Investigating Mean Integrated Squared Error calculation in R

Tahmidul Islam

7/1/2020

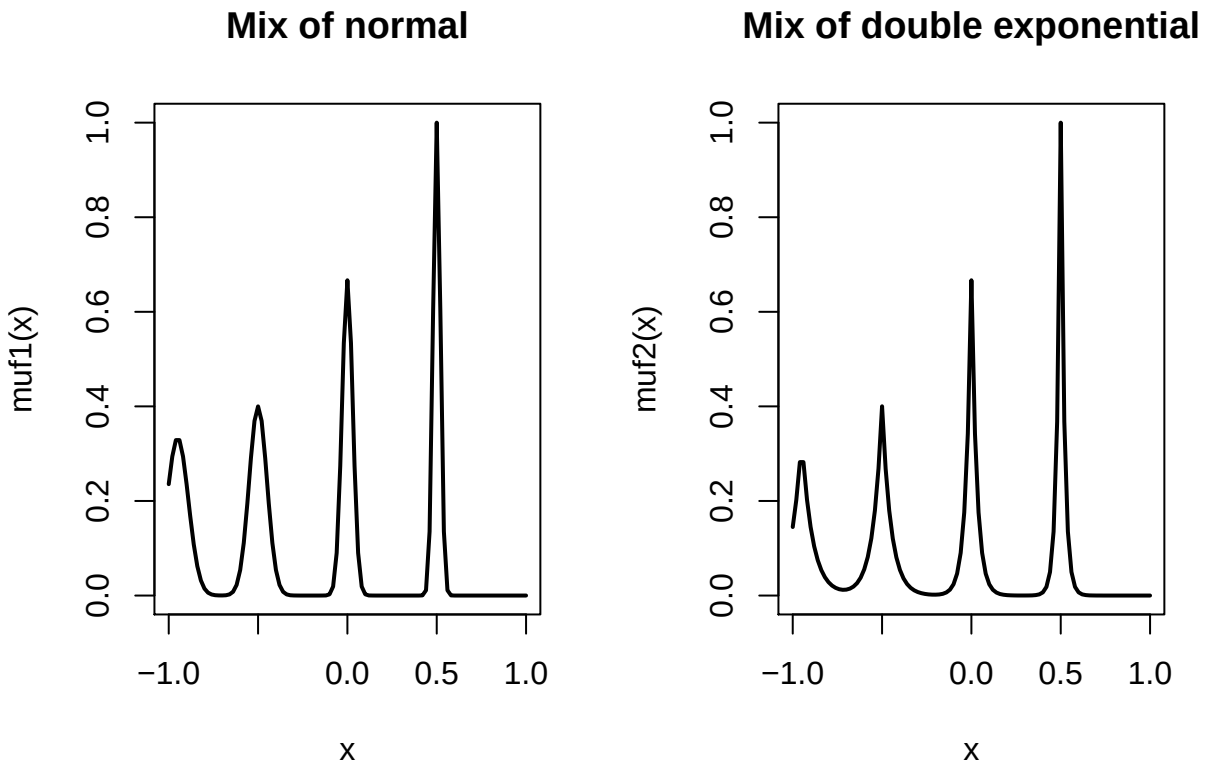
We propose two new mean functions to be used for functional data simulation. One is based on normal density functions with different location and scale parameters. The other is based on Laplace (double exponential) distribution with similar manipulation. These two mean functions contain spikes of several degrees (depending on the scale parameter). The Laplace distribution outputs non-differentiable regions at the spikes. Proposed new mean function:

$$\mu_1(t) = .25\ N(-.95, .06) + .25\ N(-.5, .05) + .25\ N(0, .03) + .25\ N(.5, .02)$$

