AutoML Challenge: AutoML Framework Using Random Space Partitioning Optimizer

Jungtaek Kim JTKIM@POSTECH.AC.KR
Pohang University of Science and Technology, Pohang 37673, Republic of Korea

Jongheon Jeong JONGHEONJ@EXBRAIN.IO
exbrain Inc., Pohang 37673, Republic of Korea

Seungjin Choi SEUNGJIN@POSTECH.AC.KR
Pohang University of Science and Technology, Pohang 37673, Republic of Korea

Abstract

Automated machine learning provides a framework where an algorithm configuration best suited to a particular problem is automatically determined without users' intervention. In this paper we present a method, referred to as Mondrian forests optimizer based on random space partitioning method, to modify the state-of-the-art system, *auto-sklearn*. We demonstrate that our method allows for incremental updating of a tree used for regression when the next candidate to be evaluated is given, while most of existing methods had to rebuild the tree. Our system, *postech.mlg_exbrain* ranked the 4th place in Final3 and Final4 phases, and 3rd place in AutoML5 phase of AutoML Challenge.

Keywords: Automated machine learning, Mondrian forests regression, random space partitioning optimizer

1. Introduction

Automated machine learning (AutoML) framework is to solve fixed-form datasets, which have different data dimension and data size, without any intervention. Generally, a machine learning problem has four steps; feature transformation, model parameter estimation, hyperparameter optimization, and algorithm selection. To solve the four steps automatically, several approaches (Fukunaga, 2002; Thornton et al., 2013; Feurer et al., 2015) have been proposed. Moreover, the practical implementations to solve the AutoML problem tackle on AutoML Challenge (Guyon et al., 2015, 2016). One of the state-of-the-art AutoML frameworks, *auto-sklearn* (Feurer et al., 2015) sequentially finds the best algorithm configuration to employ Sequential Model-based Algorithm Configuration (SMAC) (Hutter et al., 2010), which optimizes via Bayesian method using random forests.

In this paper, we propose the framework based on *auto-sklearn* with Random Space Partitioning Optimizer (RSPO), which divides the parameterized space via a probabilistic approach. We extend Mondrian forests regression (Lakshminarayanan et al., 2016) to Mondrian Forests Optimizer (MFO), which can handle all variables in the space of the parameterized algorithm configuration such as numerical and categorical variables. Furthermore, it can operate with the performance measure function which gives a response in parallel. The proposed system, *postech.mlg_exbrain* ranked the 4th place in Final3 and Final4 phases and 3rd place in AutoML5 phase.

2. Our Architecture

2.1. The Based System, *auto-sklearn* and Its Characteristics

Auto-sklearn (Feurer et al., 2015) is one of the state-of-the-art AutoML frameworks based on scikit-learn library (Pedregosa et al., 2011). It is composed of four components; meta-learning initializer, Bayesian optimizer, machine learning framework, and ensemble builder. It finds the best algorithm configuration sequentially as follows. First, meta-learning initializer suggests initial points provided from various datasets. Then, machine learning framework attempts to examine the parameterized configuration acquired from Bayesian optimizer, SMAC. Finally, newly acquired configuration builds an ensemble with previously acquired configurations.

We propose a framework based on *auto-sklearn* with MFO. The important part of *auto-sklearn* is an acquisition of the best candidate which might show the best performance. It is one of the most time-consuming steps. We utilize a method how the judicious initial point can be selected to overcome the limited resources, such as time budget and hardware limitation. Since MFO is a online and parallel algorithms, it can reduce the time to model the function and acquire the configurations. Moreover, SMAC estimates the uncertainty for Bayesian optimization via the heuristic way, where the uncertainty estimation of SMAC is computed as the uncertainty between the trees of random forests. This heuristic predictive uncertainty tends to be collapsed to 0 after a small number of iteration which causes the extrapolation of SMBO to underperform. On the contrary, MFO computes the uncertainty using the marginal distributions over nodes in the tree.

2.2. Mondrian Forests Optimizer

Algorithm 1: Mondrian Forests Optimizer

Input: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathcal{ACS}$ and y_i is sampled from the performance measure, and Time budget $\mathcal{T}$
Output: $\mathbf{x}_{best} \in \mathcal{ACS}$

