Package 'ordPens'

August 29, 2013

Type Package

Title Selection and/or Smoothing of Ordinal Predictors

Version 0.2-1

Date 2013-06-12

Author Jan Gertheiss

Maintainer Jan Gertheiss <jgerthe@uni-goettingen.de>

Description Selection and/or smoothing of ordinally scaled independent
variables using a group lasso or generalized ridge penalty

Depends grplasso, mgcv, RLRsim

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-06-14 12:23:13

R topics documented:

1

---

ordPens-package   *Selection and/or Smoothing of Ordinal Predictors*

Description

Selection and/or smoothing of ordinally scaled independent variables using a group lasso or generalized ridge penalty.

Details

Package:	ordPens
Type:	Package
Version:	0.2-1
Date:	2013-06-12
Depends:	grplasso, mgcv, RLRsim
License:	GPL-2
LazyLoad:	yes

Smoothing and selection of ordinal predictors is done by the function ordSelect; smoothing only, by ordSmooth. For ANOVA with ordinal factors, use ordAOV.

Author(s)

Jan Gertheiss

Maintainer: Jan Gertheiss, <jgerthe@uni-goettingen.de>

References

Gertheiss, J., S. Hogger, C. Oberhauser and G. Tutz (2011). *Selection of ordinally scaled independent variables with applications to international classification of functioning core sets*. Journal of the Royal Statistical Society C (Applied Statistics), 60, 377-395.

Gertheiss, J. and F. Oehrlein (2011). *Testing Relevance and Linearity of Ordinal Predictors*, Electronic Journal of Statistics, 5, 1935-1959.

Gertheiss, J. and G. Tutz (2009). *Penalized regression with ordinal predictors*. International Statistical Review, 77, 345-365.

See Also

ordSelect, ordSmooth, ordAOV

Examples

```
# smooth modeling of a random dataset
set.seed(123)
```

```
# generate (ordinal) predictors
x1 <- sample(1:8,100,replace=TRUE)
x2 <- sample(1:6,100,replace=TRUE)
x3 <- sample(1:7,100,replace=TRUE)

# the response
y <- -1 + log(x1) + sin(3*(x2-1)/pi) + rnorm(100)

# x matrix
x <- cbind(x1,x2,x3)

# lambda values
lambda <- c(1000,500,200,100,50,30,20,10,1)

# smooth modeling
o1 <- ordSmooth(x = x, y = y, lambda = lambda)

# results
round(o1$coef,digits=3)
plot(o1)

# If for a certain plot the x-axis should be annotated in a different way,
# this can (for example) be done as follows:
plot(o1, whichx = 1, xlim = c(0,9), xaxt = "n")
axis(side = 1, at = c(1,8), labels = c("no agreement","total agreement"))
```

ICFCoreSetCWP *ICF Core Set for Chronic Widespread Pain*

Description

The data set contains observed levels of ICF categories from the (comprehensive) ICF Core Set for chronic wide spread pain (CWP) and a physical health component summary measure for n = 420 patients.

Usage

```
data(ICFCoreSetCWP)
```

Format

The data frame has 420 rows and 68 columns. The first 67 columns contain observed levels of ICF categories from the (comprehensive) ICF Core Set for chronic wide spread pain (CWP). In the last column, the physical health component summary measure is given. Each row corresponds to one patient with CWP. ICF categories have discrete ordinal values between 0 and 4 (columns 1 - 50 and 67), or between -4 and 4 (columns 51 - 66). See the given references for details.

Details

The original data set contained some missing values, which have been imputed using R package `Amelia`.

The data were collected within the study *Validation of ICF Core Sets for chronic conditions*, which was a collaboration effort between the ICF Research Branch of the collaborating centers for the Family of International Classifications in German, the Classification, Terminology and standards Team from the World Health Organization and the International Society for Physical and Rehabilitation Medicine.

