Package ‘GauPro’

September 11, 2017

Type Package
Title Gaussian Process Fitting
Version 0.2.2
Author Collin Erickson
Maintainer Collin Erickson <collinberickson@gmail.com>

Description Fits a Gaussian process model to data. Gaussian processes are commonly used in computer experiments to fit an interpolating model. The model is stored as an 'R6' object and can be easily updated with new data. There are options to run in parallel (not for Windows), and 'Rcpp' has been used to speed up calculations. Other R packages that perform similar calculations include 'laGP', 'DiceKriging', 'GPfit', and 'mlegp'.

License GPL-3
LazyData TRUE

LinkingTo Rcpp, RcppArmadillo
Imports Rcpp, R6, lbfgs

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, microbenchmark, numDeriv, MASS

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-09-11 08:22:23 UTC

R topics documented:

*.GauPro_kernel .............................................. 2
+.GauPro_kernel .............................................. 3
corr_gauss_matrix ........................................... 3
corr_gauss_matrix_symC ..................................... 4
corr_gauss_matrix_sym_armaC ............................ 5
Exponential .................................................. 5
GauPro .......................................................... 6
Description

Kernel product

Usage

```r
## S3 method for class 'GauPro_kernel'
k1 * k2
```

Arguments

- `k1`: First kernel
- `k2`: Second kernel

Value

Kernel which is product of two kernels
### Kernel sum

#### Description

Kernel sum

#### Usage

```r
# S3 method for class 'GauPro_kernel'
k1 + k2
```

#### Arguments

- `k1`: First kernel
- `k2`: Second kernel

#### Value

Kernel which is sum of two kernels

#### Examples

```r
k1 <- Exponential$new(beta=1)
k2 <- Matern$new(beta=0)
k <- k1 * k2
k$k(matrix(c(2,1), ncol=1))
```

---

### Gaussian correlation

#### Description

Gaussian correlation

#### Usage

```r
corr_gauss_matrix(x, x2 = NULL, theta)
```

#### Examples

```r
k1 <- Exponential$new(beta=1)
k2 <- Matern$new(beta=0)
k <- k1 + k2
k$k(matrix(c(2,1), ncol=1))
```
**Arguments**

- `x`  First data matrix
- `x2` Second data matrix
- `theta` Correlation parameter

**Value**

Correlation matrix

**Examples**

```r
corr_gauss_matrix(matrix(1:10, ncol=1), matrix(6:15, ncol=1), 1e-2/(1:10))
```

---

**corr_gauss_matrix_symC**

*Correlation Gaussian matrix in C (symmetric)*

---

**Description**

Correlation Gaussian matrix in C (symmetric)

**Usage**

```r
corr_gauss_matrix_symC(x, theta)
```

**Arguments**

- `x`  Matrix x
- `theta` Theta vector

**Value**

Correlation matrix

**Examples**

```r
corr_gauss_matrix_symC(matrix(c(1,0,0,1),2,2), c(1,1))
```
Correlation Gaussian matrix in C using Armadillo (symmetric)

**Description**

Correlation Gaussian matrix in C using Armadillo (symmetric)

**Usage**

```
corr_gauss_matrix_sym_armaC(x, theta)
```

**Arguments**

- `x`: Matrix x
- `theta`: Theta vector

**Value**

Correlation matrix

**Examples**

```
corr_gauss_matrix_sym_armaC(matrix(c(QLPLPLQ)LRLR)Lc(QLQ))
```

---

**Exponential Kernel R6 class**

**Description**

Exponential Kernel R6 class

**Usage**

```
Exponential
```

**Format**

```
R6Class object.
```

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```
k1 <- Exponential$new(beta=0)
```
**GauPro**  

*GauPro_selector*

**Description**

GauPro_selector

**Usage**

GauPro(..., type = "Gauss")

**Arguments**

- ... Pass on
- type Type of Gaussian process, or the kind of correlation function.

**Value**

A GauPro object

**Examples**

```r
n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
#y <- sin(2*pi*x) + rnorm(n,0,1e-1)
y <- (2*x) %% 1
gp <- GauPro(x=x, y=y, parallel=FALSE)
```

---

**GauPro_base**  

*Class providing object with methods for fitting a GP model*

**Description**

Class providing object with methods for fitting a GP model

**Usage**

GauPro_base

**Format**

*R6Class* object.

**Value**

Object of *R6Class* with methods for fitting GP model.
**Fields**

- **X** Design matrix
- **Z** Responses
- **N** Number of data points
- **D** Dimension of data
- **corr** Type of correlation function
- **nug.min** Minimum value of nugget
- **nug** Value of the nugget, is estimated unless told otherwise
- **separable** Are the dimensions separable?
- **verbose** 0 means nothing printed, 1 prints some, 2 prints most.
- **useGrad** Should grad be used?
- **useC** Should C code be used?
- **parallel** Should the code be run in parallel?
- **parallel_cores** How many cores are there? It will self detect, do not set yourself.

**Methods**

**Documentation** For full documentation of each method go to https://github.com/lightning-viz/lightining-r/

```r
gaupro_gauss
```

This method is used to create object of this class with `X` and `Z` as the data.

