AutoML Challenge 2015

Isabelle Guyon
guyon@chalearn.org
Berkeley, CA, USA

Kristin Bennett
bennek@rpi.edu
RPI, USA

Marc Boullé
marc.boulle@orange-ftgroup.com
Orange, France

Gavin Cawley
G.Cawley@uea.ac.uk
UEA, UK

Hugo Jair Escalante
hugo.jair@gmail.com
INAOE, Mexico

Sergio Escalera
sergio@maia.ub.es
CVC, UAB & UB, Spain

Tin Kam Ho
tho@us.ibm.com
IBM, USA

Núria Macià
macia.nuria@gmail.com
Andorra

Bisakha Ray
bisakha.ray@nyumc.org
NYU, USA

Mehreen Saeed
mehreen.mehreen@gmail.com
FAST, Pakistan

Alexander Statnikov
statnikov@gmail.com
NYU, USA

Evelyne Viegas
evelynev@microsoft.com
Microsoft, USA

The *AutoML* Challenge is designed to promote research on reducing or removing the need for human interaction in applying machine learning (ML) to practical problems. This refers to all aspects of automating the ML process beyond model selection, hyper-parameter optimization, and model search. Automation is desired for data loading and formatting, detection and handling of skewed data and missing values, selection of learning representation and feature extraction, matching algorithms to problems, acquisition of new data (active learning), creation of appropriately sized and stratified training, validation, and test sets, selection of algorithms that satisfy resource constraints at training and run time, the ability to generate and reuse workflows, meta-learning and learning transfer, and explicative reports. Such automation is crucial for both robots and lifelong autonomous ML. In [3] details of the design of the AutoML challenge[1] are presented. In this paper we report some first results from this year's competition.

The challenge focuses on supervised learning in ML and, in particular, solving classification and regression problems, without any further human intervention, within given constraints. Data present themselves as input-output pairs that are identically and independently distributed. The models used are limited to fixed-length vectorial representations (no time series prediction). Text, speech, and video processing tasks included in the challenge are not presented in their native data representations; datasets have been preprocessed in suitable fixed-length vectorial representations. The difficulty of the challenge lies on the data complexity (class imbalance, sparsity, missing values, categorical variables). The testbed is composed of data from a wide variety of domains. Although there exist ML toolkits that can tackle all these problems, it still requires considerable human effort to find, for a given dataset, task, evaluation metric, and available computational time, the methods and hyper-parameter settings that maximize performance. The participant's challenge is to create the *perfect black box* that removes human interaction. The datasets used in the challenge present the following range of difficulty, which requires demanding hyper-parameter choices:

[1] http://codalab.org/AutoML

Table 1: Round 1 data: results of phase AutoML 1 on test set.

Participant	<Rank>	DS 1	DS 2	DS 3	DS 4	DS 5	Duration
aad.freiburg	2.80	0.5096 (1)	0.6059 (4)	0.6270 (3)	0.5802 (1)	0.8778 (5)	5988
jrl	443.80	0.4856 (2)	0.6276 (1)	0.5993 (5)	0.5292 (3)	0.8711 (8)	5987
tadej	4.20	0.4309 (9)	0.6207 (3)	0.7468 (1)	0.5549 (2)	0.8749 (6)	2728
reference	5.20	0.4568 (6)	0.5524 (8)	0.5324 (6)	0.5244 (4)	0.8934 (2)	4366

Table 2: Round 1 data: results of phase Tweakathon 1 on validation data.

Participant	<Rank>	DS 1	DS 2	DS 3	DS 4	DS 5
sjahandideh	2.40	0.5588 (1)	0.6958 (1)	0.8593 (1)	0.7067 (4)	0.9192 (5)
ideal.intel.analytics	2.60	0.5564 (2)	0.6844 (2)	0.8453 (5)	0.7376 (2)	0.9254 (2)
aad.freiburg	3.20	0.5276 (5)	0.6768 (3)	0.8553 (4)	0.8182 (1)	0.9251 (3)
reference	32.40	0.4628 (27)	0.5627 (29)	0.5276 (58)	0.5163 (31)	0.8895 (17)

Data distributions. Different intrinsic/geometrical complexity.

Tasks. Regression, binary classification, multi-class classification, and multi-label classification.

Scoring metrics. See `http://codalab.org/AutoML` for details.

