ExaGeoStatR: Harnessing HPC Capabilities for Large Scale Geospatial Modeling Using R

Sameh Abdulah, Yuxiao Li, Jian Cao, Hatem Ltaief, David E. Keyes, Marc G. Genton, and Ying Sun
Computer, Electrical and Mathematical Science and Engineering Division (CEMSE)
King Abdullah University of Science and Technology (KAUST)
Thuwal, Saudi Arabia

ABSTRACT

Large scale simulations and parallel computing techniques are becoming essential in Gaussian process calculations applications. The Gaussian log-likelihood function is used in Geospatial applications to evaluate the Gaussian model associated with a given set of measurements in existing n geographic locations. This model is represented by a set of parameters that can be used to predict missing measurements in other geographic locations. The evaluation of the Gaussian log-likelihood function requires $O(n^2)$ memory and $O(n^3)$ computation which can be considered infeasible for large datasets with existing software tools. Thus, we present ExaGeoStatR, a package for large scale geostatistics in R that computes the Gaussian log-likelihood function on shared and distributed-memory, possibly equipped with GPU, using advanced linear algebra techniques. The package provides a high-level abstraction of the underlying hardware architecture, while enhancing the R developers' productivity. The abstraction derives from the task-based programming model and the dynamic parallel runtime system (StarPU). StarPU breaks down the linear algebra operations associated with the Gaussian log-likelihood function into a set of interdependent tasks so that these tasks can easily be deployed on heterogeneous resources. The package can be used directly through the R environment without required knowledge in C, CUDA, or MPI programming languages.

Here, we demonstrate the ExaGeoStatR package by illustrating its implementation details, analyzing its performance on various parallel architectures, and assessing its accuracy using both synthetic datasets and a real sea surface temperature dataset. The performance evaluation involves spatial datasets with up to 250K observations.

ACM Reference Format:
Sameh Abdulah, Yuxiao Li, Jian Cao, Hatem Ltaief, David E. Keyes, Marc G. Genton, and Ying Sun. 2019. ExaGeoStatR: Harnessing HPC Capabilities for Large Scale Geospatial Modeling Using R. In *SC '19 Posters, November 17 - November 22, 2019, Denver, Colorado, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Gaussian processes (GPs), or Gaussian random fields (GRFs), are one of valuable tools in spatial statistics[5]. These applications include measurements regularly or irregularly located across a certain geographical region. The main motivation is to fit the GP model to spatial data for predicting missing measurements within the given geographical region. The typical inference for GRFs includes parameter estimation, stochastic simulation, and kriging (spatial prediction). Among these tasks, parameter estimation, or model fitting, is the primary and the most time-consuming one. Thus, likelihood-based inference methods for GP models are computationally expensive for large datasets.

Specifically, suppose Z is a stationary GRF with mean function $m(\cdot)$ and covariance function $C(\cdot,\cdot)$, and we observe data on a domain $D \subset \mathbb{R}^d$ at n locations, $\mathbf{s}_1, \ldots, \mathbf{s}_n$. Then, the random vector $\{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)\}^T$ follows a multivariate Gaussian distribution:

$$\forall\{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset D, \quad \{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)\}^{\mathrm{T}} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma), \qquad (1)$$

where $\boldsymbol{\mu} = \{m(\mathbf{s}_1), \ldots, m(\mathbf{s}_n)\}^{\mathrm{T}}$ and Σ are the mean vector and the covariance matrix of the n-dimensional multivariate normal distribution. Given $\boldsymbol{\mu}$ and Σ, the likelihood of observing $\mathbf{z} = \{z(\mathbf{s}_1), \ldots, z(\mathbf{s}_n)\}^{\mathrm{T}}$ at the n locations is

$$L(\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu})\right\}. \qquad (2)$$

