Towards Explainable AutoML: xAutoML

Marius Lindauer

@LindauerMarius
@AutoML_org

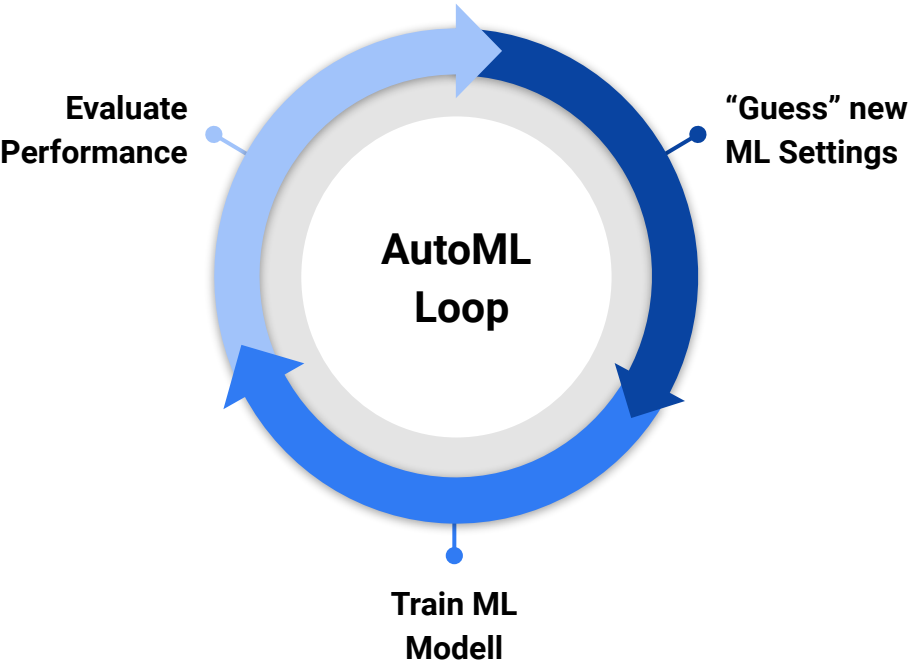
Leibniz Universität Hannover

tnt

AutoML.org Automated Machine Learning Hannover

* Slides available at automl.org/talks

AutoML Process



Optimized ML-Pipeline

Pre-Processing	Predictive Model	Post-Processing
Normalization, Feature Selection, Feature Reduction, ...	Linear model, SVM, RF, XGB, DNN, ...	

Leibniz Universität Hannover

AutoML.org

Who is using AutoML?

Users of ML
without any
deep expertise in ML

ML experts /
Data scientists …
(AutoML researchers)

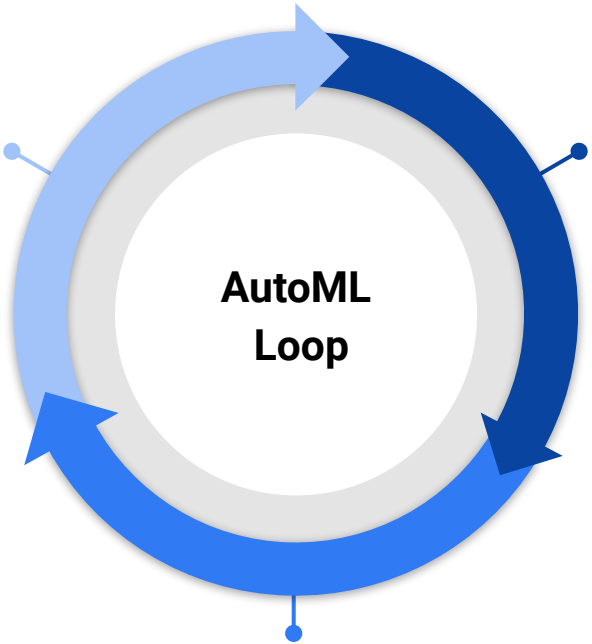
Are ML-Experts using AutoML?

- Bouthillier and Varoquaux [2020] showed that authors of NeurIPS and ICLR papers:
  - a) they often **optimize their hyperparameters** (>75%)
  - b) they often **do it manually** and don't use AutoML tools
- Crisan and Fiore-Gartland [2021] interviewed data scientists and concluded:
  - a) experts **don't necessarily trust AutoML**
  - b) **visualization of results and processes** can help to increase the acceptance of AutoML results

Previous Approach-Agnostic xAutoML Methods

Note: Of course, approach-specific methods are also possible, e.g. [Ru et al. 2021].