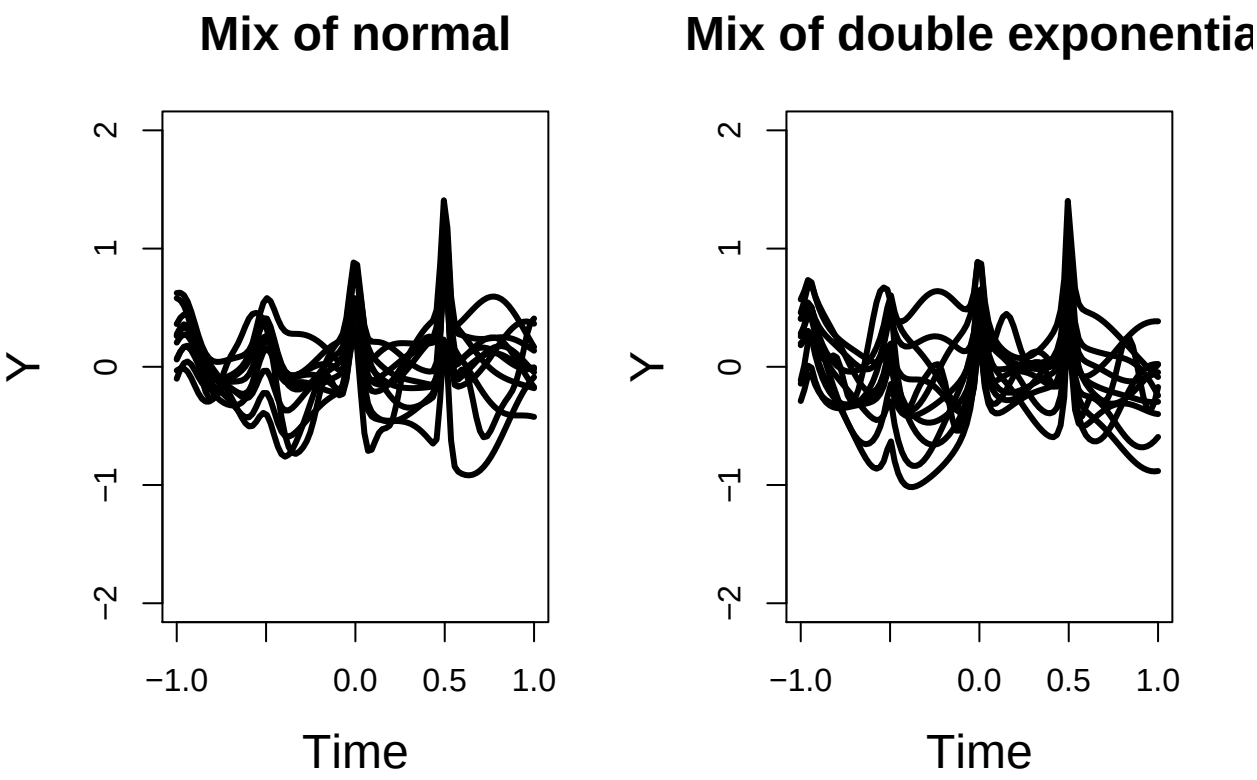
$$\mu_2(t) = .25\ dexp(-.95, .06) + .25\ dexp(-.5, .05) + .25\ dexp(0, .03) + .25\ dexp(.5, .02)$$

Where $N(a, b)$ and $dexp(a, b)$ are the density function of a normal distribution and a Laplace distribution respectively with mean a and standard deviation b.

