Package 'freqparcoord'

May 8, 2026

Version 1.0.1

Author Norm Matloff <normmatloff@gmail.com> and Yingkang Xie <yingkang.xie@gmail.com>

Maintainer Norm Matloff <normmatloff@gmail.com>

Date 2016-01-15

Title Novel Methods for Parallel Coordinates

Description New approaches to parallel coordinates plots for multivariate data visualization, including applications to clustering, outlier hunting and regression diagnostics. Includes general functions for multivariate nonparametric density and regression estimation, using parallel computation.

Depends parallel, ggplot2, GGally, FNN, mvtnorm

Suggests mgcv

LazyLoad no

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2016-01-17 10:59:33

Contents

freqparcoord 2
mlb 5
newadult 6
oliveoils 6
posjitter 7
prgeng 7
regdiag 8
rmixmvnorm 10
smoothz 11

Index **13**

freqparcoord *Frequency-based parallel coordinates.*

Description

A novel approach to the parallel coordinates method for visualizing many variables at once.

(a) Addresses the screen-clutter problem in parallel coordinates, by only plotting the "most typical" cases, meaning those with the highest estimated multivariate density values. This makes it easier to discern relations between variables, especially those whose axes are "distant" from each other.

(b) One can also plot the "least typical" cases, i.e. those with the lowest density values, in order to find outliers.

(c) One can plot only cases that are "local maxima" in terms of density, as a means of performing clustering.

Usage

```
freqparcoord(x,m,dispcols=1:ncol(x),grpvar=NULL,
      method="maxdens",faceting="vert",k=50,klm=5*k,
            keepidxs=NULL,plotidxs=FALSE,cls=NULL)
```

Arguments

x	The data, in data frame or matrix form. If there are indicator
m	Number of lines to plot for each group. A negative value in conjunction with method being "maxdens" indicates that the lowest-density lines are to be plotted. If method is "locmax", m is forced to 1.
dispcols	Numbers of the columns of x to be displayed.
grpvar	Column number for the grouping variable, if any (if none, all the data is treated as a single group); vector or factor. Must not be in dispcols. If method is "locmax", grpvar is forced to NULL
method	What to display: "maxdens" for plotting the most (or least) typical lines, "locmax" for cluster hunting, or "randsamp" for plotting a random sample of lines.
faceting	How to display groups, if present. Use "vert" for vertical stacking of group plots, "horiz" for horizontal ones, or "none" to draw all lines in one plot, color-coding by group.
k	Number of nearest neighbors to use for density estimation.
klm	If method is "locmax", number of nearest neighbors to use for finding local maxima for cluster hunting. Generally needs to be much larger than k, to avoid "noise fitting."
keepidxs	If not NULL, the indices of the rows of x that are plotted will be stored in a component idxs of the return value. The rows themselves will be in a component xdisp, ordered by x[,dispcols[1].]

plotidxs	If TRUE, lines in the display will be annotated with their case numbers, i.e. their row numbers within x. Use only with small values of m, as overplotting may occur.
cls	Cluster, if any (see the parallel package) for parallel computation.

Details

In general, a parallel coordinates plot draws each data point as a polygonal line. Say for example we have variables Height, Weight and Age (inches, pounds, years). The vertical axes are drawn, one for each variable. Then each point, "connects the dots" on the vertical axes. For instance, the point (70, 160, 28) would be represented as a segmented line connecting 70 on the Height axis, 160 on the Weight axis and 28 on the Age axis. See for example parcoord in the **MASS** package.

The problem with the parallel coordinates method is screen clutter–too many lines filling the screen. The treatment here avoids this problem by plotting only the lines having the highest estimated multivariate density (or variants discussed below).

If method = "maxdens", the m most frequent (m positive) or least frequent (m negative) rows of x will be plotted from each group, where frequency is measured by density value (the nongroup case being considered one group).

If method = "locmax", the rows having the property that their density value is highest in their klm-neighborhood will be plotted.

