Package ‘KSPM’

August 10, 2020

Title Kernel Semi-Parametric Models

Version 0.2.1

Description To fit the kernel semi-parametric model and its extensions. It allows multiple kernels and unlimited interactions in the same model. Coefficients are estimated by maximizing a penalized log-likelihood; penalization terms and hyperparameters are estimated by minimizing leave-one-out error. It includes predictions with confidence/prediction intervals, statistical tests for the significance of each kernel, a procedure for variable selection and graphical tools for diagnostics and interpretation of covariate effects. Currently it is implemented for continuous dependent variables. The package is based on the paper of Liu et al. (2007), <doi:10.1111/j.1541-0420.2007.00799.x>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports expm, CompQuadForm, DEoptim

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Catherine Schramm [aut, cre],
Aurelie Labbe [ctb],
Celia M. T. Greenwood [ctb]

Maintainer Catherine Schramm <cath.schramm@gmail.com>

Repository CRAN

Date/Publication 2020-08-10 13:32:11 UTC

R topics documented:

`case.names.kspm` *Case names of fitted models*

Description

Simple utility returning names of cases involved in a kernel semi parametric model.

Usage

```
## S3 method for class 'kspm'
case.names(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

a character vector.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, nobs.kspm, variable.names.kspm.

`coef.kspm` *Extract Model Coefficients*

Description

Returns linear and kernel coefficients for a model of class "kspm".

Usage

```
## S3 method for class 'kspm'
coef(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

Two matrices of coefficients.

`linear`	A vector of coefficients for linear part. One row is one variable.
`kernel`	A matrix of coefficients for linear part. One row is one subject, one column is one kernel part.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

See Also

kspm for fitting model.

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
coef(fit)
```

---

confint.kspm — *Confidence interavls for linear part of model parameters*

Description

Computes confidence intervals for one or more parameters in the linear part of a fitted model of class "kspm".

Usage

```
## S3 method for class 'kspm'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`parm`	a vector of names specifying which parameters are to be given confidence intervals. If missing, all parameters are considered.
`level`	the confidence level required. By default 0.95.
`...`	additional optional argument (currently unused).

Details

For objects of class "kspm", the confidence interval is based on student distribution and effective degree of freedom of the model.

Value

A matrix with column giving lower and upper confidence limits for each parameter. These are labelled as $\frac{1-level}{2}$ and $1 - \frac{1-level}{2}$ in percentage.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, summary.kspm.

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
confint(fit)
```

`cooks.distance.kspm` *Cook's distance for a Kernel Semi Parametric Model Fit*

Description

Computes the Cook's distance method for an object of class "kspm".

Usage

```
## S3 method for class 'kspm'
cooks.distance(model, ...)
```

Arguments

`model`	an model of class "kspm", usually, a result of a call to `kspm`.
`...`	furter arguments passed to or from other methods (currently unused).

Details

Cook's distance values (C_i) are computed as follows: $C_i = \frac{e_i^2 h_{ii}}{\hat{\sigma}^2 tr(H)(1-h_{ii})^2}$ where e_i is the residual of subject i, h_ii is the i th diagonal element of Hat matrix H corresponding to the leverage associated with subject i and tr(H) is the trace of the Hat matrix H.

Value

A vector containing Cook's distance values.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, residuals.kspm, rstandard.kspm, plot.kspm.

---

csm *Conventional and Social media features of 187 movies.*

Description

A dataset containing the ratings and other attributes of 187 movies.

Usage

```
csm
```

Format

A data frame with 187 rows and 13 variables:

Year year at which movies were projected on the screens
Ratings ratings
Genre genre of the movie
Gross gross income in USD
Budget budget in USD
Screens number of screens in USA
Sequel sequel
Sentiment sentiment score
Views number of views of movie trailer on Youtube
Likes number of likes of movie trailer on Youtube
Dislikes number of dislikes of movie trailer on Youtube
Comments number of comments of movie trailer on Youtube
Aggregate.Followers aggregate actor followers on Twitter

Source

https://archive.ics.uci.edu/ml/index.php

References

AHMED, Mehreen, JAHANGIR, Maham, AFZAL, Hammad, et al. Using Crowd-source based features from social media and Conventional features to predict the movies popularity. In : Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on. IEEE, 2015. p. 273-278.

derivatives — *Computing kernel function derivatives*

Description

`derivatives` is a function for "kspm" object computing pointwise partial derivatives of $h(Z)$ accroding to each Z variable.

Usage

```
derivatives(object)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.

Details

derivatives are not computed for interactions. If a variable is included in several kernels, the user may obtain the corresponding pointwise derivatives by summing the pointwise derivatives associated with each kernel.

Value

an object of class 'derivatives'

`derivmat`	a list of $n \times d$ matrix (one for each kernel) where n is the number of subjects and d the number of variables included in the kernel
`rawmat`	a $n \times q$ matrix with all variables included in the kernel part of the model q the number of variables included in the whole kernel part
`scalemat`	scaled version of rawmat
`modelmat`	matrix of correspondance between variable and kernels

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Kim, Choongrak, Byeong U. Park, and Woochul Kim. "Influence diagnostics in semiparametric regression models." Statistics and probability letters 60.1 (2002): 49:58.

