

Analyzing Sparse Functional Data with SME package

Tahmidul Islam

9/5/2020

Data

Sample size is 20. For sparse functional data setting, each subject is observed on possibly different time points for varying number of times.

Fully Sparse (fdata1)

Here, the time points are chosen randomly from a uniform distribution. First, for a given subject, n_i , the number of time points to be observed on is selected randomly from integers 2 to 10. Then, n_i time points t_{ij} s are selected from a uniform(0,1). Here, the chance of replication at the same time point is low.

Artificially Sparsified from Regular Grid (fdata2)

In this case, a grid of 100 equally spaced time points is created. A Bernoulli random variable is generated with probability 0.10 for each of the time points for its inclusion in the dataset. Here, the chance of replication at the same time point is greater.

```
fdata1 <- read.table('fdata1.txt', header = T) #Fully sparse
fdata2 <- read.table('fdata2.txt', header = T) #Sparsified
```

```
head(fdata1)
```

```
##           t           y id
## 1 0.36711119 -1.1997397  1
## 2 0.62332094 -1.6435587  1
## 3 0.87321315 -1.5343471  1
## 4 0.93903194  0.6864373  1
## 5 0.01967984  0.8496050  2
## 6 0.40799443  0.6932046  2
```

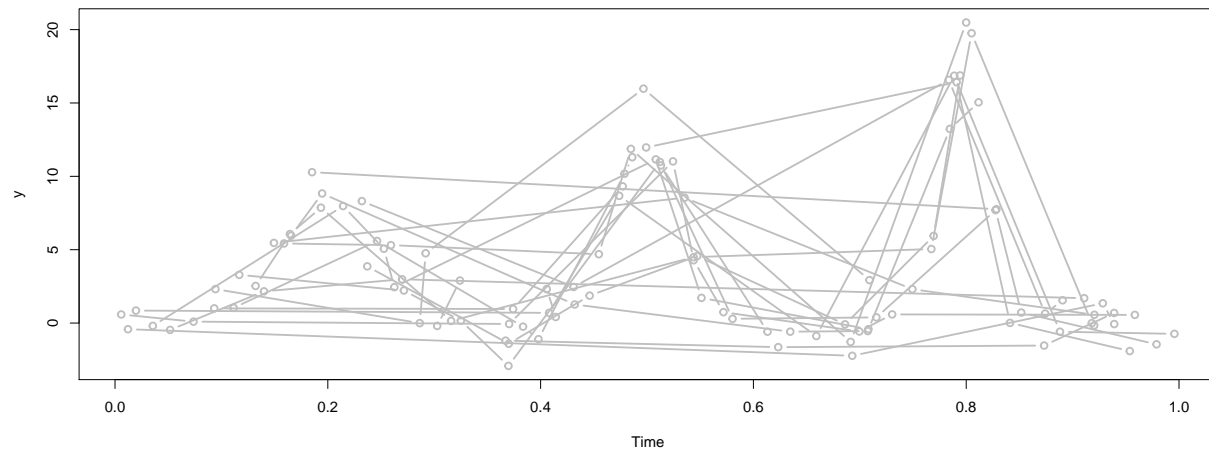
```
head(fdata2)
```

```
##           t           y id
## 1 0.02020202  2.728347  1
## 2 0.14141414  4.881323  1
## 3 0.24242424  7.694276  1
## 4 0.29292929  4.343300  1
## 5 0.32323232  3.352391  1
## 6 0.40404040  2.479632  1
```

