

Package ‘kosel’

October 13, 2022

Title Variable Selection by Revisited Knockoffs Procedures

Version 0.0.1

Description Performs variable selection for many types of L1-regularised regressions using the revisited knockoffs procedure. This procedure uses a matrix of knockoffs of the covariates independent from the response variable Y . The idea is to determine if a covariate belongs to the model depending on whether it enters the model before or after its knockoff. The procedure suits for a wide range of regressions with various types of response variables. Regression models available are exported from the R packages 'glmnet' and 'ordinalNet'. Based on the paper linked to via the URL below: Gegout A., Gueudin A., Karmann C. (2019) <[arXiv:1907.03153](https://arxiv.org/pdf/1907.03153)>.

URL <https://arxiv.org/pdf/1907.03153.pdf>

License GPL-3

Depends R (>= 1.1)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports glmnet, ordinalNet

Suggests graphics

NeedsCompilation no

Author Clemence Karmann [aut, cre],
Aurelie Gueudin [aut]

Maintainer Clemence Karmann <clemence.karmann@gmail.com>

Repository CRAN

Date/Publication 2019-07-18 10:44:06 UTC

R topics documented:

ko.glm	2
ko.ordinal	3
ko.sel	4
Index	8

 ko.glm

Statistics of the knockoffs procedure for glmnet regression models.

Description

Returns the vector of statistics W of the revisited knockoffs procedure for regressions available in the R package `glmnet`. Most of the parameters come from `glmnet()`. See [glmnet documentation](#) for more details.

Usage

```
ko.glm(x, y, family = "gaussian", alpha = 1,
       type.gaussian = ifelse(nvars < 500, "covariance", "naive"),
       type.logistic = "Newton", type.multinomial = "ungrouped",
       nVal = 50, random = FALSE)
```

Arguments

<code>x</code>	Input matrix, of dimension <code>nobs</code> x <code>nvars</code> ; each row is an observation vector. Can be in sparse matrix format (inherit from class <code>"sparseMatrix"</code> as in package <code>Matrix</code> ; not yet available for <code>family="cox"</code>)
<code>y</code>	Response variable. Quantitative for <code>family="gaussian"</code> , or <code>family="poisson"</code> (non-negative counts). For <code>family="binomial"</code> should be either a factor with two levels, or a two-column matrix of counts or proportions (the second column is treated as the target class; for a factor, the last level in alphabetical order is the target class). For <code>family="multinomial"</code> , can be a <code>nc >= 2</code> level factor, or a matrix with <code>nc</code> columns of counts or proportions. For either <code>"binomial"</code> or <code>"multinomial"</code> , if <code>y</code> is presented as a vector, it will be coerced into a factor. For <code>family="cox"</code> , <code>y</code> should be a two-column matrix with columns named <code>'time'</code> and <code>'status'</code> . The latter is a binary variable, with <code>'1'</code> indicating death, and <code>'0'</code> indicating right censored. The function <code>Surv()</code> in package <code>survival</code> produces such a matrix.
<code>family</code>	Response type: <code>"gaussian"</code> , <code>"binomial"</code> , <code>"poisson"</code> , <code>"multinomial"</code> , <code>"cox"</code> . Not available for <code>"mgaussian"</code> .
<code>alpha</code>	The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$. <code>alpha=1</code> is the lasso penalty, and <code>alpha=0</code> the ridge penalty. The default is 1.
<code>type.gaussian</code>	See <code>glmnet</code> documentation.
<code>type.logistic</code>	See <code>glmnet</code> documentation.
<code>type.multinomial</code>	See <code>glmnet</code> documentation.
<code>nVal</code>	Length of lambda sequence - default is 50.
<code>random</code>	If TRUE, the matrix of knockoffs is different for every run. If FALSE, a seed is used so that the knockoffs are the same. The default is FALSE.

Value

A vector of dimension nvars corresponding to the statistics W.

See Also

[ko.sel](#)

Examples

```
# see ko.sel
```

 ko.ordinal

Statistics of the knockoffs procedure for ordinalNet regression models.

Description

Returns the vector of statistics W of the revisited knockoffs procedure for regressions available in the R package ordinalNet. Most of the parameters come from ordinalNet(). See [ordinalNet documentation](#) for more details.

Usage

```
ko.ordinal(x, y, family = "cumulative", reverse = FALSE,
  link = "logit", alpha = 1, parallelTerms = TRUE,
  nonparallelTerms = FALSE, nVal = 100, warn = FALSE,
  random = FALSE)
```

Arguments

- | | |
|---------|---|
| x | Covariate matrix, of dimension nobs x nvars; each row is an observation vector. It is recommended that categorical covariates are converted to a set of indicator variables with a variable for each category (i.e. no baseline category); otherwise the choice of baseline category will affect the model fit. |
| y | Response variable. Can be a factor, ordered factor, or a matrix where each row is a multinomial vector of counts. A weighted fit can be obtained using the matrix option, since the row sums are essentially observation weights. Non-integer matrix entries are allowed. |
| family | Specifies the type of model family. Options are "cumulative" for cumulative probability, "sratio" for stopping ratio, "cratio" for continuation ratio, and "acat" for adjacent category. |
| reverse | Logical. If TRUE, then the "backward" form of the model is fit, i.e. the model is defined with response categories in reverse order. For example, the reverse cumulative model with K+1 response categories applies the link function to the cumulative probabilities $P(Y \geq 2)$, \dots , $P(Y \geq K+1)$, rather than $P(Y \leq 1)$, \dots , $P(Y \leq K)$. |