See Also

plot.derivatives

deviance.kspm *Model deviance*

Description

Returns the deviance of a fitted model object of class "kspm".

Usage

```
## S3 method for class 'kspm'
deviance(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`, for which the deviance is desired.
`...`	additional optional argument (currently unused).

Details

This function extracts deviance of a model fitted using `kspm` function. The returned deviance is the residual sum of square (RSS).

Value

The value of the deviance extracted from the object `object`.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm, extractAIC.kspm

Examples

```
x <- 1:15
y <- 3*x + rnorm(15, 0, 2)
fit <- kspm(y, kernel = ~ Kernel(x, kernel.function = "linear"))
deviance(fit)
```

`energy` *Energy consumption measuring hourly during 22 days*

Description

A dataset containing the energy consumption and other attributes during 22 days.

Usage

```
energy
```

Format

A data frame with 504 rows and 7 variables:

power energy consumption

date date

Temperature temperature

P pression

HR humidity rate

hour hour (categorical)

hour.num hour (numerical)

Source

https://iles-ponant-edf-sei.opendatasoft.com, https://www.infoclimat.fr

`extractAIC.kspm` *Extract AIC from a Kernel Semi Parametric Model*

Description

Computes the Akaike Information Criterion (AIC) for a kspm fit.

Usage

```
## S3 method for class 'kspm'
extractAIC(fit, scale = NULL, k = 2,
  correction = FALSE, ...)
```

Arguments

`fit`	fitted model, usually the result of kspm.
`scale`	option not available for kspm fit.
`k`	numeric specifying the 'weight' of the effective degrees of freedom (edf) part in the AIC formula. See details.
`correction`	boolean indicating if the corrected AIC should be computed instead of standard AIC, may be `TRUE` only for `k=2`. See details.
`...`	additional optional argument (currently unused).

Details

The criterion used is $AIC = nlog(RSS) + k(n - edf)$ where RSS is the residual sum of squares and edf is the effective degree of freedom of the model. `k = 2` corresponds to the traditional AIC, using `k = log(n)` provides Bayesian Information Criterion (BIC) instead. For `k=2`, the corrected Akaike's Information Criterion (AICc) is obtained by $AICc = AIC + \frac{2(n-edf)(n-edf+1)}{(edf-1)}$.

Value

`extractAIC.kspm` returns a numeric value corresponding to AIC. Of note, the AIC obtained here differs from a constant to the AIC obtained with `extractAIC` applied to a lm object. If one wants to compare a `kspm` model with a `lm` model, it is preferrable to compute again the `lm` model using kspm function by specifying `kernel = NULL` and apply `extractAIC` method on this model.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

See Also

stepKSPM for variable selection procedure based on AIC.

Examples

```
x <- 1:15
y <- 3*x + rnorm(15, 0, 2)
fit <- kspm(y, kernel = ~ Kernel(x, kernel.function = "linear"))
extractAIC(fit)
```

`fitted.kspm` *Extract Model Fitted values*

Description

Returns fitted values for a model of class "kspm".

Usage

```
## S3 method for class 'kspm'
fitted(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

The vector of fitted values.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

See Also

kspm for fitting model, residuals.kspm, coef.kspm, nobs.kspm.

Examples

```
x <- 1:15
z <- runif(15, 1, 6)
y <- 3*x + z^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(z,
kernel.function = "polynomial", d = 2, rho = 1, gamma = 0))
fitted(fit)
```

`flexible.summary` *Summarizing Kernel Semi parametric Model Fits with flexible parameters for Davies' approximation method*

Description

for flexibility in summary method for an object of class "summary.kspm"

Usage

```
flexible.summary(object, method = "davies", acc = 1e-06, lim = 10000)
```

Arguments

`object`	an object of class "summary.kspm", usually, a result of a call to `summary.kspm`.
`method`	method to approximate the chi square distribution in p-value computation, default is 'davies', another possibility is 'imhof'.
`acc, lim`	see davies and imhof functions in CompQuadForm package.

Details

the description of the model, including coefficients for the linear part and if asked for, test(s) of variance components associated with kernel part.

Value

Computes and returns the followimg summary statistics of the fitted kernel semi parametric model given in object

`residuals`	residuals
`coefficients`	a $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t statistic and corresponding (two sided) p value for the linear part of the model.
`sigma`	the square root of the estimated variance of the random error $\sigma^2 = \frac{RSS}{edf}$ where RSS is the residual sum of squares and edf is the effective degree of freedom.
`edf`	effective degrees of freedom
`r.squared`	R^2, the fraction of variance explained by the model, $1 - \frac{\sum e_i^2}{\sum (y_i - y^*)^2}$ where y^* is the mean of y_i if there is an intercept and zero otherwise.
`adj.r.squared`	the above R^2 statistics, adjusted, penalizing for higher p.
`score.test`	a $q \times 3$ matrix with colums for the estimated lambda, tau and p value for the q kernels for which a test should be performed.
`global.p.value`	p value from the score test for the global model.
`sample.size`	sample size (all: global sample size, inc: complete data sample size).

