BOAT: Building Auto-Tuners with Structured Bayesian Optimization

Valentin Dalibard, Michael Schaarschmidt, Eiko Yoneki
Presented by Harrison Brown for R244
Auto-tuners

- Difficult to manually tune complex configuration parameters for various problems
  - Compiler flags, configuration files, number and assignment of machines, etc
- Can expose configuration parameters and performance metrics to black box optimizers
  - May take thousands of iterations on complex problems
    - For systems problems with long evaluation times this process fails
    - Does not leverage any contextual information about problem
- OpenTuner – ensembles of various algorithms (evolutionary, hill climbing, etc)
- Spearmint – traditional Bayesian Optimization
Key Terms

- **Gaussian Process** – collection of random variables
  - Every finite linear combination of variables is normally distributed
- **Parametric models** – fixed number of parameters
  - Feedforward neural networks, linear regression, logistic regression
- **Non-parametric models** – unbounded number of parameters
  - K-nearest neighbors

---

**Algorithm 1 The Bayesian optimization methodology**

**Input:** Objective function $f()$

**Input:** Acquisition function $\alpha()$

1. Initialize the Gaussian process $G$
2. for $i = 1, 2, \ldots$ do
   3. Sample point: $x_t \leftarrow \text{arg max}_x \alpha(G(x))$
   4. Evaluate new point: $y_t \leftarrow f(x_t)$
   5. Update the Gaussian process: $G \leftarrow G \mid (x_t, y_t)$
3. end for
BOAT Contributions

• Novel algorithm – Structured Bayesian Optimization
  • Structured probabilistic model provided by developer
    • Discards regions of low performance where traditional Bayesian Optimization over explores
    • Semi-parametric model – developer provides parametric parts to describe general behavior

• BOAT – a framework to allow developers to build auto-tuners for their systems
  • To be used in situations where generic autotuners fail
  • Model allows for probabilistic inference
    • Can make predictions without large computational cost
Using BOAT and SBO

• Configuration Space
• Objective function and runtime measurements
• Probabilistic model of system behavior
  • Semi-parametric model
    • Constructor – prior distribution to sample model parameters
    • A parametric function for a given input that returns a prediction
  • DAG model, allows combination of multiple semi-parametric models
    • Exploits conditional independence to train independently given the measured outputs
    • Allows maximization of expected improvement in SBO
**BOAT Recommended Usage**

- Initially use generic probabilistic model – regular Bayesian optimization
- Incrementally add structure until convergence
  - Unclear on how long this process typically takes
Java Garbage Collection Case Study

- Tuning garbage collection flags of JVM database (Cassandra)
  - Only 3 parameters, very small domain
- Objective: 99th Percentile Latency using YCSB cloud benchmark
- Spearmint Converges within 16 iterations (4 hours)
- BOAT converges to within 10% of best-found performance by 2nd iteration
Neural Network Training Case Study

- In Tensorflow, users must set what available machines to be used and assign work
- Input: NN architecture, available machines, batch size
- Tuning synchronous distributed SGD
  - Parameters: worker machines, parameter servers, workload partition
  - Objective: minimize average iteration time
- OpenTuner only marginally better than uniform GPUs assignment (9.82s)
- BOAT completed within 2 hours, significant gains if architectures take weeks to train
Novel algorithm and framework for probabilistic models

- Easy to build probabilistic models with little effort

Significant gains can be made on complex problems such as neural network tuning

- Useful as black box optimizers may often fail in these domains
- If a developer has contextual knowledge, that should be leveraged
BOAT Criticisms

- BOAT does not give information about performance with incorrect contextual information
- Niche contribution – enough knowledge to provide model, not enough to set the configuration parameters
- Motivation states "auto-tuners like [...] OpenTuner [...] usually require thousands of evaluations"
  - More evidence is warranted in form of case studies / experiments
  - OpenTuner used 7 projects
  - Time versus iterations. BO has high iteration overhead
- OpenTuner and Spearmint - Python, C++ user friendly
- Performance gains vs usability
References

