PERCEPTRON

CS6140

Predrag Radivojac
KHOURY COLLEGE OF COMPUTER SCIENCES
NORTHEASTERN UNIVERSITY

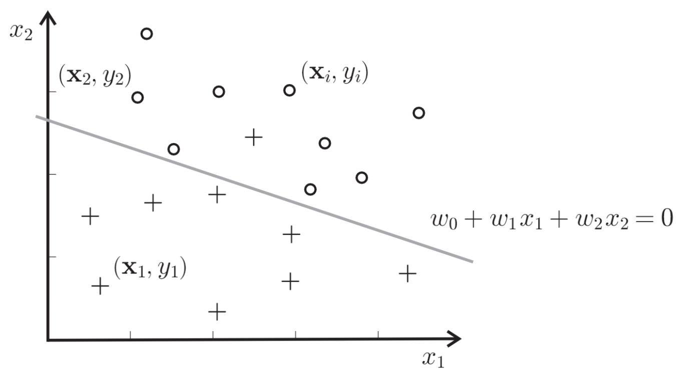
Spring 2021

LINEAR CLASSIFICATION

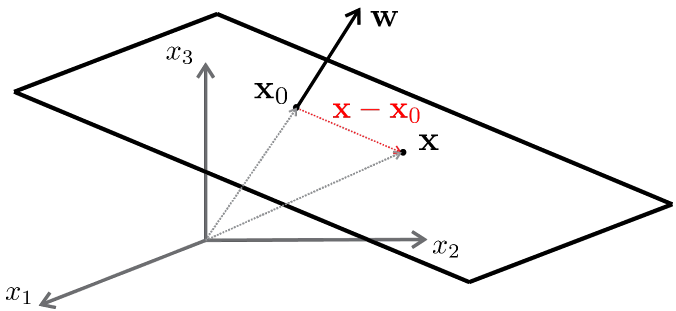
Given: a set of observations $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$

Objective: find best linear separator $f(\boldsymbol{x}) = w_0 + \sum_{j=1}^{d} w_j x_j$

$\mathcal{X} = \mathbb{R} \times \mathbb{R}$, $\mathcal{Y} = \{0, 1\}$



Equation of the Plane



A plane is defined using:

1. a point $\mathbf{x}_0$ lying in the plane
2. a vector $\mathbf{w}$ normal to the plane

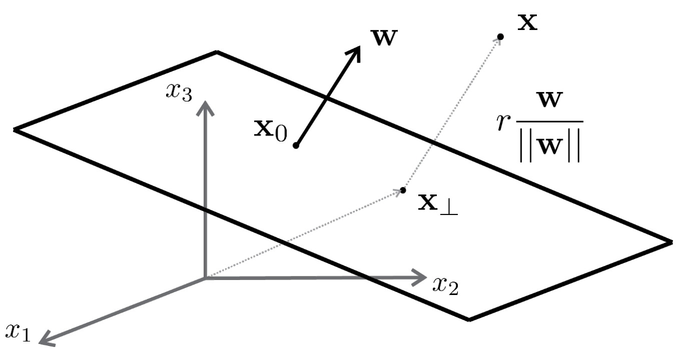
Let $\mathbf{x}$ be on the plane defined by $\mathbf{w}$ and $\mathbf{x}_0$:

$$\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0$$

$$\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{x}_0 = 0$$

$$\boxed{\mathbf{w}^T\mathbf{x} + w_0 = 0}$$

Distance from Point to the Plane



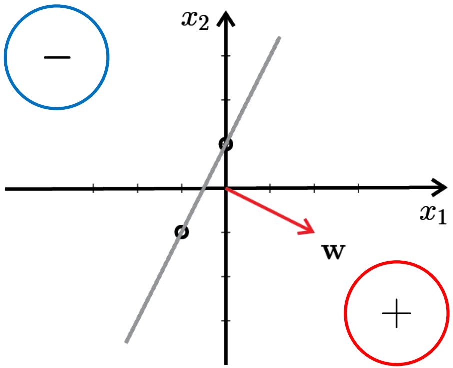
$$\mathbf{x} = \text{outside the plane}$$

