DoE Fellowship Essays

immediate

January 4, 2021

1 Field of Interest

> In terms a general audience would understand, describe an important, outstanding challenge in mathematics, statistics or computer science that you would like to pursue in your research. (1/3)
>
> ---
>
> Describe the particular mathematics, statistics or computer science problem that you would like to pursue in your research. What would be the impact on high-performance computing and on science, engineering and/or society in general if this challenge could be successfully addressed? (2/3)

Bayesian inference, or the application of Bayes' Law to find the likelihood of a hypothesis, is used to solve optimization problems under uncertainty (e.g. noise, lack of data). As society continues to use computers to make critical decisions, including engineering designs and intelligent machines, addressing the uncertainty reduces the probability of failures that can lead to deaths and financial loss. However, directly solving the Bayesian inference problem is intractable for high-dimensional problems since the cost scales exponentially with the dimension. My research will bypass this cost by casting the problem as a variational inference problem, permitting the use of optimization, specifically gradient-based, methods.

This research will have three components: mathematical analysis, algorithmic development, and implementation. I will investigate ways to improve the theoretical convergence of first-order and approximate second-order (i.e., Hessian approximations) methods by using different distance measures for the objective function, adaptive step policies, and inexact gradients. I will then develop heuristics to accelerate the empirical performance of the gradient-based methods by using better line search strategies, low-rank approximations, and randomized methods. Finally, the numerical methods will be implemented on distributed-memory machines and used to solve the Bayesian inference problem in various applications.

If successful, these gradient-based methods provide a novel, scalable algorithm for large-scale uncertainty quantification on high-performance machines. Each iteration requires gradients or approximate Hessians, which scales with matrix operations and avoids the Hessian needed in Newton-Krylov. In my ongoing research on a parallel linear program solver, I designed heuristics that reduce runtime by several orders of magnitude. The hypothesis is that similar heuristics can be discovered here, enabling faster convergence than sampling-based methods. Altogether, this algorithm will enable high-fidelity simulations in less time towards scientific problems that are of interest to the DoE, such as the modeling of the Antarctic ice sheet and rare-event prediction for nuclear propulsion plants.

2 High-Performance Computing

What is the most complex calculation you have run on a high-performance machine as part of your research experience? Or if you haven't run a high-performance computing system, tell us about the most complex computational problem you have tackled. (1/2)

Imagine if you were given access to resources 100 times more powerful than what you have access to. What would that enable you to do, and what do you perceive the mathematical and computer science challenges to be? (1/2)

The most complex program I have run on a high-performance machine is a parallel algorithm for solving positive linear programs (LP), which is an LP with non-negative values in its formulation. I developed the code using Python and petsc4py, a Python interface of the PETSc library. I then used the program to solve problems like graph matchings and densest subgraph for graphs with up to a billion edges on the supercomputer Stampede2. The most challenging aspect of this research was creating solutions to problems that existed only at the implementation level, such as numerical overflow in smooth max calculations. While there existed sequential solutions, such as masking and using Python's indexing system, these operations could not be performed in parallel with the petscp4py library. To circumvent these issues, I designed modifications and proved correctness to the numerical method, such as showing the preservation of large values in smooth min does not affect convergence, normalizing smooth max prevents numerical overflow, and creating an optimization subproblem to increase step sizes can reduce the solve time from hours and days to minutes.

Given compute resources 100 times more powerful than what I have now, achieving linear speedup depends on the type of hardware improvements. Consider the LP from above. While 100 times more nodes with the same power will allow large decision vectors and constraint matrices from an LP to be split into smaller components, thus reducing the work per processor, synchronizations after each iteration of the LP will be the dominating cost. If given access to nodes that are 100 times more powerful, then the local matrix computations can be solved in less time. However, communication from the matrix computations between processors will be the bottleneck. Therefore, achieving linear speedup will require an algorithm redesign. This includes finding ways to reduce synchronization by using approximate matrix computations and asynchronous iterations. To reduce communication, decomposing a large LP into multiple smaller LPs, such as by using graph partitions, will enable each LP subproblem to run independently for multiple iterations before communicating between nodes.

3 Program of Study

> Describe how the courses listed in your planned program of study would help prepare you to address the challenges you have described in questions 1 and 2. Discuss your rationale for choosing these courses. How will the science or engineering application courses you have selected impact your research?

I have built my program of study to gain a fundamental understanding of optimization algorithms and relevant applications in computational science. With these selected courses, I can leverage insights from the theory of an optimization problem or special properties of the application to modify an algorithm to have both strong theoretical guarantees and empirical performances for solving the Bayesian inference problem on emerging high-performance computing systems.

The courses linear optimization (ISyE6661) and nonlinear optimization (ISyE6663) introduce the theory and algorithms for various convex and non-convex optimization problems. I have completed computational data analysis (CSE6740), which covers algorithms in machine learning, Bayesian statistics, and information theoretic learning. These three courses will provide a firm understanding on the strengths and limitations of methods for solving the Bayesian inference problem. With the optimization courses, I will learn how to exploit the underlying theory of gradient-based methods to design faster algorithms.

One of the main applications I want to solve with Bayesian inference is uncertainty quantification in computational science problems involving either fluid dynamics and geoscience. Therefore, I selected viscous fluid flow (AE6009) to learn about fluid mechanics. I also selected computational fluid dynamics (AE6042) to learn how to translate a physics of a problem into mathematics (using finite-difference or finite-volume) and then how to solve these problems with my own computational fluid dynamics code. These two courses will also expose me to the special properties of the flow-based problem, including sparsity or structures in the discretizations, which I can use to develop specially catered methods with better performance.

Finally, I have learned how to write high-performance code in the course high performance parallel computing (CSE6230), which emphasizes the design, implementation, and analysis of high-performance codes on multicore and coprocessor machines. From this course, I have the necessary training to exploit high-performance machines and modify numerical methods to solve optimization problems efficiently at scale.