Supporting Documentation for Go Code Business Challenge

Erin Hodgess

November 18, 2020

We are using techniques from modern tools such as geostatistics for our submission. are very informative and are effective for prediction purposes. We are using the statistics language `R`, Fortran, OpenACC, and the Message Passing Interface (MPI) to generate our analysis.

For our application, we consider the time series from the Go Code Colorado website:
(`https://data.colorado.gov/Business/Retail-Reports-by-City-in-Colorado/2yhn-3dbj`). We selected the locations of Denver, Boulder, Fort Collins, and Colorado Springs. We have historical monthly data from each location from January 2016 - July 2020.

We make some assumptions for the models. We assume first order stationarity; that is:

$$E[z_t] = \mu$$

for all values of t. We also assume second order stationarity:

$$Cov(z_{t1}, z_{t2}) = Cov(t_2 - t_1, 0)$$

for all $t \in \mathcal{N}$.

For the geostatistics portion, the process is a bit more complicated. We will be using ordinary kriging, which is a type of weight linear regression. We start with a spatio-temporal vector:

$$\mathbf{z} = z(s_1, t_1), \ldots, z(s_m, t_n)$$

where the s_i and t_j are the locations and times of the response variable. We use assumptions for the first and second order stationarity.

$$\begin{aligned} E[\mathbf{z}] &= \mu \text{ first order,} \\ \mathbf{C_{st}} &= Cov(z(s,t), z(s+h, t+u)), \\ &= Cov(z(0,0), z(h,u)) \text{ second order} \end{aligned}$$

for all $h, u \in \mathcal{N}$.

We use these assumptions to produce our covariance matrix, our overall mean, and our predictions at various times and locationa.

We utilized high performance tools on a Windows laptop to speed up our process. `R` does have a function for spatio-temporal kriging, but it can be slow. We constructed Fortran programs using OpenACC to both speed up the process and enlarge the input data size.

Finally, we took our output grid and produced a KML file with the times, locations, and our predicted values. This is readable by Google Earth. By selecting a pin, we can see the location and the predicted value.

We believe that this model building technique can be very useful. Having a dynamic map gives the viewer a better sense of how predictions can change over time.

‘