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||}$$

$$\mathbf{w}^T\mathbf{x} + w_0 = \underbrace{\mathbf{w}^T\mathbf{x}_\perp + w_0}_{0} + r||\mathbf{w}||$$

$$\boxed{r = \frac{\mathbf{w}^T\mathbf{x} + w_0}{||\mathbf{w}||}}$$

EXAMPLE



$$x_2 = 2x_1 + 1 \quad \text{or} \quad 2x_1 - x_2 + 1 = 0$$

$$\mathbf{x}, \mathbf{w} \in \mathbb{R}^2$$

$$\mathbf{w}^T\mathbf{x} + w_0 = 0$$

where $\mathbf{w} = (2, -1)$ and $w_0 = 1$.

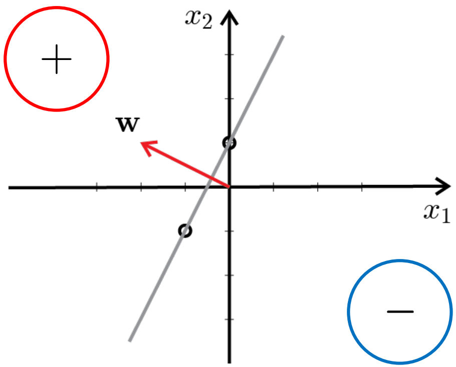
$$r = \frac{\mathbf{w}^T\mathbf{x} + w_0}{||\mathbf{w}||}$$

$$\mathbf{x} = (0, 0) \implies r = \tfrac{1}{\sqrt{5}}$$

$$\mathbf{x} = (-1, 1) \implies r = -\tfrac{2}{\sqrt{5}}$$

Vector $\mathbf{w}$ defines what side of the plane is positive.

EXAMPLE



$$x_2 = 2x_1 + 1$$

What if $\mathbf{w} = (-2, 1)$?

$\mathbf{x}, \mathbf{w} \in \mathbb{R}^2$

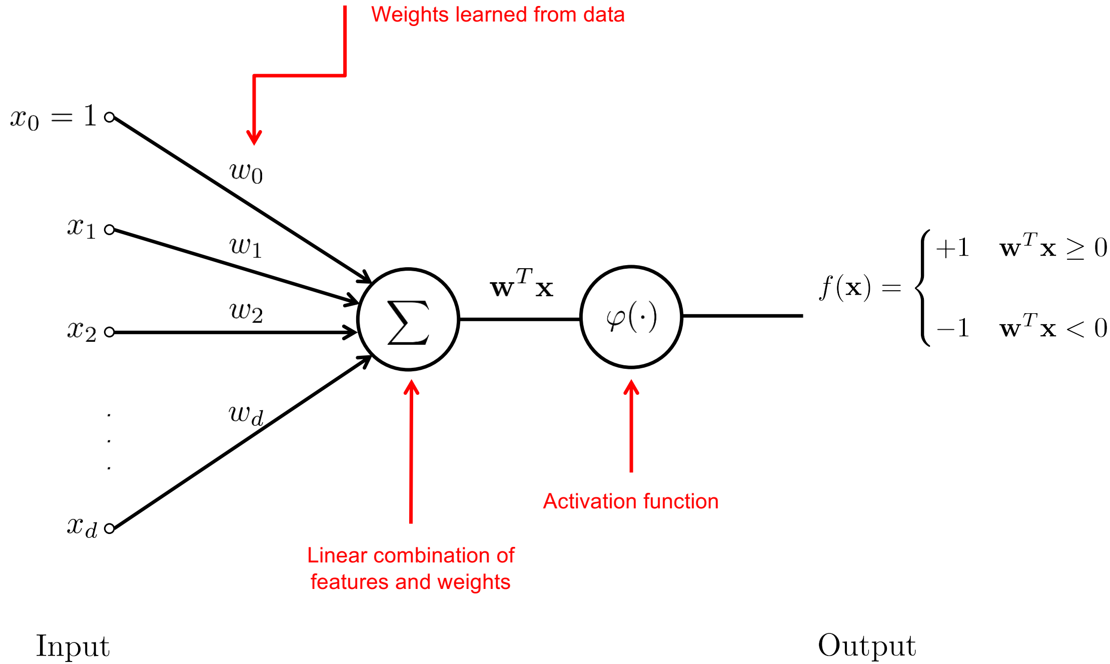
$\mathbf{w}^T\mathbf{x} + w_0 = 0$
where $\mathbf{w} = (-2, 1)$ and $w_0 = -1$.