link	Specifies the link function. The options supported are logit, probit, complementary log-log, and cauchit.
alpha	The elastic net mixing parameter, with $0 \leq \alpha \leq 1$. $\alpha=1$ corresponds to the lasso penalty, and $\alpha=0$ corresponds to the ridge penalty.
parallelTerms	Logical. If TRUE, then parallel coefficient terms will be included in the model. parallelTerms and nonparallelTerms cannot both be FALSE.
nonparallelTerms	Logical. If TRUE, then nonparallel coefficient terms will be included in the model. parallelTerms and nonparallelTerms cannot both be FALSE. Default is FALSE. nonparallelTerms = TRUE is highly discouraged.
nVal	Length of lambda sequence - default is 100.
warn	Logical. If TRUE, the following warning message is displayed when fitting a cumulative probability model with nonparallelTerms=TRUE (i.e. nonparallel or semi-parallel model). "Warning message: For out-of-sample data, the cumulative probability model with nonparallelTerms=TRUE may predict cumulative probabilities that are not monotone increasing." The warning is displayed by default, but the user may wish to disable it.
random	If TRUE, the matrix of knockoffs is different for every run. If FALSE, a seed is used so that the knockoffs are the same. The default is FALSE.

Value

A vector of dimension nvars corresponding to the statistics W.

Note

nonparallelTerms = TRUE is highly discouraged because the knockoffs procedure does not suit well to this setting.

See Also

[ko.sel](#)

Examples

```
# see ko.sel
```

ko.sel

Variable selection with the knockoffs procedure.

Description

Performs variable selection from an object (vector of statistics W) returned by [ko.glm](#) or [ko.ordinal](#).

Usage

```
ko.sel(W, print = FALSE, method = "stats")
```

Arguments

W	A vector of length nvars corresponding to the statistics W. Object returned by the functions ko.glm or ko.ordinal.
print	Logical. If TRUE, positive statistics W are displayed in increasing order. If FALSE, nothing is displayed. If method = 'manual', print is automatically TRUE.
method	Can be 'stats', 'gaps' or 'manual'. If 'stats', the threshold used is the W-threshold. If 'gaps', the threshold used is the gaps-threshold. If 'manual', the user can choose its own threshold using the graph of the positive statistics W sorted in increasing order.

Value

A list containing two elements:

- threshold A positive real value corresponding to the threshold used.
- estimation A binary vector of length nvars corresponding to the variable selection: $1*(W \geq \text{threshold})$. 1 indicates that the associated covariate belongs to the estimated model.

References

Gegout-Petit Anne, Gueudin Aurelie, Karmann Clemence (2019). *The revisited knockoffs method for variable selection in L1-penalised regressions*, arXiv:1907.03153.

See Also

[ko.glm](#), [ko.ordinal](#)

Examples

```
library(graphics)

# linear Gaussian regression
n = 100
p = 20
set.seed(11)
x = matrix(rnorm(n*p), nrow = n, ncol = p)
beta = c(rep(1,5), rep(0,15))
y = x%*%beta + rnorm(n)
W = ko.glm(x,y)
ko.sel(W, print = TRUE)

# logistic regression
n = 100
```

```

p = 20
set.seed(11)
x = matrix(runif(n*p, -1,1),nrow = n,ncol = p)
u = runif(n)
beta = c(c(3:1),rep(0,17))
y = rep(0, n)
a = 1/(1+exp(0.1-x**beta))
y = 1*(u>a)
W = ko.glm(x,y, family = 'binomial', nVal = 50)
ko.sel(W, print = TRUE)

# cumulative logit regression
n = 100
p = 10
set.seed(11)
x = matrix(runif(n*p),nrow = n,ncol = p)
u = runif(n)
beta = c(3,rep(0,9))
y = rep(0, n)
a = 1/(1+exp(0.8-x**beta))
b = 1/(1+exp(-0.6-x**beta))
y = 1*(u<a) + 2*((u>=a) & (u<b)) + 3*(u>=b)
W = ko.ordinal(x,as.factor(y), nVal = 20)
ko.sel(W, print = TRUE)

# adjacent logit regression
n = 100
p = 10
set.seed(11)
x = matrix(rnorm(n*p),nrow = n,ncol = p)
U = runif(n)
beta = c(5,rep(0,9))
alpha = c(-2,1.5)
M = 2
y = rep(0, n)
for(i in 1:n){
  eta = alpha + sum(beta*x[i,])
  u = U[i]
  Prob = rep(1,M+1)
  for(j in 1:M){
    Prob[j] = exp(sum(eta[j:M]))
  }
  Prob = Prob/sum(Prob)
  C = cumsum(Prob)
  C = c(0,C)
  j = 1
  while((C[j]> u) || (u >= C[j+1])){j = j+1}
  y[i] = j
}
W = ko.ordinal(x,as.factor(y), family = 'acat', nVal = 10)
ko.sel(W, method = 'manual')

```

0.4

```
# How to use randomness?
n = 100
p = 20
set.seed(11)
x = matrix(rnorm(n*p),nrow = n,ncol = p)
beta = c(5:1,rep(0,15))
y = x%%beta + rnorm(n)
Esti = 0
for(i in 1:100){
  W = ko.glm(x,y, random = TRUE)
  Esti = Esti + ko.sel(W, method = 'gaps')$estimation
}
Esti
```

Index

ko.glm, [2](#), [4](#), [5](#)
ko.ordinal, [3](#), [4](#), [5](#)
ko.sel, [3](#), [4](#), [4](#)