowed.
- `nfolds` The number of folds in the cross validation.
- `a` A vector with the a grid of values of $\alpha$ to be used.
- `lambda` A vector with the a grid of values of $\lambda$ to be used.
- `folds` If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `ncores` The number of cores to use. If it is more than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.
- `graph` If graph is TRUE (default value) a filled contour plot will appear.
- `col.nu` A number parameter for the filled contour plot, taken into account only if graph is TRUE.
- `seed` If seed is TRUE the results will always be the same.

Details

A k-fold cross validation is performed.
Cross validation for the transformation-free linear regression for compositional responses and predictors

**Value**

If graph is TRUE a filed contour a filled contour will appear. A list including:

- `mspe`: The MSPE where rows correspond to the $\alpha$ values and the columns to the number of principal components.
- `best.par`: The best pair of $\alpha$ and $\lambda$.
- `performance`: The minimum mean squared error of prediction.
- `runtime`: The run time of the cross-validation procedure.

**Author(s)**

Michail Tsagris

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>

**References**


**See Also**

`alfa.ridge, ridge.tune`

**Examples**

```r
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
alfaridge.tune( y, x, nfolds = 10, a = seq(0.1, 1, by = 0.1),
               lambda = seq(0, 1, by = 0.1) )
```

---

Cross validation for the transformation-free linear regression for compositional responses and predictors

**Description**

Cross validation for the transformation-free linear regression for compositional responses and predictors.
Cross validation for the transformation-free linear regression for compositional responses and predictors

Usage

```
cv.tflr(y, x, nfolds = 10, folds = NULL, seed = FALSE)
```

Arguments

- **y**: A matrix with compositional response data. Zero values are allowed.
- **x**: A matrix with compositional predictors. Zero values are allowed.
- **nfolds**: The number of folds to be used. This is taken into consideration only if the `folds` argument is not supplied.
- **folds**: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- **seed**: If seed is TRUE the results will always be the same.

Details

A k-fold cross validation for the transformation-free linear regression for compositional responses and predictors is performed.

Value

A list including:

- **runtime**: The runtime of the cross-validation procedure.
- **kl**: The Kullback-Leibler divergences for all runs.
- **js**: The Jensen-Shannon divergences for all runs.
- **perf**: The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

```
tflr, cv.olscompcomp, klalfapcr.tune
```

Examples

```
library(MASS)
y <- rdiri(214, runif(3, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- cv.tflr(y, x)
mod
```
Cross-validation for the constrained linear least squares for compositional responses and predictors

**Description**

Cross-validation for the constrained linear least squares for compositional responses and predictors.

**Usage**

```r
cv.olscompcomp(y, x, rs = 5, tol = 1e-4, nfolds = 10, folds = NULL, seed = FALSE)
```

**Arguments**

- `y`: A matrix with compositional response data. Zero values are allowed.
- `x`: A matrix with compositional predictors. Zero values are allowed.
- `rs`: The number of times to run the constrained optimisation using different random starting values each time.
- `tol`: The threshold upon which to stop the iterations of the constrained optimisation.
- `nfolds`: The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
- `folds`: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `seed`: If seed is TRUE the results will always be the same.

**Details**

The function performs k-fold cross-validation for the least squares regression where the beta coefficients are constrained to be positive and sum to 1.

**Value**

A list including:

- `runtime`: The runtime of the cross-validation procedure.
- `kl`: The Kullback-Leibler divergences for all runs.
- `js`: The Jensen-Shannon divergences for all runs.
- `perf`: The average Kullback-Leibler divergence and average Jensen-Shannon divergence.

**Author(s)**

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>. 
Density of the Dirichlet distribution

See Also

ols.compcomp, cv.tflr, klalfapcr.tune

Examples

library(MASS)
y <- rdiri(214, runif(3, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- cv.tflr(y, x)
mod

Density of the Dirichlet distribution

Description

Density of the Dirichlet distribution.

Usage

diri.density(x, a, logged = FALSE)

Arguments

x A vector or a matrix with compositional data.
a A vector of the non-negative alpha parameters.
logged Do you want the logarithm of the density values? TRUE or FALSE.

Value

The density value(s).

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

fd.density, rdiri, diri.nr
Density of the Flexible Dirichlet distribution

Examples

```r
a <- runif(3, 1, 5)
x <- rdiri(100, a)
a <- diri.nr(x)$param
x <- diri.density(x, a)
```

Description

Density of the Flexible Dirichlet distribution

Usage

```r
fd.density(x, alpha, prob, tau)
```

Arguments

- `x`: A vector or a matrix with compositional data.
- `alpha`: A vector of the non-negative \( \alpha \) parameters.
- `prob`: A vector of the clusters’ probabilities. It must sum to one.
- `tau`: The non-negative scalar \( \tau \) parameter.

Details

For more information see the references.

Value

The density value(s).

Author(s)

Michail Tsagris ported from the R package FlexDir. <mtsagris@uoc.gr>.

References


See Also

`fd.est`, `rfd`
Density values of a Dirichlet distribution

Examples

```r
alpha <- c(12, 11, 10)
prob <- c(0.25, 0.25, 0.5)
tau <- 8
x <- rfd(20, alpha, prob, tau)
fd.density(x, alpha, prob, tau)
```

Description

Density values of a Dirichlet distribution.

Usage

`ddiri(x, a, logged = TRUE)`

Arguments

- `x`: A matrix containing compositional data. This can be a vector or a matrix with the data.
- `a`: A vector of parameters. Its length must be equal to the number of components, or columns of the matrix with the compositional data and all values must be greater than zero.
- `logged`: A boolean variable specifying whether the logarithm of the density values to be returned. It is set to TRUE by default.

Details

The density of the Dirichlet distribution for a vector or a matrix of compositional data is returned.

Value

A vector with the density values.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Giorgos Athineou `<gioathineou@gmail.com>`

References

See Also

diri.nr, diri.est, diri.contour, rdiri

Examples

```r
x <- rdiri(100, c(5, 7, 4, 8, 10, 6, 4))
a <- diri.est(x)
f <- ddiri(x, a$param)
sum(f)
a
```

Description

Dirichlet random values simulation.

Usage

```r
rdiri(n, a)
```

Arguments

- `n` The sample size, a numerical value.
- `a` A numerical vector with the parameter values.

Details

The algorithm is straightforward, for each vector, independent gamma values are generated and then divided by their total sum.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

**Dirichlet regression**

See Also
----------------------

`diri.est`, `diri.nr`, `diri.contour`

Examples
----------------------

```r
x <- rdiri(100, c(5, 7, 1, 3, 10, 2, 4))
diri.est(x)
```

**Description**
----------------------

Dirichlet regression.

**Usage**
----------------------

```r
diri.reg(y, x, plot = TRUE, xnew = NULL)
diri.reg2(y, x, xnew = NULL)
```

**Arguments**
----------------------

- **y**: A matrix with the compositional data (dependent variable). Zero values are not allowed.
- **x**: The predictor variable(s), they can be either continuous or categorical or both.
- **plot**: A boolean variable specifying whether to plot the leverage values of the observations or not. This is taken into account only when `xnew = NULL`.
- **xnew**: If you have new data use it, otherwise leave it `NULL`.

**Details**
----------------------

A Dirichlet distribution is assumed for the regression. This involves numerical optimization. The function "diri.reg2" allows for the covariates to be linked with the precision parameter $\phi$ via the exponential link function $\phi = e^{xb}$.

**Value**
----------------------

A list including:

- **runtime**: The time required by the regression.
- **loglik**: The value of the log-likelihood.
- **phi**: The precision parameter. If covariates are linked with it (function "diri.reg2"), this will be a vector.
- **phipar**: The coefficients of the phi parameter if it is linked to the covariates.
Dirichlet regression

std.phi  The standard errors of the coefficients of the phi parameter is it linked to the covariates.
log.phi  The logarithm of the precision parameter.
std.logphi  The standard error of the logarithm of the precision parameter.
be  The beta coefficients.
seb  The standard error of the beta coefficients.
sigma  The covariance matrix of the regression parameters (for the mean vector and the phi parameter) in the function "diri.reg2".
lev  The leverage values.
est  For the "diri.reg" this contains the fitted or the predicted values (if xnew is not NULL). For the "diri.reg2" if xnew is NULL, this is also NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

js.compreg, kl.compreg, ols.compreg, comp.reg, alfa.reg

Examples

x <- as.vector(iris[, 4])
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
mod1 <- diri.reg(y, x)
mod2 <- diri.reg2(y, x)
mod3 <- comp.reg(y, x)
Distance based regression models for proportions

Description

Distance based regression models for proportions.

Usage

```r
ols.prop.reg(y, x, cov = FALSE, tol = 1e-07, maxiters = 100)
helling.prop.reg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

- `y`: A numerical vector proportions. 0s and 1s are allowed.
- `x`: A matrix or a data frame with the predictor variables.
- `cov`: Should the covariance matrix be returned? TRUE or FALSE.
- `tol`: The tolerance value to terminate the Newton-Raphson algorithm. This is set to $10^{-9}$ by default.
- `maxiters`: The maximum number of iterations before the Newton-Raphson is terminated automatically.

Details

We are using the Newton-Raphson, but unlike R’s built-in function "glm" we do no checks and no extra calculations, or whatever. Simply the model. The functions accept binary responses as well (0 or 1).

Value

A list including:

- `sse`: The sum of squares of errors for the "ols.prop.reg" function.
- `be`: The estimated regression coefficients.
- `seb`: The standard error of the regression coefficients if "cov" is TRUE.
- `covb`: The covariance matrix of the regression coefficients in "ols.prop.reg" if "cov" is TRUE.
- `H`: The Hellinger distance between the true and the observed proportions in "helling.prop.reg".
- `iters`: The number of iterations required by the Newton-Raphson.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.
Divergence based regression for compositional data

References

See Also
propreg, beta.reg

Examples
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(100 * 2), ncol = 2)
a1 <- ols.prop.reg(y, x)
a2 <- helling.prop.reg(y, x)

Divergence based regression for compositional data

Description
Regression for compositional data based on the Kullback-Leibler the Jensen-Shannon divergence and the symmetric Kullback-Leibler divergence.

Usage
kl.compreg(y, x, B = 1, ncores = 1, xnew = NULL, tol = 1e-07, maxiters = 50)
js.compreg(y, x, B = 1, ncores = 1, xnew = NULL)
symkl.compreg(y, x, B = 1, ncores = 1, xnew = NULL)

Arguments
y A matrix with the compositional data (dependent variable). Zero values are allowed.
x The predictor variable(s), they can be either continuous or categorical or both.
B If B is greater than 1 bootstrap estimates of the standard error are returned. If B=1, no standard errors are returned.
ncores If ncores is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If B=1, this is not taken into consideration.
xnew If you have new data use it, otherwise leave it NULL.
tol The tolerance value to terminate the Newton-Raphson procedure.
maxiters The maximum number of Newton-Raphson iterations.
Details

In the kl.compreg the Kullback-Leibler divergence is adopted as the objective function. In case of problematic convergence the "multinom" function by the "nnet" package is employed. This will obviously be slower. The js.compreg uses the Jensen-Shannon divergence and the symkl.compreg uses the symmetric Kullback-Leibler divergence. There is no actual log-likelihood for neither regression.

Value

A list including:

- runtime The time required by the regression.
- iters The number of iterations required by the Newton-Raphson in the kl.compreg function.
- loglik The log-likelihood. This is actually a quasi multinomial regression. This is basically minus the half deviance, or \(- \sum_{i=1}^n y_i \log \hat{y}_i / \hat{y}_i\).
- be The beta coefficients.
- cov The covariance matrix of the beta coefficients, if bootstrap is chosen, i.e. if B > 1.
- est The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

diri.reg, js.compreg, ols.compreg, comp.reg
Divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation

Examples

library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1 <- kl.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)

Description

Divergence based regression for compositional data with compositional data in the covariates side using the $\alpha$-transformation.

Usage

kl.alfapcr(y, x, covar = NULL, a, k, xnew = NULL, B = 1, ncores = 1, tol = 1e-07, maxiters = 50)

Arguments

- **y**: A numerical matrix with compositional data with or without zeros.
- **x**: A matrix with the predictor variables, the compositional data. Zero values are allowed.
- **covar**: If you have other covariates as well put them here.
- **a**: The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
- **k**: A number at least equal to 1. How many principal components to use.
- **xnew**: A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.
- **B**: If B is greater than 1 bootstrap estimates of the standard error are returned. If B=1, no standard errors are returned.
- **ncores**: If ncores is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If B=1, this is not taken into consideration.
- **tol**: The tolerance value to terminate the Newton-Raphson procedure.
- **maxiters**: The maximum number of Newton-Raphson iterations.
**Details**

The $\alpha$-transformation is applied to the compositional data first, the first $k$ principal component scores are calculated and used as predictor variables for the Kullback-Leibler divergence based regression model.

**Value**

A list including:

- `runtime` The time required by the regression.
- `iters` The number of iterations required by the Newton-Raphson in the kl.compreg function.
- `loglik` The log-likelihood. This is actually a quasi multinomial regression. This is basically minus the half deviance, or $-\sum_{i=1}^{n} y_i \log \hat{y}_i$.
- `be` The beta coefficients.
- `seb` The standard error of the beta coefficients, if bootstrap is chosen, i.e. if $B > 1$.
- `est` The fitted values of $x_{new}$ if $x_{new}$ is not NULL.

**Author(s)**

Initial code by Abdulaziz Alenazi. Modifications by Michail Tsagris.

R implementation and documentation: Abdulaziz Alenazi <a.alenazi@nbu.edu.sa> and Michail Tsagris <mtsagris@uoc.gr>.

**References**


**See Also**

klalfapcr.tune, tflr, pcr, glm.pcr, alfa.pcr, alfa.pcr.tune

**Examples**

```r
library(MASS)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- alfa.pcr(y = y, x = x, a = 0.7, k = 1)
mod
```
**Divergence matrix of compositional data**

**Description**
Divergence matrix of compositional data.

**Usage**
```r
divergence(x, type = "kullback_leibler", vector = FALSE)
```

**Arguments**
- `x`: A matrix with the compositional data.
- `type`: This is either "kullback_leibler" (Kullback-Leibler, which computes the symmetric Kullback-Leibler divergence) or "jensen_shannon" (Jensen-Shannon) divergence.
- `vector`: For return a vector instead a matrix.

**Details**
The function produces the distance matrix either using the Kullback-Leibler (distance) or the Jensen-Shannon (metric) divergence. The Kullback-Leibler refers to the symmetric Kullback-Leibler divergence.

**Value**
if the vector argument is FALSE a symmetric matrix with the divergences, otherwise a vector with the divergences.

**Author(s)**
Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

**References**


**See Also**
`comp.knn, js.compreg`
Empirical likelihood for a one sample mean vector hypothesis testing

Examples

```
x <- as.matrix(iris[1:20, 1:4])
x <- x / rowSums(x)
divergence(x)
```

Description

Empirical likelihood for a one sample mean vector hypothesis testing.

Usage

```
el.test1(x, mu, R = 1, ncores = 1, graph = FALSE)
```

Arguments

- `x`: A matrix containing Euclidean data.
- `mu`: The hypothesized mean vector.
- `R`: If R is 1 no bootstrap calibration is performed and the classical p-value via the $\chi^2$ distribution is returned. If R is greater than 1, the bootstrap p-value is returned.
- `ncores`: The number of cores to use, set to 1 by default.
- `graph`: A boolean variable which is taken into consideration only when bootstrap calibration is performed. If TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Multivariate hypothesis test for a one sample mean vector. This is a non parametric test and it works for univariate and multivariate data.

Value

A list with the outcome of the function `el.test` which includes the -2 log-likelihood ratio, the observed P-value by chi-square approximation, the final value of Lagrange multiplier $\lambda$, the gradient at the maximum, the Hessian matrix, the weights on the observations (probabilities multiplied by the sample size) and the number of iteration performed. In addition the runtime of the procedure is reported. In the case of bootstrap, the bootstrap p-value is also returned.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>
Empirical likelihood hypothesis testing for two mean vectors

References


See Also

eel.test1, hotel1T2, james, hotel2T2, maov, el.test2, comp.test

Examples

x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10, 0.5) ) )
el.test1(x, mu = numeric(10) )
eel.test1(x, mu = numeric(10) )

---

Empirical likelihood hypothesis testing for two mean vectors

Description

Empirical likelihood hypothesis testing for two mean vectors.

Usage

el.test2(y1, y2, R = 0, ncores = 1, graph = FALSE)

Arguments

y1
A matrix containing the Euclidean data of the first group.
y2
A matrix containing the Euclidean data of the second group.
R
If R is 0, the classical chi-square distribution is used, if R = 1, the corrected chi-square distribution (James, 1954) is used and if R = 2, the modified F distribution (Krishnamoorthy and Yanping, 2006) is used. If R is greater than 3 bootstrap calibration is performed.
ncores
How many to cores to use.
graph
A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Empirical likelihood is a non parametric hypothesis testing procedure for one sample. The generalization to two (or more samples) is via searching for the mean vector that minimizes the sum of the two test statistics.
Empirical likelihood hypothesis testing for two mean vectors

Value
A list including:

- **test**: The empirical likelihood test statistic value.
- **modif.test**: The modified test statistic, either via the chi-square or the F distribution.
- **dof**: The degrees of freedom of the chi-square or the F distribution.
- **pvalue**: The asymptotic or the bootstrap p-value.
- **mu**: The estimated common mean vector.
- **runtime**: The runtime of the bootstrap calibration.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
eel.test2, maovjames, maov, hotel2T2, james, comp.test

Examples
```
el.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 0 )
el.test2( y1 = as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 1 )
el.test2( y1 =as.matrix(iris[1:25, 1:4]), y2 = as.matrix(iris[26:50, 1:4]), R = 2 )
```
Estimating location and scatter parameters for compositional data

Description
Estimating location and scatter parameters for compositional data in a robust and non robust way.

Usage
```r
comp.den(x, type = "alr", dist = "normal", tol = 1e-09)
```

Arguments
- `x`: A matrix containing compositional data. No zero values are allowed.
- `type`: A boolean variable indicating the transformation to be used. Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.
- `dist`: Takes values "normal", "t", "skewnorn", "rob" and "spatial". They first three options correspond to the parameters of the normal, t and skew normal distribution respectively. If it set to "rob" the MCD estimates are computed and if set to "spatial" the spatial median and spatial sign covariance matrix are computed.
- `tol`: A tolerance level to terminate the process of finding the spatial median when `dist = "spatial"`. This is set to 1e-09 by default.

Details
This function calculates robust and non robust estimates of location and scatter.

Value
A list including: The mean vector and covariance matrix mainly. Other parameters are also returned depending on the value of the argument "dist".

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

spatmed.reg,multivt

Examples

library(MASS)
x <- as.matrix(iris[,1:4])
x <- x / rowSums(x)
comp.den(x)
comp.den(x, type = "alr", dist = "t")
comp.den(x, type = "alr", dist = "spatial")

Estimation of the probability left outside the simplex when using the alpha-transformation

Estimation of the probability left outside the simplex when using the alpha-transformation

Description

Estimation of the probability left outside the simplex when using the alpha-transformationn.

Usage

probout(mu, su, a)

Arguments

mu The mean vector.
su The covariance matrix.
a The value of \( \alpha \).

Details

When applying the \( \alpha \)-transformation based on a multivariate normal there might be probability left outside the simplex as the space of this transformation is a subspace of the Euclidean space. The function estimates the missing probability via Monte Carlo simulation using 40 million generated vectors.
Estimation of the value of alpha in the folded model

Value

The estimated probability left outside the simplex.

Author(s)

Michail Tsagris.
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

alfa, alpha.mle, a.est, rfolded

Examples

