Untitled

new

4/4/2025

Initialization

```
library(knitr)
library(varycoef)
```

```
## Warning: package 'varycoef' was built under R version 4.4.3
```

```
library(ggplot2)
library(gridExtra)
library(Matrix)
library(here)
```

```
## here() starts at C:/Users/s-edw/Documents
```

```
library(svc)
set.seed(123)

calc_C_phi = function(coords, phi) {
  n = nrow(coords)
  dists = as.matrix(dist(coords)) ^ 2
  C_phi = exp(-0.5 / phi * dists)
}

mcmc = 1000
```

Purpose

The purpose of this simulation study is compare the performance between our method

and varycoef R package. Specfically, we will assess the bias, computational speed, average length of credible interval

and coverage

Setting

1 replications (due to time constraint)

We will perform 1000 iterations of mcmc

We will vary by 100 or 500 sample size

We will vary by number of betas (2 or 3) and priors (low or high)

metric

Bias

How long to run in seconds

```
gibbs1 = data.frame()
coef1 = data.frame()

  lat = seq(0, 9, by = 1)
  lon = seq(0, 9, by = 1)

  coords = as.matrix(expand.grid(lat, lon))
  colnames(coords) = c("lat", "lon")

  n = nrow(coords)
  p = 2

  # generating w
  sigmasq_w = 1
  phi_w = 2

  C_w = calc_C_phi(coords, phi_w)
  w = MASS::mvrnorm(1, rep(0, n), sigmasq_w * C_w)
```

```r
# generating beta's
sigmasq_1 = 1
phi_1 = 2

C_1 = calc_C_phi(coords, phi_1)
beta_1_mean = 3
beta_1 = MASS::mvrnorm(1, rep(beta_1_mean, n),sigmasq_1 * C_1 )

sigmasq_2 = 1
phi_2 = 2

C_2 = calc_C_phi(coords, phi_2)
beta_2_mean = -5
beta_2 = MASS::mvrnorm(1, rep(beta_2_mean, n),sigmasq_2 * C_2)

# generating X
X_1 = rnorm(n, 0, 10)
X_2 = rnorm(n, 10, 100)

# generating epsilon
tausq = 0.0001
epsilon = rnorm(n, mean = 0, sd = sqrt(tausq))

Y = X_1 * beta_1 + X_2 * beta_2 + w + epsilon

k = 2

lat_knots = unique(lat)
lat_knots = lat_knots[seq(1, length(lat_knots), by = k)]

lon_knots = unique(lon)
lon_knots = lon_knots[seq(1, length(lon_knots), by = k)]

knots = as.matrix(expand.grid(lat_knots, lon_knots))

df = data.frame(coords, Y, X_1, X_2, w, beta_1, beta_2, epsilon)

knots_df = data.frame(knots)
colnames(knots_df) = c("lat", "lon")

knots_df = merge(knots_df, df, by = c("lat", "lon"), all.x = TRUE, all.y = FALSE)

X = as.matrix(df[, c("X_1", "X_2")])
Y_star = knots_df$Y
X_star = as.matrix(knots_df[, c("X_1", "X_2")])

m = nrow(knots)
l = 0
u = 4
start_time <- Sys.time()
  output = svc::svclm_beta(Y = Y,
                           X = X,
                           s = coords,
```

```
                          Y_knots = Y_star,
                          X_knots = X_star,
                          knots = knots,
                          beta_knots_start = matrix(0, nrow = m, ncol = p),
                          phi_beta_start = rep(l + (u - l) / 2, p),
                          sigmasq_beta_start = rep(1, p),
                          tausq_start = 1,
                          phi_beta_proposal_sd = rep(0.5, p),
                          phi_beta_lower = rep(l, p),
                          phi_beta_upper = rep(u, p),
                          mcmc = mcmc)
  end_time <- Sys.time()
  time_taken <- as.numeric(difftime(end_time, start_time, units = "secs"))
  gibbs1 = rbind(gibbs1,c(mean(colMeans(output$beta_samples[800:1000, ,1]) -3),mean(colMeans(output$beta
  X = cbind(1,X)
  start_time <- Sys.time()
  fit <- SVC_mle(y = Y, X = X,locs = coords)
  end_time <- Sys.time()
  time_taken <- as.numeric(difftime(end_time, start_time, units = "secs"))

  coef1 = rbind(coef1,c(fit$coefficients[2]-3,fit$coefficients[3]+5,time_taken))
```

```
combined_table <- rbind(colMeans(gibbs1), colMeans(coef1))

rownames(combined_table) <- c("Gibbs Sampler", "varycoef")
kable(combined_table, caption = "Scenario 1: (2 betas and low priors) - 100 sample size",col.names = c(
```

Table 1: Scenario 1: (2 betas and low priors) - 100 sample size

	Beta 1 - Bias	Beta 2 - Bias	Time to compute on average
Gibbs Sampler	-9.2243968	4.9657138	0.0549161
varycoef	0.3422649	0.6374244	4.8320041

```
# combined_table <- rbind(colMeans(gibbs2), colMeans(coef2))
#
# rownames(combined_table) <- c("Gibbs Sampler", "varycoef")
# kable(combined_table, caption = "Scenario 2: (2 betas and low priors) - 500 sample size",col.names =
```

Interpretations of results

1. Our model (Gibbs sampler) has high bias compared to varycoef
2. Our model (Gibbs sampler) is quick to compute compared to varycoef

Next steps

Improve the model Gibbs sampler for better performance (bias) Add another metric (EES/sec) to compare between two methods