Train In Data

Sequential Search

Hyperparameter response surface

Find the hyperparameters that minimize (or maximize) a performance metric.

$$
\begin{aligned}
\lambda^{(*)} &\approx \operatorname*{argmin}_{\lambda \in \Lambda} \operatorname*{mean}_{x \in \mathcal{X}^{(\text{valid})}} \mathcal{L}\left(x; \mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right). \\
&\equiv \operatorname*{argmin}_{\lambda \in \Lambda} \Psi(\lambda) \\
&\approx \operatorname*{argmin}_{\lambda \in \{\lambda^{(1)} \ldots \lambda^{(S)}\}} \Psi(\lambda) \equiv \hat{\lambda}
\end{aligned}
$$

Hyperparameter response surface

Find the hyperparameters that minimize (or maximize) a performance metric.

$$
\begin{aligned}
\lambda^{(*)} &\approx \underset{\lambda \in \Lambda}{\operatorname{argmin}} \underset{x \in \mathcal{X}^{(\text{valid})}}{\operatorname{mean}} \mathcal{L}\left(x; \mathcal{A}_\lambda(\mathcal{X}^{(\text{train})})\right). \\
&\equiv \underset{\lambda \in \Lambda}{\operatorname{argmin}} \Psi(\lambda) \\
&\approx \underset{\lambda \in \{\lambda^{(1)} \ldots \lambda^{(S)}\}}{\operatorname{argmin}} \Psi(\lambda) \equiv \hat{\lambda}
\end{aligned}
$$

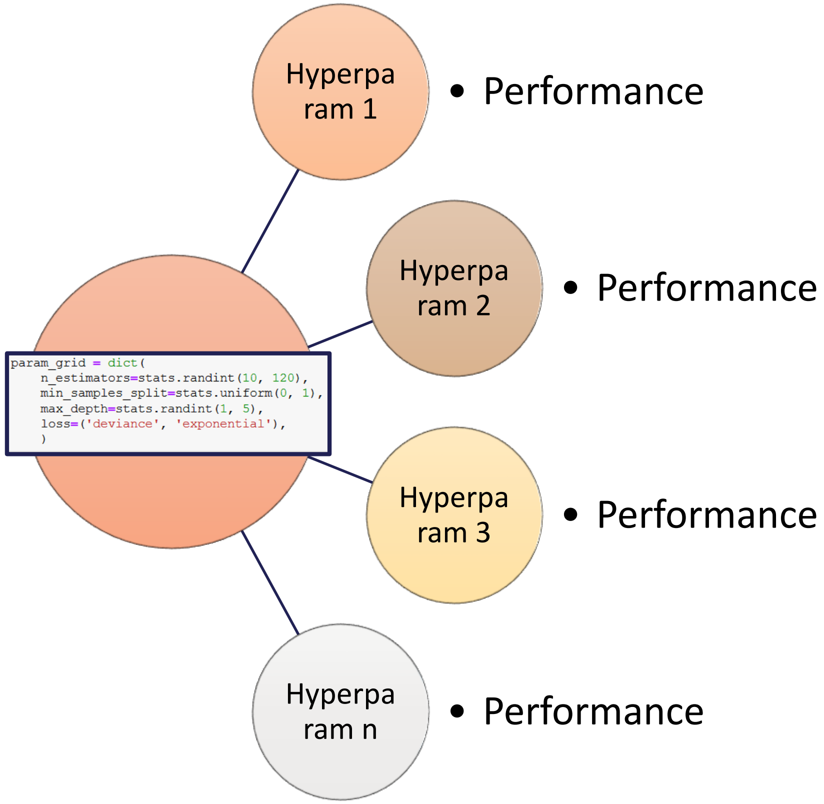
Response surface

- **A**: Algorithm
  - A_λ: Hyperparameters
- **X**: Dataset
- **L**: Function to optimise
  - Performance metric