```r
## Not run:
s <- c(0.1490676523, -0.4580818209, 0.0020395316, -0.0047446076, -0.4580818209,
1.5227259250, 0.0002596411, 0.0074836251, 0.0020395316, 0.0002596411,
0.0365384838, -0.0471448849, -0.0047446076, 0.0074836251, -0.0471448849,
0.0611442781)
s <- matrix(s, ncol = 4)
m <- c(1.715, 0.914, 0.115, 0.167)
probout(m, s, 0.5)
## End(Not run)
```
Details

This is a function for choosing or estimating the value of $\alpha$ in the folded model (Tsagris and Stewart, 2019).

Value

A list including:

- runtime: The runtime of the algorithm.
- best: The estimated optimal $\alpha$ of the folded model.
- loglik: The maximimised log-likelihood of the folded model.
- p: The estimated probability inside the simplex of the folded model.
- mu: The estimated mean vector of the folded model.
- su: The estimated covariance matrix of the folded model.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

alfa.profile, alfa, alfainv, alpha.mle

Examples

```r
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
a.est(x)
```
Estimation of the value of alpha via the profile log-likelihood

Description

Estimation of the value of \( \alpha \) via the alfa profile log-likelihood.

Usage

alfa.profile(x, a = seq(-1, 1, by = 0.01))

Arguments

- \( x \): A matrix with the compositional data. Zero values are not allowed.
- \( a \): A grid of values of \( \alpha \).

Details

For every value of \( \alpha \) the normal likelihood (see the reference) is computed. At the end, the plot of the values is constructed.

Value

A list including:

- \( \text{res} \): The chosen value of \( \alpha \), the corresponding log-likelihood value and the log-likelihood when \( \alpha = 0 \).
- \( \text{ci} \): An asymptotic 95% confidence interval computed from the log-likelihood ratio test.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

alfa.tune, alfa, alfainv
Examples

```r
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
alfa.profile(x)
```

Description

Exponential empirical likelihood for a one sample mean vector hypothesis testing.

Usage

```r
eel.test1(x, mu, tol = 1e-06, R = 1)
```

Arguments

- `x`: A matrix containing Euclidean data.
- `mu`: The hypothesized mean vector.
- `tol`: The tolerance value used to stop the Newton-Raphson algorithm.
- `R`: The number of bootstrap samples used to calculate the p-value. If `R = 1` (default value), no bootstrap calibration is performed.

Details

Multivariate hypothesis test for a one sample mean vector. This is a non parametric test and it works for univariate and multivariate data.

Value

A list including:

- `p`: The estimated probabilities.
- `lambda`: The value of the Lagrangian parameter $\lambda$.
- `iter`: The number of iterations required by the newton-Raphson algorithm.
- `info`: The value of the log-likelihood ratio test statistic along with its corresponding p-value.
- `runtime`: The runtime of the process.
Exponential empirical likelihood hypothesis testing for two mean vectors

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
el.test1, hotel1T2, james, hotel2T2, maov, el.test2, comp.test

Examples
```r
x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10, 0.5) ) )
eel.test1(x, numeric(10) )
eel.test1(x, numeric(10) )
```

Exponential empirical likelihood hypothesis testing for two mean vectors

Exponential empirical likelihood hypothesis testing for two mean vectors

Description
Exponential empirical likelihood hypothesis testing for two mean vectors.

Usage
```r
eel.test2(y1, y2, tol = 1e-07, R = 0, graph = FALSE)
```

Arguments
- `y1`: A matrix containing the Euclidean data of the first group.
- `y2`: A matrix containing the Euclidean data of the second group.
- `tol`: The tolerance level used to terminate the Newton-Raphson algorithm.
- `R`: If R is 0, the classical chi-square distribution is used, if R = 1, the corrected chi-square distribution (James, 1954) is used and if R = 2, the modified F distribution (Krishnamoorthy and Yanping, 2006) is used. If R is greater than 3 bootstrap calibration is performed.
- `graph`: A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.
Exponential empirical likelihood hypothesis testing for two mean vectors

Details

Exponential empirical likelihood is a non parametric hypothesis testing procedure for one sample. The generalization to two (or more samples) is via searching for the mean vector that minimises the sum of the two test statistics.

Value

A list including:

- **test**: The empirical likelihood test statistic value.
- **modif.test**: The modified test statistic, either via the chi-square or the F distribution.
- **dof**: The degrees of freedom of the chi-square or the F distribution.
- **pvalue**: The asymptotic or the bootstrap p-value.
- **mu**: The estimated common mean vector.
- **runtime**: The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

el.test2, maovjames, maov, hotel2T2, james, comp.test
Examples

```r
y1 = as.matrix(iris[1:25, 1:4])
y2 = as.matrix(iris[26:50, 1:4])
eel.test2(y1, y2)
eel.test2(y1, y2)
eel.test2( y1, y2 )
```

Description

Fast estimation of the value of \( \alpha \).

Usage

```r
alfa.tune(x, B = 1, ncores = 1)
```

Arguments

- `x` A matrix with the compositional data. No zero values are allowed.
- `B` If no (bootstrap based) confidence intervals should be returned this should be 1 and more than 1 otherwise.
- `ncores` If ncores is greater than 1 parallel computing is performed. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.

Details

This is a faster function than `alfa.profile` for choosing the value of \( \alpha \).

Value

A vector with the best alpha, the maximised log-likelihood and the log-likelihood at \( \alpha = 0 \), when \( B = 1 \) (no bootstrap). If \( B > 1 \) a list including:

- `param` The best alpha and the value of the log-likelihood, along with the 95% bootstrap based confidence intervals.
- `message` A message with some information about the histogram.
- `runtime` The time (in seconds) of the process.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>
Fitting a Dirichlet distribution

References

See Also
alfa.profile, alfa, alfainv

Examples
library(MASS)
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
alfa.tune(x)
alfa.profile(x)

Description
Estimation of the parameters of a fitted Dirichlet distribution.

Usage
diri.est(x, type = "mle")

Arguments
x A matrix containing compositional data.
type If you want to estimate the parameters use type="mle". If you want to estimate the mean vector along with the precision parameter, the second parametrisation of the Dirichlet, use type="prec".

Details
Maximum likelihood estimation of the parameters of a Dirichlet distribution is performed.

Value
A list including:
loglik The value of the log-likelihood.
param The estimated parameters.
phi The estimated precision parameter, if type = "prec".
a The estimated mean vector, if type = "prec".
runtime The run time of the maximisation procedure.
Fitting a Dirichlet distribution via Newton-Raphson

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
diri.nr, diri.contour, rdiri, ddiri

Examples
x <- rdiri(100, c(5, 7, 1, 3, 10, 2, 4))
diri.est(x)
diri.est(x, type = "prec")

Description
Fitting a Dirichlet distribution via Newton-Raphson.

Usage
diri.nr(x, type = 1, tol = 1e-07)

Arguments
x A matrix containing compositional data. Zeros are not allowed.
type Type can either be 1, so that the Newton-Raphson is used for the maximisation of the log-likelihood, as Minka (2012) suggested or it can be 1. In the latter case the Newton-Raphson algorithm is implemented involving matrix inversions. In addition an even faster implementation has been implemented (in C++) in the package Rfast and is used here.
tol The tolerance level indicating no further increase in the log-likelihood.
Details

Maximum likelihood estimation of the parameters of a Dirichlet distribution is performed via Newton-Raphson. Initial values suggested by Minka (2003) are used. The estimation is super faster than "diri.est" and the difference becomes really apparent when the sample size and or the dimensions increase. In fact this will work with millions of observations. So in general, I trust this one more than "diri.est".

The only problem I have seen with this method is that if the data are concentrated around a point, say the center of the simplex, it will be hard for this and the previous methods to give estimates of the parameters. In this extremely difficult scenario I would suggest the use of the previous function with the precision parametrization "diri.est(x, type = "prec")". It will be extremely fast and accurate.

Value

A list including:

iter The number of iterations required. If the argument "type" is set to 2 this is not returned.
loglik The value of the log-likelihood.
param The estimated parameters.
runtime The run time of the procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

diri.est, diri.contour rdiri, ddiri

Examples

```r
x <- rdiri( 100, c(5, 7, 5, 8, 10, 6, 4) )
diri.nr(x)
diri.nr(x, type = 2)
diri.est(x)
```
Description

Fitting a Flexible Dirichlet distribution.

Usage

```r
fd.est(x, ini.iter = 50, final.iter = 100)
```

Arguments

- `x`: A matrix or a dataframe containing the compositional data.
- `ini.iter`: Number of iterations for the initial SEM step. Default value is 50.
- `final.iter`: Number of iterations for the final EM step. Default value is 100.

Details

For more information see the references.

Value

A list including:

- `alpha`: Estimated values of the parameter vector `alpha`.
- `prob`: Estimated values of the parameter vector `p`.
- `tau`: Estimated value of the parameter `tau`.
- `loglik`: The Log-likelihood value.

Author(s)

Michail Tsagris ported from the R package FlexDir. <mtsagris@uoc.gr>.

References


See Also

`rfd`, `fd.density`
Gaussian mixture models for compositional data

Examples

```r
## Not run:
x <- rfd(n = 50, a = c(12, 11, 10), p = c(0.25, 0.25, 0.5), tau = 4)
ela <- fd.est(x, ini.iter = 10, final.iter = 20)
ela
## End(Not run)
```

Gaussian mixture models for compositional data

Description

Gaussian mixture models for compositional data.

Usage

```r
mix.compnorm(x, g, model, type = "alr")
```

Arguments

- **x**: A matrix with the compositional data.
- **g**: How many clusters to create.
- **model**: The type of model to be used.
  1. "EII": All groups have the same diagonal covariance matrix, with the same variance for all variables.
  2. "VII": Different diagonal covariance matrices, with the same variance for all variables within each group.
  3. "EEI": All groups have the same diagonal covariance matrix.
  4. "VEI": Different diagonal covariance matrices. If we make all covariance matrices have determinant 1, (divide the matrix with the $p$-th root of its determinant) then all covariance matrices will be the same.
  5. "EVI": Different diagonal covariance matrices with the same determinant.
  6. "VVI": Different diagonal covariance matrices, with nothing in common.
  7. "EEE": All covariance matrices are the same.
  8. "EEV": Different covariance matrices, but with the same determinant and in addition, if we make them have determinant 1, they will have the same trace.
  9. "VEV": Different covariance matrices but if we make the matrices have determinant 1, then they will have the same trace.
  10. "VVV": Different covariance matrices with nothing in common.
  11. "EVE": Different covariance matrices, but with the same determinant. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.
12. "VVE": Different covariance matrices, but they have something in common with their directions. Calculate the eigenvectors of each covariance matrix and you will see the similarities.

13. "VEE": Different covariance matrices, but if we make the matrices have determinant 1, then they will have the same trace. In addition, calculate the eigenvectors for each covariance matrix and you will see the extra similarities.

14. "EVV": Different covariance matrices, but with the same determinant.

type

Either the additive ("alr") or the isometric ("ilr") log-ratio transformation is to be used.

Details

A log-ratio transformation is applied and then a Gaussian mixed model is constructed.

Value

A list including:

- mu: A matrix where each row corresponds to the mean vector of each cluster.
- su: An array containing the covariance matrix of each cluster.
- prob: The estimated mixing probabilities.
- est: The estimated cluster membership values.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

bic.mixcompnorm, rmixcomp, mixnorm.contour

Examples

```r
## Not run:
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod1 <- mix.compnorm(x, 3, model = "EII")
mod2 <- mix.compnorm(x, 4, model = "VII")
## End(Not run)
```
Generate random folds for cross-validation

Description

Random folds for use in a cross validation are generated. There is the option for stratified splitting as well.

Usage

makefolds(ina, nfolds = 10, stratified = TRUE, seed = FALSE)

Arguments

ina A variable indicating the groupings.
nfolds The number of folds to produce.
stratified A boolean variable specifying whether stratified random (TRUE) or simple random (FALSE) sampling is to be used when producing the folds.
seed A boolean variable. If set to TRUE, the folds will always be the same.

Details

I was inspired by the command in the package TunePareto in order to do the stratified version.

Value

A list with nfolds elements where each elements is a fold containing the indices of the data.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

See Also

rda.tune

Examples

a <- makefolds(iris[, 5], nfolds = 5, stratified = TRUE)
table(iris[a[,1]], 5)  ## 10 values from each group
Greenacre’s power transformation

**Description**

Greenacre’s power transformation.

**Usage**

`green(x, theta)`

**Arguments**

- `x` A matrix with the compositional data.
- `theta` The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\theta = 0$ the log transformation is applied.

**Details**

Greenacre’s transformation is applied to the compositional data.

**Value**

A matrix with the power transformed data.

**Author(s)**

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

**References**


**See Also**

`alfa`

**Examples**

```r
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- green(x, 0.1)
y2 <- green(x, 0.2)
rbind( colMeans(y1), colMeans(y2) )
```
Helper Frechet mean for compositional data

Description

Helper Frechet mean for compositional data.

Usage

frechet2(x, di, a, k)

Arguments

x
A matrix with the compositional data.
di
A matrix with indices as produced by the function "dista" of the package "Rfast" or the function "nn2" of the package "RANN". Better see the details section.
a
The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied and the closed geometric mean is calculated.
k
The number of nearest neighbours used for the computation of the Frechet means.

Details

The power transformation is applied to the compositional data and the mean vector is calculated. Then the inverse of it is calculated and the inverse of the power transformation applied to the last vector is the Frechet mean.

What this helper function do is to speed up the Frechet mean when used in the $\alpha$-k-NN regression. The $\alpha$-k-NN regression computes the Frechet mean of the k nearest neighbours for a value of $\alpha$ and this function does exactly that. Suppose you want to predict the compositional value of some new predictors. For each predictor value you must use the Frechet mean computed at various nearest neighbours. This function performs these computations in a fast way. It is not the fastest way, yet it is a pretty fast way. This function is being called inside the function aknn.reg.

Value

A list where each element contains a matrix. Each matrix contains the Frechet means computed at various nearest neighbours.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.
References

See Also
alfa, alfainv, profile

Examples
## Not run:
library(MASS)
library(Rfast)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
xnew <- x[1:10, ]
x <- x[-c(1:10), ]
k <- 2:5
di <- Rfast::dista(xnew, x, k = max(k), index = TRUE, square = TRUE )
est <- frechet2(x, di, 0.2, k)
## End(Not run)
Helper functions for the Kullback-Leibler regression

\[ \text{der} \quad \text{An vector to put the first derivative there.} \]

\[ \text{der2} \quad \text{An empty matrix to put the second derivatives there, the Hessian matrix will be put here.} \]

\[ \text{id} \quad \text{A help vector with indices.} \]

\[ \text{b1} \quad \text{The matrix with the initial estimated coefficients.} \]

\[ \text{n} \quad \text{The sample size} \]

\[ \text{p} \quad \text{The number of columns of the design matrix.} \]

\[ \text{d} \quad \text{The dimensionality of the simplex, that is the number of columns of the compositional data minus 1.} \]

Details

These are help functions for the \texttt{kl.compreg} function. They are not to be called directly by the user.

Value

For \texttt{kl.compreg2} a list including:

\[ \text{iters} \quad \text{The number of iterations required by the Newton-Raphson.} \]

\[ \text{loglik} \quad \text{The loglikelihood.} \]

\[ \text{be} \quad \text{The beta coefficients.} \]

\[ \text{est} \quad \text{The fitted or the predicted values (if xnew is not NULL).} \]

For \texttt{klcompreg.boot} a list including:

\[ \text{loglik} \quad \text{The loglikelihood.} \]

\[ \text{be} \quad \text{The beta coefficients.} \]

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

\texttt{diri.reg, js.compreg, ols.compreg, comp.reg}
Hotelling’s multivariate version of the 1 sample t-test for Euclidean data

Examples

```r
library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1 <- kl.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)
```

Description

Hotelling’s test for testing one Euclidean population mean vector.

Usage