```r
update(Xnew=NULL, Znew=NULL, Xall=NULL, Zall=NULL, restarts = 5, param_update = T, nug.update = sel)
```

This method updates the model, adding new data if given, then running optimization again.

**Examples**

```r
n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
y <- sin(2*pi*x) + rnorm(n,0,1e-1)
geps <- GauPro(x=x, y=y, parallel=FALSE)
```

---

**GauPro_Gauss**

_Corr Gauss GP using inherited optim_

---

**Description**

Corr Gauss GP using inherited optim

**Usage**

GauPro_Gauss
Format

    *R6Class* object.

Value

Object of *R6Class* with methods for fitting GP model.

Examples

```r
n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
y <- sin(2*pi*x) + rnorm(n,0,1e-1)
gp <- GauPro(x=x, y=y, parallel=FALSE)
```

---

### GauPro_kernel

**Kernel R6 class**

Description

Kernel R6 class

Usage

GauPro_kernel

Format

*R6Class* object.

Value

Object of *R6Class* with methods for fitting GP model.

Examples

```r
#k <- GauPro_kernel$new()
```
**GauPro_kernel_beta**  
*Beta Kernel R6 class*

**Description**

This is the base structure for a kernel that uses $\beta = \log_{10}(\theta)$ for the lengthscale parameter. It standardizes the params because they all use the same underlying structure. Kernels that inherit this only need to implement $k$ and $dC_dparams$.

**Usage**

GauPro_kernel_beta

**Format**

*R6Class* object.

**Value**

Object of *R6Class* with methods for fitting GP model.

**Examples**

```r
# k1 <- Matern52$new(beta=0)
```

---

**GauPro_kernel_model**  
*GauPro model that uses kernels*

**Description**

Class providing object with methods for fitting a GP model. Allows for different kernel and trend functions to be used.

**Usage**

GauPro_kernel_model

**Format**

*R6Class* object.

**Value**

Object of *R6Class* with methods for fitting GP model.
Fields

- **X** Design matrix
- **Z** Responses
- **N** Number of data points
- **D** Dimension of data
- **corr** Type of correlation function
- **nug.min** Minimum value of nugget
- **nug** Value of the nugget, is estimated unless told otherwise
- **separable** Are the dimensions separable?
- **verbose** 0 means nothing printed, 1 prints some, 2 prints most.
- **useGrad** Should grad be used?
- **useC** Should C code be used?
- **parallel** Should the code be run in parallel?
- **parallel_cores** How many cores are there? It will self detect, do not set yourself.

Methods

**Documentation** For full documentation of each method go to https://github.com/lightning-viz/lightining-r/

- `new(xL ZL corr\"gauss\"L verbose\]PL separable\]tL usec\]fLusegrad\]tL parallel\]tL nug\]nest\]tL NNN)`
  This method is used to create object of this class with `x` and `Z` as the data.

- `update(xnew=NULLL Znew=NULLL Xall=NULLL Zall=NULLL, restarts = 5, param_update = T, nug.update = sel)`
  This method updates the model, adding new data if given, then running optimization again.

Examples

```r
n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
y <- sin(2*pi*x) + rnorm(n,0,1e-1)
gp <- GauPro_kernel_model$new(X=x, Z=y, kernel=Gaussian$new(1), parallel=FALSE)
gp$predict(.454)
```

---

**GauPro_trend**

**Trend R6 class**

**Description**

Trend R6 class

**Usage**

GauPro_trend
**Gaussian**

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
# k <- GauPro_trend$new()
```

---

### Gaussian

#### Gaussian Kernel R6 class

**Description**

Gaussian Kernel R6 class

**Usage**

Gaussian

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
k1 <- Gaussian$new(beta=0)
```
Gaussian_devianceC Calculate the Gaussian deviance in C

Description
Calculate the Gaussian deviance in C

Usage
Gaussian_devianceC(theta, nug, X, Z)

Arguments
theta Theta vector
nug Nugget
X Matrix X
Z Matrix Z

Value
Correlation matrix

Examples
Gaussian_devianceC(c(1,1), 1e-8, matrix(c(1,0,0,1),2,2), matrix(c(1,0,2,1)))

Gaussian_hessianC Calculate Hessian for a GP with Gaussian correlation

Description
Calculate Hessian for a GP with Gaussian correlation

Usage
Gaussian_hessianC(XX, X, Z, Kinv, mu_hat, theta)

Arguments
XX The vector at which to calculate the Hessian
X The input points
Z The output values
Kinv The inverse of the correlation matrix
mu_hat Estimate of mu
theta Theta parameters for the correlation
Gaussian_hessianCC

Value

Matrix, the Hessian at XX

Examples