$\mathcal{MF} =$ None
for $t < \mathcal{T}$ **do**
  if $\mathcal{MF} ==$ None **then**
    Build Mondrian forests, $\mathcal{MF}$ for $\mathcal{D}$
  else
    Extend $\mathcal{MF}$ with $\{(\mathbf{x}_{new,j}, y_{new,j})\}_{j=1}^{K}$
  end
  Draw seed configurations $\in \mathcal{ACS}$ of local search for min_for_search times
  Search the neighbors of seed configurations and find the candidates, whose responses of the acquisition function are higher
  Merge the randomly sampled configurations $\in \mathcal{ACS}$ with the candidates queried from the acquisition function
  Update the best K configurations, $\{(\mathbf{x}_{new,j}, y_{new,j})\}_{j=1}^{K}$ into $\mathcal{D}$
end
return $\mathbf{x}_{best} \in \mathcal{ACS}$ where $\mathbf{x}_{best}$ is the configuration which has the largest y_i of $(\mathbf{x}_i, y_i) \in \mathcal{D}$

We propose the novel Bayesian optimizer using a probabilistic random space partitioning, extended from Mondrian forests (Lakshminarayanan et al., 2014). Since labels are hard to acquire and the regression model is sequentially updated, own properties of Mondrian forests are suitable for the SMBO context. To deal with the AutoML problem, we extend Mondrian forests to MFO as in Alg. 1. 1) MFO can divide the categorical variable space

as well as the numerical variables to compare the configurations in the parameterized algorithm configuration space. We extend Mondrian forests regression in two ways, one-vs-rest partitioning and random partitioning in one-hot vector space for categorical variable. One-vs-rest partitioning for categorical variable is the simple extension, a decision tree is split by one-vs-rest criterion. It lets the configurations compare in one tree. One-hot vector space partitioning for categorical variable is the Mondrian process extension. The vector space is divided with a probabilistic approach. In this paper, we implement MFO to use one-vs-rest partitioning for simplicity. 2) MFO can parallelize SMBO to acquire a new point without actual observation. The new configuration holds the node distribution, which the configuration passes. Sequentially MFO runs in parallel and the actual response is updated after the unknown function is observed.

In Alg. 1, $\mathcal{ACS}$ denotes a product space of all parameterized algorithm configuration; algorithms, hyperparameters, and model parameters. MFO returns the best candidate which has been acquired to employ Bayesian optimization. Since the searching space is large and high dimensional, a local search method is applied in acquiring an algorithm configuration.

3. AutoML Challenge Results

Table 1: The results for AutoML Challenge Final3, Final4, and AutoML5 phases. All ranks are from https://competitions.codalab.org/competitions/2321. Rank with parentheses is the average of five datasets for each round.

Final3		**Final4**		**AutoML5**	
Team	**Rank**	**Team**	**Rank**	**Team**	**Rank**
aad_freiburg	1 (1.80)	aad_freiburg	1 (1.60)	aad_freiburg	1 (1.60)
djajetic	2 (2.00)	ideal.intel.analytics	2 (3.60)	djajetic	2 (2.60)
ideal.intel.analytics	3 (3.80)	abhishek4	3 (5.40)	**postech.mlg_exbrain**	3 (4.60)
asml.intel.com	3 (3.80)	**postech.mlg_exbrain**	4 (5.80)		
postech.mlg_exbrain	4 (5.40)				

AutoML Challenge (Guyon et al., 2015, 2016) has started in December 2014. It has 5 rounds, excluding round 0, and each round is composed of three phases, except for round 0 and 5. For 5 rounds, binary classification, multi-class classification, multi-label classification, and regression tasks are solved. Our system, *postech.mlg_exbrain* ranked the 4th place in Final3 and Final4 phases and 3rd place in AutoML5 phase as shown in Tab. 1.

4. Conclusion

In this paper, we propose the AutoML system using RSPO, implemented as MFO. Since the AutoML problem is under online and sequential setting, MFO is proper to solve the problem. Our system ranked the 4th place in Final3 and Final4 phases of AutoML Challenge and 3rd place in AutoML5 phase which is the last phase of the AutoML Challenge final round.

References

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 (NIPS)*, volume 28, 2015.

A. Fukunaga. Automated discovery of composite SAT variable-selection heuristics. In *Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI)*, pages 641–648, 2002.

I. Guyon, K. Bennett, G. Cawley, H. J. Escalante, S. Escalera, T. K. Ho, N. Macia, B. Ray, M. Saeed, A. Statnikov, et al. Design of the 2015 ChaLearn AutoML challenge. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015.

I. Guyon, I. Chaabane, H. J. Escalante, S. Escalera, D. Jajetic, J. R. Lloyd, N. Macía, B. Ray, L. Romaszko, M. Sebag, A. Statnikov, S. Treguer, and E. Viegas. A brief review of the ChaLearn AutoML challenge. In *Proceedings of AutoML 2016 Workshop on the International Conference on Machine Learning (ICML)*, 2016.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration (extended version). Technical Report 10-TR-SMAC, UBC, 2010.

B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, 2014.

B. Lakshminarayanan, D. M. Roy, and Y. W. Teh. Mondrian forests for large-scale regression when uncertainty matters. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 847–855, 2013.