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

Schweiger, Regev, et al. "RL SKAT: an exact and efficient score test for heritability and set tests." Genetics (2017): genetics 300395.

Li, Shaoyu, and Yuehua Cui. "Gene centric gene gene interaction: A model based kernel machine method." The Annals of Applied Statistics 6.3 (2012): 1134:1161.

See Also

kspm for fitting model, predict.kspm for predictions, plot.kspm for diagnostics

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
summary.fit <- summary(fit)
flexible.summary(summary.fit, acc = 0.000001, lim = 1000)
```

---

get.parameters — *compute Kernel Semi Parametric model parameters*

Description

internal function to compute model parameters

Usage

```
get.parameters(X = NULL, Y = NULL, kernelList = NULL,
  free.parameters = NULL, n = NULL, not.missing = NULL,
  compute.kernel = NULL)
```

Arguments

X	X matrix
Y	response matrix
kernelList	list of kernels

`free.parameters`	free parameters
`n`	number of samples
`not.missing`	number of non missing samples
`compute.kernel`	boolean indicating if kernel should be computed

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

`hypercoef` *Extract Model Hyper-parameter*

Description

Returns hyper-parameters for a model of class "kspm".

Usage

```
hypercoef(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

A list of parameter.

`lambda`	A vector of penalisation arameters.
`kernel`	A vector of tunning parameters.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

See Also

kspm for fitting model.

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
hypercoef(fit)
```

---

info.kspm *Giving information about Kernel Semi parametric Model Fits*

Description

gives information about Kernel Semi parametric Model Fits

Usage

```
info.kspm(object, print = TRUE)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`print`	logical, if `TRUE`, table of information are printed.

Value

`info.kspm` returns a table of information whose each row corresponds to a kernel included in the model and columns are:

`type`	type of object used to define the kernel
`dim`	dimension of data used in the model
`type.predict`	type of object the user should provide in predict.kspm function
`dim.predict`	dimension of object the user should provide in predict.kspm function

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm, predict.kspm

`Kernel` *Create a Kernel Object*

Description

Create a kernel object, to use as variable in a model formula.

Usage

```
Kernel(x, kernel.function, scale = TRUE, rho = NULL, gamma = NULL,
  d = NULL)
```

Arguments

`x`	a formula, a vector or a matrix of variables grouped in the same kernel. It could also be a symetric matrix representing the Gram matrix, associated to a kernel function, already computed by the user.
`kernel.function`	type of kernel. Possible values are `"gaussian"`, `"linear"`, `"polynomial"`, `"sigmoid"`, `"inverse.quadratic"` or `"equality"`. See details below. If `x` is a Gram matrix, associated to a kernel function, already computed by the user, `kernel.function` should be equal to `"gram.matrix"`.
`scale`	boolean indicating if variables should be scaled before computing the kernel.
`rho, gamma, d`	kernel function hyperparameters. See details below.

Details

To use inside kspm() function. Given two $p-$dimensional vectors x and y,

- the Gaussian kernel is defined as $k(x,y) = exp\left(-\frac{\|x-y\|^2}{\rho}\right)$ where $\| x - y \|$ is the Euclidean distance between x and y and $\rho > 0$ is the bandwidth of the kernel,
- the linear kernel is defined as $k(x,y) = x^T y$,
- the polynomial kernel is defined as $k(x,y) = (\rho x^T y + \gamma)^d$ with $\rho > 0$, d is the polynomial order. Of note, a linear kernel is a polynomial kernel with $\rho = d = 1$ and $\gamma = 0$,
- the sigmoid kernel is defined as $k(x,y) = tanh(\rho x^T y + \gamma)$ which is similar to the sigmoid function in logistic regression,
- the inverse quadratic function defined as $k(x,y) = \frac{1}{\sqrt{\|x-y\|^2+\gamma}}$ with $\gamma > 0$,
- the equality kernel defined as $k(x,y) = \left\{ \begin{array}{ll} 1 & if x = y \\ 0 & otherwise \end{array} \right.$.

Of note, Gaussian, inverse quadratic and equality kernels are measures of similarity resulting to a matrix containing 1 along the diagonal.

Value

A Kernel object including all parameters needed in computation of the model

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

`kernel.function` *Kernel Functions*

Description

These functions transform a $n \times p$ matrix into a $n \times n$ kernel matrix.

Usage

```
kernel.gaussian(x, rho = ncol(x))

kernel.linear(x)

kernel.polynomial(x, rho = 1, gamma = 0, d = 1)

kernel.sigmoid(x, rho = 1, gamma = 1)

kernel.inverse.quadratic(x, gamma = 1)

kernel.equality(x)
```

Arguments

`x`	a $n \times p$ matrix
`gamma, rho, d`	kernel hyperparameters (see details)

Details

Given two $p-$dimensional vectors x and y,

- the Gaussian kernel is defined as $k(x,y) = exp\left(-\frac{\|x-y\|^2}{\rho}\right)$ where $\| x - y \|$ is the Euclidean distance between x and y and $\rho > 0$ is the bandwidth of the kernel,
- the linear kernel is defined as $k(x,y) = x^T y$,
- the polynomial kernel is defined as $k(x,y) = (\rho x^T y + \gamma)^d$ with $\rho > 0$, d is the polynomial order. Of note, a linear kernel is a polynomial kernel with $\rho = d = 1$ and $\gamma = 0$,
- the sigmoid kernel is defined as $k(x,y) = tanh(\rho x^T y + \gamma)$ which is similar to the sigmoid function in logistic regression,

- the inverse quadratic function defined as $k(x,y) = \frac{1}{\sqrt{\|x-y\|^2+\gamma}}$ with $\gamma > 0$,
- the equality kernel defined as $k(x,y) = \begin{cases} 1 & if x = y \\ 0 & otherwise \end{cases}$.

