Package 'greta.gp'

September 6, 2022

Type Package

Title Gaussian Process Modelling in 'greta'

Version 0.2.0

Description Provides a syntax to create and combine Gaussian process kernels in 'greta'. You can then them to define either full rank or sparse Gaussian processes. This is an extension to the 'greta' software, Golding (2019) <doi:10.21105/joss.01601>.

License Apache License (>= 2)

URL https://github.com/greta-dev/greta.gp, https://greta-dev.github.io/greta.gp/

BugReports https://github.com/greta-dev/greta.gp/issues

Depends greta (>= 0.4.2), R (>= 3.1.0)

Imports cli, glue, tensorflow (>= 2.7.0)

Suggests covr, knitr, rmarkdown, spelling, testthat (>= 3.1.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.2.0

SystemRequirements Python (>= 2.7.0) with header files and shared library; TensorFlow (v1.14; https://www.tensorflow.org/); TensorFlow Probability (v0.7.0; https://www.tensorflow.org/probability/)

NeedsCompilation no

Author Nick Golding [aut, cph] (<https://orcid.org/0000-0001-8916-5570>), Jian Yen [ctb], Nicholas Tierney [aut, cre] (<https://orcid.org/0000-0003-1460-8722>)

Maintainer Nicholas Tierney <nicholas.tierney@gmail.com>

Repository CRAN

Date/Publication 2022-09-06 07:10:06 UTC

R topics documented:

---

gp *Define a Gaussian process*

Description

Define Gaussian processes, and project them to new coordinates.

Usage

```
gp(x, kernel, inducing = NULL, n = 1, tol = 1e-04)

project(f, x_new, kernel = NULL)
```

Arguments

`x, x_new`	greta array giving the coordinates at which to evaluate the Gaussian process
`kernel`	a kernel function created using one of the `kernel()` methods
`inducing`	an optional greta array giving the coordinates of inducing points in a sparse (reduced rank) Gaussian process model
`n`	the number of independent Gaussian processes to define with the same kernel
`tol`	a numerical tolerance parameter, added to the diagonal of the self-covariance matrix when computing the cholesky decomposition. If the sampler is hitting a lot of numerical errors, increasing this parameter could help
`f`	a greta array created with `gp$gp` representing the values of one or more Gaussian processes

Details

`gp()` returns a greta array representing the values of the Gaussian process(es) evaluated at `x`. This Gaussian process can be made sparse (via a reduced-rank representation of the covariance) by providing an additional set of inducing point coordinates `inducing`. `project()` evaluates the values of an existing Gaussian process (created with `gp()`) to new data.

Value

A greta array

Examples

```
## Not run:
# build a kernel function on two dimensions
k1 <- rbf(lengthscales = c(0.1, 0.2), variance = 0.6)
k2 <- bias(variance = lognormal(0, 1))
K <- k1 + k2

# use this kernel in a full-rank Gaussian process
f <- gp(1:10, K)

# or in sparse Gaussian process
f_sparse <- gp(1:10, K, inducing = c(2, 5, 8))

# project the values of the GP to new coordinates
f_new <- project(f, 11:15)

# or project with a different kernel (e.g. a sub-kernel)
f_new_bias <- project(f, 11:15, k2)

## End(Not run)
```

---

greta.gp *Gaussian process modelling in greta*

Description

A greta module to create and combine covariance functions and use them to build Gaussian process models in greta. See `kernels()` and `gp()`

---

kernels *Gaussian process kernels*

Description

Create and combine Gaussian process kernels (covariance functions) for use in Gaussian process models.

Usage

```
bias(variance)

constant(variance)

white(variance)

iid(variance, columns = 1)
```

```
rbf(lengthscales, variance, columns = seq_along(lengthscales))

rational_quadratic(
  lengthscales,
  variance,
  alpha,
  columns = seq_along(lengthscales)
)

linear(variances, columns = seq_along(variances))

polynomial(variances, offset, degree, columns = seq_along(variances))

expo(lengthscales, variance, columns = seq_along(lengthscales))

mat12(lengthscales, variance, columns = seq_along(lengthscales))

mat32(lengthscales, variance, columns = seq_along(lengthscales))

mat52(lengthscales, variance, columns = seq_along(lengthscales))

cosine(lengthscales, variance, columns = seq_along(lengthscales))

periodic(period, lengthscale, variance)
```

Arguments

`variance`, `variances`
: (scalar/vector) the variance of a Gaussian process prior in all dimensions (`variance`) or in each dimension (`variances`)

`columns`
: (scalar/vector integer, not a greta array) the columns of the data matrix on which this kernel acts. Must have the same dimensions as lengthscale parameters.

`alpha`
: (scalar) additional parameter in rational quadratic kernel

`offset`
: (scalar) offset in polynomial kernel

`degree`
: (scalar) degree of polynomial kernel

`period`
: (scalar) the period of the Gaussian process

`lengthscale`, `lengthscales`
: (scalar/vector) the correlation decay distance along all dimensions (`lengthscale`) or each dimension ((`lengthscales`)) of the Gaussian process

Details

The kernel constructor functions each return a *function* (of class `greta_kernel`) which can be executed on greta arrays to compute the covariance matrix between points in the space of the Gaussian process. The `+` and `*` operators can be used to combine kernel functions to create new kernel functions.

Note that `bias` and `constant` are identical names for the same underlying kernel.

`iid` is equivalent to `bias` where all entries in `columns` match (where the absolute euclidean distance is less than 1e-12), and `white` where they don't; i.e. an independent Gaussian random effect.

Value

greta kernel with class "greta_kernel"

Examples

```
## Not run:
# create a radial basis function kernel on two dimensions
k1 <- rbf(lengthscales = c(0.1, 0.2), variance = 0.6)

# evaluate it on a greta array to get the variance-covariance matrix
x <- greta_array(rnorm(8), dim = c(4, 2))
k1(x)

# non-symmetric covariance between two sets of points
x2 <- greta_array(rnorm(10), dim = c(5, 2))
k1(x, x2)

# create a bias kernel, with the variance as a variable
k2 <- bias(variance = lognormal(0, 1))

# combine two kernels and evaluate
K <- k1 + k2
K(x, x2)

# other kernels
constant(variance = lognormal(0, 1))
white(variance = lognormal(0, 1))
iid(variance = lognormal(0,1))
rational_quadratic(lengthscales = c(0.1, 0.2), alpha = 0.5, variance = 0.6)
linear(variances = 0.1)
polynomial(variances = 0.6, offset = 0.8, degree = 2)
expo(lengthscales = 0.6 ,variance = 0.9)
mat12(lengthscales = 0.5, variance = 0.7)
mat32(lengthscales = 0.4, variance = 0.8)
mat52(lengthscales = 0.3, variance = 0.9)
cosine(lengthscales = 0.68, variance = 0.8)
periodic(period = 0.71, lengthscale = 0.59, variance = 0.2)

## End(Not run)
```

Index