I2ML :: EVALUATION AND TUNING

Set-Based Performance Metrics

$J \in \{1, \dots, n\}^m$: m-dimensional index vector for a dataset $\mathcal{D} \in \mathbb{D}_n$, which also induces $\mathcal{D}_J = \left(\mathcal{D}^{(J^{(1)})}, \dots, \mathcal{D}^{(J^{(m)})}\right) \in \mathbb{D}_m$

$\mathbf{y}_J = \left(y^{(J^{(1)})}, \dots, y^{(J^{(m)})}\right) \in \mathcal{Y}^m$: vector of labels

$\mathbf{F}_{J,f} = \left(f(\mathbf{x}^{(J^{(1)})}), \dots, f(\mathbf{x}^{(J^{(m)})})\right) \in \mathbb{R}^{m \times g}$: matrix of prediction scores regarding a model f

General **performance measure**: $\rho : \bigcup_{m \in \mathbb{N}} \left(\mathcal{Y}^m \times \mathbb{R}^{m \times g}\right) \to \mathbb{R}$ maps every m-dimensional label vector $\mathbf{y}_J$ and its matrix of prediction scores $\mathbf{F}_{J,f}$ to a scalar performance value.

$\rho_L(\mathbf{y}, \mathbf{F}) = \sum_{i=1}^m L(y^{(i)}, \mathbf{F}^{(i)})$: performance measure induced by an arbitrary point-wise loss L

P : set of all performance measures ρ

Generalization Error

The **generalization error** GE is the performance of a model induced by $\mathcal{I}_{\boldsymbol{\lambda}}$ from datasets $\mathcal{D}_{\text{train}} \sim (\mathbb{P}_{xy})^{m_{\text{train}}}$ evaluated with performance measure ρ over a dataset $\mathcal{D}_{\text{test}} \sim (\mathbb{P}_{xy})^{m_{\text{test}}}$ when $m_{\text{test}} \to \infty$, i.e.,

$$\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, m_{\text{train}}, \rho) = \lim_{m_{\text{test}} \to \infty} \mathbb{E}\left[\rho\left(\mathbf{y}, \mathbf{F}_{J_{\text{test}}, f_{\mathcal{D}_{\text{train}}, \boldsymbol{\lambda}}}\right)\right],$$

where $f_{\mathcal{D}_{\text{train}}, \boldsymbol{\lambda}} = \mathcal{I}(\mathcal{D}_{\text{train}}, \boldsymbol{\lambda})$ and the expectation is taken over both datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}} (= \mathcal{D}_{J_{\text{test}}})$.

Data Splitting and Resampling

$\mathcal{S} = ((J_{\text{train},1}, J_{\text{test},1}), \dots, (J_{\text{train},K}, J_{\text{test},K})) \in \mathbb{S}_K$: **resampling strategy** consisting of K train-test-splits $(J_{\text{train},i}, J_{\text{test},i})$

Estimator of the generalization error $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, m_{\text{train}}, \rho)$:

$$\widehat{\mathrm{GE}}_{\mathcal{S}}(\mathcal{I}, \boldsymbol{\lambda}, \rho) = \mathrm{agr}\Big(\rho\left(\mathbf{y}_{J_{\text{test},1}}, \mathbf{F}_{J_{\text{test},1}, f_{\mathcal{D}_{\text{train},1}, \boldsymbol{\lambda}}}\right), \\ \vdots \\ \rho\left(\mathbf{y}_{J_{\text{test},K}}, \mathbf{F}_{J_{\text{test},K}, f_{\mathcal{D}_{\text{train},K}, \boldsymbol{\lambda}}}\right)\Big),$$

where the aggregating function agr is often the mean and $m_{\text{train}} \approx m_{\text{train},1} \approx \dots \approx m_{\text{train},K}$ and $m_{\text{train}} = \mathrm{mode}(m_{\text{train},1}, \dots, m_{\text{train},K})$

Resampling Strategies

K-fold cross-validation : splits the data into K roughly equally-sized partitions. Uses each part once as test set and joins the others for training

Leave-one-out cross validation : n-fold cross-validation

Repeated **subsampling** / Monte Carlo cross-validation : for $p \in (0, 1)$ randomly draws K training sets of size $\lfloor p \cdot n \rfloor$ without replacement from the data and uses the data not drawn as the corresponding K test sets

Bootstrap sampling : Similar to repeated subsampling but the training data is randomly drawn with replacement