Of note, Gaussian, inverse quadratic and equality kernels are measures of similarity resulting to a matrix containing 1 along the diagonal.

Value

A $n \times n$ matrix.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

`kernel.list` *List of kernel parts included in the kernel semi parametric model*

Description

internal method for listing all kernel parts included in the model

Usage

```
kernel.list(formula, data, names)
```

Arguments

`formula`	kernel part formula provided in the `kspm` function.
`data`	data provided in the `kspm` function.
`names`	row names of samples as they are evaluated in `kspm` function.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

`kernel.matrix` *Kernel matrix*

Description

These functions transform a $n \times p$ matrix into a $n \times n$ kernel matrix.

Usage

```
kernel.matrix(Z, whichkernel, rho = NULL, gamma = NULL, d = NULL)
```

Arguments

`Z`	a $n \times p$ matrix
`whichkernel`	kernel function
`gamma, rho, d`	kernel hyperparameters (see details)

Details

Given a $n \times p$ matrix, this function returns a $n \times n$ matrix where each cell represents the similarity between two samples defined by two $p-$dimensional vectors x and y,

- the Gaussian kernel is defined as $k(x,y) = exp\left(-\frac{\|x-y\|^2}{\rho}\right)$ where $\| x - y \|$ is the Euclidean distance between x and y and $\rho > 0$ is the bandwidth of the kernel,
- the linear kernel is defined as $k(x,y) = x^T y$,
- the polynomial kernel is defined as $k(x,y) = (\rho x^T y + \gamma)^d$ with $\rho > 0$, d is the polynomial order. Of note, a linear kernel is a polynomial kernel with $\rho = d = 1$ and $\gamma = 0$,
- the sigmoid kernel is defined as $k(x,y) = tanh(\rho x^T y + \gamma)$ which is similar to the sigmoid function in logistic regression,
- the inverse quadratic function defined as $k(x,y) = \frac{1}{\sqrt{\|x-y\|^2 + \gamma}}$ with $\gamma > 0$,
- the equality kernel defined as $k(x,y) = \left\{ \begin{array}{ll} 1 & if x = y \\ 0 & otherwise \end{array} \right.$.

Value

A $n \times n$ matrix.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kernel.gaussian, kernel.linear, kernel.polynomial, kernel.equality, kernel.sigmoid, kernel.inverse.quadratic.

`kernel.method` *some internal methods in computation of kernel semi parametric model*

Description

internal methods

Usage

```
comb(x, ...)

check.integer(N)

asOneSidedFormula(object)

splitFormula(form, sep = "/")

computes.Kernel(x, ind, nameKernel, not.missing = NULL)

computes.Kernel.interaction(x, ind, nameKernel, not.missing = NULL)

computes.KernelALL(kernelList, not.missing = NULL)

renames.Kernel(object, names)

objects.Kernel(formula)
```

Arguments

`x`	list of objects
`...`	other arguments
`N`	numeric value
`object`	formula provided in the kernel part of `kspm` function
`form`	formula
`sep`	separator
`ind`	index value
`nameKernel`	name of kernel
`not.missing`	non missing values
`kernelList`	list of kernels
`names`	name of kernel
`formula`	formula

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

`kspm` *Fitting Kernel Semi Parametric model*

Description

kspm is used to fit kernel semi parametric models.

Usage

```
kspm(response, linear = NULL, kernel = NULL, data = NULL,
  level = 1, control = kspmControl())
```

Arguments

`response`	a character with the name of the response variable or a vector containing the outcome or a matrix with outcome in the first column.
`linear`	an optional object of class "formula": a symbolic description of the linear part of the model to be fitted or a vector or a matrix containing covariates included in the linear part of the model. Default is intercept only. The details of model specification are given under 'Details'.
`kernel`	an object of class "formula": a symbolic description of the kernel part of the model to be fitted. If missing a linear model is fitted using lm function. The details of model specification are given under 'Details'.
`data`	an optional data frame containing the variables in the model. If NULL (default), data are taken from the workspace.
`level`	printed information about the model (0: no information, 1: information about kernels included in the model (default))
`control`	see kspmControl.

