USING CUDA AS A SPATIO-TEMPORAL KRIGING TOOL

Erin Hodgess

Western Governors University, Salt Lake City, UT, USA

ABSTRACT

We are enhancing our high performance tool research to include a mix of MPI (Message Passing Interface) and CUDA (Compute Unified Device Architectecture) programming, in order to speed up the processing time. Using these new processes, we have seen tremendous speed-up in spatio-temporal kriging. The NVIDIA graphics cards provide the vehicle for the latest version of CUDA. We can use a Windows laptop, with the Windows Subsystem for Linux (WSL 2), utilizing the Ubuntu 22.04 software. We see that not only does the processing time is significantly reduced, but larger data sets are able to be accommodated. Adding the CUDA subroutines greatly enhances spatio-temporal kriging model building.

KEYWORDS

spatio-temporal kriging, R, CUDA

1. INTRODUCTION

We have seen that using high performance tools can greatly enhance spatio-temporal and spatial data analysis. We have worked with these tools using supercomputers [1] and Windows laptops with NVIDIA graphics cards [2]. We previously used a combination of MPI and OpenACC to achieve improved results. Our previous work [2] compared our method against a MacBook Air, a Windows laptop, and a Windows laptop with the NVIDIA graphics cards. We saw significant speedup there as well. We also put together tools using OpenMP for the Raspberry Pi [3]. These applications would be useful for researchers with a limited computing budget. Our latest iteration now applies CUDA into existing processes in order to both speed up processing time, as well as allowing for large data set sizes. As we will show, this can be effectively completed on a Windows laptop with an NVIDIA graphics card. We also utilize WSL2 to aid with development. There is an existing function, krigeST, which is currently in place in the R statistical language [4], in package space time [5], [6]. We build CUDA subroutines into the solutions of the spatio-temporal kriging equation, and these produce excellent results.

1. DEFINITION OF THE PROBLEM

Suppose we have a spatio-temporal vector:

$$z = z(s_{1,}t_1), z(s_2, t_1), \dots, z(s_m, t_n),$$

where $z(s_i, t_j)$ are the locations and times of the response variable.

We make the assumption of first-order stationarity:

DOI:10.5121/ijitmc.2023.12201

$$E[z] = \mu$$

We can also make the assumption of second order stationarity for the covariance of the spatio-temporal model:

$$C_{st} = Cov\big(z(s,t), z(w,v)\big)$$

The spatial distance is h and the temporal distance is u, such that:

$$|s - w| = h$$
$$|t - v| = u$$

for all values of s, w, t, v.

We consider various sample and model estimates for our variogram. We have settled on 3 particular models: separable, product sum, and sum-metric. The separable model breaks space and time components independently. This is a easy solution, but not always very accurate, in terms of the real world. Next, we discuss the product sum model. This model considers an additive combination of spatial and temporal components. Finally, we produce the sum-metric model. The model considers spatial, temporal, and a term for anistrophy. This is the most appropriate model for real world data.

We can produce an overall sample mean estimate:

$$\hat{\mu} = (1' C_{sm}^{-1}\ 1)^{-1} \cdot (1' C_{sm}^{-1}\ z),$$

where $C_{sm}^{-1}$is

$$C_{sm} = C_s(h) + C_t(u) + C_{joint}\left(\sqrt{(h^2 + \kappa \cdot u^2}\right)$$

In the previous equation, the κ is the anisotropy value; that is, characteristics that show considerable variation based on direction.

The one step kriging forecast is:

$$\hat{z} = \hat{\mu} + c_0 C_{sm}^{-1}(z - \mu 1)$$

in which c_0 is the covariance to the new value. This can be easily extended to multiple locations and times.

2. METHODOLOGY

We use the R [4] program as our basic tool. The main spatial and temporal functions are found in [5 - 8]. We also use the pbdDMAT package [9] for large data sets.

We then build CUDA Fortran subroutines to tie into R. We use these subroutines to produce the individual covariance matrices and construct their total. Finally, we use CUDA to invert the total sum-metric covariance matrix.

