Machine Learning and Intelligent Systems

Linear Classifiers: The Perceptron

Maria A. Zuluaga

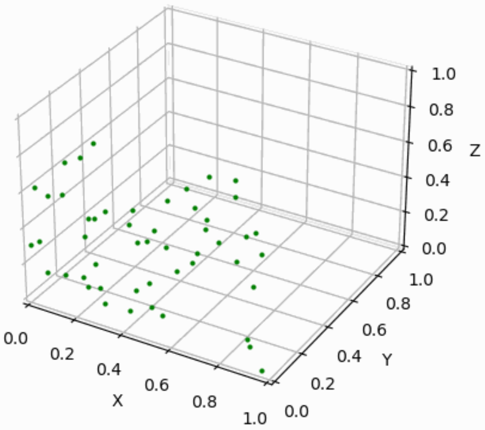
Nov 10, 2023

EURECOM - Data Science Department

Table of contents

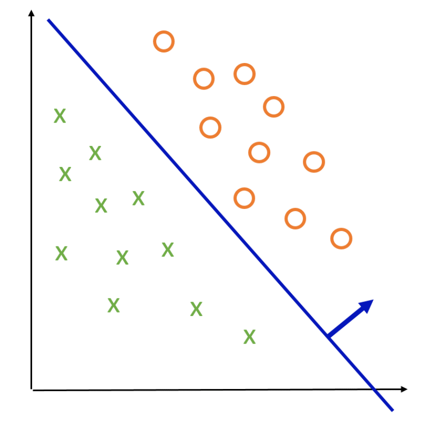
The Perceptron

Motivation: The Curse of Dimensionality



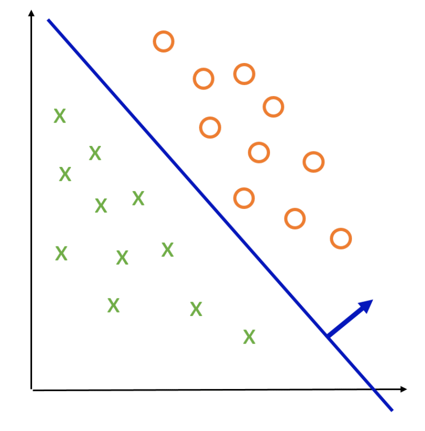
see `03_curse.ipynb`

Intuition



- There exists a hyperplane $\mathcal{H}$ that separates the data

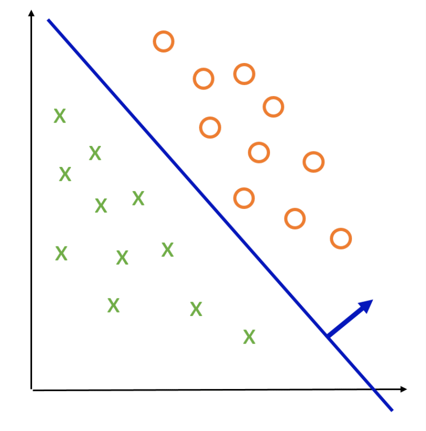
Intuition



- There exists a hyperplane $\mathcal{H}$ that separates the data

$$\mathcal{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + \hat{b} = 0\}$$

Intuition



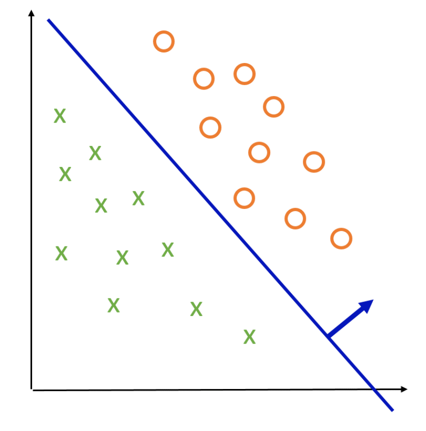
- There exists a hyperplane $\mathcal{H}$ that separates the data

$$\mathcal{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + \hat{b} = 0\}$$

- To classify a new point:

$$\hat{\mathbf{w}}^T\mathbf{x} + \hat{b} > 0 \quad \text{Positive sample}$$

Intuition



- There exists a hyperplane $\mathcal{H}$ that separates the data

$$\mathcal{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + \hat{b} = 0\}$$

- To classify a new point:

$$\hat{\mathbf{w}}^T\mathbf{x} + \hat{b} > 0 \quad \text{Positive sample}$$

$$\hat{\mathbf{w}}^T\mathbf{x} + \hat{b} < 0 \quad \text{Negative sample}$$

Assumptions

Data Assumptions:

- Binary classification : $y_i \in \{-1, 1\}$
- Data is linearly separable

Assumptions

Data Assumptions:

- Binary classification : $y_i \in \{-1, 1\}$
- Data is linearly separable

Model Assumption:

