Binary Linear Classification

Example : Credit Approval Application

User attribute vector

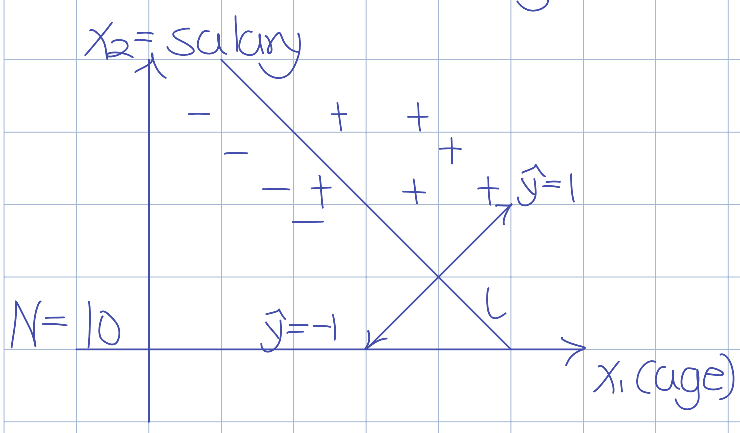
$\underline{x} = (\text{age}, \text{gender}, \text{salary}, \ldots) \in \mathbb{R}^d$

Output $y = \begin{cases} +1, & \text{if approved} \\ -1, & \text{if declined} \end{cases}$

Historical Data

$\underset{\text{Dataset}}{D} = \{(\underline{x_1}, y_1), (\underline{x_2}, y_2), \ldots, (\underline{x_N}, y_N)\}$

$d=2$ $\quad \underline{x} = (x_1 = \text{age}, x_2 = \text{salary})$



$l: \omega_1 x_1 + \omega_2 x_2 + b = 0$

$E_{in}(Q) = \frac{1}{10}$

Propose Model, Linear Classification

Weight Vector : $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$

Constant : $b \in \mathbb{R}$

Given Input : $\underline{x} = (x_1, x_2, x_3, \ldots, x_d) \in \mathbb{R}^d$

Compute : $\sum_{i=1}^{d} w_i x_i > b \longrightarrow \hat{y} = +1$

$$\sum_{i=1}^{d} w_i x_i < b \longrightarrow \hat{y} = -1$$

Model Parameters

$$\Omega = \{ \underline{w}, b \}$$

Training

Loss Function ($E_{in}(\Omega)$)

$$E_{in}(\Omega) = \text{Average number of miss classified points}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \underbrace{1}_{\text{indicator function}} \{ \underbrace{y_i}_{\text{true label}} \neq \hat{y}_i \} \longleftarrow \text{predicated label}$$

$$1\{t\} = \begin{cases} 1, & t \text{ is true} \\ 0, & t \text{ is false} \end{cases}$$

$$\Omega^* = \arg\min_{\Omega} E_{in}(\Omega)$$

We will approximately minimize $E_{in}(\Omega)$ using the perceptron learning algorithm

Slight change in notation

Expanded Dimension

$$\underline{x_i} \in \mathbb{R}^d \longrightarrow \underline{x_i} \in \mathbb{R}^{d+1}$$

Originally

$$\underline{x} = (x_1, x_2, x_3, \ldots, x_d) \in \mathbb{R}^d$$

New

$$\underline{x} = (x_0 = 1, x_1, x_2, x_3, \ldots, x_d) \in \mathbb{R}^{d+1}$$

$$\underline{w} = (\underbrace{-b}_{=w_0}, w_1, w_2, w_3, \ldots, w_d) \in \mathbb{R}^{d+1}$$

Decision Rule

Compute

$$\underline{w^T x} = \sum_{i=0}^{d} w_i x_i > 0 \longrightarrow \hat{y} = +1$$

$$\underline{w^T x} < 0 \longrightarrow \hat{y} = -1$$

$$\sum_{i=0}^{d} w_i x_i = w_0 x_0 + \sum_{i=1}^{d} w_i x_i \underset{\hat{y}=-1}{\overset{\hat{y}=1}{\gtrless}} 0$$

$$\updownarrow$$

$$\sum_{i=1}^{d} w_i x_i \underset{\hat{y}=-1}{\overset{\hat{y}=1}{\gtrless}} b$$

Decision Rule

$\hat{y} = h_{\underline{w}}(\underline{x})$ $\underline{w}$ (weights), $\underline{x}$ (input)

$= \text{sign}(\underline{w^T x})$

$$\text{sign}(t) = \begin{cases} +1, & t > 0 \\ -1, & t < 0 \end{cases}$$

Binary Linear Classification Problems $\underline{w} \in \mathbb{R}^{d+1}$

Decision Rule, $y = h_{\underline{w}}(\underline{x}) \triangleq \text{sign}(\underline{w^T x})$

Loss Function, $E_{in}(\underline{w}) = \frac{1}{N} \sum_{i=1}^{N} 1\{ y_i \neq h_{w}(\underline{x_i}) \}$

Notation

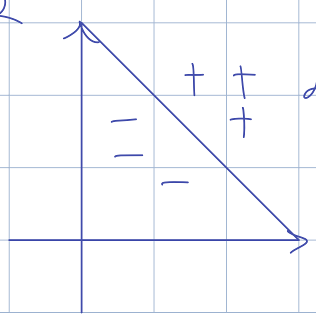
Vector: $\underline{x}$ Matrices: $\underline{X}$

Scalar: x

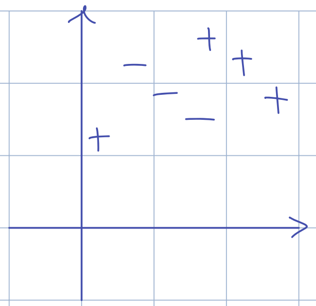
Select $\underline{w}$ that minimizes $E_{in}(\underline{w})$

Good News: When data set D is separable, the perception learning algorithm can efficiently compute the classification rule

$d=2$



Linearly Separable Dataset



Not Linearly Separable

For non separable datasets PLA can be extended to get good heuristics

Perceptron Learning Algorithm (Learning Rate Of 1)

Input, Training Dataset D that is linearly separable

Output, $\underline{w} \in \mathbb{R}^{d+1}$ that achieves

$$E_{in}(\underline{w}) = 0$$

Initialization

Initialize $\underline{w}$ in an arbitrary fashion

$$\underline{w} = (\overset{w_0}{0}, \overset{w_1}{0}, \overset{w_2}{0}, \ldots, \overset{w_d}{0}) \in \mathbb{R}^{d+1}$$

Step 1 : Check if $E_{in}(\underline{w}) = 0$?

If yes, stop output $\underline{w}$

Step2 : Let $(\underline{x_n}, y_n)$ be a misclassified point in D

(If $(\underline{x_n}, y_n)$ is on the boundary, treat it as misclassified)

If $y_n = +1$ then $\underline{w} \longleftarrow \underline{w} + \underline{x_n}$

If $y_n = -1$ then $\underline{w} \longleftarrow \underline{w} - \underline{x_n}$

Go To Step 1