Hybrid Bayesian Evolutionary Optimization for Hyperparameter Tuning

Lukas Atkinson
Frankfurt University of Applied Sciences
Frankfurt a. M., Germany
lukas.atkinson@fb2.fra-uas.de

Robin Müller-Bady
Frankfurt University of Applied Sciences
Frankfurt a. M., Germany
mueller-bady@fb2.fra-uas.de

Martin Kappes
Frankfurt University of Applied Sciences
Frankfurt a. M., Germany
kappes@fb2.fra-uas.de

ABSTRACT

In this paper, we present a Hybrid Bayesian–Evolutionary tuning algorithm (HBEtune) for tuning machine learning algorithms or evolutionary algorithms, and analyze its performance. HBEtune combines meta-EA and Bayesian optimization techniques.

As hyperparameter tuning is a noisy, black-box optimization problem with expensive target functions, practical tuners must aim to minimize the number of necessary samples. In our method, we guide the EA's recombination operator towards more promising samples by employing the expected improvement acquisition criterion commonly used in Bayesian optimization. The expected improvement is evaluated on a surrogate model using a Gaussian process regression.

HBEtune shows generally competitive performance when compared with the state of the art *irace* tuner. Performance is analyzed across a suite of synthetic and real-world benchmark problems.

CCS CONCEPTS

• **Theory of computation** → **Mathematical optimization**; • **Computing methodologies** → *Gaussian processes.*

KEYWORDS

parameter tuning, algorithm configuration, evolutionary algorithms, Bayesian optimization, Gaussian process regression

ACM Reference Format:
Lukas Atkinson, Robin Müller-Bady, and Martin Kappes. 2020. Hybrid Bayesian Evolutionary Optimization for Hyperparameter Tuning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389952

1 INTRODUCTION

Automated hyperparameter tuning is a challenging optimization problem since it amounts to black-box global optimization of a noisy, expensive to evaluate, and possibly multimodal target function. While evolutionary algorithms have been applied in this space, they rely on an accurate fitness metric, and directly using the target function is not appropriate here. Fitness can be approximated e.g. by using hierarchical populations and by using surrogate models [6]. Surrogate models can also be used to *inform* the evolutionary operators e.g. by ranking candidates on a reduced model, instead of evaluating them directly on a possibly-misleading model [1] [10].

Bayesian Optimization (BO) is a data-efficient strategy for optimizing noisy black-box functions. It provides a principled approach for dealing with uncertainty, which addresses both noisiness and optimal allocation of evaluations. BO uses a surrogate model representing our prior belief of the behavior of the target function, and an acquisition function (or loss function) for selecting the next sample to be evaluated [13]. The model of choice is a Gaussian Process Regression (GPR) [11] since this model can provide both a prediction and a measure of model uncertainty. Acquisition by expected improvement (EI) over the best known value has shown good performance in practice [14]. EI provides a principled way to include model uncertainty in an exploration–exploitation tradeoff. After BO via GPR+EI was proposed for efficient global optimization [7], it has also seen substantial use in hyperparameter tuning [5][14], also for the case of tuning EAs [12].

This paper presents and evaluates HBEtune, a tuning algorithm combining an EA-style metaheuristic with BO techniques. HBEtune is comparable to SPO [2] which also builds a surrogate model, or irace [9] which instead performs iterated racing in order to make accurate fitness comparisons between individuals. Previous hybrid Bayesian–Evolutionary approaches include ParEGO [8] which uses an EA to optimize the EI surface, and SAPEO [16] which evaluates individuals on a surrogate model only if the model is fairly certain.

2 PROPOSED APPROACH

HBEtune combines an irace-style population-based optimization approach with Bayesian optimization to guide the evolution, in particular by implementing GPR-guided mutation (see Fig. 1). This is intended to provide good tuning results at small evaluation budgets, possibly before the tuner can converge.

Per EA generation, a GPR model is trained on the observations, which also provides a parameter relevance estimation. Guided mutation generates offspring by first creating random candidates $c \sim \mathcal{N}(p)$ in the parent p's neighborhood. The candidate $o = \arg\max_c EI(c)$ with maximal EI is evaluated on the true objective and competes against the parent during selection. One parent can also produce offspring via local search on the EI surface. Once the budget is exhausted, the best configuration is selected according to its upper confidence bound.