- The decision boundary is a hyperplane:

$$\mathcal{H} = \{\mathbf{x} : \hat{w}^T \mathbf{x} + b = 0\}$$

- $\mathbf{w}$: Weight vector that defines the hyperplane
- b: bias

Formulation

The classifier:

$$y_i = h(\mathbf{x}_i) = \text{sign}(\mathbf{w}^T \mathbf{x}_i + b)$$

Formulation

The classifier:

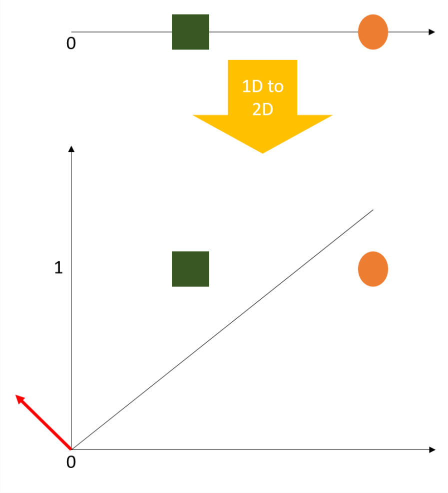
$$y_i = h(\mathbf{x}_i) = \text{sign}(\mathbf{w}^T\mathbf{x}_i + b)$$

- As dealing with b can be complicated, we will absorb it into the weights vector $\mathbf{w}$.
- We use a similar procedure as with w_0 (see first lecture).

$$\mathbf{x}_i \text{ becomes } \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}$$

$$\mathbf{w} \text{ becomes } \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}$$

Geometrical interpretation



Formulation

The new notations leads to the same expression:

$$\begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}^T \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = \mathbf{w}^T \mathbf{x}_i + b$$

Formulation

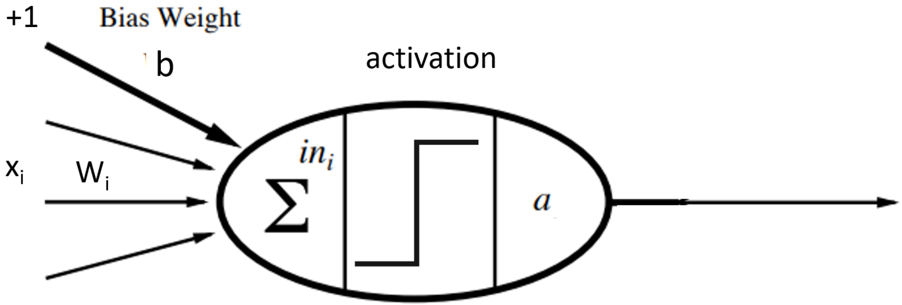
The new notations leads to the same expression:

$$\begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix}^T \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} = \mathbf{w}^T \mathbf{x}_i + b$$

Allowing to simplify the expression for h:

$$y_i = h(\mathbf{x}_i) = \mathrm{sign}(\mathbf{w}^T \mathbf{x}_i)$$

Representation



Error function: The perceptron criterion

Given two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$, with $\mathcal{C}_1$ associated to $y = 1$ and $\mathcal{C}_2$ associated to $y = -1$:

- Patterns $\mathbf{x}_i \in \mathcal{C}_1$ satisfy $\mathbf{w}^T \mathbf{x}_i > 0$
- Patterns $\mathbf{x}_i \in \mathcal{C}_2$ satisfy $\mathbf{w}^T \mathbf{x}_i < 0$

Error function: The perceptron criterion

Given two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$, with $\mathcal{C}_1$ associated to $y = 1$ and $\mathcal{C}_2$ associated to $y = -1$:

- Patterns $\mathbf{x}_i \in \mathcal{C}_1$ satisfy $\mathbf{w}^T\mathbf{x}_i > 0$
- Patterns $\mathbf{x}_i \in \mathcal{C}_2$ satisfy $\mathbf{w}^T\mathbf{x}_i < 0$

In words, the points belonging to the two classes sit in opposite sides of the hyperplane define by the vector $\mathbf{w}$.

Error function: The perceptron criterion

Given two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$, with $\mathcal{C}_1$ associated to $y = 1$ and $\mathcal{C}_2$ associated to $y = -1$:

- Patterns $\mathbf{x}_i \in \mathcal{C}_1$ satisfy $\mathbf{w}^T \mathbf{x}_i > 0$
- Patterns $\mathbf{x}_i \in \mathcal{C}_2$ satisfy $\mathbf{w}^T \mathbf{x}_i < 0$

In words, the points belonging to the two classes sit in opposite sides of the hyperplane define by the vector $\mathbf{w}$.

This means a point correctly classified satisfies:

$$\mathbf{w}^T \mathbf{x}_i y_i > 0$$

Error function: The perceptron criterion

Given two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$, with $\mathcal{C}_1$ associated to $y = 1$ and $\mathcal{C}_2$ associated to $y = -1$:

- Patterns $\mathbf{x}_i \in \mathcal{C}_1$ satisfy $\mathbf{w}^T \mathbf{x}_i > 0$
- Patterns $\mathbf{x}_i \in \mathcal{C}_2$ satisfy $\mathbf{w}^T \mathbf{x}_i < 0$

In words, the points belonging to the two classes sit in opposite sides of the hyperplane define by the vector $\mathbf{w}$.

