Introduction

Loris Jautakas

September 10, 2023

1 Informal Definitions

Definition 1: /machine_learning

A computer program learns from an experience E with respect to a class of tasks T and performance measure P if its performance at tasks in T measured by P improves from using E.

Definition 2: [machine_learning]/ loss_function

Consider:

1. Feature space X
2. Label space Y

Then L is some loss function $L : Y \times Y \to \mathbb{R}$, where: $L(f(x), y)$ is the loss predicting $f(x)$ when the true label is y.

Definition 3: [machine_learning, loss_function]/ learning_problem

A learning problem is a tuple

$$\boxed{(X, Y, L, F)} \tag{1}$$

1. X is the input space, also called the feature space.
2. Y is the output space, also called the label space.
3. L is some loss function.
4. F is a set of functions $f : X \to Y$.

The goal is to get as close as in F to the function:

$$\boxed{f^* \stackrel{\text{def}}{=} \arg\min_{y \in Y} L(f(x), y)} \tag{2}$$

Definition 4: [learning_problem]/ supervised_learning_problem

Consider

1. Learning problem (X, Y, L, F)
2. Training dataset $T \subseteq (X \times Y)^{<\omega}$
3. D as some probability measure over $X \times Y$.

A supervised learning problem is a learning problem

$$\boxed{(X, Y, L, F, T, D)} \tag{3}$$

Definition 5: [learning_problem]/ training_error

Consider

- (X, Y, L, F) a supervised learning problem
- $f \in F$ a function

Define the loss of a function f:

- $$\boxed{L(y_i, f(x_i)) \stackrel{\text{def}}{=} \mathbb{1}_{y_i \neq f(x_i)}} \tag{4}$$
- $$\boxed{\begin{aligned} E_n(f) &= \frac{1}{n} \sum_{i \in n} L(f(x_i), y_i) \qquad (5) \\ &= \frac{1}{n} \sum_{i \in n} L(y_i, f(x_i)) \qquad (6) \end{aligned}}$$

Definition 6: [supervised_learning_problem]/ linear_classifier

Let $(X, \langle \cdot, \cdot \rangle)$ be some inner product space. Consider any learning problem $(X, \{0, 1\}, F, T, D)$. A linear classifier learns:

1. $\vec{w} \in X$
2. $r \in \mathbb{R}$

such that:

$$\boxed{f(x) \stackrel{\text{def}}{=} \begin{cases} 1 & \langle \vec{w}, \vec{x} \rangle > r \\ -1 & \text{o.w.} \end{cases}} \tag{7}$$

$$\tag{8}$$

is a good predictor for D.

Example 1.1: /linear_classifier_visualization

Being parallel to the hyperplane is the same as being orthogonal to the normal vector $\vec{w}$. Thus if the inner product between $\vec{x}$ and $\vec{w}$ is positive, then $\vec{x}$ is on the same side of the hyperplane as $\vec{w}$ and if it is negative, then it is on the other side. The separating plane can be thought of $\{\vec{x} \in X | \langle \vec{w} | \vec{x} \rangle = b\}$

Make this in inkscape, 3D makes more sense

Every plane in $\mathbb{R}^n$ can be characterized by its normal vector w, plus some offset $r \in \mathbb{R}$ (i.e. how much you go from the origin along the normal vector before you hit the plane). We then classify by whether a vector in X is along the same direction as the normal vector with offset r, i.e. $\langle w, x \rangle > r$.