```r
set.seed(0)
n <- 40
x <- matrix(runif(n*2), ncol=2)
f1 <- function(a) {sin(2*pi*a[1]) + sin(6*pi*a[2])}
y <- apply(x,1,f1) + rnorm(n,0,.01)
gp <- GauPro(x,y, verbose=2, parallel=FALSE);gp$theta
gp$hessian(c(.2,.75), useC=TRUE) # Should be -38.3, -5.96, -5.96, -389.4 as 2x2 matrix
```

Gaussian_hessianCC  Gaussian hessian in C

Description

Gaussian hessian in C

Usage

Gaussian_hessianCC(XX, X, Z, Kinv, mu_hat, theta)

Arguments

- **XX**: point to find Hessian at
- **X**: matrix of data points
- **Z**: matrix of output
- **Kinv**: inverse of correlation matrix
- **mu_hat**: mean estimate
- **theta**: correlation parameters

Value

Hessian matrix
Gaussian_hessianR  

Calculate Hessian for a GP with Gaussian correlation

Description

Calculate Hessian for a GP with Gaussian correlation

Usage

Gaussian_hessianR(XX, X, Z, Kinv, mu_hat, theta)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>The vector at which to calculate the Hessian</td>
</tr>
<tr>
<td>X</td>
<td>The input points</td>
</tr>
<tr>
<td>Z</td>
<td>The output values</td>
</tr>
<tr>
<td>Kinv</td>
<td>The inverse of the correlation matrix</td>
</tr>
<tr>
<td>mu_hat</td>
<td>Estimate of mu</td>
</tr>
<tr>
<td>theta</td>
<td>Theta parameters for the correlation</td>
</tr>
</tbody>
</table>

Value

Matrix, the Hessian at XX

Examples

```r
set.seed(0)
n <- 40
x <- matrix(runif(n*2), ncol=2)
f1 <- function(a) {sin(2*pi*a[1]) + sin(6*pi*a[2])}
y <- apply(x,1,f1) + rnorm(n,0,.01)
gp <- GauPro(x,y, verbose=2, parallel=FALSE); gp$theta
gp$hessian(c(0.2,.75), useC=FALSE) # Should be -38.3, -5.96, -5.96, -389.4 as 2x2 matrix
```

kernel_product  

Gaussian Kernel R6 class

Description

Gaussian Kernel R6 class

Usage

kernel_product
**kernel_sum**

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
k1 <- Exponential$new(beta=1)
k2 <- Matern32$new(beta=2)
k <- k1 + k2
k$k(matrix(c(2,1), ncol=1))
```

---

**Description**

Gaussian Kernel R6 class

**Usage**

`kernel_sum`

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
k1 <- Exponential$new(beta=1)
k2 <- Matern32$new(beta=2)
k <- k1 + k2
k$k(matrix(c(2,1), ncol=1))
```
Matern32  

Matern 3/2 Kernel R6 class

Description

Matern 3/2 Kernel R6 class

Usage

Matern32

Format

R6Class object.

Value

Object of R6Class with methods for fitting GP model.

Examples

k1 <- Matern32$new(beta=0)

Matern52  

Matern 5/2 Kernel R6 class

Description

Matern 5/2 Kernel R6 class

Usage

Matern52

Format

R6Class object.

Value

Object of R6Class with methods for fitting GP model.

Examples

k1 <- Matern52$new(beta=0)
Periodic Kernel R6 class

Description
Periodic Kernel R6 class

Usage
Periodic

Format
R6Class object.

Value
Object of R6Class with methods for fitting GP model.

Examples
k1 <- Periodic$new(p=1, alpha=1)

plot.GauPro
Plot for class GauPro

Description
Plot for class GauPro

Usage
## S3 method for class 'GauPro'
plot(x, ...)

Arguments
x Object of class GauPro
...
Additional parameters

Value
Nothing
Examples

n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
y <- sin(2*pi*x) + rnorm(n,0,1e-1)
gp <- GauPro(X=x, Z=y, parallel=FALSE)
if (requireNamespace("MASS", quietly = TRUE)) {
  plot(gp)
}

predict.GauPro

Predict for class GauPro

Description

Predict for class GauPro

Usage

## S3 method for class 'GauPro'
predict(object, XX, se.fit = F, covmat = F,
        split_speed = T, ...)

Arguments

object Object of class GauPro
XX new points to predict
se.fit Should standard error be returned (and variance)?
covmat Should the covariance matrix be returned?
split_speed Should the calculation be split up to speed it up?
... Additional parameters

Value

Prediction from object at XX

Examples

n <- 12
x <- matrix(seq(0,1,length.out = n), ncol=1)
y <- sin(2*pi*x) + rnorm(n,0,1e-1)
gp <- GauPro(X=x, Z=y, parallel=FALSE)
predict(gp, .448)
RatQuad

### Rational Quadratic Kernel R6 class

**Description**
Rational Quadratic Kernel R6 class

**Usage**
RatQuad

**Format**
*R6Class* object.

**Value**
Object of *R6Class* with methods for fitting GP model.

**Examples**
```r
k1 <- RatQuad$new(beta=0, alpha=0)
```

---

trend_0

### Trend R6 class

**Description**
Trend R6 class

**Usage**
trend_0

**Format**
*R6Class* object.

**Value**
Object of *R6Class* with methods for fitting GP model.

**Examples**
```r
t1 <- trend_0$new()
```
trend_c

*Trend R6 class*

**Description**

Trend R6 class

**Usage**

trend_c

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
t1 <- trend_c$new()
```

