

Package ‘FastGP’

October 17, 2026

Type Package

Title Efficiently Using Gaussian Processes with Rcpp and RcppEigen

Version 1.5

Date 2026-10-17

Maintainer Giri Gopalan <gopalan88@gmail.com>

Description Contains Rcpp and RcppEigen implementations of matrix operations useful for Gaussian process models, such as the inverse and determinant of a symmetric and positive-definite Toeplitz matrix, sampling from multivariate normal distributions, evaluation of the log-density of a multivariate normal vector, and Bayesian inference for latent variable Gaussian process models with elliptical slice sampling (Murray, Adams, and MacKay 2010).

License GPL-2

Imports Rcpp, MASS, mvtnorm, stats

Suggests microbenchmark

LinkingTo Rcpp, RcppEigen

Repository CRAN

NeedsCompilation yes

RoxygenNote 7.3.1

Author Giri Gopalan [aut, cre],
Luke Bornn [aut]

Date/Publication 2026-10-17 19:00:02 UTC

Contents

ess	2
rcpp_matrix_ops	3
rcpp_rmvnorm	5
Index	6

Description

The `ess` function uses elliptical slice sampling to draw from the posterior distribution within a Bayesian model in which the prior is a Gaussian process/multivariate normal. If the covariance matrix corresponding to the prior is also Toeplitz, use `ess_Toep` for a faster version.

Elliptical slice sampling is a valid Markov chain Monte Carlo (MCMC) algorithm for posterior sampling when the prior is a Gaussian process/multivariate normal. The prior is assumed to have 0 mean, but a non-zero mean can be used by shifting the location in `log.lik`, and shifting the location of the returned samples likewise.

Murray, I., Adams, R., & MacKay, D. (2010, March). Elliptical slice sampling. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 541-548). JMLR Workshop and Conference Proceedings.

Usage

```
ess(log.lik,Y, Sig, N_mcmc,burn_in,N,flag)
ess_Toep(log.lik,Y, Sig, N_mcmc,burn_in,N)
```

Arguments

<code>log.lik</code>	Log-lik function in model which is assumed to take two arguments: the first contains the parameters/latent variables and the second the observed data <code>Y</code> .
<code>Y</code>	Observed data.
<code>Sig</code>	Covariance matrix associated with the prior distribution on the parameters/latent variable vector.
<code>N_mcmc</code>	Number of desired MCMC samples.
<code>burn_in</code>	Number of burn-in iterations.
<code>N</code>	Dimensionality of parameter/latent variable vector.
<code>flag</code>	Set to TRUE for MASS implementation of <code>mvrnorm</code> , FALSE for FastGP implementation of <code>rcpp_rmvnorm</code> .

Author(s)

Giri Gopalan gopalan88@gmail.com

Examples

```
# See demo/FastGPDemo.r.
set.seed(123)
library(MASS)

N <- 100
```

```

sigma <- 20 #variance parameter in the covariance function
phi <- 100 #decay parameter for the squared-exponential kernel
x_seq <- seq(N)
Sig <- as.matrix(sigma*exp(-as.matrix(dist(x_seq))^2/phi)) + .001*diag(N) #test covariance function

A <- 1 #amplitude of the sin function used for our signal
T <- 5 #period of the sin function used as our signal

#A function that creates a signal, which can be toggled for multiple shapes.
#In the present iteration we use a sin function with variable period.
signal <- function(t) {
  return(A*sin(t/T))
}
#Generate a copy of the signal on the time scale of 1...N
t <- mvrnorm(n = 1, mu = rep(0,N), Sigma = Sig , tol = 1e-6)
s <- signal(seq(1:N)+t)+rnorm(N,sd=sqrt(.001))

#log-lik function
log_lik <- function(w,s){
  return (rcpp_log_dmvnorm(S = .001*diag(1,N),mu=signal(seq(1:N)+w),s,istoep=TRUE))
}

mcmc_samples <- ess_Toep(log_lik, s, Sig, 1000, 250, N)
par(mfrow=c(1,1))
plot(colMeans(mcmc_samples), type="l")
points(t, type="l", lwd=3, col="red")
points(colMeans(mcmc_samples) + 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")
points(colMeans(mcmc_samples) - 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")

mcmc_samples <- ess(log_lik, s, Sig, 1000, 250, N, flag = TRUE)
par(mfrow=c(1,1))
plot(colMeans(mcmc_samples), type="l")
points(t, type="l", lwd=3, col="red")
points(colMeans(mcmc_samples) + 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")
points(colMeans(mcmc_samples) - 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")

mcmc_samples <- ess(log_lik, s, Sig, 1000, 250, N, flag = FALSE)
par(mfrow=c(1,1))
plot(colMeans(mcmc_samples), type="l")
points(t, type="l", lwd=3, col="red")
points(colMeans(mcmc_samples) + 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")
points(colMeans(mcmc_samples) - 2* apply(mcmc_samples, 2, sd), type="l", lty="dashed")

```

Description

Performs matrix operations using Rcpp and RcppEigen, and some custom C++ to handle symmetric and positive-definite Toeplitz matrices.