General Setup



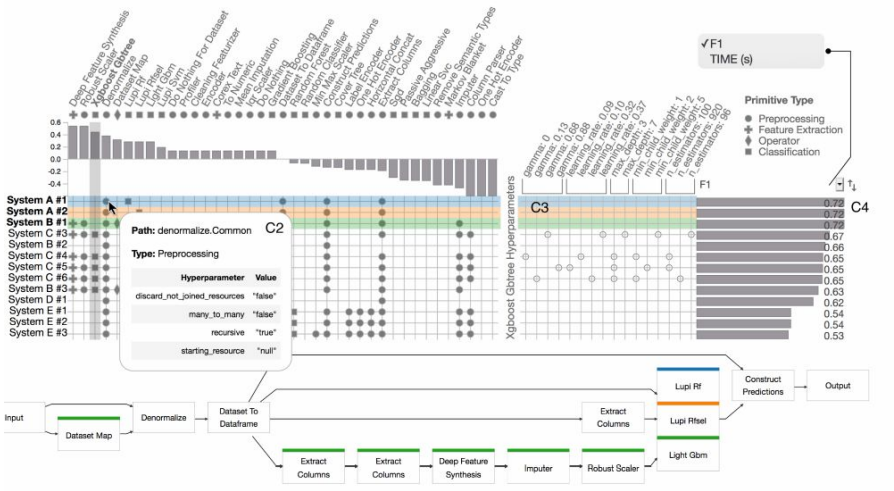
$$c^{(t)} = \mathcal{L}(\theta^*, \mathcal{D}_{\text{val}}; \lambda^{(t)})^*$$

$$\lambda^{(t)}$$

$$\theta^* \in \min_{\theta} \mathcal{L}(\theta; \mathcal{D}_{\text{train}}, \lambda^{(t)})$$

* can in principle be any kind of cost function.

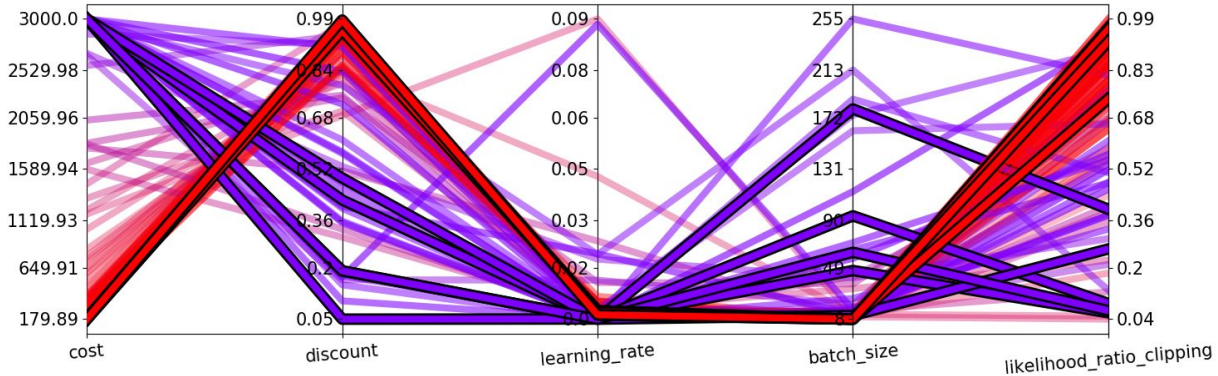
Visualization of Pipelines



Source: [Ono et al. 2020]

- Visualization of sampled pipelines (incl. algorithms and hyperparameters) and their performance
- More descriptive analysis

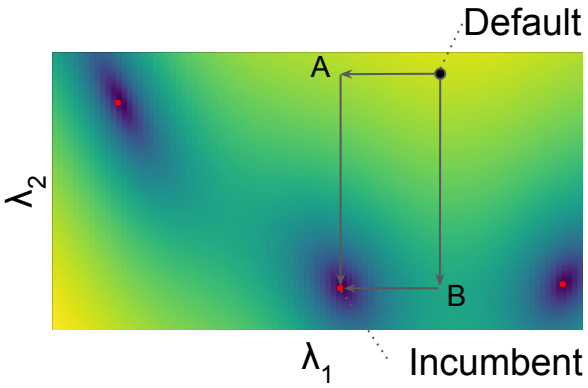
Parallel Coordinate Plots



PPO on cartpole; Source: [Lindauer et al. 2019]

- Visualization of sampling in high-dimensional hyperparameter spaces [Golovin et al. 2017]
- Allows to identify:
  - optimization focus of AutoML optimiziers
  - well-performing combination of settings
  - Interaction effects between settings (to some degree)
- Rather qualitative, less quantitative analysis
- Follow up: Conditional parallel coordinate plots [Weidele et al. 2019]

Ablation Studies



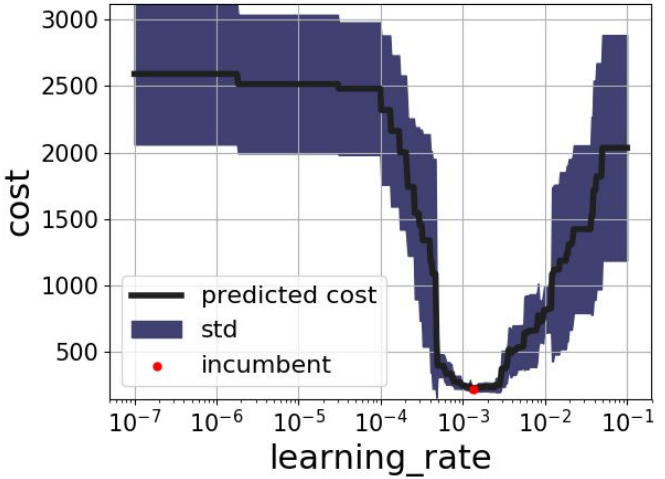
- AutoML can start from some default settings
  a. Defined by the algorithm developer
  b. User expertise

