Automatic Exploration of Machine Learning Experiments on OpenML

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

February 5, 2018

Abstract

For any machine learning problem it is essential to have the right dataset. Meta-learning is a branch of machine learning, that focuses on hyperparameter optimization of machine learning algorithms. To date no large, open datset exists to support and benchmark meta-learning approaches. With this work we provide one such dataset for six different machine learning algorithms, which not only can be used for meta-learning but also for the better understanding of the explored hyperparameter space of these algorithms.

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. Several approaches like meta-learning or bayesian optimization exist for solving this issue. While showing promising results, bayesian optimization and other hyperparameter optimization techniques require a large overhead. Meta-learning has shown to decrease this overhead [Feurer et al., 2015], but no large and open source meta-learning datasets exist to date. In this paper we 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. 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.

2 Description of the bot

We specified six machine learning algorithms with their specific hyperparameters. We choosed the six supervised learning algorithms elastic net (`glmnet` package), decision tree (`rpart`), k-nearest neighbors (`kknn`), support vector machines (`svm`), random forest (`ranger`) and gradient boosting (`xgboost`). They represent algorithms that are used very often and cover a broad range of different types of algorithms. The algorithms with their specific hyperparameters and hyperparameter constraints can be found in table 1. Some hyperparameters are transformed after they have been drawn, for example to get more results in regions where we expect more interesting results.

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	-
	respect.unordered.factors	logical	-	-	-
	min.node.size	numeric	0	1	-
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

Furthermore we choosed the classification tasks (datasets) from the OpenML100 Benchmarksuite [Bischl et al., 2017] as benchmark datasets, only including datasets without missing data and with binary outcome resulting in 76 datasets.

The datasets with their specific characteristics can be found in table **??**.

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 benchmark datasets
4. Download the dataset from OpenML
5. Benchmark the specified algorithm on the specified dataset with 10-fold cross-validation
6. Upload the benchmark results with time measurements to OpenML with the identification tag `mlrRandomBot`

The code for the bot can be found on github (`https://github.com/ja-thomas/OMLbots`), the R packages `mlr` [Bischl et al., 2016] and `OpenML` [Casalicchio et al., 2017] were used for the whole process.

In total more than 6 millions (XXX) runs of the random bot on different server platforms (OH NO!) like Azure were executed.

3 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, 2017] and download the .csv files.

4 Validity of the results

DISTRIBUTION OF HYPERPARAMETERS AND DATASETS

5 Potential usage of the results

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

  - 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

6 Literature

Other related projects and papers (e.g. AutoWeka)

7 to-Do

- Think about the title
- Weakness: Dimension is too high, e.g. for xgboost. The best regions are not explored enough. Maybe add later.

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*, Jun 2017. doi: 10.1007/s00180-017-0742-2. URL https://doi.org/10.1007/s00180-017-0742-2.

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

P. Probst and D. Kühn. OpenML R Bot Benchmark Data. 12 2017. doi: 10.6084/m9.figshare.5727073.v1. URL https://figshare.com/articles/OpenML_R_Bot_Benchmark_Data/5727073.

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.