The (i,j)-th element of Σ is $\Sigma_{ij} = C(\mathbf{s}_i, \mathbf{s}_j)$, where the covariance function $C(\mathbf{s}_i, \mathbf{s}_j)$ is assumed to have a parametric form with unknown vector of parameters $\boldsymbol{\theta}$. Various classes of valid covariance functions can be found in Cressie [3]. For simplicity, in this work, we assume the mean vector $\boldsymbol{\mu}$ to be zero to focus on estimating the covariance parameters. We choose the most popular isotropic Matérn covariance kernel, which is specified as,

$$\Sigma_{ij} = C(\|\mathbf{s}_i - \mathbf{s}_j\|) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\|\mathbf{s}_i - \mathbf{s}_j\|}{\beta}\right)^{\nu}\mathcal{K}_\nu\left(\frac{\|\mathbf{s}_i - \mathbf{s}_j\|}{\beta}\right), \qquad (3)$$

where $\|\mathbf{s}_i - \mathbf{s}_j\|$ is the (Euclidean or great-circle) distance between $\mathbf{s}_i$ and $\mathbf{s}_j$, $\Gamma(\cdot)$ is the gamma function, $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν, and σ^2, $\beta > 0$, and $\nu > 0$ are the key parameters of the covariance function controlling the variance, spatial range, and smoothness, respectively. The Matérn covariance kernel is highly flexible and includes the exponential and Gaussian kernels as special cases. The variance, spatial range, and smoothness parameters, σ^2, β, and ν determine the properties of the GRF.

The Gaussian log-likelihood function includes a Cholesky factorization operation which requires $O(n^2)$ memory and $O(n^3)$ computation. This implies that the standard methods and traditional algorithms for GRFs are computationally infeasible for large datasets. On the other hand, technological advances in sensor networks along

SC '19 , November 17 - November 22, 2019, Denver, Colorado, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

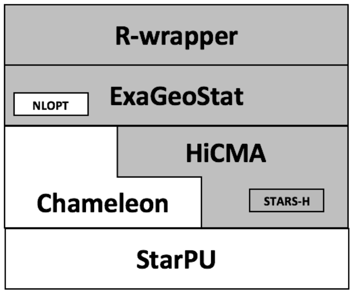

Figure 1: ExaGeoStatR Software Stack

with the investments to data monitoring, collection, resource management provide massive open-access spatial datasets [4]. Therefore, unprecedented data availability and challenging computational issues call for novel methods, algorithms, and software packages to deal with modern "Big Data" problems.

This work presents ExaGeoStatR, a high-performance package in R for geostatistical applications, that depends on an architecturally independent C-based software called ExaGeoStat [1]. ExaGeoStatR is able to fit Gaussian process models and provide spatial predictions for geostatistics application in large scale domains through both exact and approximation (i.e., Diagonal Super Tile (DST) and Tile Low-Rank (TLR)) computations. This study aims at highlighting the capabilities of the ExaGeoStatR exact computation since it can be considered as a benchmark for the performance of other computation methods. Moreover, the evaluation of the DST and the TLR approximations has been already covered in [1] and [2].

The package also includes a synthetic dataset generator for generating large spatial datasets with the exact prespecified covariance function. Such large datasets can be used to perform broader scientific experiments related to large scale computational geostatistics applications. Besides its ability to deal with different hardware architectures such as multicore systems, GPUs, and distributed systems, ExaGeoStatR utilizes the underlying hardware architectures to its full extent. Existing assessments on ExaGeoStat show the ability of the software to handle up to two million spatial locations on manycore systems [2].

2 THE EXAGEOSTATR SOFTWARE STACK

The ExaGeoStatR package presents a solution toward exascale computing for geostatistical modeling and prediction based on the Cholesky factorization to drive the maximum likelihood estimation (MLE). The package provides a set of R-wrapper functions to the existing C-based software (i.e.,ExaGeoStat). ExaGeoStat depends on a set of libraries to provide the parallel solutions of the MLE operation on different hardware architectures. Figure 1 shows the structure of ExaGeoStatR software. It has four main layers: the R-wrapper functions, ExaGeoStat, which includes the upper-level functions of the software; the linear algebra libraries, i.e., Chameleon and HiCMA, which provides solvers for the linear algebra operations; and the StarPU runtime, which translates the software for execution on the appropriate underlying hardware.

3 RESULTS

We propose a set of experiments to assess the accuracy and performance of the proposed ExaGeoStatR package. The experiments involve synthetic and real datasets assessments on different hardware architectures (i.e., shared-memory, GPUs, and distributed-memory).

We compare ExaGeoStatR package with a common existing R packages, i.e., fields [6] and geoR [7]. The results show a huge improvement in performance compared to fields [6] and geoR packages. For instance, the results show that with $22,500$ locations, ExaGeoStatR is 33 times faster than fields and 92 times faster that geoR using pure CPU shared-memory system. We also reported the performance on heterogeneous hardware, i.e., CPUs/GPUs. The results show that ExaGeoStatR have a good scalability with increasing the number of used GPUs. The results show that adding GPU accelerators improves the overall execution time compared to the pure CPU executions. Finally, using distributed systems, ExaGeoStatR shows linear scalability with a different number of nodes with the ability to handle large dataset up to 250K using 256 nodes.