3. Discussion of the Data Analysis

We use the Irish Wind data set, as described in [10]. There are daily average wind speed values at 12 locations from 1961 – 1978.

We start with 3 months of daily data, and produce kriging estimates. We then increase the number of daily values by 1 year at all 12 locations.

We use a regular Windows laptop, and run the krigeST function [5], on our data sets. We then run the new procedure on a Windows laptop with WSL2, which has an NVIDIA graphics card. This has the code which includes the CUDA Fortran subroutines, which are called from R [4].

Table 1: Results in Minutes

	Windows	CUDA
3 mo	0.713	0.583
1 yr	3.493	2.111
2 yr	14.577	4.933
3 yr		8.533
4 yr		9.045

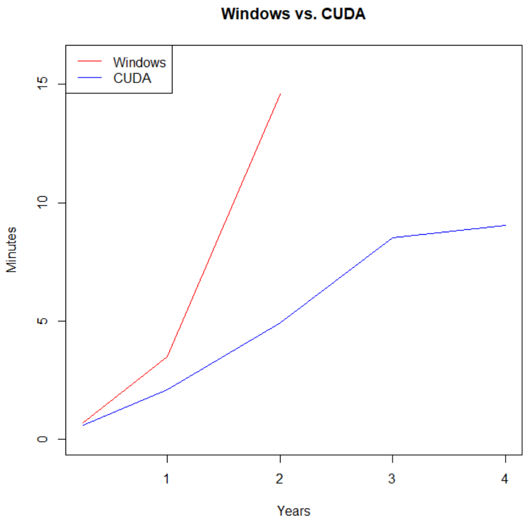

Figure 1: Results in Minutes

4. Conclusion

We believe that CUDA can be an extremely effective tool, when added into R processing. We see that the times are dramatically better, and the amount of data that can be processed is more extensive.

As the next step in our research, we are constructing using just R [4] and CUDA, bypassing the MPI stage altogether.

High performance tools are not merely items for supercomputers. For laptops with NVIDIA cards, we can exploit these for better quality research.

REFERENCES

[1] Hodgess, E.M., and Mhoon, K. (2018). "Advances in Ordinary Kriging Using the Stampede and Bridges Supercomputers", Journal of Geological Research and Engineering, 6, pp. 14-18.

[2] Hodgess, E.M., and Mhoon, K (2019). "Ordinary Kriging for Spatio-Temporal Data Sets Using a Windows Laptop", Proceedings of the Geociencias Meeting, Havana, Cuba, April 2019.

[3] Hodgess, E. (2022). "High Performance Computing on the Raspberry Pi", International Journal of Computational Science and Information Technology, Volume 10, Number 1, February 2022.

[4] R Foundation for Statistical Computing. R: A Language and Environment for Statistical Computing, [ref. April 8, 2024]. URL: www.r-project.org

[5] Pebesma, E. (2012). "spacetime: Spatio-Temporal Data in R", Journal of Statistical Software, 51(7), 1 – 30.

[6] Bivand, R., Pebesma, E., and Gomez-Rubio, V. (2012). Applied Spatial Statistics with R, Second Edition, Springer Publishing, New York.

[7] Graler, B., Pebesma, E., and Heuvelink, G.B.M. (2016). "Spatio-Temporal Interpolation Using gstat", R Journal, Vol 8, Number 1, pp. 204-218.

[8] Pebesma, E. (2004). "Multivariable Geostatistics in S: the gstat package", Computers and Geosciences, 30, 683-691.

[9] Schmidt, D., Chen, W.C., Ostrouchov, G., and Patel, P. (2012). "pbdDMAT: Distributed Matrix Algebra Computation", R package.

[10] Haslett, J., and Raftery, A.E. (1989). "Space-time Modelling with Long-memory Dependence: Assessing Ireland's Wind Power Resource" (with Discussion). Applied Statistics, 38, 1-50.