

# High-performance computing considerations in hyperparameter search

Billy Braithwaite

CSC - IT Center for Science

January 21, 2022

# The new fad: machine learning



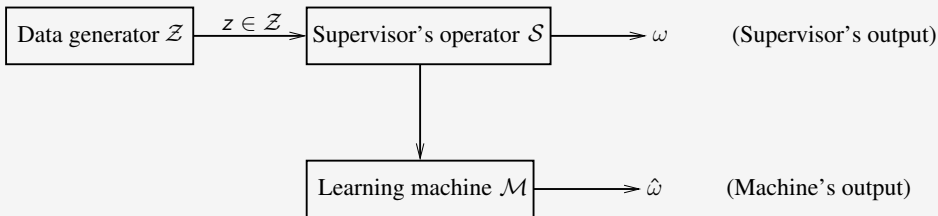
Rebranding of particular statistical problems:

- Statistical learning theory (SLT):  $(\mathcal{Z}_{\text{data}}, \omega_{\text{labels}})$ 
  - Pattern recognition
  - Regression
- Self-organization<sup>a</sup>:  $(\mathcal{Z}_{\text{data}})$ 
  - Clustering(ish)
- Reinforcement learning:  $(\mathcal{Z}_{\text{data}}, \phi(\cdot))$ 
  - Markovian Decision Process
  - Robotics

---

<sup>a</sup>Unsupervised learning

# SLT: an indirect attack to statistical inference via examples



# Problem of model selection

## Model selection

Find  $\{\mathcal{M}_i\}_i^N$  such that given metric  $\mathcal{L}(f(\mathcal{Z}), \hat{\omega})$  is minimized/maximized

## Model sampling

Find  $\mathcal{M}$  under various sampling conditions

- Bootstrapping
- Cross-validation

## Hyperparameter search

Find  $\mathcal{M}$  with different model configurations

- #Layers in a Neural Network
- Margin threshold in SVM

# Grid and Random Search

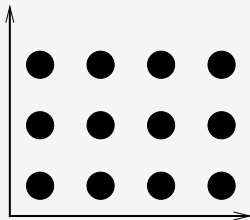


Figure: Grid search

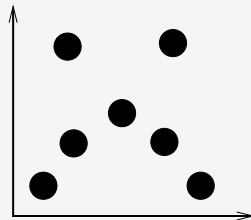


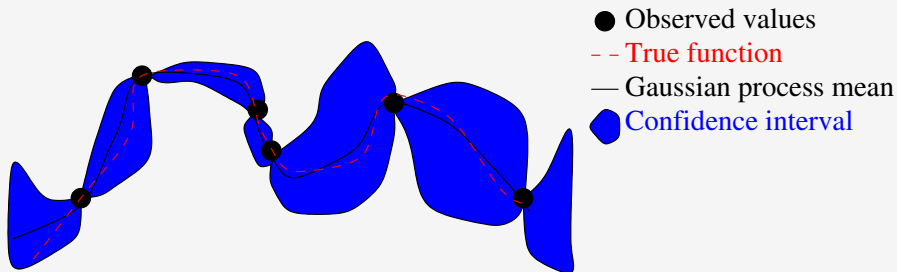
Figure: Random search

## HPC aspect

Embarrassingly parallel: each point evaluated independently.

- Small, limited number of hyperparameters
- Well suited for categorical hyperparameters (solver type, loss function, solver constraints)

# Adaptive (Bayesian) Search



## HPC aspect

$\mathcal{M}$  evaluation sequential.

- Search space may be partitioned
- Multifidelity search

# Asynchronous Successive Halting<sup>1</sup>

## Algorithm 2 Asynchronous Successive Halving (ASHA)

```

input minimum resource  $r$ , maximum resource  $R$ , reduction factor  $\eta$ , minimum early-stopping rate  $s$ 
function ASHA()
  repeat
    for for each free worker do
       $(\theta, k) = \text{get\_job}()$ 
       $\text{run.then.return.val.loss}(\theta, r\eta^{s+k})$ 
    end for
    for completed job  $(\theta, k)$  with loss  $l$  do
      Update configuration  $\theta$  in rung  $k$  with loss  $l$ .
    end for
  until desired
end function

function get\_job()
  // Check if there is a promotable config
  for  $k = \lfloor \log_\eta(R/r) \rfloor - s - 1, \dots, 1, 0$  do
    candidates = top.k(rung  $k$ ,  $\lfloor \text{rung } k / \eta \rfloor$ )
    promotable =  $\{t \in \text{candidates} : t \text{ not promoted}\}$ 
    if  $|\text{promotable}| > 0$  then
      return promotable[0],  $k + 1$ 
    end if
  // If not, grow bottom rung.
  Draw random configuration  $\theta$ .
  return  $\theta, 0$ 
  end for
end function

```

## Principle of ASHA

- Each configuration is ranked w.r.t loss and resources
- Highest ranked configurations promoted to the next iteration.
- Configurations promoted when available.

<sup>1</sup>Li, Liam, et al. "A system for massively parallel hyperparameter tuning."

# Problem with model selection

## Statistical

- A large number of model evaluations may lead to overfitting.
- Model comparison using loss function.
- Criteria for optimality in a given dataset?

## Computational

- Efficient partition of search space.
- Focus on efficient evaluation of  $\mathcal{M}$ ?
- Programming language constraints (Python's GIL; C, C++, Julia may be cumbersome to implement)



## Q&amp;A

