I2ML :: BASICS

Data

$\mathcal{X} \subseteq \mathbb{R}^p$: p-dimensional **feature space** / input space
Usually we assume categorical features to be numerically encoded.

$\mathcal{Y}$: **target space**
e.g.: $\mathcal{Y} = \mathbb{R}$ for regression, $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$ for binary classification, $\mathcal{Y} = \{1, \dots, g\}$ for multi-class classification with g classes

$\mathbf{x} = (x_1, \dots, x_p)^\top \in \mathcal{X}$: **feature vector** / covariate vector

$y \in \mathcal{Y}$: **target variable** / output variable
Concrete samples are called labels

$\left(\mathbf{x}^{(i)}, y^{(i)}\right) \in \mathcal{X} \times \mathcal{Y}$: i -th **observation** / sample / instance / example

$\mathbb{D} = \bigcup_{n \in \mathbb{N}} (\mathcal{X} \times \mathcal{Y})^n$: **set of all finite data sets**

$\mathbb{D}_n = (\mathcal{X} \times \mathcal{Y})^n \subseteq \mathbb{D}$: **set of all finite data sets of size** n

$\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right) \in \mathbb{D}_n$: **data set** of size n. An n-tuple, a family indexed by $\{1, \dots, n\}$. We use $\mathcal{D}_n$ to emphasize its size.

$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$: **data sets for training and testing**
Often: $\mathcal{D} = \mathcal{D}_{\text{train}} \,\dot\cup\, \mathcal{D}_{\text{test}}$

$\mathbb{P}_{xy}$: **joint probability distribution on** $\mathcal{X} \times \mathcal{Y}$

Classification

$o_k(y) = \mathbb{I}(y = k) \in \{0, 1\}$: multiclass one-hot encoding, if y is class k
We write $\mathbf{o}(y)$ for the g-length encoding vector and $o_k^{(i)} = o_k(y^{(i)})$

$\pi_k = \mathbb{P}(y = k)$: **prior probability** for class k
In case of binary labels we might abbreviate: $\pi = \mathbb{P}(y = 1)$.

Model and Learner

Model / Hypothesis: $f : \mathcal{X} \to \mathbb{R}^g$ maps features to predictions, often parametrized by $\boldsymbol{\theta} \in \Theta$ (then we write $f_{\boldsymbol{\theta}}(\mathbf{x})$ or $f(\mathbf{x}|\boldsymbol{\theta})$).

$\Theta \subseteq \mathbb{R}^d$: **parameter space**

$\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_d) \in \Theta$: model **parameter** vector
Some models may traditionally use different symbols.

$\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R}^g \mid f \text{ belongs to a certain functional family}\}$:
Hypothesis space – set of functions to which we restrict learning

Learner / Inducer $\mathcal{I} : \mathbb{D} \times \Lambda \to \mathcal{H}$ takes a training set $\mathcal{D}_{\text{train}} \in \mathbb{D}$, produces model $f : \mathcal{X} \to \mathbb{R}^g$, with hyperparam. configuration $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$.
We also write $\mathcal{I} : \mathbb{D} \times \Lambda \to \Theta$ or $\mathcal{I}_{\boldsymbol{\lambda}} : \mathbb{D} \to \Theta$

$\boldsymbol{\Lambda} = \boldsymbol{\Lambda_1} \times \boldsymbol{\Lambda_2} \times \dots \times \boldsymbol{\Lambda}_\ell \subseteq \mathbb{R}^\ell$: **hyperparameter space**
$\boldsymbol{\Lambda}_j$ are usually bounded real or integer intervals or a finite categorical set

$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_\ell) \in \boldsymbol{\Lambda}$: **hyperparameter configuration**

$r = y - f(\mathbf{x})$ or $r^{(i)} = y^{(i)} - f\left(\mathbf{x}^{(i)}\right)$: (i-th) **residual** in regression

Classification

$\pi_k(\mathbf{x}) : \mathcal{X} \to [0, 1]$ **probability prediction** for class k, approximates $\mathbb{P}(y = k \mid \mathbf{x})$; for binary we abbreviate with $\pi(\mathbf{x})$ for $\mathbb{P}(y = 1 \mid \mathbf{x})$.

$f_k(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$: **scoring** / discriminant **function** for class k;
for binary we use $f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x})$

$h(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$: **hard label function**;
Typically created by $h(\mathbf{x}) = \underset{k \in \{1,\dots,g\}}{\arg\max} f_k(\mathbf{x})$ or
$h(\mathbf{x}) = \underset{k \in \{1,\dots,g\}}{\arg\max} \pi_k(\mathbf{x})$

$yf(\mathbf{x})$ or $y^{(i)} f\left(\mathbf{x}^{(i)}\right)$: **margin** for ($i$-th) observation in binary classification

$c \in \mathbb{R}$, s.t. $h(\mathbf{x}) := [\pi(\mathbf{x}) \geq c]$ or $h(\mathbf{x}) := [f(\mathbf{x}) \geq c]$: **threshold** for hard label assignment in binary case (common: $c = 0$ for scoring, $c = 0.5$ for probabilistic classifiers)

$\hat{y}$, $\hat{f}$, $\hat{h}$, $\hat{\pi}_k(\mathbf{x})$, $\hat{\pi}(\mathbf{x})$ and $\hat{\theta}$
The hat symbol denotes **learned** functions and parameters.

