Data Science Theory and Practices

The Perceptron Algorithm

Perfectly Separable and Inseparable Data

Joseph Chuang-Chieh Lin
Dept. CSIE, Tamkang University

31st May 2021

Learning to Classify

- A core problem underlying many machine learning applications.

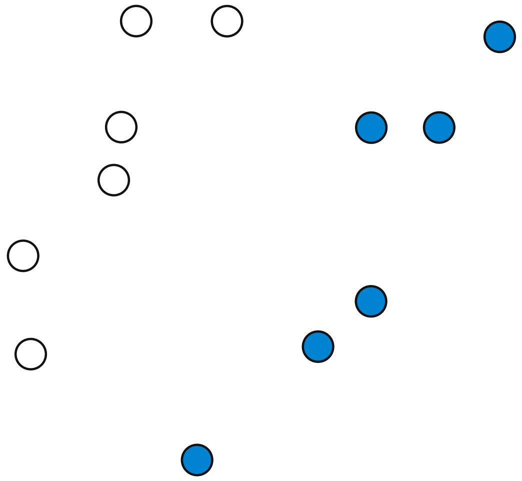
Learning to Classify

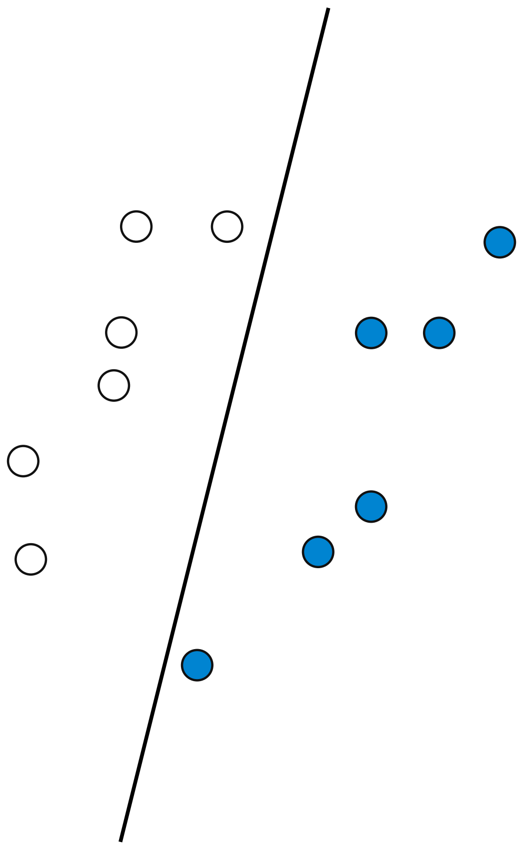
- A core problem underlying many machine learning applications.
- To help ground our discussion, we begin by a specific learning algorithm: the **Perceptron algorithm**.