Details

The kernel semi parametric model refers to the following equation $Y_i = X_i\beta + h(Z_i) + e_i$ with $i = 1..n$ where n is the sample size, Y is the univariate response, $X\beta$ is the linear part, $h(Z)$ is the kernel part and e are the residuals. The linear part is defined using the `linear` argument by specifying the covariates X. It could be either a formula, a vector of length n if only one variable is included in the linear part or a $n \times p$ design matrix containing the values of the p covariates included in the linear part (columns), for each individuals (rows). By default, an intercept is included. To remove the intercept term, use formula specification and add the term `-1`, as usual. Kernel part is defined using the `kernel` argument. It should be a formula of `Kernel` object(s). For a multiple kernel semi parametric model, `Kernel` objects are separated by the usual signs `"+"`, `"*"` and `":"` to specify addition and interaction between kernels. Specification formats of each `Kernel` object may be different. See Kernel for more information about their specification.

Value

kspm returns an object of class kspm.

An object of class kspm is a list containing the following components:

linear.coefficients
: matrix of coefficients associated with linear part, the number of coefficients is the number of terms included in linear part

kernel.coefficients
: matrix of coefficients associated with kernel part, the number of rows is the sample size included in the analysis and the number of columns is the number of kernels included in the model

lambda
: penalization parameter(s)

fitted.values
: the fitted mean values

residuals
: the residuals, that is response minus the fitted values

sigma
: standard deviation of residuals

Y
: vector of responses

X
: design matrix for linear part

K
: kernel matrices computed by the model

n.total
: total sample size

n
: sample size of the model (model is performed on complete data only)

edf
: effective degree of freedom

linear.formula
: formula corresponding to the linear part of the model

kernel.info
: information about kernels included in the model such as matrices of covariates (Z), kernel function (type), values of hyperparameters (rho, gamma, d). A boolean indicates if covariates were scaled (kernel.scale) and if TRUE, kernel.mean, kernel.sd and Z.scale give information about scaling. kernel.formula indicates the formula of the kernel and free.parameters indicates the hyperparameters that were estimated by the model.

Hat
: The hat matrix H such that $\hat{Y} = HY$

L
: A matrix corresponding to $I - \sum_{\ell=1}^{L} K_\ell G_\ell^{-1} M_\ell$ according to our notations

XLX_inv
: A matrix corresponding to $(XLX)^{-1}$

GinvM
: A list of matrix, each corresponding to a kernel and equaling $G_\ell^{-1} M_\ell$ according to our notations

control
: List of control parameters

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

Kim, Choongrak, Byeong U. Park, and Woochul Kim. "Influence diagnostics in semiparametric regression models." Statistics and probability letters 60.1 (2002): 49:58.

Oualkacha, Karim, et al. "Adjusted sequence kernel association test for rare variants controlling for cryptic and family relatedness." Genetic epidemiology 37.4 (2013): 366:376.

See Also

summary.kspm for summary, predict.kspm for predictions, plot.kspm for diagnostics

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
summary(fit)
```

---

kspmControl — *Control various aspects of the optimisation problem*

Description

Allow the user to set some characteristics of the optimisation algorithm

Usage

```
kspmControl(interval.upper = NA, interval.lower = NA, trace = FALSE,
  optimize.tol = .Machine$double.eps^0.25, NP = NA, itermax = 500,
  CR = 0.5, F = 0.8, initialpop = NULL, storepopfrom = itermax + 1,
  storepopfreq = 1, p = 0.2, c = 0,
  reltol = sqrt(.Machine$double.eps), steptol = itermax,
  parallel = FALSE)
```

Arguments

`interval.upper`	integer or vetor of initial maximum value(s) allowed for parameter(s)
`interval.lower`	integer or vetor of initial maximum value(s) allowed for parameter(s)
`trace`	boolean. If TRUE parameters value at each iteration are displayed.
`optimize.tol`	if optimize function is used. See optimize
`NP`	if DEoptim function is used. See DEoptim.control

`itermax`	if DEoptim function is used. See DEoptim.control
`CR`	if DEoptim function is used. See DEoptim.control
`F`	if DEoptim function is used. See DEoptim.control
`initialpop`	if DEoptim function is used. See DEoptim.control
`storepopfrom`	if DEoptim function is used. See DEoptim.control
`storepopfreq`	if DEoptim function is used. See DEoptim.control
`p`	if DEoptim function is used. See DEoptim.control
`c`	if DEoptim function is used. See DEoptim.control
`reltol`	if DEoptim function is used. See DEoptim.control
`steptol`	if DEoptim function is used. See DEoptim.control
`parallel`	if DEoptim function is used. See DEoptim.control

Details

When only one hyperparameter should be estimated, the optimisation problem calls the optimize function from `stats` basic package. Otherwise, it calls the DEoptim function from the package `DEoptim`. In both case, the parameters are choosen among the initial interval defined by `interval.lower` and `interval.upper`.

Value

`search.parameters` is an iterative algorithm estimating model parameters and returns the following components:

`lambda`	tuning parameters for penalization.
`beta`	vector of coefficients associated with linear part of the model, the size being the number of variable in linear part (including an intercept term).
`alpha`	vector of coefficients associated with kernel part of the model, the size being the sample size.
`Ginv`	a matrix used in several calculations. $Ginv = (\lambda I + K)^{-1}$.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

link get.parameters for computation of parameters at each iteration logLik.kspm *Log Likelihood of a kspm Object*

Description

Returns the Log Likelihood value of the kernel semi parametric model represented by `obect` evaluated at the estimated coefficients.

Usage

```
## S3 method for class 'kspm'
logLik(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to kspm.
`...`	additional optional argument (currently unused).