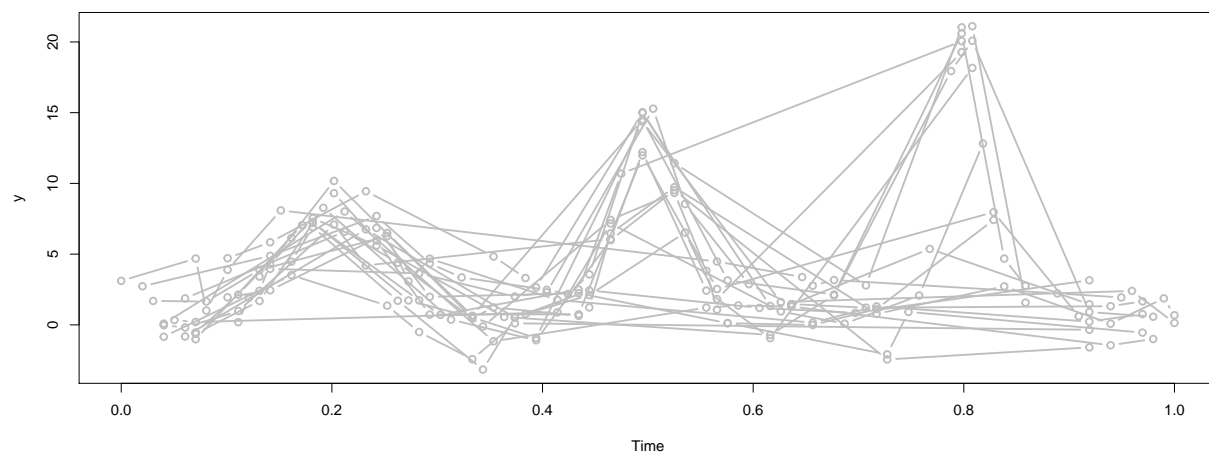
Function to display functional data

```
# plot functional data
plotFdata <- function(id, x, y) {
  plot(x, y, type='n', xlab="Time", ylab="y")
  for(i in unique(id)){
    lines(x[id == i], y[id == i], col = 'grey', type = 'b', lwd = 2)
  }
}

plotFdata(fdata1$id, fdata1$t, fdata1$y)
```



```
plotFdata(fdata2$id, fdata2$t, fdata2$y)
```



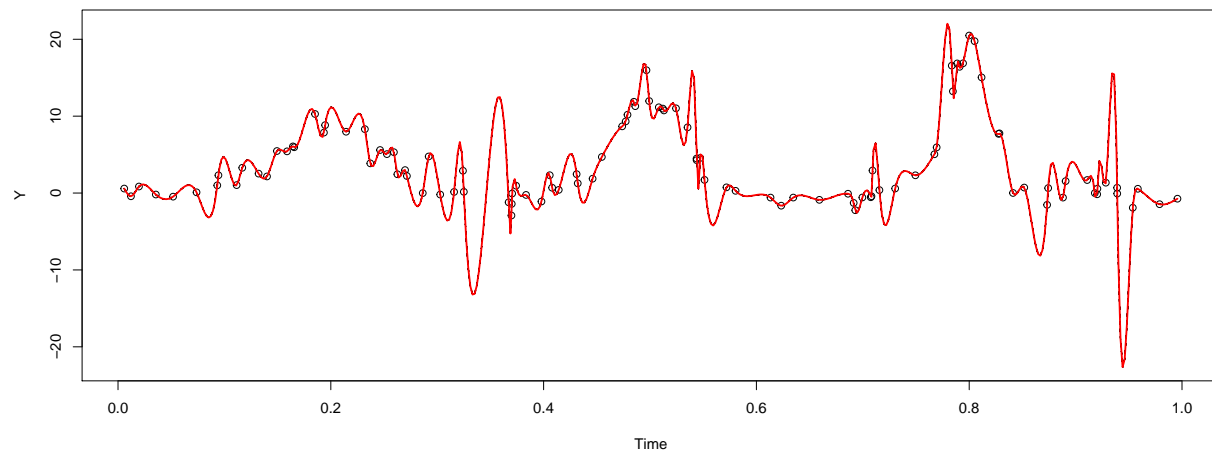
SME package

```
#install.packages("https://cran.r-project.org/src/contrib/Archive/sme/sme_1.0.2.tar.gz")
library(sme)
```

```
fit1 <- sme(data.frame(tme = fdata1$t, ind = as.factor(fdata1$id), y = fdata1$y),
  criteria = 'AIC', maxIter = 10000)

fit2 <- sme(data.frame(tme = fdata2$t, ind = as.factor(fdata2$id), y = fdata2$y),
  criteria = 'AIC', maxIter = 10000)
```

```
plot(fit1)
```



```
plot(fit2)
```

