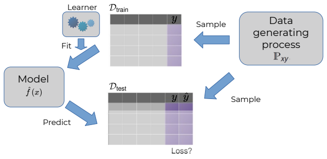
Introduction to Machine Learning

Evaluation

In a nutshell

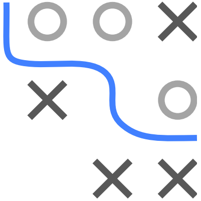


Learner
$\mathcal{D}_{\text{train}}$
y
Sample
Data generating process $\mathbb{P}_{xy}$
Fit
Model $\hat{f}(x)$
$\mathcal{D}_{\text{test}}$
y $\hat{y}$
Sample
Predict
Loss?

Learning goals

- Understand what the Generalization Error is
- Get an overview on how we evaluate performance of learners
- Learn about some evaluation metrics
- Understand why we do resampling

ESTIMATING THE GENERALIZATION ERROR



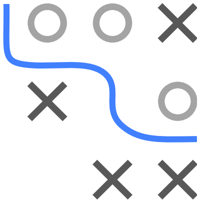
- For a fixed model, we are interested in the Generalization Error (GE): $\mathrm{GE}\left(\hat{f}, L\right) := \mathbb{E}\left[L\left(y, \hat{f}(\mathbf{x})\right)\right]$, i.e. the expected error the model makes for data $(\mathbf{x}, y) \sim \mathbb{P}_{xy}$.
- We need an estimator for the GE with m test observations:

$$\widehat{\mathrm{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x},y)} \left[L\left(y, \hat{f}(\mathbf{x})\right)\right]$$

- However, if $(\mathbf{x}, y) \in \mathcal{D}_{\text{train}}$, $\widehat{\mathrm{GE}}(\hat{f}, L)$ will be biased via overfitting the training data.
- Thus, we estimate the GE using unseen data $(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}$:

$$\widehat{\mathrm{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{test}}} \left[L\left(y, \hat{f}(\mathbf{x})\right)\right]$$

METRICS



But what is $L\left(y, \hat{f}(\mathbf{x})\right)$?

$L\left(y, \hat{f}(\mathbf{x})\right)$ will always indicate how good the target matches our prediction. While we can always use the (inner) loss function that we trained the model on as outer loss, this may not always be ideal:

- Explicit values of loss functions may not have a **meaningful interpretation** beyond ordinality.
- The loss function may not be applicable to all models that we are interested in comparing (**model agnostic**ism), e.g. when comparing generative and discriminative approaches.

Thus, there also exist evaluation metrics that are not based on inner losses. Yet, they can (still) be faced with these problems:

- They might be not **useful** (for a specific use case, e.g. when we have imbalanced data).
- They might be im**proper**, i.e. they might draw false conclusions.

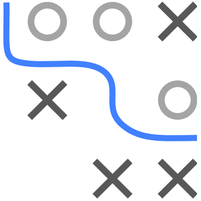
DEEP DIVE: PROPERNESS

- A scoring rule **S** is proper relative to $\mathcal{F}$ if (where a low value of the scoring rule is better):

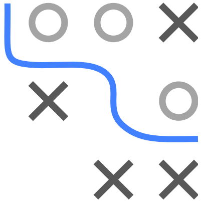
$$\mathbf{S}(Q, Q) \leq \mathbf{S}(F, Q) \forall F, Q \in \mathcal{F}$$

with $\mathcal{F}$ being a convex class of probability measures.

- This means that a scoring rule should be optimal for the actual data target distribution, i.e. we are rewarded for properly modeling the target.



