Package 'leapgp'

July 24, 2024

Type Package

Title Localized Ensemble of Approximate Gaussian Processes

Version 1.0.0

Author Kellin Rumsey [aut, cre]

Maintainer Kellin Rumsey <knrumsey@lanl.gov>

Description An emulator designed for rapid sequential emulation (e.g., MCMC applications). Works via extension
of the laGP approach by Gramacy and Apley (2015). Details are given in Rumsey et al. (2023).

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports laGP, RANN, cluster

Suggests knitr, rmarkdown, testthat, lhs, tictoc, RColorBrewer

NeedsCompilation no

VignetteBuilder knitr

R topics documented:

`leapGP` *Localized Ensemble of Approximate Gaussian Processes*

Description

Function to train or initialize a leapGP model, as described in Rumsey et al. (2023).

Usage

```
leapGP(
  X,
  y,
  M0 = ceiling(sqrt(length(y))),
  rho = NA,
  scale = FALSE,
  n = ceiling(sqrt(length(y))),
  start = NA,
  verbose = FALSE,
  justdoit = FALSE,
  ...
)
```

Arguments

X	a matrix of training locations (1 row for each training instance)
y	a vector of training responses (length(y) should equal nrow(X))
M0	the number of prediction hubs desired. Defaults to ceiling(sqrt(length(Y))).
rho	(optional). The parameter controlling time-accuracy tradeoff. Can also be specified during prediction.
scale	logical. Do we want the scale parameter to be returned for predictions? If TRUE, the matrix K^{-1} will be stored for each hub.
n	local neighborhood size (for laGP)
start	number of starting points for neighborhood (between 6 and n inclusive)
verbose	logical. Should status be printed? Deault is FALSE
justdoit	logical. Force leapGP to run using specified parameters (may take a long time and/or cause R to crash).
...	optional arguments to be passed to laGP()

Details

The leapGP is extends the laGP framework of Gramacy & Apley (2015). The methods are equivalent for rho=1, but leapGP trades memory for speed when rho < 1. The method is described in Rumsey et al. (2023) where they demonstrate that leapGP is faster than laGP for sequential predictions and is also generally more accurate for some settings of rho.

Value

an object of class leapGP with fields X, y, and hubs. Also returns scale parameter if scale=TRUE

References

Gramacy, R. B., & Apley, D. W. (2015). Local Gaussian process approximation for large computer experiments. Journal of Computational and Graphical Statistics, 24(2), 561-578.

Rumsey, K. N., Huerta, G., & Derek Tucker, J. (2023). A localized ensemble of approximate Gaussian processes for fast sequential emulation. Stat, 12(1), e576.

Examples

```
# Generate data
f <- function(x){
   1.3356*(1.5*(1-x[1]) + exp(2*x[1] - 1)*sin(3*pi*(x[1] - 0.6)^2) +
   exp(3*(x[2]-0.5))*sin(4*pi*(x[2] - 0.9)^2))
}
X <- matrix(runif(200), ncol=2)
y <- apply(X, 1, f)

# Generate data for prediction
Xtest <- matrix(runif(200), ncol=2)
ytest <- apply(Xtest, 1, f)

# Train initial model
mod <- leapGP(X, y, M0 = 30)
# Make sequential predictions
pred <- rep(NA, 100)
for(i in 1:100){
  mod <- predict_leapGP(mod, matrix(Xtest[i,], nrow=1), rho=0.9)
  pred[i] <- mod$mean
}
```

`predict_leapGP` *Predict Method for leapGP*

Description

Predict method for an object of class leapGP. Returns a (possibly modified) leapGP object as well as a prediction (with uncertainty, if requested).

Usage

```
predict_leapGP(
  object,
  newdata,
  rho = 0.95,
  scale = FALSE,
  n = ceiling(sqrt(length(y))),
  start = NA,
  M_max = Inf,
  ...
)
```

Arguments

`object`	An object of class `leapGP`
`newdata`	New data
`rho`	parameter controlling time-accuracy tradeoff (default is `rho=0.95`)
`scale`	logical. Do we want the scale parameter to be returned for predictions? If TRUE, the matrix K^{-1} will be stored for each hub.
`n`	local neighborhood size

start	number of starting points for neighborhood (between 6 and n inclusive)
M_max	the maximum number of hubs allowed (used to upper bound the run time)
...	optional arguments to be passed to laGP()

Details

The leapGP is extends the laGP framework of Gramacy & Apley (2015). The methods are equivalent for rho=1, but leapGP trades memory for speed when rho < 1. The method is described in Rumsey et al. (2023) where they demonstrate that leapGP is faster than laGP for sequential predictions and is also generally more accurate for some settings of rho.

Value

A list containing values mean, hubs X and y. If scale=TRUE the list also contains field sd.

References

Gramacy, R. B., & Apley, D. W. (2015). Local Gaussian process approximation for large computer experiments. Journal of Computational and Graphical Statistics, 24(2), 561-578.

Rumsey, K. N., Huerta, G., & Derek Tucker, J. (2023). A localized ensemble of approximate Gaussian processes for fast sequential emulation. Stat, 12(1), e576.

Examples

```
# Generate data
f <- function(x){
   1.3356*(1.5*(1-x[1]) + exp(2*x[1] - 1)*sin(3*pi*(x[1] - 0.6)^2) +
   exp(3*(x[2]-0.5))*sin(4*pi*(x[2] - 0.9)^2))
}
X <- matrix(runif(200), ncol=2)
y <- apply(X, 1, f)

# Generate data for prediction
Xtest <- matrix(runif(200), ncol=2)
ytest <- apply(Xtest, 1, f)

# Train initial model
mod <- leapGP(X, y, M0 = 30)
# Make sequential predictions
pred <- rep(NA, 100)
for(i in 1:100){
  mod <- predict_leapGP(mod, matrix(Xtest[i,], nrow=1), rho=0.9)
  pred[i] <- mod$mean
}
```

Index