Package 'fso'

October 13, 2022

Version 2.1-2

Date 2022-09-26

Title Fuzzy Set Ordination

Author David W. Roberts <droberts@montana.edu>

Maintainer David W. Roberts <droberts@montana.edu>

Description Fuzzy set ordination is a multivariate analysis used in ecology to
    relate the composition of samples to possible explanatory variables. While
    differing in theory and method, in practice, the use is similar to 'constrained
    ordination.' The package contains plotting and summary functions as well as
    the analyses.

Depends labdsv

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-09-26 15:30:02 UTC

R topics documented:

---

fso   *Fuzzy Set Ordination*

Description

Compute a fuzzy set for samples along a specified environmental or experimental gradient based on sample similarities and gradient values as weights. The fuzzy set memberships represent the degree to which a sample is similar to one end of the gradient while not similar to the other.

Usage

```
## S3 method for class 'formula'
fso(formula,dis,data,permute=FALSE,...)
## Default S3 method:
fso(x,dis,permute=FALSE,...)
## S3 method for class 'fso'
summary(object,...)
```

Arguments

formula	a formula in the form of ~x+y+z (no LHS)
dis	a dist object such as that returned by dist, dsvdis, or 'vegdist'
data	a data frame that holds variables listed in the formula
permute	if FALSE, estimate probabilities from Z distribution for correlation; if numeric, estimate probabilities from permutation of input
x	a numerical vector, a matrix, or numeric dataframe
object	an object of class 'fso'
...	generic arguments for future use

Details

The algorithm converts the input to a full symmetric similarity matrix and bounds [0,1] (if necessary). It then calculates several fuzzy sets:

$$mu_a(i) = (x_i - min(x))/(max(x) - min(x))$$

$$mu_b(i) = 1 - mu_a(i)$$

$$mu_c(i) = \left(\sum_j mu_a(j) \times y_{i,j} i\right) / \sum_j \mu_a(j)$$

$$mu_d(i) = (\sum_j mu_b(j) \times y_{i,j}) / \sum_j \mu_b(j)$$

A separate fuzzy set ordination is calculated for each term in the formula. If x is a matrix or dataframe a separate fuzzy set ordination is calculated for each column or field.

If permute is numeric, the permutation is performed permute-1 times, and the probability is estimated as $(correlations >= observed + 1)/permute$

Value

An object of class 'fso' which has the following elements:

mu	the fuzzy membership values for individual plots in the fuzzy set. If x is a matrix or dataframe then mu is also a matrix of the same dimension.
data	a copy of data vector or matrix y
r	the correlation between the original vector and the fuzzy set. If x is a matrix or dataframe then r is a vector with length equal to the number of columns in the matrix or dataframe.
p	the probability of obtaining a correlation between the data and fuzzy set as large as observed
d	the correlation of pair-wise distances among each fuzzy set compared to the dissimilarity matrix from which the fso was constructed
var	the variable name(s) from matrix y

Note

Fuzzy set ordination is a method of multivariate analysis employed in vegetation analysis.

fso can be run with the first argument either a dataframe or a formula (with no left hand side). The formula version has distinct advantages:

1) The data= argument allows the user to specify a data frame containing the variables of interest. In this way variables need not be local.

2) The formula version handles categorical variables by converting them to dummy variables. In the default version, all variables must be quantitative or binary.

3) The formula version is somewhat more graceful about handling missing values in the data.

Author(s)

David W. Roberts <droberts@montana.edu>

References

Roberts, D.W. 1986. Ordination on the basis of fuzzy set theory. Vegetatio 66:123-131.

Roberts, D.W. 2007. Statistical analysis of multidimensional fuzzy set ordinations. Ecology 89:1246-1260.

Roberts, D.W. 2009. Comparison of multidimensional fuzzy set ordination with CCA and DB-RDA. Ecology. 90:2622-2634.

