AutoML System
Using Random Space Partitioning Optimizer

Jungtaek Kim[1], Jongheon Jeong[2], and Seungjin Choi[1]

[1]Department of Computer Science and Engineering,
Pohang University of Science and Technology, Pohang 37673, Republic of Korea
[2]exbrain Inc., Pohang 37673, Republic of Korea

The 15th KYUTECH-POSTECH Joint Workshop on Neuroinformatics
Aug 23, 2016

Machine Learning Group

Table of Contents

Motivation: AutoML

- Recently, deep neural networks solves many problems in machine learning community.
  - They take charge of the part of the preprocessing and representation learning.
  - They are usually trained end-to-end.
- However relatively small training dataset and the environment with low computing performance are not suitable for employing DNNs.
- The traditional machine learning algorithms also require automated machine learning (AutoML) framework easy to apply in the particular problems.

Motivation: Our Architecture, *postech.mlg_exbrain*

- To build the automated framework for various machine learning algorithms, we should search a huge cross product space.
  - It is formed by hyperparameter, model parameter, and algorithm vectors.
  - It contains categorical variables as well as numerical variables.
- The existing regression method on the huge searching space is not proper to find the best candidate of algorithm configuration.
- Random space partitioning method can be a way to find the best algorithm configuration.

General Machine Learning Framework

- A model parameter learning is

Supervised learning case

$$\underset{\theta \in \Theta}{\text{argmin}}\, \mathcal{L}(f(\theta; \lambda_i, F_j, A_i, \{(\mathbf{x}_k, y_k)\}_{k=1}^{n}))$$

Unsupervised learning case

$$\underset{\theta \in \Theta}{\text{argmin}}\, \mathcal{L}(f(\theta; \lambda_i, F_j, A_i, \{\mathbf{x}_k\}_{k=1}^{n}))$$

where $\mathcal{L}$ is a loss function, f is a predictive model, θ is a parameter vector, λ_i is the hyperparameter vector of the chosen algorithm A_i, and F_j is the chosen feature vector. $\mathbf{x}_k$ is an input value and y_k is a output value.

AutoML

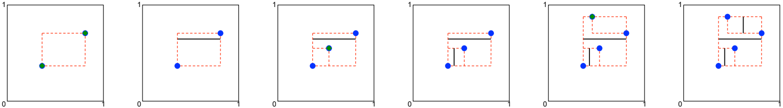
- Each problem optimizes model parameters, hyperparameters, and algorithms.
  - Hyperparameter optimization

$$\underset{\theta \in \Theta, \lambda_i \in \Lambda}{\operatorname{argmin}} \mathcal{L}(f(\theta, \lambda_i; F_j, A_i, \{\mathbf{x}_k, y_k\}_{k=1}^{n}))$$

  - Algorithm selection

$$\underset{\theta \in \Theta, \lambda_i \in \Lambda, A_i \in \mathcal{A}}{\operatorname{argmin}} \mathcal{L}(f(\theta, \lambda_i, A_i; F_j, \{\mathbf{x}_k, y_k\}_{k=1}^{n}))$$

Random Space Partitioning Method: Mondrian Process (Roy and Teh, 2009)



- Probability distribution over k-d tree data structure.
- Multidimensional generalization of Poisson process.
- Constructing multidimensional generalization of the stick-breaking process.

