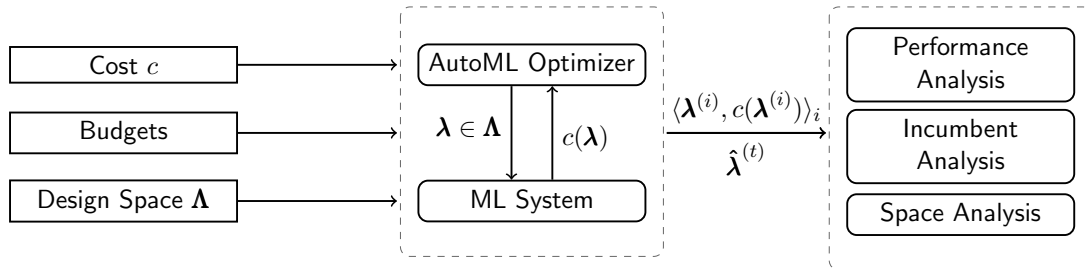


# AutoML: Interpretability

Incumbent Analysis and Local Hyperparameter Importance

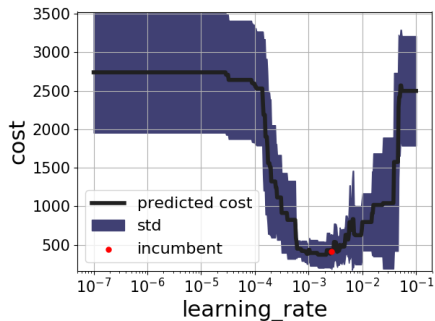
Bernd Bischl   Frank Hutter   Lars Kotthoff  
Marius Lindauer   Joaquin Vanschoren

# Idea



~> focus on why is the eventually returned configuration a good choice

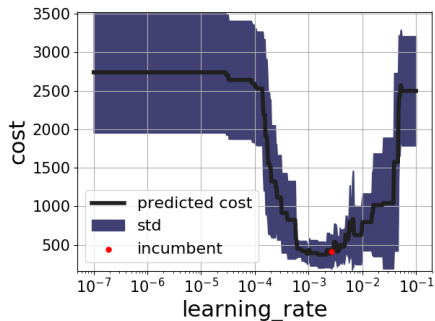
# Local Importance [Biedenkapp et al. 2018]



- Typical question of users:
  - ▶ How would the performance change if we change hyperparameter  $\lambda_i$ ?

Source: [Lindauer et al. 2019]

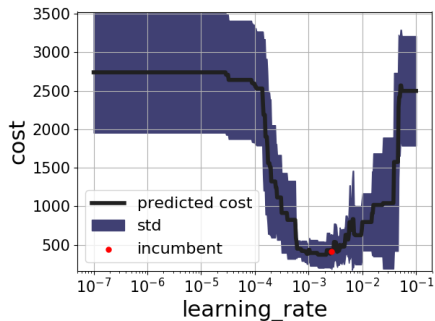
# Local Importance [Biedenkapp et al. 2018]



Source: [Lindauer et al. 2019]

- Typical question of users:
  - ▶ How would the performance change if we change hyperparameter  $\lambda_i$ ?
- Problem: Running full study is often too expensive
  - ▶ Each run of an ML-system is potential expensive

# Local Importance [Biedenkapp et al. 2018]



Source: [Lindauer et al. 2019]

- Typical question of users:
  - ▶ How would the performance change if we change hyperparameter  $\lambda_i$ ?
- Problem: Running full study is often too expensive
  - ▶ Each run of an ML-system is potential expensive
- Key Ideas:
  - ▶ Re-use probabilistic models as trained in BO
  - ▶ Plot performance change around  $\hat{\lambda}^{(t)}$  along each dimension

$$\text{VAR}_{\boldsymbol{\lambda}}(i) = \sum_{v \in \boldsymbol{\Lambda}_i} (\mathbb{E}_{v \sim \boldsymbol{\Lambda}_i} [L(\boldsymbol{\lambda})] - L(\boldsymbol{\lambda}[\boldsymbol{\lambda}_i := v]))^2 \quad (1)$$

