Model-Agnostic Influence Analysis for Performance Data

Rahul Sridhar
University of California, Irvine
Lawrence Livermore National Laboratory, CA
rsridha2@uci.edu

Rushil Anirudh
Lawrence Livermore National Laboratory, CA
anirudh1@llnl.gov

Jayaraman J. Thiagarajan
Lawrence Livermore National Laboratory, CA
jayaramanthi1@llnl.gov

Nikhil Jain
Lawrence Livermore National Laboratory, CA
jain6@llnl.gov

Todd Gamblin
Lawrence Livermore National Laboratory, CA
gamblin2@llnl.gov

ABSTRACT

Execution time of an application is affected by several performance parameters, e.g. number of threads, decomposition, etc. Hence, an important problem in high performance computing is to study the influence of these parameters on the performance of an application. Additionally, quantifying the influence of individual parameter configurations (data samples) on performance also aids in identifying sub-domains of interest in high-dimensional parameter spaces. Conventionally, such analysis is performed using a surrogate model, which introduces its own bias that is often non-trivial to undo, leading to inaccurate results. In this work, we propose an entirely data-driven, model-agnostic influence analysis approach based on recent advances in analyzing functions on graphs. We show that the problem of identifying influential parameters (features) and configurations (samples) can be effectively addressed within this framework.

CCS CONCEPTS

• **General and reference → Performance**; • **Computing methodologies →** *Machine learning*; • **Mathematics of computing →** *Graph theory*;

KEYWORDS

HPC performance analysis, parameter influence, sample influence, spectral graph analysis

ACM Reference format:
Rahul Sridhar, Rushil Anirudh, Jayaraman J. Thiagarajan, Nikhil Jain, and Todd Gamblin. 2017. Model-Agnostic Influence Analysis for Performance Data. In *Proceedings of SC, The International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, Nov 2017 (SC17),* 2 pages.
https://doi.org/10.475/123_4

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-ABS-739487).

SC17, Nov 2017, Denver, CO, USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

In the domain of high performance computing (HPC), performance measures such as a program's execution time are sensitive to changes in performance parameters such as the number of threads, power cap, order of loop parallelization, and other application-specific parameters. Additionally, there are performance variations across different sub-domains of the parametric space. In such scenarios, it becomes imperative to quantify the influence of each parameter and parameter configuration (data sample) on performance (refer Table 1). Influence analysis primarily entails quantifying the importance of parameters and samples in predicting variations in performance. The primary benefits of studying these influences on performance are three-fold:

- Performance variations and optimization can be controlled by tweaking only the most influential parameters
- Sub-domains of the parameter space that result in the highest/lowest performance can be identified
- Performance under all possible parameter configurations can be approximately predicted using a smaller number of data samples and parameters, thereby alleviating the need for exhaustive simulations

Influence Analysis				
	Parameter 1	...	Parameter n	Performance
Config. 1	...	...	...	...
...	...	...	...	...
Config. m	...	...	...	...

Table 1: Different parameters and parameter configurations have different influences on program performance

Existing approaches: Conventional machine learning models that are used for quantifying influence are implicitly founded on certain assumptions, such as a linear relationship between the parameters and the response variable, which lead to biased and inaccurate estimates of influences. We propose a model-agnostic approach that uses techniques from signal processing on graphs [1] constructed on the parameter space to quantify influence.

Our Approach: The goal of this work is to not only infer relatively unbiased estimates of parameter and sample influences, but also to regularize predictive model learning using the influence patterns. Apart from making minimal assumptions about the underlying data domain, another advantage of our approach is that it can easily be extended to incorporate domain-specific knowledge - if the domain expert knows that certain sub-domains of the parameter space are inherently more influential than others, this can be fused into the analysis prior to making any inference.

2 METHODOLOGY

We perform our analysis on data collected from Kripke, a parallel Sn transport mini-app [3]. Each data sample consists of a unique configuration of the parameters and the corresponding execution time of the program. We construct an undirected, weighted graph over the parameter space, where each node of the graph corresponds to a parameter configuration. We then define a real-valued signal over the graph, where the signal value for each node is its execution time. Edges between nodes are weighted using a distance metric known as the Gower distance [2] (a metric used especially when the data is composed of discrete and continuous parameters), and only the edges between a node and its top k neighbors are retained.

Analogous to a signal in the time domain, a Fourier frequency decomposition of the graph signal can be performed to identify its constituent frequency components, where frequency is a measure of how much the signal varies from a node to its neighbors. **Parameter influence estimation**: Changes in graph connectivity will result in changes in the Fourier frequency spectrum. We make use of this fact to estimate parameter influence using a leave-one-parameter-out (LOPO) strategy.

More specifically, we construct a graph using a LOPO strategy for each parameter (with the signal unchanged) and compute its Fourier frequency spectrum. Graphs obtained when removing weak parameters should behave very similar to the original graph with all parameters, resulting in nearly the same spectral signature, whereas graphs constructed without an influential parameter should result in a very different spectral signature. We quantify the differences in frequency spectra using the *energy in high frequencies*, which is the absolute difference of the Euclidean norms of the high frequency components of both spectra. Consequently, influential parameters will have higher energy than weak parameters.

Sample influence estimation: Alternately, in order to quantify the influence of each data sample, we study the changes in the low frequencies of the spectra when the signal of a sample is replaced by the average value of the signal of its neighbors. If the sample is not influential, doing such a modification of its signal would result in minimal changes in the low frequency components of the spectrum, whereas we can expect relatively higher changes if the sample is influential.

3 RESULTS

We find that the order of loop parallelization has the highest influence on program execution time, followed by the number of threads, and other Kripke-specific parameters. The results closely align with the intuition of domain experts. By also performing a leave-two-parameters-out analysis, we show that the observed results could not have been obtained due to higher-order correlations between two parameters suppressing the influences of each other. We also performed the analysis on other datasets from Kripke with different initial settings and made similar observations. Finally, we also compute the influence of individual data samples and build predictive models with and without the most influential samples to show that the removal of these samples is highly detrimental to model performance.

4 CONCLUSION AND FUTURE WORK

In summary, we present a novel, model-agnostic approach for quantifying parameter and sample influence and prove its efficacy by studying their influences on the performance of a HPC application. Our immediate next steps entail regularizing predictive model learning using the obtained influence patterns. We also plan on looking at ways to perform transfer learning from Kripke to other related HPC applications.

REFERENCES

[1] David I. Shuman et al. 2013. The emerging field of signal processing on graphs: Extending high- dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30, 3 (2013), 83–98. https://doi.org/10.1109/MSP.2012.2235192

[2] J. C. Gower. 1971. A General Coefficient of Similarity and Some of Its Properties. *Biometrics* 27, 4 (1971), 857–871. https://doi.org/10.2307/2528823

[3] Adam J. Kunen, Teresa S. Bailey, and Peter N. Brown. 2015. Kripke - A Massively Parallel Transport Mini-App. *American Nuclear Society Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications, and the Monte Carlo Method*. https://codesign.llnl.gov/pdfs/Kripke_ANS_2015_Paper.pdf