Automatic Exploration of Machine Learning Experiments on OpenML

by Daniel Kühn*, Philipp Probst*, Janek Thomas and Bernd Bischl

May 2, 2018

Abstract

Understanding the influence of hyperparameters on the performance of a machine learning algorithm is an important part of finding a well performing and adequately tuned algorithm for a given dataset. As to date no dataset exists to support this required understanding for state of the art algorithms like gradient boosting or random forest, this paper presents a large and open dataset for this problem (see [Probst and Kühn, 2018] to access the dataset). The dataset contains the performance results (AUC, accuracy and Brier score) of six different machine learning algorithms, the used hyperparameters that were set by a random search, runtimes and meta-data for each dataset. It can be used for meta-learning, finding good defaults, measuring the tunability of algorithms and hyperparameters, benchmarking of algorithms and other tasks mainly related to hyperparameters.

1 Introduction

When applying machine learning on real world datasets, users have to choose from a large selection of different machine learning algorithms with many of these algorithms offering a set of hyperparameters, which can be specified by the user and can have a significant influence on the performance of the algorithm. Since there is no free lunch in algorithm selection and one can not expect one algorithm to outperform all the others [Wolpert, 2001], a crucial question practitioners have to face on a daily basis therefore is the selection of the "right" algorithm with the "right" hyperparameters for a given dataset. This problem is not easy to solve, because choices are many, the evaluation of a single machine learning run usally is computationally expensive and the hyperparameter-space is complex [Claesen and Moor, 2015]. A usual approach is to run a tuning algorithm like bayesian optimization to find the best hyperparameter setting. While showing promising results, bayesian optimization and other hyperparameter optimization techniques require a large overhead.

Meta-learning tries to decrease this overhead [Feurer et al., 2015b], by using information of previous algorithm runs on other datasets. A requirement for this, is to have a meta-learning dataset, that contains the information of previous runs. With this paper we provide a open accesible dataset that contains information of previous runs of six different machine learning algorithms on 38 classification datasets. As large datasets like Imagenet [Deng et al., 2009] have shown to improve the progress of machine learning, we hope to support the development of new meta-learning algorithms with this dataset.

We first describe how we created such a dataset by executing random machine learning experiments and storing the results on OpenML [Vanschoren et al., 2013], an open source database for machine learning problems. Then the possibilities of accessing this dataset are shortly presented. Finally we briefly discuss potential usage of the dataset.

2 Related literature

There is quite a few literature that use meta-learning approaches. A successful example is auto-sklearn [Feurer et al., 2015a] which is a automatic machine learning algorithm that uses information of previous experiments on other datasets. They use a meta-learning approach for warmstarting their tuning algorithm.

On the other hand, to the best of our knowledge there are currently no papers and projects that provide data for meta-learning approaches, like we do in this paper. The only relevant paper we could found is the paper by Reif [2012] which presents a dataset for meta-learning based on 83 datasets, six classification algorithms, and 49 meta-features. The dataset is not anymore available on the university wegpage as the researcher does not work there anymore. We tried to circumvent this potential problem by uploading our results on figshare which is meant to be a durable and open platform for storing and sharing data.

3 Creating the dataset

In this section we describe in detail how we create the dataset.

3.1 Algorithms

To create the dataset six supervised machine learning algorithms were used in R with predefined ranges for their relevant hyperparameters and they were run on 38 classification tasks. The following frequently used algorithms were chosen: elastic net (`glmnet`), decision tree (`rpart`), k-nearest neighbors (`kknn`), support vector machines (`svm`), random forest (`ranger`) and gradient boosting (`xgboost`). These algorithms cover a broad range of approaches to machine learning. For each algorithm the available hyperparameters were explored in a predefined range (see table 1). Values of some of these ranges were transformed by the function found in column *trafo* to explore the range in a non-uniform

manner. This is useful, if, for example, minor changes in a hyperparameter are not expected to have a significant impact on the performance of an algorithm.

algorithm	hyperparameter	type	lower	upper	trafo
glmnet	alpha	numeric	0	1	-
	lambda	numeric	-10	10	2^x
rpart	cp	numeric	0	1	-
	maxdepth	integer	1	30	-
	minbucket	integer	1	60	-
	minsplit	integer	1	60	-
kknn	k	integer	1	30	-
svm	kernel	discrete	-	-	-
	cost	numeric	-10	10	2^x
	gamma	numeric	-10	10	2^x
	degree	integer	2	5	-
ranger	num.trees	integer	1	2000	-
	replace	logical	-	-	-
	sample.fraction	numeric	0	1	-
	mtry	numeric	0	1	$x \cdot p$
	respect.unordered.factors	logical	-	-	-
	min.node.size	numeric	0	1	n^x
xgboost	nrounds	integer	1	5000	-
	eta	numeric	-10	0	2^x
	subsample	numeric	0	1	-
	booster	discrete	-	-	-
	max_depth	integer	1	15	-
	min_child_weight	numeric	0	7	2^x
	colsample_bytree	numeric	0	1	-
	colsample_bylevel	numeric	0	1	-
	lambda	numeric	-10	10	2^x
	alpha	numeric	-10	10	2^x

Table 1: Hyperparameters of the algorithms. p refers to the number of variables and n to the number of observations.

