

## ST 758 Project 2

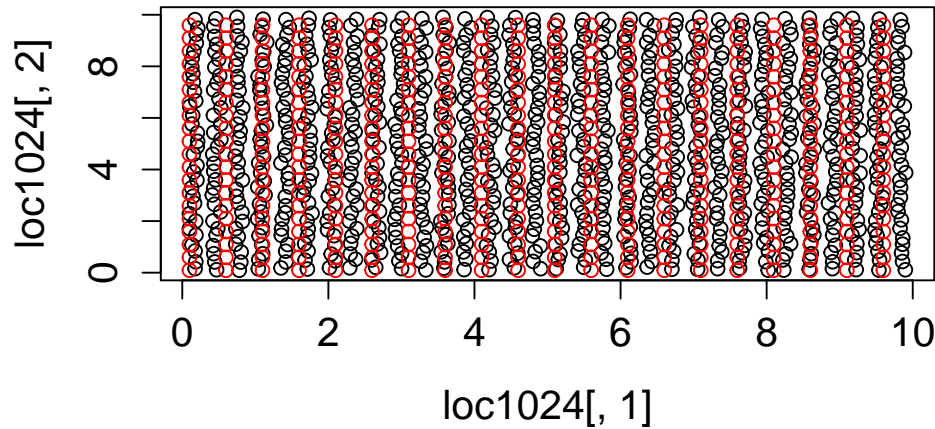
Jessica Miller

Nov. 10, 2016

For this assignment, I focused on the two issues that would slow down the computation of the Gaussian log-likelihood: the locations and the matrix inverse. To reduce the number of locations, I created grids to approximate the given locations. To calculate the matrix inverse, I investigated two different methods in R.

The more locations included, the larger the distance matrix and the longer it takes to compute the log-likelihood. To explore this, I identified a group of knots. Using knots is fairly common practice in spatial statistics to approximate a group of locations, even those that are clustered. The locations for this assignment are very grid-like and so using another grid of knots is not an extreme idea. For the group of 1024 locations, I used a grid evenly spaced by 0.5 units. For my initial calculations, I used a grid spaced as such for each group of locations, no matter the density of the locations. See Figure 1 for the grid using for the 1024 locations.

Figure 1: The 1024 locations, plotted in black, and the grid used to approximate them, plotted in red.



After I created my grids to approximate the locations, I continued to the calculation of the log-likelihood. I calculated two covariance matrices for each grid, following the equations from the project guidelines. From there, I focused on speed: how can I calculate the inverse of the covariance matrix, and thus the log-likelihood, the quickest? I investigated the timings using the `solve()` and `chol2inv()` functions in R. I generated 400 different multivariate Normal responses with mean 0 and sigma corresponding to the appropriate locations' covariance matrix. Using these, I calculated each log-likelihood and averaged their system speeds. In Table 1, we see that the `chol2inv()` function calculates the inverse more quickly.

Table 1: System timings (seconds) for each grid, covariance function, and matrix inversion method.

Cov. Fun.		1024	2025	4096
Matern	solve	0.077	0.087	0.088
	chol2inv	0.036	0.040	0.040
Exp.	solve	0.092	0.079	0.089
	chol2inv	0.042	0.036	0.040

After this initial analysis of speed, I moved on to improving accuracy. I increased the density of the grids for the 2025 and 4096 locations. The 2025 locations were increased to a spacing of .375 units and the 4096 locations were increased to a spacing of .25 units to better approximate the original sites. The speed decreased significantly for the exponential covariance function, as seen in Table 2.

Table 2: System timings (seconds) for the 2025 and 4096 grids, and each covariance function and matrix inversion method.

Cov. Fun.		2025	4096
Matern	solve	0.079	0.089
	chol2inv	0.035	0.041
Exp.	solve	0.605	4.660
	chol2inv	0.236	2.065

It is interesting that the Matern covariance function did not increase in timing as much as the exponential covariance function did. The `chol2inv()` R function continued to calculate the inverse more quickly than the `solve()` function.

To optimize both speed and accuracy of the calculation of the Gaussian log-likelihood, I would use the higher grid densities when using the Matern covariance function and the lower grid densities when using the exponential covariance function. This promises the accuracy of the higher densities but the speed of the lower densities when using either covariance function.