

# **Introduction to Machine Learning**

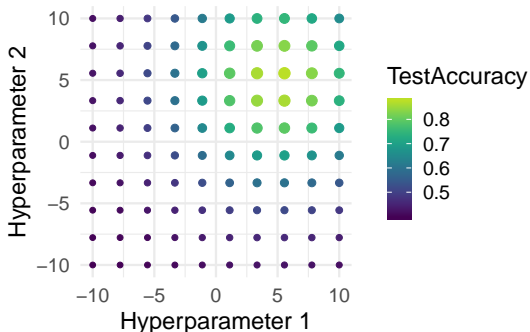
## **Hyperparameter Tuning - Techniques**

[compstat-lmu.github.io/lecture\\_i2ml](https://compstat-lmu.github.io/lecture_i2ml)

# GRID SEARCH

- Simple technique which is still quite popular, tries all HP combinations on a multi-dimensional discretized grid
- For each hyperparameter a finite set of candidates is predefined
- Then, we simply search all possible combinations in arbitrary order

Grid search over 10x10 points



# GRID SEARCH

## Advantages

- Very easy to implement
- All parameter types possible
- Parallelizing computation is trivial

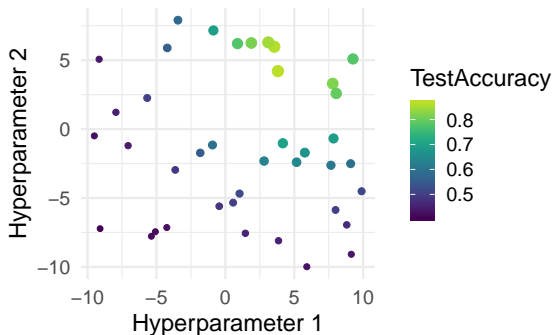
## Disadvantages

- Scales badly: Combinatorial explosion
- Inefficient: Searches large irrelevant areas
- Arbitrary: Which values / discretization?

# RANDOM SEARCH

- Small variation of grid search
- Uniformly sample from the region-of-interest

Random search over 100 points



# RANDOM SEARCH

## Advantages

- Like grid search: Very easy to implement, all parameter types possible, trivial parallelization
- Anytime algorithm: Can stop the search whenever our budget for computation is exhausted, or continue until we reach our performance goal.
- No discretization: each individual parameter is tried with a different value every time

## Disadvantages

- Inefficient: many evaluations in areas with low likelihood for improvement
- Scales badly: high dimensional hyperparameter spaces need *lots* of samples to cover.

# TUNING EXAMPLE

Tuning random forest with random search and 5CV on the sonar data set for AUC:

Parameter	Type	Min	Max
<code>num.trees</code>	integer	3	500
<code>mtry</code>	integer	5	50
<code>min.node.size</code>	integer	10	100