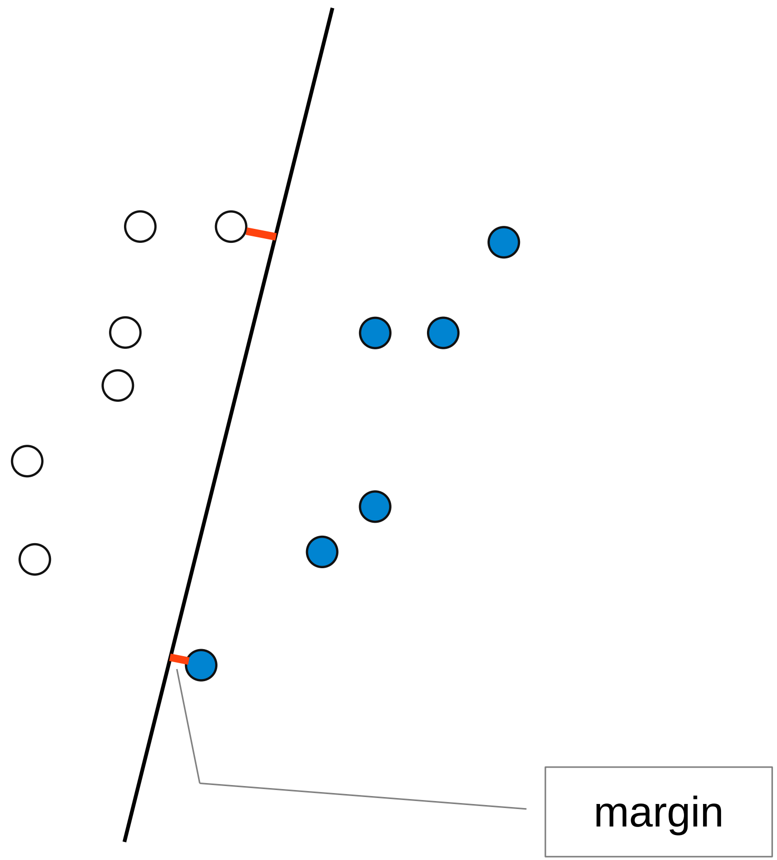
Learning to Classify

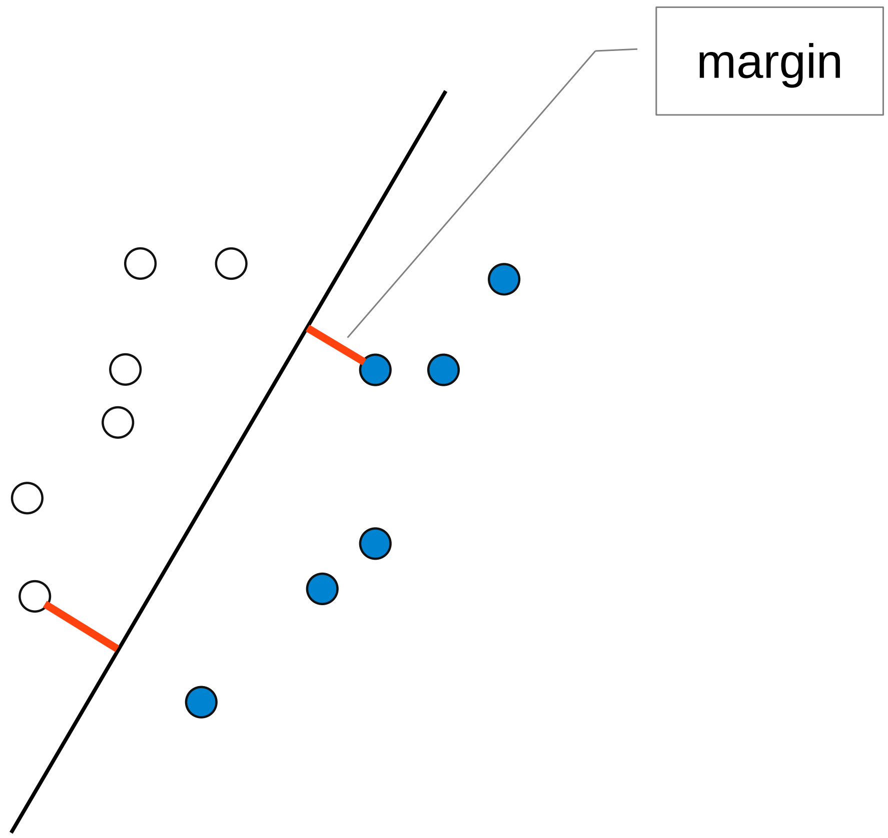
- A core problem underlying many machine learning applications.
- To help ground our discussion, we begin by a specific learning algorithm: the **Perceptron algorithm**.
  - Assigning positive and negative weights to features.
  - Each positive example has a positive sum of weights.
  - Each negative example has a negative sum of weights.

