

# 랜덤 공간 분할 최적화기를 이용한 기계학습 자동화 시스템

김정택<sup>1</sup>, 정종헌<sup>2</sup>, 최승진<sup>1</sup>

<sup>1</sup>포항공과대학교 컴퓨터공학과  
37673, 경상북도 포항시 남구 청암로 77  
<sup>2</sup>엑스브레인  
37673, 경상북도 포항시 남구 청암로 77

2016 한국BI데이터마이닝학회 추계학술대회  
2016년 11월 18일



# 목차

동기

일반적인 기계학습 프레임워크

기계학습 자동화

몬드리안 프로세스와 몬드리안 포레스트

제안 시스템

몬드리안 포레스트 최적화기

AutoML Challenge 결과

실험: 몬드리안 포레스트 최적화기



# 동기: 기계학습 자동화

- ▶ 최근, 기계학습을 비롯한 많은 분야에서 딥 신경망이 높은 성능을 보이고 있음.
- ▶ 딥 신경망은 성능외에도 **전처리 과정**과 **표현학습 (representation learning)**을 해결하는 기능도 있음. 또한, **종단간 학습(end-to-end learning)**도 가능.
- ▶ 이러한 특성들은 알고리즘의 적용을 쉽게 만들면서 성능도 올리므로 많은 분야에서 사용이 가능.
- ▶ 하지만, 상대적으로 **적은 양의 데이터셋**과 **낮은 컴퓨팅 성능** 환경에서는 딥 신경망의 적용이 어려움.
- ▶ 따라서 전통적인 기계학습 알고리즘을 자동으로 적용할 **기계학습 자동화 시스템**이 필요.



## 동기: 제안 시스템, *postech.mlg\_exbrain*

- ▶ 다양한 기계학습 알고리즘을 적용하기 위한 자동화 시스템을 만들기 위해선, **넓은 알고리즘·하이퍼파라미터·매개변수의 외적 공간**에서 알맞은 알고리즘 구성을 찾아야 함.
  - ▶ 이 공간에는 숫자 변수(numerical variable)뿐만 아니라 분류 변수(categorical variable)도 포함.
- ▶ 기존의 회귀 방법들은 매우 넓은 공간에서 적절한 알고리즘 구성을 찾는데 부적합.
- ▶ **랜덤 공간 분할 최적화기(random space partitioning optimizer)**를 하나의 방법으로 제안.



# 알고리즘 구성 공간

- ▶ 기계학습 알고리즘을 작동하는 하나의 설정을 **알고리즘 구성**이라 부름.
- ▶ 이 알고리즘 구성의 전체 집합을 알고리즘 구성 공간으로 정의;

## 정의 1 (알고리즘 구성 공간)

$$\Theta \times \Lambda \times \mathbf{A}.$$

여기서  $\Theta$ 는 모델 매개변수의 전체 셋이며,  $\Lambda$ 는 하이퍼파라미터의 전체 셋,  $\mathbf{A}$ 는 알고리즘의 전체 셋.

- ▶ 정의 1에서 정의한 알고리즘 구성 공간에서 **가장 높은 성능을 보이는 알고리즘 구성을 찾는 것이 목표.**
- ▶ 실제로,  $\Lambda$ 는 약 250차원,  $\mathbf{A}$ 는 약 50차원으로 구성.



# 일반적인 기계학습 프레임워크

- ▶ 모델 매개변수 학습(model parameter learning)은

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

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

로 표현 가능. 여기서  $\mathcal{L}$ 은 손실 함수,  $f$ 는 예측 모델, 선택된 알고리즘  $A_i$ 의 매개변수와 하이퍼파라미터의 집합은  $\theta$ 와  $\lambda_i$ 임.  $\mathbf{x}_k$ 와  $y_k$ 는 이미 알고 있는 알고리즘 구성과 그의 성능측정값.

- ▶ 일반적인 기계학습은 알고리즘과 하이퍼파라미터가 고정된 환경에서 알맞은 매개변수를 찾는 것으로 해석 가능.



# 기계학습 자동화

- ▶ 기계학습은 다음과 같이 순차적으로 매개화 가능. 최종적으로 **기계학습 자동화 문제**를 정의할 수 있음.
  - ▶ **하이퍼파라미터 최적화**는 모델 매개변수와 하이퍼파라미터를 동시에 최적화함;

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

- ▶ **정의 2 (기계학습 자동화)** 기계학습 자동화는 **모델 매개변수, 하이퍼파라미터, 알고리즘을 동시에** 최적화함;

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



# 랜덤 공간 분할: 몬드리안 프로세스 (Roy and Teh, 2009)

---

## Algorithm 1 몬드리안 프로세스

---

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

---

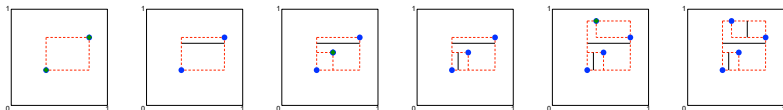
- ▶ k-d 트리의 확률적인 일반화.
- ▶ 푸아송 프로세스의 고차원 일반화.