Loss, Risk and ERM

$L : \mathcal{Y} \times \mathbb{R}^g \to \mathbb{R}_0^+$: **loss function**: Quantifies "quality" $L(y, f(\mathbf{x}))$ of prediction $f(\mathbf{x})$ (or $L(y, \pi(\mathbf{x}))$ of prediction $\pi(\mathbf{x})$) for true y.

(Theoretical) risk: $\mathcal{R} : \mathcal{H} \to \mathbb{R}$, $\mathcal{R}(f) = \mathbb{E}_{((\mathbf{x},y) \sim \mathbb{P}_{xy})}[L(y, f(\mathbf{x}))]$

Empirical risk: $\mathcal{R}_{\text{emp}} : \mathcal{H} \to \mathbb{R}$, $\mathcal{R}_{\text{emp}}(f) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$,

analogously: $\mathcal{R}_{\text{emp}} : \Theta \to \mathbb{R}$; $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)$

Empirical risk minimization (ERM): $\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \Theta}{\arg\min} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$

Bayes-optimal model: $f^* = \arg\min_{f : \mathcal{X} \to \mathbb{R}^g} \mathcal{R}(f)$

Regularized risk: $\mathcal{R}_{\text{reg}} : \mathcal{H} \to \mathbb{R}, \mathcal{R}_{\text{reg}}(f) = \mathcal{R}_{\text{emp}}(f) + \lambda \cdot J(f)$ with

Regression Losses

L2 loss / squared error:

- $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ or $L(y, f(\mathbf{x})) = 0.5(y - f(\mathbf{x}))^2$
- Convex and differentiable, non-robust against outliers
- Optimal constant model: $\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} y^{(i)} = \bar{y}$
- Optimal model over $\mathbb{P}_{xy}$ for unrestricted $\mathcal{H}$: $\hat{f}(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$

L1 loss / absolute error:

- $L(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$
- Convex and more robust, non-differentiable
- Optimal constant model: $\hat{f}(\mathbf{x}) = \text{med}(y^{(1)}, \dots, y^{(n)})$
- Optimal model over $\mathbb{P}_{xy}$ for unrestricted $\mathcal{H}$: $\hat{f}(\mathbf{x}) = \text{med}[y|\mathbf{x}]$

Classification Losses

0-1-loss (binary case)
$L(y, h(\mathbf{x})) = \mathbb{I}(y \neq h(\mathbf{x}))$
$L(y, f(\mathbf{x})) = \mathbb{I}(yf(\mathbf{x}) < 0)$ for $\mathcal{Y} = \{-1, +1\}$
Discontinuous, usually results in NP-hard optimization

Brier score (binary case)
$L(y, \pi(\mathbf{x})) = (\pi(\mathbf{x}) - y)^2$ for $\mathcal{Y} = \{0, 1\}$
Least-squares on probabilities

Log-loss / Bernoulli loss / binomial loss (binary case)
$L(y, \pi(\mathbf{x})) = -y \log(\pi(\mathbf{x})) - (1 - y) \log(1 - \pi(\mathbf{x}))$ for $\mathcal{Y} = \{0, 1\}$
$L(y, \pi(\mathbf{x})) = \log(1 + (\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})})^{-y})$ for $\mathcal{Y} = \{-1, +1\}$

Assuming a logit-link $\pi(\mathbf{x}) = \exp(f(\mathbf{x}))/(1 + \exp(f(\mathbf{x})))$:
$L(y, f(\mathbf{x})) = -y \cdot f(\mathbf{x}) + \log(1 + \exp(f(\mathbf{x})))$ for $\mathcal{Y} = \{0, 1\}$
$L(y, f(\mathbf{x})) = \log(1 + \exp(-y \cdot f(\mathbf{x})))$ for $\mathcal{Y} = \{-1, +1\}$
Penalizes confidently-wrong predictions heavily

Brier score (multi-class case)
$L(y, \pi(\mathbf{x})) = \sum_{k=1}^{g} (\pi_k(\mathbf{x}) - o_k(y))^2$

Log-loss (multi-class case)
$L(y, \pi(\mathbf{x})) = -\sum_{k=1}^{g} o_k(y) \log(\pi_k(\mathbf{x}))$

Optimal constant models
0-1-loss: $h(\mathbf{x}) \in \underset{j \in 0,1}{\arg\max} \sum_{i=1}^{n} \mathbb{I}(y^{(i)} = j)$
Brier and log-loss (binary): $\hat{\pi}(\mathbf{x}) = \bar{y}$
Brier and log-loss (multiclass): $\hat{\pi}(\mathbf{x}) = \left(\frac{1}{n} \sum_{i=1}^{n} o_1^{(i)}, \dots, \frac{1}{n} \sum_{i=1}^{n} o_g^{(i)}\right)$