Meta-Learning for Symbolic Hyperparameter Defaults

Pieter Gijsbers*
University of Eindhoven
Eindhoven, Netherlands

Florian Pfisterer*
Ludwig-Maximilians-University
Munich, Germany

Jan N. van Rijn
LIACS, Leiden University
Leiden, Netherlands

Bernd Bischl
Ludwig-Maximilians-University
Munich, Germany

Joaquin Vanschoren
University of Eindhoven
Eindhoven, Netherlands

ABSTRACT

Hyperparameter optimization in machine learning (ML) deals with the problem of empirically learning an optimal algorithm configuration from data, usually formulated as a black-box optimization problem. In this work, we propose a zero-shot method to meta-learn symbolic default hyperparameter configurations that are expressed in terms of the properties of the dataset. This enables a much faster, but still data-dependent, configuration of the ML algorithm, compared to standard hyperparameter optimization approaches. In the past, symbolic and static default values have usually been obtained as hand-crafted heuristics. We propose an approach of learning such symbolic configurations as formulas of dataset properties from a large set of prior evaluations on multiple datasets by optimizing over a grammar of expressions using an evolutionary algorithm. We evaluate our method on surrogate empirical performance models as well as on real data across 6 ML algorithms on more than 100 datasets and demonstrate that our method indeed finds viable symbolic defaults.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**.

KEYWORDS

Hyperparameter Optimization, Metalearning

ACM Reference Format:
Pieter Gijsbers, Florian Pfisterer, Jan N. van Rijn, Bernd Bischl, and Joaquin Vanschoren. 2021. Meta-Learning for Symbolic Hyperparameter Defaults. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459532

1 INTRODUCTION & RELATED WORK

The performance of most machine learning (ML) algorithms is greatly influenced by their hyperparameter settings. While various methods exist to automatically optimize them, the additional complexity and effort cause many practitioners to forgo optimization. Defaults provide a fallback but are often static and do not take properties of the dataset into account, even though the success of tuning hyperparameters suggests values should be based on properties of the data. Contrary to static defaults, *symbolic defaults* should be a function of the meta-features (dataset characteristics) of the dataset, such as the number of features. A well-known example for a symbolic default is the random forest algorithm's default $mtry = \sqrt{p}$ for the number of features sampled in each split. In this paper we explore how such formulas can be obtained in a principled, empirical manner, especially when multiple hyperparameters interact, and have to be considered simultaneously[1]. We propose to learn such symbolic default configurations by optimizing over a grammar of potential expressions, in a manner similar to symbolic regression [3] using evolutionary algorithms. We validate our approach across a variety of state-of-the-art ML algorithms and propose default candidates for use by practitioners. The proposed approach is general and can be used for any algorithm as long as their performance is empirically measurable on instances in a similar manner.

*Both authors contributed equally to the paper

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '21 Companion, July 10–14, 2021, Lille, France
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3459532

2 METHOD

A symbolic configuration is a set of functions, one for each hyperparameter of the algorithm. Each function maps the meta-features of the given dataset to a value for a hyperparameter, e.g. $mtry = \sqrt{p}$. Note that it is not needed for any or all meta-features to be used in the mapping, i.e. the function may be constant (static) or only use few meta-features. We want to learn a symbolic default configuration $\lambda(.)$ for algorithm $\mathcal{A}$ that minimizes the expected risk induced by the model produced by $\mathcal{A}_{\boldsymbol{\lambda}}$ across datasets.

We define a context-free grammar of transformations, which define the space of potential expressions for all functions $\lambda(.)$. We select a small set of simple dataset characteristics for use in formulas, e.g. number of observations, features, or missing values. Given K datasets, a risk function $R(\lambda(.), \mathcal{D}_i)$ that denotes the risk induced by the model learnt using algorithm $\mathcal{A}$ with symbolic configuration $\lambda(.)$ on dataset $\mathcal{D}_i$, we can formulate a global objective to minimize: $R_{\mathcal{D}}(\lambda(.)) = \frac{1}{K}\sum_{i=1}^{K} R(\lambda(.), \mathcal{D}_i)$. As estimating $R_{\mathcal{D}}(\lambda(.))$ empirically using cross-validation (CV) is costly in practice, we instead employ *surrogate models* that approximate $R_{\mathcal{D}}(\lambda(.))$.

Meta-learning. To create surrogate models, we collect data about the performance of randomly sampled constant configurations. These configurations are evaluated across all datasets using 10-fold CV. For each dataset we train a random forest model mapping hyperparameter configurations to expected performance. We can then approximate the average risk of $\lambda(.)$ by querying each surrogate

[1]Code available at https://github.com/PGijsbers/symbolicdefaults

model after first computing the real configuration values using the dataset's characteristics.