A set of S in R^d, $d = 2$





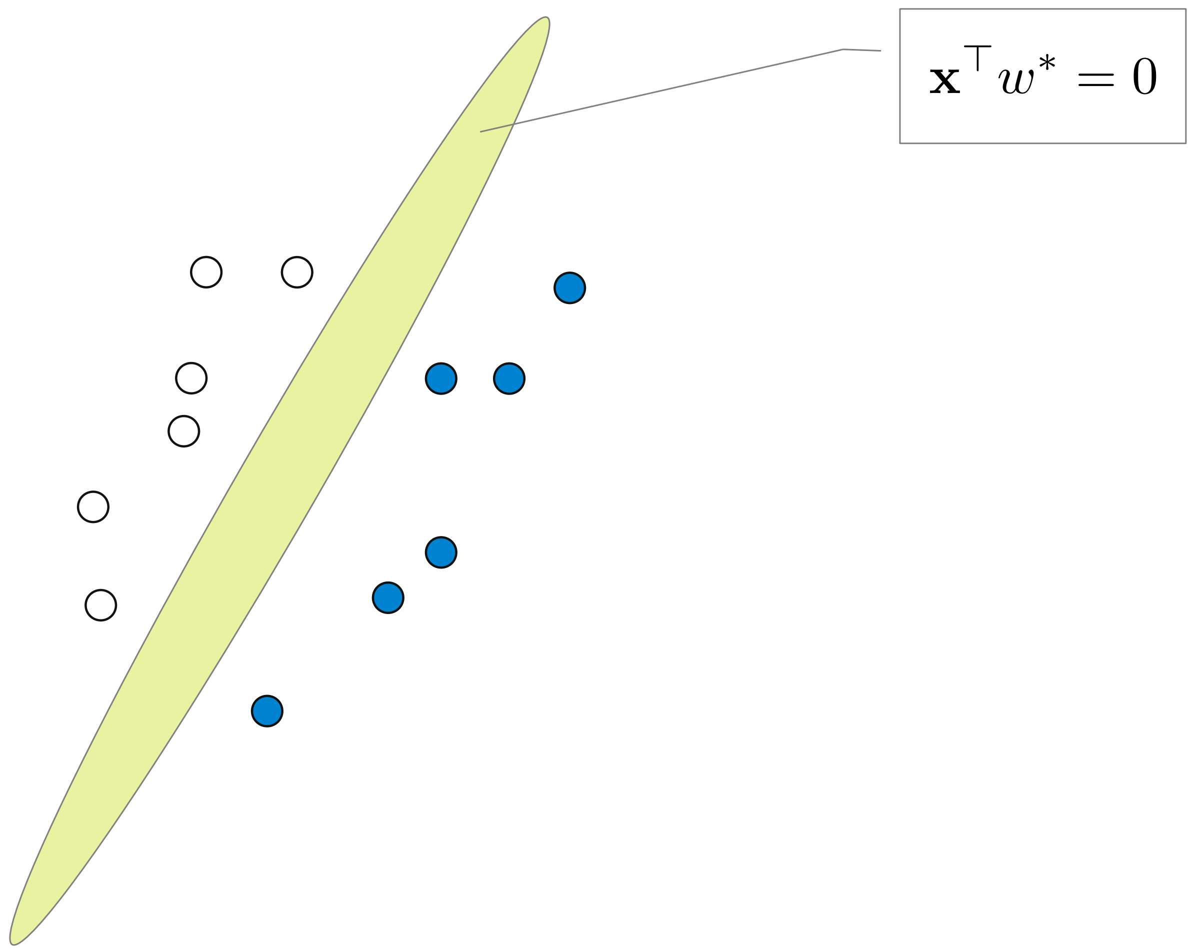

margin

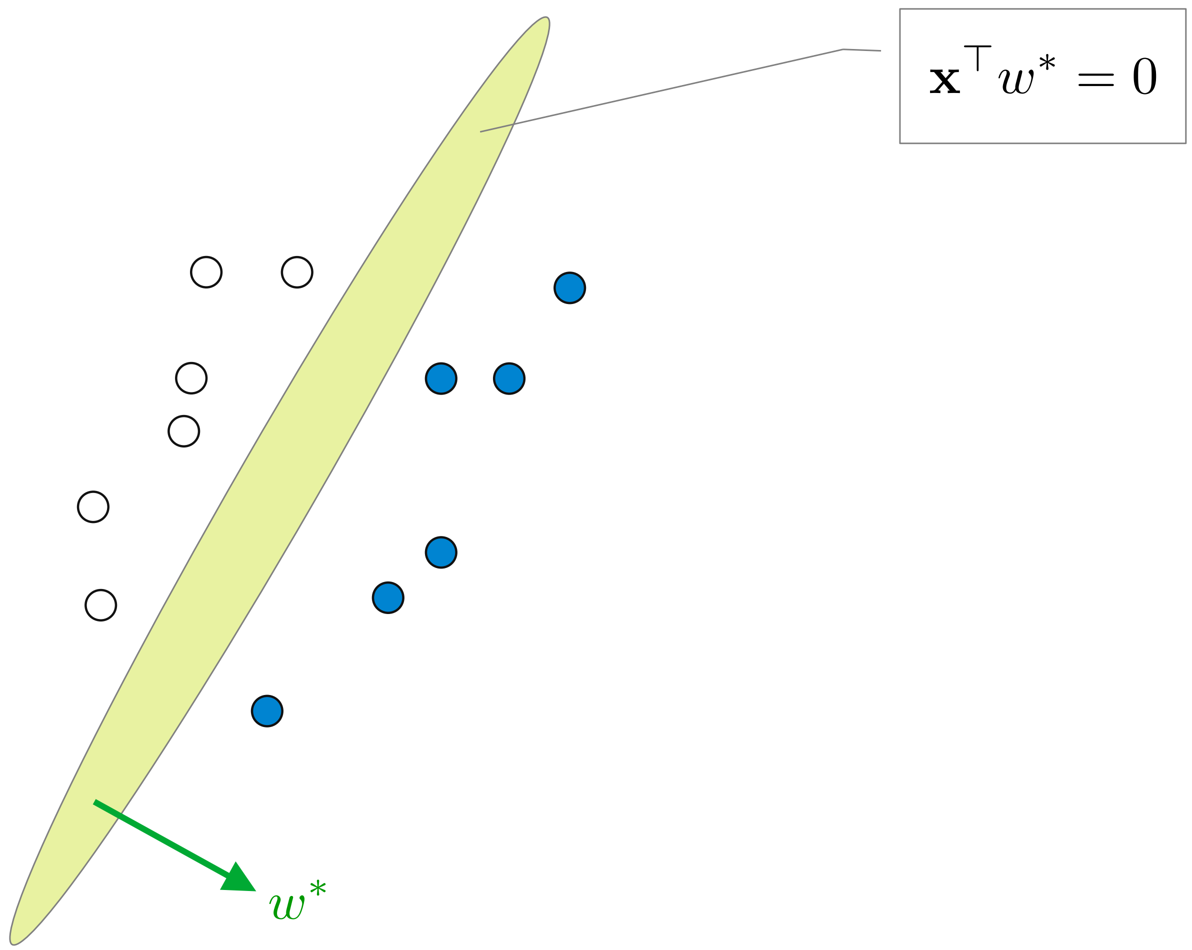

margin