This means a point correctly classified satisfies:

$$\mathbf{w}^T \mathbf{x}_i y_i > 0$$

Question: How can we quantify errors?

Error function: The perceptron criterion

- A natural choice of error function is the 0/1 loss.
- Problems: 0/1 loss is a piecewise constant function of $\mathbf{w}$, with discontinuities wherever a change in $\mathbf{w}$ causes the decision boundary to move across points.
- Not a good choice when using the gradient of the error function.

Error function: The perceptron criterion

- A natural choice of error function is the 0/1 loss.
- Problems: 0/1 loss is a piecewise constant function of $\mathbf{w}$, with discontinuities wherever a change in $\mathbf{w}$ causes the decision boundary to move across points.
- Not a good choice when using the gradient of the error function.

The perceptron criterion:

$$E_p(\mathbf{w}) = -\sum_{i \in \mathcal{M}} \mathbf{w}^T \mathbf{x}_i y_i$$

Error function: The perceptron criterion

- A natural choice of error function is the 0/1 loss.
- Problems: 0/1 loss is a piecewise constant function of $\mathbf{w}$, with discontinuities wherever a change in $\mathbf{w}$ causes the decision boundary to move across points.
- Not a good choice when using the gradient of the error function.

The perceptron criterion:

$$E_p(\mathbf{w}) = -\sum_{i \in \mathcal{M}} \mathbf{w}^T \mathbf{x}_i y_i$$

The perceptron criterion associates zero error with any pattern that is correctly classified, whereas for a misclassified pattern $\mathbf{x}_i$ it tries to minimize the quantity $\mathbf{w}^T \mathbf{x}_i$, with $\mathcal{M}$ the set of misclassified points.

The Learning Process

- We need to obtain an expression for the gradient of the perceptron criterion

The Learning Process

- We need to obtain an expression for the gradient of the perceptron criterion
- We will use **stochastic gradient descent (SGB)** to minimize the error function
- From last lecture: In SGB, the gradient is approximated by a gradient at a single sample i randomly chosen at each iteration

The Learning Process

- We need to obtain an expression for the gradient of the perceptron criterion
- We will use **stochastic gradient descent (SGB)** to minimize the error function
- From last lecture: In SGB, the gradient is approximated by a gradient at a single sample i randomly chosen at each iteration
- A change in the weight vector $\mathbf{w}$ is given by

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \alpha \nabla E_p(\mathbf{w}) =$$

The Learning Process

- We need to obtain an expression for the gradient of the perceptron criterion
- We will use **stochastic gradient descent (SGB)** to minimize the error function
- From last lecture: In SGB, the gradient is approximated by a gradient at a single sample i randomly chosen at each iteration
- A change in the weight vector $\mathbf{w}$ is given by

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \alpha \nabla E_p(\mathbf{w}) = \mathbf{w}^{(\tau)} + \alpha \mathbf{x}_i y_i$$

The Perceptron Training Algorithm

Algorithm 1 The perceptron training algorithm

```
Initialize w
while TRUE do
    m ← 0
    for each (x_i, y_i) ∈ D do
        if (w^T x_i) y_i < 0 then
            w ← w + α x_i y_i
            m ← m + 1
        end if
    end for
    if m = 0 then return
    end if
end while
```

A Running Example

```
initialize w
while TRUE do:
    m=0
    foreach (x_i, y_i) do
        if (w^T x_i) y_i ≤ 0
            w = w + α y_i x_i
            m=m+1

    if m = 0
      break
```

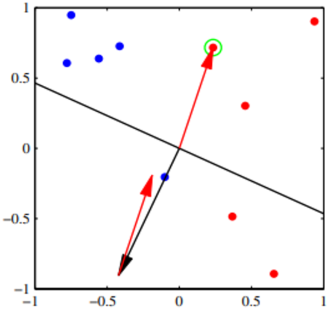
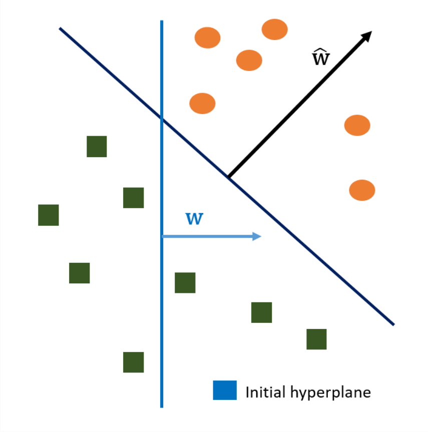


Illustration adapted from Fig 4.7 PRML C. Bishop

Convergence

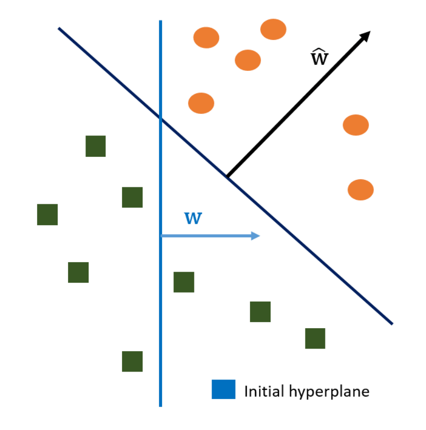
- The Perceptron provides a strong formal guarantee of convergence.
- If the data is linearly separable, the perceptron always finds a separating hyperplane in a finite number of steps.



