STDA Homework 1

Seokjun Choi

April 10, 2020

Note:
You can get full, run-able code files at my github page: Visit https://github.com/letsjdosth/SpaTempoDA
Because I provide the full code file separately, in this report, I will show key-code block only instead of bringing the full, verbose code.

1 Problem 1

Suppose we want to simulate a random vector $Y \sim N(\mu, \Sigma)$. If Σ (Matern) is symmetric and positive definite, it can be represented using the Cholesky decomposition $\Sigma = LL'$, where L is a lower triangular matrix. Consider the following algorithm for simulating Y:

- **Calculate the matrix L.**
- **Sample $Z \sim N(0, I)$, where I is the $n \times n$ identity matrix.**
- **Let $Y = \mu + LZ$**

1.1 Problem 1-(a)

Show that Y generated in this way has the correct distribution. You may use the fact that a linear function of a multivariate normal random variable is again multivariate normal; just show the mean and variance are correct.

From the elementary observation from mathematical statistics, we have for $a, b \in \mathcal{R}$ and for a random variable X,

$$E[aX + b] = aE[X] + b$$
$$Var[aX + b] = a^2 Var[X]$$

And its multivariate version is, for proper $A \in \mathcal{R}^{n \times n}$, $b \in \mathcal{R}^n$, and for a random vector X,

$$E[Ax + b] = AE[X] + b$$
$$Var[AX + b] = AVar[X]A'$$

Then, using this observation and the fact at this problem, we can get

$$Y \sim N(\mu + 0, LIL')$$

Since $LL' = \Sigma$ by our setting, we get $Y \sim N(\mu, \Sigma)$, so we can confirm that the algorithm is valid.

1.2 Problem 1-(b)

Write a function or a few lines of code in R to implement this method for arguments mu and Sigma. You may use the built-in function chol for the Cholesky decomposition and rnorm to generate Z.

Here is implementation. Also you may see the 'HW1sol_problem.R' file.

#R code

```
rnorm_chol_algorithm = function(mu, Sigma) {
    # mu is vector
    # Sigma should be symmetric, positive definite matrix
    dimension = dim(Sigma)
    vecZ = rnorm(dimension, 0, 1)
    chol_upperL = chol(Sigma)
    return(mu + t(chol_upperL) %*% vecZ)
}
```

1.3 Problem 1-(c)

For a mean and covariance function of your choosing, use your code from (b) and make a few plots illustrating realizations of a Gaussian process on [0;1], but changing the different parameters in the model. These differences will be easier to see if you keep the same Z sample but just change mu and Sigma.

For convenience, I use fields$Matern function to calculate covariance matrix from parameters of Matern. test2: $\nu = 0.5$ test3: $\nu = 0.1$ test4: $\nu = 1$

2 Problem 2

The file CAtemps.RData contains two R objects of class SpatialPointsDataFrame, called CAtemp and CAgrid. CAtemp contains a average temperatures from 1961-1990 at 200 locations (latitude and longitude) in California in degrees Fahrenheit, along with their elevation in meters. CAgrid contains elevations in meters over a grid of locations. I've given you some code to get started with this data in HW1.R

Consider the following model for the temperature data.

$$Y_i = \mu(s_i;\beta) + e(s_i;\sigma^2,\rho,\tau)$$

where $\mu(s_i;\beta) = \beta_0 + \beta_1 Longitude(s) + \beta_2 Latitude(s) + \beta_3 Elevation(s)$ and $e(s_i;\sigma^2,\rho,\tau)$ is a zero mean stationary Gaussian process with exponential covariance function.

Another way of writing this is as

$$Y_i = \mu(s_i;\beta) + Z(s_i;\sigma^2,\rho) + \epsilon_i$$

where now Z is mean zero Gaussian process like e but without the nugget term, and the ϵ are iid $N(0,\tau^2)$, independent of Z. This is important because we want to predict $\mu(s_i;\beta)+Z(s_i;\sigma^2,\rho)$ without the measurement error.

2.1 Problem 2-(a)

Using the CAtemp data, form a preliminary estimate of β using ordinary least squares and make a color plot of the residuals. Include your estimates and plot.