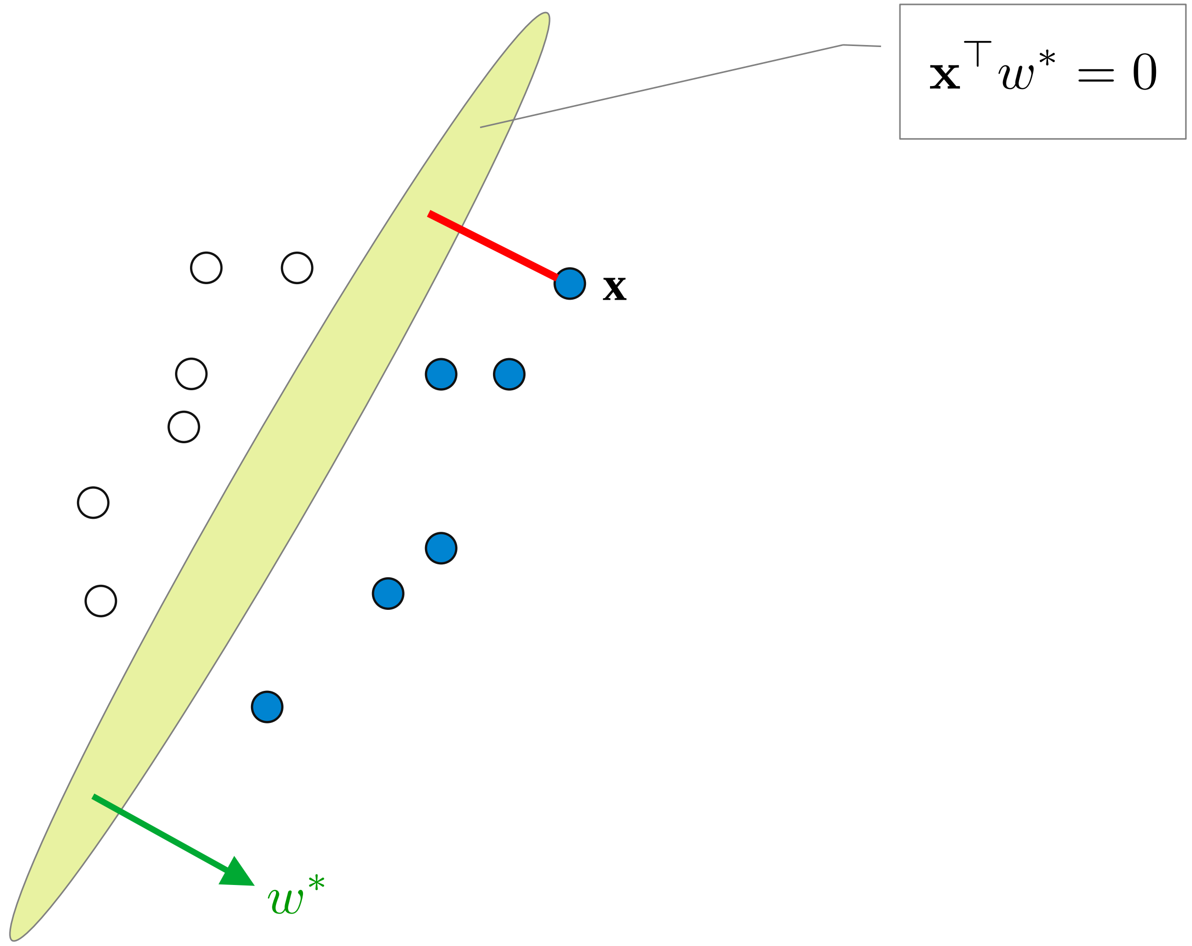
Data and the assumption

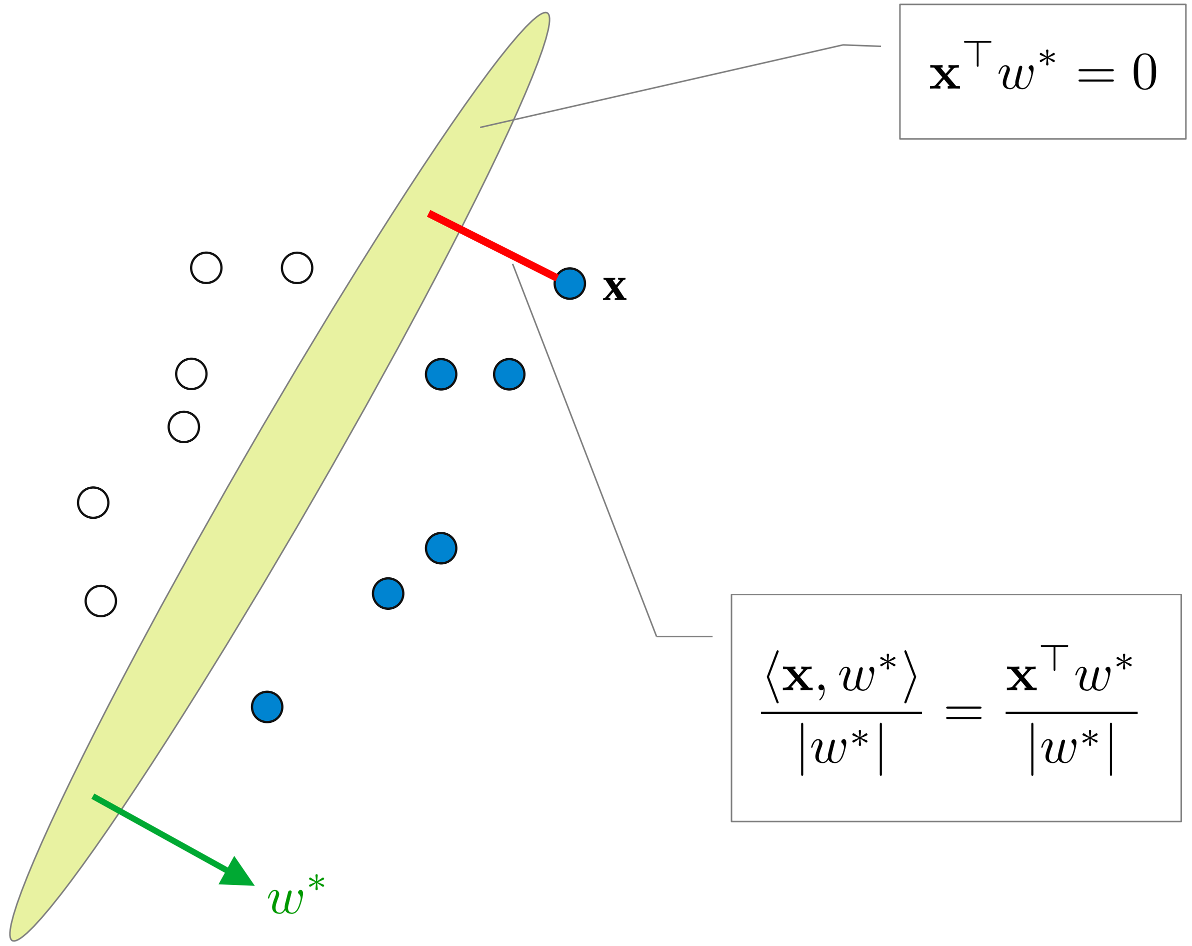
- Given a set of points in R^d space.
  - Each is labeled as positive (+) or negative (−).

- Assumption: There exists a vector w^* such that
  - For each positive example $\mathbf{x} \in S$, $\mathbf{x}^\top w^* \geq 1$
  - For each negative example $\mathbf{x} \in S$, $\mathbf{x}^\top w^* \leq -1$