Examples

```
library(labdsv)
data(bryceveg)
data(brycesite)
dis <- dsvdis(bryceveg,'bray/curtis')
elev.fso <- fso(brycesite$elev,dis)
elev.fso <- fso(~elev,dis,data=brycesite)
```

```
plot.fso(elev.fso)
summary(elev.fso)
```

---

mfso   *Multidimensional Fuzzy Set Ordination*

Description

A multidimensional extension of fuzzy set ordination (FSO) that constructs a multidimensional ordination by mapping samples from fuzzy topological space to Euclidean space for statistical analysis. MFSO can be used in exploratory or testing modes.

Usage

```
## S3 method for class 'formula'
mfso(formula,dis,data,permute=FALSE,lm=TRUE,scaling=1,...)
## Default S3 method:
mfso(x,dis,permute=FALSE,scaling=1,lm=TRUE,notmis=NULL,...)
## S3 method for class 'mfso'
summary(object,...)
```

Arguments

formula	Model formula, with no left hand side. Right hand side gives the independent variables to use in fitting the model
dis	a dist object of class 'dist' returned from dist, 'vegdist', or [dsvdis]
data	a data frame containing the variables specified in the formula
permute	a switch to control how the probability of correlations is calculated. permute=FALSE (the default) uses a parametric Z distribution approximation; permute=n permutes the independent variables (permute-1) times and estimates the probability as (m+1)/(permute) where m is the number of permuted correlations greater than or equal to the observed correlation.
lm	a switch to control scaling of axes after the first axis. If lm=TRUE (the default) each axis is constructed independently, and then subjected to a Gram-Schmidt orthogonalization to all previous axes to preserve only the the variability that is uncorrelated with all previous axes. If lm=FALSE, the full extent of all axes is preserved without correcting for correlation with previous axes.
scaling	a switch to control how the initial fuzzy set axes are scaled: 1 = use raw μ membership values, 2 = relativize μ values [0,1], 3 = relativize μ values [0,1] and multiply by respective correlation coefficient.
x	a quantitative matrix or dataframe. One axis will be fit for each column
notmis	a vector passed from the formula version of mfso to control for missing values in the data
object	an object of class 'mfso'
...	generic arguments for future use

Details

mfso performs individual fso calculations on each column of a data frame or matrix, and then combines those fso axes into a higher dimensional object. The algorithm of fuzzy set ordination is described in the help file for [fso]. The key element in mfso is the Gram-Schmidt orthogonalization, which ensures that each axis is independent of all previous axes. In practice, each axis is regressed against all previous axes, and the residuals are retained as the result.

Value

an object of class 'mfso' with components:

mu	a matrix of fuzzy set memberships of samples, analogous to the coordinates of the samples along the axes, one column for each axis
data	a dataframe containing the independent variables as columns
r	a vector of correlation coefficients, one for each axis in order
p	a vector of probabilities of observing correlations as high as observed
var	a vector of variables names used in fitting the model
gamma	a vector of the fraction of variance for an axis that is independent of all previous axes

Note

MFSO is an extension of single dimensional fuzzy set ordination designed to achieve low dimensional representations of a dissimilarity or distance matrix as a function of environmental or experimental variables.

If you set lm=FALSE, an mfso is equivalent to an fso, but the plotting routines differ. For an mfso, the plotting routine plots each axis against all others in turn; for an fso the plotting routine plots each axis against the environmental or experimental variable it is derived from.

Author(s)

David W. Roberts <droberts@montana.edu>

References

Roberts, D.W. 2007. Statistical analysis of multidimensional fuzzy set ordinations. Ecology 89:1246-1260.

Roberts, D.W. 2009. Comparison of multidimensional fuzzy set ordination with CCA and DB-RDA. Ecology. 90:2622-2634.

Examples

```
require(labdsv)
data(bryceveg) # returns a vegetation dataframe
data(brycesite) # returns a dataframe of environmental variables
dis.bc <- dsvdis(bryceveg,'bray/curtis')
            # returns an object of class sQuote{dist}
demo.mfso <- mfso(~elev+slope+av,dis.bc,data=brycesite) # creates the mfso
```

```
    summary(demo.mfso)
    ## Not run: plot(demo.mfso)
```

---

plot.fso   *Plotting Routines for Fuzzy Set Ordinations*

Description

A set of routines for plotting, highlighting points, or identifying the distribution of a third variable on an fso.