# 몬드리안 트리와 몬드리안 포레스트

(Lakshminarayanan *et al.*, 2015)

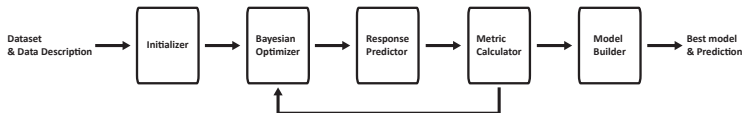


- ▶ 몬드리안 트리는 데이터가 유한한 몬드리안 프로세스.
- ▶ 몬드리안 포레스트는 몬드리안 트리의 앙상블.
- ▶ 트리의 분할은 정의역에 의해서만 결정되며, 함수의 값은 영향을 끼치지 않음.
- ▶ 유한한 수명(lifetime) 매개변수가 트리의 분할수를 결정 (일반적인 결정 트리의 최대 깊이와 같음).



# 제안 시스템, *postech.mlg\_exbrain*

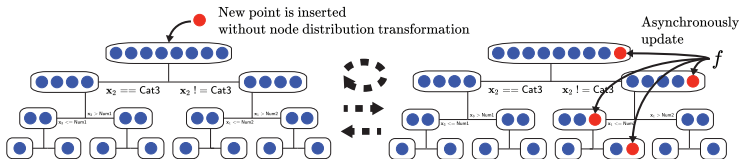
- ▶ 기반 시스템, *auto-sklearn* (Feurer *et al.*, 2015)
  - ▶ 네 구성요소: 메타학습 초기화기(meta-learning), 베이지안 최적화기(Bayesian optimizer), 기계학습 프레임워크(machine learning framework), 앙상블 제조기(ensemble builder).
  - ▶ **랜덤 포레스트** 기반 베이지안 최적화기, SMAC (Hutter *et al.*, 2010).
- ▶ 제안 시스템



- ▶ 다섯 구성요소: 메타학습 초기화기, **베이지안 최적화기**, 성능측정값 예측기(response predictor), 측정기준 계산기(metric calculator), 모델 제조기(model builder).
- ▶ 몬드리안 포레스트 기반 베이지안 최적화기, **몬드리안 포레스트 최적화기**.



# 몬드리안 포레스트 최적화기



- ▶ 랜덤 공간 분할 최적화기.
- ▶ 몬드리안 포레스트 회귀에서 확장 (Lakshminarayanan *et al.*, 2016).
- ▶ 숫자 변수와 분류 변수와 같은 **모든 변수**를 최적화.
- ▶ 몬드리안 포레스트 최적화기와 실제 성능측정값 추출기가 **병렬**로 작동.



# AutoML Challenge 결과 (Guyon et al., 2015)

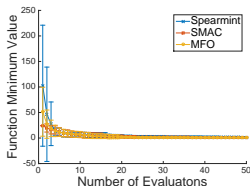
- ▶ 총 세 단계(AutoML, Tweakathon, Final)로 구성되어 있는 5개의 라운드가 진행.
- ▶ 다섯 라운드에 걸쳐서 이진 분류, 다중 클래스 분류, 다중 레이블 분류, 회귀 문제를 해결.
- ▶ 총 30개의 데이터셋이 제공.

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)	<b>postech.mlg_exbrain</b>	3 (4.60)
asml.intel.com	3 (3.80)	<b>postech.mlg_exbrain</b>	4 (5.80)		
<b>postech.mlg_exbrain</b>	4 (5.40)				

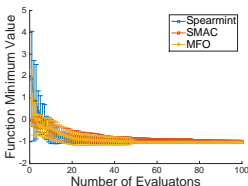


# 실험: 몬드리안 포레스트 최적화기

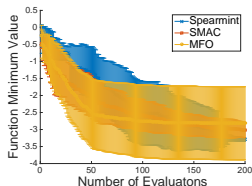
- ▶ 3개의 전역 최적화 벤치마크에 대해서 전역 최적값을 찾는 실험.



(g) Branin 함수



(h) Camelback 함수

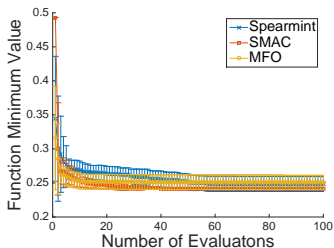


(i) Hartman 6D 함수

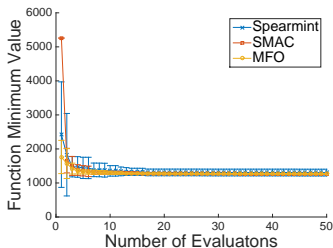
Figure 1: Spearmint는 10번 반복. SMAC과 MFO는 50번 반복.

# 실험: 몬드리안 포레스트 최적화기

- ▶ SVM과 LDA에 대해 각각 3개의 하이퍼파라미터를 최적화.



(a) SVM



(b) LDA

Figure 2: Spearmint는 10번 반복. SMAC과 MFO는 50번 반복.