- Question: **Which settings (λ) changes had the bigger impact on the performance?**

- Characteristic:
  a. Quantitative analysis
  b. Hard to visualize in high-dimensional spaces
  c. Rather local: subspace between default and incumbent setting

- Efficiency in high-dimensional spaces:
  a. Greedy approach [Fawcett and Hoos 2016]
  b. Use of surrogate models
     instead of expensive function evaluations
     (e.g., from Bayesian Optimization) [Biedenkapp et al. 2017]

ICE Curves and LPI



PPO on cartpole
Source: [Lindauer et al. 2019]

- ICE Curve [Goldstein et al. 2017]: individual effect of one feature for an individual observation
  - Slice through the space in one dimension at a given observation
- LPI: Local (Hyper-)Parameter Importance [Biedenkapp et al. 2018]
  - Same idea as ICE curves, but single ICE curve centered at the incumbent setting returned by an AutoML tool
  - Quantitative importance of hyperparameters:

$$LPI(h \mid \lambda) = \frac{Var_{v \in \Lambda_h} \hat{c}(\lambda[\lambda_h = v])}{\sum_{h' \in \mathcal{H}} Var_{w \in \Lambda_{h'}} \hat{c}(\lambda[\lambda_h = w])}$$

fANOVA for Hyperparameter Importance

	fANOVA	LPI
discount	19.32	38.88
learning rate	3.70	35.4
batch size	15.77	21.5
# units 1	1.86	0.07
# units 2	0.39	0.01

PPO on cartpole
Source: [Lindauer et al. 2019]

- Fraction of explained variance by main and interaction effects of hyperparameters can be quantified by

$$\mathbb{V}_{\mathcal{H}' \subset \mathcal{H}} = \frac{\frac{1}{||\Lambda_{\mathcal{H}'}||} \int \hat{c}(\lambda_{\mathcal{H}'})^2 d\lambda_{\mathcal{H}'}}{\frac{1}{||\Lambda||} \int (\hat{y}(\lambda) - \hat{c}_\varnothing)^2 d\lambda}$$

- Efficient computation on a RF as surrogate model [Hutter et al. 2014]
- Allows to study importance across datasets [van Rijn & Hutter 2018, Sharma et al. 2019]

Explaining HPO via Partial Dependence Plots

Julia Moosbauer[1], Julia Herbinger[1], Giuseppe Casalicchio[1], Marius Lindauer[2], Bernd Bischl[1]

[1] LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

[2] Leibniz Universität Hannover

PDP: Partial Dependence Plots [Friedman 2001]

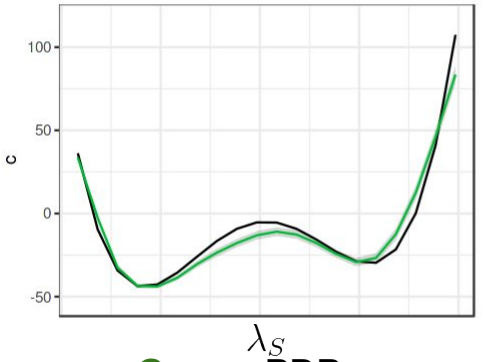
For, a subset S of the hyperparameters, the partial dependence function is:

$$c_S(\lambda_S) := \mathbb{E}_{\lambda_C}\left[c(\lambda)\right] = \int_{\Lambda_C} c(\lambda_S, \lambda_C) d\mathbb{P}(\lambda_C)$$

and can be approximated by Monte-Carlo integration:

$$\hat{c}_S\left(\lambda_S\right) = \frac{1}{n}\sum_{i=1}^{n} \hat{m}\left(\lambda_S, \lambda_C^{(i)}\right)$$

where $\left(\lambda_C^{(i)}\right)_{i=1,\ldots,n} \sim \mathbb{P}(\lambda_C)$ and λ_S for a set of grid points.



Green: PDP
Black: Ground truth

→ Average of ICE curves.

[Hutter et al. 2014] showed how to do this efficiently for RFs as surrogate models.

Quantifying Uncertainties in PDPs

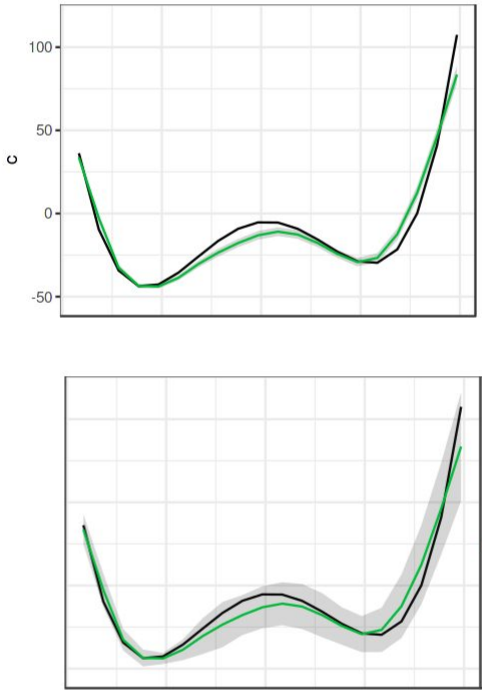
$$
\begin{aligned}
&\hat{s}_S^2(\lambda_S) \\
&= \mathbb{V}_{\hat{c}}\left[\hat{c}_S\left(\lambda_S\right)\right] \\
&= \mathbb{V}_{\hat{c}}\left[\frac{1}{n}\sum_{i=1}^{n}\hat{c}\left(\lambda_S, \lambda_C^{(i)}\right)\right] \\
&= \frac{1}{n^2} 1^\top \hat{K}\left(\lambda_S\right)\ 1.
\end{aligned}
$$

→ requires a kernel correctly specifying the covariance structure (e.g., GPs).