Convergence

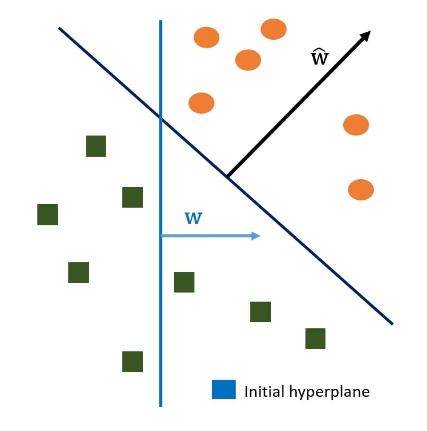
- The Perceptron provides a strong formal guarantee of convergence.
- If the data is linearly separable, the perceptron always finds a separating hyperplane in a finite number of steps.

- **Question:** From the figure, how can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?



Convergence: Setup

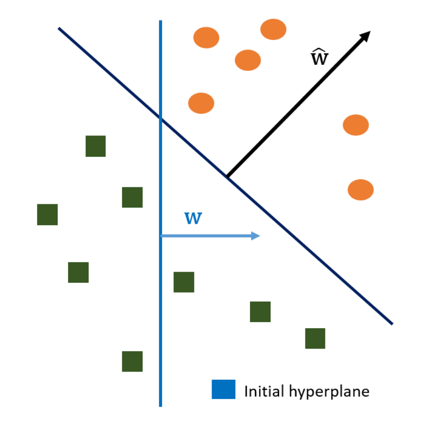
How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?



Convergence: Setup

How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?

1. $\mathbf{w}^T\hat{\mathbf{w}}$:

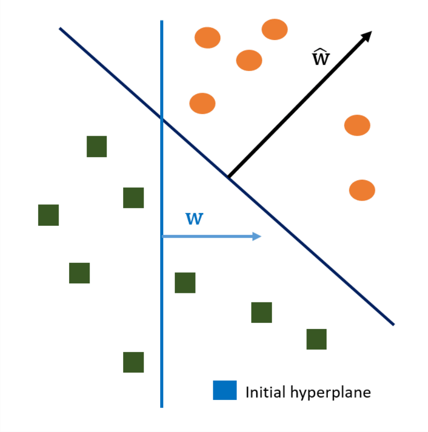


Convergence: Setup

How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?

1. $\mathbf{w}^T \hat{\mathbf{w}}$: Measures alignment between $\mathbf{w}$ and $\hat{\mathbf{w}}$

2. $\mathbf{w}^T \mathbf{w}$:

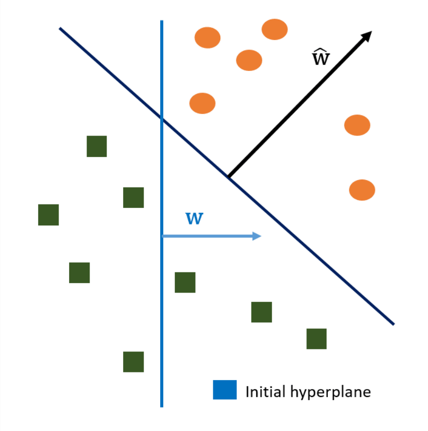


Convergence: Setup

How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?

1. $\mathbf{w}^T\hat{\mathbf{w}}$: Measures alignment between $\mathbf{w}$ and $\hat{\mathbf{w}}$

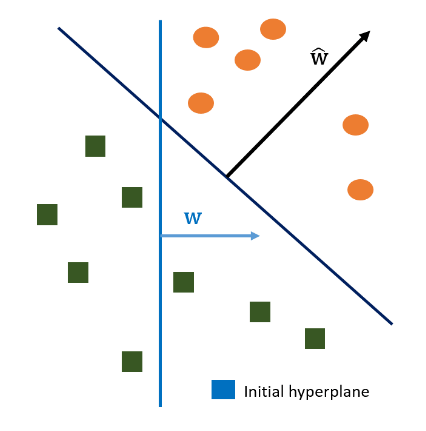
2. $\mathbf{w}^T\mathbf{w}$: Guarantees that an increase in $\mathbf{w}^T\hat{\mathbf{w}}$ is not just because $\mathbf{w}$ is growing



Convergence: Setup

How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?