Class imbalance. Balanced vs. unbalanced class proportions.

Sparsity. Full vs. sparse matrices.

Missing values. Presence vs. absence of missing values.

Categorical variables. Presence vs. absence of categorical features.

Irrelevant variables. Presence vs. absence of additional irrelevant data.

Number of training examples (P_{tr}). Small vs. large number of training examples.

Number of features (N). Small vs. large number of features.

Aspect ratio of the training data matrix (P_{tr}/N). $P_{tr} \gg N, P_{tr} \ = N$, or $P_{tr} \ll N$.

The challenge is run in multiple phases grouped in six rounds. Round 0 (Preparation) is a practice round with publicly available datasets, which is followed by five rounds of progressive difficulty (Novice, Intermediate, Advanced, Expert, and Master). Each round introduces 5 new datasets. Except for rounds 0 and 5, all rounds include three phases: AutoML, Tweakathon, and Final. Submissions are made in Tweakathon phases only, and immediate performance feed-back is provided on the leaderboard for each dataset using a development test set (validation set). The results of the latest submission are shown on the leaderboard and such submission automatically migrates to the next phases: the Final phase, which evaluates performance on the test set, and the AutoML phase of the following round. Submitting code makes it possible to participate in the next AutoML phase and in every subsequent phases without new submissions. However, participating in previous rounds is not a prerequisite for entering new rounds.

The challenge uses the Codalab platform for submissions. Codalab provides computational resources shared by all participants. To ensure fairness, when a code submission is evaluated, its execution time is limited to a given time budget, which varies from dataset to dataset. Participants who submit results—instead of code—are not constrained by the time budget since their code is run on their own platform. This is advantageous for entries counting towards the Final phases (immediately following a Tweakathon). Participants wishing to also enter the AutoML phases, which require submitting code, can submit both results and code (simultaneously). The results do not need to be produced by the submitted code; if a participant does not want to share personal code, he/she can submit the sample code provided by the organizers together with his/her results.

As of September 2015, more than 500 people registered and downloaded data, and more than 60 teams are actively participating. The top ranking participants significantly outperform the reference entry in rounds 0 and 1 (Tables, 1, 2, and 3). The methods of several participating teams are described in [2], [4], and [5].

The general strategy adopted by the participants was to (1) Reduce the search space with filter methods; some participants used meta-learning to design their filters with the help of

Table 3: Round 1 data: results of phase Final 1 on test set.

Participant	**<Rank>**	**DS 1**	**DS 2**	**DS 3**	**DS 4**	**DS 5**
aad.freiburg	2.20	0.5269 (4)	0.6378 (2)	0.8457 (2)	0.7925 (1)	0.9364 (2)
ideal.intel.analytics	3.20	0.5537 (1)	0.6458 (1)	0.8130 (8)	0.7153 (3)	0.9344 (3)
asml.intel.com	4.60	0.5441 (2)	0.6310 (3)	0.8191 (6)	0.6569 (7)	0.9280 (5)
reference	34.20	0.4722 (17)	0.5524 (33)	0.5324 (63)	0.5244 (37)	0.8934 (21)

classical machine learning dataset, typically taken from the UCI repository. (2) Reduce the number of hyper-parameters using versions of the algorithms that optimize them internally with some embedded methods. (3) Use an ensemble method to grow an ever improving ensemble until the computational budget is exhausted, e.g. the method proposed in [1]. Our presentation will explain in more details the successful methods presented at the AutoML 2015 workshop and ICML.

References

[1] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *21st International Conference on Machine Learning*, pages 18–. ACM, 2004.

[2] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Methods for improving bayesian optimization for automl. In *AutoML Workshop, International Conference on Machine Learning 2015*, 2015.

[3] Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, Núria Macià, Bisakha Ray, Mehreen Saeed, Alexander Statnikov, and Evelyne Viegas. Design of the 2015 chalearn automl challenge. In *The International Joint Conference on Neural Networks*, July 2015.

[4] Tadej Stajner. Autokit: Automatic machine learning via representation and model search. In *AutoML Workshop, International Conference on Machine Learning 2015*, 2015.

[5] Abhishek Thakur and Artus Krohn-Grimberghe. Autocompete: A framework for machine learning competitions. In *AutoML Workshop, International Conference on Machine Learning 2015*, 2015.