2.2 Problem 2-(b)

Estimate the variogram non-parametrically and then fit the exponential variogram to it using weighted least squares. Make and include a plot of the nonparametric and parametric variogram functions. Also store your parameter estimates and report them.

2.3 Problem 2-(c)

We will now form the GLS estimate of β by hand, rather than using the gls function.

- **Use the rdist function in fields to create a matrix of distance (in miles) between pairs of locations in CAtemp.**
- **Create the covariance matrix, plugging in your estimates from the fitted variogram. (Hint: Sum tow matrix, one without a nugget and one using the diag function to create the matrix $\tau^2 I$)**
- **Invert the covariance matrix and store it for later reference.**
- **Create the X matrix (Hint: Use cbind.)**
- **Put all the pieces together to form $\hat{\beta}_{GLS}$.**

2.4 Problem 2-(d)

Calculate and plot the EBLUP of $\mu + Z$ at the location in CAgrid, plugging in your estimates from (b) and (c). Calculate and plot the (estimated) standard error of Z at each prediction location

Since $Z \sim N(0, C)$, by definition of functional distribution in weak sense, $\langle Z, x\rangle \sim N(\langle 0, x\rangle, \langle C(x), x\rangle)$ for all $x \in \mathcal{H} = \mathcal{L}^2([0,1])$. So, with eigenfunctions $\{v_i\}$, we get multivariate normal distribution for

$$\{\langle Z, v_i\rangle\}_{i=1,2,...,m} \sim Normal_m(\begin{bmatrix}\langle 0, v_1\rangle \\ \langle 0, v_2\rangle \\ ... \\ \langle 0, v_m\rangle\end{bmatrix}, \begin{bmatrix}\langle C(v_1), v_1\rangle & \langle C(v_1), v_2\rangle & ... & \langle C(v_1), v_m\rangle \\ \langle C(v_2), v_1\rangle & \langle C(v_2), v_2\rangle & ... & \langle C(v_2), v_m\rangle \\ & ... & & \\ \langle C(v_m), v_1\rangle & \langle C(v_m), v_2\rangle & ... & \langle C(v_m), v_m\rangle\end{bmatrix})$$

For simplicity, denote above expression's covariance matrix as Σ_m. Then we simply write above as

$$\{\langle Z, v_i\rangle\}_{i=1,2,...,m} \sim Normal_m(0, \Sigma_m)$$

Since the pdf of multivariate normal is well known, I write it without additional explanation. If denote the vector $[\langle Z, v_i\rangle]_i, i = 1, 2, ..., m$ as z_m, then

$$f_m(z_m) = \frac{1}{(\sqrt{2\pi})^m det(\Sigma_m)} exp(-\frac{1}{2}(z_m - 0)^T \Sigma_m^{-1}(z_m - 0))$$

Likewise, since $X := \mu + Z$,

$$\{\langle X, v_i\rangle\}_{i=1,2,...,m} \sim Normal_m(\begin{bmatrix}\langle \mu, v_1\rangle \\ \langle \mu, v_2\rangle \\ ... \\ \langle \mu, v_m\rangle\end{bmatrix}, \begin{bmatrix}\langle C(v_1), v_1\rangle & \langle C(v_1), v_2\rangle & ... & \langle C(v_1), v_m\rangle \\ \langle C(v_2), v_1\rangle & \langle C(v_2), v_2\rangle & ... & \langle C(v_2), v_m\rangle \\ & ... & & \\ \langle C(v_m), v_1\rangle & \langle C(v_m), v_2\rangle & ... & \langle C(v_m), v_m\rangle\end{bmatrix})$$

with denoting the mean vector as μ_m covariance matrix as Σ_m. Then we simply write above as

$$\{\langle X, v_i\rangle\}_{i=1,2,...,m} \sim Normal_m(\mu_m, \Sigma_m)$$

and pdf is, where $x_m = [\langle X, v_i\rangle]_i, i = 1, 2, ..., m$,

$$f_m(x_m) = \frac{1}{(\sqrt{2\pi})^m det(\Sigma_m)} exp(-\frac{1}{2}(x_m - \mu_m)^T \Sigma_m^{-1}(x_m - \mu_m))$$