

Package: matern32 (via r-universe)

May 9, 2026

Type Package

Title Kernel Ridge Regression using Matern 3/2 kernels

Version 0.2.0

Date 2025-03-16

Author T. Moudiki

Maintainer T. Moudiki <thierry.moudiki@gmail.com>

Description Interpretable Probabilistic Kernel Ridge Regression using Matern 3/2 kernels.

License BSD_3_clause Clear + file LICENSE

Imports Rcpp (>= 0.12.17), tseries

Suggests MASS, mda, misc

LinkingTo Rcpp

RoxygenNote 7.3.2

Remotes thierrymoudiki/misc

Config/pak/sysreqs libssl-dev

Repository <https://techtonique.r-universe.dev>

Date/Publication 2025-07-21 15:07:44 UTC

RemoteUrl <https://github.com/thierrymoudiki/matern32>

RemoteRef HEAD

RemoteSha adf4a696a3079a07823295bfa95bba3bffffc5db

Contents

matern32-package	2
derivatives	3
direct_sampling	3
fit_matern32	4
fit_poly	5
interactions	6
plot.matern32	6

plot_coeffs	7
plot_GCV	7
plot_heterogeneity	7
plot_interactions	8
plot_loocv	8
plot_residuals	9
predict.matern32	9
predict.poly	10
rcpp_hello_world	10
rmultivariate	11
sensi1D	11
sensi2D	12
summary.matern32	12

Index	13
--------------	-----------

matern32-package	<i>A short title line describing what the package does</i>
------------------	--

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
## Not run:
## Optional simple examples of the most important functions
## These can be in \dontrun{} and \donttest{} blocks.
```

```
## End(Not run)
```

derivatives	<i>Compute the first and second order derivatives of the response function</i>
-------------	--

Description

Compute the first and second order derivatives of the response function

Usage

```
derivatives(fit_obj, obs = NULL)
```

Arguments

fit_obj	Fitted object
---------	---------------

direct_sampling	<i>Direct sampling</i>
-----------------	------------------------

Description

Direct sampling

Usage

```
direct_sampling(  
  data = NULL,  
  n = 100L,  
  method = c("kde", "surrogate", "bootstrap"),  
  kde = NULL,  
  seed = NULL,  
  ...  
)
```

Arguments

data	A numeric vector or matrix.
n	The number of samples to draw.
method	The method to use for sampling.
kde	The kernel density estimate to use for sampling.
seed	The seed to use for sampling.
...	Additional arguments to pass to the density function.

fit_matern32

*Fit Matern 3/2 model***Description**

Fit Matern 3/2 model

Usage

```
fit_matern32(
  x,
  y,
  lambda = 10^seq(-10, 10, length.out = 100),
  l = NULL,
  method = c("chol", "solve", "svd", "eigen", "conformal"),
  with_kmeans = FALSE,
  centers = NULL,
  centering = FALSE,
  seed = 123,
  ...
)
```

Arguments

x	Matrix of predictors
y	Vector of response values
lambda	Regularization parameter
l	Hyperparameter for the Matern 3/2 kernel
method	Method to use for fitting the model
with_kmeans	Use k-means to reduce the number of observations
centers	Number of centers to use in k-means
centering	Center the response
seed	Seed for reproducibility
...	
sigma	Hyperparameter for the Matern 3/2 kernel

Examples

```
n <- 100; p <- 4

set.seed(456)
X <- matrix(rnorm(n * p), n, p) # no intercept!
y <- rnorm(n)

lams <- 10^seq(-5, 4, length.out = 50)
```

```

fit_obj <- matern32::fit_matern32(x = X, y = y, lambda = lams)

par(mfrow=c(1, 2))

plot(log(lams), fit_obj$GCV, type = 'l', main = "GCV",
      ylab = "GCV")

matplot(log(lams), t(fit_obj$coef), type = 'l',
        main = "coefficients = f(lambda)", xlab = "log(lambda)",
        ylab = "coefs")
abline(h = 0, lty = 2, lwd = 2, col = "red")

matern32::summary.matern32(fit_obj)

library(MASS)
X <- longley[,-7]
y <- longley[, 7]
fit_obj <- matern32::fit_matern32(x = X, y = y)

par(mfrow=c(1, 2))

plot(log(fit_obj$lambda), fit_obj$GCV, type = 'l', main = "GCV",
      ylab = "GCV")

matplot(log(fit_obj$lambda), t(fit_obj$coef), type = 'l',
        main = "coefficients = f(lambda)", xlab = "log(lambda)",
        ylab = "coefs")
abline(h = 0, lty = 2, lwd = 2, col = "red")

matern32::summary.matern32(fit_obj)

library(MASS)

X <- as.matrix(Boston[,-14])
y <- Boston[,14]

fit_obj <- matern32::fit_matern32(x = X, y = y, lambda = lams)

matern32::summary.matern32(fit_obj)

```

fit_poly

Polynomial regression

Description

Polynomial regression

Usage

```
fit_poly(x, y, degree = 1)
```

Arguments

x	Matrix of predictors
y	Vector of response values
degree	Degree of the polynomial

interactions	<i>Title</i>
--------------	--------------

Description

Title

Usage

```
interactions(fit_obj, index_col1 = 1, index_col2 = 2, obs = NULL)
```

Arguments

