SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization

Marius Lindauer\textsuperscript{1}
Katharina Eggensperger\textsuperscript{2}
Matthias Feurer\textsuperscript{2}
André Biedenkapp\textsuperscript{2}
Difan Deng\textsuperscript{1}
Carolin Benjamins\textsuperscript{1}
Tim Ruhkopf\textsuperscript{1}
René Sass\textsuperscript{1}
Frank Hutter\textsuperscript{2,3}

\textsuperscript{1}Leibniz University Hannover, \textsuperscript{2}University of Freiburg, \textsuperscript{3}Bosch Center for Artificial Intelligence

Abstract

Algorithm parameters, in particular hyperparameters of machine learning algorithms, can substantially impact their performance. To support users in determining well-performing hyperparameter configurations for their algorithms, datasets and applications at hand, SMAC3 offers a robust and flexible framework for Bayesian Optimization, which can improve performance within a few evaluations. It offers several facades and pre-sets for typical use cases, such as optimizing hyperparameters, solving low dimensional continuous (artificial) global optimization problems and configuring algorithms to perform well across multiple problem instances. The SMAC3 package is available under a permissive BSD-license at https://github.com/automl/SMAC3.

Keywords: Bayesian Optimization, Hyperparameter Optimization, Multi-Fidelity Optimization, Automated Machine Learning, Algorithm Configuration

1. Introduction

It is well known that setting hyperparameter configurations of machine learning algorithms correctly is crucial to achieve top performance on a given dataset (Bergstra and Bengio, 2012; Snoek et al., 2012). Besides simple random search (Bergstra and Bengio, 2012), evolutionary algorithms (Olson et al., 2016) and bandit approaches (Li et al., 2018), Bayesian Optimization (BO) (Mockus et al., 1978; Shahriari et al., 2016) is a commonly used approach for Hyperparameter Optimization (HPO) (Feurer and Hutter, 2019) because of its sample efficiency. Nevertheless, BO is fairly brittle to its own design choices (Lindauer et al., 2019b) and depending on the task at hand different BO approaches are required.

SMAC3 is a flexible open-source BO package that (i) implements several BO approaches, (ii) provides different facades, hiding unnecessary complexity and allowing easy usages, and (iii) can thus be robustly applied to different HPO tasks. Its usage in successful AutoML tools, such as auto-sklearn (Feurer et al., 2015) and the recent version of Auto-PyTorch (Zimm-
2. Different Use Cases and Modes of SMAC3

SMAC3 is designed to be robustly applicable to a wide range of different use-cases. To allow for maximal flexibility, SMAC3 does not only implement a Python interface, but also a CLI to communicate with arbitrary processes and programming languages. Furthermore, SMAC3 supports two kinds of parallelization techniques: (i) via DASK (Rocklin, 2015) and (ii) via running an arbitrary number of independent SMAC3 instances exchanging information via the file system. Last but not least, its modular design allows to combine different modules seamlessly. For a use-case at hand, the user only has to define a scenario, including the configuration space $\Lambda$ (Lindauer et al., 2019a), and choose a facade encoding different pre-sets of SMAC3, see Figure 1. These pre-sets are specifically designed to be efficient based on the characteristics of the given use-case, as described in the following.

2.1 SMAC4BB: SMAC for Low-dimensional and Continuous Black-Box Functions

The most general view on HPO is one of black-box optimization, where an unknown cost function $c$ is minimized with respect to its input hyperparameters $\lambda \in \Lambda$: one can equivalently frame this as minimizing the loss $\mathcal{L}$ on validation data $\mathcal{D}_{\text{val}}$ of a model trained on training data $\mathcal{D}_{\text{train}}$ with hyperparameters $\lambda$:

$$\lambda^* \in \arg \min_{\lambda \in \Lambda} c(\lambda) = \arg \min_{\lambda \in \Lambda} \mathcal{L}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}; \lambda).$$  