1. $\mathbf{w}^T\hat{\mathbf{w}}$: Measures alignment between $\mathbf{w}$ and $\hat{\mathbf{w}}$
2. $\mathbf{w}^T\mathbf{w}$: Guarantees that an increase in $\mathbf{w}^T\hat{\mathbf{w}}$ is not just because $\mathbf{w}$ is growing



Given an update of $\mathbf{w}$:

$$\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x},$$

Convergence: Setup

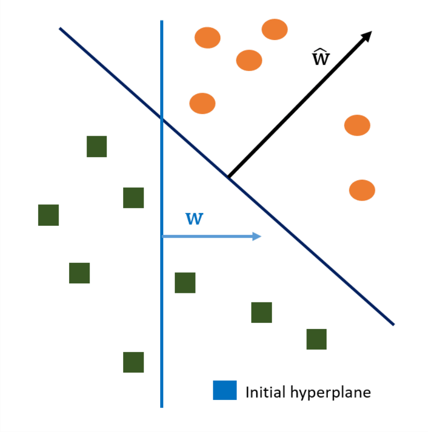
How can we measure that $\mathbf{w} \longrightarrow \hat{\mathbf{w}}$?

1. $\mathbf{w}^T\hat{\mathbf{w}}$: Measures alignment between $\mathbf{w}$ and $\hat{\mathbf{w}}$
2. $\mathbf{w}^T\mathbf{w}$: Guarantees that an increase in $\mathbf{w}^T\hat{\mathbf{w}}$ is not just because $\mathbf{w}$ is growing

Given an update of $\mathbf{w}$:

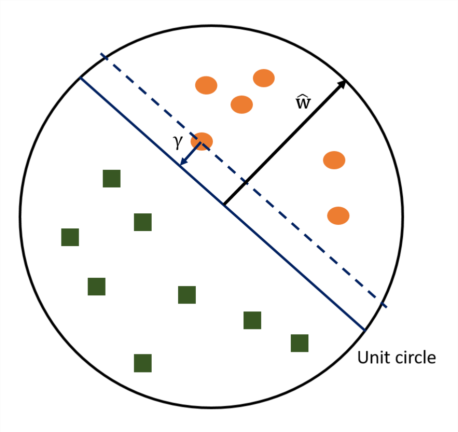
$$\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x},$$

We will see which effects this has on $\mathbf{w}^T\hat{\mathbf{w}}$ and $\mathbf{w}^T\mathbf{w}$



Setup and Definitions

Suppose $\exists \hat{\mathbf{w}}$ such that $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i) > 0 \, \forall \, (\mathbf{x}_i, y_i) \in \mathcal{D}$.

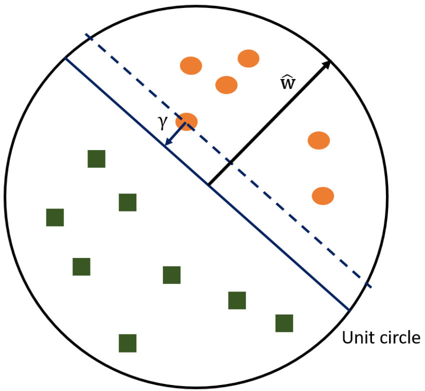


Setup and Definitions

Suppose $\exists \hat{\mathbf{w}}$ such that $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i) > 0 \, \forall \, (\mathbf{x}_i, y_i) \in \mathcal{D}$.

We rescale every point in $\mathcal{D}$ and the $\hat{\mathbf{w}}$ such that:

$$\|\mathbf{x}_i\| \leq 1 \, \forall \, \mathbf{x}_i \in \mathcal{D} \qquad \text{and} \qquad \|\hat{\mathbf{w}}\| = 1$$



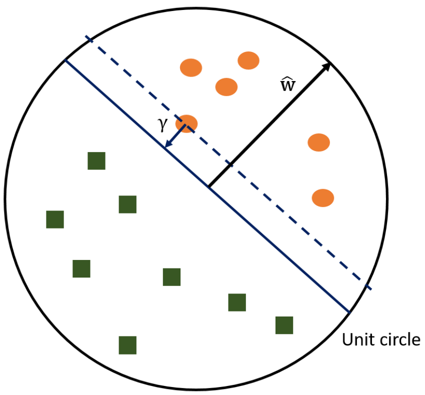
Setup and Definitions

Suppose $\exists \hat{\mathbf{w}}$ such that $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i) > 0 \, \forall \, (\mathbf{x}_i, y_i) \in \mathcal{D}$.

We rescale every point in $\mathcal{D}$ and the $\hat{\mathbf{w}}$ such that:

$$\|\mathbf{x}_i\| \leq 1 \, \forall \, \mathbf{x}_i \in \mathcal{D} \qquad \text{and} \qquad \|\hat{\mathbf{w}}\| = 1$$

In words: All features live within a unit sphere.



Setup and Definitions

Suppose $\exists \hat{\mathbf{w}}$ such that $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i) > 0 \, \forall \, (\mathbf{x}_i, y_i) \in \mathcal{D}$.