Special thanks go to the following participating study centers: Ankara University, Turkey; Azienda Ospedaliera di Sciacca, Italy; Donauspital, Vienna, Austria; Drei-Burgen-Klinik, Bad Muenster, Germany; Edertal Klinik, Bad Wildungen, Germany; Fachklinik Bad Bentheim, Germany; Hospital das Clinicas, School of Medicine, University of Sao Paulo, Brazil; Hospital San Juan Bautista, Catamarca, Argentina; Istituto Scientifico di Montescano, Italy; Istituto Scientifico di Veruno, Italy; Kaiser-Franz-Josef-Spital, Vienna, Austria; Klinik am Regenbogen, Nittenau, Germany; Klinik Bavaria Kreischa, Germany; Klinik Hoher Meissner, Bad Sooden-Allendorf, Germany; Klinikum Berchtesgadener Land, Schoenau, Germany; Kuwait Physical Medicine and Rehabilitation Society, Safat, Kuwait; National Institute for Medical Rehabilitation, Budapest, Hungary; Neuro-Orthopaedisches Krankenhaus und Zentrum fuer Rehabilitative Medizin Soltau, Germany; Praxis fuer Physikalische Medizin und Rehabilitation, Goettingen, Germany; Rehabilitationsklinik Seehof der Bundesversicherungsanstalt fuer Angestellte, Teltow, Germany; Rehaklinik Rheinfelden, Switzerland; Spanish Society of Rheumatology, Madrid, Spain; University Hospital Zurich, Switzerland; University of Santo Tomas, Quelonchy, Philippines.

Most special thanks go to all the patients participating in the study.

If you use the data, please cite the following two references.

References

Cieza, A., G. Stucki, M. Weigl, L. Kullmann, T. Stoll, L. Kamen, N. Kostanjsek, and N. Walsh (2004). *ICF Core Sets for chronic widespread pain*. Journal of Rehabilitation Medicine, Suppl. 44, 63-68.

Gertheiss, J., S. Hogger, C. Oberhauser and G. Tutz (2011). *Selection of ordinally scaled independent variables with applications to international classification of functioning core sets*. Journal of the Royal Statistical Society C (Applied Statistics), 60, 377-395.

Examples

```
# load the data
data(ICFCoreSetCWP)

# available variables
names(ICFCoreSetCWP)

# adequate coding of x matrix (using levels 1,2,...)
p <- ncol(ICFCoreSetCWP) - 1
n <- nrow(ICFCoreSetCWP)
add <- c(rep(1,50),rep(5,16),1)
add <- matrix(add,n,p,byrow=TRUE)
x <- ICFCoreSetCWP[,1:p] + add
```

```
# make sure that also a coefficient is fitted for levels
# that are not observed in the data
addrow <- c(rep(5,50),rep(9,16),5)
x <- rbind(x,addrow)
y <- c(ICFCoreSetCWP$phcs,NA)

# some lambda values
lambda <- c(600,500,400,300,200,100)

# smoothing and selection
modelICF <- ordSelect(x = x, y = y, lambda = lambda)

# results
plot(modelICF)

# plot a selected ICF category (e.g. e1101 'drugs')
# with adequate class labels
plot(modelICF, whichx = 51, xaxt = "n")
axis(side = 1, at = 1:9, labels = -4:4)
```

---

ordAOV   *ANOVA for factors with ordered levels*

Description

This function performs analysis of variance when the factor(s) of interest has/have ordinal scale level. For testing, values from the null distribution are simulated.

Usage

```
ordAOV(x, ...)

## Default S3 method:
ordAOV(x, y, type = c("RLRT", "LRT"), nsim = 10000, ...)
```

Arguments

x	a vector or matrix of integers 1,2,... giving the observed levels of the ordinal factor(s). If x is a matrix, it is assumed that each column corresponds to one ordinal factor.
y	the vector of response values.
type	the type of test to carry out: likelihood ratio ("LRT") or restricted likelihood ratio ("RLRT").
nsim	number of values to simulate from the null distribution.
...	additional arguments to LRTSim and RLRTSim, respectively.

Details

The method assumes that ordinal factor levels (contained in vector/columns of matrix x) take values 1,2,...,max, where max denotes the highest level of the respective factor observed in the data. Every level between 1 and max has to be observed at least once.

The method uses a mixed effects formulation of the usual one- or multi-factorial ANOVA model (with main effects only) while penalizing (squared) differences of adjacent means. Testing for equal means across factor levels is done by (restricted) likelihood ratio testing for a zero variance component in a linear mixed model. For simulating values from the finite sample null distribution of the (restricted) likelihood ratio statistic, the algorithms implemented in Package RLRsim are used. See LRTSim and RLRTSim for further information.

If x is a vector (or one-column matrix), one-factorial ANOVA is applied, and it is simulated from the exact finite sample null distribution as derived by Crainiceanu & Ruppert (2004). If x is a matrix, multi-factorial ANOVA (with main effects only) is done, and the approximation of the finite sample null distribution proposed by Greven et al. (2008) is used. Simulation studies by Gertheiss (2013) suggest that for ANOVA with ordinal factors RLRT should rather be used than LRT.