$$\text{VAR}_{\boldsymbol{\lambda}}(i) = \sum_{v \in \boldsymbol{\Lambda}_i} (\mathbb{E}_{v \sim \boldsymbol{\Lambda}_i} [L(\boldsymbol{\lambda})] - L(\boldsymbol{\lambda}[\boldsymbol{\lambda}_i := v]))^2 \quad (1)$$

$$\text{LPI}(i \mid \boldsymbol{\lambda}) = \frac{\text{VAR}_{\boldsymbol{\lambda}}(i)}{\sum_j \text{VAR}_{\boldsymbol{\lambda}}(j)} \quad (2)$$

$$\text{VAR}_{\boldsymbol{\lambda}}(i) = \sum_{v \in \boldsymbol{\Lambda}_i} (\mathbb{E}_{v \sim \boldsymbol{\Lambda}_i} [L(\boldsymbol{\lambda})] - L(\boldsymbol{\lambda}[\boldsymbol{\lambda}_i := v]))^2 \quad (1)$$

$$\text{LPI}(i \mid \boldsymbol{\lambda}) = \frac{\text{VAR}_{\boldsymbol{\lambda}}(i)}{\sum_j \text{VAR}_{\boldsymbol{\lambda}}(j)} \quad (2)$$

~> While fixing all other hyperparameters to the incumbent value,  
the hyperparameter with the highest variance is the most important one



# Ablation Study for Importance

- Users often start from some kind of default configuration
  - ① As given in the documentation
  - ② Or as always used in the last time

# Ablation Study for Importance

- Users often start from some kind of default configuration
  - ① As given in the documentation
  - ② Or as always used in the last time
- **Key Idea:** Going from the default to the automatically optimized configuration, which choices were important?

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42]\end{aligned}$$

# Ablation Study for Importance

- Users often start from some kind of default configuration
  - ① As given in the documentation
  - ② Or as always used in the last time
- **Key Idea:** Going from the default to the automatically optimized configuration, which choices were important?

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42]\end{aligned}$$

- Cheap approach: Assess  $\lambda^{(\text{end})}$  with each hyperparameter value from  $\lambda^{(\text{start})}$

# Ablation Study for Importance

- Users often start from some kind of default configuration
  - ① As given in the documentation
  - ② Or as always used in the last time
- **Key Idea:** Going from the default to the automatically optimized configuration, which choices were important?

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42]\end{aligned}$$

- Cheap approach: Assess  $\lambda^{(\text{end})}$  with each hyperparameter value from  $\lambda^{(\text{start})}$
- Expensive approach: Try all mixtures of  $\lambda^{(\text{end})}$  and  $\lambda^{(\text{start})}$ 
  - ▶ Only feasible for small spaces and fairly cheap ML systems

# Ablation Study for Importance

- Users often start from some kind of default configuration
  - ① As given in the documentation
  - ② Or as always used in the last time
- **Key Idea:** Going from the default to the automatically optimized configuration, which choices were important?

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42]\end{aligned}$$

- Cheap approach: Assess  $\lambda^{(\text{end})}$  with each hyperparameter value from  $\lambda^{(\text{start})}$
- Expensive approach: Try all mixtures of  $\lambda^{(\text{end})}$  and  $\lambda^{(\text{start})}$ 
  - ▶ Only feasible for small spaces and fairly cheap ML systems
- Trade-off: Find a way from  $\lambda^{(\text{start})}$  to  $\lambda^{(\text{end})}$  in a greedy fashion [Fawcett and Hoos. 2016]

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

1st Iteration:

$$\lambda^{(1)} = [0.98, 1, 0, 100] \quad L_1 = 19\%$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

1st Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 0, 100] & L_1 &= 19\% \\ \lambda^{(2)} &= [1, 2.42, 0, 100] & L_2 &= 20\%\end{aligned}$$



# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

1st Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 0, 100] & L_1 &= 19\% \\ \lambda^{(2)} &= [1, 2.42, 0, 100] & L_2 &= 20\% \\ \lambda^{(3)} &= [1, 1, 1, 100] & L_3 &= 7\%\end{aligned}$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

1st Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 0, 100] & L_1 &= 19\% \\ \lambda^{(2)} &= [1, 2.42, 0, 100] & L_2 &= 20\% \\ \lambda^{(3)} &= [1, 1, 1, 100] & L_3 &= 7\% \\ \lambda^{(4)} &= [1, 1, 0, 42] & L_4 &= 16\%\end{aligned}$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

