Package 'vdar'

October 12, 2022

Type Package

Title Discriminant Analysis Incorporating Individual Uncertainties

Version 0.1.3-2

Author Solveig Pospiech [aut, cre]

Maintainer Solveig Pospiech <s.pospiech@hzdr.de>

Description The qda() function from package 'MASS' is extended to calculate a weighted linear (LDA) and quadratic discriminant analysis (QDA) by changing the group variances and group means based on cell-wise uncertainties.
The uncertainties can be derived e.g. through relative errors for each individual measurement (cell), not only row-wise or column-wise uncertainties.
The method can be applied compositional data (e.g. portions of substances, concentrations) and non-compositional data.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports compositions

Suggests ggplot2, ggthemes, ggtern

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-04 19:40:02 UTC

R topics documented:

`calc_estimate_true_var`

Estimate true group variance

Description

Estimation of true group variance incorporating observation wise variances. The function uses the data from x and the individual variances for each observation, for example derived from uncertainties, to calculate a 'true' group variance. The variance of the matrix is corrected for the sum of the individual variances of the data set, which is normalized to the number of rows of the matrix.

Usage

```
calc_estimate_true_var(x, ...)

## Default S3 method:
calc_estimate_true_var(x, individual_var, force_pos_def = T, ...)

## S3 method for class 'rmult'
calc_estimate_true_var(x, individual_var, force_pos_def = T, ...)
```

Arguments

`x`	a matrix of data
`...`	...
`individual_var`	a matrix of cell-wise uncertainties, corresponding to the entries of 'x'
`force_pos_def`	force positive definiteness of the new group variances, default TRUE

Value

matrix of corrected group variance

Methods (by class)

- `default`: for class matrix or data.frame
- `rmult`: for class rmult

Author(s)

Solveig Pospiech, K. Gerald v.d. Boogaart

dataobs *Simulated observation data*

Description

A data set of 200 simulated observations with two observed variables and two groups, non-compositional

Usage

```
dataobs
```

Format

A data frame with 200 rows and 3 columns

Var1 simulated observed variable

Var2 simulated observed variable

Group Factor with levels 'Group 1' and 'Group 2'

dataobs_coda *Simulated observation of compositional data*

Description

A data set of 200 simulated observations with three observed variables and two groups, compositional

Usage

```
dataobs_coda
```

Format

A data frame with 200 rows and 4 columns

Var1 simulated observed variable, compositional

Var2 simulated observed variable, compositional

Var3 simulated observed variable, compositional

Group Factor with levels 'Group 1' and 'Group 2'

datatrue *Simulated true data*

Description

A data set of 200 simulated 'true' data, from which the observations are deduced, with two observed variables and two groups, non-compositional

Usage

```
datatrue
```

Format

A data frame with 200 rows and 3 columns

Var1 simulated variable

Var2 simulated variable

Group Factor with levels 'Group 1' and 'Group 2'

datatrue_coda *Simulated true compositional data*

Description

A data set of 200 simulated 'true' data, from which the observations are deduced, with three observed variables and two groups, compositional

Usage

```
datatrue_coda
```

Format

A data frame with 200 rows and 4 columns

Var1 simulated variable, compositional

Var2 simulated variable, compositional

Var3 simulated variable, compositional

Group Factor with levels 'Group 1' and 'Group 2'

force_posdef *Force positive definiteness*

Description

Function to force positive definiteness on a matrix.

Usage

```
force_posdef(x, verbose = T)
```

Arguments

x	matrix
verbose	logical, default TRUE. Should the function print the corrected eigenvalues?

Value

positive definite matrix

Author(s)

Solveig Pospiech

generalized_mean *Generalized mean*

Description

Calculates the generalized mean of a data set by using a given group variance and individual, observation-wise variances for each observation of the data set

Usage

```
generalized_mean(x, ...)

## Default S3 method:
generalized_mean(
  x,
  var,
  individual_var = matrix(0, nrow = nrow(x), ncol = ncol(x)),
  ...
)

## S3 method for class 'rmult'
```

```
generalized_mean(
  x,
  var,
  individual_var = matrix(0, nrow = nrow(x), ncol = ncol(x)^2),
  ...
)
```

Arguments

x	a matrix containing the data for which the mean should be calculated
...	not implemented
var	a matrix containing the corrected (estimated true) group variances
individual_var	a matrix containing individual variances. Default is a 0 - matrix with the dimensions of x, can be used for implementing the individual uncertainties

Value

vector of lenght of ncol(x) of generalized means

Methods (by class)

- `default`: for class matrix or data.frame
- `rmult`: for class rmult of package 'compositions'

Author(s)

Solveig Pospiech, K. Gerald v.d. Boogaart

`predict.vqda` *predict.vqda*

Description

Classify multivariate observations in conjunction with qda() or lda() of class 'vqda' or 'vlda'.