Details

The function returns the Log Likelihood computed as follow: $logLik = -\frac{1}{2}RSS$ where RSS is the residual sum of squares.

Value

logLik of kspm fit

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

See Also

kspm, extractAIC.kspm, deviance.kspm

Examples

```
x <- 1:15
y <- 3*x + rnorm(15, 0, 2)
fit <- kspm(y, kernel = ~ Kernel(x, kernel.function = "linear"))
logLik(fit)
```

`lossFunction.looe` *Computation of the leave one out error (LOOE) in kernel semi parametric model*

Description

internal function to optimize model for estimating hyperparameters based on LOOE

Usage

```
lossFunction.looe(param. = NULL, Y. = NULL, X. = NULL,
  kernelList. = NULL, n. = NULL, not.missing. = NULL,
  compute.kernel. = NULL, print.lambda. = FALSE)
```

Arguments

`param.`	initial parameter values.
`Y.`	response matrix.
`X.`	X matrix (linear part).
`kernelList.`	list of kernels (kernel part).
`n.`	nb of samples.
`not.missing.`	nb of non missing samples.
`compute.kernel.`	boolean. If TRUE, the kernel matrix is computed at each iteration. Should be TRUE when hyperparameters of kernel functions should be estimated by the model.
`print.lambda.`	boolean. If TRUE, values of tunning parameters (lambda) are printed at each iteration.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

`nobs.kspm` *Extract the number of observations from a Kernel Semi parametric Model Fit*

Description

Extract the number of observations use to estimate the model coefficients. This is principally intented to be used in computing BIC (see extractAIC.kspm).

Usage

```
## S3 method for class 'kspm'
nobs(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

A single number (integer).

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, extractAIC.kspm.

Examples

```
x <- 1:15
y <- 3*x + rnorm(15, 0, 2)
fit <- kspm(y, kernel = ~ Kernel(x, kernel.function = "linear"))
nobs(fit)
```

`plot.derivatives` *Plot derivatives of a kspm object*

Description

Plot of derivatives for kernel part of a kspm model.

Usage

```
## S3 method for class 'derivatives'
plot(x, subset = NULL, xlab = NULL,
  ylab = NULL, ...)
```

Arguments

`x`	an object of class "derivatives", usually, a result of a call to `derivatives`.
`subset`	if a subset of the plots is required, specify the names of the variable for which plot of derivatives is required.
`xlab`	x label
`ylab`	y label
`...`	further arguments passed to or from other methods.

Details

X axis represents the raw data used as input in kernel part of the model. Y axis represents the pointwise derivative values i.e. the derivatives of fitted value according to the variable of interest.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Kim, Choongrak, Byeong U. Park, and Woochul Kim. "Influence diagnostics in semiparametric regression models." Statistics and probability letters 60.1 (2002): 49:58.

See Also

derivatives

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
plot(derivatives(fit))
```

`plot.kspm` *Plot Diagnostics for a kspm Object*

Description

Five plots (selectable by `which`) are currently available: a plot of residuals against fitted values, a scale Location plot of $\sqrt{|\ residuals\ |}$ against fitted values, a Normal Q Q plot for residuals, a plot of Cook's distances versus row labels and a plot of residuals against leverages. By default, the first three and 5 are provided.

Usage

```
## S3 method for class 'kspm'
plot(x, which = c(1:3, 5), cook.levels = c(0.5, 1),
  id.n = 3, labels.id = names(x$residuals), cex.id = 0.75,
  col.id = "blue", ...)
```

Arguments

`x`	an object of class "kspm", usually, a result of a call to `kspm`.
`which`	if a subset of the plots is required, specify a subset of the numbers 1:5.
`cook.levels`	levels of Cook's distance at which to draw contours.
`id.n`	number of points to be labelled in each plot, starting with the most extreme.
`labels.id`	vector of labels, from which the labels for extreme points will be chosen. NULL uses names associated to response specified in `kspm`.
`cex.id`	size of point labels.
`col.id`	color of point labels.
`...`	further arguments passed to or from other methods.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Kim, Choongrak, Byeong U. Park, and Woochul Kim. "Influence diagnostics in semiparametric regression models." Statistics and probability letters 60.1 (2002): 49:58.

See Also

kspm for fitting the model, summary.kspm for summary

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
plot(fit)
```

predict.kspm — *Predicting Kernel Semi parametric Model Fits*

Description

predict method for class "kspm".

Usage

```
## S3 method for class 'kspm'
predict(object, newdata.linear = NULL,
  newdata.kernel = NULL, interval = "none", level = 0.95, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`newdata.linear`	should be a data frame or design matrix of variables used in the linear part
`newdata.kernel`	a list containing data frame or design matrix of variables used in each kernel part depending on the specification format of each kernel. When a kernel has been specified using `kernel.function = "gram.matrix"` in `Kernel` function, the user should also provide the Gram matrix associated to the new data points in `newdata.kernel`. The function info.kspm may help to correctly specify it.
`interval`	type of interval calculation. If `"none"` (default), no interval is computed, if `"confidence"`, the confidence interval is computed, if `"prediction"`, the prediction interval is computed.
`level`	confidence level. Default is `level = 0.95` meaning 95% confidence/prediction interval.
`...`	further arguments passed to or from other methods.