Value

In case of one-factorial ANOVA, a list of class htest containing the following components (see also exactLRT and exactRLRT):

statistic	the observed (restricted) likelihood ratio.
p	p-value for the observed test statistic.
method	a character string indicating what type of test was performed and how many values were simulated to determine the critical value.
sample	the samples from the null distribution returned by LRTSim and RLRTSim, respectively.

In case of multi-factorial ANOVA, a list (of lists) with the jth component giving the results above when testing the main effect of factor j.

Author(s)

Jan Gertheiss

References

Crainiceanu, C. and D. Ruppert (2004). *Likelihood ratio tests in linear mixed models with one variance component*, Journal of the Royal Statistical Society B, 66, 165-185.

Gertheiss, J. (2013). *ANOVA for Factors with Ordered Levels*, Preprint. (available on request)

Gertheiss, J. and F. Oehrlein (2011). *Testing Relevance and Linearity of Ordinal Predictors*, Electronic Journal of Statistics, 5, 1935-1959.

Greven, S., C. Crainiceanu, H. Kuechenhoff, and A. Peters (2008). *Restricted likelihood ratio testing for zero variance components in linear mixed models*, Journal of Computational and Graphical Statistics, 17, 870-891.

Scheipl, F., S. Greven, and H. Kuechenhoff (2008). *Size and power of tests for a zero random effect variance or polynomial regression in additive and linear mixed models*, Computational Statistics & Data Analysis, 52, 3283-3299.

See Also

[LRTSim](), [RLRTSim]()

Examples

```
# load some data
data(ICFCoreSetCWP)

# the pysical health component summary
y <- ICFCoreSetCWP$phcs

# consider the first ordinal factor
x <- ICFCoreSetCWP[,1]

# adequate coding
x <- as.integer(x - min(x) + 1)

# ANOVA
ordAOV(x, y, type = "RLRT", nsim=1000000)
```

---

ordSelect   *Selection and smoothing of dummy coefficients of ordinal predictors*

Description

Fits dummy coefficients of ordinally scaled independent variables with a group lasso penalty on differences of adjacent dummy coefficients.

Usage

```
ordSelect(x, ...)

## Default S3 method:
ordSelect(x, y, u = NULL, z = NULL,
  offset = rep(0,length(y)), lambda, nu = 1, zeta = 1,
  model = c("linear", "logit", "poisson"), penscale = sqrt,
  scalex = TRUE, scalez = TRUE, scaleu = TRUE,
  nonpenx = NULL, nonpenz = NULL, nonpenu = NULL,
  intercept = TRUE, eps = 1e-3, ...)
```

Arguments

x	the matrix of ordinal predictors, with each column corresponding to one predictor and containing numeric values from {1,2,...}; for each covariate, category 1 is taken as reference category with zero dummy coefficient.
y	the response vector.

u	a matrix (or `data.frame`) of additional categorical (nominal) predictors, with each column corresponding to one (additional) predictor and containing numeric values from {1,2,...}; corresponding dummy coefficients will be regularized using a groupwise simple Ridge penalty, and for each covariate category 1 is taken as reference category.
z	a matrix (or `data.frame`) of additional metric predictors, with each column corresponding to one (additional) predictor; corresponding coefficients will be regularized using a simple Lasso penalty.
offset	vector of offset values.
lambda	vector of penalty parameters (in decreasing order). Optimization starts with the first component. See details below.
nu	additional tuning parameter to control the strength of the penalty imposed on dummy coefficients corresponding to u. See details below.
zeta	additional tuning parameter to control the strength of the penalty imposed on coefficients corresponding to z. See details below.
model	the model which is to be fitted. Possible choices are "linear" (default), "logit" or "poisson". See details below.
penscale	rescaling function to adjust the value of the penalty parameter to the degrees of freedom of the parameter group. See the references below.
scalex, scaleu, scalez	
	logical. Should (split/dummy-coded) design matrices corresponding to x, u, resp. z be scaled to have unit variance over columns? See details below.
nonpenx, nonpenu, nonpenz	
	vectors of indices indicating columns of x, u, resp. z whose regression coefficients are not penalized.
intercept	logical. Should a (non-penalized) intercept be included in the model? Default is TRUE.
eps	a (small) constant to be added to the columnwise standard deviations when scaling design matrices, to control the effect of very small stds. See details below.
...	additional arguments.