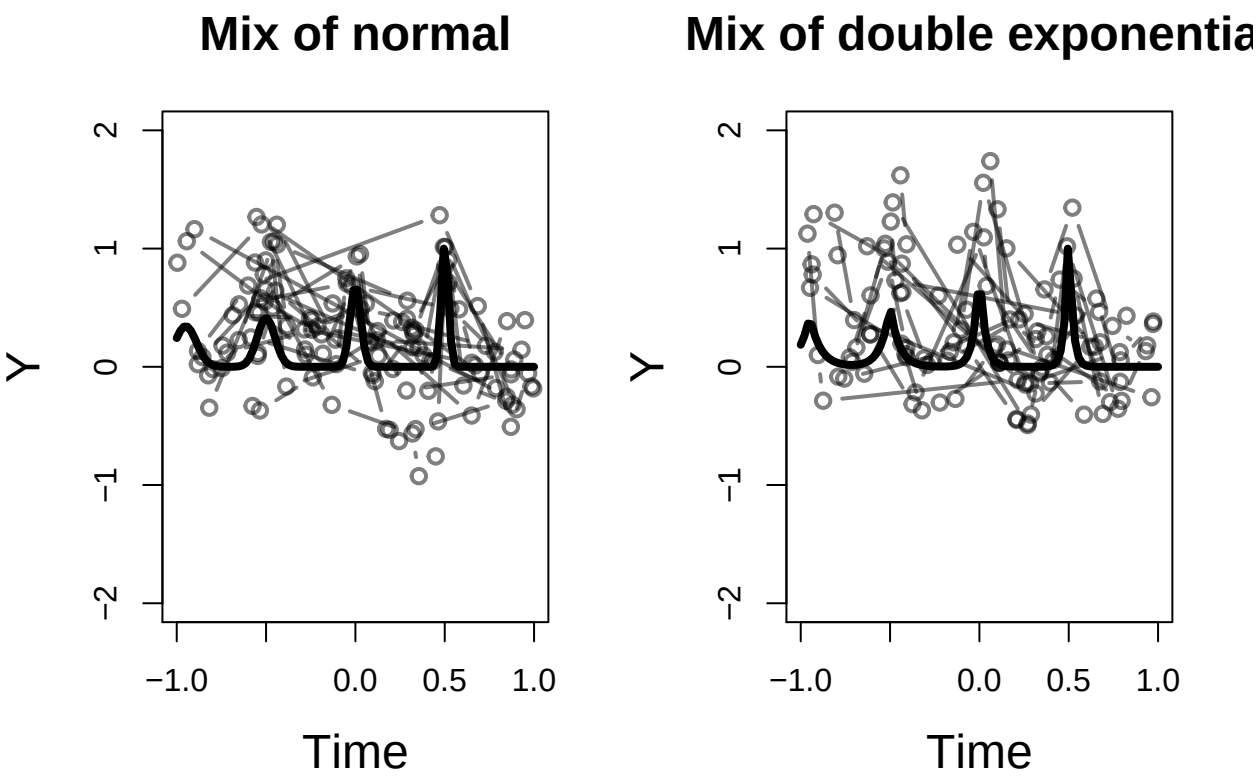
Let us plot the mean funtions.



Now simulate some random functions using the mean functions as the prior mean in the Gaussian process with Gaussian covariance function.



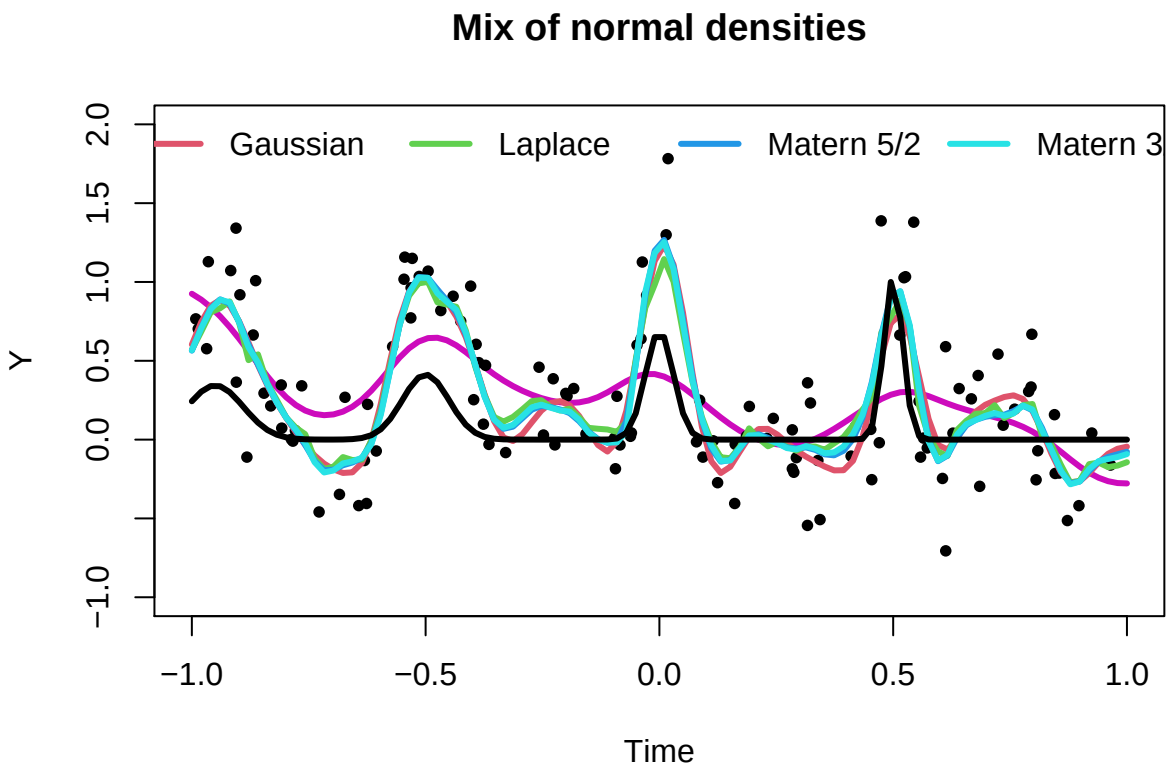
Now we generate sparse functional data which are actually observed in practice. To introduce sparsity, first we sample any integer n_t between 2 to 10. This is the number of time points each function will be observed on. Further we sample n_t time points from a uniform (-1, 1).



