I2ML :: CHEAT SHEET

The **I2ML**: Introduction to Machine Learning course offers an introductory and applied overview of "supervised" Machine Learning. It is organized as a digital lecture.

Hyperparameter Tuning

Hyperparameter Tuning:

Hyperparameters $\boldsymbol{\lambda}$: parameters that are *inputs* to the training problem, in which a learner $\mathcal{I}$ minimizes the empirical risk on a training data set to find optimal **model parameters** $\boldsymbol{\theta}$ that define fitted model $\hat{f}$.
Hyperparameter optimization (HPO) / Tuning is the process of finding a well-performing hyperparameter configuration (HPC) $\boldsymbol{\lambda} \in \tilde{\Lambda}$ for an learner $\mathcal{I}_{\boldsymbol{\lambda}}$.
The general HPO problem is defined as:

$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \tilde{\Lambda}} c(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\lambda} \in \tilde{\Lambda}} \widehat{\mathrm{GE}}(\mathcal{I}, \mathcal{J}, \rho, \boldsymbol{\lambda}),$$

with $\boldsymbol{\lambda}^*$ as theoretical optimum, and $c(\boldsymbol{\lambda})$ is short for estim. gen. error when $\mathcal{I}$, resampling splits $\mathcal{J}$, performance measure ρ are fixed.

Components of a tuning problem:

The dataset, the learner (tuned), the learner's hyperparameters and their respective regions-of-interest over which optimization is done, the performance measure, a (resampling) procedure for estimating the predictive performance.

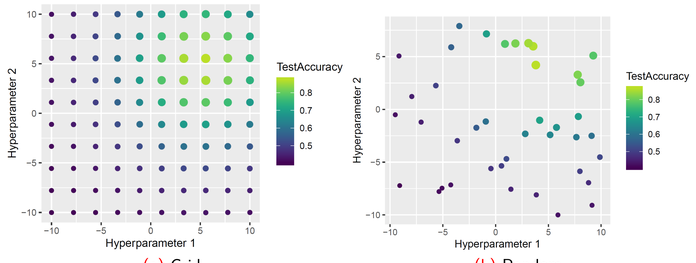
Tuning is hard:

Tuning is derivative-free which is a black-box problem. Every evaluation is **expensive** and the answer we get from that evaluation is **not exact, but stochastic** in most settings. The space of hyperparameters we optimize over has a non-metric, complicated structure.

Basic Techniques

Grid Search:

For each hyperparameter a finite set of candidates is predefined and searches all possible combinations in arbitrary order.



(a) Grid (b) Random

Random Search:

Small variation of Grid Search. Uniformly sample from the region-of-interest.
Note: Both grid and random search are very easy to implement, all parameter types possible, have trivial parallelization. However, both are also inefficient and scale badly.

Bayesian Optimization (BO):

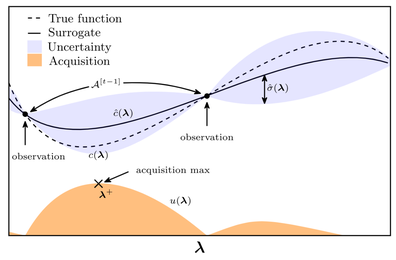
BO sequentially iterates:

1. **Approximate** $\boldsymbol{\lambda} \mapsto c(\boldsymbol{\lambda})$ by (nonlin) regression model $\hat{c}(\boldsymbol{\lambda})$, from evaluated configurations (archive)
2. **Propose candidates** via optimizing an acquisition function that is based on the surrogate $\hat{c}(\boldsymbol{\lambda})$
3. **Evaluate** candidate(s) proposed in 2, then go to 1

Important trade-off: **Exploration** (evaluate candidates in under-explored areas) vs. **exploitation** (search near promising areas)

Surrogate Model:

- Probabilistic modeling of $C(\boldsymbol{\lambda}) \sim (\hat{c}(\boldsymbol{\lambda}), \hat{\sigma}(\boldsymbol{\lambda}))$ with posterior mean $\hat{c}(\boldsymbol{\lambda})$ and uncertainty $\hat{\sigma}(\boldsymbol{\lambda})$.
- Typical choices for numeric spaces are Gaussian Processes; random forests for mixed spaces



Acquisition Function:

- Balance exploration (high $\hat{\sigma}$) vs. exploitation (low $\hat{c}$).
- Lower confidence bound (LCB): $a(\boldsymbol{\lambda}) = \hat{c}(\boldsymbol{\lambda}) - \kappa \cdot \hat{\sigma}(\boldsymbol{\lambda})$
- Expected improvement (EI): $a(\boldsymbol{\lambda}) = \mathbb{E}\left[\max\left\{c_{\min} - C(\boldsymbol{\lambda}), 0\right\}\right]$

where ($c_{\min}$ is best cost value from archive)

- Optimizing $a(\boldsymbol{\lambda})$ is still difficult, but cheap(er)

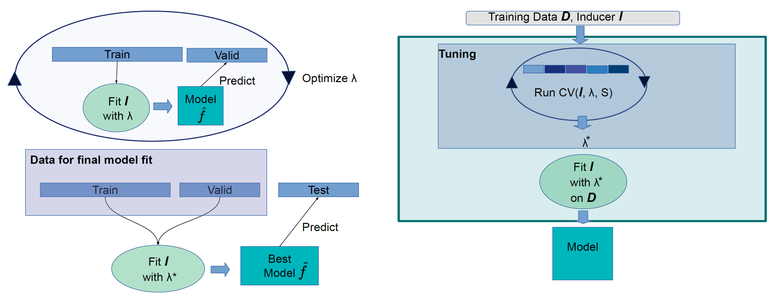
Nested Resampling

Problem of Tuning

Need to **select an optimal learner** without compromising the **accuracy of the performance estimate** for that learner. For this **untouched test set** is needed.

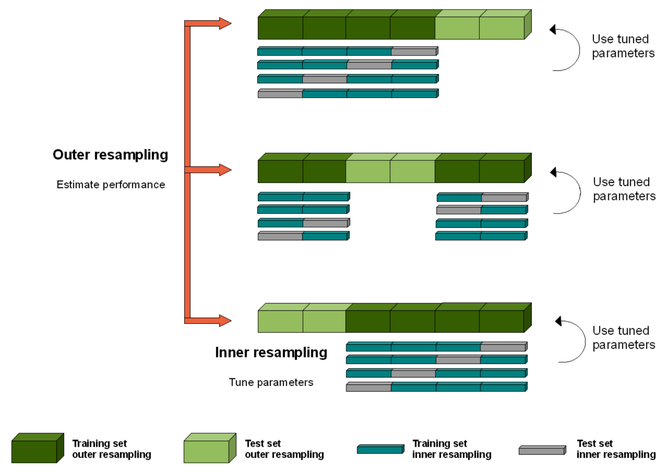
Train-validation-test

A 3-way split is the simplest method: During tuning, a learner is trained on the **training set**, evaluated on the **validation set** and after the best model configuration λ^* is selected, we re-train on the joint (training+validation) set and evaluate the model's performance on the **test set**.



If we want to tune over a set of candidate HP configurations λ_i; $i = 1, \ldots$ with 4-fold CV in the inner resampling and 3-fold CV in the outer loop. The outer loop is visualized as the light green and dark green parts.

Nested Resampling



The outer loop is visualized as the light green and dark green parts. This is with 4-fold CV in the inner resampling and 3-fold CV in the outer loop.