```r
hotel1T2(x, M, a = 0.05, R = 999, graph = FALSE)
```

Arguments

- `x`: A matrix containing Euclidean data.
- `a`: The significance level, set to 0.05 by default.
- `M`: The hypothesized mean vector.
- `R`: If R is 1 no bootstrap calibration is performed and the classical p-value via the F distribution is returned. If R is greater than 1, the bootstrap p-value is returned.
- `graph`: A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Multivariate hypothesis test for a one sample mean vector. This is the multivariate analogue of the one sample t-test. The p-value can be calculated either asymptotically or via bootstrap.

Value

A list including:

- `m`: The sample mean vector.
- `info`: The test statistic, the p-value, the critical value and the degrees of freedom of the F distribution (numerator and denominator). This is given if no bootstrap calibration is employed.
- `pvalue`: The bootstrap p-value is bootstrap is employed.
- `runtime`: The runtime of the bootstrap calibration.
Hotelling's multivariate version of the 2 sample t-test for Euclidean data

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also
eel.test1, el.test1, james, hotel2T2, maov, el.test2, comp.test

Examples

x <- Rfast::rmvnorm(100, numeric(10), diag( rexp(10,0.5) ) )
hotel1T2(x, numeric(10), R = 1)
hotel1T2(x, numeric(10), R = 999, graph = TRUE)

Description

Hotelling's test for testing the equality of two Euclidean population mean vectors.

Usage

hotel2T2(x1, x2, a = 0.05, R = 999, graph = FALSE)

Arguments

x1 A matrix containing the Euclidean data of the first group.
x2 A matrix containing the Euclidean data of the second group.
a The significance level, set to 0.05 by default.
R If R is 1 no bootstrap calibration is performed and the classical p-value via the F distribution is returned. If R is greater than 1, the bootstrap p-value is returned.
graph A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

Details

Multivariate analysis of variance assuming equality of the covariance matrices. The p-value can be calculated either asymptotically or via bootstrap.
Hypothesis testing for two or more compositional mean vectors

Value

A list including:

- **mesoi**: The two mean vectors.
- **info**: The test statistic, the p-value, the critical value and the degrees of freedom of the F distribution (numerator and denominator). This is given if no bootstrap calibration is employed.
- **pvalue**: The bootstrap p-value is bootstrap is employed.
- **note**: A message informing the user that bootstrap calibration has been employed.
- **runtime**: The runtime of the bootstrap calibration.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

-james, maov, el.test2, comp.test

Examples

```r
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 1 )
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 1 )
```

Description

Hypothesis testing for two or more compositional mean vectors.

Usage

```r
comp.test(x, ina, test = "james", R = 0, ncores = 1, graph = FALSE)
```
Hypothesis testing for two or more compositional mean vectors

Arguments

\( \mathbf{x} \) A matrix containing compositional data.

\( \text{ina} \) A numerical or factor variable indicating the groups of the data.

\( \text{test} \) This can take the values of "james" for James’ test, "hotel" for Hotelling’s test, "maov" for multivariate analysis of variance assuming equality of the covariance matrices, "maovjames" for multivariate analysis of variance without assuming equality of the covariance matrices. "el" for empirical likelihood or "eel" for exponential empirical likelihood.

\( R \) This depends upon the value of the argument "test". If the test is "maov" or "maovjames", \( R \) is not taken into consideration. If test is "hotel", then \( R \) denotes the number of bootstrap resamples. If test is "james", then \( R \) can be 1 (chi-square distribution), 2 (F distribution), or more for bootstrap calibration. If test is "el", then \( R \) can be 0 (chi-square), 1 (corrected chi-square), 2 (F distribution) or more for bootstrap calibration. See the help page of each test for more information.

\( \text{ncores} \) How many to cores to use. This is taken into consideration only if test is "el" and \( R \) is more than 2.

\( \text{graph} \) A boolean variable which is taken into consideration only when bootstrap calibration is performed. If TRUE the histogram of the bootstrap test statistic values is plotted. This is taken into account only when \( R \) is greater than 2.

Details

The idea is to apply the \( \alpha \)-transformation, with \( \alpha = 1 \), to the compositional data and then use a test to compare their mean vectors. See the help page of each test for more information. The function is visible so you can see exactly what is going on.

Value

A list including:

\( \text{result} \) The outcome of each test.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


Inverse of the alpha-transformation


See Also

maovjames, maov, hotel2T2, el.test2

Examples

ina <- rep(1:2, each = 50)
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "james", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "hotel", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "el", R = 0 )
comp.test( as.matrix(iris[1:100, 1:4]), ina, test = "eel", R = 0 )

Inverse of the alpha-transformation

Inverse of the \(\alpha\)-transformation

Description

The inverse of the \(\alpha\)-transformation.

Usage

\texttt{alfainv(x, a, h = TRUE)}

Arguments

\texttt{x} A matrix with Euclidean data. However, they must lie within the feasible, acceptable space. See references for more information.

\texttt{a} The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \(\alpha = 0\), the inverse of the isometric log-ratio transformation is applied.

\texttt{h} If \texttt{h = TRUE} this means that the multiplication with the Helmer sub-matrix will take place. It is set to \texttt{TRUE} by default.
Details
The inverse of the \( \alpha \)-transformation is applied to the data. If the data lie outside the \( \alpha \)-space, NAs will be returned for some values.

Value
A matrix with the pairwise distances.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
alfa, alfadist

Examples

```r
library(MASS)
x <- as.matrix(fgl[1:10, 2:9])
x <- x / rowSums(x)
y <- alfa(x, 0.5)$aff
alfainv(y, 0.5)
```
James multivariate version of the t-test

**Arguments**

- **y1**: A matrix containing the Euclidean data of the first group.
- **y2**: A matrix containing the Euclidean data of the second group.
- **a**: The significance level, set to 0.05 by default.
- **R**: If R is 1 no bootstrap calibration is performed and the classical p-value via the F distribution is returned. If R is greater than 1, the bootstrap p-value is returned.
- **graph**: A boolean variable which is taken into consideration only when bootstrap calibration is performed. IF TRUE the histogram of the bootstrap test statistic values is plotted.

**Details**

Multivariate analysis of variance without assuming equality of the covariance matrices. The p-value can be calculated either asymptotically or via bootstrap. The James test (1954) or a modification proposed by Krishnamoorthy and Yanping (2006) is implemented. The James test uses a corrected chi-square distribution, whereas the modified version uses an F distribution.

**Value**

A list including:

- **note**: A message informing the user about the test used.
- **mesoi**: The two mean vectors.
- **info**: The test statistic, the p-value, the correction factor and the corrected critical value of the chi-square distribution if the James test has been used or, the test statistic, the p-value, the critical value and the degrees of freedom (numerator and denominator) of the F distribution if the modified James test has been used.
- **pvalue**: The bootstrap p-value if bootstrap is employed.
- **runtime**: The runtime of the bootstrap calibration.

**Author(s)**

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

**References**


**See Also**

hotel2T2, maovjames, el, comp.test
Examples

```r
ejames(as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 1 )
ejames(as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]), R = 2 )
ejames(as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
hotel2T2( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
```

Description

Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions.

Usage

```r
kl.diri(a, b, type = "KL")
```

Arguments

- `a`: A vector with the parameters of the first Dirichlet distribution.
- `b`: A vector with the parameters of the second Dirichlet distribution.
- `type`: A variable indicating whether the Kullback-Leibler divergence ("KL") or the Bhattacharyya distance ("bhatt") is to be computed.

Details

Note that the order is important in the Kullback-Leibler divergence, since this is asymmetric, but not in the Bhattacharyya distance, since it is a metric.

Value

The value of the Kullback-Leibler divergence or the Bhattacharyya distance.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
diri.est, diri.nr

Examples

library(MASS)
a <- runif(10, 0, 20)
b <- runif(10, 1, 10)
kl.diri(a, b)
kl.diri(b, a)
kl.diri(a, b, type = "bhatt")
kl.diri(b, a, type = "bhatt")

Description

Log-likelihood ratio test for a Dirichlet mean vector.

Usage

dirimean.test(x, a)

Arguments

x A matrix with the compositional data. No zero values are allowed.
a A compositional mean vector. The concentration parameter is estimated at first. If the elements do not sum to 1, it is assumed that the Dirichlet parameters are supplied.

Details

Log-likelihood ratio test is performed for the hypothesis the given vector of parameters "a" describes the compositional data well.

Value

If there are no zeros in the data, a list including:

param A matrix with the estimated parameters under the null and the alternative hypothesis.
loglik The log-likelihood under the alternative and the null hypothesis.
info The value of the test statistic and its relevant p-value.
Log-likelihood ratio test for a symmetric Dirichlet distribution

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

sym.test, diri.nr, diri.est, rdiri, ddiri

Examples

x <- rdiri(100, c(1, 2, 3))
dirimean.test(x, c(1, 2, 3))
dirimean.test(x, c(1, 2, 3)/6)

Log-likelihood ratio test for a symmetric Dirichlet distribution

Log-likelihood ratio test for a symmetric Dirichlet distribution

Description

Log-likelihood ratio test for a symmetric Dirichlet distribution.

Usage

sym.test(x)

Arguments

x A matrix with the compositional data. No zero values are allowed.

Details

Log-likelihood ratio test is performed for the hypothesis that all Dirichlet parameters are equal.
Value

A list including:

- **est.par**: The estimated parameters under the alternative hypothesis.
- **one.par**: The value of the estimated parameter under the null hypothesis.
- **res**: The loglikelihood under the alternative and the null hypothesis, the value of the test statistic, its relevant p-value and the associated degrees of freedom, which are actually the dimensionality of the simplex, \( D - 1 \), where \( D \) is the number of components.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

diri.nr, diri.est, rdiri, dirimean.test

Examples

```r
x <- rdiri(100, c(5, 7, 1, 3, 10, 2, 4))
sym.test(x)
x <- rdiri(100, c(5, 5, 5, 5, 5))
sym.test(x)
```

Description

Mixture model selection via BIC.

Usage

```r
bic.mixcompnorm(x, G, type = "alr", graph = TRUE)
```
Mixture model selection via BIC

Arguments

- **x**: A matrix with compositional data.
- **G**: A numeric vector with the number of components, clusters, to be considered.
- **type**: The type of transformation to be used, either additive log-ratio ("alr") or the isometric log-ratio ("ilr").
- **graph**: A boolean variable, TRUE or FALSE specifying whether a graph should be drawn or not.

Details

The alr or the ilr-transformation is applied to the compositional data first and then mixtures of multivariate Gaussian distributions are fitted. BIC is used to decide on the optimal model and number of components.

Value

- A plot with the BIC of the best model for each number of components versus the number of components. A list including:
  - **mod**: A message informing the user about the best model.
  - **BIC**: The BIC values for every possible model and number of components.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

- mix.compnorm, mixnorm.contour, rmixcomp

Examples

```r
## Not run:
x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
bic.mixcompnorm(x, 1:3, type = "alr", graph = FALSE)
bic.mixcompnorm(x, 1:3, type = "ilr", graph = FALSE)
## End(Not run)
```
MLE for the multivariate t distribution

Description

MLE of the parameters of a multivariate t distribution.

Usage

```r
multivt(y, plot = FALSE)
```

Arguments

- `y`: A matrix with continuous data.
- `plot`: If plot is TRUE the value of the maximum log-likelihood as a function of the degrees of freedom is presented.

Details

The parameters of a multivariate t distribution are estimated. This is used by the functions `comp.den` and `bivt.contour`.

Value

A list including:

- `center`: The location estimate.
- `scatter`: The scatter matrix estimate.
- `df`: The estimated degrees of freedom.
- `loglik`: The log-likelihood value.
- `mesos`: The classical mean vector.
- `covariance`: The classical covariance matrix.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

MLE of distributions defined in the (0, 1) interval

See Also

bivt.contour, comp.den, rmvt

Examples

```r
x <- as.matrix(iris[, 1:4])
multivt(x)
```

Description

MLE of distributions defined in the (0, 1) interval.

Usage

```r
beta.est(x, tol = 1e-07)
ibeta.est(x, tol = 1e-07)
logitnorm.est(x)
hsecant01.est(x, tol = 1e-07)
simplex.est(x, tol = 1e-07)
kumar.est(x, tol = 1e-07)
zilogitnorm.est(x)
```

Arguments

- **x**: A numerical vector with proportions, i.e. numbers in (0, 1) (zeros and ones are not allowed).
- **tol**: The tolerance level up to which the maximisation stops.

Details

Maximum likelihood estimation of the parameters of the beta distribution is performed via Newton-Raphson. The distributions and hence the functions does not accept zeros. "logitnorm.mle" fits the logistic normal, hence no newton-Raphson is required and the "hypersecant01.mle" and "simplex.est" use the golden ratio search as is it faster than the Newton-Raphson (less computations). The "zilogitnorm.est" stands for the zero inflated logistic normal distribution.

Value

A list including:

- **iters**: The number of iterations required by the Newton-Raphson.
- **loglik**: The value of the log-likelihood.
- **param**: The estimated parameters. In the case of "hypersecant01.est" this is called "theta" as there is only one parameter.
Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Manos Papadakis <papadakm95@gmail.com>

References
You can also check the relevant wikipedia pages.

See Also
diri.nr2,

Examples
x <- rbeta(1000, 1, 4)
beta.est(x)
ibeta.est(x)

x <- runif(1000)
hsecant01.est(x)
logitnorm.est(x)
ibeta.est(x)

x <- rbeta(1000, 2, 5)
x[sample(1:1000, 50)] <- 0
ibeta.est(x)

**MLE of the folded model for a given value of alpha**

**Mono-log-normal distribution**

MLE of the folded model for a given value of $\alpha$.

Usage
alpha.mle(x, a)
amle(a, x)
MLE of the folded model for a given value of alpha

Arguments

x A matrix with the compositional data. No zero values are allowed.
a A value of $\alpha$.

Details

This is a function for choosing or estimating the value of $\alpha$ in the folded model (Tsagris and Stewart, 2020). It is called by a.est.

Value

If "alpha.mle" is called, a list including:

- iters The number of iterations the EM algorithm required.
- loglik The maximized log-likelihood of the folded model.
- p The estimated probability inside the simplex of the folded model.
- mu The estimated mean vector of the folded model.
- su The estimated covariance matrix of the folded model.

If "a.mle" is called, the log-likelihood is returned only.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

alfa.profile, alfa, alfainv, a.est

Examples

```r
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
mod <- alfa.tune(x)
mod
alpha.mle(x, mod[1])
```
Multivariate analysis of variance

Description

Multivariate analysis of variance with assuming equality of the covariance matrices.

Usage

maov(x, ina)

Arguments

x
A matrix containing Euclidean data.

ina
A numerical or factor variable indicating the groups of the data.

Details

Multivariate analysis of variance assuming equality of the covariance matrices.

Value

A list including:

note
A message stating whether the F or the chi-square approximation has been used.

result
The test statistic and the p-value.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

maovjames, hotel2T2, james, comp.test

Examples

maov( as.matrix(iris[,1:4]), iris[,5] )
maovjames( as.matrix(iris[,1:4]), iris[,5] )
Multivariate analysis of variance (James test)

Description
Multivariate analysis of variance without assuming equality of the covariance matrices.

Usage
maovjames(x, ina, a = 0.05)

Arguments
x A matrix containing Euclidean data.
ina A numerical or factor variable indicating the groups of the data.
a The significance level, set to 0.005 by default.

Details
Multivariate analysis of variance without assuming equality of the covariance matrices.

Value
A vector with the next 4 elements:
test The test statistic.
correction The value of the correction factor.
corr.critical The corrected critical value of the chi-square distribution.
p-value The p-value of the corrected test statistic.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
maov,hotel2T2,james,comp.test
Examples

```r
maov( as.matrix(iris[,1:4]), iris[,5] )
maovjames( as.matrix(iris[,1:4]), iris[,5] )
```

Description

Multivariate kernel density estimation.

Usage

```r
mkde(x, h, thumb = "silverman")
```

Arguments

- `x`: A matrix with Euclidean (continuous) data.
- `h`: The bandwidth value. It can be a single value, which is turned into a vector and then into a diagonal matrix, or a vector which is turned into a diagonal matrix.
- `thumb`: Do you want to use a rule of thumb for the bandwidth parameter? If no, leave it "none", or else put "estim" for maximum likelihood cross-validation, "scott" or "silverman" for Scott’s and Silverman’s rules of thumb respectively.

Details

The multivariate kernel density estimate is calculated with a (not necessarily given) bandwidth value. It is used a wrapper for the function `comp.kerncontour`.

Value

A vector with the density estimates calculated for every vector.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


Multivariate linear regression

See Also

mkde.tune, comp.kerncontour

Examples

mkde( as.matrix(iris[, 1:4]), thumb = "scott"

mkde( as.matrix(iris[, 1:4]), thumb = "silverman"

Description

Multivariate linear regression.

Usage

multivreg(y, x, plot = TRUE, xnew = NULL)

Arguments

y A matrix with the Euclidean (continuous) data.

x A matrix with the predictor variable(s), they have to be continuous.

plot Should a plot appear or not?

xnew If you have new data use it, otherwise leave it NULL.

Details

The classical multivariate linear regression model is obtained.

Value

A list including:

suma A summary as produced by \texttt{lm}, which includes the coefficients, their standard error, t-values, p-values.

r.squared The value of the $R^2$ for each univariate regression.

resid.out A vector with number indicating which vectors are potential residual outliers.

x.leverage A vector with number indicating which vectors are potential outliers in the predictor variables space.

out A vector with number indicating which vectors are potential outliers in the residuals and in the predictor variables space.

est The predicted values if xnew is not NULL.
Multivariate normal random values simulation on the simplex

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
diri.reg, js.compreg, kl.compreg, ols.compreg, comp.reg

Examples
library(MASS)
x <- as.matrix(iris[, 1:2])
y <- as.matrix(iris[, 3:4])
multivreg(y, x, plot = TRUE)

Description
Multivariate normal random values simulation on the simplex.

Usage
rcompnorm(n, m, s, type = "alr")

Arguments
n The sample size, a numerical value.
m The mean vector in \( R^d \).
s The covariance matrix in \( R^d \).
type The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.

Details
The algorithm is straightforward, generate random values from a multivariate normal distribution in \( R^d \) and brings the values to the simplex \( S^d \) using the inverse of a log-ratio transformation.
Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation

Value
A matrix with the simulated data.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also

comp.den, rdiri, rcompt, rcompsn

Examples

```r
x <- as.matrix(iris[, 1:2])
m <- colMeans(x)
s <- var(x)
y <- rcompnorm(100, m, s)
comp.den(y)
ternary(y)
```

Description
Multivariate or univariate regression with compositional data in the covariates side using the \( \alpha \)-transformation.

Usage