Otherwise, m random rows will be displayed.

The lines will be color-coded according to density value. Density values are computed separately within groups.

If cls is non-null, the computation will be done in parallel. See knndens.

The data is centered and scaled using scale before analysis, including before any grouping operations. Thus the selected rows are still plotted on the scale of the entire data set; for instance, a vertical axis value of 0 corresponds to the mean of the given variable. If some variable is constant, scaling is impossible, and an error message, "arguments imply differing number of rows: 0, 1," will appear. In such case, try a larger value of m.

Density estimation is done through the k-Nearest Neighbor method, in the function smoothz. (Due to use above-mentioned use of scale, this is meaningful even if some variables are of the indicator/dummy type, i.e. 1-0 valued to indicate the presence or absence of some trait. This way such variables are comparable to the continuous ones in the distance compuations.) For any point, the k nearest data points are found, requiring powers of distances in a denominator. With large, discrete data, the denominator may be 0. In such cases, it is recommended that you apply jitter or (from this package) posjitter. The same visual patterns will emerge.

As with any exploratory tool, the user should experiment with the values of the arguments, especially the klm argument with the method "locmax".

Note that with long-tailed distributions, the scaled data will be disproportionately negative. Thus the magnitude of the scaled variables should be viewed relative to each other, rather than to 0.

If you use too large a value for k, it may be larger than some group size, generating an error message like "k should be less than sample size." If so, try a smaller k. If a plot would contain only one line, this may cause a problem with some graphics systems.

Value

Object of type "gg" (**ggplot2** object), with components `idxs` and `xdisp` added if `keepidxs` is not NULL (see argument `keepidxs` above).

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu> and Yingkang Xie <yingkang.xie@gmail.com>

Examples

```
# baseball player data courtesy of UCLA Stat. Dept., www.socr.ucla.edu
data(mlb)

# plot baseball data, broken down by position category (infield,
# outfield, etc.); plot the 5 higest-density values in each group
freqparcoord(mlb,5,4:6,7,method="maxdens")
# we see that the most typical pitchers are tall and young, while the
# catchers are short and heavy

# same, but no grouping
freqparcoord(mlb,5,4:6,method="maxdens")

# find the outliers, 1 for each position
freqparcoord(mlb,-1,4:6,7)
# for instance we see an infielder of average height and weight, but
# extremely high age, worth looking into

# do the same, but also plot and retain the indices of the rows being
# plotted, and the rows themselves
p <- freqparcoord(mlb,-1,4:6,7,keepidxs=4,plotidxs=TRUE)
p
p$idxs
p$xdisp
# ah, that outlier infielder was case number 674,
# Julio Franco, 48 years old!

# olive oil data courtesy of Dr. Martin Theus
data(oliveoils)
olv <- oliveoils

# there are 9 olive-oil producing areas of Italy, named Area here
# check whether the area groups have distinct patterns (yes)
freqparcoord(olv,1,3:10,1,k=15)

# same check but looking at within-group variation (turns out that some
# variables are more diverse in some areas than others)
freqparcoord(olv,25,3:10,1,k=15)
# yes, definitely, e.g. wide variation in stearic in Sicily

# look at it without stacking the groups
freqparcoord(olv,25,3:10,1,faceting="none",k=15)
# prettier this way, with some patterns just as discernible
```

```
## Not run:
# programmers and engineers in Silicon Valley, 2000 census
data(prgeng)
pg <- prgeng

# compare men and women
freqparcoord(pg,10,dispcols=c(1,3,8),grpvar=7,faceting="horiz")
# men seem to fall into 2 subgroups, one with very low wages; let's get
# a printout of the plotted points, grouped by gender
p <-
   freqparcoord(pg,10,dispcols=c(1,3,8),grpvar=7,faceting="horiz",keepidxs=7);
p$xdisp
# ah, there are some wages like $3000; delete those and look again;
pg1 <- pg[pg$wageinc >= 40000 & pg$wkswrkd >= 48,]
freqparcoord(pg1,50,dispcols=c(1,3,8),grpvar=7,faceting="horiz",keepidxs=7)
# the women seem to fall in 2 age groups, but not the men, worth further
# analysis
# note that all have the same education, a bachelor's degree, the
# most frequent level

# generate some simulated data with clusters at (0,0), (1,2) and (3,3),
# and see whether "locmax" (clustering) picks them up
cv <- 0.5*diag(2)
x <- rmixmvnorm(10000,2,3,list(c(0,0),c(1,2),c(3,3)),list(cv,cv,cv))
p <- freqparcoord(x,m=1,method="locmax",keepidxs=1,k=50,klm=800)
p$xdisp  # worked well in this case, centers near (0,0), (1,2), (3,3)

# see how well outlier detection works
x <- rmixmvnorm(10000,2,3,list(c(0,0),c(1,2),c(8,8)),list(cv,cv,cv),
   wts=c(0.49,0.49,0.02))
# most of the outliers should be out toward (8,8)
p <- freqparcoord(x,m=-10,keepidxs=1)
p$xdisp

## End(Not run)
```