We will fit our GP based model to the generated data and compare with the fit with the PACE (Yao, Mueller, and Wang (2005)) method based on Functional Principal Component Analysis (FPCA), a key technique for functional data analysis, for sparsely or densely sampled random trajectories and time courses, via the Principal Analysis by Conditional Estimation (PACE) algorithm.

Mixed of normal density mean function

We can inspect the fit of the mean function estimation by GP based method and compare with PACE. For GP method, we have used four different covariance kernels: Gaussian (RBF), Laplace, Matern 5/2 and Matern 3/2.



Error in mean function estimation

Mean Squared Error (MSE)

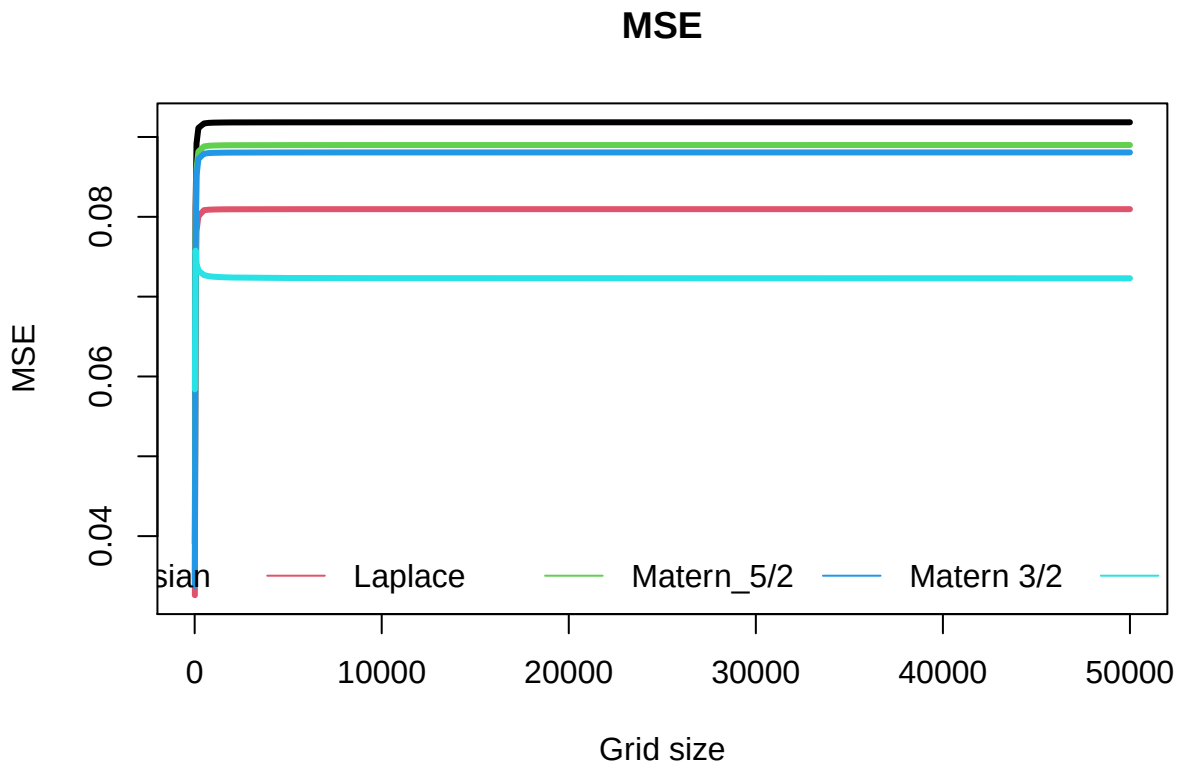
We want to measure the goodness of fit of our GP based method and compare the performance with PACE. The basic error metric we can use for mean function estimation is the Mean Squared Error (MSE). Since we know the true mean function, we can compute the following:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{N}\Big(\hat{\mu}(t_i) - \mu(t_i)\Big)^2.$$