BO with Gaussian Processes (GP) is the traditional choice for HPO on continuous spaces with few dimensions. SMAC3 builds on top of existing GP implementations and offers several acquisition functions, including LCB (Srinivas et al., 2010), TS (Thompson, 1933), PI and EI (Jones et al. 1998) and variants, e.g., EI per second (Snoek et al., 2012) for evaluations with different runtimes and logEI (Hutter et al., 2010) for heavy-tailed cost distributions. The pre-set of SMAC4BB follows commonly used components and comprises a Sobol sequence as initial design, a GP with 5/2-Matérn Kernel and EI as acquisition function, similar to Snoek et al. (2012).
SMAC3: A Versatile Bayesian Optimization Package for HPO

2.2 SMAC4HPO: SMAC for CASH and Structured Hyperparameter Optimization

SMAC3 can also tackle the combined algorithm selection and hyperparameter optimization problem (CASH) (Thornton et al., 2013), and searches for a well-performing $A_i$ from a set of algorithms $\mathbf{A}$ and its hyperparameters $\mathbf{\lambda} \in \mathbf{\Lambda}_i$:

$$ (A^*, \lambda^*) \in \arg \min_{A_i \in \mathbf{A}, \lambda \in \mathbf{\Lambda}_i} c(A_i, \lambda) = \arg \min_{A_i \in \mathbf{A}, \lambda \in \mathbf{\Lambda}_i} \mathcal{L}(D_{\text{train}}, D_{\text{val}}; A_i(\lambda)). $$

The hyperparameter sub-space $\mathbf{\Lambda}_i$ for an algorithm $A_i$ is only active if $A_i$ was chosen. Therefore, there is a conditional hierarchy structure between the top-level hyperparameter choosing $A_i$ and the subspace $\mathbf{\Lambda}_i$. SMAC3 models these spaces by using a random forest (Breimann, 2001), with SMAC having been the first BO approach to use this type of model (Hutter et al., 2011). SMAC3 supports multiple levels of conditionalities, e.g., one top-level hyperparameter choosing a classification algorithm and another sub-level hyperparameter an optimizer (for training NNs) enabling a second sub-level of hyperparameters. The pre-set of SMAC4HPO is based on the tuning by Lindauer et al. (2019b) and combines a Sobol sequence as initial design, a RF as surrogate model and logEI as acquisition function.

2.3 SMAC4MF: SMAC for Expensive Tasks and Automated Deep Learning

On some HPO tasks, e.g., optimizing hyperparameters and architectures for deep learning, training many models might be too expensive. Multi-fidelity optimization is a common approach, for cases where we can observe cheaper approximations of the true cost:

$$ \lambda^* \in \arg \min_{\lambda \in \mathbf{\Lambda}} c(\lambda, b_{\text{max}}) = \arg \min_{\lambda \in \mathbf{\Lambda}} \mathcal{L}(D_{\text{train}}, D_{\text{val}}; \lambda, b_{\text{max}}). $$

Here, a configuration is evaluated with a budget $b \leq b_{\text{max}}$ (e.g., number of epochs, dataset size or channels of CNN) to obtain a cheap proxy of the true cost function at $b_{\text{max}}$. SMAC3 follows the principle of BOHB (Falkner et al., 2018) in combining Hyperband (Li et al., 2018) and BO, where the surrogate model is fitted on the highest budget-level with sufficient observations. However, its RF models tend to yield much better performance than BOHB’s TPE models, see Figure 2 for illustrative examples. With the exception of the RF, the pre-set of SMAC4MF is similar to Falkner et al. (2018) and consists of a random initial design and Hyperband as intensification method.