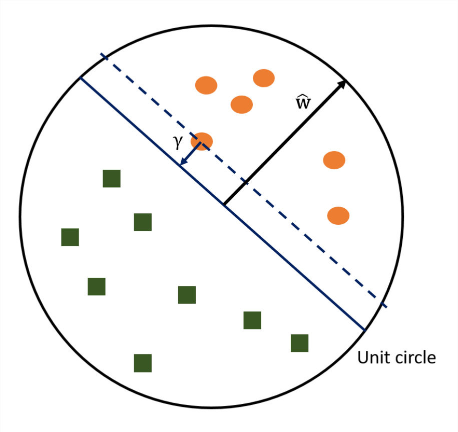
We rescale every point in $\mathcal{D}$ and the $\hat{\mathbf{w}}$ such that:

$$\|\mathbf{x}_i\| \leq 1 \, \forall \, \mathbf{x}_i \in \mathcal{D} \qquad \text{and} \qquad \|\hat{\mathbf{w}}\| = 1$$

In words: All features live within a unit sphere.

<u>Definition:</u> The margin γ of the hyperplane

$$\gamma = \min_{(\mathbf{x}_i, y_i) \in \mathcal{D}} |\hat{\mathbf{w}}^T \mathbf{x}_i| \tag{1}$$



Setup and Definitions

Suppose $\exists \hat{\mathbf{w}}$ such that $y_i(\hat{\mathbf{w}}^T \mathbf{x}_i) > 0 \, \forall \, (\mathbf{x}_i, y_i) \in \mathcal{D}$.

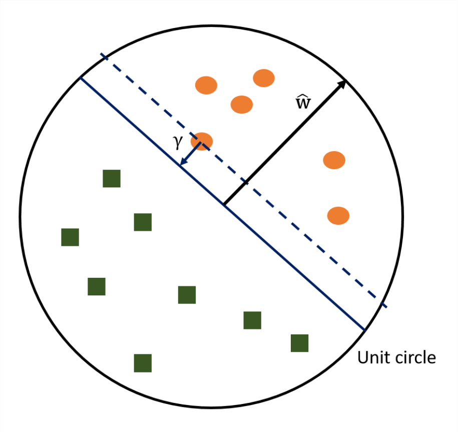
We rescale every point in $\mathcal{D}$ and the $\hat{\mathbf{w}}$ such that:

$$\|\mathbf{x}_i\| \leq 1 \, \forall \, \mathbf{x}_i \in \mathcal{D} \qquad \text{and} \qquad \|\hat{\mathbf{w}}\| = 1$$

In words: All features live within a unit sphere.

Definition: The margin γ of the hyperplane

$$\gamma = \min_{(\mathbf{x}_i, y_i) \in \mathcal{D}} |\hat{\mathbf{w}}^T \mathbf{x}_i| \tag{1}$$



Theorem

If the above holds, the perceptron algorithm takes at most $1/\gamma^2$ to converge.

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$:

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$:

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Given an update of $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$, we replace the expression in $\mathbf{w}^T\hat{\mathbf{w}}$:

$$\mathbf{w}^T\hat{\mathbf{w}} = (\mathbf{w} + y\mathbf{x})^T\hat{\mathbf{w}}$$

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Given an update of $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$, we replace the expression in $\mathbf{w}^T\hat{\mathbf{w}}$:

$$\begin{aligned}\mathbf{w}^T\hat{\mathbf{w}} &= (\mathbf{w} + y\mathbf{x})^T\hat{\mathbf{w}} \\ &= \mathbf{w}^T\hat{\mathbf{w}} + (y\mathbf{x})^T\hat{\mathbf{w}}\end{aligned}$$

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Given an update of $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$, we replace the expression in $\mathbf{w}^T\hat{\mathbf{w}}$:

$$\begin{aligned}
\mathbf{w}^T\hat{\mathbf{w}} &= (\mathbf{w} + y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + (y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + y\hat{\mathbf{w}}^T\mathbf{x}
\end{aligned}$$

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Given an update of $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$, we replace the expression in $\mathbf{w}^T\hat{\mathbf{w}}$:

$$\begin{aligned}
\mathbf{w}^T\hat{\mathbf{w}} &= (\mathbf{w} + y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + (y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + y\hat{\mathbf{w}}^T\mathbf{x}
\end{aligned}$$

We know that the second term is positive and, from the the margin definition (Eq. 1), we know that $|\hat{\mathbf{w}}^T\mathbf{x}|$ is, at least, γ. Replacing this, we get the following lower bound:

Proof - Step 1: Effect on $\mathbf{w}^T\hat{\mathbf{w}}$

For the proof, we need to keep in mind that:

- $y(\mathbf{w}^T\mathbf{x}) < 0$: because an update only occurs when there is a mistake
- $y(\hat{\mathbf{w}}^T\mathbf{x}) > 0$: because the separating hyperplane $\hat{\mathbf{w}}$ classifies correctly every point

