A Parallel Spatiotemporal Kriging Algorithm

Hongda Hu[1]

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University
129 Luoyu Road, Wuhan, Hubei, China 430079
Email: dadagiser@163com

1. Introduction

In spatial statistics, geostatistics is quite strong for modeling spatial or spatiotemporal processes. Apart from regionalized variable theories, a critical part is how to design a valid variogram for modeling spatiotemporal autocorrelation or variability. To this end, a product-sum model of combining spatial and temporal variograms into a spatiotemporal variogram has been proposed early (De Cesare et al., 2001).

The spatiotemporal kriging interpolation based on the product-sum model is useful for statistical predictions. However, the procedure of spatiotemporal kriging is computation-intensive due to heavy load of calculating spatiotemporal distances and variograms. From the previous work by Kerry and Hawick (1998), it is known that computing parallelism is feasibly applied to kriging. Recently, the parallel computing approach to fast geostatistical areal interpolation has also been proposed by Guan et al. (2011). In this paper we innovatively present a parallel spatiotemporal kriging algorithm implemented with R snowfall package, and demonstrate its application to spatiotemporal interpolation of air temperature in East China.

2. Spatiotemporal Kriging

2.1 Product-sum Variogram Model

The spatiotemporal random field Z(S,T) is assumed to be intrinsic stationary, if the variance of increments between two spatiotemporal random variables $\mathrm{Var}(Z(s+h_s,t+h_t)-Z(s,t))$ is a function of spatial distance h_s and temporal distance h_t. Then the function $\gamma_{st}(h_s, h_t)$ called spatiotemporal variogram is shown as equation 1.

$$\gamma_{st}(h_s,h_t)=\frac{1}{2}\mathrm{Var}\big(Z(s+h_s,t+h_t)-Z(s,t)\big)=\frac{1}{2}E(Z(s+h_s,t+h_t)-Z(s,t))^2 \quad (1)$$

The product-sum of purely spatial variogram and purely temporal variogram, reflecting the mixed effects of purely spatial variability and purely temporal variability, can generate a valid spatiotemporal variogram. The product-sum spatiotemporal variogram function $\gamma_{st}(h_s, h_t)$ is defined as equation 2.

$$\gamma_{st}(h_s,h_t)=(k_1C_t(0)+k_2)\gamma_s(h_s)+(k_1C_s(0)+k_3)\gamma_t(h_t)-k_1\gamma_s(h_s)\gamma_t(h_t) \quad (2)$$

where γ_s is the spatial variogram, γ_t is the temporal variogram, $C_s(0)$ is the spatial sill, $C_t(0)$ is the temporal sill, and k_1,k_2,k_3 can be calculated by equation 3.

$$\begin{cases} k_1=[C_s(0)+C_t(0)-C_{st}(0)]/C_s(0)*C_t(0) \\ k_2=1-k_1*C_t(0) \\ k_3=1-k_1*C_s(0) \end{cases} \quad (3)$$

where $C_{st}(0)$ is the spatiotemporal sill, and it can be practically choosen from the larger one in $C_s(0)$ and $C_t(0)$.

2.2 Parallel Spatiotemporal Kriging Algorithm

In algorithm, the steps of spatiotemporal kriging are similar to the ordinary spatial kriging, that is,

(1) calculating the empirical spatial and temporal variograms at different lags to fit the theoretical variogram models(eg., the spherical model);
(2) applying the fitted spatial and temporal variogram functions to build product-sum spatiotemporal variogram model;
(3) choosing the samples within the spatiotemporal range at each unobserved location ;
(4) computing the variogram coefficient matrix of these samples and the variogram vector between each unobserved location and its sample sites;
(5) computing the weight vector and the estimation at each unobserved location.

Through analysis of computational complexity of each step, it is found that the last three steps have consumed the largest portion of computing time. To deal with this situation, our study is intended to explore parallel computation techniques applied in these time-consuming steps. Obviously, the data interpolation of each unobserved location is an independent task which can be regarded as a parallel unit. The essential part of this algorithm is data parallelism. By data parallelism, it means that coordinates of each unobserved location, coordinates of its nearby sample sites and the sample attribute values are assigned to a free node. Then sample spatial distances and temporal distances are calculated in each node, and the results are used to calculate the sample spatiotemporal variograms. But we realize that two adjacent points have so many identical nearby samples within the spatiotemporal range, especially when unobserved locations are dense. So the identical sample distances and variograms are calculated twice redundantly. Although this kind of computation can be parallelized, it wastes computational resource and pose a significant increase of computing time.

To solve this problem, spatiotemporal variograms between all the sample pairs in the interpolation region should be calculated and saved in the host node. The host node broadcasts all the sample spatiotemporal variograms to each node so that variograms between any sample pairs can be queried. However, this method not only reduces redundant computation, but also brings unnecessary computation. That is because variograms between some sample pairs faraway from each other may not be used for the data interpolation of any unobserved locations. Nevertheless, this unnecessary computation can be ignored as compared to the redundant computation in most cases.