`mlb` *Major Leage Baseball player data set.*

Description

Heights, weights, ages etc. of major league baseball players. A new variable has been added, consolidating positions into Infielders, Outfielders, Catchers and Pitchers.

Included here with the permission of the UCLA Statistics Department.

Usage

```
data(mlb); mlb
```

newadult *UCI adult income data set, adapted*

Description

This data set is adapted from the Adult data from the UCI Machine Learning Repository, which was in turn adapted from Census data on adult incomes and other demographic variables. The UCI data is used here with permission from Ronny Kohavi.

The variables are:

- `gt50`, which converts the original `>50K` variable to an indicator variable; 1 for income greater than $50,000, else 0
- `edu`, which converts a set of education levels to approximate number of years of schooling
- `age`
- `gender`, 1 for male, 0 for female
- `mar`, 1 for married, 0 for single

Usage

```
data(newadult); newadult
```

oliveoils *Italian olive oils data set.*

Description

Italian olive oils data set, as used in *Graphics of Large Datasets: Visualizing a Million*, by Antony Unwin, Martin Theus and Heike Hofmann, Springer, 2006. Included here with permission of Dr. Martin Theus.

Usage

```
data(oliveoils); oliveoils
```

posjitter *Add positive jitter.*

Description

Similar to jitter, but only generating values in (0,1). A typical example of use is for an age variable, which in many data sets is truncated to the lowest integer.

Usage

```
posjitter(x)
```

Arguments

x	Vector to which jitter is to be added.

Value

The vector `x + runif(length(x))`.

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu>

prgeng *Silicon Valley programmers and engineers*

Description

This data set is adapted from the 2000 Census (5% sample, person records). It is restricted to programmers and engineers in the Silicon Valley area.

The variable codes, e.g. occupational codes, are available from the Census Bureau, at http://www.census.gov/prod/cen2000/doc/pums.pdf. (Short code lists are given in the record layout, but longer ones are in the appendix Code Lists.)

The variables are:

- age, with a U(0,1) variate added for jitter
- cit, citizenship; 1-4 code various categories of citizens; 5 means noncitizen (including permanent residents
- educ: 01-09 code no college; 10-12 means some college; 13 is a bachelor's degree, 14 a master's, 15 a professiona deal and 16 is a doctorate
- engl, English proficiency
- occ, occupation
- birth, place of birth

- wageinc, wage income
- wkswrkd, number of weeks worked
- yrentry, year of entry to the U.S. (0 for natives)
- powpuma, location of work
- gender, 1 for male, 2 for female

Usage

```
data(prgeng); prgeng
```

---

regdiag *Diagnosing regression model fit using parallel coordinates.*

Description

Performs parametric regression model fit diagnostics, based on freqparcoord. One axis is the "divergences," the differences beween the parametric and nonparametric estimates of the population regression function, while the other axes are the predictor variables. Note that the divergences are NOT the parametric model residuals, e.g. differences between fitted model values and response ("Y") values.