$$r = \frac{\mathbf{w}^T\mathbf{x} + w_0}{||\mathbf{w}||}$$

$$\mathbf{x} = (0, 0) \implies r = -\tfrac{1}{\sqrt{5}}$$

$$\mathbf{x} = (-1, 1) \implies r = \tfrac{2}{\sqrt{5}}$$

Perceptron


Weights learned from data
$x_0 = 1$
x_1
x_2
x_d
w_0
w_1
w_2
w_d
$\sum$
$\mathbf{w}^T\mathbf{x}$
$\varphi(\cdot)$
$f(\mathbf{x}) = \begin{cases} +1 & \mathbf{w}^T\mathbf{x} \geq 0 \\ -1 & \mathbf{w}^T\mathbf{x} < 0 \end{cases}$
Activation function
Linear combination of
features and weights
Input
Output

Perceptron Training Algorithm

Algorithm 1 Perceptron training algorithm. The algorithm loops over the training data $\mathcal{D}$ until either the weight vector is unchanged for a pre-specified number of steps or the maximum number of steps is exceeded.

Input:

Training data: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathcal{X} = \{1\} \times \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$
Learning parameter: $\eta \in (0, 1]$
Termination criteria; e.g., the maximum number of steps

Initialization:

$\mathbf{w} \leftarrow \mathbf{0}$

Weight learning:

repeat until termination criteria are satisfied
  draw the next labeled example $(\mathbf{x}, y)$ from $\mathcal{D}$
  if $(\mathbf{w}^T\mathbf{x} \geq 0 \;\wedge\; y = -1) \;\vee\; (\mathbf{w}^T\mathbf{x} < 0 \;\wedge\; y = +1)$
    $\mathbf{w} \leftarrow \mathbf{w} + \eta y \mathbf{x}$
  end
end

Output:

Weight vector $\mathbf{w} \in \mathbb{R}^{d+1}$

Pocket algorithm

Perceptron training:

- uses negative reinforcement
- ignores correct predictions

Idea:

- keep the best-so-far $\boldsymbol{w}$ "in the pocket"
- determine best $\boldsymbol{w}$ by the run of correct classifications

Result:

- mimimizes error rate

Algorithm 1 Pocket algorithm. The algorithm loops over the training data $\mathcal{D}$ until either $\mathbf{w}_{\text{pocket}}$ is unchanged for a pre-specified number of steps or the maximum number of steps is exceeded.

Input:

Training data: $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathcal{X} = \{1\} \times \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$
Learning parameter: $\eta \in (0, 1]$
Termination criteria; e.g. the maximum number of steps

Initialization:

$\mathbf{w} \leftarrow \mathbf{w}_{\text{pocket}} \leftarrow \mathbf{0}$
$\text{run} \leftarrow \text{run}_{\text{pocket}} \leftarrow 0$

Weight learning:

```
repeat until termination criteria are satisfied
    draw the next labeled example (x, y) from D
    if (w^T x ≥ 0 ∧ y = −1) ∨ (w^T x < 0 ∧ y = +1)
        if run > run_pocket
            w_pocket ← w
            run_pocket ← run
        end
        w ← w + ηyx
        run ← 0
    else
        run ← run + 1
    end
end
if run > run_pocket
  w_pocket ← w
end
```

Output:

Weight vector $\mathbf{w}_{\text{pocket}} \in \mathbb{R}^{d+1}$