1st Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 0, 100] & L_1 &= 19\% \\ \lambda^{(2)} &= [1, 2.42, 0, 100] & L_2 &= 20\% \\ \lambda^{(3)} &= [1, 1, 1, 100] & L_3 &= 7\% \\ \lambda^{(4)} &= [1, 1, 0, 42] & L_4 &= 16\%\end{aligned}$$

$\rightsquigarrow$  1st step:  $\lambda_2$  – flipping hyperparameter 3

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(s1)} &= [1, 1, \textcolor{blue}{1}, 100] & L &= 7\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

2nd Iteration:

$$\lambda^{(1)} = [\textcolor{blue}{0.98}, 1, 1, 100] \quad L_1 = 6\%$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(s1)} &= [1, 1, 1, 100] & L &= 7\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

2nd Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 1, 100] & L_1 &= 6\% \\ \lambda^{(2)} &= [1, 2.42, 1, 100] & L_2 &= 7\%\end{aligned}$$

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(s1)} &= [1, 1, 1, 100] & L &= 7\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

2nd Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 1, 100] & L_1 &= 6\% \\ \lambda^{(2)} &= [1, 2.42, 1, 100] & L_2 &= 7\% \\ \lambda^{(3)} &= [1, 1, 1, 42] & L_3 &= 5\%\end{aligned}$$

$\rightsquigarrow$  2nd step:  $\lambda_3$  – flipping hyperparameter 4

# Greedy Ablation Study

Given:

$$\begin{aligned}\lambda^{(\text{start})} &= [1, 1, 0, 100] & L_{\text{start}} &= 20\% \\ \lambda^{(s1)} &= [1, 1, 1, 100] & L &= 7\% \\ \lambda^{(s2)} &= [1, 1, 1, 42] & L &= 5\% \\ \lambda^{(\text{end})} &= [0.98, 2.42, 1, 42] & L_{\text{end}} &= 4\%\end{aligned}$$

3rd Iteration:

$$\begin{aligned}\lambda^{(1)} &= [0.98, 1, 1, 100] & L_1 &= 4\% \\ \lambda^{(2)} &= [1, 2.42, 1, 100] & L_2 &= 5\%\end{aligned}$$

$\rightsquigarrow$  2nd step:  $\lambda_3$  – flipping hyperparameter 1

# Greedy Ablation Study

Ablation Path:

$$\lambda^{(\text{start})} = [1, 1, 0, 100] \quad L_{\text{start}} = 20\%$$

$$\lambda^{(s1)} = [1, 1, 1, 100] \quad L = 7\%$$

$$\lambda^{(s1)} = [1, 1, 1, 42] \quad L = 5\%$$

$$\lambda^{(s3)} = [0.98, 1, 1, 42] \quad L = 4\%$$

$$\lambda^{(s4)} = [0.98, 2.42, 1, 42] \quad L = 4\%$$

$$\lambda^{(\text{end})} = [0.98, 2.42, 1, 42] \quad L_{\text{end}} = 4\%$$



# Greedy Ablation Pseudo Code

---

**Algorithm 1** Greedy Ablation

---

**Input** : Algorithm  $\mathcal{A}$  with configuration space  $\Lambda$ , start configuration  $\lambda^{(\text{start})}$ ,  
end configuration  $\lambda^{(\text{end})}$ , cost metric  $c$

$\lambda \leftarrow \lambda^{(\text{start})};$

$P \leftarrow [] ;$

# Greedy Ablation Pseudo Code

---

**Algorithm 2** Greedy Ablation

---

**Input** : Algorithm  $\mathcal{A}$  with configuration space  $\mathbf{\Lambda}$ , start configuration  $\lambda^{(\text{start})}$ ,  
end configuration  $\lambda^{(\text{end})}$ , cost metric  $c$

$\lambda \leftarrow \lambda^{(\text{start})};$

$P \leftarrow [];$

**foreach**  $t \in \{1 \dots |\mathbf{\Lambda}|\}$  **do**

|

# Greedy Ablation Pseudo Code

---

## Algorithm 3 Greedy Ablation

---

**Input** : Algorithm  $\mathcal{A}$  with configuration space  $\mathbf{\Lambda}$ , start configuration  $\lambda^{(\text{start})}$ ,  
end configuration  $\lambda^{(\text{end})}$ , cost metric  $c$