The question addressed is, "In what regions of predictor space is the parametric fit poorer?" To answer that, the divergences are grouped into upper and lower tails; e.g. if tail is set to 0.10, we find the data points that have divergences in the lower and upper 10%, then plot both groups, as well as the middle.

The parallel coordinates plot then can be used to identify regions in which the parametric model tends to either under- or overpredict the response, thus indicating possible addition of interaction or polynomial terms.

Furthermore, in the case for regdiag in which an lm object is input, the adjusted R-squared value for the parametric model and the R-squared value from the nonparametric fit are computed. If the nonparametric value is substantially larger than the parametric one, this is an indication of some deficiency in the parametric model, thus providing some quantitative information on whether inclusion of interaction and/or polynomial terms may be useful.

The term *regression* is used in the sense of condtional mean response given the predictors. Thus parametric classification models such as the logistic may also be used, with the regression function being the condtional probability of response = 1, given the predictors.

Usage

```
regdiag(regout, tail=0.10, k=NULL, m=5,
      checkna = TRUE, cls = NULL, nchunks = length(cls))
regdiagbas(preds, resp, parest, tail=0.10, k=NULL, m=5,
      checkna = TRUE, cls = NULL, nchunks = length(cls))
```

Arguments

regout	Output of lm or glm
preds	Matrix of predictor values.
resp	Vector of response values.
parest	Parametric model estimates of the population regression function at the predictor data points.
tail	Proportion of most negative and most positive divergences to use in grouping.
k	See freqparcoord.
m	See freqparcoord.
checkna	See freqparcoord.
cls	See freqparcoord.
nchunks	See freqparcoord.

Details

The population regression function (including the case of a probability function in a classification problem) is estimated nonparametrically at the observation points, using knnreg.

The nonparametric estimates are subtracted from the parametric ones, yielding the divergences. A frequency-parallel coordinates plot is displayed as described above.

The R-squared values are available in the situation noted earlier. The nonparametric R-squared value is calculated as the squared correlation between estimated regression value and the response value.

It is possible that in one of the tail groups the response value is constant, in which case an error message appears. If so, try a larger value of tail.

Value

An object of type "gg" (a **ggplot2** object, displays when printed), with new components added:

- The nonparametric regression estimates, in nonparest.
- In the case of a linear model specified via regout, the adjusted R-squared value for the parametric model, in paramr2, and nonparamr2, the R-squared value from the nonparametric fit. The latter is the squared correlation between predicted and actual response values

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu> and Yingkang Xie <yingkang.xie@gmail.com>

Examples

```
data(mlb)

lmout <- lm(mlb$Weight ~ mlb$Height + mlb$Age)
p <- regdiag(lmout,0.10,k=50,m=25)
p
```

```
# taller, older players are overpredicted, with shorter, younger players
# underpredicted; suggests that adding quadratic terms for Height, Age
# may help in the tails
# let's compare the R-squared values
p$paramr2
p$nonparamr2
# not much difference (param. model a bit better), possibly due to
# small sample size

# doing it "the long way" (showing use without an lm/glm object)
parest <- lmout$fitted.values
regdiagbas(mlb[c("Height","Age")], mlb$Weight,parest,0.10,k=50,m=25)

data(prgeng)
pg <- prgeng
pg1 <- pg[pg$wageinc >= 40000 & pg$wkswrkd >= 48,]
l1 <- lm(wageinc ~ age+educ+sex,data=pg1)
p <- regdiag(l1)
p
p$paramr2
p$nonparamr2
# young men's wages underpredicted, older women overpredicted; both
# R-squared values low, but nonpar is about 27% higher, indicating room
# for improvement; interaction and polynomial terms may help

## Not run:
data(newadult)
g1 <- glm(gt50 ~ edu + age + gender + mar, data=newadult, family=binomial)
regdiag(g1)
# parametric model underpredicts older highly-educated married men,
# and overpredicts young female lesser-educated singles; might try adding
# interaction terms

## End(Not run)
```

`rmixmvnorm` *Random vectors from mixtures of multivariate normal distributions.*

Description

Generates random vectors from mixtures of multivariate normal distributions.