Approximation:

$$
\hat{s}_S^2\left(\lambda_S\right) \approx \frac{1}{n}\sum_{i=1}^{n}\hat{K}\left(\lambda_S\right)_{i,i}
$$

→ Model-agnostic (local) approximation



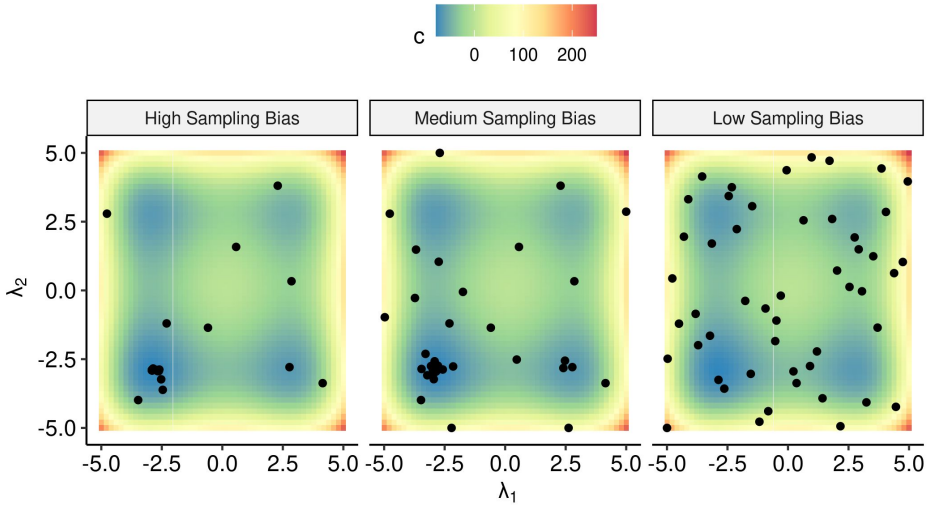
Ground truth
PDP
Uncertainty

Problem: Biased Sampling

- PDPs assume that the data is i.i.d.
- Obviously not the case for efficient AutoML tools with a focus on high-performance regions

- Example:
  - BO with GPs and LCB
  - Different exploration rate for LCB to show different sampling bias

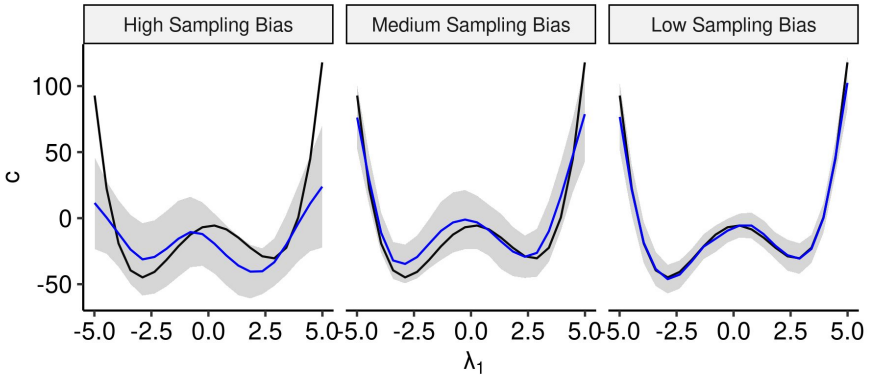
$$\text{LCB}(\lambda) = \mu(\lambda) + \beta \cdot \sigma(\lambda)$$