$\lambda \leftarrow \lambda^{(\text{start})}$ ;

$P \leftarrow []$  ;

**foreach**  $t \in \{1 \dots |\mathbf{\Lambda}|\}$  **do**

**foreach**  $\delta \in \Delta(\lambda, \lambda^{(\text{end})})$  **do**

$\lambda'_\delta \leftarrow \text{apply } \delta \text{ to } \lambda$ ;

        evaluate  $c(\lambda'_\delta)$ ;

# Greedy Ablation Pseudo Code

---

**Algorithm 4** Greedy Ablation

---

**Input** : Algorithm  $\mathcal{A}$  with configuration space  $\Lambda$ , start configuration  $\lambda^{(\text{start})}$ , end configuration  $\lambda^{(\text{end})}$ , cost metric  $c$

$\lambda \leftarrow \lambda^{(\text{start})};$

$P \leftarrow [];$

**foreach**  $t \in \{1 \dots |\Lambda|\}$  **do**

**foreach**  $\delta \in \Delta(\lambda, \lambda^{(\text{end})})$  **do**

$\lambda'_\delta \leftarrow \text{apply } \delta \text{ to } \lambda;$

        evaluate  $c(\lambda'_\delta);$

    Determine most important change  $\delta^* \in \arg \min_{\delta \in \Delta(\lambda, \lambda^{(\text{end})})} c(\lambda_\delta);$

$\lambda \leftarrow \text{apply } \delta^* \text{ to } \lambda;$

$P.\text{append}(\delta^*);$

# Greedy Ablation Pseudo Code

---

**Algorithm 5** Greedy Ablation

---

**Input** : Algorithm  $\mathcal{A}$  with configuration space  $\Lambda$ , start configuration  $\lambda^{(\text{start})}$ ,  
end configuration  $\lambda^{(\text{end})}$ , cost metric  $c$

$\lambda \leftarrow \lambda^{(\text{start})};$

$P \leftarrow [];$

**foreach**  $t \in \{1 \dots |\Lambda|\}$  **do**

**foreach**  $\delta \in \Delta(\lambda, \lambda^{(\text{end})})$  **do**

$\lambda'_\delta \leftarrow \text{apply } \delta \text{ to } \lambda;$

        evaluate  $c(\lambda'_\delta);$

    Determine most important change  $\delta^* \in \arg \min_{\delta \in \Delta(\lambda, \lambda^{(\text{end})})} c(\lambda_\delta);$

$\lambda \leftarrow \text{apply } \delta^* \text{ to } \lambda;$

$P.\text{append}(\delta^*);$

**return** Ablation path  $P$

---

## Remarks on Ablation

- Even this greedy ablation requires  $\mathcal{O}(n^2)$  steps

# Remarks on Ablation

- Even this greedy ablation requires  $\mathcal{O}(n^2)$  steps
- ~→ We can also speedup that up by using surrogate models  
[Biedenkapp et al. 2017]

# Remarks on Ablation

- Even this greedy ablation requires  $\mathcal{O}(n^2)$  steps
- $\rightsquigarrow$  We can also speedup that up by using surrogate models  
[Biedenkapp et al. 2017]
- Common observations:
  - ① Some hyperparameters might not matter ( $\lambda_2$  in the example)



# Remarks on Ablation

- Even this greedy ablation requires  $\mathcal{O}(n^2)$  steps
- $\rightsquigarrow$  We can also speedup that up by using surrogate models  
[Biedenkapp et al. 2017]
- Common observations:
  - 1 Some hyperparameters might not matter ( $\lambda_2$  in the example)
  - 2 Often only a few of the hyperparameters have an big impact

# Remarks on Ablation

- Even this greedy ablation requires  $\mathcal{O}(n^2)$  steps
- $\rightsquigarrow$  We can also speedup that up by using surrogate models  
[Biedenkapp et al. 2017]
- Common observations:
  - 1 Some hyperparameters might not matter ( $\lambda_2$  in the example)
  - 2 Often only a few of the hyperparameters have an big impact
  - 3 You have plateaus in your ablation path because of interaction effects