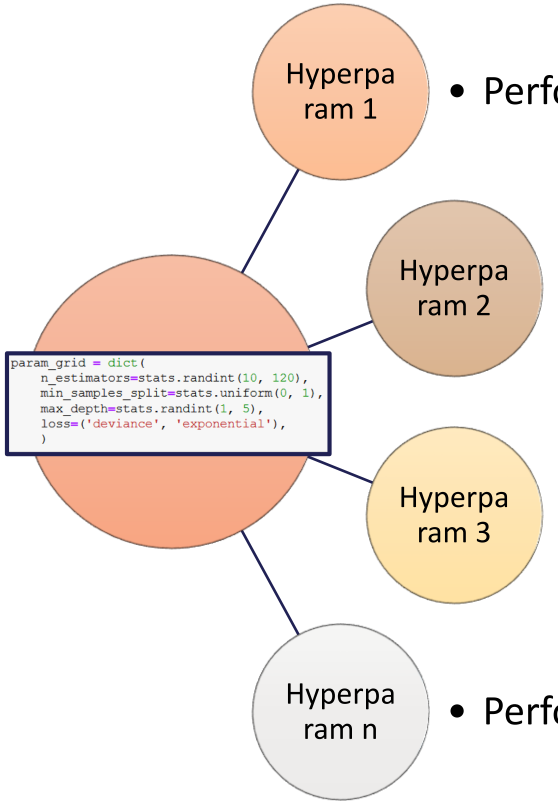
Hyperparameter Tuning: Challenges

- We can't define a formula to find the hyperparameters ➔ **black box** function
- The response surface is not differentiable ➔ it does not have a gradient
- Try different combinations of hyperparameter and evaluate model performance

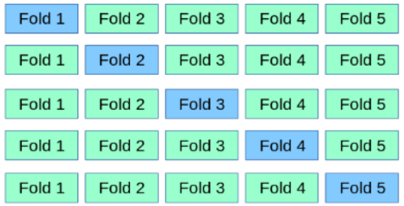
Hyperparameter Search



Hyperparameter Search

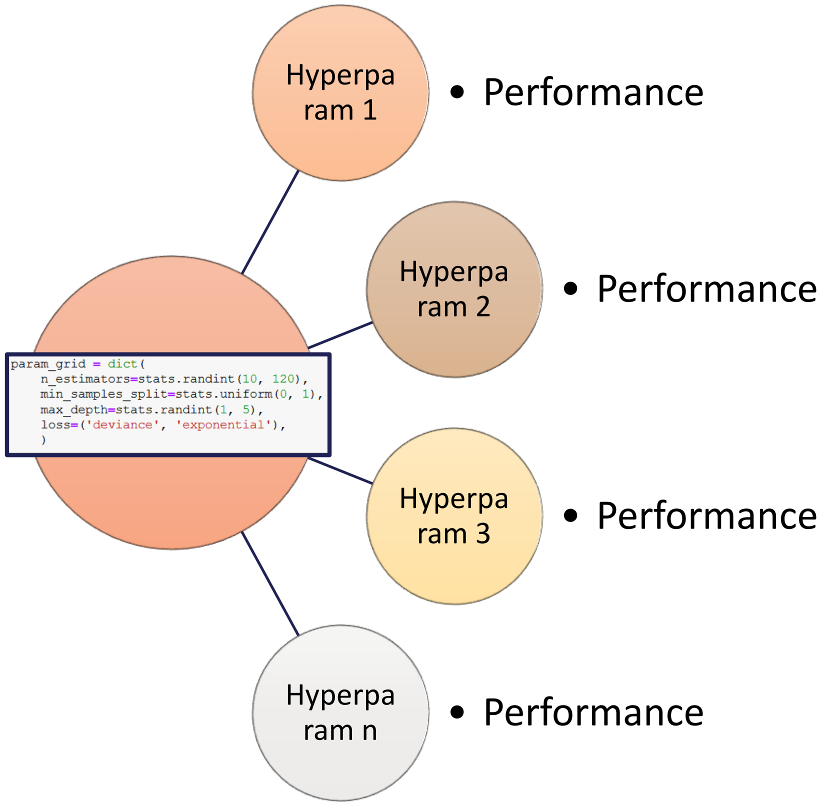


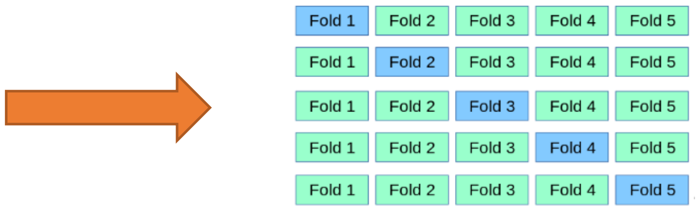
- Performance
- Performance
- Performance
- Performance