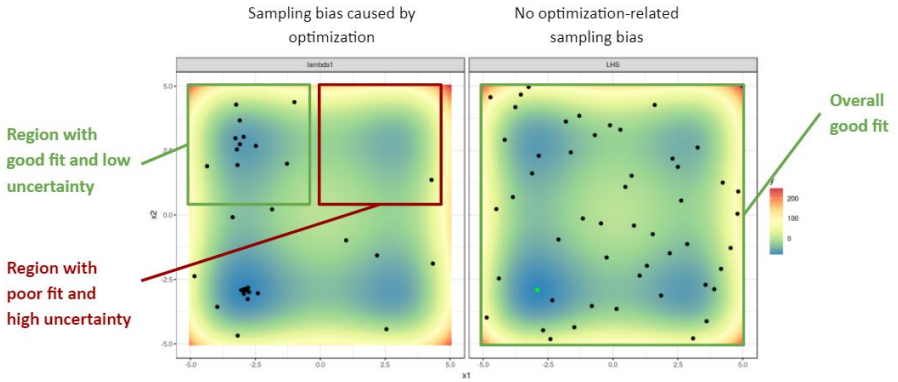


Impact of the Sampling Bias

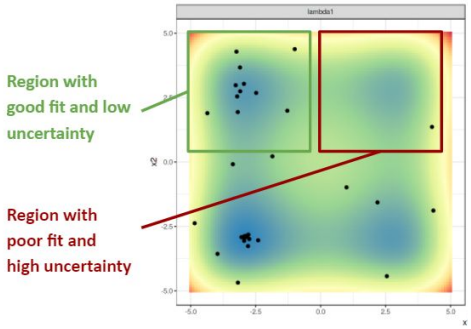
- Simply using all observations from AutoML tools might lead to misleading PDPs
- Uncertainty estimates help to quantify the poor fits

→ of course, sampling bias is wanted and the solution cannot be to change the sampling behavior





Partitioning of Space



Partition space to obtain interpretable subspaces $\mathcal{N}'$.

Uncertainty variation across all ICE estimates:

$$L\left(\lambda_S, \mathcal{N}'\right) = \sum\nolimits_{i \in \mathcal{N}} \left(\hat{s}^2 \left(\lambda_S, \lambda_C^{(i)} \right) - \hat{s}^2_{S|\mathcal{N}'} \left(\lambda_S \right) \right)^2$$

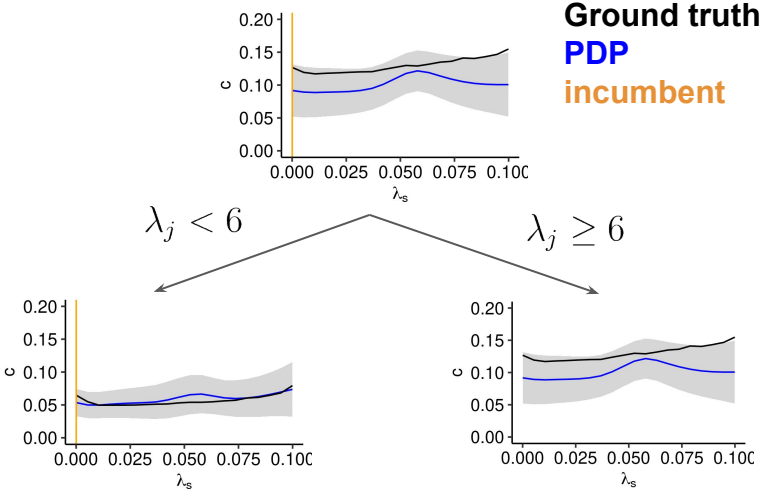
$$\hat{s}^2_{S|\mathcal{N}'} \left(\lambda_S \right) := \frac{1}{|\mathcal{N}'|} \sum_{i \in \mathcal{N}'} \hat{s}^2 \left(\lambda_S, \lambda_C^{(i)} \right)$$

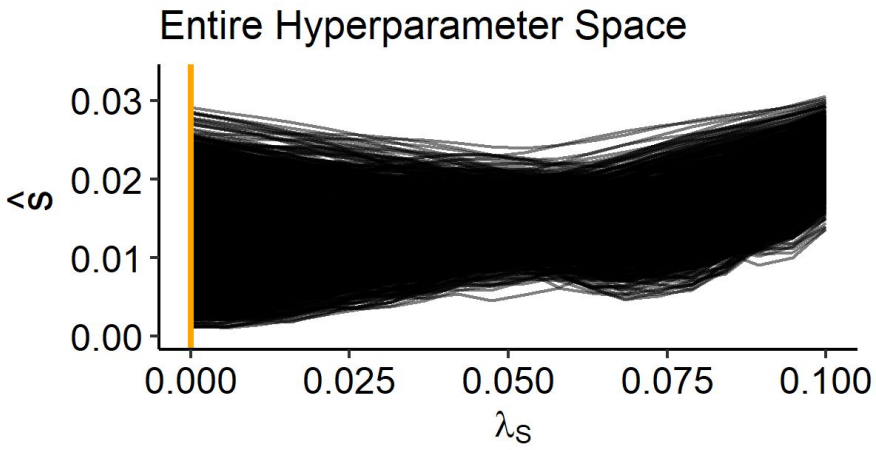
→ **Uncertainty structure of ICE curves should maximally agree**

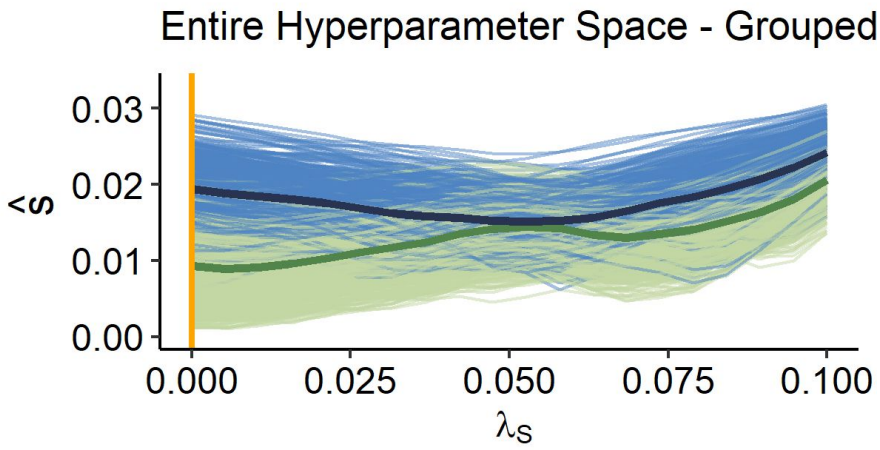
Split Loss = Aggregation over all grid points:

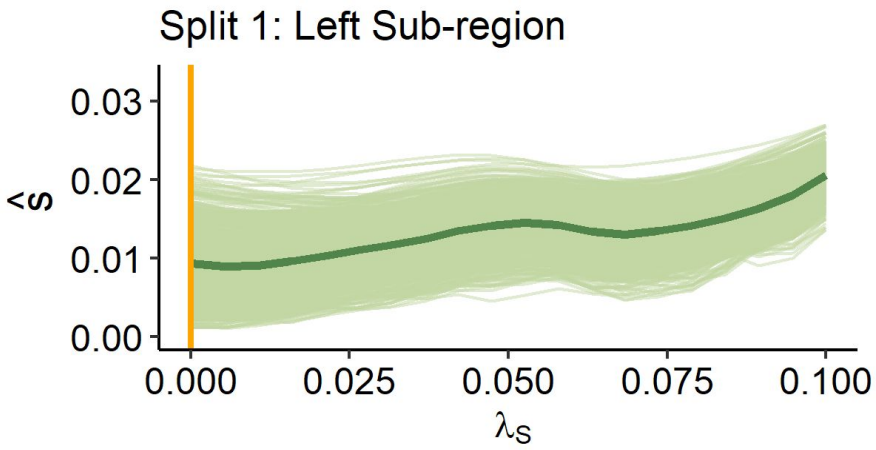
$$\mathcal{R}_{L2}(\mathcal{N}') = \sum\nolimits_{g=1}^{G} L(\lambda_S^{(g)}, \mathcal{N}')$$

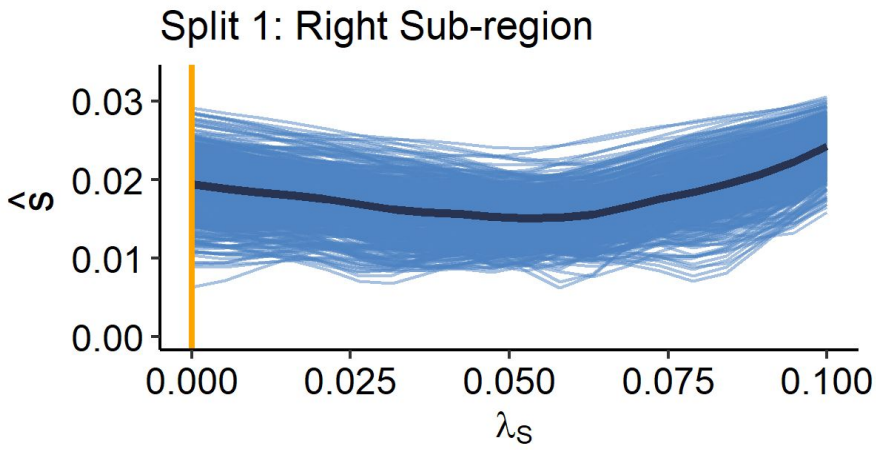
Note (i): Partition only along the marginalized dimensions











Leibniz Universität Hannover

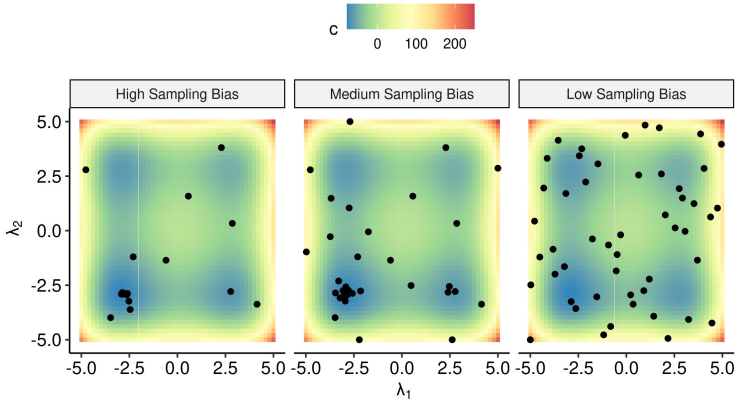
AutoML.org

Empirical Results

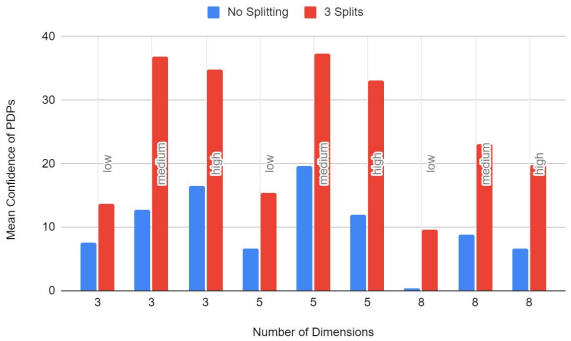
Effect of Splitting on an Artificial Function

