Spatio Temporal Gaussian Process Real

David Arthur

November 2, 2020

```
library(mvtnorm)
library(coda)
```

```
## Warning: package 'coda' was built under R version 3.5.3
```

```
space.kern <- function(D, l2){
  exp(-1/(2*l2)*D)
}

space.D <- function(lat_long){
  D <- as.matrix(dist(lat_long, diag = TRUE, upper = TRUE))^2
}

time.kern <- function(D, l2){
  exp(-1/(2*l2)*D)
}

time.D <- function(tt){
  D <- as.matrix(dist(tt, diag = TRUE, upper = TRUE))^2
}

gau.pro.spat.temp <- function(data, niter, l2_space = 0.005, l2_time = 10,
                              prior_a = 2, prior_b = 2, var_init = 2){
  Y <- log(data[,3]+1)
  tt <- unique(data[,2])

  Ymean <- mean(Y)
  Ysd <- sd(Y)
  Y <- (Y-Ymean)/Ysd

  lat_long <- unique(data[,4:5])
  rownames(lat_long) <- NULL

  N <- length(tt)*nrow(lat_long)

  D_space <- space.D(lat_long)
  K_space <- space.kern(D_space, l2_space) + diag(10e-06, nrow(lat_long))
  D_time <- time.D(tt)
  K_time <- time.kern(D_time, l2_time) + diag(10e-06, length(tt))

  K_space_inv <- solve(K_space)
  K_time_inv <- solve(K_time)

  f_save <- matrix(0, niter, length(Y))
  sig2_save <- rep(0, niter)
  tau2_space_save <- rep(0, niter)
```

```
  tau2_time_save <- rep(0, niter)

  sig2_a <- prior_a
  sig2_b <- prior_b
  tau2_time_a <- prior_a
  tau2_time_b <- prior_b
  tau2_space_a <- prior_a
  tau2_space_b <- prior_b

  sig2_save[1] <- var_init
  tau2_space_save[1] <- var_init
  tau2_time_save[1] <- var_init

  for (i in 2:niter){
    K_space_time <- kronecker(tau2_space_save[i-1]*K_space,
                          tau2_time_save[i-1]*K_time)
    sig2I <- diag(sig2_save[i-1], N)
    f_Sigma <- K_space_time + sig2I

    f_Mu <- K_space_time%*%solve(f_Sigma, Y)
    f_Sigma <- K_space_time - K_space_time%*%solve(f_Sigma, K_space_time)

    f_save[i,] <- rmvnorm(1, f_Mu, f_Sigma)

    sig2_save[i] <- 1/rgamma(1, sig2_a + N/2,
                             sig2_b + 0.5*t(Y-f_save[i,])%*%(Y-f_save[i,]))

    tau2_space_Sigma <- kronecker(K_space_inv, 1/tau2_time_save[i-1]*K_time_inv)

    tau2_space_save[i] <- 1/rgamma(1, tau2_space_a + N/2,
                                tau2_space_b + 0.5*t(f_save[i,])%*%tau2_space_Sigma%*%f_save[i,])

    tau2_time_Sigma <- kronecker(1/tau2_space_save[i]*K_space_inv, K_time_inv)

    tau2_time_save[i] <- 1/rgamma(1, tau2_time_a + N/2,
                                  tau2_time_b + 0.5*t(f_save[i,])%*%tau2_time_Sigma%*%f_save[i,])
  }

  cbind(f_save, sig2_save, tau2_space_save, tau2_time_save)
}

load(file = "C://Users/david/Documents/R/Stat 695/Project/uber/trip_data.RData")

# gaupro_out1 <- gau.pro.spat.temp(trip_data, niter = 500, var_init = 100)
# gaupro_out2 <- gau.pro.spat.temp(trip_data, niter = 500, var_init = 10)
# gaupro_out3 <- gau.pro.spat.temp(trip_data, niter = 500, var_init = 0.01)
#
# save(gaupro_out1, file = "C://Users/david/Documents/R/Stat 695/Project/uber/st_gaupro_out1.RData")
# save(gaupro_out2, file = "C://Users/david/Documents/R/Stat 695/Project/uber/st_gaupro_out2.RData")
# save(gaupro_out3, file = "C://Users/david/Documents/R/Stat 695/Project/uber/st_gaupro_out3.RData")
#
# f_out <- gaupro_out[,1:600]
# f_pred <- colMeans(exp(f_out*sd(log(trip_data$n_trips+1)) + mean(log(trip_data$n_trips+1))))
```

```
#
# f_pred_mat <- matrix(f_pred, nrow = 24, ncol = 25, byrow = FALSE)
# trip_mat <- matrix(trip_data$n_trips, nrow = 24, ncol = 25, byrow = FALSE)
#
# for (i in 1:25){
#   plot(0:23, trip_mat[,i], main = paste("Number of Hourly Trips (with Predictions) for Location", i),
#           xlab = "Hour", ylab = "Number of Trips")
#   lines(0:23, f_pred_mat[,i], col = "red")
# }
```