I2ML :: EVALUATION AND TUNING

Set-Based Performance Metrics

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

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

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

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

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

Generalization Error

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

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

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

Data Splitting and Resampling

$\mathcal{S} = ((J_{\mathrm{train},1}, J_{\mathrm{test},1}), \ldots, (J_{\mathrm{train},K}, J_{\mathrm{test},K}))$: **resampling strategy** consisting of K train-test-splits $(J_{\mathrm{train},i}, J_{\mathrm{test},i})$

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

$$\begin{aligned}\widehat{\mathrm{GE}}_{\mathcal{S}}(\mathcal{I}, \boldsymbol{\lambda}, \rho) = \mathrm{agr}\Big(&\rho\Big(\mathbf{y}_{J_{\mathrm{test},1}}, \boldsymbol{F}_{J_{\mathrm{test},1}, f_{\mathcal{D}_{\mathrm{train},1}, \boldsymbol{\lambda}}}\Big), \\ &\vdots \\ &\rho\Big(\mathbf{y}_{J_{\mathrm{test},K}}, \boldsymbol{F}_{J_{\mathrm{test},B}, f_{\mathcal{D}_{\mathrm{train},K}, \boldsymbol{\lambda}}}\Big)\Big),\end{aligned}$$

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

Resampling Strategies

Cross-validation

Leave-one-out cross validation

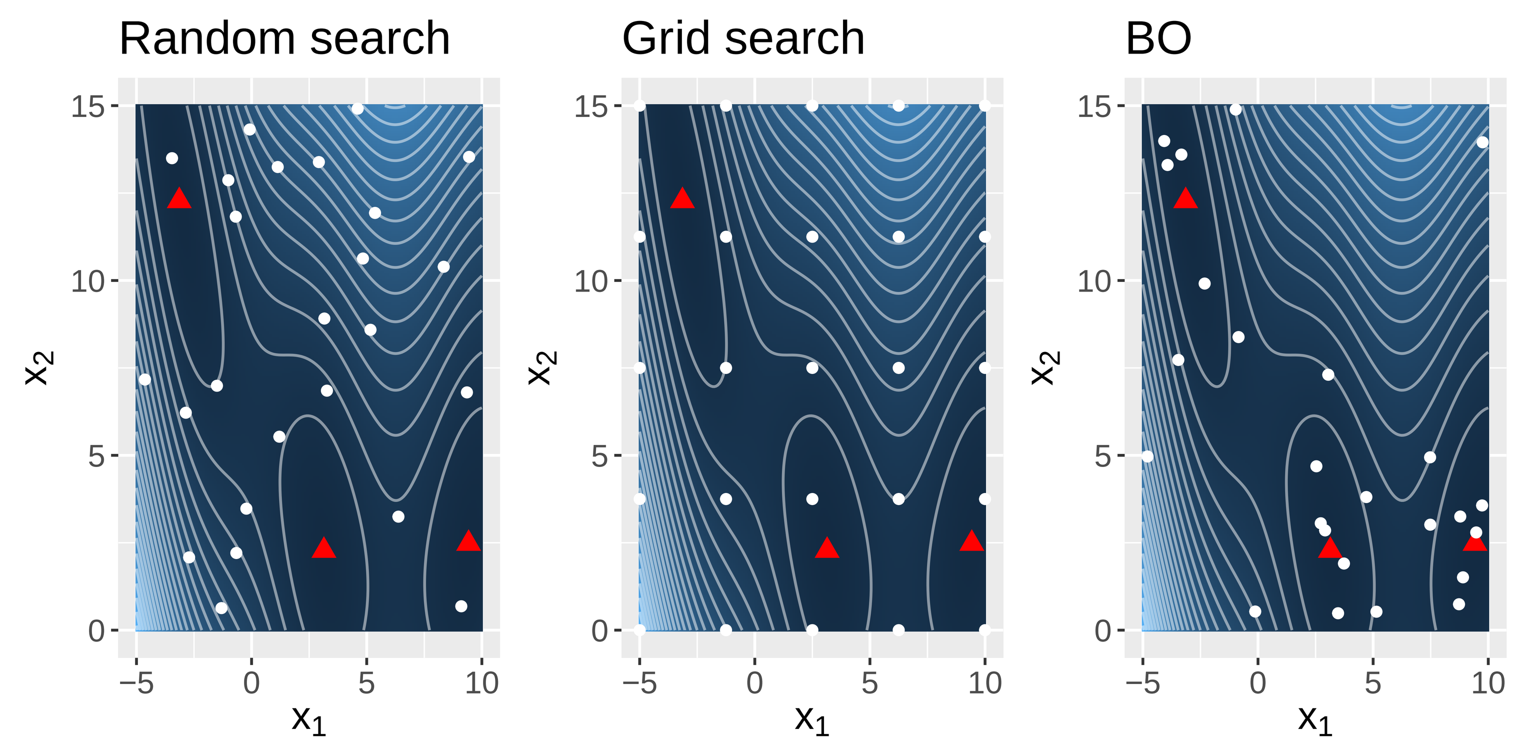
Repeated **subsampling** / Monte Carlo cross-validation

Bootstrap sampling

Tuning

$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \widehat{\mathrm{GE}}_{\mathcal{S}}(\mathcal{I}, \boldsymbol{\lambda}, \rho)$: optimal hyperparameter

Black-Box Optimization Techniques



Grid search

Random search

Nested Resampling