```r
alfa.pcr(y, x, a, k, model = "gaussian", xnew = NULL)
```

Arguments

- **y**: A numerical vector containing the response variable values. They can be continuous, binary, discrete (counts). This can also be a vector with discrete values or a factor for the multinomial regression (model = "multinomial").
- **x**: A matrix with the predictor variables, the compositional data.
Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation

\( a \)  
The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \( \alpha = 0 \) the isometric log-ratio transformation is applied.

\( k \)  
A number at least equal to 1. How many principal components to use.

\( \text{model} \)  
The type of regression model to fit. The possible values are "gaussian", "multinomial", "binomial" and "poisson".

\( \text{xnew} \)  
A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.

Details

The \( \alpha \)-transformation is applied to the compositional data first, the first \( k \) principal component scores are calculated and used as predictor variables for a regression model. The family of distributions can be either, "normal" for continuous response and hence normal distribution, "binomial" corresponding to binary response and hence logistic regression or "poisson" for count response and poisson regression.

Value

A list including:

\( \text{be} \)  
If linear regression was fitted, the regression coefficients of the \( k \) principal component scores on the response variable \( y \).

\( \text{mod} \)  
If another regression model was fitted its outcome as produced in the package \text{Rfast}.

\( \text{per} \)  
The percentage of variance explained by the first \( k \) principal components.

\( \text{vec} \)  
The first \( k \) principal components, loadings or eigenvectors. These are useful for future prediction in the sense that one needs not fit the whole model again.

\( \text{est} \)  
If the argument "\( \text{xnew} \)" was given these are the predicted or estimated values, otherwise it is NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

\text{pcr}, \text{glm.pcr}, \text{alfapcr.tune}
Multivariate regression with compositional data

Examples

```r
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- alfa.pcr(y = y, x = x, 0.7, 1)
mod
```

Multivariate regression with compositional data

Description

Multivariate regression with compositional data.

Usage

```r
comp.reg(y, x, type = "classical", xnew = NULL, yb = NULL)
```

Arguments

- `y`: A matrix with compositional data. Zero values are not allowed.
- `x`: The predictor variable(s), they have to be continuous.
- `type`: The type of regression to be used, "classical" for standard multivariate regression, or "spatial" for the robust spatial median regression. Alternatively you can type "lmfit" for the fast classical multivariate regression that does not return standard errors whatsoever.
- `xnew`: This is by default set to NULL. If you have new data whose compositional data values you want to predict, put them here.
- `yb`: If you have already transformed the data using the additive log-ratio transformation, put it here. Otherwise leave it NULL. This is intended to be used in the function `alfareg.tune` in order to speed up the process.

Details

The additive log-ratio transformation is applied and then the chosen multivariate regression is implemented. The alr is easier to explain than the ilr and that is why the latter is avoided here.

Value

A list including:

- `runtime`: The time required by the regression.
- `be`: The beta coefficients.
- `seb`: The standard error of the beta coefficients.
- `est`: The fitted values of xnew if xnew is not NULL.
Multivariate skew normal random values simulation on the simplex

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
multivreg, spatmed.reg, js.compreg, diri.reg

Examples
library(MASS)
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
x <- as.vector(iris[, 4])
mod1 <- comp.reg(y, x)
mod2 <- comp.reg(y, x, type = "spatial")

Description
Multivariate skew normal random values simulation on the simplex.

Usage
rcompsn(n, xi, Omega, alpha, dp = NULL, type = "alr")

Arguments
n The sample size, a numerical value.
xi A numeric vector of length \(d\) representing the location parameter of the distribution.
Omega A \(d \times d\) symmetric positive-definite matrix of dimension.
alpha A numeric vector which regulates the slant of the density.
dp A list with three elements, corresponding to xi, Omega and alpha described above. The default value is FALSE. If dp is assigned, individual parameters must not be specified.
type The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.
Multivariate t random values simulation on the simplex

Details
The algorithm is straightforward, generate random values from a multivariate t distribution in $R^d$ and brings the values to the simplex $S^d$ using the inverse of a log-ratio transformation.

Value
A matrix with the simulated data.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
comp.den, rdiri, rcomppnorm, rmvt

Examples
x <- as.matrix(iris[, 1:2])
par <- sn::msn.mle(y = x)$dp
y <- rcompt(100, dp = par)
comp.den(y, dist = "skewnorm")
ternary(y)
Multivariate t random values simulation on the simplex

Arguments

- **n**: The sample size, a numerical value.
- **m**: The mean vector in \( R^d \).
- **s**: The covariance matrix in \( R^d \).
- **dof**: The degrees of freedom.
- **type**: The alr (type = "alr") or the ilr (type = "ilr") is to be used for closing the Euclidean data onto the simplex.

Details

The algorithm is straightforward, generate random values from a multivariate t distribution in \( R^d \) and brings the values to the simplex \( S^d \) using the inverse of a log-ratio transformation.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

- `comp.den`, `rdiri`, `rcompnorm`, `rmvt`

Examples

```r
x <- as.matrix(iris[, 1:2])
m <- Rfast::colmeans(x)
s <- var(x)
y <- rcompt(100, m, s, 10)
comp.den(y, dist = "t")
ternary(y)
```
Description

Non linear least squares regression for compositional data.

Usage

```r
ols.compreg(y, x, B = 1, ncores = 1, xnew = NULL)
```

Arguments

- `y`: A matrix with the compositional data (dependent variable). Zero values are allowed.
- `x`: The predictor variable(s), they have to be continuous.
- `B`: If `B` is greater than 1 bootstrap estimates of the standard error are returned. If `B`=1, no standard errors are returned.
- `ncores`: If `ncores` is 2 or more parallel computing is performed. This is to be used for the case of bootstrap. If `B`=1, this is not taken into consideration.
- `xnew`: If you have new data use it, otherwise leave it NULL.

Details

The ordinary least squares between the observed and the fitted compositional data is adopted as the objective function. This involves numerical optimization since the relationship is non linear. There is no log-likelihood.

Value

A list including:

- `runtime`: The time required by the regression.
- `beta`: The beta coefficients.
- `covb`: The covariance matrix of the beta coefficients, if bootstrap is chosen, i.e. if `B` > 1.
- `est`: The fitted of `xnew` if `xnew` is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>`.
Perturbation operation

References


See Also

diri.reg, js.compreg, kl.compreg, comp.reg, comp.reg, alfa.reg

Examples

library(MASS)
x <- as.vector(fgl[, 1])
y <- as.matrix(fgl[, 2:9])
y <- y / rowSums(y)
mod1 <- ols.compreg(y, x, B = 1, ncores = 1)
mod2 <- js.compreg(y, x, B = 1, ncores = 1)

---

Perturbation operation

Description

Perturbation operation.

Usage

perturbation(x, y, oper = "+")

Arguments

x A matrix with the compositional data.
y Either a matrix with compositional data or a vector with compositional data. In either case, the data may not be compositional data, as long as they non negative.
oper For the summation this must be "*" and for the negation it must be "/". According to Aitchison (1986), multiplication is equal to summation in the log-space, and division is equal to negation.

Details

This is the perturbation operation defined by Aitchison (1986).

Value

A matrix with the perturbed compositional data.
Power operation

**Author(s)**
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

**References**

**See Also**
<code>power</code>

**Examples**
```r
x <- as.matrix(iris[1:15, 1:4])
y <- as.matrix(iris[21:35, 1:4])
perturbation(x, y)
perturbation(x, y[1, ])
```

---

**Description**
Power operation.

**Usage**
```r
pow(x, a)
```

**Arguments**
- `x`: A matrix with the compositional data.
- `a`: Either a vector with numbers of a single number.

**Details**
This is the power operation defined by Aitchison (1986). It is also the starting point of the $\alpha$-transformation.

**Value**
A matrix with the power transformed compositional data.

**Author(s)**
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>
Principal component generalised linear models

References


See Also

perturbation, alfa

Examples

\[
x \leftarrow \text{as.matrix(iris[1:15, 1:4])}
\]
\[
a \leftarrow \text{runif(1)}
\]
\[
pow(x, a)
\]

Description

Principal component generalised linear models.

Usage

pcr(y, x, k = 1, xnew = NULL)
glm.pcr(y, x, k = 1, xnew = NULL)

Arguments

y A numerical vector, a real values vector or a numeric vector with 0 and 1 (binary) or a vector with discrete (count) data.
x A matrix with the predictor variable(s), they have to be continuous.
k A number greater than or equal to 1. How many principal components to use. In the case of "pcr" this can be a single number or a vector. In the second case you get results for the sequence of principal components.
xnew If you have new data use it, otherwise leave it NULL.

Details

Principal component regression is performed with linear, binary logistic or Poisson regression, depending on the nature of the response variable. The principal components of the cross product of the independent variables are obtained and classical regression is performed. This is used in the function alfa.pcr.
Principal component generalised linear models

Value

A list including:

- **be** The beta coefficients of the predictor variables computed via the principal components if "pcr" is used.
- **model** The summary of the logistic or Poisson regression model.
- **per** The percentage of variance of the predictor variables retained by the k principal components.
- **vec** The principal components, the loadings.
- **est** The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

- alfa.pcr, alfafpcr.tune

Examples

```r
library(MASS)
x <- as.matrix(fgl[, 2:9])
y <- as.vector(fgl[, 1])
mod1 <- pcr(y, x, 1)
mod2 <- pcr(y, x, 2)
mod <- pcr(y, x, k = 1:4)  # many results at once

x <- as.matrix(iris[, 1:4])
y <- rbinom(150, 1, 0.6)
mod <- glm.pcr(y, x, k = 1)
```
Projection pursuit regression for compositional data

Description

Projection pursuit regression for compositional data.

Usage

```r
comp.ppr(y, x, nterms = 3, type = "alr", xnew = NULL, yb = NULL )
```

Arguments

- `y` A matrix with the compositional data.
- `x` A matrix with the continuous predictor variables or a data frame including categorical predictor variables.
- `nterms` The number of terms to include in the final model.
- `type` Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.
- `xnew` If you have new data use it, otherwise leave it NULL.
- `yb` If you have already transformed the data using a log-ratio transformation put it here. Otherwise leave it NULL.

Details

This is the standard projection pursuit. See the built-in function "ppr" for more details.

Value

A list including:

- `runtime` The runtime of the regression.
- `mod` The produced model as returned by the function "ppr".
- `est` The fitted values of `xnew` if `xnew` is not NULL.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Quasi binomial regression for proportions

See Also

compknn.tune, rda, alfa

Examples

```r
x <- as.matrix( iris[, 1:4] )
x <- x / rowSums(x)
ina <- iris[, 5]
mod <- comp.knn(x, x, ina, a = 1, k = 5)
table(ina, mod)
mod2 <- alfa.knn(x, x, ina, a = 1, k = 5)
table(ina, mod2)
```

Description

Quasi binomial regression for proportions.

Usage

```r
propreg(y, x, varb = "quasi", tol = 1e-07, maxiters = 100)
propregs(y, x, varb = "quasi", tol = 1e-07, logged = FALSE, maxiters = 100)
```

Arguments

- `y` A numerical vector proportions. 0s and 1s are allowed.
- `x` For the "propreg" a matrix with data, the predictor variables. This can be a matrix or a data frame. For the "propregs" this must be a numerical matrix, where each columns denotes a variable.
- `tol` The tolerance value to terminate the Newton-Raphson algorithm. This is set to $10^{-9}$ by default.
- `varb` The type of estimate to be used in order to estimate the covariance matrix of the regression coefficients. There are two options, either "quasi" (default value) or "glm". See the references for more information.
- `logged` Should the p-values be returned (FALSE) or their logarithm (TRUE)?
- `maxiters` The maximum number of iterations before the Newton-Raphson is terminated automatically.

Details

We are using the Newton-Raphson, but unlike R's built-in function "glm" we do no checks and no extra calculations, or whatever. Simply the model. The "propregs" is to be used for very many univariate regressions. The "x" is a matrix in this case and the significance of each variable (column of the matrix) is tested. The function accepts binary responses as well (0 or 1).
Quasi binomial regression for proportions

Value

For the "propreg" function a list including:

- **iters**: The number of iterations required by the Newton-Raphson.
- **varb**: The covariance matrix of the regression coefficients.
- **phi**: The phi parameter is returned if the input argument "varb" was set to "glm", otherwise this is NULL.
- **info**: A table similar to the one produced by "glm" with the estimated regression coefficients, their standard error, Wald test statistic and p-values.

For the "proregs" a two-column matrix with the test statistics (Wald statistic) and the associated p-values (or their logarithm).

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Manos Papadakis <papadakm95@gmail.com>.

References


See Also

- anova_propreg univglms, score.glms, logistic_only

Examples

```r
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(100 * 3), ncol = 3)
a <- propreg(y, x)
y <- rbeta(100, 1, 4)
x <- matrix(rnorm(400 * 100), ncol = 400)
b <- propregs(y, x)
mean(b[, 2] < 0.05)
```
Regression with compositional data using the alpha-transformation

Description

Regression with compositional data using the $\alpha$-transformation.

Usage

alfa.reg(y, x, a, xnew = NULL, yb = NULL, seb = FALSE)

Arguments

y A matrix with the compositional data.

x A matrix with the continuous predictor variables or a data frame including categorical predictor variables.

a The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied and the solution exists in a closed form, since it the classical mutivariate regression.

xnew If you have new data use it, otherwise leave it NULL.

yb If you have already transformed the data using the $\alpha$-transformation with the same $\alpha$ as given in the argument "a", put it here. Otherwise leave it NULL. This is intended to be used in the function alfareg.tune in order to speed up the process. The time difference in that function is small for small samples. But, if you have a few thousands and or a few more components, there will be bigger differences.

seb Do you want the standard error of the coefficients to be returned? In the alfareg.tune function this extra computation that is avoided can save some time.

Details

The $\alpha$-transformation is applied to the compositional data first and then multivariate regression is applied. This involves numerical optimisation.

Value

A list including:

runtime The time required by the regression.

be The beta coefficients.

seb The standard error of the beta coefficients.

est The fitted values for xnew if xnew is not NULL.
Regularised and flexible discriminant analysis for compositional data using the alpha-transformation

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References

See Also
alfareg.tune, diri.reg, js.compreg, kl.compreg, ols.compreg, comp.reg

Examples
library(MASS)
x <- as.vector(fgl[1:40, 1])
y <- as.matrix(fgl[1:40, 2:9])
y <- y / rowSums(y)
mod <- alfa.reg(y, x, 0.2)

Description
Regularised and flexible discriminant analysis for compositional data using the α-transformation.

Usage
alfa.rda(xnew, x, ina, a, gam = 1, del = 0)
alfa.fda(xnew, x, ina, a)
**Regularised and flexible discriminant analysis for compositional data using the alpha-transformation**

**Arguments**

- `xnew`: A matrix with the new compositional data whose group is to be predicted. Zeros are allowed, but you must be careful to choose strictly positive values of $\alpha$.
- `x`: A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values of $\alpha$.
- `ina`: A group indicator variable for the available data.
- `a`: The value of $\alpha$ for the $\alpha$-transformation.
- `gam`: This is a number between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.
- `del`: This is a number between 0 and 1. It is the weight of the LDA and QDA.

**Details**

For the `alfa.rda`, the covariance matrix of each group is calculated and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. `gam` is the weight of the pooled covariance matrix and `1-gam` is the weight of the spherical covariance matrix. $Sa = gam \times Sp + (1-gam) \times sp$. Then it is a compromise between LDA and QDA. `del` is the weight of `Sa` and `1-del` the weight of each group covariance group. This function is a wrapper for `alfa.rda`.

For the `alfa.fda` a flexible discriminant analysis is performed. See the R package `fda` for more details.

**Value**

For the `alfa.rda` a list including:

- `prob`: The estimated probabilities of the new data of belonging to each group.
- `scores`: The estimated scores of the new data of each group.
- `est`: The estimated group membership of the new data.

For the `alfa.fda` a list including:

- `mod`: A `fda` object as returned by the command `fda` of the R package `mda`.
- `est`: The estimated group membership of the new data.

**Author(s)**

Michail Tsagris

R implementation and documentation: Michail Tsagris `<mtsagris@uoc.gr>` and Giorgos Athineou `<gioathineou@gmail.com>`
Regularised discriminant analysis for Euclidean data

References


See Also

rda, alfa, alfard.tune

Examples

x <- as.matrix(iris[,1:4])
x <- x / rowSums(x)
ina <- iris[,5]
mod <- alfa.rda(x, x, ina, 0)
table(ina, mod$est)
mod2 <- alfa.fda(x, x, ina, 0)
table(ina, mod2$est)

Description

Regularised discriminant analysis for Euclidean data.

Usage

rda(xnew, x, ina, gam = 1, del = 0)

Arguments

xnew A matrix with the new data whose group is to be predicted. They have to be continuous.
x A matrix with the available data. They have to be continuous.
ing A group indicator variable for the available data.
gam This is a number between 0 and 1. It is the weight of the pooled covariance and the diagonal matrix.
del This is a number between 0 and 1. It is the weight of the LDA and QDA.
Details

The covariance matrix of each group is calculated and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. $\text{gam}$ is the weight of the pooled covariance matrix and $1-\text{gam}$ is the weight of the spherical covariance matrix, $\text{Sa} = \text{gam} \times \text{Sp} + (1-\text{gam}) \times \text{sp}$. Then it is a compromise between LDA and QDA. $\text{del}$ is the weight of $\text{Sa}$ and $1-\text{del}$ the weight of each group covariance group. This function is a wrapper for alfa.rda.

Value

A list including:

- \text{prob} The estimated probabilities of the new data of belonging to each group.
- \text{scores} The estimated scores of the new data of each group.
- \text{est} The estimated group membership of the new data.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

rda.tune, alfa

Examples

