Next Generation of Empirical Performance Prediction

Hadar Shavit

Chair of AI Methodology
RWTH Aachen University
shavit@aim.rwth-aachen.de

Abstract

Empirical performance models (EPMs) predict algorithm performance without execution, enabling applications such as algorithm selection, surrogate-based optimisation, and benchmarking. However, their effectiveness is currently constrained by the quality of feature representations and the predictive models themselves. My thesis advances EPMs by addressing both limitations. To further enhance usability and foster broader adoption, I also introduce a Python library that unifies state-of-the-art methods under a single API. These contributions aim to make EPMs more accurate, versatile, and accessible to the broader AI community.

Introduction

Empirical performance prediction is a fundamental task with many applications. At its core, it seeks to predict performance of an algorithm from features or hyperparameter configurations, without executing the algorithm. Empirical performance models (EPMs) have numerous applications, including surrogate-based optimisation, algorithm selection, explainability, benchmarking, and more (Hutter et al. 2014). My thesis aims to improve EPMs for different use cases by exploring two directions:

- Improving the features provided to the EPM
- Building better performance models

The ultimate goal of my thesis is to develop better EPMs that are more widely used and beneficial in practice.

Formal definition

In empirical performance prediction, given a feature set $\mathbf{z} \in \mathcal{F}$ which describe a problem instance (*e.g.*, a SAT instance or a machine learning dataset), and/or a (hyper)parameter configuration of an algorithm $\theta \in \mathbf{\Theta}$ (*e.g.*, SAT solver configuration or hyperparameters of a neural network), the goal is to build a surrogate model f that predicts the performance of the algorithm $f : \mathcal{F} \times \mathbf{\Theta} \to P$. In some use cases (such as model-based configuration (Hutter, Hoos, and Leyton-Brown 2011)), it is also necessary to predict the uncertainty of the prediction, commonly expressed in terms of variance σ^2: $f : \mathcal{F} \times \mathbf{\Theta} \to P \times V$. Furthermore, there exist also specialised EPMs that predict the distribution of the performance of stochastic algorithms (Eggensperger, Lindauer, and Hutter 2018): $f : \mathcal{F} \times \mathbf{\Theta} \to \Delta(P)$.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related work

Research on EPMs has a long history. For example, SATzilla 2004 (Nudelman et al. 2004) predicts the running times of SAT solvers to perform algorithm selection. The authors also introduced the SATzilla feature set, which describes SAT instances. In 2007, SATzilla achieved state-of-the-art performance, winning the SAT competition, by using an improved hierarchical EPM (Xu et al. 2008). Later, SMAC (Hutter, Hoos, and Leyton-Brown 2011) employed a random forest as an EPM to predict the performance of $\mathcal{NP}$-complete problem solvers (such as MIP and SAT solvers) over an instance set, with the goal of configuring such solvers to obtain well-performing configurations for the given set of instances.

EPMs have prominent usage in benchmarking in various fields, such as machine learning (Pfisterer et al. 2022) and $\mathcal{NP}$-complete solvers (Eggensperger et al. 2015). Additionally, EPMs have prominent usage in algorithm selection, where the best algorithm needs to be selected based on cheaply extracted features (Xu et al. 2008). Additional use-cases include explainability (Hutter, Hoos, and Leyton-Brown 2013) as well as data-center operations (Lee, Phanishayee, and Mahajan 2025).

Hutter et al. (2014) evaluated multiple EPMs in various scenarios and found that a modified random forest works best for the task. Today, various EPMs are employed in practice, most commonly Gaussian processes (Rasmussen and Williams 2006), random forests (Breiman 2001), and XGBoost (Chen and Guestrin 2016), depending on the task, the amount of available data, and the type of feature space (categorical, continuous, or mixed).

Research questions

My thesis addresses three main research questions:

- How can we obtain better features for empirical performance prediction?
- How can we design more effective empirical performance models?

- How can we encourage their broader adoption in diverse fields?

Since starting my PhD, I have already made significant progress toward answering all three research questions, with additional progress expected in the coming year.

Research progress

In this section I provide a description of the published projects from my thesis.

Revisiting SATzilla features in 2024. In this work (Shavit and Hoos 2024), we introduced an improved version of the SATzilla feature extraction tool, which describes SAT instances. The previous version of the tool suffered from various errors that resulted in many instances missing even the most basic features. In the new version, we modernised the feature extraction process and showed that, using our improved feature extraction tool, it is possible to achieve better results on three downstream tasks: satisfiability prediction, performance prediction, and algorithm selection.

ASF: Algorithm selection framework. While many methods to build EPMs and algorithm selectors have been introduced, there exists no package that allows for easy usage of these methods as well as their comparison. A unified implementation is especially valuable for the advancement of EPMs, since EPMs are used in diverse application fields with redundant efforts. ASF is a library that provides an easy-to-use interface for empirical performance prediction and algorithm selection by integrating multiple state-of-the-art methods within a unified API. It also allows stacking EPMs and selectors, as well as tuning them. While an early version of ASF is already available[1], further development is underway, including the implementation of more methods and features.

Works under review. Additional projects which are currently under review or in an advanced state of preparation include:

- Improving the performance of EPMs used in model-based algorithm configuration through dynamic ensembling.
- Investigating new methods to create more accurate EPMs.
- Better explainability and insights of the problems using EPMs.
- Assessing dimensionality reduction methods to create EPMs for high dimensional problems.

Take-home message

The key message of my thesis is that empirical performance models can be made more accurate, versatile, and usable by jointly improving feature representations, model design, and accessibility. Through new methods and tools, I aim to broaden their adoption and impact in AI research and practice.

[1] https://github.com/hadarshavit/asf

References

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016)*, 785–794.

Eggensperger, K.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2015. Efficient Benchmarking of Hyperparameter Optimizers via Surrogates. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2015)*, 1114–1120.

Eggensperger, K.; Lindauer, M.; and Hutter, F. 2018. Neural Networks for Predicting Algorithm Runtime Distributions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 1442–1448.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential Model-Based Optimization for General Algorithm Configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION 2011)*, volume 6683, 507–523.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2013. Identifying Key Algorithm Parameters and Instance Features Using Forward Selection. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION 2013)*, volume 7997 of *Lecture Notes in Computer Science*, 364–381.

Hutter, F.; Xu, L.; Hoos, H. H.; and Leyton-Brown, K. 2014. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206: 79–111.

Lee, S.; Phanishayee, A.; and Mahajan, D. 2025. Forecasting GPU Performance for Deep Learning Training and Inference. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2025)*, 493–508.

Nudelman, E.; Leyton-Brown, K.; Hoos, H. H.; Devkar, A.; and Shoham, Y. 2004. Understanding Random SAT: Beyond the Clauses-to-Variables Ratio. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, 438–452.

Pfisterer, F.; Schneider, L.; Moosbauer, J.; Binder, M.; and Bischl, B. 2022. YAHPO Gym - An Efficient Multi-Objective Multi-Fidelity Benchmark for Hyperparameter Optimization. In *Proceedings of the International Conference on Automated Machine Learning (AutoML 2022)*, volume 188, 3/1–39.

Rasmussen, C. E.; and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Shavit, H.; and Hoos, H. H. 2024. Revisiting SATZilla Features in 2024. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT 2024)*, volume 305, 27:1–27:26.

Xu, L.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2008. SATzilla: Portfolio-based Algorithm Selection for SAT. *Journal of Artificial Intelligence Research*, 32: 565–606.