Usage

```
## S3 method for class 'fso'
plot(x, which="all", xlab = x$var, ylab="mu(x)",
    title="",r=TRUE,pch=1,...)
## S3 method for class 'fso'
points(x, overlay, which="all", col=2, cex=1, pch=1, ...)
## S3 method for class 'fso'
plotid(ord, which="all", xlab=ord$var, ylab="mu(x)", title="",
    r=TRUE, pch=1, labels=NULL, ...)
## S3 method for class 'fso'
hilight(ord, overlay, which=1, cols = c(2, 3, 4, 5,
    6, 7), symbol = c(1, 3, 5), ...)
## S3 method for class 'fso'
chullord(ord, overlay, which = 1, cols = c(2, 3, 4, 5,
    6, 7), ltys = c(1, 2, 3), ...)
## S3 method for class 'fso'
boxplot(x, ...)
```

Arguments

x	an object of class 'fso'
ord	an object of class 'fso'
which	a switch to control which axis is plotted
r	a switch to control printing the correlation coefficient in the plot
fso	an object of class 'fso' from fso
overlay	a logical vector of the same length as the number of points in the plot
labels	a vector of labels to print next to the identified points
symbol	an integer or vector of integers to control which symbols are printed in which order on the plot by specifying values to pch
ltys	an integer or vector of integers to control the line styles of convex hull polygons
xlab	text label for X axis
ylab	text label for Y axis

title	an overall title for the plot (equivalent to main)
pch	the symbol for plotting
col	the color for plotted symbols
cex	the character expansion factor (font size)
cols	an integer vector specifying color order
...	arguments to pass to the underlying plot function

Details

Fuzzy set ordinations (FSO) are almost inherently graphical, and routines to facilitate plotting and overlaying are essential to work effectively with them.

A fuzzy set ordination object (an object of class 'fso') may contain one or more axes. In the simplest case, for a single-axis fso, the plot routine plots the underlying raw data on the X axis and the fuzzy set memberships on the Y axis, including by default the correlation coefficient in the upper left corner. For fsos containing multiple axes, the default (which="all") is to plot the raw data on the X axis, the respective fuzzy set memberships on the Y axis, plotting all axes in turn with a prompt to move to the next panel. This is often effective. It is also possible to plot a single panel out of the set of axes, specifying the axis as an integer with, e.g., "which = 2."

The 'points' function can be used to highlight or identify specific points in the plot. The 'points' function requires a logical vector (TRUE/FALSE) of the same length as the number of points in the plot. The default behavior is to color the points with a respective TRUE value red. It is possible to control the color (with col=), size (with cex=) and symbol (with pch=) of the points.

The 'plotid' function can be used to label or identify specific points with the mouse. Clicking the left mouse button adjacent to a point causes the point to be labeled, offset in the direction of the click relative to the point. Clicking the right mouse button exits the routine. The default (labels=NULL) is to label points with the row number in the data.frame (or position in the vector) for the point. Alternatively, specifying a vector of labels (labels=) prints the respective labels. If the data were derived from a data.frame, the row.names of the data.frame are often a good choice, but the labels can also be used with a factor vector to identify the distribution of values of a factor in the ordination (but see hilight as well).

The 'hilight' function identifies the factor values of points in the ordination, using color and symbols to identify unique values (up to 18 values by default). The colors and symbols used can be specified by the 'cols=' and 'symbol=' arguments, which should both be integers or integer vectors. The default of colors 2, 3, 4, 5, 6, 7 and symbols 1, 3, 5 shows well in most cases, but on colored backgrounds you may need to adjust 'cols='. If you have a factor with more than 18 classes you will need to augment the 'symbol=' vector with more values.

The 'chullord' function plots convex hulls around all points sharing the same value for a factor variable, and colors all points of that value to match. The convention on colors follows 'hilight'.

The 'boxplot' function plots boxplots of the μ membership values for the fuzzy sets in the fso.

Note

The plotting and highlighting routines for fso are designed to match the same routines for other ordinations in package labdsv.