Details

`predict.kspm` produces predicted values. If a new dataset is not specified, it will return the fitted values from the original data (complete data used in the model specification). If `predict.kspm` is applied to a new dataset, all variables used in the original model should be provided in `newdata.linear` and `newdata.kernel` arguments but only complete data may be provided. Setting `interval` specifies computation of confidence or prediction intervals at the specified `level`.

Value

`predict.kspm` returns a vector of predictions or a matrix containing the following components if `interval` is set:

`fit`	predictions.
`lwr`	lower bound of confidence/prediction intervals.
`upr`	upper bound of confidence/prediction intervals.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm, summary.kspm.

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
predict(fit, interval = "confidence")
```

---

print.kspm — *Print results from a Kernel Semi parametric Model Fit*

Description

print method for class "kspm".

Usage

```
## S3 method for class 'kspm'
print(x, ...)

## S3 method for class 'summary.kspm'
print(x, ...)
```

Arguments

`x`	an object used to select a method. Usually, a result of a call to `kspm` or a result from `summary.kspm`.
`...`	additional optional argument (currently unused).

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, summary.kspm

residuals.kspm *Extract residuals from a Kernel Semi Parametric Model*

Description

Returns the vector of residuals for a model fit of class "kspm".

Usage

```
## S3 method for class 'kspm'
residuals(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

A vector of residuals. The vector length is the number of observations used in model coefficients estimation (see nobs.kspm).

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, nobs.kspm, rstandard.kspm.

Examples

```
x <- 1:15
y <- 3*x + rnorm(15, 0, 2)
fit <- kspm(y, kernel = ~ Kernel(x, kernel.function = "linear"))
residuals(fit)
```

rstandard.kspm	*Standardized residuals for Kernel Semi parametric Model Fits*

Description

computes standardized residuals for an object of class "kspm".

Usage

```
## S3 method for class 'kspm'
rstandard(model, ...)
```

Arguments

`model`	an model of class "kspm", usually, a result of a call to `kspm`.
`...`	furter arguments passed to or from other methods (currently unused).

Details

Standardized residuals t_i are obtained by $t_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$ where e_i is the residual, $\hat{\sigma}$ is the estimated standard deviation of the errors and h_{ii} is the leverage of subject i, i.e. the i th diagonal element of the Hat matrix.

Value

a vector containing the standardized residuals.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, residuals.kspm, cooks.distance.kspm, plot.kspm.

search.parameters	*Optimisation to cumpute hyperparameter in Kernel Semi Parametric model*

Description

internal function to optimize model for estimating hyperparameters

Usage

```
search.parameters(Y = NULL, X = NULL, kernelList = NULL, n = NULL,
  not.missing = NULL, compute.kernel = NULL, controlKspm = NULL)
```

Arguments

`Y`	response matrix
`X`	X matrix
`kernelList`	of kernels
`n`	nb of samples
`not.missing`	nb of non missing samples
`compute.kernel`	boolean kernel computation
`controlKspm`	control parameters

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

`sigma.kspm` *Extract residuals standard deviation*

Description

Returns the residuals standard deviation (sigma) for object of class "kspm".

Usage

```
## S3 method for class 'kspm'
sigma(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Details

The value returned by the method is $\sqrt{\frac{RSS}{edf}}$ where RSS is the residual sum of squares and edf is the effective degree of freedom.

Value

typically a number, the estimated standard deviation of the errors ("residual standard deviation")

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm for fitting model, summary.kspm, residuals.kspm, nobs.kspm, deviance.kspm.

stepKSPM *Choose a model by AIC or BIC in a Stepwise Algorithm*

Description

Performs stepwise model selection for Kernel Semi Parametric Model by AIC or BIC.

Usage

```
stepKSPM(object, data = NULL, linear.lower = NULL,
  linear.upper = NULL, kernel.lower = NULL, kernel.upper = NULL,
  direction = "both", k = 2, kernel.param = "fixed", trace = TRUE)
```

Arguments

`object`	an object of class "kspm" with only one kernel.
`data`	data.
`linear.lower`	one side formula corresponding to the smallest set of variables that should be included in the linear part of the model.
`linear.upper`	one side formula corresponding to the largest set of variables that may be included in the linear part of the model.
`kernel.lower`	one side formula corresponding to the smallest set of variables that should be included in the kernel part of the model.
`kernel.upper`	one side formula corresponding to the largest set of variables that may be included in the kernel part of the model.
`direction`	the mode of stepwise search, can be one of "both" (default), "backward", or "forward".
`k`	type of information criteria used for the variable selection. If `k=2` AIC is used (default), if `k=log(n)`, BIC is used instead.
`kernel.param`	define if hyperparameters should be fixed (`"fixed"`) or reestimated at each iteration (`"change"`). Tu use the last option, hyperparameter of model provided in `object` should have been estimated by the model.
`trace`	integer. If positive, information is printed during the running of step.kspm. Larger values may give more information on the fitting process.

Details

This procedure may be done on `kspm` object defined with only one kernel part and for which a data frame including all variables was provided. Selection may be done on linear part only, on kernel part only or on both at the same time. To perform selection on linear (resp. kernel) part only, `kernel.lower` and `kernel.upper` (resp. `linear.lower` and `linear.upper`) should contain all the variables that should stay in the model for kernel (resp. linear) part.

Value

stepKSPM returns the selected model.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

extractAIC.kspm

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 4)
z3 <- rnorm(15, 6, 2)
z4 <- runif(15, -10, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
dfrm <- data.frame(x = x, z1 = z1, z2 = z2, z3 = z3, z4 = z4, y = y)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2 + z3 + z4,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0), data = dfrm)
stepKSPM(fit, k = 2, data = dfrm)
```

