Preliminary Evaluation of Hyperopt Algorithms on HPOLib

James Bergstra, Brent Komer, Chris Eliasmith
David Warde-Farley

University of Waterloo
Université de Montréal

Abstract:

Model selection, also known as hyperparameter tuning, can be viewed as a blackbox optimization problem. Recently the HPOlib benchmarking suite was advanced to facilitate algorithm comparison between hyperparameter optimization algorithms. We compare seven optimization algorithms implemented in the Hyperopt optimization package, including a new annealing-type algorithm and a new family of Gaussian Process-based SMBO methods, on four screening problems from HPOLib.

On these screening problems Gaussian Processes (GPs) are the most call-efficient. Vanilla GP-based methods (stationary RBF kernels and maximum likelihood kernel parameter estimation) provide a near-perfect ability to optimize the benchmarks. Despite being slower than more heuristic baselines, a Theano-based GP-SMBO implementation requires at most a few seconds to produce a candidate evaluation point. We compare this vanilla approach to Hybrid Monte-Carlo integration of the kernel lengthscales and fail to find compelling advantages of this more expensive procedure.

Optimization Algorithms

1. Random Search
2. Annealed Random Search
3. Tree of Parzen Estimators (TPE)
4. Regression Tree SMBO
5. GP-UCB SMBO (ML lengthscale)
6. GP-EI SMBO (Max. Lik. lengthscale)
7. GP-EI SMBO (HMC lengthscales)

Search Domains

1. Branin-Hoo (Branin):
   2D continuous multimodal
2. LDA-on-grid (LDA)
   3D discretized continuous
3. SVM-on-grid (SVM)
   3D discretized continuous
4. Hartmann-6 (Har6)
   6D continuous multimodal

Experiment 1: Convergence Rate

For each method on each search problem, repeat 10 times:
  Seed the search with 15 random draws (different on each trial).
  After earch iteration, determine best score to date; average across trials.
Want to see: steep downward slope starting at t=15.

Experiment 2: Success Rate

For each method on each search problem, repeat 10 times.
  Seed the search with 15 random draws (different on each trial).
  After T=50 (for easy searches) or T=100 steps, report best value.
Want to see: tight boxplot around global optimum, indicating consistent discovery of global optimum.

Experiment 3: Overhead of Optimization

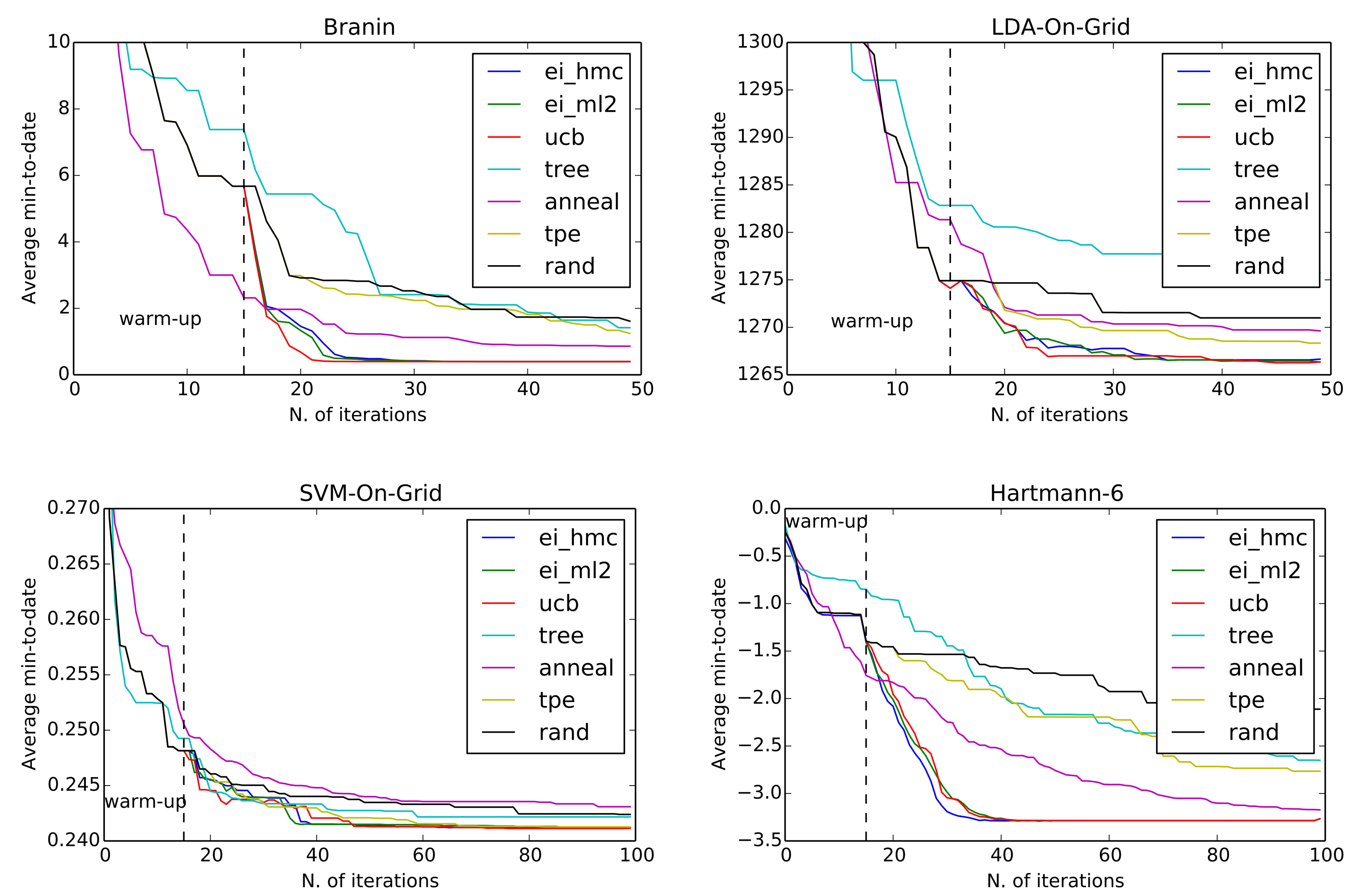
For each method on each search problem, repeat 10 times.
  Seed the search with 15 random draws (different on each trial).
  After each iteration measure how long it takes the optimizer to suggest a new trial point.
Want to see: consistently small amount of time. Optimizer will be wasted on Black box functions that evaluate in less time than the optimizer overhead.