```r
x <- as.matrix(iris[, 1:4])
ina <- iris[, 5]
mod <- rda(x, x, ina)
table(ina, mod$est)
```
Ridge regression

Description

Ridge regression.

Usage

ridge.reg(y, x, lambda, B = 1, xnew = NULL)

Arguments

y
A real valued vector. If it contains percentages, the logit transformation is applied.

x
A matrix with the predictor variable(s), they have to be continuous.

lambda
The value of the regularisation parameter \( \lambda \).

B
If B = 1 (default value) no bootstrap is performed. Otherwise bootstrap standard errors are returned.

xnew
If you have new data whose response value you want to predict put it here, otherwise leave it as is.

Details

This is used in the function alfa.ridge. There is also a built-in function available from the MASS library, called lm.ridge.

Value

A list including:

beta
The beta coefficients.

seb
The standard error of the coefficients. If B > 1 the bootstrap standard errors will be returned.

est
The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


Ridge regression plot

See Also

\texttt{ridge.tune, alfa.ridge, ridge.plot}

Examples

\begin{verbatim}
  y <- as.vector(iris[, 1])
  x <- as.matrix(iris[, 2:4])
  mod1 <- ridge.reg(y, x, lambda = 0.1)
  mod2 <- ridge.reg(y, x, lambda = 0)
\end{verbatim}

Description

A plot of the regularised regression coefficients is shown.

Usage

\begin{verbatim}
  ridge.plot(y, x, lambda = seq(0, 5, by = 0.1))
\end{verbatim}

Arguments

- \texttt{y} A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e. strictly within 0 and 1 they are mapped into R using the logit transformation. In any case, they must be continuous only.
- \texttt{x} A numeric matrix containing the continuous variables. Rows are samples and columns are features.
- \texttt{lambda} A grid of values of the regularisation parameter $\lambda$.

Details

For every value of $\lambda$ the coefficients are obtained. They are plotted versus the $\lambda$ values.

Value

A plot with the values of the coefficients as a function of $\lambda$.

Author(s)

Michail Tsagris

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>
Ridge regression with compositional data in the covariates side using the alpha-transformation

References

See Also
ridge.reg, ridge.tune, alfa.ridge, alfaridge.plot

Examples
```r
y <- as.vector(iris[, 1])
x <- as.matrix(iris[, 2:4])
ridge.plot(y, x, lambda = seq(0, 2, by = 0.1) )
```

Description
Ridge regression with compositional data in the covariates side using the \( \alpha \)-transformation.

Usage
afa.ridge(y, x, a, lambda, B = 1, xnew = NULL)

Arguments

- **y**: A numerical vector containing the response variable values. If they are percentages, they are mapped onto \( R \) using the logit transformation.
- **x**: A matrix with the predictor variables, the compositional data. Zero values are allowed, but you must be careful to choose strictly positive \( \alpha \)-values.
- **a**: The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \( \alpha = 0 \) the isometric log-ratio transformation is applied.
- **lambda**: The value of the regularisation parameter, \( \lambda \).
- **B**: If \( B > 1 \) bootstrap estimation of the standard errors is implemented.
- **xnew**: A matrix containing the new compositional data whose response is to be predicted. If you have no new data, leave this NULL as is by default.

Details
The \( \alpha \)-transformation is applied to the compositional data first and then ridge components regression is performed.
Ridge regression with the alpha-transformation plot

Value

The output of the ridge.reg.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

ridge.reg, alfaridge.tune, alfaridge.plot

Examples

library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x/ rowSums(x)
mod1 <- alfa.ridge(y, x, a = 0.5, lambda = 0.1, B = 1, xnew = NULL)
mod2 <- alfa.ridge(y, x, a = 0.5, lambda = 1, B = 1, xnew = NULL)

Ridge regression with the alpha-transformation plot

Ridge regression plot

Description

A plot of the regularised regression coefficients is shown.

Usage

alfaridge.plot(y, x, a, lambda = seq(0, 5, by = 0.1) )
Ridge regression with the alpha-transformation plot

Arguments

y
A numeric vector containing the values of the target variable. If the values are proportions or percentages, i.e., strictly within 0 and 1 they are mapped into R using the logit transformation. In any case, they must be continuous only.

x
A numeric matrix containing the continuous variables.

a
The value of the $\alpha$-transformation. It has to be between -1 and 1. If there are zero values in the data, you must use a strictly positive value.

lambda
A grid of values of the regularisation parameter $\lambda$.

Details

For every value of $\lambda$ the coefficients are obtained. They are plotted versus the $\lambda$ values.

Value

A plot with the values of the coefficients as a function of $\lambda$.

Author(s)

Michail Tsagris

R implementation and documentation: Giorgos Athineou <gioathineou@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>

References


See Also

ridge.plot, alfa.ridge

Examples

library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
alfaridge.plot(y, x, a = 0.5, lambda = seq(0, 5, by = 0.1) )
Description

Simulation of compositional data from Gaussian mixture models.

Usage

rmixcomp(n, prob, mu, sigma, type = "alr")

Arguments

- **n**: The sample size
- **prob**: A vector with mixing probabilities. Its length is equal to the number of clusters.
- **mu**: A matrix where each row corresponds to the mean vector of each cluster.
- **sigma**: An array consisting of the covariance matrix of each cluster.
- **type**: Should the additive ("type=alr") or the isometric (type="ilr") log-ratio be used? The default value is for the additive log-ratio transformation.

Details

A sample from a multivariate Gaussian mixture model is generated.

Value

A list including:

- **id**: A numeric variable indicating the cluster of simulated vector.
- **x**: A matrix containing the simulated compositional data. The number of dimensions will be + 1.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

mix.compnorm,bic.mixcompnorm
Examples

```r
p <- c(1/3, 1/3, 1/3)
mu <- matrix(nrow = 3, ncol = 4)
s <- array(dim = c(4, 4, 3))
x <- as.matrix(iris[, 1:4])
ina <- as.numeric(iris[, 5])
mu <- rowsum(x, ina) / 50
s[, , 1] <- cov(x[ina == 1, ])
s[, , 2] <- cov(x[ina == 2, ])
s[, , 3] <- cov(x[ina == 3, ])
y <- rmixcomp(100, p, mu, s, type = "alr")
```

Description

Simulation of compositional data from the Flexible Dirichlet distribution.

Usage

```r
rfd(n, alpha, prob, tau)
```

Arguments

- `n` The sample size.
- `alpha` A vector of the non-negative \(\alpha\) parameters.
- `prob` A vector of the clusters’ probabilities that must sum to one.
- `tau` The positive scalar \(\tau\) parameter.

Details

For more information see the references.

Value

A matrix with compositional data.

Author(s)

Michail Tsagris ported from the R package FlexDir. `<mtsagris@uoc.gr>`.
Simulation of compositional data from the folded normal distribution

References


See Also

fd.est, fd.density

Examples

alpha <- c(12, 11, 10)
prob <- c(0.25, 0.25, 0.5)
x <- rfd(100, alpha, prob, 7)

Simulation of compositional data from the folded normal distribution

Description

Simulation of compositional data from the folded model normal distribution.

Usage

rfolded(n, mu, su, a)

Arguments

n The sample size.
mu The mean vector.
su The covariance matrix.
a The value of \( \alpha \).

Details

A sample from the folded model is generated.

Value

A matrix with compositional data.
Spatial median regression

Author(s)

Michail Tsagris.
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

alfa, alpha.mle, a.est

Examples

s <- c(0.1490676523, -0.4580818209, 0.0020395316, -0.0047446076, -0.4580818209,
1.5227259250, 0.0002596411, 0.0074836251, 0.0020395316, 0.0002596411,
0.0365384838, -0.0471448849, -0.0047446076, 0.0074836251, -0.0471448849,
0.0611442781)
s <- matrix(s, ncol = 4)
m = c(1.715, 0.914, 0.115, 0.167)
x = rfolded(100, m, s, 0.5)
a.est(x)

spatmed.reg(y, x, xnew = NULL, tol = 1e-07, ses = FALSE)

Arguments

y A matrix with the compositional data. Zero values are not allowed.
x The predictor variable(s), they have to be continuous.
xnew If you have new data use it, otherwise leave it NULL.
tol The threshold upon which to stop the iterations of the Newton-Rapshon algorithm.

ses If you want to extract the standard errors of the parameters, set this to TRUE. Be careful though as this can slow down the algorithm dramatically. In a run example with 10,000 observations and 10 variables for y and 30 for x, when ses = FALSE the algorithm can take 0.20 seconds, but when ses = TRUE it can go up to 140 seconds.
Details

The objective function is the minimization of the sum of the absolute residuals. It is the multivariate generalization of the median regression. This function is used by \texttt{comp.reg}.

Value

A list including:

- \texttt{iter}: The number of iterations that were required.
- \texttt{runtime}: The time required by the regression.
- \texttt{be}: The beta coefficients.
- \texttt{seb}: The standard error of the beta coefficients is returned if \texttt{ses=}TRUE and NULL otherwise.
- \texttt{est}: The fitted of \texttt{xnew} if \texttt{xnew} is not NULL.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris \texttt{<mtsagris@uoc.gr>} and Giorgos Athineou \texttt{<gioathineou@gmail.com>}

References


See Also

\texttt{multivreg, comp.reg, alfa.reg, js.compreg, diri.reg}

Examples

```r
library(MASS)
x <- as.matrix(iris[, 3:4])
y <- as.matrix(iris[, 1:2])
mod1 <- spatmed.reg(y, x)
mod2 <- multivreg(y, x, plot = FALSE)
```
Ternary diagram

Description

Ternary diagram.

Usage

\texttt{ternary(x, means = TRUE, pca = FALSE, colour = NULL)}

Arguments

- \texttt{x}: A matrix with the compositional data.
- \texttt{means}: A boolean variable. Should the closed geometric mean and the arithmetic mean appear (\text{TRUE}) or not (\text{FALSE})?
- \texttt{pca}: Should the first PCA calculated by Aitchison (1983) described appear? If yes, then this should be \text{TRUE}, or \text{FALSE} otherwise.
- \texttt{colour}: If you want the points to appear in different colour put a vector with the colour numbers or colours.

Details

The first PCA is calculated using the centred log-ratio transformation as Aitchison (1983, 1986) suggested. If the data contain zero values, the first PCA will not be plotted. There are two ways to create a ternary graph. The one I used here, where each edge is equal to 1 and the one Aitchison (1986) uses. For every given point, the sum of the distances from the edges is equal to 1. Zeros in the data appear with green circles in the triangle and you will also see NaN in the closed geometric mean.

Value

The ternary plot and a 2-row matrix with the means. The closed geometric and the simple arithmetic mean vector and or the first principal component will appear as well if the user has asked for them.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References

Ternary diagram of regression models

See Also

comp.den, alfa, diri.contour, comp.kerncontour

Examples

```r
x <- as.matrix(iris[, 1:3])
x <- x / rowSums(x)
ternary(x, means = TRUE, pca = TRUE)
```

Description

Ternary diagram of regression models.

Usage

```r
ternary.reg(y, est, id, labs)
```

Arguments

- `y`: A matrix with the compositional data.
- `est`: A matrix with all fitted compositional data for all regression models, one under the other.
- `id`: A vector indicating the regression model of each fitted compositional data set.
- `labs`: The names of the regression models to appear in the legend.

Details

The points first appear on the ternary plot. Then, the fitted compositional data appear with different lines for each regression model.

Value

The ternary plot and lines for the fitted values of each regression model.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
ternary, diri.contour, comp.kerncontour

Examples
x <- cbind(1, rnorm(50) )
a <- exp( x %*% matrix( rnorm(6,0,0.4), ncol = 3) )
y <- matrix(NA, 50, 3)
for (i in 1:50) y[i, ] <- rdiri(1, a[i, ])
est <- comp.reg(y, x[, -1], xnew = x[, -1])$est
ternary.reg(y, est, id = rep(1, 50), labs = “ALR regression”)

The additive log-ratio transformation and its inverse
The additive log-ratio transformation and its inverse

Description
The additive log-ratio transformation and its inverse.

Usage
alr(x)
alrinv(y)

Arguments
x A numerical matrix with the compositional data.
y A numerical matrix with data to be closed into the simplex.

Details
The additive log-ratio transformation with the first component being the common divisor is applied.
The inverse of this transformation is also available.

Value
A matrix with the alr transformed data (if alr is used) or with the compositional data (if the alrinv is used).

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References
The alpha-distance

See Also

alfa, \ link{alfainv} alfa.profile, alfa.tune

Examples

library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- alr(x)
x¹ <- alrinv(y)

---

The alpha-distance  The α-distance

Description

This is the Euclidean (or Manhattan) distance after the α-transformation has been applied.

Usage

alfadist(x, a, type = "euclidean", square = FALSE)
alfadista(xnew, x, a, type = "euclidean", square = FALSE)

Arguments

xnew  A matrix or a vector with new compositional data.
x  A matrix with the compositional data.
a  The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $α = 0$, the isometric log-ratio transformation is applied.
type  Which type distance do you want to calculate after the α-transformation, "euclidean", or "manhattan".
square  In the case of the Euclidean distance, you can choose to return the squared distance by setting this TRUE.

Details

The α-transformation is applied to the compositional data first and then the Euclidean or the Manhattan distance is calculated.

Value

For "alfadist" a matrix including the pairwise distances of all observations or the distances between xnew and x. For "alfadista" a matrix including the pairwise distances of all observations or the distances between xnew and x.
The alpha-k-NN regression for compositional response data

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
alfa, alfainv, alfa.reg, esov

Examples
library(MASS)
x <- as.matrix(fgl[1:20, 2:9])
x <- x / rowSums(x)
alfadist(x, 0.1)
alfadist(x, 1)

The alpha-k-NN regression for compositional response data

Description
The $\alpha$-k-NN regression for compositional response data.

Usage
aknn.reg(xnew, y, x, a = seq(0.1, 1, by = 0.1), k = 2:10,
apostasi = "euclidean", rann = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xnew</td>
<td>A matrix with the new predictor variables whose compositions are to be predicted.</td>
</tr>
<tr>
<td>y</td>
<td>A matrix with the compositional response data. Zeros are allowed.</td>
</tr>
<tr>
<td>x</td>
<td>A matrix with the available predictor variables.</td>
</tr>
<tr>
<td>a</td>
<td>The value of $\alpha$. As zero values in the compositional data are allowed, you must be careful to choose strictly positive values of $\alpha$. If negative values are passed, the positive ones are used only.</td>
</tr>
</tbody>
</table>
The alpha-k-NN regression with compositional predictor variables

k
The number of nearest neighbours to consider. It can be a single number or a vector.

apostasi
The type of distance to use, either "euclidean" or "manhattan".

rann
If you have large scale datasets and want a faster k-NN search, you can use k-d trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details
The \( \alpha \)-k-NN regression for compositional response variables is applied.

Value
A list with the estimated compositional response data for each value of \( \alpha \) and k.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
alfa.reg, comp.ppr, comp.reg, kl.compreg

Examples
```
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
x <- iris[, 4]
mod <- aknn.reg(x, y, x, a = c(0.4, 0.5), k = 2:3, apostasi = "euclidean")
```

The alpha-k-NN regression with compositional predictor variables

Description
The \( \alpha \)-k-NN regression with compositional predictor variables.

Usage
```
alfa.knn.reg(xnew, y, x, a = 1, k = 2:10, apostasi = "euclidean", method = "average")
```
The alpha-k-NN regression with compositional predictor variables

Arguments

xnew  A matrix with the new compositional predictor variables whose response is to be predicted. Zeros are allowed.
y  The response variable, a numerical vector.
x  A matrix with the available compositional predictor variables. Zeros are allowed.
a  A single value of $\alpha$. As zero values in the compositional data are allowed, you must be careful to choose strictly positive values of $\alpha$. If negative values are passed, the positive ones are used only. If the data are already alpha-transformed, you can make this NULL.
k  The number of nearest neighbours to consider. It can be a single number or a vector.
apostasi  The type of distance to use, either "euclidean" or "manhattan".
method  If you want to take the average of the responses of the k closest observations, type "average". For the median, type "median" and for the harmonic mean, type "harmonic".

Details

The $\alpha$-k-NN regression with compositional predictor variables is applied.

Value

A matrix with the estimated response data for each value of k.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

aknn.reg, alfa.knn, alfa.pcr, alfa.ridge

Examples

library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y <- fgl[, 1]
mod <- alfa.knn.reg(x, y, x, a = 0.5, k = 2:4)
The alpha-transformation

Description
The $\alpha$-transformation.

Usage

alfa(x, a, h = TRUE)
alef(x, a)

Arguments

x A matrix with the compositional data.
a The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
h A boolean variable. If is TRUE (default value) the multiplication with the Helmert sub-matrix will take place. When $\alpha = 0$ and $h = \text{FALSE}$, the result is the centred log-ratio transformation (Aitchison, 1986). In general, when $h = \text{FALSE}$ the resulting transformation maps the data onto a singular space. The sum of the vectors is equal to 0. Hence, from the simplex constraint the data go to another constraint.

Details
The $\alpha$-transformation is applied to the compositional data. The command "alef" is the same as "alfa(x, a, h = FALSE)", but returns a different element as well and is necessary for the functions a.est, a.mle and alpha.mle.

Value

A list including:
sa The logarithm of the Jacobian determinant of the $\alpha$-transformation. This is used in the "profile" function to speed up the computations.
sk If the "alef" was called, this will return the sum of the $\alpha$-power transformed data, prior to being normalised to sum to 1. If $\alpha = 0$, this will not be returned.
aff The $\alpha$-transformed data.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>
The ESOV-distance

References


See Also

alfainv, alfa.profile, alfa.tune a.est, alpha.mle

Examples

library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
y1 <- alfa(x, 0.2)$aff
y2 <- alfa(x, 1)$aff
rbind( colMeans(y1), colMeans(y2) )
y3 <- alfa(x, 0.2)$aff
dim(y1) ; dim(y3)
rowSums(y1)
rowSums(y3)

The ESOV-distance

Description

The ESOV-distance.

Usage

esov(x)
esova(xnew, x)
es(x1, x2)

Arguments

x A matrix with compositional data.
xnew A matrix or a vector with new compositional data.
x1 A vector with compositional data.
x2 A vector with compositional data.
Details

The ESOV distance is calculated.

Value

For "esov" a matrix including the pairwise distances of all observations or the distances between xnew and x. For "esova" a matrix including the pairwise distances of all observations or the distances between xnew and x. For "es" a number, the ESOV distance between x1 and x2.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

alfadist, comp.knn, js.compreg

Examples

library(MASS)
x <- as.matrix(fgl[1:20, 2:9])
x <- x / rowSums(x)
esov(x)

frechet(x, a)
The Frechet mean for compositional data

Arguments

- **x**: A matrix with the compositional data.
- **a**: The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \( \alpha = 0 \) the isometric log-ratio transformation is applied and the closed geometric mean is calculated. You can also provide a sequence of values of alpha and in this case a matrix of Frechet means will be returned.

Details

The power transformation is applied to the compositional data and the mean vector is calculated. Then the inverse of it is calculated and the inverse of the power transformation applied to the last vector is the Frechet mean.

Value

If \( \alpha \) is a single value, the function will return a vector with the Frechet mean for the given value of \( \alpha \). Otherwise the function will return a matrix with the Frechet means for each value of \( \alpha \).

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

alfa,alfainv,profile

Examples