The accuracy assessment is also important to validate the estimation results of ExaGeoStatR. Thus, we used a set of synthetic datasets and one real dataset,i.e., sea surface temperature, for this purpose. We used ExaGeoStatR synthetic data generator to generate the synthetic target datasets. The results show an accurate model parameter vector estimation by ExaGeoStatR with different variance, smoothness, and range values of the given spatial data. We also used sea surface data from the Indian Ocean to assess the prediction quality of the estimated parameters. The results show high prediction accuracy in four different days of temperature data.

4 CONCLUSIONS AND FUTURE WORK

We present ExaGeoStatR package for large scale Geostatistics applications in R. The package provides parallel computations for the Gaussian maximum likelihood function using shared-memory, GPUs, and distributed-memory systems. Thus, large scale Gaussian calculations in R become possible with ExaGeoStatR by mitigating its memory space and computing restrictions.

We analyzed and assessed the performance of exact computation variant of ExaGeoStatR against existing well-known R packages, i.e., geoR and fields. The evaluation shows a large difference in ExaGeoStatR performance compared to the other two packages. The accuracy evaluation also shows that ExaGeoStatR performs very well in both synthetic and real datasets. We focused on exact computation to show the advantage of using ExaGeoStatR over the aforementioned R packages and its unique ability to run on different existing hardware architectures. However, ExaGeoStatR is not limited by exact computation, it also includes two approximation computation variants, Diagonal Super Tile (DST) and Tile Low-Rank (TLR) methods [1, 2]. Moreover, the package is designed to be extensible to other approximation methods of Gaussian process calculations in the future.

Our goal from the beginning is to abstract the parallel execution functions to the R developer. The developer needs only to specify some parameters to define the hardware architecture and the package will take care of optimizing the execution on the target hardware with the aid of the underlying runtime software. The package can be used directly through the R environment without required knowledge in C, CUDA, or MPI programming languages. By this way, we increase the portability of our package and make it more suitable for R community developers.

As our future work, ExaGeoStatR will provide the necessary built-in functions to support the aforementioned extensions for more complex applications related to Gaussian process calculations.

REFERENCES

[1] Sameh Abdulah, Hatem Ltaief, Ying Sun, Marc G Genton, and David E Keyes. 2018. ExaGeoStat: A High Performance Unified Software for Geostatistics on Manycore Systems. *IEEE Transactions on Parallel and Distributed Systems* 29, 12 (2018), 2771–2784.

[2] Sameh Abdulah, Hatem Ltaief, Ying Sun, Marc G Genton, and David E Keyes. 2018. Parallel Approximation of the Maximum Likelihood Estimation for the Prediction of Large-Scale Geostatistics Simulations. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 98–108.

[3] Noel Cressie. 2015. *Statistics for Spatial Data*. John Wiley & Sons.

[4] Andrew O. Finley, Sudipto Banerjee, and Alan E. Gelfand. 2015. spBayes for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models. *Journal of Statistical Software* 63, 13 (2015), 1–28. http://www.jstatsoft.org/v63/i13/

[5] Alan E Gelfand and Erin M Schliep. 2016. Spatial Statistics and Gaussian Processes: A Beautiful Marriage. *Spatial Statistics* 18 (2016), 86–104.

[6] Douglas Nychka, Reinhard Furrer, John Paige, and Stephan Sain. 2017. fields: Tools for Spatial Data. https://doi.org/10.5065/D6W957CT R package version 9.7.

[7] Paulo J. Ribeiro Jr and Peter J. Diggle. 2016. *geoR: Analysis of Geostatistical Data*. https://CRAN.R-project.org/package=geoR R package version 1.7-5.2.

5 ACKNOWLEDGMENTS

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST). We would like to thank NVIDIA for hardware donations in the context of a GPU Research Center, Intel for support in the form of an Intel Parallel Computing Center award, Cray for support provided during the Center of Excellence award to the Extreme Computing Research Center at KAUST, and KAUST IT Research Computing for their hardware support on the GPU-based system. This research made use of the resources of the KAUST Supercomputing Laboratory.