3.2 Datasets

These algorithms are run on a subset of the OpenML100 Benchmark suite [Bischl et al., 2017], which consists of 100 classification datasets carefully curated from the thousands of datasets available on OpenML. We only include datasets without missing data and with binary outcome resulting in 38 datasets. The datasets with their specific characteristics can be found in table 2.

Data_id	Name	nObs	nFeat	majPerc	numFeat	catFeat
3	kr-vs-kp	3196	37	0.52	0	37
31	credit-g	1000	21	0.70	7	14
37	diabetes	768	9	0.65	8	1
44	spambase	4601	58	0.61	57	1
50	tic-tac-toe	958	10	0.65	0	10
151	electricity	45312	9	0.58	7	2
312	scene	2407	300	0.82	294	6
333	monks-problems-1	556	7	0.50	0	7
334	monks-problems-2	601	7	0.66	0	7
335	monks-problems-3	554	7	0.52	0	7
1036	sylva_agnostic	14395	217	0.94	216	1
1038	gina_agnostic	3468	971	0.51	970	1
1046	mozilla4	15545	6	0.67	5	1
1049	pc4	1458	38	0.88	37	1
1050	pc3	1563	38	0.90	37	1
1063	kc2	522	22	0.80	21	1
1067	kc1	2109	22	0.85	21	1
1068	pc1	1109	22	0.93	21	1
1120	MagicTelescope	19020	12	0.65	11	1
1176	Internet-Advertisements	3279	1559	0.86	1558	1
1220	Click_prediction_small	39948	12	0.83	11	1
1461	bank-marketing	45211	17	0.88	7	10
1462	banknote-authentication	1372	5	0.56	4	1
1464	blood-transfusion-service-center	748	5	0.76	4	1
1467	climate-model-simulation-crashes	540	21	0.91	20	1
1471	eeg-eye-state	14980	15	0.55	14	1
1479	hill-valley	1212	101	0.50	100	1
1480	ilpd	583	11	0.71	9	2
1485	madelon	2600	501	0.50	500	1
1486	nomao	34465	119	0.71	89	30
1487	ozone-level-8hr	2534	73	0.94	72	1
1489	phoneme	5404	6	0.71	5	1
1504	steel-plates-fault	1941	34	0.65	33	1
1510	wdbc	569	31	0.63	30	1
1570	wilt	4839	6	0.95	5	1
4134	Bioresponse	3751	1777	0.54	1776	1
4135	Amazon_employee_access	32769	10	0.94	0	10
4534	PhishingWebsites	11055	31	0.56	0	31

Table 2: Included datasets with meta-data. *nObs* are the number of observations, *nFeat* the number of Features, *majPerc* the percentage of observations with the most common class, *numFeat* the number of numeric features and *catFeat* the number of categorical features.

3.3 Execution of the bot

Following the search paradigm of random search the bot iteratively executes several steps:

1. Randomly draw one of the six algorithms
2. Randomly draw a hyperparameter setting of the chosen algorithm
3. Randomly draw one of the 38 binary classification benchmark datasets
4. Download the dataset from OpenML
5. Benchmark the specified algorithm on the specified dataset with 10-fold cross-validation (the standardized splits for the cross-validation are specified by OpenML)

6. Upload the benchmark results with hyperparameters, performance measures and time measurements to OpenML with the tag `mlrRandomBot` used for identification

An advantage of using random search, instead of for example grid search is, that further experiments can be easily added to the existing ones. For example, one could easily increase the hyperparameter space for a specific algorithm or add a complete new algorithm by just adding new experiments to the existing ones.

The code for the bot can be found on GitHub (https://GitHub.com/ja-thomas/OMLbots). Further algorithms can be easily added by adding an algorithm with specific hyperparameters in the `R/botSetLearnerParamPairs.R` file of the GitHub repository. The main function `runBot` executes the bot with specific settings (number of experiments, temporary file, sample configurations, etc.).

The R packages `mlr` [Bischl et al., 2016] and `OpenML` [Casalicchio et al., 2017] were used for the whole process.

3.4 Extraction of results

After having run more than 6 million benchmark experiments the results of the bot are downloaded from OpenML. Because of technical reasons on one dataset (data.id = 4135) all algorithms except of `rpart` and `ranger` provide errors, so we exclude it and 38 datasets are left.

For each of the algorithms we only take 500000 experiments to obtain a dataset which is balanced regarding the experiment runs. They are chosen by the following procedure: for each algorithm, a threshold B is set (see below) and, if the number of results for a dataset exceeds B, we draw randomly B of the results obtained for this algorithm and this dataset. For each algorithm, the threshold value B is chosen for each algorithm separately to exactly obtain 500000 results for each algorithm.

For `kknn` we only executed 30 experiments per dataset because this number of experiments is high enough to cover the hyperparameter space (that only consists of the parameter k for $k \in \{1, ..., 30\}$) appropriately, resulting in 1140 experiments. In total this results in around 2.5 million experiments.

The distribution of the runs on the datasets and algorithms can be seen in table 3.