Author(s)

David W. Roberts <droberts@montana.edu>

Examples

```
require(labdsv) # to obtain access to data sets and dissimilarity function
data(bryceveg) # vegetation data
data(brycesite) # environmental data
dis.bc <- dsvdis(bryceveg,'bray/curtis') # produce \sQuote{dist} object
demo.fso <- fso(~elev+slope+av,dis.bc,data=brycesite)
## Not run: plot(demo.fso)
## Not run: hilight(demo.mfso,brycesite$quad)
```

---

plot.mfso   *Plotting Routines for Multidimensional Fuzzy Set Ordinations*

Description

A set of routines for plotting, identifying, or highlighting points in a multidimensional fuzzy set
ordination (MFSO).

Usage

```
## S3 method for class 'mfso'
plot(x, dis=NULL, pch=1, ax=NULL, ay=NULL, ...)
## S3 method for class 'mfso'
points(x, overlay, col=2, pch=1, ...)
## S3 method for class 'mfso'
plotid(ord, dis=NULL, labels=NULL, ...)
## S3 method for class 'mfso'
hilight(ord, overlay, cols = c(2, 3, 4, 5,
    6, 7), symbol = c(1, 3, 5), ...)
## S3 method for class 'mfso'
chullord(ord, overlay, cols = c(2, 3, 4, 5,
    6, 7), ltys = c(1, 2, 3), ...)
## S3 method for class 'mfso'
boxplot(x, ...)
## S3 method for class 'mfso'
thull(ord,var,grain,ax=1,ay=2,col=2,grid=50,
        nlevels=5,levels=NULL,lty=1,numitr=100,...)
```

Arguments

x	an object of class 'mfso'
ax	X axis number
ay	Y axis number
ord	an object of class 'mfso'

`mfso`	an object of class 'mfso'
`dis`	an object of class 'dist' from dist, [dsvdis](#), or 'vegdist'
`overlay`	a logical vector of the same length as the number of points in the plot
`labels`	a vector of labels to print next to the identified points
`symbol`	an integer or vector of integers to control which symbols are printed in which order on the plot by specifying values to pch
`ltys`	an integer or vector of integers to control the line styles of convex hull polygons
`pch`	the symbol to plot
`col`	the color to use for plotted symbols
`cols`	an integer vector for color order
`var`	a variable to fit with a tensioned hull
`grain`	the size of the moving window used to calculate the tensioned hull
`grid`	the number of cells in the image version of the tensioned hull
`nlevels`	the number of contour levels to plot the tensioned hull
`levels`	a logical variable to control plotting the contours on the tensioned hull
`lty`	the line type to use in drawing the contours
`numitr`	the number of random iterations to use to compute the probability of obtaining as small a tensioned hull as observed
`...`	arguments to pass to function points

Details

Multidimensional fuzzy set ordinations (MFSO) are almost inherently graphical, and routines to facilitate plotting and overlaying are essential to work effectively with them.

A multidimensional fuzzy set ordination object (an object of class 'mfso') generally contains at least two axes, and may contain many more. By default, the plot routine plots all possible axis pairs in order. If 'ax' and 'ay' are specified only a single plot is produced with X axis ax and Y axis ay. If 'dist' object is passed with the 'dis=' argument, the final panel is a plot of the dissimilarity or distance matrix values on the X axis and the pair-wise ordination distances on the Y axis with the correlation coefficient in the upper left hand corner.

The 'points' function can be used to highlight or identify specific points in the plot. The 'points' function requires a logical vector (TRUE/FALSE) of the same length as the number of points in the plot. The default behavior is to color the points with a respective TRUE value red. It is possible to control the color (with col=), size (with cex=) and symbol (with pch=) of the points.

The 'plotid' function can be used to label or identify specific points with the mouse. Clicking the left mouse button adjacent to a point causes the point to be labeled, offset in the direction of the click relative to the point. Clicking the right mouse button exits the routine. The default (labels=NULL) is to label points with the row number in the data.frame (or position in the vector) for the point. Alternatively, specifying a vector of labels (labels=) prints the respective labels. If the data were derived from a data.frame, the row.names of the data.frame are often a good choice, but the labels can also be used with a factor vector to identify the distribution of values of a factor in the ordination (but see hilight as well).

