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.

Basic Notations

Important **notations** used in the whole course

$\mathcal{X}$: p-dim. **input space**

Usually we assume $\mathcal{X} = \mathbb{R}^p$, but categorical **features** can occur as well.

$\mathcal{Y}$: **target space**

For example, $\mathcal{Y} = \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$, $\mathcal{Y} = \{-1, 1\}$, $\mathcal{Y} = \{1, \dots, g\}$ or $\mathcal{Y} = \{\mathrm{label}_1, \dots, \mathrm{label}_g\}$.

$\mathbf{x}$: **feature vector**

$\mathbf{x} = (x_1, \dots, x_p)^T \in \mathcal{X}$.

y : **target / label / output**

$y \in \mathcal{Y}$.

$\mathbb{P}_{xy}$: **probability distribution**

Joint probability distribution on $\mathcal{X} \times \mathcal{Y}$.

$p(\mathbf{x}, y)$ or $p(\mathbf{x}, y \mid \boldsymbol{\theta})$: **joint pdf**

Joint probability density function for $\mathbf{x}$ and y.

Note: This lecture is mainly developed from a frequentist perspective. If parameters appear behind the $|$, this is for better reading, and does not imply that we condition on them in a Bayesian sense (but this notation would actually make a Bayesian treatment simple). So formally, $p(\mathbf{x}|\theta)$ should usually be understood to mean $p_\theta(\mathbf{x})$ or $p(\mathbf{x}, \theta)$ or $p(\mathbf{x}; \theta)$.

Definitions

$(\mathbf{x}^{(i)}, y^{(i)})$: i -th **observation** or **instance**

$\mathcal{D} = \left\{ \left(\mathbf{x}^{(1)}, y^{(1)}\right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)}\right) \right\}$

data set with n observations.

$\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{test}}$: data for training and testing

Often, $\mathcal{D} = \mathcal{D}_{\text{train}} \dot{\cup}\, \mathcal{D}_{\text{test}}$.

$f(\mathbf{x})$ or $f(\mathbf{x} \mid \boldsymbol{\theta}) \in \mathbb{R}$ or $\mathbb{R}^g$: prediction function (**model**)

We might suppress $\boldsymbol{\theta}$ in notation.

$h(\mathbf{x})$ or $h(\mathbf{x}|\boldsymbol{\theta}) \in \mathcal{Y}$

Discrete prediction for classification.

$\boldsymbol{\theta} \in \Theta$: model **parameters**

Some models may traditionally use different symbols.

$\mathcal{H}$: **hypothesis space**

f lives here, restricts the functional form of f.

$\epsilon = y - f(\mathbf{x})$ or $\epsilon^{(i)} = y^{(i)} - f\left(\mathbf{x}^{(i)}\right)$

Residual in regression.

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

With, $\mathcal{Y} = \{-1, 1\}$.

$\pi_k(\mathbf{x}) = \mathbb{P}(y = k \mid \mathbf{x})$: **posterior probability** for class k, given x

In case of binary labels we might abbreviate $\pi(\mathbf{x}) = \mathbb{P}(y = 1 \mid \mathbf{x})$.

$\pi_k = \mathbb{P}(y = k)$: **prior probability** for class k

In case of binary labels we might abbreviate $\pi = \mathbb{P}(y = 1)$.

$\mathcal{L}(\boldsymbol{\theta})$ and $\ell(\boldsymbol{\theta})$: Likelihood and log-Likelihood for a parameter $\boldsymbol{\theta}$

These are based on a statistical model.

$\hat{y}$, $\hat{f}$, $\hat{h}$, $\hat{\pi}_k(\mathbf{x})$, $\hat{\pi}(\mathbf{x})$ and $\hat{\boldsymbol{\theta}}$

These are learned functions and parameters (These are estimators of corresponding functions and parameters).

Note: With a slight abuse of notation we write random variables, e.g., $\mathbf{x}$ and y, in lowercase, as normal variables or function arguments. The context will make clear what is meant.

Important terms

Model: $f : \mathcal{X} \to \mathbb{R}^g$ is a function that maps feature vectors to predictions.

Learner: takes a data set with features and outputs (**training set**, $\in \mathcal{X} \times \mathcal{Y}$) and produces a **model** (which is a function $f : \mathcal{X} \to \mathbb{R}^g$)

Learning = Representation + Evaluation + Optimization.

Representation: (Hypothesis space) Defines which kind of model structure of f can be learned from the data.

Example: Linear functions, Decision trees etc.

Evaluation: A metric that quantifies how well a specific model performs on a given data set. Allows us to rank candidate models in order to choose the best one.

Example: Squared error, Likelihood etc.

Optimization: Efficiently searches the hypothesis space for good models.

Example: Gradient descent, Quadratic programming etc.

Loss function: The "goodness" of a prediction $f(\mathbf{x})$ is measured by a loss function $L\,(y, f(\mathbf{x}))$

Through **loss**, we calculate the prediction error and the choice of the loss has a major influence on the final model

Risk Minimization: The ability of a model f to reproduce the association between x and y that is present in the data $\mathcal{D}$ can be measured by the average loss: the **empirical risk**.

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right).$$

Learning then amounts to **empirical risk minimization** – figuring out which model f has the smallest average loss:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \mathcal{R}_{\text{emp}}(f).$$