Main Insights:

- For higher-dimensional problems, PDPs are potentially more uncertain
- Mean Confidence (MC) increases with deeper trees
- More to gain for high-sampling bias cases



Styblinski-Tang function



Explaining LCBench [Zimmer et al. 2021]

Name	Range	log	type
Number of layers	$[1, 5]$	no	int
Max. number of units	$[64, 512]$	yes	int
Batch size	$[16, 512]$	yes	int
Learning rate (SGD)	$[1e^{-4}, 1e^{-1}]$	yes	float
Weight decay	$[1e^{-5}, 1e^{-1}]$	no	float
Momentum	$[0.1, 0.99]$	no	float
Max. dropout rate	$[0.0, 1.0]$	no	float

Hyperparameter	δ MC (%)	δ OC (%)	δ NLL (%)
Batch size	40.8 (14.9)	61.9 (13.5)	19.8 (19.5)
Learning rate	50.2 (13.7)	57.6 (14.4)	17.9 (20.5)
Max. dropout	49.7 (15.4)	62.4 (11.9)	17.4 (18.2)
Max. units	51.1 (15.2)	58.6 (12.7)	24.6 (22.0)
Momentum	51.7 (14.5)	58.3 (12.7)	19.7 (21.7)
Number of layers	30.6 (16.4)	50.9 (16.6)	13.8 (32.5)
Weight decay	36.3 (22.6)	61.0 (13.1)	11.9 (19.7)

Improvement of mean confidence (MC), confidence close to incumbent (OC), and negative log-likelihood (NLL) after 6 splits

Setting:

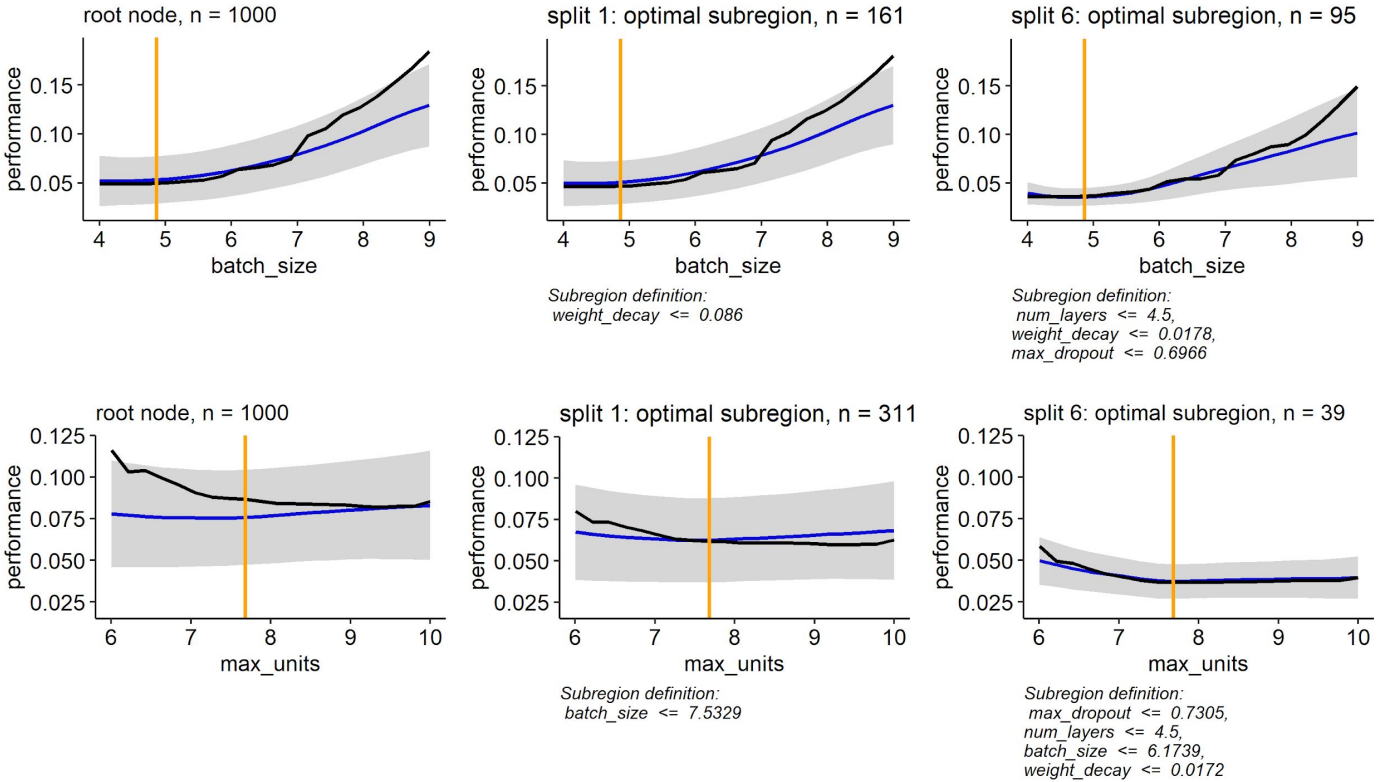
- Small configuration space of **Auto-PyTorch Tabular**
- Training of a RF as surrogate on LCBench with 2000 randomly sampled configurations
- Bayesian Optimization with 200 function evaluations