Given an update of $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$, we replace the expression in $\mathbf{w}^T\hat{\mathbf{w}}$:

$$\begin{aligned}
\mathbf{w}^T\hat{\mathbf{w}} &= (\mathbf{w} + y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + (y\mathbf{x})^T\hat{\mathbf{w}} \\
&= \mathbf{w}^T\hat{\mathbf{w}} + y\hat{\mathbf{w}}^T\mathbf{x}
\end{aligned}$$

We know that the second term is positive and, from the the margin definition (Eq. 1), we know that $|\hat{\mathbf{w}}^T\mathbf{x}|$ is, at least, γ. Replacing this, we get the following lower bound:

$$\mathbf{w}^T\hat{\mathbf{w}} = \mathbf{w}^T\hat{\mathbf{w}} + y\hat{\mathbf{w}}^T\mathbf{x} \geq \mathbf{w}^T\hat{\mathbf{w}} + \gamma \tag{2}$$

Proof - Step 2: Effect on $\mathbf{w}^T\mathbf{w}$

Let is now replace the update $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$ in the second expression $\mathbf{w}^T\mathbf{w}$:

$$\mathbf{w}^T\mathbf{w} = (\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x})$$

Proof - Step 2: Effect on $\mathbf{w}^T\mathbf{w}$

Let is now replace the update $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$ in the second expression $\mathbf{w}^T\mathbf{w}$:

$$
\begin{aligned}
\mathbf{w}^T\mathbf{w} &= (\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x}) \\
&= \mathbf{w}^T\mathbf{w} + 2y\mathbf{w}^T\mathbf{x} + y^2\mathbf{x}^T\mathbf{x}
\end{aligned}
$$

Proof - Step 2: Effect on $\mathbf{w}^T\mathbf{w}$

Let is now replace the update $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$ in the second expression $\mathbf{w}^T\mathbf{w}$:

$$\begin{aligned}\mathbf{w}^T\mathbf{w} &= (\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x}) \\ &= \mathbf{w}^T\mathbf{w} + 2y\mathbf{w}^T\mathbf{x} + y^2\mathbf{x}^T\mathbf{x}\end{aligned}$$

About this expression we know that:

- $2y\mathbf{w}^T\mathbf{x} < 0$ (why?)

Proof - Step 2: Effect on $\mathbf{w}^T\mathbf{w}$

Let is now replace the update $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$ in the second expression $\mathbf{w}^T\mathbf{w}$:

$$\begin{aligned}\mathbf{w}^T\mathbf{w} &= (\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x}) \\ &= \mathbf{w}^T\mathbf{w} + 2y\mathbf{w}^T\mathbf{x} + y^2\mathbf{x}^T\mathbf{x}\end{aligned}$$

About this expression we know that:

- $2y\mathbf{w}^T\mathbf{x} < 0$ (why?)
- $0 \leq y^2\mathbf{x}^T\mathbf{x} \leq 1$ (why?)

Proof - Step 2: Effect on $\mathbf{w}^T\mathbf{w}$

Let is now replace the update $\mathbf{w} \longleftarrow \mathbf{w} + y\mathbf{x}$ in the second expression $\mathbf{w}^T\mathbf{w}$:

$$\begin{aligned}\mathbf{w}^T\mathbf{w} &= (\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x}) \\ &= \mathbf{w}^T\mathbf{w} + 2y\mathbf{w}^T\mathbf{x} + y^2\mathbf{x}^T\mathbf{x}\end{aligned}$$

About this expression we know that:

- $2y\mathbf{w}^T\mathbf{x} < 0$ (why?)
- $0 \leq y^2\mathbf{x}^T\mathbf{x} \leq 1$ (why?)

Taking this into account, we have:

$$(\mathbf{w} + y\mathbf{x})^T(\mathbf{w} + y\mathbf{x}) = \mathbf{w}^T\mathbf{w} + 2y\mathbf{w}^T\mathbf{x} + y^2\mathbf{x}^T\mathbf{x} \leq \mathbf{w}^T\mathbf{w} + 1 \tag{3}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T \mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T \hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T\mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T\hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Starting from Eq. 5, by definition of the dot product we have:

$$M\gamma \leq \|\mathbf{w}\|\|\hat{\mathbf{w}}\| \cos\theta \qquad\qquad \theta \text{ the angle between the two}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T\mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T\hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Starting from Eq. 5, by definition of the dot product we have:

$$\begin{aligned} M\gamma &\leq \|\mathbf{w}\|\|\hat{\mathbf{w}}\|\cos\theta && \theta \text{ the angle between the two} \\ &\leq \|\mathbf{w}\| && \text{by definition of } \cos, \cos\theta \leq 1 \end{aligned}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T\mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T\hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Starting from Eq. 5, by definition of the dot product we have:

$$\begin{aligned} M\gamma &\leq \|\mathbf{w}\|\|\hat{\mathbf{w}}\|\cos\theta && \theta \text{ the angle between the two} \\ &\leq \|\mathbf{w}\| && \text{by definition of } \cos, \cos\theta \leq 1 \\ &\leq \sqrt{\mathbf{w}^T\mathbf{w}} && \text{by definition of} \|\cdot\| \end{aligned}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T\mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T\hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Starting from Eq. 5, by definition of the dot product we have:

$$\begin{aligned}
M\gamma &\leq \|\mathbf{w}\|\|\hat{\mathbf{w}}\|\cos\theta && \theta \text{ the angle between the two} \\
&\leq \|\mathbf{w}\| && \text{by definition of } \cos, \cos\theta \leq 1 \\
&\leq \sqrt{\mathbf{w}^T\mathbf{w}} && \text{by definition of} \|\cdot\| \\
&\leq \sqrt{M} && \text{by replacing Eq. 4}
\end{aligned}$$

Proof - Step 3: M updates

After M updates in the perceptron algorithm, from Eq. 2 and 3 the following should hold:

$$\mathbf{w}^T\mathbf{w} \leq M \tag{4}$$

$$\mathbf{w}^T\hat{\mathbf{w}} \geq M\gamma \tag{5}$$

Starting from Eq. 5, by definition of the dot product we have:

$$\begin{aligned}
M\gamma &\leq \|\mathbf{w}\|\|\hat{\mathbf{w}}\|\cos\theta && \theta \text{ the angle between the two} \\
&\leq \|\mathbf{w}\| && \text{by definition of } \cos, \cos\theta \leq 1 \\
&\leq \sqrt{\mathbf{w}^T\mathbf{w}} && \text{by definition of} \|\cdot\| \\
&\leq \sqrt{M} && \text{by replacing Eq. 4}
\end{aligned}$$

From which we can obtain an expression for M:

$$M^2\gamma^2 \leq M$$

$$M \leq \frac{1}{\gamma^2}$$

Perceptron's Convergence

The expression which we have obtained:

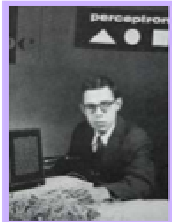
$$M \leq \frac{1}{\gamma^2}$$

is telling us that the number of updates M is upper bounded by a constant.

Exercise: Given this theorem, what can you say about the margin of a classifier? What is most desirable?

History & Limitations

Some History

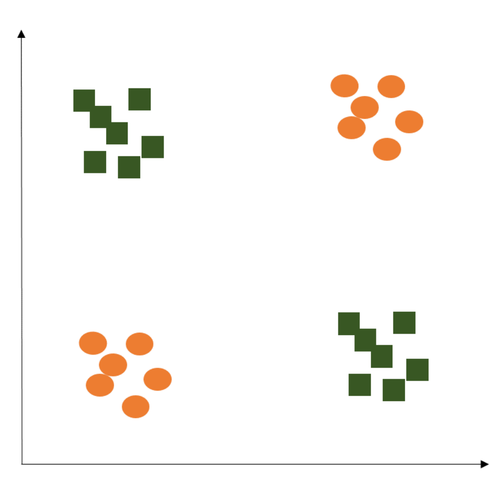


Frank Rosenblatt
1928–1969

Rosenblatt's perceptron played an important role in the history of machine learning. Initially, Rosenblatt simulated the perceptron on an IBM 704 computer at Cornell in 1957, but by the early 1960s he had built special-purpose hardware that provided a direct, parallel implementation of perceptron learning. Many of his ideas were encapsulated in "Principles of Neuro-dynamics: Perceptrons and the Theory of Brain Mechanisms" published in 1962. Rosenblatt's work was criticized by Marvin Minksy, whose objections were published in the book "Perceptrons", co-authored with Seymour Papert. This book was widely misinterpreted at the time as showing that neural networks were fatally flawed and could only learn solutions for linearly separable problems. In fact, it only proved such limitations in the case of single-layer networks such as the perceptron and merely conjectured (incorrectly) that they applied to more general network models. Unfortunately, however, this book contributed to the substantial decline in research funding for neural computing, a situation that was not reversed until the mid-1980s. Today, there are many hundreds, if not thousands, of applications of neural networks in widespread use, with examples in areas such as handwriting recognition and information retrieval being used routinely by millions of people.

Source: PRML - C. Bishop

The XOR Problem



Other limitations

- The algorithm does not converge when the data are not separable
- When the data is separable, there are many solutions, and which one is found depends on the starting values
- The **finite** number of steps can be very large.

Recap

Recap

In this lecture...

- We introduced the perceptron algorithm, a linear classifier that guarantees convergence
- The perceptron looks for a hyperplane that can linearly separate data
- We saw that it guarantees a solution for linearly separable data
- But we also saw that it has numerous limitations

Key Concepts

- Hyperplane
- The Perceptron Criterion
- Linearly separable data
- Convergence

References

Further Reading and Useful Material

Source	Notes
Pattern Recognition and Machine Learning	Sec 4.1.7
The Elements of Statistical Learning	Sec. 4.5
Rosenblatt's article	The Perceptron (link)