```r
library(MASS)
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
frechet(x, 0.2)
frechet(x, 1)
```
The Helmert sub-matrix

Description
The Helmert sub-matrix.

Usage
helm(n)

Arguments
n  A number greater than or equal to 2.

Details
The Helmert sub-matrix is returned. It is an orthogonal matrix without the first row.

Value
A \((n - 1) \times n\) matrix.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>.

References


See Also
alfa, alfainv

Examples
helm(3)
helm(5)
The k-nearest neighbours using the alpha-distance

Description
The k-nearest neighbours using the alpha-distance.

Usage
alfann(xnew, x, a, k = 10, rann = FALSE)

Arguments
xnew A matrix or a vector with new compositional data.
x A matrix with the compositional data.
a The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$, the isometric log-ratio transformation is applied.
k The number of nearest neighbours to search for.
rann If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details
The $\alpha$-transformation is applied to the compositional data first and the indices of the k-nearest neighbours using the Euclidean distance are returned.

Value
A matrix including the indices of the nearest neighbours of each xnew from x.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

See Also
alfa.knn,aknn.reg,alfa,alfainv
The k-NN algorithm for compositional data

Examples

```r
library(MASS)
xnew <- as.matrix(fgl[1:20, 2:9])
xnew <- xnew / rowSums(xnew)
x <- as.matrix(fgl[-c(1:20), 2:9])
x <- x / rowSums(x)
b <- alfann(xnew, x, a = 0.1, k = 10)
```

Description

The k-NN algorithm for compositional data with and without using the power transformation.

Usage

```r
comp.knn(xnew, x, ina, a = 1, k = 5, type = "S", apostasi = "ESOV", mesos = TRUE)
```

```r
alfa.knn(xnew, x, ina, a = 1, k = 5, type = "S", mesos = TRUE, apostasi = "euclidean", rann = FALSE)
```

Arguments

- `xnew`: A matrix with the new compositional data whose group is to be predicted. Zeros are allowed, but you must be careful to choose strictly positive values of \( \alpha \) or not to set `apostasi = "Ait"`. 
- `x`: A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values of \( \alpha \) or not to set `apostasi = "Ait"`. 
- `ina`: A group indicator variable for the available data. 
- `a`: The value of \( \alpha \). As zero values in the compositional data are allowed, you must be careful to choose strictly positive values of \( \alpha \). You have the option to put `a = NULL`. In this case, the `xnew` and `x` are assumed to be the already \( \alpha \)-transformed data. 
- `k`: The number of nearest neighbours to consider. It can be a single number or a vector. 
- `type`: This can be either "S" for the standard k-NN or "NS" for the non standard (see details). 
- `apostasi`: The type of distance to use. For the `comp.knn` this can be one of the following: "ESOV", "taxicab", "Ait", "Hellinger", "angular" or "CS". See the references for them. For the `alfa.knn` this can be either "euclidean" or "manhattan". 
- `mesos`: This is used in the non standard algorithm. If `TRUE`, the arithmetic mean of the distances is calculated, otherwise the harmonic mean is used (see details).
The k-NN algorithm for compositional data

If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

Details

The k-NN algorithm is applied for the compositional data. There are many metrics and possibilities to choose from. The standard algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours. The non standard algorithm is slower but perhaps more accurate. For every group is finds the k nearest neighbours to the new observation. It then computes the arithmetic or the harmonic mean of the distances. The new point is allocated to the class with the minimum distance.

Value

A vector with the estimated groups.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <m.t.sagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

compknn.tune,rda,alfa,esov

Examples

x <- as.matrix( iris[, 1:4] )
x <- x/ rowSums(x)
ing <- iris[, 5]
Total variability

mod <- comp.knn(x, x, ina, a = 1, k = 5)
table(ina, mod)
mod2 <- alfa.knn(x, x, ina, a = 1, k = 5)
table(ina, mod2)

Description

Total variability.

Usage

totvar(x, a = 0)

Arguments

x A numerical matrix with the compositional data.
a The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the centred log-ratio transformation is used.

Details

The $\alpha$-transformation is applied and the sum of the variances of the transformed variables is calculated. This is the total variability. Aitchison (1986) used the centred log-ratio transformation, but we have extended it to cover more geometries, via the $\alpha$-transformation.

Value

The total variability of the data in a given geometry as dictated by the value of $\alpha$.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

alfa, \link{alfainv,} alfa.profile, alfa.tune
Examples

```r
x <- as.matrix(iris[, 1:4])
x <- x / rowSums(x)
totvar(x)
```

Description

Transformation-free linear regression for compositional responses and predictors.

Usage

```r
tflr(y, x, xnew = NULL)
```

Arguments

- `y`: A matrix with the compositional response. Zero values are allowed.
- `x`: A matrix with the compositional predictors. Zero values are allowed.
- `xnew`: If you have new data use it, otherwise leave it NULL.

Details

The transformation-free linear regression for compositional responses and predictors is implemented.

Value

A list including:

- `runtime`: The time required by the regression.
- `loglik`: The log-likelihood $-\sum_{i=1}^{n} y_i \log \hat{y}_i / \hat{y}_i$.
- `be`: The beta coefficients.
- `est`: The fitted values of xnew if xnew is not NULL.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tuning of the alpha generalised correlations between two compositional datasets

See Also

cv.tflr, ols.compcomp, kl.alfapcr

Examples

library(MASS)
y <- rdiri(214, runif(3, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- tflr(y, x, x)
mod

Tuning of the alpha generalised correlations between two compositional datasets

Tuning of the alpha generalised correlations between two compositional datasets

Description

Tuning of the alpha generalised correlations between two compositional datasets.

Usage

acor.tune(y, x, a, type = "dcor")

Arguments

y A matrix with the compositional data.
x A matrix with the compositional data.
a The range of values of the power transformation to search for the optimal one. If zero values are present it has to be greater than 0.
type the type of correlation to compute, the distance correlation ("edist"), the canonical correlation type 1 ("cancor1") or the canonical correlation type 2 ("cancor2"). See details for more information.

Details

The $\alpha$-transformation is applied to each composition and then the distance correlation or the canonical correlation is computed. If type = "cancor1" the function returns the value of $\alpha$ that maximizes the product of the eigenvalues. If type = "cancor2" the function returns the value of $\alpha$ that maximizes the the largest eigenvalue.

Value

A list including:

alpha The optimal value of $\alpha$.
acor The maximum value of the acor.
runtime The runtime of the optimization.
Tuning of the bandwidth \( h \) of the kernel using the maximum likelihood cross validation

**Author(s)**
Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

**See Also**

acor, alfa.profile, alfa, alfainv

**Examples**

```r
y <- rdiri(30, runif(3) )
x <- rdiri(30, runif(4) )
acor(y, x, a = 0.4)
```

**Description**

Tuning of the bandwidth \( h \) of the kernel using the maximum likelihood cross validation.

**Usage**

```r
mkde.tune( x, low = 0.1, up = 3, s = cov(x) )
```

**Arguments**

- `x`: A matrix with Euclidean (continuous) data.
- `low`: The minimum value to search for the optimal bandwidth value.
- `up`: The maximum value to search for the optimal bandwidth value.
- `s`: A covariance matrix. By default it is equal to the covariance matrix of the data, but can change to a robust covariance matrix, MCD for example.

**Details**

Maximum likelihood cross validation is applied in order to choose the optimal value of the bandwidth parameter. No plot is produced.

**Value**

A list including:

- `hopt`: The optimal bandwidth value.
- `maximum`: The value of the pseudo-log-likelihood at that given bandwidth value.
Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation

**Author(s)**
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

**References**

**See Also**
mkde, comp.kerncontour

**Examples**
```r
library(MASS)
mkde.tune(as.matrix(iris[, 1:4]), c(0.1, 3) )
```

**Description**
Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the $\alpha$-transformation.

**Usage**
```r
klalfapcr.tune(y, x, covar = NULL, nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1),
folds = NULL, graph = FALSE, tol = 1e-07, maxiters = 50, seed = FALSE)
```

**Arguments**
- `y` A numerical matrix with compositional data with or without zeros.
- `x` A matrix with the predictor variables, the compositional data. Zero values are allowed.
- `covar` If you have other continuous covariates put them here.
- `nfolds` The number of folds for the K-fold cross validation, set to 10 by default.
- `maxk` The maximum number of principal components to check.
Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation

\( \alpha \)  
The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \( \alpha = 0 \) the isometric log-ratio transformation is applied.

folds  
If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

graph  
If graph is TRUE (default value) a filled contour plot will appear.

tol  
The tolerance value to terminate the Newton-Raphson procedure.

maxiters  
The maximum number of Newton-Raphson iterations.

seed  
If seed is TRUE the results will always be the same.

Details

The M-fold cross validation is performed in order to select the optimal values for \( \alpha \) and k, the number of principal components. The \( \alpha \)-transformation is applied to the compositional data first, the first k principal component scores are calculated and used as predictor variables for the Kullback-Leibler divergence based regression model. This procedure is performed M times during the M-fold cross validation.

Value

A list including:

mspe  
A list with the KL divergence for each value of \( \alpha \) and k in every fold.

performance  
A matrix with the KL divergence for each value of \( \alpha \) averaged over all folds. If graph is set to TRUE this matrix is plotted.

best.perf  
The minimum KL divergence.

params  
The values of \( \alpha \) and k corresponding to the minimum KL divergence.

Author(s)

Initial code by Abdulaziz Alenazi. Modifications by Michail Tsagris.

R implementation and documentation: Abdulaziz Alenazi <a.alenazi@nbu.edu.sa> Michail Tsagris <mtsagris@uoc.gr>

References


Tuning of the k-NN algorithm for compositional data

See Also

kl.alfapcr, cv.tflr, pcr, glm.pcr, alfapcr.tune

Examples

library(MASS)
y <- rdiri(214, runif(4, 1, 3))
x <- as.matrix(fgl[, 2:9])
x <- x / rowSums(x)
mod <- klalfapcr.tune(y = y, x = x, a = c(0.7, 0.8))
mod

Description

Tuning of the k-NN algorithm for compositional data with and without using the power or the \( \alpha \)-transformation. In addition, estimation of the rate of correct classification via M-fold cross-validation.

Usage

compknn.tune(x, ina, nfolds = 10, k = 2:5, type = "S", mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "ESOV", folds = NULL,
stratified = FALSE, seed = FALSE, graph = FALSE)

alfaknn.tune(x, ina, nfolds = 10, k = 2:5, type = "S", mesos = TRUE,
a = seq(-1, 1, by = 0.1), apostasi = "euclidean", rann = FALSE, folds = NULL,
stratified = FALSE, seed = FALSE, graph = FALSE)

Arguments

x A matrix with the available compositional data. Zeros are allowed, but you must be careful to choose strictly positive values of \( \alpha \) or not to set apostasi = "Ait".

ina A group indicator variable for the available data.
nfolds The number of folds to be used. This is taken into consideration only if the folds argument is not supplied.
k A vector with the nearest neighbours to consider.
type This can be either "S" for the standard k-NN or "NS" for the non standard (see details).
mesos This is used in the non standard algorithm. If TRUE, the arithmetic mean of the distances is calculated, otherwise the harmonic mean is used (see details).
a A grid of values of \( \alpha \) to be used only if the distance chosen allows for it.
Tuning of the k-NN algorithm for compositional data

apostasi The type of distance to use. For the comp.knn this can be one of the following: "ESOV", "taxicab", "Ait", "Hellinger", "angular" or "CS". See the references for them. For the alfa.knn this can be either "euclidean" or "manhattan".

ran If you have large scale datasets and want a faster k-NN search, you can use kd-trees implemented in the R package "RANN". In this case you must set this argument equal to TRUE. Note however, that in this case, the only available distance is by default "euclidean".

folds If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.

stratified Do you want the folds to be created in a stratified way? TRUE or FALSE.

seed If seed is TRUE the results will always be the same.

graph If set to TRUE a graph with the results will appear.

Details

The k-NN algorithm is applied for the compositional data. There are many metrics and possibilities to choose from. The standard algorithm finds the k nearest observations to a new observation and allocates it to the class which appears most times in the neighbours. The non standard algorithm is slower but perhaps more accurate. For every group is finds the k nearest neighbours to the new observation. It then computes the arithmetic or the harmonic mean of the distances. The new point is allocated to the class with the minimum distance.

Value

A list including:

ela A matrix or a vector (depending on the distance chosen) with the averaged over all folds rates of correct classification for all hyper-parameters ($\alpha$ and k).

performance The estimated rate of correct classification.

best_a The best value of $\alpha$. This is returned for "ESOV" and "taxicab" only.

best_k The best number of nearest neighbours.

runtime The run time of the cross-validation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

comp.knn,rda,alfa

Examples

```r
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
ina <- iris[, 5]
mod1 <- compknn.tune(x, ina, a = seq(1, 1, by = 0.1) )
mod2 <- alfaknn.tune(x, ina, a = seq(-1, 1, by = 0.1) )
```

Tuning of the projection pursuit regression for compositional data

Description

Tuning of the projection pursuit regression for compositional data In addition, estimation of the rate of correct classification via K-fold cross-validation.

Usage

```r
compppr.tune(y, x, nfolds = 10, folds = NULL, seed = FALSE, nterms = 1:10,
type = "alr", yb = NULL, B = 1000 )
```

Arguments

- `y`: A matrix with the available compositional data, but zeros are not allowed.
- `x`: A matrix with the continuous predictor variables.
- `nfolds`: The number of folds to use.
- `folds`: If you have the list with the folds supply it here.
- `seed`: If seed is TRUE the results will always be the same.
- `nterms`: The number of terms to try in the projection pursuit regression.
Tuning of the projection pursuit regression for compositional data

**type**

Either "alr" or "ilr" corresponding to the additive or the isometric log-ratio transformation respectively.

**yb**

If you have already transformed the data using a log-ratio transformation put it here. Otherwise leave it NULL.

**B**

The number of bootstrap re-samples to use for the unbiased estimation of the performance of the projection pursuit regression. If $B = 1$, no bootstrap is applied.

**Details**

The function performs tuning of the projection pursuit regression algorithm.

**Value**

A list including:

- $k_1$ The average Kullback-Leibler divergence.
- $bc\_perf$ The bootstrap bias corrected average Kullback-Leibler divergence. If no bootstrap was performed this is equal to the average Kullback-Leibler divergence.
- runtime The run time of the cross-validation procedure.

**Author(s)**

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

**References**


**See Also**

comp.ppr, comp.reg, alfa

**Examples**

```r
x <- as.matrix(iris[, 1:4])
x <- x/ rowSums(x)
ina <- iris[, 5]
mod <- compknn.tune(x, ina, a = seq(1, 1, by = 0.1) )
```
Tuning the number of PCs in the PCR with compositional data using the alpha-transformation

Description

This is a cross-validation procedure to decide on the number of principal components when using regression with compositional data (as predictor variables) using the \( \alpha \)-transformation.

Usage

```r
alfapcr.tune(y, x, model = "gaussian", nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1),
folds = NULL, ncores = 1, graph = TRUE, col.nu = 15, seed = FALSE)
```

Arguments

- \( y \) A vector with either continuous, binary or count data.
- \( x \) A matrix with the predictor variables, the compositional data. Zero values are allowed.
- \( model \) The type of regression model to fit. The possible values are "gaussian", "binomial" and "poisson".
- \( nfolds \) The number of folds for the K-fold cross validation, set to 10 by default.
- \( maxk \) The maximum number of principal components to check.
- \( a \) The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If \( \alpha = 0 \) the isometric log-ratio transformation is applied and the solution exists in a closed form, since it the classical mutivariate regression. The estimated bias correction via the (Tibshirani and Tibshirani (2009) criterion is applied.
- \( folds \) If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- \( ncores \) How many cores to use. If you have heavy computations or do not want to wait for long time more than 1 core (if available) is suggested. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down th process.
- \( graph \) If graph is TRUE (default value) a filled contour plot will appear.
- \( col.nu \) A number parameter for the filled contour plot, taken into account only if graph is TRUE.
- \( seed \) If seed is TRUE the results will always be the same.

Details

The \( \alpha \)-transformation is applied to the compositional data first and the function "pcr.tune" or "glm-pcr.tune" is called.
Value

If graph is TRUE a fileld contour a filled contour will appear. A list including:

- `mspe`: The MSPE where rows correspond to the $\alpha$ values and the columns to the number of principal components.
- `best.par`: The best pair of $\alpha$ and number of principal components.
- `performance`: The minimum mean squared error of prediction.
- `runtime`: The time required by the cross-validation procedure.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

`alfa`, `profile`, `alfa.pcr`, `pcr.tune`, `glmpcr.tune`, `glm`

Examples

```r
library(MASS)
y <- as.vector(fgl[, 1])
x <- as.matrix(fgl[, 2:9])
x <- x/rowSums(x)
mod <- alfapcr.tune(y, x, nfolds = 10, maxk = 50, a = seq(-1, 1, by = 0.1))
```

Description

Tuning the parameters of the regularised discriminant analysis for Euclidean data.
Tuning the parameters of the regularised discriminant analysis

Usage

