

Homework 2
Stat 597a: Spatial Models
Due Tuesday, October 10 in class

Hand in a paper copy of your solutions. Turn in a written answer to Problem 1 and then the R code and plots you produce for the rest of the problems. Please use comments to show where each problem number begins in your code.

In this assignment we will carry out the analysis of mourning dove counts from section 7.4.2 of the Handbook of Spatial Statistics (you can download it from Canvas), using the `geoRglm` package in R.

1. First, read over this section. There is one key change we will make compared to Wikle's analysis, which is that I have first transformed the coordinates from longitude and latitude to UTM coordinates. This is because the `geoRglm` package only uses Euclidean distances. Using it with longitude and latitude will not give accurate distances between the points. Another way we could do it, if we were coding this from scratch, would be to keep longitude and latitude, but work with great circle distances when calculating the distance matrix.

Given this new coordinate system, the prior for the range parameter ϕ that we'll use is a discrete uniform distribution from 500 to 300,000, in increments of 500. The reasons for the discrete prior are again due to the constraints of the `geoRglm` package.

With this change, write down the three layers of the hierarchical model as defined on slide 8 of Lecture 8. That is, what distributions make up the data model, process model, and prior model?

2. Transform the original observations using $Z_i = \log(Y_i)$ and use classical geo-statistical techniques to get preliminary estimates of σ^2 and ϕ by treating the Z_i as normally distributed given η (i.e. fit a model with a nugget, and extract just the estimates for σ^2 and ϕ).
3. Use `model.glm.control`, `prior.glm.control`, `mcmc.control`, and `pois.krige.bayes` to run an initial MCMC chain, fixing ϕ at your estimate from (1). The goal here is to experiment with changing `S.scale` to achieve an acceptance rate of about 60% for the process samples, which is optimal for the algorithm `pois.krige.bayes` is using to sample the vector of process values.

Hint: Here is some code to get you started.

```

sim.model <- model.glm.control(cov.model = ??, kappa = ??)
sim.prior <- prior.glm.control(phi.prior = ??, phi = ??)
sim.mcmc <- mcmc.control(S.scale = ??, Htrunc = "default",
                        S.start = "random",
                        burn.in = 0, thin = 10, n.iter = 100000)
sim.posterior <- pois.krige.bayes(??,
                                model=sim.model, prior=sim.prior,
                                mcmc.input=sim.mcmc)

```

The choice of `n.iter = 100000` and `thin = 10` is just to keep things manageable at this stage. We will eventually run a much longer chain. I suggest you always keep `burn.in = 0` and then discard the initial samples yourself after seeing the results.

4. Now duplicate and modify the code from (3) to include sampling ϕ . Experiment with changing `S.scale` and `phi.scale` to get acceptance rates of about 60% and 25%, respectively. (Note: I found the acceptance rates fluctuated over the course of my chain. Just aim to get in the right ballpark.) I suggest you also take `thin = 100` and `n.iter = 100000` here.

Make trace plots and ACF plots of the parameters σ^2 , ϕ , and β . You will likely see a LOT of autocorrelation.

5. There is not much beyond changing `S.scale` and `phi.scale` that we can do to reduce the autocorrelation. So we will run a long chain and subsample it. Run the same code as in part (4), increasing to `thin = 1e4` and `n.iter = 1e7`.

Now do the following with your sampled parameters.

- Make trace plots and ACF plots. Choose a burn-in to discard and make them again.
 - Calculate the effective sample size you have for each parameter. If they are not at least 100 for each parameter, go back and run a longer chain.
 - Plot the marginal posterior distributions for each parameter using histograms and/or kernel density estimators.
6. Do one final run of the chain, the same as in (5), but this time specifying the burn-in value you chose in (5) and also modifying the code to include prediction at the locations in `dove.grid`. Create two color or grayscale plots, showing the posterior mean and standard deviation for the underlying mean surface ($\exp \eta$) at each location in `dove.grid`.