Key References:

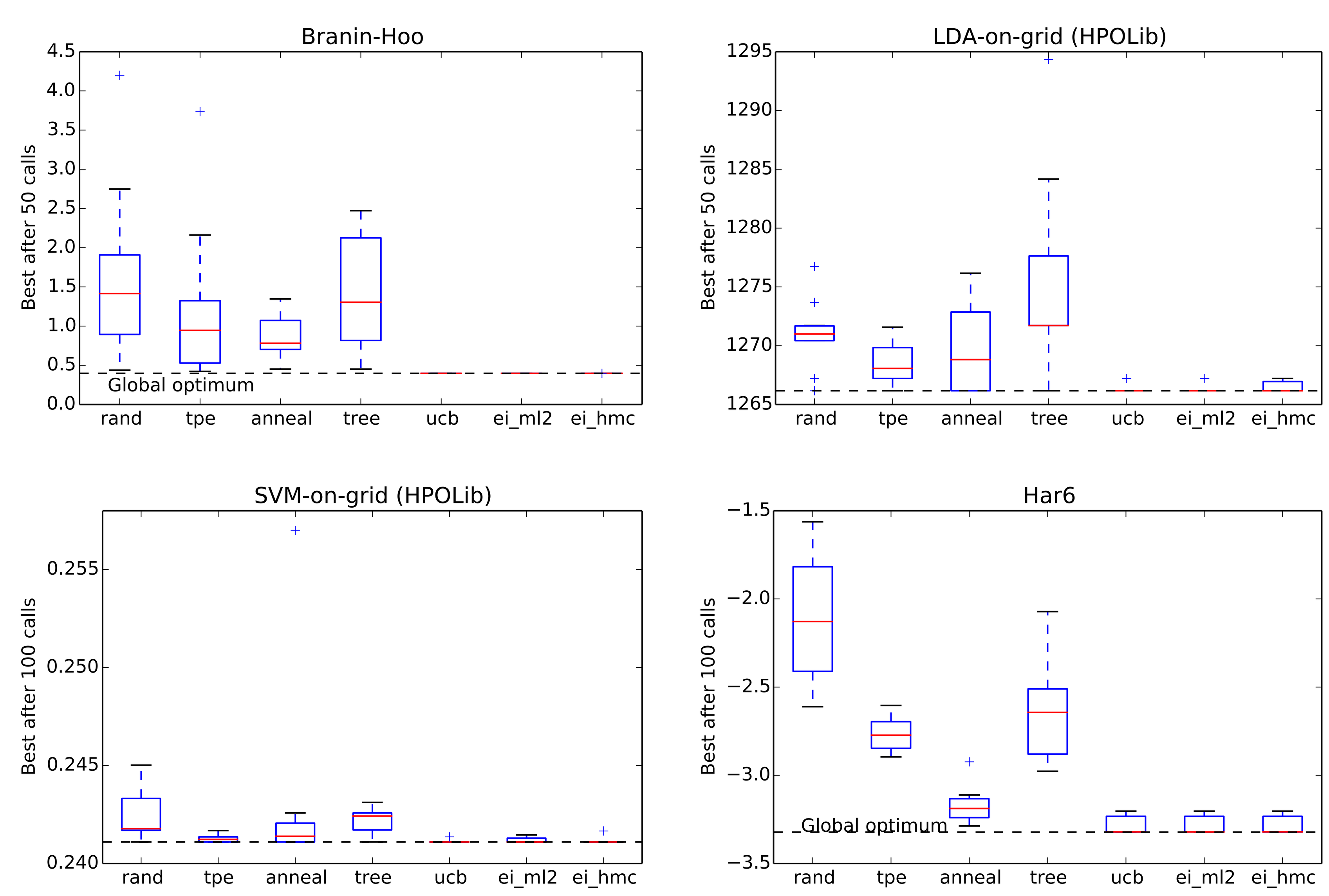
Hyperopt Project: http://hyperopt.github.io/hyperopt
HPOLib Project: http://www.automl.org
Theano Project: http://deeplearning.net/software/theano

Jones DR. J. Global Optimization 2001.
Bergstra J, Yamins D, Cox DD. SciPy 2013.
Snoek J, Larochelle H, Adams RP. NIPS 2012.
Eggensperger K, Feurer M, Hutter F., Bergstra J. Snoek J, Hoos H, Leyton-Brown K (2013)

Results 1: Convergence Rate



Results 2: Success Rate



Results 3: Overhead of Optimization