Usage

```
## S3 method for class 'vqda'
predict(object, newdata, newerror, prior = object$prior, ...)

## S3 method for class 'vlda'
predict(object, newdata, newerror, prior = object$prior, ...)
```

Arguments

object	object of class 'vqda' or 'vlda'.
newdata	data frame or matrix of cases to be classified or, if object has a formula, a data frame with columns of the same names as the variables used. A vector will be interpreted as a row vector. If newdata is missing, an attempt will be made to retrieve the data used to fit the qda object.
newerror	data frame or matrix of uncertainties corresponding to the cases in 'newdata'.
prior	the prior probabilities of group membership. If unspecified, the prior of the object are used.
...	...

Value

list containing the following components: class factor containing the predicted group likelihood matrix of dimension 'number of samples' x 'number of groups', containing the likelihood for each sample to belong to one of the groups grouping original grouping of the samples, copied from the input object

list containing the following components: class factor containing the predicted group likelihood matrix of dimension 'number of samples' x 'number of groups', containing the likelihood for each sample to belong to one of the groups grouping original grouping of the samples, copied from the input object

Methods (by class)

- vqda: predict() for class 'vqda'
- vlda: predict() for class 'vlda'

Author(s)

Solveig Pospiech, package 'MASS'

---

uncertainties *Simulated observation uncertainties*

Description

A data set of 200 simulated uncertainties with two variables and two groups, non-compositional

Usage

```
uncertainties
```

Format

A data frame with 200 rows and 3 columns

Var1 simulated observed variable

Var2 simulated observed variable

Group Factor with levels 'Group 1' and 'Group 2'

`uncertainties_coda` *Simulated observation uncertainties of compositional data*

Description

A data set of 200 simulated uncertainties with three variables and two groups, compositional

Usage

```
uncertainties_coda
```

Format

A data frame with 200 rows and 4 columns

Var1 simulated observed variable, compositional

Var2 simulated observed variable, compositional

Var3 simulated observed variable, compositional

Group Factor with levels 'Group 1' and 'Group 2'

`vlda` *Weighted Linear Discriminant Analysis*

Description

Extension of the qda() of package 'MASS' (not the lda() function) to calculate a LDA incorporating individual, cell-wise uncertainties, e.g. if the uncertainties are expressed as individual variances for each measurand.

Usage

```
vlda(x, uncertainties, grouping, prior)
```

Arguments

x	frame or matrix containing the data to be discriminated
uncertainties	data frame or matrix containing the values for uncertainties per cell. Uncertainties should be relative errors, e.g. the relative standard deviation of the measurand
grouping	a factor or character vector specifying the group for each observation (row).
prior	the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.

Details

Uncertainties can be considered in a statistical analysis either by each measured variable, by each observation or by using the individual, cell-wise uncertainties. There are several methods for incorporating variable-wise or observation-wise uncertainties into a QDA, most of them using the uncertainties as weights for the variables or observations of the data set. The term 'cell-wise uncertainties' describe a data set of d analysed variables where each observation has an individual uncertainty for each of the d variables conforming it. Hence, a data set of $n \times d$ data values has associated a data set of $n \times d$ individual uncertainties. Instead of weighting the columns or rows of the data set, the vlda() function uses uncertainties to recalculate better estimates of the group variances and group means. It is internally very similar to the vqda function, but with an averaged group variance for all groups. If the presence of uncertainties is not accounted for, the decision rules are based on the group variances calculated by the given data set. But this observed group variance might deviate notably from the group variance, which can be estimated including the uncertainties. This methodological framework does not only allow to incorporate cell-wise uncertainties, but also would largely be valid if the information about the co-dependency between uncertainties within each observation would be reported.

Value

object of class 'vlda' containing the following components: prior the prior probabilities used. counts counts per group. means the group means. generalizedMeans the group means calculated by the function generalized_mean groupVarCorrected the group variances calculated by the function calc_estimate_true_var lev the levels of the grouping factor. grouping the factor specifying the class for each observation.

Author(s)

Solveig Pospiech, package 'MASS'

References

Pospiech, S., R. Tolosana-Delgado and K.G. van den Boogaart (2020) Discriminant Analysis for Compositional Data Incorporating Cell-Wise Uncertainties, Mathematical Geosciences

Examples

```
# for non-compositional data:
data("dataobs")
data("uncertainties")
mylda = vlda(x = dataobs[, 1:2], uncertainties = uncertainties[, 1:2], grouping = dataobs$Group)
mypred = predict(mylda, newdata = dataobs[, 1:2], newerror = uncertainties[, 1:2])
forplot = cbind(dataobs, LG1 = mypred$posterior[,1])
if (require("ggplot2")) {
  scatter_plot = ggplot(data = forplot, aes(x = Var1, y = Var2)) +
    geom_point(aes(shape = Group, color = LG1))
  if (require("ggthemes")) {
    scatter_plot = scatter_plot +
        scale_color_gradientn(colours = colorblind_pal()(5))
  }
  scatter_plot
}

# for compositional data
data("dataobs_coda")
data("uncertainties_coda")
require(compositions)
# generate ilr-transformation (from package 'compositions')
data_ilr = ilr(dataobs_coda[, 1:3])
uncert_ilr = t(simplify2array(apply(uncertainties_coda[, 1:3],1,
                      function(Delta) clrvar2ilr(diag(Delta)))))
uncert_ilr = compositions::rmult(uncert_ilr) # change class into rmult from package 'compositions'
mylda_coda = vlda(x = data_ilr, uncertainties = uncert_ilr, grouping = dataobs_coda$Group)
mypred_coda = predict(mylda_coda, newdata = data_ilr, newerror = uncert_ilr)
forplot_coda = cbind(dataobs_coda, LG1 = mypred_coda$posterior[,1])
# if 'ggtern' is installed, you can plot via ggtern:
# if (require("ggtern")) {
#   ternary_plot = ggtern(data = forplot_coda, aes(x = Var1, y = Var2, z = Var3)) +
#     geom_point(aes(shape = Group, color = LG1))
#   if (require("ggthemes")) {
#     ternary_plot = ternary_plot +
#         scale_color_gradientn(colours = colorblind_pal()(5))
#   }
#   ternary_plot
# }
```