- Train K models
- Estimate K metrics

Hyperparameter Search



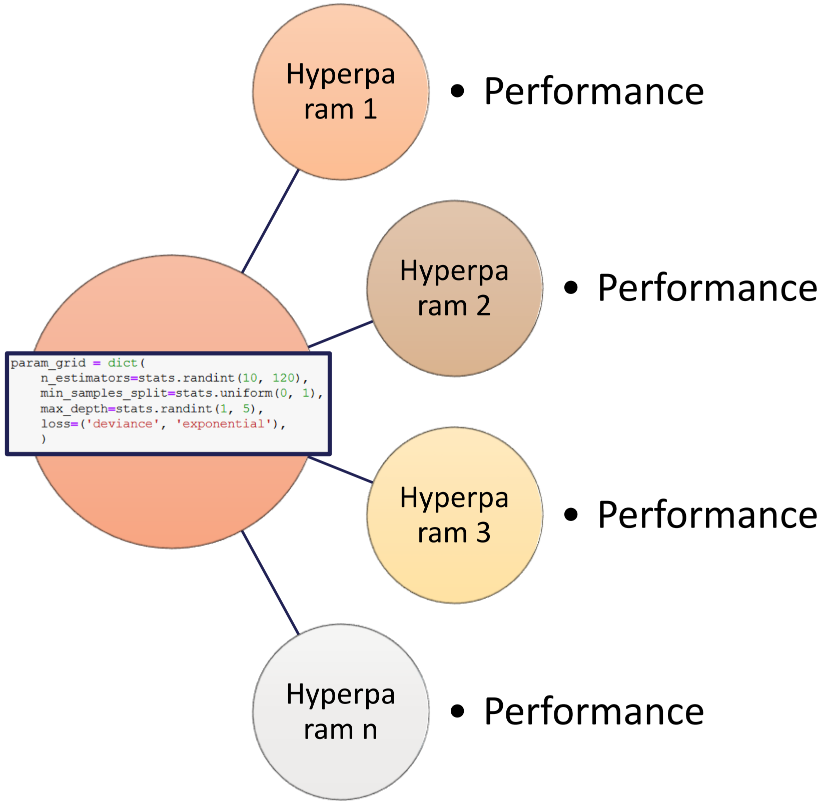


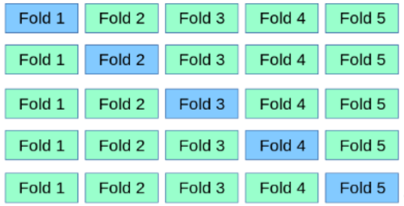
- Train K models
- Estimate K metrics

If models are simple (i.e., linear models, tree based algorithms):

- Grid Search
- Random Search

Hyperparameter Search



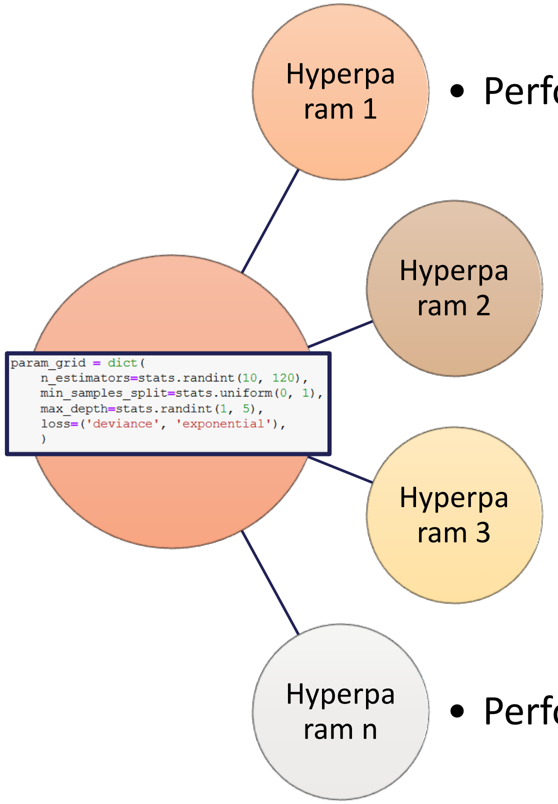


- Train K models
- Estimate K metrics

If models are complex (i.e., Neural Networks):

- Training the model is very costly (time and money).
- Trying all possible combinations is **not an option**

Hyperparameter Search



- Performance
- Performance
- Performance
- Performance

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

- Train K models
- Estimate K metrics

If models are complex (i.e., Neural Networks):

- Select smartly which hyperparameters we are going to evaluate.

Sequential Search

- Grid Search and Random Search generate all the candidate points up front and evaluate them in parallel.

- **Sequential search** techniques pick a few hyperparameter settings, evaluate their quality, then decide where to sample next.

  - ✓ Iterative and sequential process
  - ✓ Not parallelizable
  - ✓ Goal: make fewer evaluations, only of those most promising candidate hyperparameters

Sequential Search Trade-off

- Sequential search techniques pick a few hyperparameter settings, evaluate their quality, then **decide where to sample next**.

- Trade-off:
  - ➢ Less ML model training time × time to estimate where to sample next

- Sequential search makes sense when the **evaluation procedure** (training the model – performance) takes much longer than the process of **evaluating where to sample next**.

Bayesian Optimization

- **Bayesian optimization** is a sequential strategy for global optimization of black-box functions, which does not assume any functional forms.

- **Bayesian optimization** is usually employed to optimize expensive-to-evaluate functions.

THANK YOU

www.trainindata.com