Task_id	Name	nObs	nFeat	glmnet	rpart	kknn	svm	ranger	xgboost
3	kr-vs-kp	3196	37	15563	15432	2047	10982	64274	20832
31	credit-g	1000	21	62070	82169	2470	61029	109493	69675
37	diabetes	768	9	19800	15441	1308	12003	37682	14885
43	spambase	4601	58	12482	24317	1129	19334	13714	4331
49	tic-tac-toe	958	10	3164	1456	568	10469	50808	17988
219	electricity	45312	9	17493	12983	1493	1477	8012	17604
3485	scene	2407	300	6613	13455	2215	18740	16802	15886
3492	monks-problems-1	556	7	19915	15126	2167	7201	24045	31420
3493	monks-problems-2	601	7	21331	13707	1119	32236	17148	27746
3494	monks-problems-3	554	7	11687	16409	1415	9811	13310	4806
3889	sylva_agnostic	14395	217	14937	15823	1302	2338	9412	2581
3891	gina_agnostic	3468	971	16228	5151	3973	5716	5509	1370
3896	ada_agnostic	4562	49	6466	21345	740	10121	17577	21558
3899	mozilla4	15545	6	18326	17471	17178	5422	23323	11812
3902	pc4	1458	38	7423	19411	2375	12064	41952	4453
3903	pc3	1563	38	41542	20906	697	23641	20270	13758
3913	kc2	522	22	24857	35387	2503	29320	37325	19032
3917	kc1	2109	22	19315	30099	2031	10229	43443	27240
3918	pc1	1109	22	21069	16882	2803	13893	43430	24097
3954	MagicTelescope	19020	12	15531	7477	2433	3908	9671	8143
6566	Internet-Advertisements	3279	1559	3562	1592	1591	4712	5055	6085
7295	Click_prediction_small	39948	12	18476	6265	1992	430	8308	2759
9889	wilt	4839	6	13586	5876	1257	15433	17879	2033
9910	Bioresponse	3751	1777	497	2484	193	91	3358	745
9911	Amazon_employee_access	32769	10	0	1006	13	0	8422	0
9914	wilt	4839	6	38850	51914	1893	61853	79466	48238
9946	wdbc	569	31	27993	18335	1218	31745	18776	25201
9952	phoneme	5404	6	29017	12333	2921	4851	20338	21043
9957	qsar-biodeg	1055	42	27983	19397	3621	16272	46830	10744
9967	steel-plates-fault	1941	34	23691	11875	1344	9819	31453	50465
9970	hill-valley	1212	101	5541	18159	990	17450	23708	8674
9971	ilpd	583	11	11994	18393	2533	27338	34297	6448
9976	madelon	2600	501	3114	8444	979	5461	14828	4695
9977	nomao	34465	119	510	5358	926	926	5303	3913
9978	ozone-level-8hr	2534	73	22266	17	3571	11294	28409	5992
9980	climate-model-simulation-crashes	540	21	34417	15388	1641	3941	15750	11723
9983	eeg-eye-state	14980	15	39043	22740	994	987	17154	1882
10093	banknote-authentication	1372	5	2819	21734	257	843	17098	24979
10101	blood-transfusion-service-center	748	5	68190	70024	4357	76402	77018	63140
14951	eeg-eye-state	14980	15	5626	13777	2242	8738	12208	21240
14952	PhishingWebsites	11055	31	806	1085	1001	987	7099	0
14965	bank-marketing	45211	17	4473	5878	2430	1524	7666	1484
14966	Bioresponse	3751	1777	499	970	1367	56	3359	495
14971	Click_prediction_small	39948	12	2403	4988	2639	644	5011	2306
34536	Internet-Advertisements	3279	1559	7939	13158	1941	9081	10976	3744
34537	PhishingWebsites	11055	31	1995	2146	948	1489	9167	947
34539	Amazon_employee_access	32769	10	1	2046	15	0	8347	0
145677	Bioresponse	3751	1777	497	493	1303	413	3201	982
145804	tic-tac-toe	958	10	52740	46496	776	59439	59811	46885
145833	bank-marketing	45211	17	2497	8195	1789	1154	7023	731
145834	banknote-authentication	1372	5	6136	14482	444	5477	11698	3804
145836	blood-transfusion-service-center	748	5	9913	10873	525	19718	29205	6713
145839	climate-model-simulation-crashes	540	21	10367	20437	1633	500	17126	8879
145847	hill-valley	1212	101	30887	9991	2729	19954	6867	19667
145848	ilpd	583	11	3030	3441	2879	680	7584	9806
145853	madelon	2600	501	5133	2479	512	4873	18977	4542
145854	nomao	34465	119	3356	6031	3273	564	15002	1900
145855	ozone-level-8hr	2534	73	15427	5988	385	13048	48135	5202
145857	phoneme	5404	6	31676	8966	1108	12447	15998	30277
145862	qsar-biodeg	1055	42	13787	10324	577	11428	46097	30877
145872	steel-plates-fault	1941	34	18318	13451	562	41612	33402	22344
145878	wdbc	569	31	11494	10815	1107	11031	19263	19672
145953	kr-vs-kp	3196	37	47925	9945	797	10202	6521	18684
145972	credit-g	1000	21	21949	10291	529	10343	18538	18276
145976	diabetes	768	9	11967	996	525	3982	20917	26799
145979	spambase	4601	58	14969	2561	1498	9857	8977	18534
146012	electricity	45312	9	6842	5971	899	907	5559	4095
146064	monks-problems-1	556	7	44832	44840	3655	46225	51274	46066
146065	monks-problems-2	601	7	39221	51082	560	55628	68403	44439
146066	monks-problems-3	554	7	3991	1490	3533	5312	11171	5196
146085	Internet-Advertisements	3279	1559	1504	987	525	658	4054	3218

Table 2: Included datasets, number of observations (nObs), number of features (nFeat) and number of runs of each algorithm