Usage

```
rcppeigen_invert_matrix(A)
rcppeigen_get_det(A)
rcppeigen_get_chol(A)
tinv(A)
tdet(A)
rcpp_distance(A, nrow, ncol)
```

Arguments

A	Matrix to perform operation on.
nrow	Number of rows in A.
ncol	Number of columns in A.

Details

Functions with "rcppeigen" directly call RcppEigen implementations of the associated functions, specifically for inverse, determinant, and lower triangular Cholesky factor. The function `tinv` computes an inverse and `tdet` returns the log-determinant of a symmetric and positive-definite Toeplitz matrix using methods from Trench and Durbin from "Matrix Computations" by Golub and Van Loan. `Rcpp_distance` computes the pairwise distances between rows of input matrix.

Author(s)

`gopalan88@gmail.com`

Examples

```
# See demo/FastGPDemo.R

N <- 5
sigma <- 1 #variance parameter in the covariance function
phi <- 5 #decay parameter for the squared-exponential kernel
x_seq <- seq(N)
Sig <- as.matrix(sigma*exp(-as.matrix(dist(x_seq))^2/phi)) + .001*diag(N) #test covariance function

rcppeigen_invert_matrix(Sig)
rcppeigen_get_det(Sig)
rcppeigen_get_chol(Sig)
tinv(Sig)
#NOTE: tdet returns the log of determinant
tdet(Sig)
rcpp_distance(matrix(seq(1,N^2),nrow=N), N, N)
```

Description

These functions allow for the sampling of and evaluation of the log-density of a multivariate normal vector.

Usage

```
rcpp_log_dmvnorm(S,mu,x, istoep)
rcpp_rmvnorm(n,S,mu)
rcpp_rmvnorm_Toep(n,S,mu)
rcpp_rmvnorm_stable(n,S,mu)
```

Arguments

S	Covariance matrix of associated multivariate normal.
n	Number of (independent) samples to generate.
mu	Mean vector.
x	Vector of observations to evaluate the log-density of.
istoep	set this to TRUE if S is Toeplitz.

Author(s)

Giri Gopalan gopalan88@gmail.com

Examples

```
#See demo/FastGPDemo.R, which includes timing comparisons
set.seed(123)

N <- 5
sigma <- 1 #variance parameter in the covariance function
phi <- 5 #decay parameter for the squared-exponential kernel
x_seq <- seq(N)
Sig <- as.matrix(sigma*exp(-as.matrix(dist(x_seq))^2/phi)) + .001*diag(N) #test covariance function
mu <- rep(0, N)

rcpp_log_dmvnorm(Sig,mu,rep(.1,N), istoep = TRUE)
rcpp_rmvnorm(n=1,Sig,mu)
rcpp_rmvnorm_Toep(n=1,Sig,mu)
rcpp_rmvnorm_stable(n=1,Sig,mu)
```

Index

`durbin(rcpp_matrix_ops)`, [3](#)

`ess`, [2](#)

`ess-Toep(ess)`, [2](#)

`rcpp_distance(rcpp_matrix_ops)`, [3](#)

`rcpp_log_dmvnorm(rcpp_rmvnorm)`, [5](#)

`rcpp_matrix_ops`, [3](#)

`rcpp_rmvnorm`, [5](#)

`rcpp_rmvnorm_stable(rcpp_rmvnorm)`, [5](#)

`rcpp_rmvnorm-Toep(rcpp_rmvnorm)`, [5](#)

`rcppeigen_get_chol(rcpp_matrix_ops)`, [3](#)

`rcppeigen_get_chol_diag`
 `(rcpp_matrix_ops)`, [3](#)

`rcppeigen_get_det(rcpp_matrix_ops)`, [3](#)

`rcppeigen_get_diag(rcpp_matrix_ops)`, [3](#)

`rcppeigen_invert_matrix`
 `(rcpp_matrix_ops)`, [3](#)

`rcppeigen_pivot_chol(rcpp_matrix_ops)`,
 [3](#)

`tdet(rcpp_matrix_ops)`, [3](#)

`tinvc(rcpp_matrix_ops)`, [3](#)

`toep_chol_inv(rcpp_matrix_ops)`, [3](#)

`toep_logdet(rcpp_matrix_ops)`, [3](#)

`trench(rcpp_matrix_ops)`, [3](#)