```
  rda.tune(x, ina, nfolds = 10, gam = seq(0, 1, by = 0.1), del = seq(0, 1, by = 0.1),
  ncores = 1, folds = NULL, stratified = TRUE, seed = FALSE)
```

Arguments

- `x`: A matrix with the data.
- `ina`: A group indicator variable for the available data.
- `nfolds`: The number of folds in the cross validation.
- `gam`: A grid of values for the \( \gamma \) parameter as defined in Tsagris et al. (2016).
- `del`: A grid of values for the \( \delta \) parameter as defined in Tsagris et al. (2016).
- `ncores`: The number of cores to use. If more than 1, parallel computing will take place.
  It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.
- `folds`: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `stratified`: Do you want the folds to be created in a stratified way? TRUE or FALSE.
- `seed`: If seed is TRUE the results will always be the same.

Details

Cross validation is performed to select the optimal parameters for the regularised discriminant analysis and also estimate the rate of accuracy.

The covariance matrix of each group is calculated and then the pooled covariance matrix. The spherical covariance matrix consists of the average of the pooled variances in its diagonal and zeros in the off-diagonal elements. \( \gamma \) is the weight of the pooled covariance matrix and \( 1-\gamma \) is the weight of the spherical covariance matrix, \( S_a = \gamma \cdot S_p + (1-\gamma) \cdot s_p \). Then it is a compromise between LDA and QDA. \( \delta \) is the weight of \( S_a \) and \( 1-\delta \) the weight of each group covariance group. This function is a wrapper for `alfa.rda`.

Value

A list including: If graph is TRUE a plot of the performance versus the number of principal components will appear.

- `per`: An array with the estimate rate of correct classification for every fold. For each of the M matrices, the row values correspond to \( \gamma \) and the columns to the \( \delta \) parameter.
- `percent`: A matrix with the mean estimated rates of correct classification. The row values correspond to \( \gamma \) and the columns to the \( \delta \) parameter.
- `se`: A matrix with the standard error of the mean estimated rates of correct classification. The row values correspond to \( \gamma \) and the columns to the \( \delta \) parameter.
- `result`: The estimated rate of correct classification along with the best \( \gamma \) and \( \delta \) parameters.
- `runtime`: The time required by the cross-validation procedure.
Tuning the principal components with GLMs

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also

rda, alfa

Examples

mod <- rda.tune(as.matrix(iris[, 1:4]), iris[, 5], gam = seq(0, 1, by = 0.2), del = seq(0, 1, by = 0.2) )
mod

Tuning the principal components with GLMs

Tuning the principal components with GLMs

Description

Tuning the number of principal components in the generalised linear models.

Usage

pcr.tune(y, x, nfolds = 10, maxk = 50, folds = NULL, ncores = 1, seed = FALSE, graph = TRUE)
glmpcr.tune(y, x, nfolds = 10, maxk = 10, folds = NULL, ncores = 1, seed = FALSE, graph = TRUE)
multinompcr.tune(y, x, nfolds = 10, maxk = 10, folds = NULL, ncores = 1, seed = FALSE, graph = TRUE)
Arguments

\(y\) \hspace{1cm} \text{A real valued vector for "pcr.tune". A real valued vector for the "glmcr.tune" with either two numbers, 0 and 1 for example, for the binomial regression or with positive discrete numbers for the poisson. For the "multinomcr.tune" a vector or a factor with more than just two values. This is a multinomial regression.}

\(x\) \hspace{1cm} \text{A matrix with the predictor variables, they have to be continuous.}

\(n\text{folds}\) \hspace{1cm} \text{The number of folds in the cross validation.}

\(\text{maxk}\) \hspace{1cm} \text{The maximum number of principal components to check.}

\(\text{folds}\) \hspace{1cm} \text{If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.}

\(\text{ncores}\) \hspace{1cm} \text{The number of cores to use. If more than 1, parallel computing will take place. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.}

\(\text{seed}\) \hspace{1cm} \text{If seed is TRUE the results will always be the same.}

\(\text{graph}\) \hspace{1cm} \text{If graph is TRUE a plot of the performance for each fold along the values of } \alpha \text{ will appear.}

Details

Cross validation is performed to select the optimal number of principal components in the GLMs or the multinomial regression. This is used by \texttt{alfapcr.tune}.

Value

If graph is TRUE a plot of the performance versus the number of principal components will appear. A list including:

\(\text{msp}\) \hspace{1cm} \text{A matrix with the mean deviance of prediction or mean accuracy for every fold.}

\(\text{mpd}\) \hspace{1cm} \text{A vector with the mean deviance of prediction or mean accuracy, each value corresponds to a number of principal components.}

\(k\) \hspace{1cm} \text{The number of principal components which minimizes the deviance or maximises the accuracy.}

\(\text{performance}\) \hspace{1cm} \text{The optimal performance, MSE for the linear regression, minimum deviance for the GLMs and maximum accuracy for the multinomial regression.}

\(\text{runtime}\) \hspace{1cm} \text{The time required by the cross-validation procedure.}

Author(s)

Michail Tsagris

\text{R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>}. 
Tuning the value of alpha in the alpha-regression

References


See Also

`pcr.tune, glm.pcr, alfa.pcr, alfapcr.tune`

Examples

```r
library(MASS)
x <- as.matrix(fgl[, 2:9])
y <- rpois(214, 10)
glm.pcr.tune(y, x, nfolds = 10, maxk = 20, folds = NULL, ncores = 1)
```

---

Tuning the value of alpha in the alpha-regression

Tuning the value of $\alpha$ in the $\alpha$-regression

Description

Tuning the value of $\alpha$ in the $\alpha$-regression.

Usage

```r
alfareg.tune(y, x, a = seq(0.1, 1, by = 0.1), nfolds = 10, folds = NULL, ncores = 1, seed = FALSE, graph = FALSE)
```

Arguments

- `y`: A matrix with compositional data. Zero values are allowed.
- `x`: A matrix with the continuous predictor variables or a data frame including categorical predictor variables.
- `a`: The value of the power transformation, it has to be between -1 and 1. If zero values are present it has to be greater than 0. If $\alpha = 0$ the isometric log-ratio transformation is applied.
- `nfolds`: The number of folds to split the data.
- `folds`: If you have the list with the folds supply it here. You can also leave it NULL and it will create folds.
- `nc`: The number of cores to use. If you have a multicore computer it is advisable to use more than 1. It makes the procedure faster. It is advisable to use it if you have many observations and or many variables, otherwise it will slow down the process.
Tuning the value of alpha in the alpha-regression

seed
If seed is TRUE the results will always be the same.

graph
If graph is TRUE a plot of the performance for each fold along the values of \( \alpha \) will appear.

Details
The \( \alpha \)-transformation is applied to the compositional data and the numerical optimisation is performed for the regression, unless \( \alpha = 0 \), where the coefficients are available in closed form.

Value
A plot of the estimated Kullback-Leibler divergences (multiplied by 2) along the values of \( \alpha \) (if graph is set to TRUE). A list including:

- runtime
  The runtime required by the cross-validation.

- kula
  A matrix with twice the Kullback-Leibler divergence of the observed from the fitted values. Each row corresponds to a fold and each column to a value of \( \alpha \). The average over the columns equal the next argument, "kl".

- kl
  A vector with twice the Kullback-Leibler divergence of the observed from the fitted values. Every value corresponds to a value of \( \alpha \).

- opt
  The optimal value of \( \alpha \).

- value
  The minimum value of twice the Kullback-Leibler.

Author(s)
Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr> and Giorgos Athineou <gioathineou@gmail.com>

References


See Also
alfa.reg, alfa

Examples
library(MASS)
y <- as.matrix(fgl[1:40, 2:4])
y <- y /rowSums(y)
x <- as.vector(fgl[1:40, 1])
mod <- alfareg.tune(y, x, a = c(0.2, 0.35, 0.05), nfolds = 5)
Unit-Weibull regression models for proportions

Description

Unit-Weibull regression models for proportions.

Usage

unitweib.reg(y, x, tau = 0.5)

Arguments

y  A numerical vector proportions. 0s and 1s are allowed.
x  A matrix or a data frame with the predictor variables.
tau The quantile to be used for estimation. The default value is 0.5 yielding the median.

Details

See the reference paper.

Value

A list including:

loglik The loglikelihood of the regression model.
info A matrix with all estimated parameters, their standard error, their Wald-statistic and its associated p-value.

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References


See Also

propreg, beta.reg
Examples

```r
y <- exp(-rweibull(100, 1, 1))
x <- matrix(rnorm(100 * 2), ncol = 2)
a <- unitweib.reg(y, x)
```

Description

Zero adjusted Dirichlet regression.

Usage

```r
zadr(y, x, xnew = NULL, tol = 1e-05)
mixreg(param, z)
```

Arguments

- `y`: A matrix with the compositional data (dependent variable). The number of observations with at least one zero value should not be more than the columns of the predictor variables. Otherwise, the initial values will not be calculated.
- `x`: The predictor variable(s), they can be either continuous or categorical or both.
- `xnew`: If you have new data use it, otherwise leave it NULL.
- `tol`: A tolerance level to terminate the maximisation process.
- `param`: Some arguments passed on to the mixreg helper function.
- `z`: Some arguments passed on to the mixreg helper function.

Details

A zero adjusted Dirichlet regression is being fitted. The likelihood consists of two components. The contributions of the non-zero compositional values and the contributions of the compositional vectors with at least one zero value. The second component may have many different sub-categories, one for each pattern of zeros. The function "mixreg" is a helper function and is not intended to be called directly by the user.

Value

A list including:

- `runtime`: The time required by the regression.
- `loglik`: The value of the log-likelihood.
- `phi`: The precision parameter. If covariates are linked with it (function "diri.reg2"), this will be a vector.
Zero adjusted Dirichlet regression

be  The beta coefficients.
seb The standard error of the beta coefficients.
sigma The covariance matrix of the regression parameters (for the mean vector and the phi parameter) in the function "diri.reg2".
est The fitted or the predicted values (if xnew is not NULL).

Author(s)

Michail Tsagris
R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>

References


See Also

diri.reg, kl.compreg, ols.compreg, alfa.reg

Examples