Usage

```
rmixmvnorm(n,dm,nmix,means,covs,wts=rep(1/nmix,nmix))
```

Arguments

`n`	Number of random vectors to generate.
`dm`	Dimension i.e. length of each random vector.

nmix	Number of components in the mixture.
means	Mean vectors of the MV normal distributions; an R list of nmix vectors of length dm each
covs	Covariance matrices of the MV normal distributions; an R list of nmix , each dm x dm.
wts	Mixture probabilities.

Value

An n by dm matrix of random vectors of length dm, grouped by MV normal distribution of origin.

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu>

---

smoothz *Smoothing functions.*

Description

Routines for k-Nearest Neighbor density and regression estimation, optionally using parallel computation.

Usage

```
smoothz(z,sf,k,checkna=TRUE,cls=NULL,nchunks=length(cls),scalefirst=FALSE)
smoothzpred(newx,oldx,oldxregest,checkna=TRUE,cls=NULL,nchunks=length(cls))
knnreg(data,k)
knndens(data,k)
```

Arguments

z	The data, in data frame or matrix form. In the regression case, the response variable is assumed to be in the last column.
sf	Smoothing function (unquoted), knnreg for regression or knndens for density estimation.
k	Number of nearest neighbors.
nchunks	Number of chunks to break the computation into.
newx	New X data to predict from
oldx	X-variable values in the training set.
oldxregest	Estimated regression values in the training set.
checkna	If TRUE, remove any row having at least one NA value.
cls	Cluster to use (see the parallel package) for parallel computation.
data	Data to be smoothed.
scalefirst	Apply scale to the data before smoothing.

Details

The smoothed values are calculated at the input data points (needed in this form for another application). So, for instance, the i-th value of the output of `smoothz` in the regression case is the estimated regression function at the i-th row of `z`.

The density estimates are not mormalized to having total hypervolume equal to 1.0.

In the case of non-null `nchunks`, smoothing is done within-chunk only. The smoothed value at a point will be computed only from its neighbors in the point's chunk.

The `smoothzpred` function applies only to the regression case. It is assumed that `smoothz` has been previously called on `oldx`, yielding regression function estimates `oldxregest` at those points. The `smoothzpred` function then finds, for each point `newx[i]`, the closest point `oldx[j]` in `oldx`, and uses the corresponding value `oldxregest[j]` as the predicted value at `newx[i]`.

Value

Vector of smoothed values, or in the case of `smoothzpred`, vector of predicted Y values for `newx`.

Author(s)

Norm Matloff <matloff@cs.ucdavis.edu> and Yingkang Xie <yingkang.xie@gmail.com>

Examples

```
# programmers and engineers in Silicon Valley, 2000 census, age 25-65
data(prgeng)
pg <- prgeng
pg1 <- pg[pg$age >= 25 & pg$age <= 65,]
estreg <- smoothz(pg1[,c(1,8)],sf=knnreg,k=100)
age <- pg1[,1]
p <- ggplot(data.frame(age,estreg))
p + geom_smooth(aes(x=age,y=estreg))
# peak earnings appear to occur around age 45
```

Index

freqparcoord, 2, *8*, *9*

jitter, *7*

knndens, *3*
knndens *(*smoothz*)*, 11
knnreg, *9*
knnreg *(*smoothz*)*, 11

mlb, 5

newadult, 6

oliveoils, 6

posjitter, 7
prgeng, 7

regdiag, 8
regdiagbas *(*regdiag*)*, 8
rmixmvnorm, 10

scale, *11*
smoothz, 11
smoothzpred *(*smoothz*)*, 11