



Introduction to Machine Learning

Week 3

Olivier JAYLET

School of Information Technology and Engineering

Definition

The goal of the supervised learning approach is to learn a **mapping** from inputs \mathbf{x} to outputs y , given a *labeled set* of input-output pairs:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n,$$

where:

- \mathcal{D} is the **training set**.
- n is the number of training examples.
- \mathbf{x}_i : Each training example is a vector of numbers called *features, attributes, covariates, or explanatory variables*:
 - They are usually stored on an $n \times p$ **design matrix**.
 - Their structure may be more complex, such as an image, a text, a sequence, a graph, ...
- y_i is the **response variable**:
 - It can be a *categorical* or *nominal* variable from a finite set.
 - Or a *real-valued scalar*.

As we know the real value of y_i , it is possible to compare the prediction with the observable and therefore compute **error metrics**.

Classification

When the response variable y_i is categorical, the problem is known as **classification** (or pattern recognition).

- detecting if an e-mail is ham or spam
- recognizing parts of speech (verbs, subject, pronouns, etc.)
- face detection on an image
- market segmentation

Regression

When the response variable y_i is a real-valued scalar, the problem is known as **regression**.

- predict the wage of an individual
- predict the value of a financial asset
- predict the temperature at any location in a building

Supervised learning

In supervised learning problems, we assume that there exists a relationship between the input variables x and the output variable y :

$$y = f(x) + \epsilon,$$

where f is a fixed but unknown function of the predictors, and ϵ is a random error term.

Bias

Definition

Systematic error in model predictions.

For a model prediction $\hat{f}(x)$ and true function $f(x)$:

$$\text{Bias}(x) = \mathbb{E}[\hat{f}(x)] - f(x)$$

High bias \Rightarrow underfitting.

Variance

Definition

Measures sensitivity of model predictions to training data variations.

$$\text{Var}(\hat{f}(x)) = \mathbb{E}\left[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2\right]$$

High variance \Rightarrow overfitting.

Misspecification Bias

- Let's consider a quite general model: $y = f(X) + \varepsilon$.
- Assume that X and y are test observations drawn from the population.
- The expected (squared) prediction error, or EPE, is equal to :

$$\begin{aligned}E(y - \hat{y})^2 &= E[f(X) + \varepsilon - \hat{f}(X)]^2 \\&= \underbrace{E[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\varepsilon)}_{\text{Irreducible}}\end{aligned}$$

- The focus of Machine Learning is to estimate f with the aim of minimizing the reducible error.
- Reducible error = $[\text{Bias}(\hat{f}(X))]^2 + \text{Var}(\hat{f}(X))$.

Misspecification Bias

Assuming that the data are generated by a specific model, or that the model is correctly specified (i.e. $f(x) = \hat{f}(x)$),

remains to assume that the (misspecification) bias is zero:

$$Bias(\hat{f}) = 0$$

$$E[f(X) - \hat{f}(X)]^2 = 0$$

Estimating f

We are interested in estimating the function f , for two main reasons:

- to **predict** the value of y for some inputs that may nor be available.
- to estimate the **impact** of X on y i.e., for *inference* purposes.

Estimating f for prediction

If we are interested in estimating f for prediction purposes:

- we want to get $\hat{y} = \hat{f}(x)$ where \hat{f} is the estimation of f
- we may not be interested that much in the exact form of \hat{f} and may view it as a black box... as long as it gets accurate predictions

Estimating f for inference

When we are interested in estimating the mapping from x to y for inference purposes, we want to know **how variations in the inputs x affect the output y .**

In that case, we may want to know what are the important predictors among x that can explain the variations of the response.

Besides, we may want to know more about the relationship between predictors and the response:

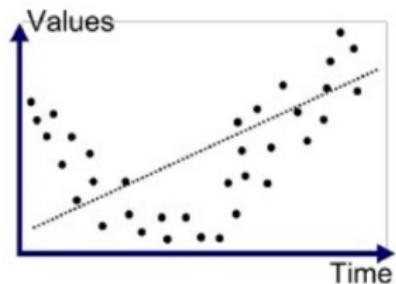
- what is the magnitude?
- what is the sign of the relationship?
- is it linear? non-linear?

Accuracy Vs. Interpretability Trade-off

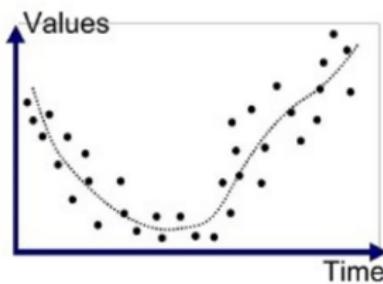
There is therefore a trade-off between prediction accuracy and model interpretability.

Depending on the goal of the estimation, one might prefer giving-up some accuracy and turn to more restrictive model to get more interpretable results.

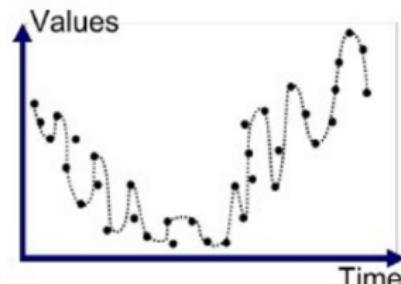
Model Selection



Underfitted

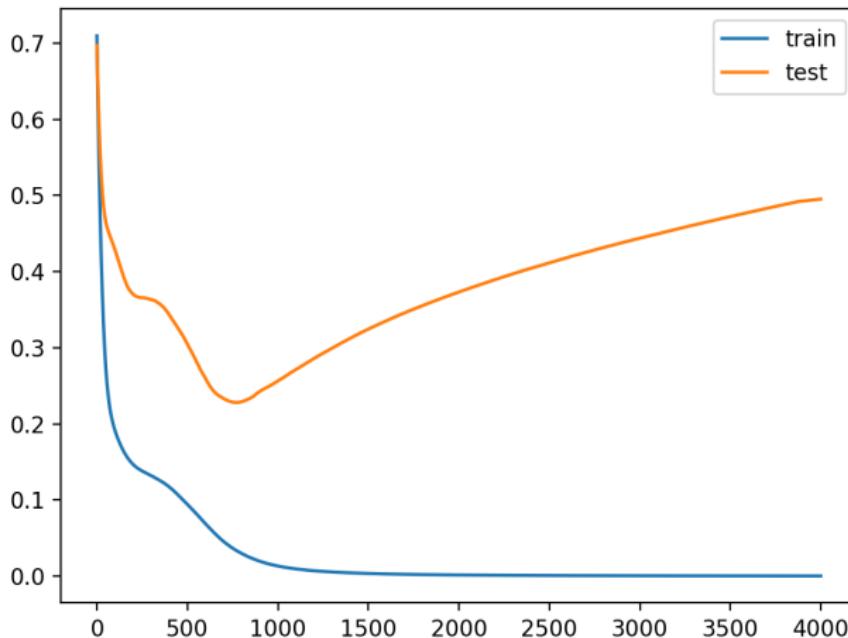


Good Fit/R robust



Overfitted

Train and test Loss



Definition

A **Cost function**¹ measures the performance of a machine learning model for given data.

It quantifies the error between predicted and expected values and present that error in the form of a single real number.

It is often denoted as :

$$\mathcal{L}(\mathcal{D}),$$

where : $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$

¹Also called **Loss function**

Regression Loss functions

$$\text{Mean Squared Error (MSE)} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

$$\text{Mean Absolute Error (MAE)} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Binary Classification Loss functions

Misclassification Rate :

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(y_i \neq \hat{y}_i) \quad (3)$$

Binary Cross Entropy Loss :

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (4)$$