2.4 SMAC4AC: SMAC for Algorithm Configuration

A more general view on the problem of HPO is called algorithm configuration (AC) (Hutter et al. 2009; Ansótegui et al., 2009; López-Ibáñez et al., 2016), where the goal is to determine a well-performing robust configuration across a set of problem instances $i$ from a finite set $\mathcal{I}$:

$$ \lambda^* \in \arg \min_{\lambda \in \mathbf{\Lambda}} c(\lambda) = \arg \min_{\lambda \in \mathbf{\Lambda}} \sum_{i \in \mathcal{I}} c(\lambda, i) $$.  (4)

The origin of SMAC lies in AC (Hutter et al., 2011). It inherits the ideas of aggressive racing (Hutter et al., 2009) which evaluates less promising candidate configurations on only a few instances, and to collect sufficient empirical evidence on many instances if the candidate
could become the next incumbent configuration. Furthermore, SMAC3 supports imputation of right-censored observations and a logEI to model heavy-tailed cost distributions; it also uses a customized hyperparameter configuration of the RF surrogate model for AC. SMAC successfully optimized configuration spaces with more than 300 hyperparameters for SAT solvers (Hutter et al., 2017), yielding speedups of up to several orders of magnitude. The pre-set of SMAC4AC follows Hutter et al. (2011) and combines a single default configuration as initial design, a RF as surrogate model (with different hyperparameter settings compared to SMAC4HPO), logEI as acquisition function and aggressive racing as intensification mechanism.

3. Brief Empirical Comparison

To provide an impression on the sequential performance of SMAC3, we compared it against random search (Bergstra and Bengio, 2012)—although we consider it a weak baseline (Turner et al., 2020)—Hyperband (Li et al. 2018), Dragonfly (Kandasamy et al., 2020) and BOHB (Falkner et al., 2018) on the surrogate benchmark for HPO on DNNs on the letter dataset (Falkner et al., 2018), a joint HPO+NAS benchmark (Klein and Hutter, 2019) on the Naval-Propulsion dataset and a pure NAS benchmark (Zela et al., 2020).1 As shown in Figure 2, SMAC3’s multi-fidelity approach (see Sec. 2.3) performs as well as Hyperband in the beginning, performs best in the middle, until SMAC3’s pure BO with RFs catches up in the end. For the whole time, SMAC3 consistently outperforms Dragonfly and in the later phases, also BOHB. For a larger empirical study, incl. SMAC3, we refer to Eggensperger et al. (2021).

4. Related Work

With the initial success of BO for HPO (Hutter et al., 2011; Snoek et al. 2012; Bergstra et al., 2013), there were many follow up tools in recent years (Bakshy et al., 2018; Nardi et al. 2019; Kandasamy et al., 2020; Balandat et al., 2020; Li et al., 2021; Head et al. 2021). SMAC3’s advantage lies on the one hand in the efficient use of random forests as surrogate model for higher dimensional and complex spaces, and on the other hand, in its flexibility of combining different state-of-the-art BO and intensification strategies, such as aggressive

---

1. See https://github.com/automl/HPOBench/ for details regarding the experimental setup.
racing and multi-fidelity approaches. In addition, evolutionary algorithms are also known as efficient black-box optimizers (Fortin et al., 2012; Olson et al., 2016; Loshchilov and Hutter, 2016. Rapin and Teytaud, 2018. Awad et al. 2021). Although there is the common belief that BO is particularly efficient for small budgets and evolutionary algorithms for cheap function evaluations (Feurer and Hutter, 2019), choosing the right optimizer for a given task is still an open problem. To address this, first systems have been introduced that schedule several optimizers sequentially to make use of their respective strengths (Awad et al. 2020; Lan et al., 2020; Turner et al., 2020).

5. Outlook

Although SMAC3 performs robustly on many HPO tasks, it does not exploit their landscape structure. As a next step, we plan to integrate local BO approaches (Eriksson et al., 2019). Furthermore, SMAC3 provides an easy-to-use \texttt{fmin}-API and facades, but choosing SMAC3’s own hyperparameters might still be challenging (Lindauer et al., 2019b). Therefore, we plan to add mechanisms to adaptively select SMAC3’s settings on the fly, e.g., via Bandits (Hoffman et al., 2011) or reinforcement learning (Biedenkapp et al., 2020).

Acknowledgements

Robert Bosch GmbH is acknowledged for financial support. This work has partly been supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant no. 716721. The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 37/935-1 FUGG.

References