Here the time points t_i's form a fixed grid of points over the support of the function. Here we can choose $N = 100, 500, 1000, \ldots$ time points from the interval $(-1, 1)$ and compute the MSE. We want to investigate the effect of the choice of N.

Mix of normal densities

n	Gaussian	Laplace	Matern_5/2	Matern 3/2	PACE
10	0.0390399	0.0326009	0.0346213	0.0337489	0.0583866
50	0.0810898	0.0696550	0.0767159	0.0754654	0.0757671
100	0.0892001	0.0781923	0.0860114	0.0851498	0.0741517
200	0.0911269	0.0800937	0.0881719	0.0872720	0.0732679
500	0.0916986	0.0808330	0.0888221	0.0879065	0.0726948
750	0.0917670	0.0808898	0.0889017	0.0879834	0.0725626
1000	0.0917932	0.0809147	0.0889328	0.0880132	0.0724959
1500	0.0918141	0.0809363	0.0889580	0.0880372	0.0724288
2000	0.0918226	0.0809454	0.0889684	0.0880471	0.0723951
5000	0.0918346	0.0809536	0.0889835	0.0880612	0.0723342
10000	0.0918376	0.0809557	0.0889875	0.0880649	0.0723138
50000	0.0918397	0.0809571	0.0889903	0.0880675	0.0722975



Mean Integrated Squared Error (MISE)

The mean integrated squared error (MISE) is defined by

$$E||\hat{f} - f||^2 = E \int \left(\hat{f}(t) - f(t)\right)^2 dt.$$

Where f is the true function. We first compute the integrated squared error.

```r
# Compute error
rint <- integrate(residfunc, lower = -1 , upper = 1, muf = muf1, est = gpsmooth,
          est_arg = fet.muf1.rbf, subdivisions = 10000)$value
rint <- cbind(rint,
integrate(residfunc, lower = -1 , upper = 1, muf = muf1, est = gpsmooth,
          est_arg = fet.muf1.lap, subdivisions = 10000)$value,

integrate(residfunc, lower = -1 , upper = 1, muf = muf1, est = gpsmooth,
          est_arg = fet.muf1.m52, subdivisions = 10000)$value,

integrate(residfunc, lower = -1 , upper = 1, muf = muf1, est = gpsmooth,
          est_arg = fet.muf1.m32, subdivisions = 10000)$value)

integrate(residfunc, lower = -1 , upper = 1, muf = muf1, est = fpcamu,
          est_arg = fpca1, subdivisions = 10000)$value
```

```
## Error in integrate(residfunc, lower = -1, upper = 1, muf = muf1, est = fpcamu, : maximum number of su
```