---

summary.kspm *Summarizing Kernel Semi parametric Model Fits*

Description

summary method for an object of class "kspm"

Usage

```
## S3 method for class 'kspm'
summary(object, kernel.test = "all",
  global.test = FALSE, ...)
```

Arguments

object	an object of class "kspm", usually, a result of a call to kspm.
kernel.test	vector of characters indicating for which kernel a test should be performed. Default is "all". If "none", no test will be performed.
global.test	logical, if TRUE, a global test for kernel part is computed.
...	further arguments passed to or from other methods.

Details

the description of the model, including coefficients for the linear part and if asked for, test(s) of variance components associated with kernel part.

Value

Computes and returns the followimg summary statistics of the fitted kernel semi parametric model given in object

`residuals`	residuals
`coefficients`	a $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t statistic and corresponding (two sided) p value for the linear part of the model.
`sigma`	the square root of the estimated variance of the random error $\sigma^2 = \frac{RSS}{edf}$ where RSS is the residual sum of squares and edf is the effective degree of freedom.
`edf`	effective degrees of freedom
`r.squared`	R^2, the fraction of variance explained by the model, $1 - \frac{\sum e_i^2}{\sum (y_i - y^*)^2}$ where y^* is the mean of y_i if there is an intercept and zero otherwise.
`adj.r.squared`	the above R^2 statistics, adjusted, penalizing for higher p.
`score.test`	a $q \times 3$ matrix with colums for the estimated lambda, tau and p value for the q kernels for which a test should be performed.
`global.p.value`	p value from the score test for the global model.
`sample.size`	sample size (all: global sample size, inc: complete data sample size).

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Liu, D., Lin, X., and Ghosh, D. (2007). Semiparametric regression of multidimensional genetic pathway data: least squares kernel machines and linear mixed models. Biometrics, 63(4), 1079:1088.

Schweiger, Regev, et al. "RL SKAT: an exact and efficient score test for heritability and set tests." Genetics (2017): genetics 300395.

Li, Shaoyu, and Yuehua Cui. "Gene centric gene gene interaction: A model based kernel machine method." The Annals of Applied Statistics 6.3 (2012): 1134:1161.

See Also

kspm for fitting model, predict.kspm for predictions, plot.kspm for diagnostics

Examples

```
x <- 1:15
z1 <- runif(15, 1, 6)
z2 <- rnorm(15, 1, 2)
y <- 3*x + (z1 + z2)^2 + rnorm(15, 0, 2)
fit <- kspm(y, linear = ~ x, kernel = ~ Kernel(~ z1 + z2,
kernel.function = "polynomial", d= 2, rho = 1, gamma = 0))
summary(fit)
```

`test.function` *Score Tests for kernel part in kernel semi parametric model*

Description

Perform score tests for kernel part in kernel semi parametric model

Usage

```
test.1.kernel(object)

test.global.kernel(object)

test.k.kernel(object, kernel.name)
```

Arguments

`object`	an object of class "kspm"
`kernel.name`	vector of character listing names of kernels for which test should be performed

Value

p values

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

References

Schweiger, Regev, et al. "RL SKAT: an exact and efficient score test for heritability and set tests." Genetics (2017): genetics 300395.

Li, Shaoyu, and Yuehua Cui. "Gene centric gene gene interaction: A model based kernel machine method." The Annals of Applied Statistics 6.3 (2012): 1134:1161.

Oualkacha, Karim, et al. "Adjusted sequence kernel association test for rare variants controlling for cryptic and family relatedness." Genetic epidemiology 37.4 (2013): 366:376.

Ge, Tian, et al. "A kernel machine method for detecting effects of interaction between multidimensional variable sets: An imaging genetics application." Neuroimage 109 (2015): 505:514.

`variable.names.kspm` *Variable names of fitted models*

Description

Simple utility returning names of variables involved in a kernel semi parametric model.

Usage

```
## S3 method for class 'kspm'
variable.names(object, ...)
```

Arguments

`object`	an object of class "kspm", usually, a result of a call to `kspm`.
`...`	additional optional argument (currently unused).

Value

a list of character vectors. The first element correspond to the names of variables included in the linear part of the model. Then, a vector containing names of variables including in kernel part is provided for each kernel.

Author(s)

Catherine Schramm, Aurelie Labbe, Celia Greenwood

See Also

kspm, summary.kspm, case.names.kspm.

Index