```r
x <- as.vector(iris[, 4])
y <- as.matrix(iris[, 1:3])
y <- y / rowSums(y)
mod1 <- diri.reg(y, x)
y[sample(1:450, 15)] <- 0
mod2 <- zadr(y, x)
```
Index

* Additive log-ratio-transformation
  The additive log-ratio transformation and its inverse, 122
* Bhattacharyya distance
  Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, 77
* Compositional data
  Compositional-package, 5
* Contour plots
  Compositional-package, 5
* Dirichlet mean vector
  Log-likelihood ratio test for a Dirichlet mean vector, 78
* Dirichlet distribution
  Contour plot of a Dirichlet distribution in S^2, 12
  Contour plot of a Flexible Dirichlet distribution in S^2, 13
  Density values of a Dirichlet distribution, 37
  Dirichlet random values simulation, 38
  Fitting a Dirichlet distribution, 59
  Fitting a Dirichlet distribution via Newton-Raphson, 60
  Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, 77
* Dirichlet regression
  Dirichlet regression, 39
* Discriminant analysis
  Compositional-package, 5
* Equality of the covariance matrices
  Hotelling's multivariate version of the 2 sample t-test for Euclidean data, 71
  Multivariate analysis of variance, 86
* Euclidean distance
  The alpha-distance, 123
  The k-nearest neighbours using the alpha-distance, 132
* Frechet mean
  Helper Frechet mean for compositional data, 67
  The Frechet mean for compositional data, 129
* GLMs
  Quasi binomial regression for proportions, 103
* GLM
  Tuning the principal components with GLMs, 148
* Gaussian mixture models
  Simulation of compositional data from Gaussian mixture models, 115
  Simulation of compositional data from the folded normal distribution, 117
* Gaussian mixtures
  Gaussian mixture models for compositional data, 63
* Gaussianmixture model
  Contour plot of a Gaussian mixture model in S^2, 14
* Hypothesis test
  Empirical likelihood for a one sample mean vector hypothesis testing, 47
  Exponential empirical likelihood for a one sample mean vector hypothesis testing, 55
  Hotelling's multivariate version
of the 1 sample t-test for Euclidean data, 70

* Kullback-Leibler divergence
  Helper functions for the Kullback-Leibler regression, 68

* Kullback-Leibler divergence
  Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, 77

* Log-likelihood ratio test
  Log-likelihood ratio test for a Dirichlet mean vector, 78
  Log-likelihood ratio test for a symmetric Dirichlet distribution, 79

* MLE
  Compositional-package, 5

* Manhattan distance
  The alpha-distance, 123
  The k-nearest neighbours using the alpha-distance, 132

* Multivariate analysis of variance
  Hotelling’s multivariate version of the 2 sample t-test for Euclidean data, 71
  James multivariate version of the t-test, 75
  Multivariate analysis of variance, 86
  Multivariate analysis of variance (James test), 87

* Multivariate hypothesis testing
  Empirical likelihood hypothesis testing for two mean vectors, 48
  Exponential empirical likelihood hypothesis testing for two mean vectors, 56
  Hypothesis testing for two or more compositional mean vectors, 72

* Newton-Raphson
  Fitting a Dirichlet distribution via Newton-Raphson, 60

* No equality of the covariance matrices
  Multivariate analysis of variance (James test), 87

* Non parametric test
  Empirical likelihood hypothesis testing for two mean vectors, 48

* Quasi regression
  Quasi binomial regression for proportions, 103

* Regression
  Compositional-package, 5

* Regularised discriminant analysis
  Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation, 27
  Regularised and flexible discriminant analysis for compositional data using the alpha-transformation, 106
  Regularised discriminant analysis for Euclidean data, 108
  Tuning of the k-NN algorithm for compositional data, 141
  Tuning of the projection pursuit regression for compositional data, 143
  Tuning the parameters of the regularised discriminant analysis, 146

* Ridge regression
  Cross validation for the ridge regression, 29

* Tuning of the hyper-parameters
  Tuning the parameters of the regularised discriminant analysis, 146

* Unequality of the covariance matrices
  James multivariate version of the t-test, 75

* bandwidth tuning
  Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation, 138

* bivariate normal distribution
  Contour plot of the folded model in $S^2$, 17
  Contour plot of the normal distribution in $S^2$, 19
* bivariate skew skewnormal distribution
  Contour plot of the skew skew-normal distribution in $S^2$, 20

* bivariate t distribution
  Contour plot of the t distribution in $S^2$, 21

* compositional data
  Hypothesis testing for two or more compositional mean vectors, 72

* contour plot
  Contour plot of a Dirichlet distribution in $S^2$, 12
  Contour plot of a Flexible Dirichlet distribution in $S^2$, 13
  Contour plot of a Gaussian mixture model in $S^2$, 14
  Contour plot of the folded model in $S^2$, 17
  Contour plot of the kernel density estimate in $S^2$, 18
  Contour plot of the normal distribution in $S^2$, 19
  Contour plot of the skew skew-normal distribution in $S^2$, 20
  Contour plot of the t distribution in $S^2$, 21

* cross-validation
  Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation, 27

* density values
  Density values of a Dirichlet distribution, 37

* fractional response
  Quasi binomial regression for proportions, 103

* inverse transformation
  Inverse of the alpha-transformation, 74

* k-NN algorithm
  Projection pursuit regression for compositional data, 102
  The k-NN algorithm for compositional data, 133

* kernel density estimate
  Multivariate kernel density estimation, 88

* kernel density
  Contour plot of the kernel density estimate in $S^2$, 18
  Tuning of the bandwidth $h$ of the kernel using the maximum likelihood cross validation, 138

* location and scatter
  Estimating location and scatter parameters for compositional data, 50

* maximum likelihood cross validation
  Tuning of the bandwidth $h$ of the kernel using the maximum likelihood cross validation, 138

* maximum likelihood estimation
  Estimation of the value of alpha in the folded model, 52
  Fitting a Dirichlet distribution, 59
  Fitting a Dirichlet distribution via Newton-Rapshon, 60
  MLE of the folded model for a given value of alpha, 84

* maximum log-likelihood estimation
  Fast estimation of the value of alpha, 58

* mean vector
  Empirical likelihood for a one sample mean vector hypothesis testing, 47
  Exponential empirical likelihood for a one sample mean vector hypothesis testing, 55
  Hotelling’s multivariate version of the 1 sample t-test for Euclidean data, 70

* mixtures of normal distributions
  Mixture model selection via BIC, 80

* model based clustering
  Gaussian mixture models for compositional data, 63

* model selection
Mixture model selection via BIC, 80
* multivariate kernel
  Multivariate kernel density estimation, 88
* multivariate linear regression
  Multivariate linear regression, 89
* multivariate normal distribution
  Multivariate normal random values simulation on the simplex, 90
* multivariate regression
  Dirichlet regression, 39
  Helper functions for the Kullback-Leibler regression, 68
  Non linear least squares regression for compositional data, 97
  Regression with compositional data using the alpha-transformation, 105
  Spatial median regression, 118
* multivariate regression
  Multivariate regression with compositional data, 93
* multivariate skew normal distribution
  Multivariate skew normal random values simulation on the simplex, 94
* multivariate t distribution
  MLE for the multivariate t distribution, 82
  Multivariate t random values simulation on the simplex, 95
* non parametric test
  Exponential empirical likelihood hypothesis testing for two mean vectors, 56
  Hypothesis testing for two or more compositional mean vectors, 72
* ordinary least squares
  Non linear least squares regression for compositional data, 97
* orthogonal matrix
  The Helmert sub-matrix, 131
* parameters tuning
  Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, 31
* plot
  Ridge regression plot, 111
  Ridge regression with the alpha-transformation plot, 113
* principal components regression
  Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, 91
  Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, 145
* principal components
  Tuning the principal components with GLMs, 148
* profile log-likelihood
  Estimation of the value of alpha via the profile log-likelihood, 54
* random values simulation
  Dirichlet random values simulation, 38
  Multivariate normal random values simulation on the simplex, 90
  Multivariate skew normal random values simulation on the simplex, 94
  Multivariate t random values simulation on the simplex, 95
* random vectors simulation
  Simulation of compositional data from Gaussian mixture models, 115
  Simulation of compositional data from the folded normal distribution, 117
* regularisation
  Ridge regression, 110
* ridge regression
  Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, 31
  Ridge regression, 110
Ridge regression plot, 111
Ridge regression with compositional data in the covariates side using the alpha-transformation, 112
Ridge regression with the alpha-transformation plot, 113

* robust estimation
Estimating location and scatter parameters for compositional data, 50

* spatial median regression
Multivariate regression with compositional data, 93
Spatial median regression, 118

* symmetric Dirichlet distribution
Log-likelihood ratio test for a symmetric Dirichlet distribution, 79

* ternary plot
Ternary diagram, 120
Ternary diagram of regression models, 121

* tuning
Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, 145

* visualisation
Ternary diagram, 120
Ternary diagram of regression models, 121

a.est, 52, 85, 118, 127, 128
a.est (Estimation of the value of alpha in the folded model), 52
a.mle, 127
a.mle (MLE of the folded model for a given value of alpha), 84
acor, 138
acor (Alpha generalised correlations between two compositional datasets), 8
acor.tune, 8
acor.tune (Tuning of the alpha generalised correlations between two compositional datasets), 137
Aithison’s simple zero replacement strategy, 6
aknn.reg, 67, 126, 132
aknn.reg (The alpha-k-NN regression for compositional response data), 124
aknnreg.tune, 23
aknnreg.tune (Cross validation for the alpha-k-NN regression for compositional response data), 24
alef (The alpha-transformation), 127
alfa (The alpha-transformation), 127
alfa.fda, 25, 27, 29
alfa.fda (Regularised and flexible discriminant analysis for compositional data using the alpha-transformation), 106
alfa.knn, 126, 132
alfa.knn (The k-NN algorithm for compositional data), 133
alfa.knn.reg (The alpha-k-NN regression with compositional predictor variables), 125
alfa.pcr, 11, 100, 101, 126, 146, 150
alfa.pcr (Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation), 91
alfa.profile, 8, 53, 58, 59, 85, 123, 128, 135, 138
alfa.profile (Estimation of the value of alpha via the profile log-likelihood), 54
alfa.rda, 25, 27, 29, 107, 109, 147
alfa.rda (Regularised and flexible discriminant analysis for compositional data using the alpha-transformation), 106
alfa.reg, 40, 98, 119, 124, 125, 151, 154
alfa.reg (Regression with compositional data using the alpha-transformation), 105
alfa.ridge. 32, 110–112, 114, 126
alfa.ridge (Ridge regression with compositional data in the covariates side using the alpha-transformation), 112
alfa.tune, 54, 123, 128, 135
alfa.tune (Fast estimation of the value of alpha), 58
alfadist, 75, 129
alfadist (The alpha-distance), 123
alfadista (The alpha-distance), 123
alfafda.tune (Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation), 27
alfainv, 8, 53, 54, 59, 68, 83, 124, 128, 130–132, 138
alfainv (Inverse of the alpha-transformation), 74
alfaknn.tune (Tuning of the k-NN algorithm for compositional data), 141
alfaknnreg.tune (Cross validation for the alpha-k-NN regression with compositional predictor variables), 26
alfann (The k-nearest neighbours using the alpha-distance), 132
alfapcr.tune, 11, 45, 92, 101, 141, 149, 150
alfapcr.tune (Tuning the number of PCs in the PCR with compositional data using the alpha-transformation), 145
alfarda.tune, 108
alfarda.tune (Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation), 27
alfareg.tune, 93, 105, 106
alfareg.tune (Tuning the value of alpha in the alpha-regression), 150
alfaridge.plot, 112, 113
alfaridge.plot (Ridge regression with the alpha-transformation plot), 113
alfaridge.tune, 30, 113
alfaridge.tune (Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation), 31
All pairwise additive log-ratio transformations, 7
Alpha generalised correlations between two compositional datasets, 8
alpha.mle, 52, 53, 118, 127, 128
alpha.mle (MLE of the folded model for a given value of alpha), 84
alr, 7
alr (The additive log-ratio transformation and its inverse), 122
alr.all (All pairwise additive log-ratio transformations), 7
alrinv (The additive log-ratio transformation and its inverse), 122
anova_propreg, 104

Beta regression, 9
beta.est, 9
beta.est (MLE of distributions defined in the (0, 1) interval), 83
beta.reg, 42, 152
beta.reg (Beta regression), 9
bic.mixcompnorm, 15, 64, 115
bic.mixcompnorm (Mixture model selection via BIC), 80
bivt.contour, 13, 14, 16, 18–21, 82, 83
bivt.contour (Contour plot of the t distribution in S^2), 21

Choose the number of principal components via reconstruction error, 10
choose.pc (Choose the number of principal components via reconstruction error), 10
comp.den, 82, 83, 91, 95, 96, 121
comp.den (Estimating location and scatter parameters for compositional data), 50
comp.kerncontour, 13, 14, 16, 88, 89, 121, 122, 139
comp.kerncontour(Contour plot of the kernel density estimate in $S^2$), 18
comp.knn, 46, 129, 143
comp.knn(The k-NN algorithm for compositional data), 133
comp.ppr, 125, 144
comp.ppr (Projection pursuit regression for compositional data), 102
comp.reg, 23, 40, 43, 69, 90, 98, 106, 119, 125, 144
comp.reg (Multivariate regression with compositional data), 93
comp.test, 48, 49, 56, 57, 71, 72, 76, 86, 87
comp.test (Hypothesis testing for two or more compositional mean vectors), 72
comppknntune, 103, 134
comppknntune (Tuning of the k-NN algorithm for compositional data), 141
Compositional-package, 5
compppr.tune, 23
compppr.tune (Tuning of the projection pursuit regression for compositional data), 143
Constrained linear least squares for compositional responses and predictors, 11
Contour plot of a Dirichlet distribution in $S^2$, 12
Contour plot of a Flexible Dirichlet distribution in $S^2$, 13
Contour plot of a Gaussian mixture model in $S^2$, 14
Contour plot of a Kent distribution in $S^2$, 16
Contour plot of the folded model in $S^2$, 17
Contour plot of the kernel density estimate in $S^2$, 18
Contour plot of the normal distribution in $S^2$, 19
Contour plot of the skew skew-normal distribution in $S^2$, 20
Contour plot of the t distribution in $S^2$, 21
Cross validation for some compositional regression models, 22
Cross validation for the alpha-k-NN regression for compositional response data, 24
Cross validation for the alpha-k-NN regression with compositional predictor variables, 26
Cross validation for the regularised and flexible discriminant analysis with compositional data using the alpha-transformation, 27
Cross validation for the ridge regression, 29
Cross validation for the ridge regression with compositional data as predictor using the alpha-transformation, 31
Cross validation for the transformation-free linear regression for compositional responses and predictors, 32
Cross-validation for the constrained linear least squares for compositional responses and predictors, 34
cv.comp.reg (Cross validation for some compositional regression models), 22
cv.olscompcomp, 12, 33
cv.olscompcomp (Cross-validation for the constrained linear least squares for compositional responses and predictors), 34
cv.tflr, 35, 137, 141
cv.tflr (Cross validation for the transformation-free linear regression for compositional responses and predictors), 32
ddiri, 60, 61, 79
ddiri (Density values of a Dirichlet distribution), 37
Density of the Dirichlet distribution, 35
Density of the Flexible Dirichlet distribution, 36
Density values of a Dirichlet distribution, 37
diri.contour, 15, 18–22, 38, 39, 60, 61, 121, 122
diri.contour (Contour plot of a Dirichlet distribution in $S^2$), 12
diri.density (Density of the Dirichlet distribution), 35
diri.est, 38, 39, 61, 78–80
diri.est (Fitting a Dirichlet distribution), 59
diri.nr, 35, 38, 39, 60, 78–80
diri.nr (Fitting a Dirichlet distribution via Newton-Rapshon), 60
diri.nr2, 84
diri.reg, 9, 43, 69, 90, 94, 98, 106, 119, 154
diri.reg (Dirichlet regression), 39
diri.reg2 (Dirichlet regression), 39
Dirichlet random values simulation, 38
Dirichlet regression, 39
dirimean.test, 80
dirimean.test (Log-likelihood ratio test for a Dirichlet mean vector), 78
Distance based regression models for proportions, 41
divergence (Divergence matrix of compositional data), 46
Divergence based regression for compositional data, 42
Divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation, 44
Divergence matrix of compositional data, 46
eel.test1, 48, 71
eel.test1 (Exponential empirical likelihood for a one sample mean vector hypothesis testing), 55
eel.test2, 49
eel.test2 (Exponential empirical likelihood hypothesis testing for two mean vectors), 56
el, 76
el.test, 47
el.test1, 56, 71
el.test1 (Empirical likelihood for a one sample mean vector hypothesis testing), 47
el.test2, 48, 56, 57, 71, 72, 74
el.test2 (Empirical likelihood hypothesis testing for two mean vectors), 48
Empirical likelihood for a one sample mean vector hypothesis testing, 47
Empirical likelihood hypothesis testing for two mean vectors, 48
es (The ESOV-distance), 128
esov, 124, 134
esov (The ESOV-distance), 128
esova (The ESOV-distance), 128
Estimating location and scatter parameters for compositional data, 50
Estimation of the probability left outside the simplex when using the alpha-transformation, 51
Estimation of the value of alpha in the folded model, 52
Estimation of the value of alpha via the profile log-likelihood, 54
Exponential empirical likelihood for a one sample mean vector hypothesis testing, 55
Exponential empirical likelihood hypothesis testing for two mean vectors, 56
Fast estimation of the value of alpha, 58
fd.contour (Contour plot of a Flexible Dirichlet distribution in $S^2$), 13
fd.density, 35, 62, 117
fd.density (Density of the Flexible Dirichlet distribution), 36
fd.est, 36, 117
fd.est (Fitting a Flexible Dirichlet distribution), 62
Fitting a Dirichlet distribution, 59
INDEX

Fitting a Dirichlet distribution via Newton-Rapshon, 60
Fitting a Flexible Dirichlet distribution, 62
fold.contour (Contour plot of the folded model in S^2), 17
frechet (The Frechet mean for compositional data), 129
frechet2 (Helper Frechet mean for compositional data), 67
Gaussian mixture models for compositional data, 63
Generate random folds for cross-validation, 65
glm, 146
glm.pcr, 45, 92, 141, 150
glm.pcr (Principal component generalised linear models), 100
glm.pcr.tune, 146
glm.pcr.tune (Tuning the principal components with GLMs), 148
green (Greenacre’s power transformation), 66
Greenacre’s power transformation, 66
helling.prop.reg (Distance based regression models for proportions), 41
helm (The Helmert sub-matrix), 131
Helper Frechet mean for compositional data, 67
Helper functions for the Kullback-Leibler regression, 68
hotel1T2, 48, 56
hotel1T2 (Hotelling’s multivariate version of the 1 sample t-test for Euclidean data), 70
hotel2T2, 48, 49, 56, 57, 71, 74, 76, 86, 87
hotel2T2 (Hotelling’s multivariate version of the 2 sample t-test for Euclidean data), 71
Hotelling’s multivariate version of the 1 sample t-test for Euclidean data, 70
Hotelling’s multivariate version of the 2 sample t-test for Euclidean data, 71
hsecant01.est (MLE of distributions defined in the (0, 1) interval), 83
Hypothesis testing for two or more compositional mean vectors, 72
ibeta.est (MLE of distributions defined in the (0, 1) interval), 83
Inverse of the alpha-transformation, 74
james, 48, 49, 56, 57, 71, 72, 86, 87
james (James multivariate version of the t-test), 75
James multivariate version of the t-test, 75
js.compreg, 40, 43, 46, 69, 90, 94, 98, 106, 119, 129
js.compreg (Divergence based regression for compositional data), 42
kent.contour (Contour plot of a Kent distribution in S^2), 16
kl.alfapcr, 12, 137, 141
kl.alfapcr (Divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation), 44
kl.compreg, 23, 40, 69, 90, 98, 106, 125, 154
kl.compreg (Divergence based regression for compositional data), 42
kl.compreg2 (Helper functions for the Kullback-Leibler regression), 68
kl.diri (Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions), 77
klalfapcr.tune, 33, 35, 45
klalfapcr.tune (Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation), 139
klcompreg.boot (Helper functions for the Kullback-Leibler regression), 68
Kullback-Leibler divergence and Bhattacharyya distance between two Dirichlet distributions, 77
kumar.est (MLE of distributions defined in the (0, 1) interval), 83

lm, 89
lm.ridge, 110
Log-likelihood ratio test for a Dirichlet mean vector, 78
Log-likelihood ratio test for a symmetric Dirichlet distribution, 79

logistic_only, 104
logitnorm.est (MLE of distributions defined in the (0, 1) interval), 83

makefolds (Generate random folds for cross-validation), 65
maov, 48, 49, 56, 57, 71, 72, 74, 87
maov (Multivariate analysis of variance), 86
maovjames, 49, 57, 74, 76, 86
maovjames (Multivariate analysis of variance (James test)), 87

mix.compnorm, 15, 81, 115
mix.compnorm (Gaussian mixture models for compositional data), 63
mixnorm.contour, 13, 14, 16, 18–22, 64, 81
mixnorm.contour (Contour plot of a Gaussian mixture model in S^2), 14
mixreg (Zero adjusted Dirichlet regression), 153

Mixture model selection via BIC, 80
mkde, 139
mkde (Multivariate kernel density estimation), 88
mkde.tune, 89
mkde.tune (Tuning of the bandwidth h of the kernel using the maximum likelihood cross validation), 138
MLE for the multivariate t distribution, 82
MLE of distributions defined in the (0, 1) interval, 83

MLE of the folded model for a given value of alpha, 84
multinompcr.tune (Tuning the principal components with GLMs), 148
Multivariate analysis of variance, 86
Multivariate analysis of variance (James test), 87

Multivariate kernel density estimation, 88
Multivariate linear regression, 89
Multivariate normal random values simulation on the simplex, 90
Multivariate or univariate regression with compositional data in the covariates side using the alpha-transformation, 91
Multivariate regression with compositional data, 93
Multivariate skew normal random values simulation on the simplex, 94
Multivariate t random values simulation on the simplex, 95
multivreg, 94, 119
multivt (Multivariate linear regression), 89

multivt (MLE for the multivariate t distribution), 82

Non linear least squares regression for compositional data, 97
norm.contour, 13, 14, 16, 19, 21, 22
norm.contour (Contour plot of the normal distribution in S^2), 19

ols.compcomp, 35, 137
ols.compcomp (Constrained linear least squares for compositional responses and predictors), 11
ols.compreg, 40, 43, 69, 90, 106, 154
ols.compreg (Non linear least squares regression for compositional data), 97
ols.prop.reg (Distance based regression models for proportions), 41

pca, 11, 45, 92, 141
pcr (Principal component generalised linear models), 100
pcr.tune, 146, 150
pcr.tune (Tuning the principal components with GLMs), 148
perturbation, 6, 100
perturbation (Perturbation operation), 98
Perturbation operation, 98
pow (Power operation), 99
power, 99
Principal component generalised linear models, 100
probout (Estimation of the probability left outside the simplex when using the alpha-transformation), 51
profile, 68, 130, 146
Projection pursuit regression for compositional data, 102
prop.reg, 9
propreg, 42, 152
propreg (Quasi binomial regression for proportions), 103
propreg (Quasi binomial regression for proportions), 103
Quasi binomial regression for proportions, 103
rcompnorm, 95, 96
rcompnorm (Multivariate normal random values simulation on the simplex), 90
rcompsn, 91
rcompsn (Multivariate skew normal random values simulation on the simplex), 94
rcompt, 91
rcompt (Multivariate t random values simulation on the simplex), 95
rda, 103, 108, 134, 143, 148
rda (Regularised discriminant analysis for Euclidean data), 108
rda.tune, 25, 27, 29, 65, 109
rda.tune (Tuning the parameters of the regularised discriminant analysis), 146
rdiri, 35, 38, 60, 61, 79, 80, 91, 95, 96
rdiri (Dirichlet random values simulation), 38
Regression with compositional data using the alpha-transformation, 105
Regularised and flexible discriminant analysis for compositional data using the alpha-transformation, 106
Regularised discriminant analysis for Euclidean data, 108
rfd, 36, 62
rfd (Simulation of compositional data from the Flexible Dirichlet distribution), 116
rfolded, 52
rfolded (Simulation of compositional data from the folded normal distribution), 117
Ridge regression, 110
Ridge regression plot, 111
Ridge regression with compositional data in the covariates side using the alpha-transformation, 112
Ridge regression with the alpha-transformation plot, 113
ridge.plot, 111, 114
ridge.plot (Ridge regression plot), 111
ridge.reg, 30, 112, 113
ridge.reg (Ridge regression), 110
ridge.tune, 32, 111, 112
ridge.tune (Cross validation for the ridge regression), 29
rmixcomp, 64, 81
rmixcomp (Simulation of compositional data from Gaussian mixture models), 115
rmvt, 83, 95, 96
score.glms, 104
simplex.est (MLE of distributions defined in the (0, 1) interval), 83
Simulation of compositional data from Gaussian mixture models, 115
Simulation of compositional data from the Flexible Dirichlet
distribution, 116
Simulation of compositional data from the folded normal distribution, 117
skewnorm.contour, 18, 20, 22
skewnorm.contour (Contour plot of the skew skew-normal distribution in $S^2$), 20
Spatial median regression, 118
spatmed.reg, 51, 94
spatmed.reg (Spatial median regression), 118
sym.test, 79
sym.test (Log-likelihood ratio test for a symmetric Dirichlet distribution), 79
symkl.compreg (Divergence based regression for compositional data), 42
ternary, 122
ternary (Ternary diagram), 120
Ternary diagram, 120
Ternary diagram of regression models, 121
ternary.reg (Ternary diagram of regression models), 121
tflr, 12, 33, 45
tflr (Transformation-free linear regression for compositional responses and predictors), 136
The additive log-ratio transformation and its inverse, 122
The alpha-distance, 123
The alpha-k-NN regression for compositional response data, 124
The alpha-k-NN regression with compositional predictor variables, 125
The alpha-transformation, 127
The ESOV-distance, 128
The Frechet mean for compositional data, 129
The Helmert sub-matrix, 131
The k-nearest neighbours using the alpha-distance, 132
The k-NN algorithm for compositional data, 133
Total variability, 135
totvar (Total variability), 135
Transformation-free linear regression for compositional responses and predictors, 136
Tuning of the alpha generalised correlations between two compositional datasets, 137
Tuning of the bandwidth $h$ of the kernel using the maximum likelihood cross validation, 138
Tuning of the divergence based regression for compositional data with compositional data in the covariates side using the alpha-transformation, 139
Tuning of the k-NN algorithm for compositional data, 141
Tuning of the projection pursuit regression for compositional data, 143
Tuning the number of PCs in the PCR with compositional data using the alpha-transformation, 145
Tuning the parameters of the regularised discriminant analysis, 146
Tuning the principal components with GLMs, 148
Tuning the value of alpha in the alpha-regression, 150
Unit-Weibull regression models for proportions, 152
unitweib.reg (Unit-Weibull regression models for proportions), 152
univglms, 104
zadr (Zero adjusted Dirichlet regression), 153
Zero adjusted Dirichlet regression, 153
zeroreplace (Aithison's simple zero replacement strategy), 6
zilogitnorm.est (MLE of distributions defined in the $(0, 1)$ interval), 83