Hyperparameter Tuning

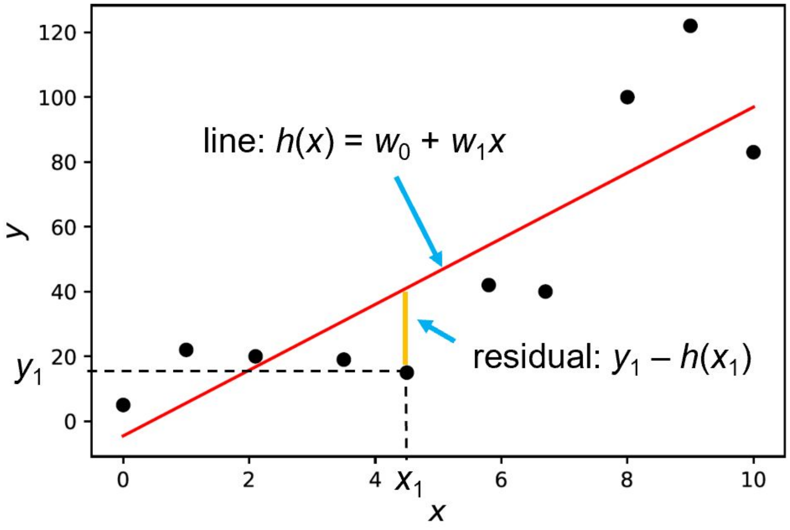
For Machine learning Algorithms

….

Parameters vs hyperparameters

Let's start simple: Model parameters in a linear model

- Parameters are being fit (i.e. found) during training.
- They are the **result** of model fitting or training.
- In a linear model, we want to find the **coefficients**.



Model parameters vs hyperparameters in a linear model

- **Remember**: model parameters are being fit (i.e. found) during training; they are the result of model fitting or training.
- Hyperparameters are being set before training.
- They specify HOW the training is supposed to happen.

Why tune hyperparameters?

- Fantasy football players ~ Hyperparameters
- Football players' positions ~ **Hyperparameter** values

- Finding the best combination of players and positions ~ Finding the best **combination** of hyperparameters

