Algorithm performance dataset from NASA open-source software

Geoffrey F. Bomarito, Patrick E. Leser, James E. Warner, William P. Leser

July 29, 2020

Background

NASA Langley Research Center has recently developed and released the open-source software Multi Model Monte Carlo with Python (`MXMCPy` - LAR-19756-1) as a general capability for computing the statistics of outputs from an expensive, high-fidelity model by leveraging faster, low-fidelity models for speedup. Given a fixed computational budget and a collection of models with varying cost/accuracy, multi model Monte Carlo (MC) seeks a sample allocation strategy across the models that results in an estimator with optimal variance reduction. `MXMCPy` is a versatile tool that enables convenient access to many existing multi-model MC approaches (over a dozen algorithms available) within one modular and extensible package [1]. With `MXMCPy`, users can easily compare existing methods to determine the best choice for their particular problem, while developers have a basis for implementing and sharing new variance reduction approaches. However, there is currently very little understanding about which algorithm will perform best for a given problem (defined by the correlation between and relative cost of the available models) without a brute force search.

Dataset Description

A recent study conducted a performance comparison across all available algorithms in `MXMCPy`. The algorithms were evaluated across a huge number of random model scenarios, where each scenario is defined by the elements of a covariance matrix (describing the correlation between models) and associated model costs (describing the run time of each model). For different numbers of available models $M \in \{2, 3, 4, 5, 6\}$, 1×10^6 model scenarios were randomly generated, and each algorithm was executed to determine the optimal estimator variance it would provide for each scenario and how it would allocate samples across the available models. Several conclusions about relative algorithm performance were drawn based on these results, but an attempt to understand when and why each algorithm behaved as they did for given covariance matrices and model costs was beyond the scope of the study.

As a byproduct of this study, a large dataset of performance data from the `MXMCPy` algorithms was produced. The dataset is stored in HDF5 file format [2] and is roughly 50GB in size. For each of the millions of model scenarios, the relevant data includes

- Algorithm name (string describing which of the `MXMCPy` algorithms was used, see [1])
- Covariance matrix (two dimensional array of size $M \times M$)
- Model costs (one dimensional array of size M)
- Optimal estimator variance (scalar)
- Sample allocation (two dimensional array describing how many times to evaluate each model, see [1])

The dataset is currently stored on the K Cluster high performance computing resource at NASA Langley Research Center.

It is important to point out that *this dataset can be straightforwardly reproduced using the open-source code* `MXMCPy`. However, the motivation to release the dataset is for convenience - *it took months of CPU time to execute the millions of random model scenarios* to produce it. Releasing the dataset will allow researchers and students to investigate the performance of different multi model MC algorithms to discover trends and patterns and to be able to predict and understand when/why algorithms behave the way they do. This knowledge would help guide future NASA research in the area of uncertainty quantification and multi model methods.

References

[1] G. F. Bomarito, J. E. Warner, P. E. Leser, W. P. Leser, and L. Morrill. Multi Model Monte Carlo with Python (MXMCPy). NASA/TM-2020-220585, 2020.

[2] The HDF Group: Hierarchical Data Format, Version 5. http://www.hdfgroup.org/HDF5/, Jan. 2016.