Details

The method assumes that categorical covariates (contained in x and u) take values 1,2,...,max, where max denotes the (columnwise) highest level observed in the data. If any level between 1 and max is not observed, a corresponding (dummy) coefficient is fitted anyway. If any level > max is not observed but possible in principle, and a corresponding coefficient is to be fitted, the easiest way is to add a corresponding row to x (and u,z) with corresponding y value being NA.

In order to adjust the strength of penalty that is imposed on nominal or metric covariates, nu, resp. zeta can be chosen. The penalty parameter used for nominal variables (contained in u) is `lambda*nu`, for metric predictors (contained in z) it is `lambda*zeta`.

If a linear regression model is fitted, response vector y may contain any numeric values; if a logit model is fitted, y has to be 0/1 coded; if a poisson model is fitted, y has to contain count data.

If scalex, scaleu or scalez are TRUE, design matrices constructed from x, u, resp. z are scaled to have unit variance over columns. In the case of x, the design matrix is split-coded, in the case

of u, it is dummy-coded, and in case of z, it is just z. If a certain x- or u- category, however, is observed only a few times, variances may become very small and scaling has enormous effects on the result and may cause numerical problems. Hence a small constant eps can be added to each standard deviation when used for scaling.

Value

An ordPen object, which is a list containing:

fitted	the matrix of fitted response values of the training data. Columns correspond to different lambda values.
coefficients	the matrix of fitted coefficients with respect to dummy-coded (ordinal or nominal) categorical input variables (including the reference category) as well as metric predictors. Columns correspond to different lambda values.
model	the type of the fitted model: "linear", "logit", or "poisson".
lambda	the used lambda values.
xlevels	a vector giving the number of levels of the ordinal predictors.
ulevels	a vector giving the number of levels of the nominal predictors.
zcovars	the number of metric covariates.

Author(s)

Jan Gertheiss

References

Gertheiss, J., S. Hogger, C. Oberhauser and G. Tutz (2011). *Selection of ordinally scaled independent variables with applications to international classification of functioning core sets*. Journal of the Royal Statistical Society C (Applied Statistics), 60, 377-395.

Meier, L., S. van de Geer and P. Buehlmann (2008). *The group lasso for logistic regression*. Journal of the Royal Statistical Society B, 70, 53-71.

Yuan, M. and Y. Lin (2006). *Model selection and estimation in regression with grouped variables*. Journal of the Royal Statistical Society B, 68, 49-67.

See Also

[plot.ordPen](#), [predict.ordPen](#), [ICFCoreSetCWP](#)

Examples

```
# smoothing and selection of ordinal covariates using a random dataset
set.seed(123)

# generate (ordinal) predictors
x1 <- sample(1:8,100,replace=TRUE)
x2 <- sample(1:6,100,replace=TRUE)
x3 <- sample(1:7,100,replace=TRUE)
```

```
# the response
y <- -1 + log(x1) + sin(3*(x2-1)/pi) + rnorm(100)

# x matrix
x <- cbind(x1,x2,x3)

# lambda values
lambda <- c(1000,500,200,100,50,30,20,10,1)

# smoothing and selection
o2 <- ordSelect(x = x, y = y, lambda = lambda)

# results
round(o2$coef,digits=3)
plot(o2)

# If for a certain plot the x-axis should be annotated in a different way,
# this can (for example) be done as follows:
plot(o2, whichx = 1, xlim = c(0,9), xaxt = "n")
axis(side = 1, at = c(1,8), labels = c("no agreement","total agreement"))
```

---

ordSmooth   *Smoothing dummy coefficients of ordinal predictors*

Description

Fits dummy coefficients of ordinally scaled independent variables with the sum of squared differences of adjacent dummy coefficients being penalized.

Usage

```
ordSmooth(x, ...)

## Default S3 method:
ordSmooth(x, y, u = NULL, z = NULL,
  offset = rep(0,length(y)), lambda, nu = 1, zeta = 1,
  model = c("linear", "logit", "poisson"), penscale = identity,
  scalex = TRUE, scalez = TRUE, scaleu = TRUE,
  nonpenx = NULL, nonpenz = NULL, nonpenu = NULL,
  intercept = TRUE, eps = 1e-3, delta = 1e-6, maxit = 25, ...)
```

Arguments

x	the matrix (or `data.frame`) of ordinal predictors, with each column corresponding to one predictor and containing numeric values from {1,2,...}; for each covariate, category 1 is taken as reference category with zero dummy coefficient.
y	the response vector.

