simulation_gp

Joonwon Lee

2023-10-17

simulation

Simulate Gaussian process

For data generation, define covariance function as

$\gamma(d) = \theta * exp(-|d|)$, and $\theta = 2$, where $d = ||x - y||_2, x, y \in \mathbb{R}^2$. distance between two input points.

```
# Set the number of points you want
n <- 100

# Define the range of the square
x_min <- 0
x_max <- 1
y_min <- 0
y_max <- 1

# Generate random x and y coordinates randomly
x <- runif(n, min = x_min, max = x_max)
y <- runif(n, min = y_min, max = y_max)

# x <- seq(0,10, length.out = n)
# y <- seq(0,10, length.out = n)

# Define covaraince function
theta <- 2 ; range = 2
d <- sqrt(outer(x,x,"-")^2 )+ sqrt(outer(y,y,"-")^2 )

# Construct n by n covariance matrix
cov_mat <- theta * exp(-d/range)
chol_matrix <- chol(cov_mat) # returns L^T where L is lower triangular.

# Generate finite subset of Gaussian Process
# whose covariance matrix is defined as above
z_list <- rnorm(n)
simulated_data <- t(chol_matrix) %*% z_list
```

covfun_name

We can fit the model using fit_model function in GpGp package by specifying covariance function as below:

matern15_scaledim: $M(x,y) = \sigma^2(1 + |D^{-1}|x - y||)exp(-||D^{-1}x - y||)$,

exponential_isotropic: $M(x,y) = \sigma^2 exp(-||x - y||/\alpha)$.

Remark: options in fit_model():

1. reorder=FALSE: maxmin ordering is not used unless nrow(locs) > 1e5.
2. m_seq: By default, a 10-neighbor approximaation is maximized , then a 30-neighbor approximation is maximized using 10 neighbor estimates as starting values.

```
# set location matrix and design matrix X.
locs <- as.matrix(data.frame(x = x, y = y))
X <- cbind(rep(1,n), locs)

# fit the model using 'matern15_scaledim'
mod1 <- fit_model(simulated_data, locs = locs, X = X ,
                  covfun_name = 'matern15_scaledim', fixed_parms = c(2,3), start_parms = c(3,2,2,0))
# parameters in matern15_scaledim: variance, range_1,...range_d, nugget
# fixed_parms = c(2,3) means fix range_1 and range_2
# starting from start_parms = c(3,1,1,0).

# fit the model using the true covariance function
mod2 <- fit_model(simulated_data, locs = locs, X = X,
                  covfun_name = 'exponential_isotropic',
fixed_parms = c(2), start_parms = c(3,2,0))
```

```
cat("Given ", n, "samples with true [theta, range] =", matrix(c(2,2),nrow=1),
 ",\n\nestimate of (theta, range ) by fitting with matern15_isotropic are :", mod1$covparms[1:2],
"\n\nestimate of (theta, range ) by fitting with true exponential_isotropic are :", mod2$covparms[1:2])
```

```
## Given  100 samples with true [theta, range] = 2 2 ,
##
## estimate of (theta, range ) by fitting with matern15_isotropic are : 326.1131 2
##
## estimate of (theta, range ) by fitting with true exponential_isotropic are : 2.559718 2
```

Generate data using matern15_scaledim:

In this section, I wanted to do the opposite, generate data using $M(x,y) = \sigma^2(1+||x-y||/\alpha)exp(-||D^{-1}x - y||/\alpha)$, and fit the model using exponential covariance function but I couldn't generate data because the covariance matrix is not positive definite. There are nearly 0 eigen values.

```
theta = 2 ; range = 1
cov_mat2 <- theta * (1+ d/range) * exp(-d/range)
round(eigen(cov_mat2)$values,3)
```

```
##   [1] 171.610  13.616  11.197   2.368   2.039   1.116   0.924   0.505   0.458
##  [10]   0.278   0.253   0.190   0.153   0.122   0.109   0.082   0.075   0.074
##  [19]   0.048   0.041   0.039   0.032   0.028   0.027   0.021   0.019   0.016
##  [28]   0.013   0.012   0.012   0.008   0.007   0.005   0.004   0.003   0.002
##  [37]   0.002   0.000   0.000   0.000   0.000  -0.001  -0.001  -0.001  -0.002
```

```
##  [46]  -0.003  -0.003  -0.003  -0.004  -0.005  -0.006  -0.007  -0.007  -0.008
##  [55]  -0.008  -0.010  -0.010  -0.011  -0.012  -0.013  -0.015  -0.016  -0.016
##  [64]  -0.018  -0.019  -0.020  -0.022  -0.024  -0.024  -0.026  -0.027  -0.029
##  [73]  -0.030  -0.031  -0.036  -0.037  -0.042  -0.043  -0.044  -0.052  -0.053
##  [82]  -0.057  -0.064  -0.072  -0.073  -0.080  -0.094  -0.101  -0.108  -0.123
##  [91]  -0.142  -0.144  -0.173  -0.217  -0.281  -0.321  -0.333  -0.610  -0.663
## [100]  -1.111
```

```
# data2 <- mvrnorm(n = n, mu = rep(0,n), Sigma = cov_mat)
```

Use MLE

For this section, I fit parameters using MLE of covariance matrix:

```
# Generate 10000 observations of multivariate normal
# having exponential covariance function
n2 = 10000
theta = 2; range = 2
cov_mat <- theta * exp(-d/range)
data <- mvrnorm(n = n2, mu = rep(0,n), Sigma = cov_mat)

# mle of covariance matrix
mle_cov = 1/n2 * t(data) %*% data

# fit model using matern15_scaledim: sigma^2 * (1+d)%*% exp(-d/range)

sigma_sq = diag(mle_cov %*% solve( (1+d/range)* exp(-d/range) ))

# fite model using true exponential: sigma^2 * exp(-d/range)

sigma_sq2 = diag(mle_cov %*% solve(exp(-d/range) ))

cat("Given ", n2, "samples with true [theta, range] =", matrix(c(2,2),nrow=1),
 ",\n\nestimate of (theta, range ) by fitting with matern15_isotropic are :", mean(sigma_sq),
"\n\nestimate of (theta, range ) by fitting with true exponential_isotropic are :", mean(sigma_sq2) )
```

```
## Given  10000 samples with true [theta, range] = 2 2 ,
##
## estimate of (theta, range ) by fitting with matern15_isotropic are : 570.7625
##
## estimate of (theta, range ) by fitting with true exponential_isotropic are : 2.002314
```