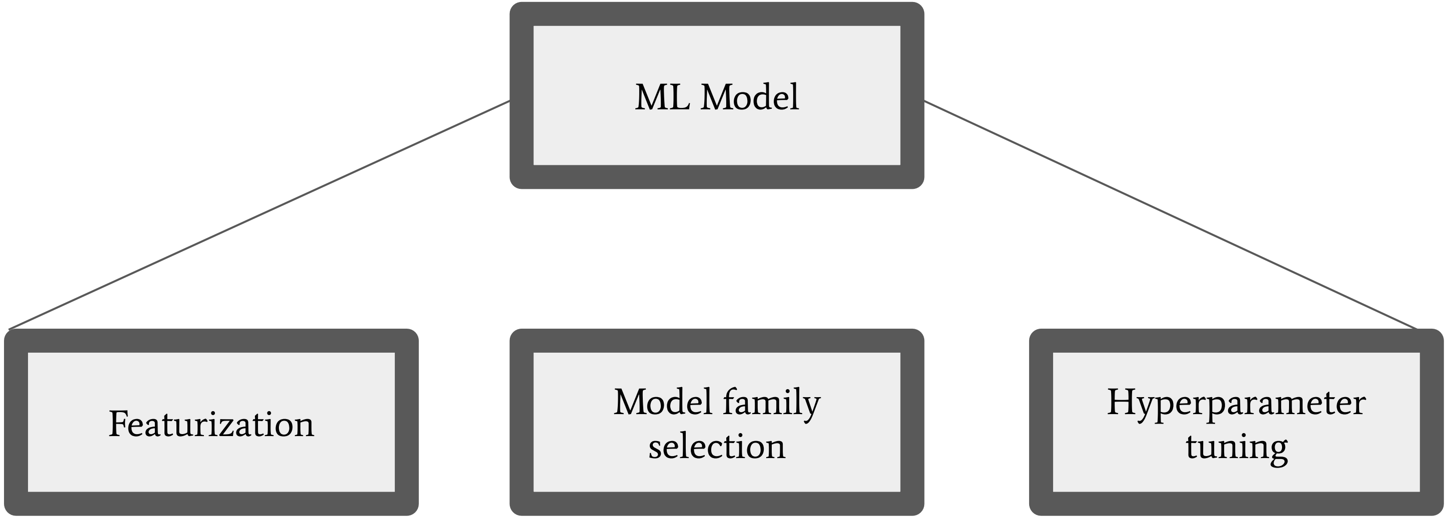
Hyperparameters

- Express high-level concepts, such as statistical assumptions
  - E.g.: regularization
- Are fixed before training or are hard to learn from data
  - E.g.: neural net architecture
- Affect objective, test time performance, computational cost
  - E.g.: # iterations or epochs

Challenges in Tuning

- Curse of dimensionality
- Non-convex optimization
- Computational cost
- Unintuitive hyperparameters
    - hyperparameters such as exploration percentage and batch size are more concrete, while others such as discounting factor and learning rate are a little less intuitive.

A Practical Definition of Tuning



Parameters: configs which your ML library learns from data

Hyperparameters: configs which your ML library does not learn from data

Tuning methods

Overview of tuning methods

- Manual search
- Random search
- Grid search
- Bayesian algorithms
- Population-based algorithms

Manual Search

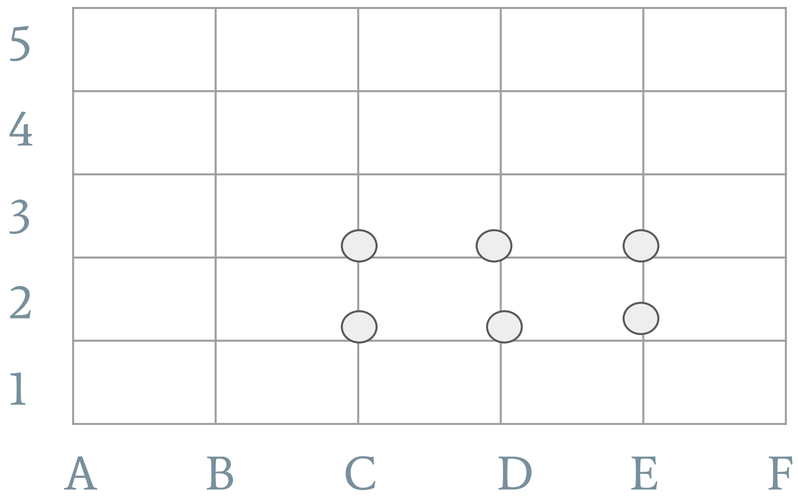
Select hyperparameter settings to try based on human intuition.

2 hyperparameters:

- [0,....,5]
- {A,B,....,F}

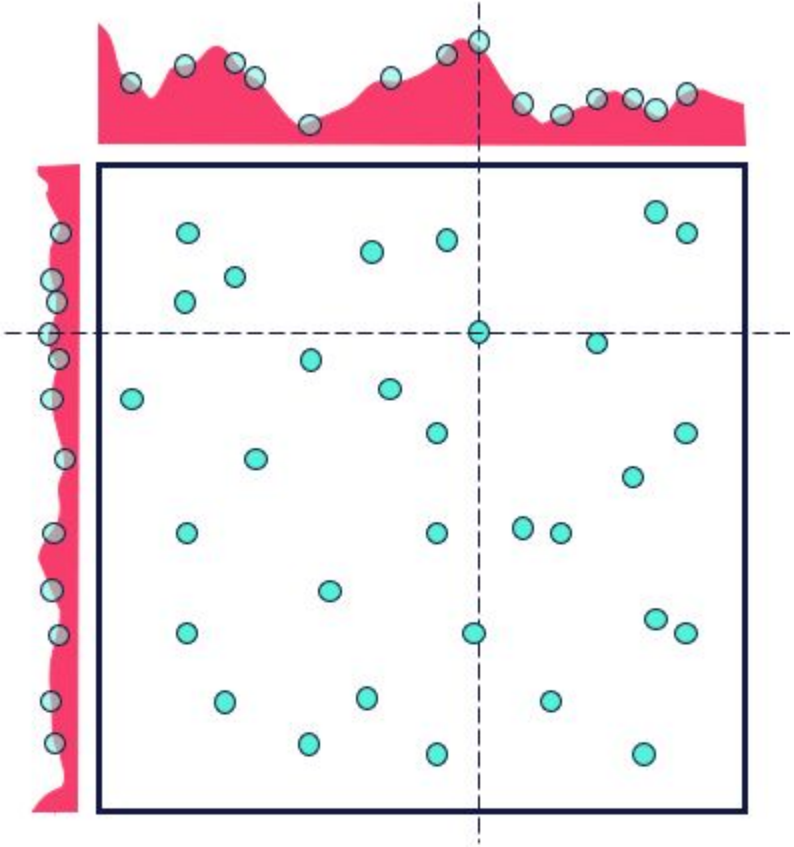
Expert knowledge tells us to try:

(2,C), (2,D), (2,E), (3,C), (3,D), (3,E)



Random Search

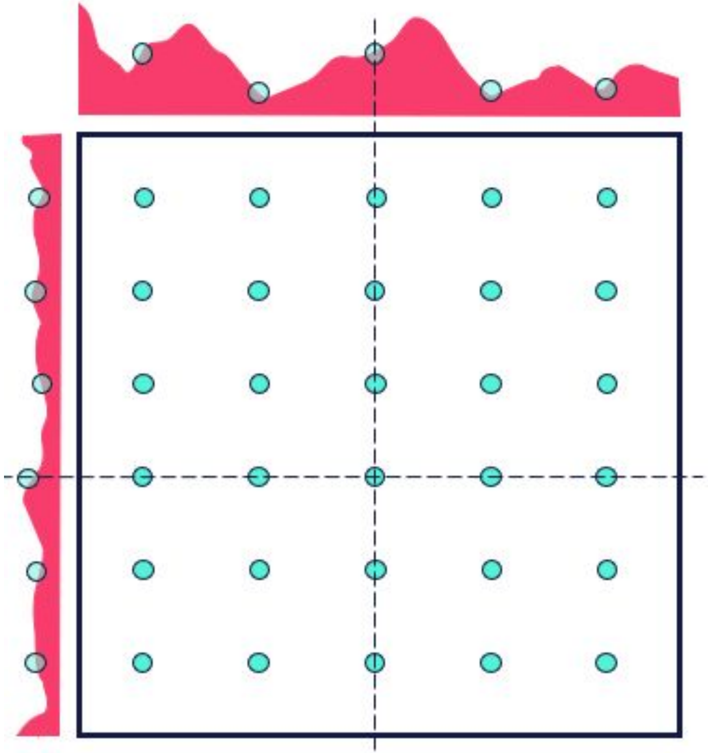
We **manually** set a **range of bounds** of the possible parameters and the algorithm makes a **search** over them for the number of iterations we set.



Random Search

Grid Search

- Set parameter ranges manually for algorithm exploration.
- Exhaustive Search Technique:
  - Implement a grid search for a thorough exploration of specified parameter ranges.
- Brute-Force Approach:
  - Understand that grid search employs a complete brute-force method.
- Execution Time Awareness:
  - Be mindful that this exhaustive process may result in longer execution times.