u	a matrix (or `data.frame`) of additional categorical (nominal) predictors, with each column corresponding to one (additional) predictor and containing numeric values {1,2,...}; corresponding dummy coefficients will be regularized using a simple Ridge penalty, and for each covariate category 1 is taken as reference category.
z	a matrix (or `data.frame`) of additional metric predictors, with each column corresponding to one (additional) predictor; corresponding coefficients will be regularized using a simple Ridge penalty.
offset	vector of offset values.
lambda	vector of penalty parameters (in decreasing order). Optimization starts with the first component. See details below.
nu	additional tuning parameter to control the strength of the penalty imposed on dummy coefficients corresponding to u. See details below.
zeta	additional tuning parameter to control the strength of the penalty imposed on coefficients corresponding to z. See details below.
model	the model which is to be fitted. Possible choices are "linear" (default), "logit" or "poisson". See details below.
penscale	rescaling function to adjust the value of the penalty parameter to the degrees of freedom of the parameter group.
scalex, scaleu, scalez	
	logical. Should (split/dummy-coded) design matrices corresponding to x, u, resp. z be scaled to have unit variance over columns? See details below.
nonpenx, nonpenu, nonpenz	
	vectors of indices indicating columns of x, u, resp. z whose regression coefficients are not penalized.
intercept	logical. Should a (non-penalized) intercept be included in the model? Default is TRUE.
eps	a (small) constant to be added to the columnwise standard deviations when scaling design matrices, to control the effect of very small stds. See details below.
delta	a small positive convergence tolerance which is used as stopping criterion for the penalized Fisher scoring when a logit or poisson model is fitted. See details below.
maxit	integer given the maximal number of (penalized) Fisher scoring iterations.
...	additional arguments.

Details

The method assumes that categorical covariates (contained in x and u) take values 1,2,...,max, where max denotes the (columnwise) highest level observed in the data. If any level between 1 and max is not observed, a corresponding (dummy) coefficient is fitted anyway. If any level > max is not observed but possible in principle, and a corresponding coefficient is to be fitted, the easiest way is to add a corresponding row to x (and u,z) with corresponding y value being NA.

In order to adjust the strength of penalty that is imposed on nominal or metric covariates, nu, resp. zeta can be chosen. The penalty parameter used for nominal variables (contained in u) is lambda*nu, for metric predictors (contained in z) it is lambda*zeta.

If a linear regression model is fitted, response vector y may contain any numeric values; if a logit model is fitted, y has to be 0/1 coded; if a poisson model is fitted, y has to contain count data.

If `scalex`, `scaleu` or `scalez` are `TRUE`, design matrices constructed from x, u, resp. z are scaled to have unit variance over columns. In the case of x, the design matrix is split-coded, in the case of u, it is dummy-coded, and in case of z, it is just z. If a certain x- or u- category, however, is observed only a few times, variances may become very small and scaling has enormous effects on the result and may cause numerical problems. Hence a small constant eps can be added to each standard deviation when used for scaling.

A logit or poisson model is fitted by penalized Fisher scoring. For stopping the iterations the criterion `sqrt(sum((b.new-b.old)^2)/sum(b.old^2)) < delta` is used.

Value

An `ordPen` object, which is a list containing:

`fitted`	the matrix of fitted response values of the training data. Columns correspond to different `lambda` values.
`coefficients`	the matrix of fitted coefficients with respect to dummy-coded (ordinal or nominal) categorical input variables (including the reference category) as well as metric predictors. Columns correspond to different lambda values.
`model`	the type of the fitted model: "linear", "logit", or "poisson".
`lambda`	the used lambda values.
`xlevels`	a vector giving the number of levels of the ordinal predictors.
`ulevels`	a vector giving the number of levels of the nominal predictors.
`zcovars`	the number of metric covariates.

Author(s)

Jan Gertheiss

References

Gertheiss, J. and G. Tutz (2009). *Penalized regression with ordinal predictors*. International Statistical Review, 77, 345-365.