Mondrian Process

Algorithm 1 Mondrian Process

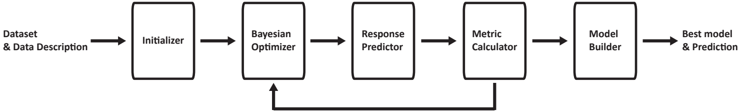
function $MONDRIAN(\Theta)$
    return $MONDRIAN\text{-}STARTED\text{-}AT(\Theta, 0)$
end function
function $MONDRIAN\text{-}STARTED\text{-}AT(\Theta, t_0)$
    $T \sim Exp(LD(\Theta))$
    $d \sim Discrete(p_1, \dots, p_D)$ where $p_d \propto (b_d - a_d)$
    $x \sim \mathcal{U}([a_d, b_d])$
    $M^< \rightarrow MONDRIAN\text{-}STARTED\text{-}AT(\Theta^<, t_0 + T)$ where $\Theta^< = \{\mathbf{z} \in \Theta | z_d \leq x\}$
    $M^> \rightarrow MONDRIAN\text{-}STARTED\text{-}AT(\Theta^>, t_0 + T)$ where $\Theta^> = \{\mathbf{z} \in \Theta | z_d \geq x\}$
end function

Mondrian Tree and Mondrian Forests (Lakshminarayanan *et al.*, 2015)

- A Mondrian tree is the restriction of a Mondrian process to the finite set of training data points.
- Mondrian forests is an ensemble of Mondrian Trees.
- The partitions are determined with respect to a covariate, not a label.
- The finite lifetime parameter controls the total number of splits (the maximum depth of standard decision tree).

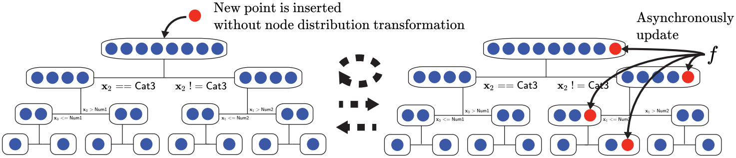
Our Architecture, *postech.mlg_exbrain*

- The based system, *auto-sklearn* (Feurer *et al.*, 2015)
  - Four components; meta-learning initializer, Bayesian optimizer, machine learning framework, and ensemble builder.
  - Bayesian optimizer, SMAC (Hutter *et al.*, 2010).
- Our system



  - Five components; meta-learning initializer, Bayesian optimizer, response predictor, metric calculator, and model builder.
  - Our optimizer, Mondrian Forests Optimizer.

Mondrian Forests Optimizer

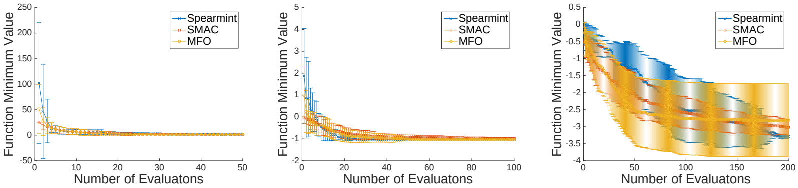


- Random space partitioning optimizer.
- Extended from Mondrian forests regression (Lakshminarayanan *et al.*, 2016).
- Handle all variables such as categorical and numerical variables.
- Run on both Mondrian forests optimizer and actual response sampler in parallel.

AutoML Challenge Results (Guyon et al., 2015, 2016)

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)				

Experiments: Mondrian Forests Optimizer

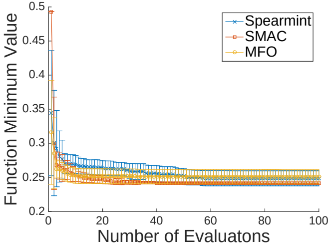


(g) Branin Function (h) Camelback Function (i) Hartman 6D Function

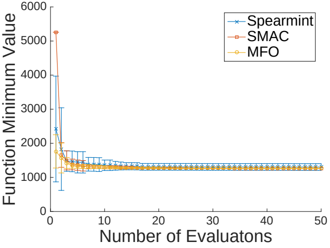
Figure 1: 10 runs for Spearmint and 50 runs for SMAC and MFO.

Machine Learning Group

Experiments: Mondrian Forests Optimizer



(a) SVM on grid



(b) LDA on grid

Figure 2: 10 runs for Spearmint and 50 runs for SMAC and MFO.

Thank you for
attending our presentation.