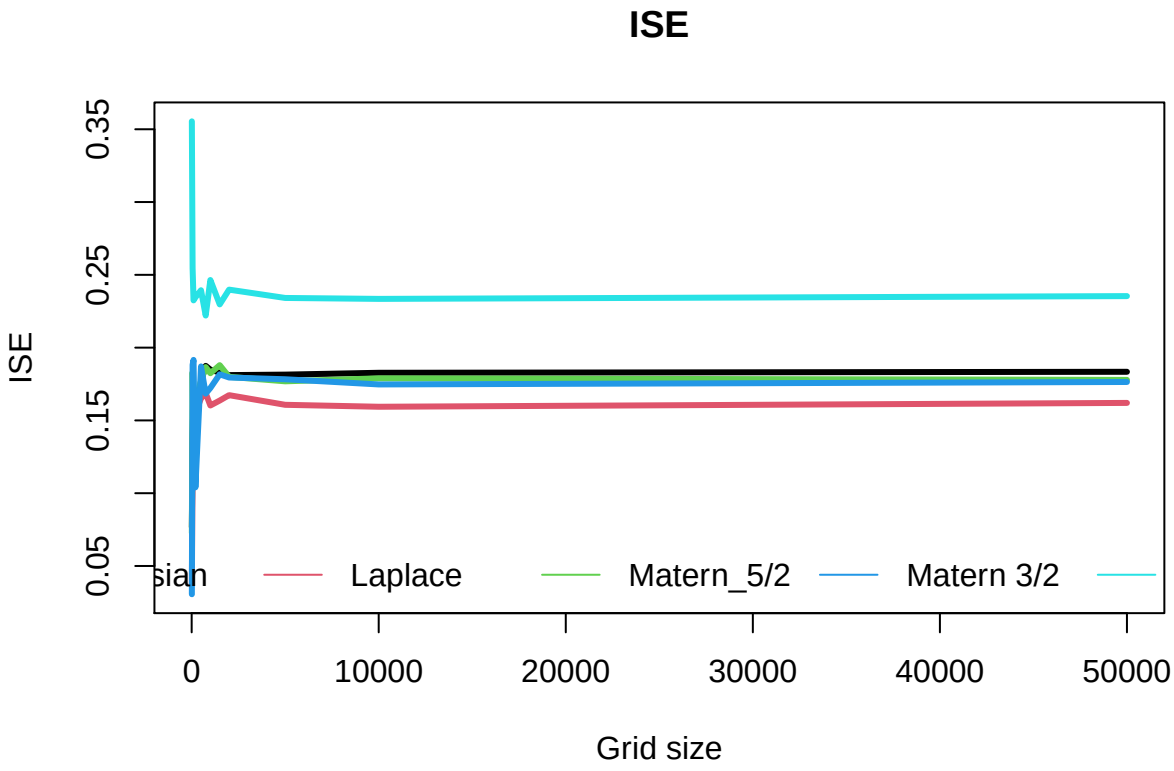
Since R's base code for integration using adaptive quadrature sometimes results in error, we use a package called 'cubature'. This package uses adative and monte-carlo integration. We show results for the mix of normal density mean function.

Method	Gaussian	Laplace	Matern_5/2	Matern 3/2	PACE
Base R	1.0622	0.1065	0.1342	0.1137	NA
Cubature	2.2536	1.2801	1.4564	1.4376	1.2298

Clearly, there are discrepencies in the integration result. We implement a simple Monte Carolo integration using uniform distribution. We generate N time points from uniform(-1,1) and approximate the integral.

$$\int_a^b f(t)dt = (b-a)\frac{1}{N}\sum f(t_i).$$

n	Gaussian	Laplace	Matern_5/2	Matern 3/2	PACE
10	0.0766391	0.0372530	0.0726518	0.0305921	0.3554319
50	0.1385882	0.1194255	0.1830429	0.1881872	0.2547085
100	0.1685069	0.1310707	0.1332573	0.1915726	0.2324583
200	0.1374494	0.1551261	0.1630506	0.1040107	0.2345263
500	0.1770121	0.1651761	0.1770614	0.1870797	0.2394253
750	0.1874870	0.1687485	0.1866356	0.1683850	0.2220219
1000	0.1848743	0.1602485	0.1824088	0.1722619	0.2464493
1500	0.1828228	0.1636531	0.1879577	0.1816469	0.2296869
2000	0.1811220	0.1673438	0.1799314	0.1794839	0.2398319
5000	0.1814504	0.1607137	0.1768480	0.1782656	0.2341142
10000	0.1827255	0.1593848	0.1789074	0.1747164	0.2334841
50000	0.1833986	0.1620169	0.1778345	0.1763995	0.2353784


ISE
0.35
0.25
0.15
0.05
ISE
sian
Laplace
Matern_5/2
Matern 3/2
0
10000
20000
30000
40000
50000
Grid size