See Also

plot.ordPen, predict.ordPen

Examples

```
# smooth modeling of a random dataset
set.seed(123)

# generate (ordinal) predictors
x1 <- sample(1:8,100,replace=TRUE)
x2 <- sample(1:6,100,replace=TRUE)
x3 <- sample(1:7,100,replace=TRUE)
```

```
# the response
y <- -1 + log(x1) + sin(3*(x2-1)/pi) + rnorm(100)

# x matrix
x <- cbind(x1,x2,x3)

# lambda values
lambda <- c(1000,500,200,100,50,30,20,10,1)

# smooth modeling
o1 <- ordSmooth(x = x, y = y, lambda = lambda)

# results
round(o1$coef,digits=3)
plot(o1)

# If for a certain plot the x-axis should be annotated in a different way,
# this can (for example) be done as follows:
plot(o1, whichx = 1, xlim = c(0,9), xaxt = "n")
axis(side = 1, at = c(1,8), labels = c("no agreement","total agreement"))
```

plot.ordPen	*Plot method for ordPen objects*

Description

Takes a fitted `ordPen` object and plots estimated dummy coefficients of ordinal predictors for different `lambda` values.

Usage

```
## S3 method for class 'ordPen'
plot(x, whichlam = NULL, whichx = NULL,
  type = NULL, xlab = NULL, ylab = NULL, main = NULL,
  xlim = NULL, ylim = NULL, col = NULL, ...)
```

Arguments

x	an ordPen object.
whichlam	a vector of indices of `lambda` values corresponding to `object$lambda` for which plotting is done; if NULL, all values from `object$lambda` are considered.
whichx	a vector of indices indicating the ordinal predictors whose dummy coefficients are plotted; e.g., set `whichx=2`, if you just want the plot for the second smooth term.
type	1-character string giving the type of plot desired, see plot.default.
xlab	a label for the x axis; if supplied then this will be used as the x label for all plots.

ylab	a label for the y axis; if supplied then this will be used as the y label for all plots.
main	a main title for the plot(s); if supplied then this will be used as the title for all plots.
xlim	the x limits; if supplied then this pair of numbers are used as the x limits for each plot.
ylim	the y limits; if supplied then this pair of numbers are used as the y limits for each plot.
col	the plotting color; can be a vector of the same length as whichlam specifying different colors for different lambda values. Default is shades of gray: the higher lambda the darker.
...	additional graphical parameters (see plot.default, or par).

Value

The function simply generates plots.

Author(s)

Jan Gertheiss

See Also

ordSelect, ordSmooth

Examples

```
# see for example
help(ordSelect)
```

---

predict.ordPen	*Predict method for ordPen objects*

Description

Obtains predictions from an ordPen object.

Usage

```
## S3 method for class 'ordPen'
predict(object, newx, newu = NULL, newz = NULL,
  offset = rep(0,nrow(as.matrix(newx))),
  type = c("link", "response"), ...)
```

Arguments

`object`	an ordPen object.
`newx`	the matrix (or `data.frame`) of new observations of the considered ordinal predictors, with each column corresponding to one predictor and containing numeric values from {1,2,...}.
`newu`	a matrix (or `data.frame`) of new observations of the additional categorical (nominal) predictors, with each column corresponding to one (additional) predictor and containing numeric values {1,2,...}.
`newz`	a matrix (or `data.frame`) of new observations of the additional metric predictors, with each column corresponding to one (additional) predictor.
`offset`	potential offset values.
`type`	the type of prediction. `type = "link"` is on the scale of linear predictors, whereas `type = "response"` is on the scale of the response variable, i.e. `type = "response"` applies the inverse link function to the linear predictors.
`...`	additional arguments.

Value

A matrix of predictions whose columns correspond to the different values of the penalty parameter `lambda` of the ordPen object.

Author(s)

Jan Gertheiss

See Also

[ordSelect](#), [ordSmooth](#)

Examples

```
# the training data
set.seed(123)

# generate (ordinal) predictors
x1 <- sample(1:8,100,replace=TRUE)
x2 <- sample(1:6,100,replace=TRUE)
x3 <- sample(1:7,100,replace=TRUE)

# the response
y <- -1 + log(x1) + sin(3*(x2-1)/pi) + rnorm(100)

# x matrix
x <- cbind(x1,x2,x3)

# lambda values
lambda <- c(1000,500,200,100,50,30,20,10,1)
```

```
# selecting and/or smoothing
o1 <- ordSmooth(x = x, y = y, lambda = lambda)
o2 <- ordSelect(x = x, y = y, lambda = lambda)

# new data
x1 <- sample(1:8,10,replace=TRUE)
x2 <- sample(1:6,10,replace=TRUE)
x3 <- sample(1:7,10,replace=TRUE)
newx <- cbind(x1,x2,x3)

# prediction
round(predict(o1, newx), digits=3)
round(predict(o2, newx), digits=3)
```

Index