Optimization. The problem we aim to solve requires optimization over a space of mathematical expressions. Several options to achieve this exist [4, 5]. We opt for a tree representation of individuals, where nodes correspond to operations and leaves to terminal symbols or numeric constants, and optimize this via genetic programming [3]. We differentiate between real-valued and integer-valued terminal symbols to account for the difference in algorithm hyperparameters. We use a $\mu + \lambda$ algorithm to evolve candidate solutions via crossover and mutation. We jointly optimize performance of solutions while preferring formulas with smaller structural depth using NSGA-II selection [1] without explicitly limiting length of the expressions.

3 RESULTS

We investigate symbolic defaults for 6 ML algorithms using a large set of meta-data, containing evaluations of over a hundred datasets available from OpenML [6]. We optimize the average logistic loss (normalized to [0,1]), but our methodology trivially extends to other performance measures. We evaluate using a leave-one-dataset-out strategy to obtain symbolic defaults. As baselines, we employ *random search* and *1-nearest neighbour*, an approach that selects the configuration that worked best on the most similar dataset, comparable to warm-starting in auto-sklearn [2].

Table 1 shows the mean and standard deviation of the normalized log-loss for each algorithm across all tasks, as predicted by surrogate models. The symbolic and constant columns denote the performance of defaults found with our approach including and excluding symbolic terminals respectively. The package column shows the best result obtained from either the scikit-learn or mlr default, and the last column denotes the best-found performance sampling 8 random real-world scores on the task for the algorithm. Note that the best rank can deviate from the best average performance.

The default mean rank is never significantly lower than that of other approaches, but in some cases, it is significantly higher. The only implementation default which does not score a significantly lower mean rank than our approach is the default for SVM, which has carefully hand-crafted defaults. For more nuance about the performance differences per dataset, Figure 1 shows the predicted normalized log-loss per dataset for SVM configurations obtained by different methods.

We further show the non-normalized log-loss per dataset obtained with 10-fold CV experiments in Figure 2. The median performance for symbolic defaults is slightly lower, though overall very similar performance is achieved by this automatically obtained symbolic default to the hand-crafted one in scikit-learn, or per-dataset recommendations from 1NN.

algorithm	symbolic	constant	package	opt. RS 8
glmnet	**0.917(.168)**	**0.928(.158)**	0.857(.154)	0.906(.080)
knn	**0.954(.148)**	0.947(.156)	0.879(.137)	**0.995(.009)**
rf	**0.946(.087)**	0.951(.074)	0.933(.085)	0.945(.078)
rpart	**0.922(.112)**	**0.925(.093)**	0.792(.141)	**0.932(.082)**
svm	**0.889(.178)**	**0.860(.207)**	**0.882(.190)**	**0.925(.084)**
xgboost	**0.995(.011)**	**0.995(.011)**	0.925(.125)	0.978(.043)

Table 1: Mean normalized log-loss (standard deviation) across all tasks with baselines. Boldface values indicate the average rank was not significantly worse than the best (underlined) of the four settings.

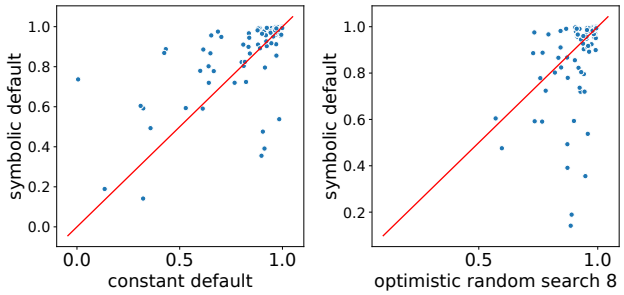

Figure 1: Normalized log-loss comparison of symbolic defaults to constant defaults (left) and budget 8 random search (right).

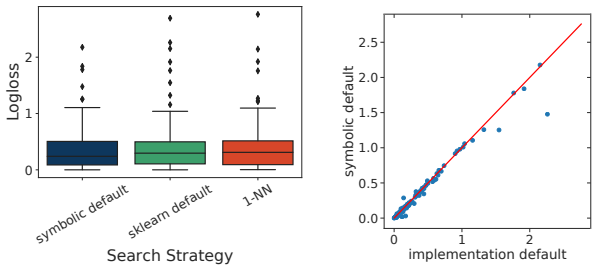

Figure 2: Comparison of 1NN, symbolic, and implementation default using log-loss across all datasets performed on real data.

ACKNOWLEDGMENTS

This material is based upon work supported by the Data Driven Discovery of Models (D3M) program run by DARPA and the Air Force Research Laboratory, and by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content.

REFERENCES

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
[2] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2962–2970.
[3] John R Koza. 1994. Genetic programming as a means for programming computers by natural selection. *Statistics and computing* 4, 2 (1994), 87–112.
[4] M. O'Neill and C. Ryan. 2001. Grammatical evolution. *IEEE Transactions on Evolutionary Computation* 5, 4 (Aug 2001), 349–358.
[5] Jan N. van Rijn, Florian Pfisterer, Janek Thomas, Andreas Müller, Bernd Bischl, and Joaquin Vanschoren. 2018. Meta learning for defaults : symbolic defaults. In *Workshop on Meta-Learning @ NeurIPS2018*.
[6] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter* 15, 2 (2014), 49–60.