The 'hilight' function identifies the factor values of points in the ordination, using color and symbols to identify unique values (up to 18 values by default). The colors and symbols used can be specified by the 'col=' and 'symbol=' arguments, which should both be integers or integer vectors. The default of colors 2, 3, 4, 5, 6, 7 and symbols 1, 3, 5 shows well in most cases, but on colored backgrounds you may need to adjust 'col='. If you have a factor with more than 18 classes you will need to augment the 'symbol=' vector with more values.

The 'chullord' function plots convex hulls around all points sharing the same value for a factor variable, and colors all points of that value to match. The convention on colors follows `hilight`.

The 'boxplot' function plots boxplots of the μ membership values in the MFSO.

The 'thull' funntion drapes a tensioned hull for variable 'var' over the plotted mfso.

Value

Note

The plotting and highlighting routines for mfso are designed to match the same routines for other ordinations in package `labdsv`.

Author(s)

David W. Roberts <droberts@montana.edu>

Examples

```
require(labdsv) # to obtain access to data sets and dissimilarity function
data(bryceveg) # vegetation data
data(brycesite) # environmental data
dis.bc <- dsvdis(bryceveg,'bray/curtis') # produce \sQuote{dist} object
demo.mfso <- mfso(~elev+slope+av,dis.bc,data=brycesite)
plot(demo.mfso)
## Not run: hilight(demo.mfso,brycesite$quad) # requires interaction
```

| step.mfso | *Step-Wise Forward Variable Selection in a Multivariate Fuzzy Set Ordination* |

Description

A simple routine to screen variables for addition to a multivariate fuzzy set ordination (MFSO). The routine operates by adding variables one at a time to an existing MFSO (which can be NULL), and calculating the correlation coefficient between the underlying dissimilarity matrix (object of class 'dist') and the pair-wise distances in the MFSO ordination.

Usage

```
step.mfso(dis,start,add,numitr=100,scaling=1)
```

Arguments

dis	a dissimilarity of distance object from dist, [dsvdis](#), or 'vegdist' or other 'dist' object
start	either NULL (to find the first variable to add) or a data.frame of binary or quantitative variables to use in the base model
add	a data.frame of binary or quantitative variables to screen for addition to the model
numitr	the number of random permutations of a vector to use in establishing the probability of observing as large an increase in correlation as observed
scaling	the scaling parameter to pass along to mfso

Details

'mfso' is intended as a tool for analysis of multiple competing hypotheses, and the analyst is expected to have a priori models to compare. Nonetheless, 'mfso' can be used in a hypothesis generating variable screening mode by maximizing the correlation between the underlying dissimilarity matrix and the pair-wise distances in the 'mfso' ordination.

The step.mfso function is an inelegant approach to step-wise forward variable selection in mfso. It considers each variable offered in turn, calculates the mfso resulting from adding that variable to the given mfso, permutes that variable 'numitr' times, and determines a probability of observing as large an increase in correlation as observed. After testing all variables for inclusion, it simply prints a table of the calculations, and the analyst has to rerun the routine adding the selected variable to data.frame 'start' and deleting it from 'add'.

While it would be nice to automate the production of the step-wise 'mfso', to date I have only implemented this limited function. In addition, model parsimony is ensured by the permutation routine, rather than an AIC-based approach, and doesn't directly penalize for degrees of freedom (number of variables).

Value

Produces a table of the analysis but does not produce any objects

Author(s)

David W. Roberts <droberts@montana.edu>

References

Roberts, D.W. 2007. Statistical analysis of multidimensional fuzzy set ordinations. Ecology in press

Examples

```
## Not run: require(labdsv) # make data available
        ## Not run: data(bryceveg)  # get vegetation data
        ## Not run: data(brycesite) # get environmental data
        ## Not run: dis.bc <- dsvdis(bryceveg,'bray.curtis') # produce dist object
        ## Not run: attach(brycesite)   # make variables easily available
```

```
## Not run: step.mfso(dis.bc,start=NULL,add=data.frame(elev,slope,av))
## Not run: step.mfso(dis.bc,start=data.frame(elev),add=data.frame(slope,av))
```

Index