A prototypical implementation is provided as an easy to configure command line tool[1] which supports parallel evaluation. The single objective can have linear or logarithmic scale. Parameters

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

[1] https://github.com/latk/hbetune.rs, includes evaluation experiments

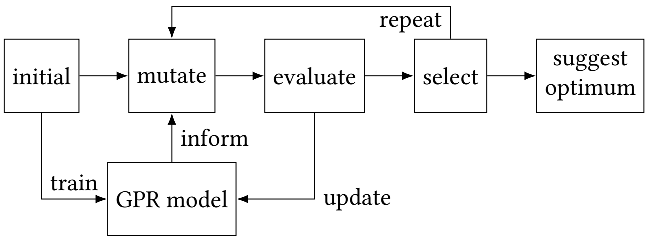

Figure 1: HBEtune optimization schema

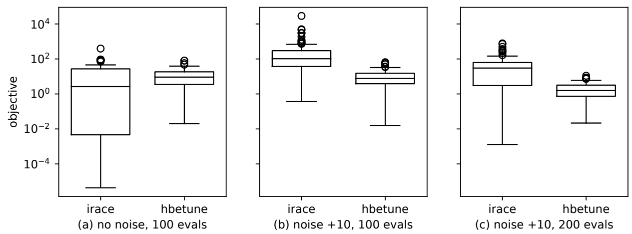

Figure 2: Performance of suggested configurations suggested by irace and HBEtune on the Goldstein-Price function (lower is better)

Table 1: Best tuner in various experiments

task	deterministic	noisy	
	100 evals	100 evals	200 evals
Rosenbrock 2D	HBE	HBE	HBE
Goldstein-Price 2D	irace*	HBE	HBE
Himmelblau 2D	HBE	HBE	HBE
Rastrigin 2D	irace	irace	irace*
HBE/Rosenbrock	150 evals: irace		
HBE/Rastrigin	150 evals: irace		
ACOTSP	150 evals: inconclusive		

*significant at $p \approx .05$, other results at $p < .001$

can be bounded and have different scales, and integer parameters are handled per [4].

3 EVALUATION

HBEtune was compared against irace [9] on a suite of problems with different budgets and noise levels. Tuning of synthetic problems was repeated 100 times, with the true value of the suggested configuration being used as utility metric. Noise-free tasks used irace in deterministic mode to conserve evaluations. Real-world tasks included tuning HBEtune itself (5 parameters), and tuning ACOTSP (6 parameters) [15]. Tuning was repeated 30 times, and the suggested configuration was sampled 40 times. The Mann-Whitney rank-sum test was applied on the results to determine whether HBEtune finds better configurations.

Table 1 summarizes the findings. Fig. 2 illustrates results on the Goldstein-Price function. The synthetic examples indicate that HBEtune is often able to find significantly better results than irace, especially on noisy tasks. However, this does not extend to the Rastrigin function or to the practical experiments.

4 CONCLUSION

HBEtune is able to outperform irace on some but not all studied problems, and could therefore be a competitive tuner for small evaluation budgets. However, further research is needed to investigate behavior on more and more complex tuning problems, and to compare against additional tuners such as hyperopt [3] or spearmint [14].

ACKNOWLEDGMENTS

This work was supported in the context of the German Federal Ministry for Economic Affairs and Energy grant no ZF4131805MS9.

REFERENCES

[1] K. S. Anderson and YuHong Hsu. 1999. Genetic crossover strategy using an approximation concept. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*. IEEE, 527–533. https://doi.org/10.1109/CEC.1999.781978

[2] Thomas Bartz-Beielstein, Christian Lasarczyk, and Mike Preuss. 2005. Sequential parameter optimization. In *Evolutionary Computation, 2005. CEC'05. IEEE Congress on*. Edinburgh, Scotland, 773–780. http://www.spotseven.de/wp-content/papercite-data/pdf/blp05.pdf

[3] James Bergstra, Daniel Yamins, and David Daniel Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning – Volume 28 (ICML '13)*. JMLR, 115–123.

[4] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. 2020. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing* 380 (2020), 20–35. https://doi.org/10.1016/j.neucom.2019.11.004

[5] Frank Hutter. 2009. *Automated configuration of algorithms for solving hard computational problems*. Ph.D. Dissertation. University of British Columbia.

[6] Yaochu Jin. 2005. A comprehensive survey of fitness approximation in evolutionary computation. *Soft computing* 9, 1 (2005), 3–12. https://doi.org/10.1007/s00500-003-0328-5

[7] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.

[8] J. Knowles. 2006. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10, 1 (Feb 2006), 50–66. https://doi.org/10.1109/TEVC.2005.851274

[9] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. 2016. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3 (2016), 43–58. https://CRAN.R-project.org/package=irace

[10] Khaled Rasheed and Haym Hirsh. 2000. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*. 628–635.

[11] Carl Edward Rasmussen and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, USA. http://www.gaussianprocess.org/gpml/

[12] Ibai Roman, Josu Ceberio, Alexander Mendiburu, and Jose A Lozano. 2016. Bayesian optimization for parameter tuning in evolutionary algorithms. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 4839–4845.

[13] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (2015), 148–175.

[14] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*. 2951–2959.

[15] Thomas Stützle. 2002. ACOTSP v1.03. http://www.aco-metaheuristic.org/aco-code/ a software package implementing various Ant Colony Optimization algorithms applied to the symmetric Traveling Salesman Problem.

[16] Vanessa Volz, Günter Rudolph, and Boris Naujoks. 2017. Surrogate-Assisted Partial Order-Based Evolutionary Optimisation. In *Evolutionary Multi-Criterion Optimization*, Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme (Eds.). Springer, Cham, 639–653. https://doi.org/10.1007/978-3-319-54157-0_43