```
fit_obj
index_col1
index_col2
obs
```

plot.matern32	<i>Plot method for matern32 objects</i>
---------------	---

Description

Plot method for matern32 objects

Usage

```
plot.matern32(
  fit_obj,
  choice = c("coeffs", "GCV", "heterogeneity", "interactions", "residuals"),
  ...
)
```

Arguments

fit_obj	Fitted object of class 'matern32'
choice	The type of plot to produce (coeffs, GCV, heterogeneity, interactions, residuals)
...	

plot_coeffs	<i>Title</i>
-------------	--------------

Description

Title

Usage

plot_coeffs(fit_obj, ...)

Arguments

fit_obj

plot_GCV	<i>Title</i>
----------	--------------

Description

Title

Usage

plot_GCV(fit_obj, ...)

Arguments

fit_obj

plot_heterogeneity	<i>Title</i>
--------------------	--------------

Description

Title

Usage

plot_heterogeneity(fit_obj, var = 1, ...)

Arguments

fit_obj

var

...

plot_interactions *Plot interactions*

Description

Plot interactions

Usage

```
plot_interactions(
  fit_obj,
  var1 = 1,
  var2 = 2,
  method = c("poly", "mars"),
  degree = 1,
  ...
)
```

Arguments

fit_obj	Fitted object of class 'matern32'
var1	Covariate 1
var2	Covariate 2
method	The method to use for smoothing the interactions
degree	The degree of the polynomial to use for smoothing
...	

plot_loocv *Title*

Description

Title

Usage

```
plot_loocv(fit_obj, ...)
```

Arguments

fit_obj
...

plot_residuals	<i>Plot residuals</i>
----------------	-----------------------

Description

Plot residuals

Usage

```
plot_residuals(fit_obj, ...)
```

Arguments

fit_obj

predict.matern32	<i>Predict from Matern 3/2 model</i>
------------------	--------------------------------------

Description

Predict from Matern 3/2 model

Usage

```
## S3 method for class 'matern32'
predict(
  fit_obj,
  newx,
  level = NULL,
  method = c("splitconformal", "kde", "surrogate", "bootstrap"),
  nsim = 250L,
  B = nsim
)
```

Arguments

fit_obj	Fitted object from fit_matern32
newx	New data matrix
level	Confidence level for prediction intervals (between 0 and 100)
method	Method to use for prediction intervals
nsim	Number of simulations to use for prediction
B	Alias for nsim

Examples

```

n <- 100 ; p <- 4

set.seed(456)
X <- matrix(rnorm(n * p), n, p) # no intercept!
y <- rnorm(n)

lams <- 10^seq(-10, 10, length.out = 50)

fit_obj <- fit_matern32(x = X, y = y, lambda = lams)

df <- data.frame(predict(fit_obj, X) - y)
colnames(df) <- paste0(round(lams, 2))
summary(df)
boxplot(df[, c(1, 10, 25, 35, 50)],
main = "distribution of in sample bias",
xlab = "lambda", ylab = "y_hat - y")

```

predict.poly	<i>Predict method for polynomial regression</i>
--------------	---

Description

Predict method for polynomial regression

Usage

```

## S3 method for class 'poly'
predict(object, newx)

```

Arguments

object	Fitted object
newx	New data

rcpp_hello_world	<i>Simple function using Rcpp</i>
------------------	-----------------------------------

Description

Simple function using Rcpp

Usage

```
rcpp_hello_world()
```

Examples

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

rmultivariate	<i>Simulate multivariate data</i>
---------------	-----------------------------------

Description

Simulate multivariate data

Usage

```
rmultivariate(
  data,
  method = c("bootstrap", "block-bootstrap"),
  n = 100L,
  block_size = 5
)
```

Arguments

data	A numeric vector or matrix.
method	The method to use for sampling.
n	The number of samples to draw.
block_size	The size of the blocks to use for the block bootstrap.
...	Additional arguments to pass to the density function.

sensi1D	<i>Title</i>
---------	--------------

Description

Title

Usage

```
sensi1D(fit_obj, index_col, h, verbose = TRUE)
```

Arguments

fit_obj	object fitted by using fit_matern32
index_col	an integer, column index
h	(must be small)

sensi2D	<i>Title</i>
---------	--------------

Description

Title

Usage

```
sensi2D(fit_obj, index_col1, index_col2, h1, h2, verbose = TRUE)
```

Arguments

fit_obj	object fitted by using fit_matern32
index_col1	an integer, column index
index_col2	an integer, second column index
h	(must be small)

summary.matern32	<i>Summary method for matern32</i>
------------------	------------------------------------

Description

Summary method for matern32

Usage

```
summary.matern32(fit_obj, obs = NULL, verbose = TRUE)
```

Arguments

fit_obj	Fitted object of class 'matern32'
obs	observation number for which to calculate the derivatives (default is NULL)
verbose	logical indicating whether to print the summary (default is TRUE)

Index

* package

matern32-package, 2

derivatives, 3

direct_sampling, 3

fit_matern32, 4

fit_poly, 5

interactions, 6

matern32 (matern32-package), 2

matern32-package, 2

plot.matern32, 6

plot_coeffs, 7

plot_GCV, 7

plot_heterogeneity, 7

plot_interactions, 8

plot_loocv, 8

plot_residuals, 9

predict.matern32, 9

predict.poly, 10

rcpp_hello_world, 10

rmultivariate, 11

sensi1D, 11

sensi2D, 12

summary.matern32, 12