Grid Search

Bayesian optimization

- Statistical approach for **minimizing noisy black-box functions.**
- Idea: learn a **statistical model of the function** from hyperparameter values to the loss function
  - Then choose parameters to minimize the loss under this model
- Main benefit: choose the hyperparameters to test not at random, but in a way that gives the **most information about the model.**
  - This lets it learn faster than grid search

Effect of Bayesian Optimization

- Downside: it’s a pretty heavyweight method
    - The updates are not as simple-to-implement as grid search
- Upside: empirically it has been demonstrated to get better results in fewer experiments
    - Compared with grid search and random search
- Pretty widely used method
    - Lots of research opportunities here

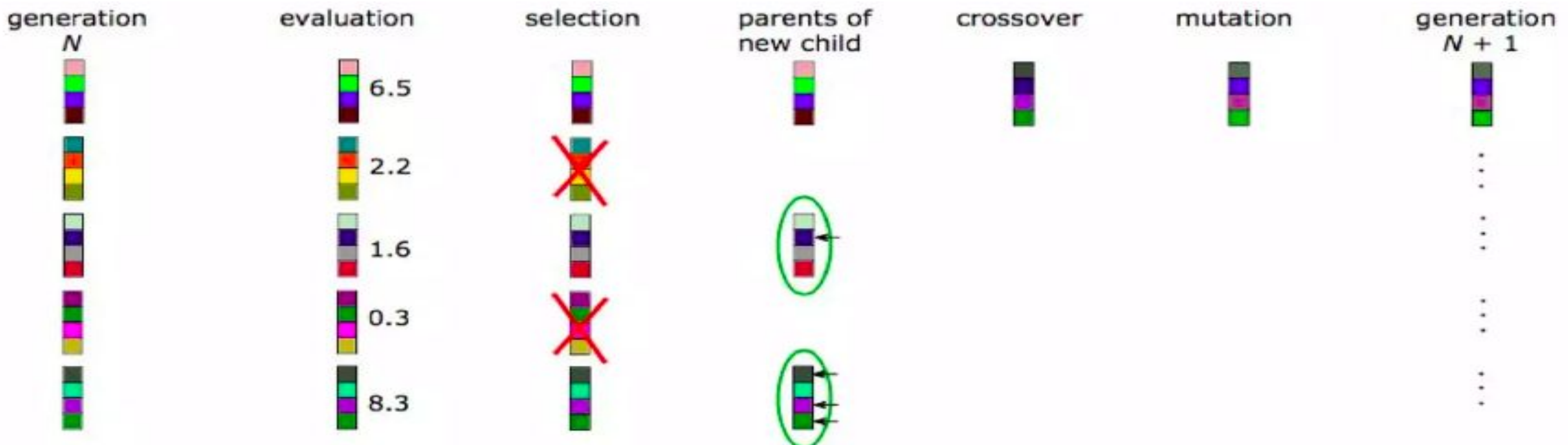
Bayesian Approach

- Make assumption on F(x) we want to maximize: F(x) is a weakly-stationary Gaussian process.
- Start with a few points randomly sampled.
- For each new evaluation, update your prior knowledge on F(x) to get a posterior.
- Using the posterior, decide which point 1= hyperpa ram combination) to try next.

- Using the posterior to decide what to try next: Exploration VS Exploitation
  - Exploration = sample point where posterior variance is largest, i.e. we know the least on F(x)
  - Exploitation = sample point where posterior mean is highest, i.e. where we expect F(x) to be max
  - Usually, some knobs define in the Bayesian optimizer how to tune exploration vs exploitation ...

Genetic Algorithm

1. At every iteration consider a population, called generation, consisting of M individuals x,
2. For every individual, evaluate its fitness function F(x) (= model accuracy, in our case)
3. Some individuals are selected to reproduce, with Prob(x is selected) proportional to the fitness F(x)
4. A new generation is produced from selected individuals: for child = 1 ... M, choose 2 parents
   a. child's genotype (= components of x) is generated by a random crossover of parents' genotype
   b. child's genotype is randomly modified by a mutation



Open source tools for tuning

	Grid search	Random search	Population -based	Bayesian	PyPi downloads last month	Github stars	License
scikit-learn	Yes	Yes			---	---	BSD
MLlib	Yes				---	---	Apache 2.0
scikit-optimize				Yes	49,189	1,278	BSD
Hyperopt		Yes		Yes	98,282	3,286	BSD
DEAP			Yes		26,700	2,789	LGPL v3
TPOT			Yes		9,057	5,609	LGPL v3
GPyOpt				Yes	4,959	451	BSD