---

vqda *Weighted Quadratic Discriminant Analysis*

Description

Extension of the qda() of package 'MASS' to calculate a QDA incorporating individual, cell-wise uncertainties, e.g. if the uncertainties are expressed as individual variances for each measurand.

Usage

```
vqda(x, uncertainties, grouping, prior)
```

Arguments

x	data frame or matrix containing the data to be discriminated
uncertainties	data frame or matrix containing the values for uncertainties per cell. Uncertainties should be relative errors, e.g. the relative standard deviation of the measurand
grouping	a factor or character vector specifying the group for each observation (row).
prior	the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.

Details

Uncertainties can be considered in a statistical analysis either by each measured variable, by each observation or by using the individual, cell-wise uncertainties. There are several methods for incorporating variable-wise or observation-wise uncertainties into a QDA, most of them using the uncertainties as weights for the variables or observations of the data set. The term 'cell-wise uncertainties' describe a data set of d analysed variables where each observation has an individual uncertainty for each of the d variables conforming it. Hence, a data set of $n \times d$ data values has associated a data set of $n \times d$ individual uncertainties. Instead of weighting the columns or rows of the data set, the vqda() function uses uncertainties to recalculate better estimates of the group variances and group means. If the presence of uncertainties is not accounted for, the decision rules are based on the group variances calculated by the given data set. But this observed group variance might deviate notably from the group variance, which can be estimated including the uncertainties. This methodological framework does not only allow to incorporate cell-wise uncertainties, but also would largely be valid if the information about the co-dependency between uncertainties within each observation would be reported.

Value

object of class 'vqda' containing the following components: `prior` the prior probabilities used. `counts` counts per group. `means` the group means. `generalizedMeans` the group means calculated by the function `generalized_mean` `groupVarCorrected` the group variances calculated by the function `calc_estimate_true_var` `lev` the levels of the grouping factor. `grouping` the factor specifying the class for each observation.

Author(s)

Solveig Pospiech, package 'MASS'

References

Pospiech, S., R. Tolosana-Delgado and K.G. van den Boogaart (2020) Discriminant Analysis for Compositional Data Incorporating Cell-Wise Uncertainties, Mathematical Geosciences

Examples

```
# for non-compositional data:
data("dataobs")
data("uncertainties")
myqda = vqda(x = dataobs[, 1:2], uncertainties = uncertainties[, 1:2], grouping = dataobs$Group)
mypred = predict(myqda, newdata = dataobs[, 1:2], newerror = uncertainties[, 1:2])
forplot = cbind(dataobs, LG1 = mypred$posterior[,1])
if (require("ggplot2")) {
  scatter_plot = ggplot(data = forplot, aes(x = Var1, y = Var2)) +
    geom_point(aes(shape = Group, color = LG1))
  if (require("ggthemes")) {
    scatter_plot = scatter_plot +
        scale_color_gradientn(colours = colorblind_pal()(5))
  }
  scatter_plot
}

# for compositional data
data("dataobs_coda")
data("uncertainties_coda")
require(compositions)
# generate ilr-transformation (from package 'compositions')
data_ilr = ilr(dataobs_coda[, 1:3])
uncert_ilr = t(simplify2array(apply(uncertainties_coda[, 1:3],1,
                      function(Delta) clrvar2ilr(diag(Delta)))))
uncert_ilr = compositions::rmult(uncert_ilr) # change class into rmult from package 'compositions'
myqda_coda = vqda(x = data_ilr, uncertainties = uncert_ilr, grouping = dataobs_coda$Group)
mypred_coda = predict(myqda_coda, newdata = data_ilr, newerror = uncert_ilr)
forplot_coda = cbind(dataobs_coda, LG1 = mypred_coda$posterior[,1])
# if 'ggtern' is installed, you can plot via ggtern:
# if (require("ggtern")) {
#   ternary_plot = ggtern(data = forplot_coda, aes(x = Var1, y = Var2, z = Var3)) +
#     geom_point(aes(shape = Group, color = LG1))
#   if (require("ggthemes")) {
#     ternary_plot = ternary_plot +
#         scale_color_gradientn(colours = colorblind_pal()(5))
#   }
#   ternary_plot
# }
```

Index