Automatic Exploration of Machine Learning Experiments on OpenML

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

January 15, 2018

1 Abstract

2 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 Wolpert2001, 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.

In this paper we describe how experiments with the random bot were executed on OpenML OpenML2013.

3 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 **??**. Some hyperparameters are transformed after they have been drawn, for example to get more results in regions where we expect more interesting results.

Furthermore we choosed the classification tasks (datasets) from the OpenML100 Benchmarksuite Bischl2017 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 XXX.

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

1. Randomly draw one of the six algorithms
2. Randomly draw a hyperparameter setting of the chosen algorithms
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` Bischl2016 and `OpenML` Casalicchio2017 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.

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 Probst2017 and download the .csv files.

5 Validity of the results

DISTRIBUTION OF HYPERPARAMETERS AND DATASETS

6 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

7 Literature

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

8 to-Do

- Think about the title
- Put the package on CRAN
- Put the database on e.g. figshare