

Package: zoomerGP (via r-universe)

May 25, 2026

Title Fast, composable Gaussian Processes in R.
Version 0.0.0.9000
Description Fast, composable Gaussian Processes in R.
License GPL (>= 3)
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.2
Config/rextendr/version 0.4.0.9000
SystemRequirements Cargo (Rust's package manager), rustc
Depends R (>= 4.2)
Suggests knitr, rmarkdown, nlopt, testthat (>= 3.0.0), tibble
VignetteBuilder knitr
Config/testthat/edition 3
Imports ggplot2, memoise, rgenoud, rlang, tidysselect
URL <http://beniamino.org/zoomerGP/>
Config/pak/sysreqs libclang-dev
Repository <https://beniaminogreen.r-universe.dev>
Date/Publication 2025-08-28 15:25:53 UTC
RemoteUrl <https://github.com/beniaminogreen/zoomergp>
RemoteRef HEAD
RemoteSha dea14b365145a45a344ae8b95da0325cf0cd1ee8

Contents

gaussian_process	2
logistic_gaussian_process	3
plot.gaussian_process	4
predict.gaussian_process	4
print.gaussian_process	5
residuals.gaussian_process	5

gaussian_process	<i>Fit a Gaussian process in R</i>
------------------	------------------------------------

Description

Fit a Gaussian process in R

Usage

```
gaussian_process(
  form,
  data,
  sparse = F,
  n_points = 5,
  training_method = "bfgs",
  noise = T
)
```

Arguments

form	the regression formula for the gaussian process. The left-hand-side gives determines outcome you are fitting the GP to, and the right-hand side gives the kernel specification.
data	the dataframe from which to source regression data.
sparse	a boolean indicating whether to use the projected process approximation to speed up training. Recommended for $N > 500$.
n_points	The number of inducing points used for the projected process approximation. Defaults to 5. If set too large, may cause numerical stability issues when calculating the gradient of the marginal likelihood.
training_method	one of 'bfgs', 'cg' (conjugate gradient), 'rgenoud' (genetic optimization using derivatives), or "coin" (coin betting). The method used to optimize the marginal likelihood (default BFGS).
noise	whether to fit a GP assuming function values are observed with noise (default TRUE). Setting FALSE is only available when sparse = F.

Examples

```
library(tibble)
library(ggplot2)
n <- 400
data <- tibble(
  x = runif(n),
  y = runif(n),
  z = x^2 + rnorm(n, 0, .1)
```

```

)
gp_model <- gaussian_process(z ~ rbf(x,y), data = data)
data$predictions <- predict(gp_model)[,1]
ggplot(data, aes(x=x)) +
  geom_point(aes(y= z)) +
  geom_line(aes(y=predictions))

```

logistic_gaussian_process

Fit a logistic Gaussian process in R for a binary outcome

Description

Fit a logistic Gaussian process in R for a binary outcome

Usage

```

logistic_gaussian_process(
  formula,
  data,
  sparse = T,
  n_points = 10,
  n_particles = 10,
  n_iter = 2000
)

```

Arguments

data	the dataframe from which to source regression data.
sparse	a Boolean indicating whether to use the projected process approximation to speed up training. Recommended for N>100.
n_points	The number of inducing points used for the projected process approximation. Defaults to 5. If set too large, may cause numerical stability issues when calculating the gradient of the marginal likelihood.
n_particles	number of particles to be used in the stein variational gradient descent simulation
form	the regression formula for the Gaussian process. The left-hand-side gives determines outcome you are fitting the GP to, and the right-hand side gives the kernel specification.

plot.gaussian_process *Plot Gaussian Process Fit*

Description

Plot Gaussian Process Fit

Usage

```
## S3 method for class 'gaussian_process'
plot(x, col = 1, add_points = T, ...)
```

Arguments

x	Gaussian Process object
col	column name for the predictor you want to plot. All other predictors are set to their medians
add_points	whether to plot datapoints in the background
...	additional arguments (ignored)

predict.gaussian_process
Predict Function Values at a Test Point

Description

Predict Function Values at a Test Point

Usage

```
## S3 method for class 'gaussian_process'
predict(object, newdata = NULL, kernel_str = NA, ...)
```

Arguments

object	An object for which predictions are desired.
newdata	a new data set to generate predictions for. Must have all of the predictor columns of the original data set or an error will be returned.
kernel_str	the label of the kernel you would like to extract predictions for. This can be useful if you are (for example) attempting to isolate a specific signal from a data set such as a periodic component from a time series. Use the print method of the Gaussian process to determine the labels for the individual kernels.
...	additional arguments (ignored)

```
print.gaussian_process
```

Print information about a Gaussian Process

Description

Print information about a Gaussian Process

Usage

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

Arguments

x	Gaussian Process object
...	additional arguments (ignored)

```
residuals.gaussian_process
```

Get Residuals for Gaussian Process

Description

Get Residuals for Gaussian Process

Usage

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

Arguments

object	An object for which residuals are desired.
...	additional arguments (ignored)

Index

`gaussian_process`, 2

`logistic_gaussian_process`, 3

`plot.gaussian_process`, 4

`predict.gaussian_process`, 4

`print.gaussian_process`, 5

`residuals.gaussian_process`, 5