We employ the dynamic load-balancing techniques to make full use of computing power and reduce computing time. Initially, it is not determined that which tasks of unobserved locations a specific node will perform. But each node will be assigned one task. Then the remaining tasks will be successively assigned to the free nodes which have finished their own tasks.

3. Experimental Analysis

This parallel algorithm implemented on the basis of R snowfall package is used for spatiotemporal interpolation of air temperature in July 2007 in East China with corresponding monthly air temperature data from January 2007 to December 2008. R

itself does not allow programs to be executed in parallel. But there exist technical solutions, e.g., snowfall for R, to deploy computational tasks over a single multicore machine and even a cluster. Snowfall supports several networking types such as socket, MPI and PVM (Knaus et al. 2009). For convenience, the socket type is used in our experiments. For experimental test, the clusters with two computers of two four-cores CPUs are configured.

The speedup results for different problem sizes are shown in fig. 1. The problem size stands for the number of unobserved locations in East China. In our case, obviously problem size 7886 provides the best speedup results, while problem size 80 does the worst. The results have explained that the parallel efficiency increases as the problem size increases. The main reason is that the major part of serial time spent on the computation of all the sample spatiotemporal variograms is almost fixed. The serial part is independent of the problem size, while the parallel time is directly proportional to the problem size.

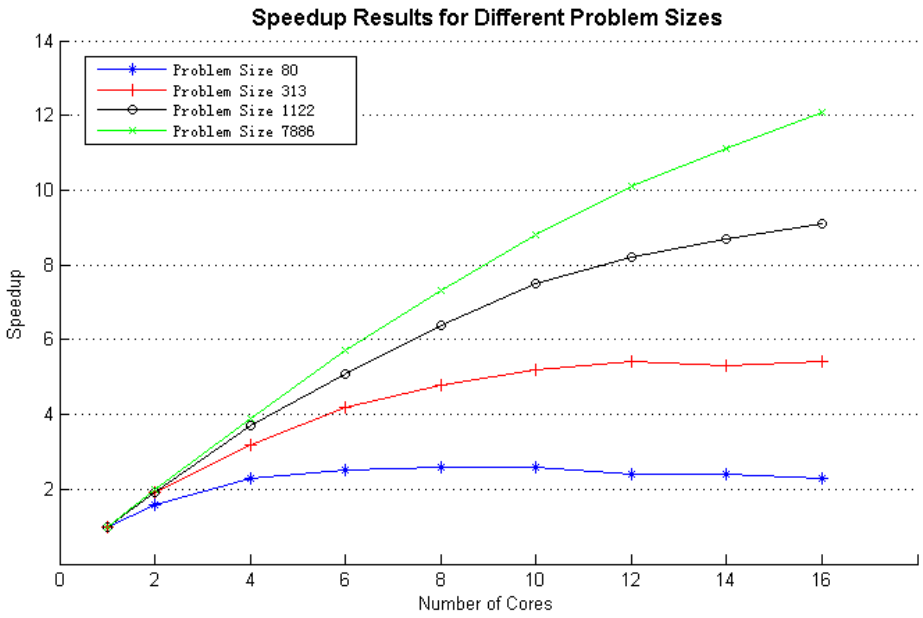

Figure 1. The speedup results for different problem sizes

4. Conclusions

The product-sum spatiotemporal variogram models space-time autocorrelaiton well, but brings intensive computation. To overcome this issue, a parallel spatiotemporal kriging algorithm is proposed in our study. The relationship between parallel efficiency and problem size is analyzed. The speedup results for different problem sizes show us that the parallel efficiency increases as the problem size increases. Thus, this parallel algorithm is appropriate for the data interpolation with dense unobserved locations.

We realize that the main problem of low parallel efficiency is that the sample spatiotemporal variograms are calculated serially in the host node and are broadcast to each node. Our ongoing work is how to parallelize this part of work in criterion of load balance. Perhaps we can divide these sample pairs into several parts and distribute these parts to each node evenly. The sample distances and variograms are calculated in each

node. Then the host node gathers these results and broadcasts all the sample variograms. Although this method increases the communication overhead, the parallel efficiency could be improved a lot. Currently we apply the network socket to implement the parallel algorithm, and the algorithm based on MPI can be further developed.

5. Acknowledgements

This research is supported by National High Technology Research and Development Program of China（No. 2011AA010502）.

6. References

Kerry K and Hawick K, 1998, Kriging interpolation on high performance computers. *Proceedings Of High Performance Computing and Networks Europe*, Amsterdam, The Netherlands, 429-438.

Guan Q F, Kyriakidis P C and Goodchild M F, 2011, A parallel computing approach to fast geostatistical areal interpolation. *International Journal of Geographical Information Science*, 25(8):1241-1267.

De Cesare L, Myers D E and Posa D, 2001, Product-sum covariance for space-time modeling: an environmental application. *Environmetrics*, 12(1): 11-23.

Knaus J, Porzelius C, Binder H, et al, 2009, Easier parallel computing in R with snowfall and sfCluster. *The R Journal*, 1(1): 54-59.