METRICS FOR CLASSIFICATION



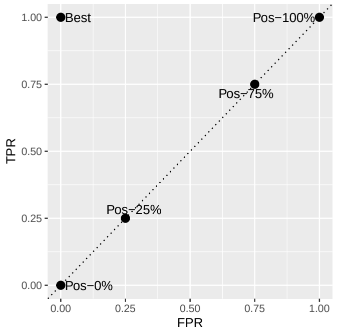
Commonly used evaluation metrics include:

- Accuracy:
  $\rho_{ACC} = \frac{1}{m}\sum_{i=1}^{m}[y^{(i)} = \hat{y}^{(i)}] \in [0, 1].$
- Misclassification error (MCE):
  $\rho_{MCE} = \frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \neq \hat{y}^{(i)}] \in [0, 1].$
- Brier Score:
  $\rho_{BS} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{\pi}^{(i)} - y^{(i)}\right)^2$
- Log-loss:
  $\rho_{LL} = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\left(\hat{\pi}^{(i)}\right) - \left(1 - y^{(i)}\right)\log\left(1 - \hat{\pi}^{(i)}\right)\right).$

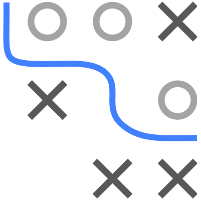
The probabalistic metrics, Brier Score and Log-Loss penalize false confidence, i.e. predicting the wrong label with high probability, heavily.

RECEIVER OPERATING CHARACTERISTICS

- Receiver operating characteristics (ROC) performs evaluation for binary classifiers beyond single metrics.
- We can assess classifiers by their TPR (y-axis) and FPR (x-axis).
- We aim to identify good classifiers who (weakly) dominate others.
- For example, the "Best" classifier in the image strictly dominates "Pos-25%" and "Pos-75%" and weakly dominates the others.



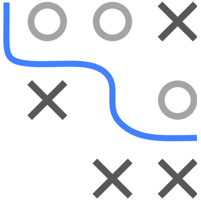
METRICS FOR REGRESSION



Commonly used evaluation metrics include:

- Sum of Squared Errors (SSE): $\rho_{SSE}(\mathbf{y}, \boldsymbol{F}) = \sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$
- Mean Squared Error (MSE): $\rho_{MSE}(\mathbf{y}, \boldsymbol{F}) = \frac{1}{m}\sum_{i=1}^{m} SSE$
- Root Mean Squared Error (RMSE): $\rho_{RMSE}(\mathbf{y}, \boldsymbol{F}) = \sqrt{MSE}$
- R-Squared: $\rho_{R^2}(\mathbf{y}, \boldsymbol{F}) = 1 - \frac{\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{m}(y^{(i)} - \bar{y})^2}$
- Mean Absolute Error (MAE):
  $\rho_{MAE}(\mathbf{y}, \boldsymbol{F}) = \frac{1}{m}\sum_{i=1}^{m}|y^{(i)} - \hat{y}^{(i)}| \in [0; \infty)$

ESTIMATING THE GENERALIZATION ERROR (BETTER)



While

$$\widehat{\mathrm{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{test}}} \left[L\left(y, \hat{f}(\mathbf{x})\right)\right]$$

will be unbiased, with a small m it will suffer from high variance. We have two options to decrease the variance:

- Increase m.
- Compute $\widehat{\mathrm{GE}}(\hat{f}, L)$ for multiple test sets and aggregate them.

With a finite amount of data, increasing m would mean to decrease the size of the training data. Thus, we focus on using multiple (B) test sets:

$$\mathcal{J} = \left((J_{\mathrm{train},1}, J_{\mathrm{test},1}), \dots, (J_{\mathrm{train},B}, J_{\mathrm{test},B})\right).$$

where we compute $\widehat{\mathrm{GE}}(\hat{f}, L)$ for each set and aggregate the estimates. These B sets are generated through **resampling**.

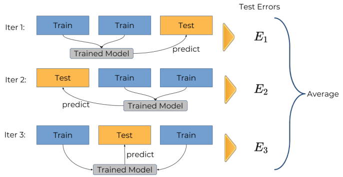
RESAMPLING

There exist a few well established resampling strategies:

- (Repeated) Hold-out / Subsampling
- Cross validation
- Bootstrap

All methods aim to generate $\mathcal{J}$ by splitting the full data set (repeatedly) into a train and test set. The model is trained on the respective train set and evaluated on the test set.

Example: 3-fold cross validation



In order to robustify performance estimates we can repeat these resamplings, e.g. we could perform 10 times 8 fold cross validation.