---

trend_LM

*Trend R6 class*

**Description**

Trend R6 class

**Usage**

trend_LM

**Format**

`R6Class` object.

**Value**

Object of `R6Class` with methods for fitting GP model.

**Examples**

```r
t1 <- trend_LM$new(D=2)
```
Index

*Topic Gaussian
 Exponential, 5
 GauPro_base, 6
 GauPro_Gauss, 7
 GauPro_kernel, 8
 GauPro_kernel_beta, 9
 GauPro_kernel_model, 9
 GauPro_trend, 10
 Gaussian, 11
 kernel_product, 14
 kernel_sum, 15
 Matern32, 16
 Matern52, 16
 Periodic, 17
 RatQuad, 19
 trend_0, 19
 trend_c, 20
 trend_LM, 20

*Topic data,
 Exponential, 5
 GauPro_base, 6
 GauPro_Gauss, 7
 GauPro_kernel, 8
 GauPro_kernel_beta, 9
 GauPro_kernel_model, 9
 GauPro_trend, 10
 Gaussian, 11
 kernel_product, 14
 kernel_sum, 15
 Matern32, 16
 Matern52, 16
 Periodic, 17
 RatQuad, 19
 trend_0, 19
 trend_c, 20
 trend_LM, 20

*Topic kriging,
 Exponential, 5
 GauPro_base, 6
 GauPro_Gauss, 7
 GauPro_kernel, 8
 GauPro_kernel_beta, 9

*Topic process,
 Exponential, 5
 GauPro_base, 6
 GauPro_Gauss, 7
 GauPro_kernel, 8
 GauPro_kernel_beta, 9
 GauPro_kernel_model, 9
 GauPro_trend, 10
 Gaussian, 11
 kernel_product, 14
 kernel_sum, 15
 Matern32, 16
 Matern52, 16
 Periodic, 17
 RatQuad, 19
 trend_0, 19
 trend_c, 20
 trend_LM, 20

*Topic regression
 Exponential, 5
 GauPro_base, 6
 GauPro_Gauss, 7
 GauPro_kernel, 8
 GauPro_kernel_beta, 9
GauPro_kernel_model, 9
GauPro_trend, 10
Gaussian, 11
kernel_product, 14
kernel_sum, 15
Matern32, 16
Matern52, 16
Periodic, 17
RatQuad, 19
trend_0, 19
trend_c, 20
trend_LM, 20
*.GauPro_kernel, 2
+.GauPro_kernel, 3
corr_gauss_matrix, 3
corr_gauss_matrix_sym armaC, 5
corr_gauss_matrix_symC, 4
Exponential, 5
GauPro, 6
GauPro_base, 6
GauPro_Gauss, 7
GauPro_kernel, 8
GauPro_kernel_beta, 9
GauPro_kernel_model, 9
GauPro_trend, 10
Gaussian, 11
Gaussian_devianceC, 12
Gaussian_hessianC, 12
Gaussian_hessianCC, 13
Gaussian_hessianR, 14
kernel_product, 14
kernel_sum, 15
Matern32, 16
Matern52, 16
Periodic, 17
dotPlot_GauPro, 17
dotsPredict_GauPro, 18
R6Class, 5, 6, 8, 9, 11, 15–17, 19, 20
RatQuad, 19
trend_0, 19
trend_c, 20
trend_LM, 20