# Simulation Sample Generator Design

Yunbi Nam

11/20/2020

In the simulation study, we consider as domain of the functional observations a triangulated surface $\mathcal{M}_{\mathcal{T}}$ with 642 nodes that approximates the brainstem. On this triangulated surface, we select 3 smooth orthonormal functions $\{\mathbf{v}_i\}_{i=1,2,3}$, from eigenfunctions of the Laplace–Beltrami operator. We then generate $n = 100$ curves of individuals randomly drawn from the $k$th class. $x_1, ..., x_{100}$ on $\mathcal{M}_{\mathcal{T}}$ by

$$x_i = u_{i1}v_1 + u_{i2}v_2 + u_{i3}v_3, \qquad i = 1, ..., n, \tag{1}$$

where $u_{i1}, u_{i2}, u_{i3}$ are independent random variables that are distributed as $u_{il} \sim N(\mu_k, {\sigma_k}^2)$. We assume that we have three classes, with $\mu_1 = 50$, $\mu_2 = 30$, and $\mu_3 = 10$, and $\sigma_1 = 1, \sigma_2 = 2$, and $\sigma_3 = 3$. The smooth functions $x_i$ are then sampled at locations $p_j \in \mathbb{R}$ with $j = 1, ..., s$ coinciding with the nodes of the triangulated surface. Moreover, at each of these points we add to the functions a Gaussian noise with mean zero and standard deviation $\sigma = 0.1$ to obtain the noisy observations denoted with $x_i(p_j)$.

As discussed in the meeting, we implicitly select $\Sigma$ as the following:

$$E[x_i(p_t)] = E[u_{i1}]v_1(p_t) + E[u_{i2}]v_2(p_t) + E[u_{i3}]v_3(p_t) = \mu_k v_1(p_t) + \mu_k v_2(p_t) + \mu_k v_3(p_t),$$

and

$$\Sigma(x_i(p_t), x_i(p_u)) = E[(x_i(p_t) - \mu_k)(x_i(p_u) - \mu_k)]$$

Let $\mathbf{X}$ be the vector of observations of $x_i$ at locations $p_1, ..., p_s$. Then

$$\mathbf{X} \sim N(\mu_k, \Omega + \sigma^2 I)$$

where

$$\mu_k = \begin{pmatrix} \mu_k(p_1) \\ \mu_k(p_2) \\ \vdots \\ \mu_k(p_s) \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Sigma(p_1,p_1) & \Sigma(p_1,p_2) & \cdots & \Sigma(p_1,p_s) \\ \Sigma(p_2,p_1) & \Sigma(p_2,p_2) & \cdots & \Sigma(p_2,p_s) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(p_s,p_1) & \Sigma(p_s,p_2) & \cdots & \Sigma(p_s,p_s) \end{pmatrix}$$

```
# Arguments:
# eigV:       r by c matrix, c eigenfunctions of the Laplace-Beltrami operator
# m:          the number of base eigenfunctions using in linear combination to generate x_ii
# n:          the number of generating samples
# coef.mean:  mean of coefficients in linear combination
# coef.var:   variance of coefficients in linear combination
# noise_x:    varaince of random noise on x
# noise_y:    varaince of random noise on y
# ---------------------------------------------------------------------
# Outputs:
# A:          n by (r+1) matrix [y X]
```

```r
# y:           n by 1 matrix
# X:           n by r matrix

mysamples <- function(eigV, R0, m = 3, n = 100, coef.mean, coef.var, noise_x, noise_y){

        r <- nrow(eigV)
        c <- ncol(eigV)

        X <- matrix(0, nrow=n, ncol=r) # n samples by r nodes

        for(i in 1:n){

                base_index <- sample(1:c,m)
                # sample different bases v_1, v_2, v_3 for each individual
                base <- eigV[, base_index]

                coef <- matrix(rnorm(m, coef.mean, coef.var), nrow=m)
                # the distribution of coefficients are same in the same class
                x <- base%*%coef + rnorm(r, 0, noise_x)
                X[i,] <- x

        }

        return(X)
}
```