Project proposal

Accelerating Bayesian Optimisation for Algorithm Selection

Supervisors:
Juergen Branke (juergen.branke@wbs.ac.uk)

Michael Pearce (scrambledpie@gmail.com)

Background: Algorithms are designed to be general purpose tools. However, multiple algorithms exist for solving a single class of problem and problem specific algorithm selection has received much attention, for example given a range of datasets, many machine learning methods exist and the best method depends on dataset properties. For learning a mapping from problem characteristics to best algorithm, Bayesian optimisation is a is receiving a great deal of attention however they are computationally rather expensive. Meanwhile, there has been much recent work on combining GPU computation to speed up Bayesian optimisation algorithms reducing overhead thereby broadening their applicability. The algorithm selection problem often requires large Monte Carlo integrals which can be prohibitively expensive on CPU while on GP can be 100x faster.

Mini-project: The goal would be to develop a GPU accelerated Bayesian optimisation method for automatic algorithm selection. This can be written in basic python using Tensorflow or Pytorch. Benchmark this against CPU version of the same algorithm.

Deliverable: Working code, paper with some empirical results on artificial benchmark problems and real world datasets if time permits.

PhD prospect: The general field of Bayesian optimization is broad and growing and GPU acceleration is a very exciting topic (GPFlowOpt and BOTorch are BO packages built on Tensorflow and Pytorch respectively). There exist many cases of Bayesian optimization that involve Monte-Carlo integrals: robust optimisation, multi objective optimisation with preference learning, quadrature optimisation, algorithm selection, where the vast parallelisation of GPUs can provide enormous speed ups. Also, there is scope in improving or adapting the Bayesian optimisation algorithm itself, for example by exploiting correlation between algorithms (for some problem instances, all algorithms do well or poorly), or aiming to return a set of good algorithms rather than the single best algorithm.

Student requirements: Programming skills.

Reference:

M. Pearce and J. Branke, "Efficient expected improvement estimation for continuous multiple ranking and selection," *2017 Winter Simulation Conference (WSC)*, Las Vegas, NV, 2017, pp. 2161-2172.