Tuning

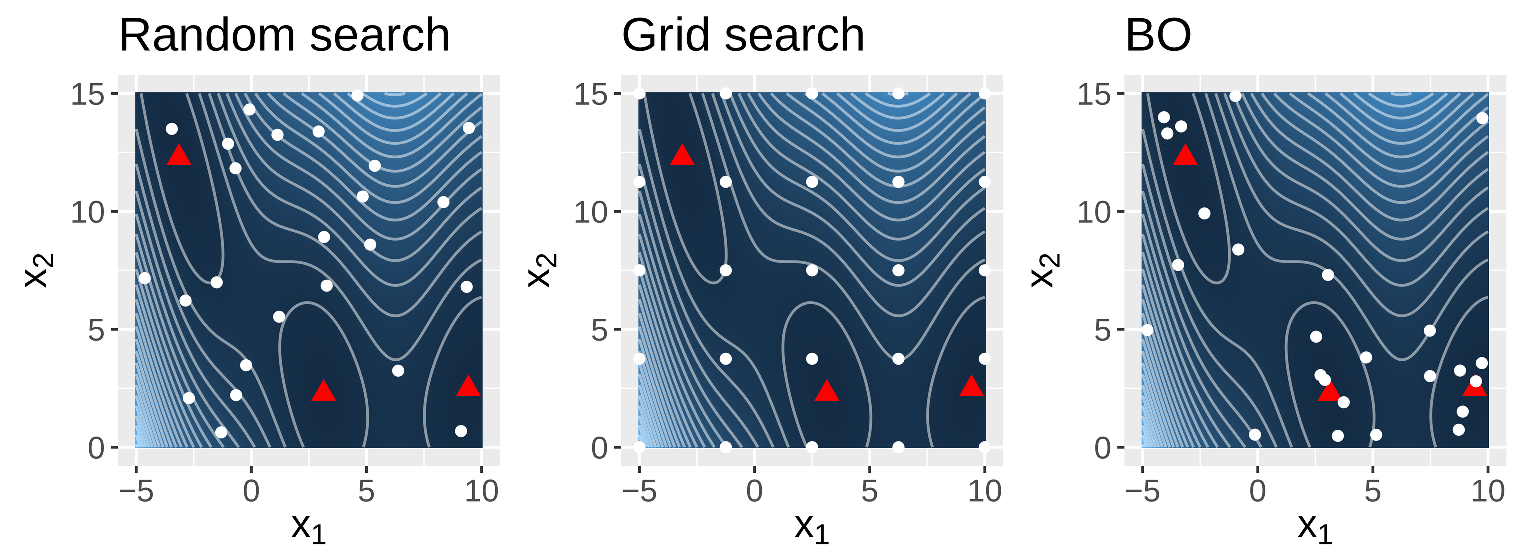
Tuner $\mathcal{T} : \mathbb{D} \times \mathbb{S}_K \times \{\mathcal{I} : \mathcal{I} \text{ is a learner}\} \times \mathrm{P} \to \mathcal{H}$ takes a dataset $\mathcal{D} \in \mathbb{D}$, and produces a model f learned with learner $\mathcal{I}_{\widehat{\boldsymbol{\lambda}}^*}$ where

$$\widehat{\boldsymbol{\lambda}}^* \approx \boldsymbol{\lambda}^* = \operatorname*{arg\,min}_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \widehat{\mathrm{GE}}_{\mathcal{S}}(\mathcal{I}, \boldsymbol{\lambda}, \rho), \text{ i.e.,}$$

the optimal hyperparameter regarding the performance measure $\rho \in \mathrm{P}$ and the resampling strategy $\mathcal{S} \in \mathbb{S}_K$ defined on $\mathcal{D}$. Here $\boldsymbol{\Lambda}$ denotes the search space which is a bounded subset of the hyperparameter space.

For fixed resampling strategy $\mathcal{S}$, learner $\mathcal{I}$ and performance measure ρ a learner $\mathcal{T}_{\mathcal{S},\mathcal{I},\rho}$ can be derived from a tuner $\mathcal{T}$. Possible hyperparameters of $\mathcal{T}_{\mathcal{S},\mathcal{I},\rho}$, so-called strategy parameters, are for simplicity's sake suppressed in this notation.

Black-Box Optimization Techniques



Random search : samples uniformly candidates from the search space

Grid search : (uniformly) discretizes the search space and samples without replacement candidates from it

Bayesian optimization : continously learns a surrogate model of the objective function and levarages it to sample new candidates from the search space while balancing exploration and exploitation

Evolutionary algorithms : are stochastic optimization methods that aim to solve optimization problems by using ideas of natural evolution

Hyperband : is a multi-fidelity method which tries to allocates more budget to promising candidates based on Successive Halving

Nested Resampling

$\mathcal{S}_{B,K} = \left(\mathcal{S}_{\text{outer}}, \left(\mathcal{S}_{\text{inner}}^{(1)}, \dots, \mathcal{S}_{\text{inner}}^{(B)}\right)\right)$: **nested resampling strategy** where $\mathcal{S}_{\text{outer}} \in \mathbb{S}_B$ defined on $\mathcal{D}$ and $\mathcal{S}_{\text{inner}}^{(i)} \in \mathbb{S}_K$ defined on $\mathcal{D}_{\text{outer,train,i}}$

Estimator of the generalization error $\mathrm{GE}(\mathcal{T}_{\mathcal{S},\mathcal{I},\rho}, m_{\text{train}}) = \mathrm{GE}(\mathcal{I}, \widehat{\boldsymbol{\lambda}}^*, m_{\text{train}}, \rho)$:

$$\widehat{\mathrm{GE}}_{\mathcal{S}_{B,K}}(\mathcal{T}_{\mathcal{S},\mathcal{I},\rho}) = \mathrm{agr}\Big(\rho\left(\mathbf{y}_{J_{\text{outer,test},1}}, \mathbf{F}_{J_{\text{outer,test},1}, f_{\mathcal{D}_{\text{outer,train},1}}}\right), \\ \vdots \\ \rho\left(\mathbf{y}_{J_{\text{outer,test},B}}, \mathbf{F}_{J_{\text{outer,test},B}, f_{\mathcal{D}_{\text{outer,train},B}}}\right)\Big),$$

where $f_{\mathcal{D}_{\text{outer,train},i}} = \mathcal{T}_{\mathcal{S}_{\text{inner}}^{(i)}, \mathcal{I}, \rho}(\mathcal{D}_{\text{outer,train},i})$ and $\mathcal{S}_{\text{inner}}^{(i)}$ has the same type of resampling strategy as $\mathcal{S}$ and $m_{\text{train}} \approx m_{\text{outer,train},1} \approx \dots \approx m_{\text{outer,train},B}$ and $m_{\text{train}} = \mathrm{mode}(m_{\text{outer,train},1}, \dots, m_{\text{outer,train},B})$