4 Access to the benchmark results

The results of the benchmarks can be accessed in different ways:

- The easiest way to access them is to go to the figshare repository [Probst and Kühn, 2018] and download the `.csv` files or the `.RData` file.
- Alternatively the code for the extraction of the data from the nightly database snapshot of OpenML can be found here: https://github.com/j

Data_id	glmnet	rpart	kknn	svm	ranger	xgboost	Total
3	15547	14633	30	19644	15139	16867	81860
31	15547	14633	30	19644	15139	16867	81860
37	15546	14633	30	15985	15139	16866	78199
44	15547	14633	30	19644	15139	16867	81860
50	15547	14633	30	19644	15139	16866	81859
151	15547	14632	30	2384	12517	16866	61976
312	6613	13455	30	18740	12985	15886	67709
333	15546	14632	30	19644	15139	16867	81858
334	15547	14633	30	19644	14492	16867	81213
335	15547	14633	30	15123	15139	10002	70474
1036	14937	14633	30	2338	7397	2581	41916
1038	15547	5151	30	5716	4827	1370	32641
1046	6466	14633	30	10121	3788	16867	51905
1049	15547	14633	30	5422	8842	11812	56286
1050	7423	14632	30	12064	15139	4453	53741
1063	15547	14633	30	19644	11357	13758	74969
1067	15547	14633	30	19644	7914	16866	74634
1068	15546	14632	30	10229	7386	16866	64689
1120	15546	14633	30	13893	8173	16866	69141
1176	15531	7477	30	3908	9760	8143	44849
1220	13005	14632	30	14451	15140	13047	70305
1461	6970	14073	30	2678	14323	2215	40289
1462	8955	14633	30	6320	15139	16867	61944
1464	15547	14632	30	19644	15139	16867	81859
1467	15547	14633	30	4441	15139	16866	66656
1471	15547	14633	30	9725	13523	16866	70324
1479	15546	14633	30	19644	15140	16867	81860
1480	15024	14633	30	19644	15139	16254	80724
1485	8247	10923	30	10334	15139	9237	53910
1486	3866	11389	30	1490	15139	5813	37727
1487	15547	6005	30	19644	15139	11194	67559
1489	15547	14633	30	17298	15139	16867	79514
1504	15547	14632	30	19644	15139	16867	81859
1510	15547	14633	30	19644	15140	16867	81861
1570	15547	14633	30	19644	15139	16867	81860
4134	15546	14632	30	19644	15139	16867	81858
4135	1493	3947	30	560	14516	2222	22768
4534	2801	3231	30	2476	15139	947	24624
Total	500000	500000	1140	500000	500000	500000	2501140

Table 3: Number of results by dataset and algorithm

a-thomas/OMLbots/blob/master/snapshot_database/database_extraction.R

5 Potential usage of the results and discussion

The results can be used to discover effects of the hyperparameters on performances of the different algorithms on different datasets.

This can be used to:

- Find good defaults for the algorithms that work well on many datasets
- Measure differences between the algorithms
- Optimize tuning algorithms:
  - Measure the tunability of algorithms and find out which parameters should be tuned [see Probst et al., 2018]

  - Use the results to get priors for tuning algorithms - in which regions of the hyperparameter space should be searched with higher probability?

- Meta-Learning: Train models that based on dataset characteristics and possibly time limitations propose hyperparameter settings that perform good on a specific dataset

A potential weakness of this dataset is that the dimension of the hyperparameters for, for example, `xgboost` is very high and the number of experiments is not high enough to explore the regions appropriately, especially regions where very high performance can be achieved. This could potentially be improved by using a tuning algorithm and by adding the results of the tuning steps to the existing database.

References

B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in r. *Journal of Machine Learning Research*, 17(170):1–5, 2016. URL http://jmlr.org/papers/v17/15-066.html.

B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren. OpenML Benchmarking Suites and the OpenML100. *ArXiv e-prints*, Aug. 2017.

G. Casalicchio, J. Bossek, M. Lang, D. Kirchhoff, P. Kerschke, B. Hofner, H. Seibold, J. Vanschoren, and B. Bischl. OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 32(3):1–15, 2017. doi: 10.1007/s00180-017-0742-2.

M. Claesen and B. D. Moor. Hyperparameter search in machine learning. *MIC 2015: The XI Metaheuristics International Conference*, 2015.

J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015a.

M. Feurer, J. T. Springenberg, and F. Hutter. Initializing bayesian hyperparameter optimization via meta-learning. 2015b.

P. Probst and D. Kühn. OpenML R Bot Benchmark Data (final subset). 2 2018. doi: 10.6084/m9.figshare.5882230.v1. URL https://figshare.com/articles/OpenML_R_Bot_Benchmark_Data_final_subset_/5882230.

P. Probst, B. Bischl, and A.-L. Boulesteix. Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *ArXiv preprints arXiv:1703.03373*, 2018.

M. Reif. A comprehensive dataset for evaluating approaches of various meta-learning tasks. In *ICPRAM*, 2012.

J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

D. H. Wolpert. The supervised learning no-free-lunch theorems. 2001.