Take-away:

→ the confidence of PDPs improves across all hyperparameters and metrics

Explaining Auto-PyTorch (cont'd) [Zimmer et al. 2021]

Ground truth
PDP
incumbent



Future Work and Conclusion

Take-Home

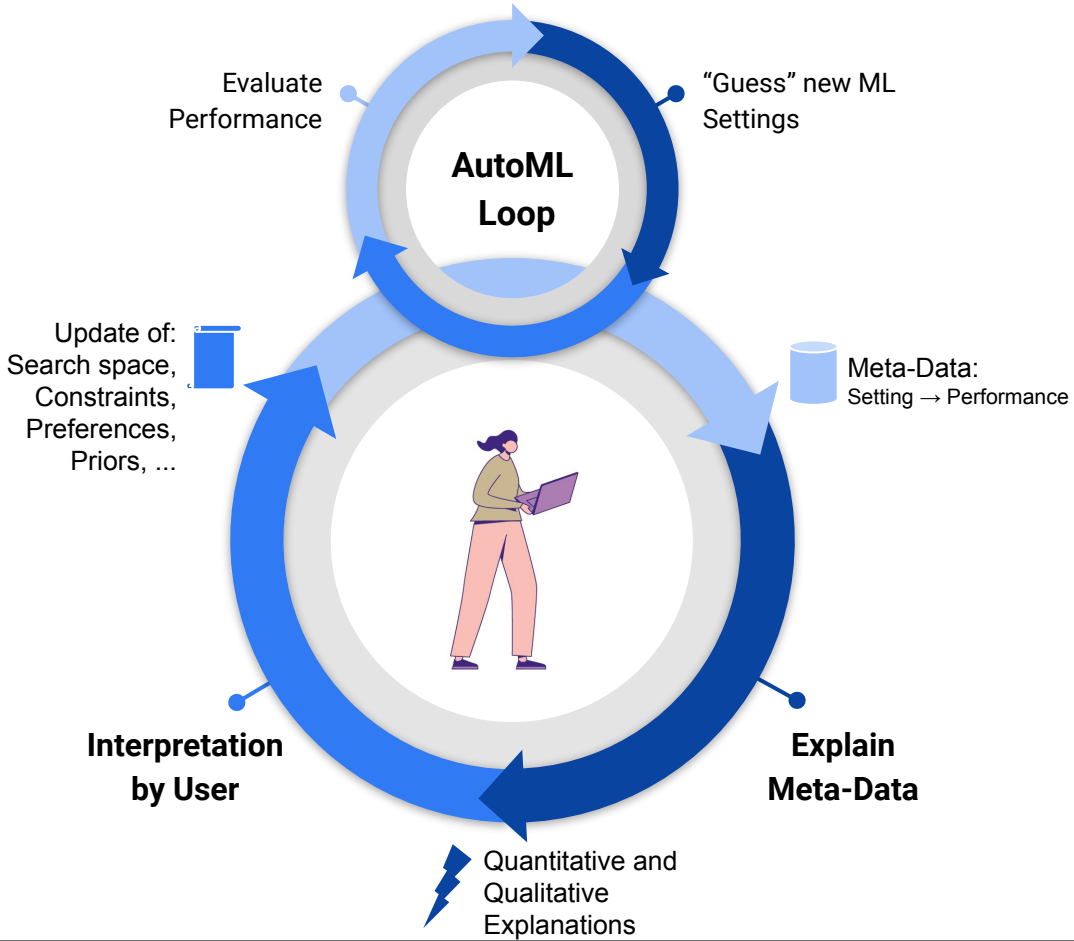
Summary:

- Explaining AutoML is important to **create trust** in them
- Common **iML methods** such as PDPs can be used to **explain AutoML**
- However, **i.i.d assumptions** might be **violated**
- PDPs can be extended to **uncertainty estimates**
- Split into subspaces with **better interpretability**

Future Work:

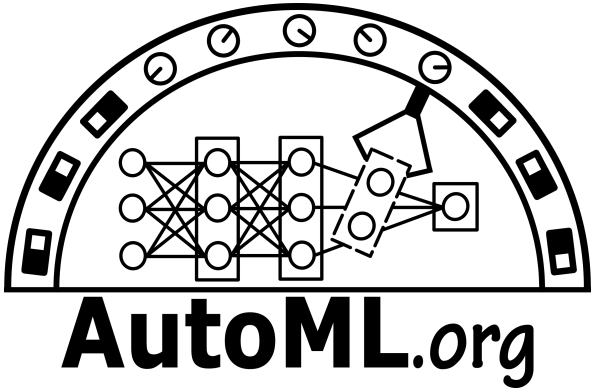
- **Additional samples** to efficiently reduce sampling bias
- Extension to **multi-fidelity setting AutoML**
- Other iML methods

Goal: Human-Centered AutoML



@LindauerMarius
@AutoML_org

Thank you!



Leibniz
Universität
Hannover

AutoML.org