$\mathbf{x}^\top w^* = 0$


$\mathbf{x}^\top w^* = 0$
w^*

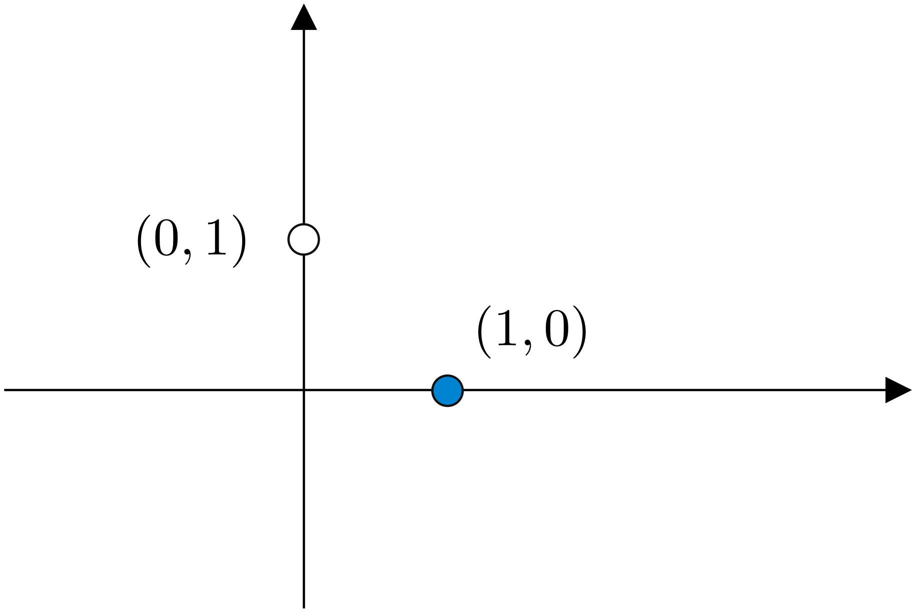

$\mathbf{x}^\top w^* = 0$
$\mathbf{x}$
w^*


$\mathbf{x}^\top w^* = 0$
$\mathbf{x}$
$\dfrac{\langle \mathbf{x}, w^* \rangle}{|w^*|} = \dfrac{\mathbf{x}^\top w^*}{|w^*|}$
w^*

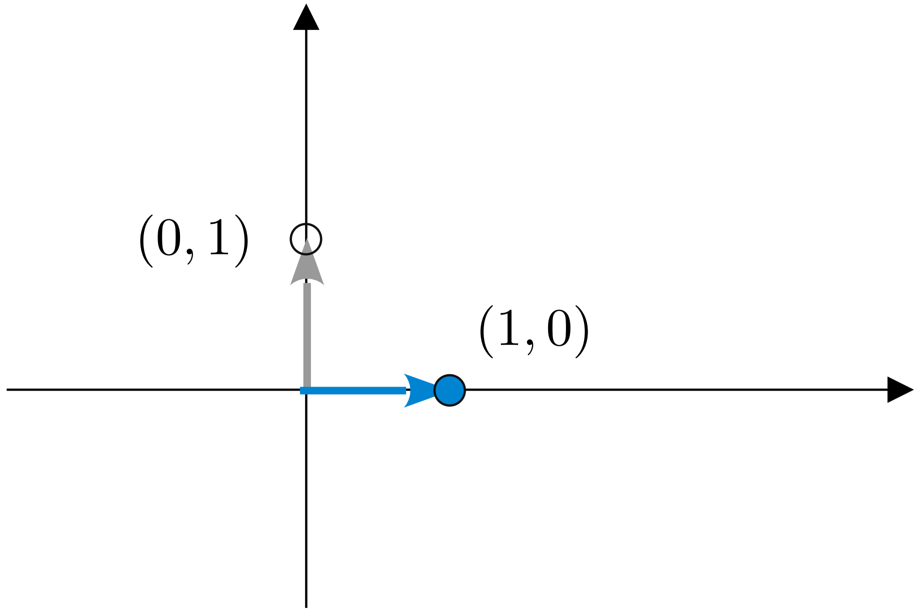
The Perceptron algorithm

- Start with $w = \mathbf{0}$.
- Repeat the following until $\mathbf{x}^\top w$ has the correct sign for all $\mathbf{x} \in S$.
  - Let $\mathbf{x} \in S$ be an example for which the sign of $\mathbf{x}^\top w$ is NOT correct.
  - Update as follows:
    - If $\mathbf{x}$ is a positive example, let $w \leftarrow w + \mathbf{x}$.
    - If $\mathbf{x}$ is a negative example, let $w \leftarrow w - \mathbf{x}$.

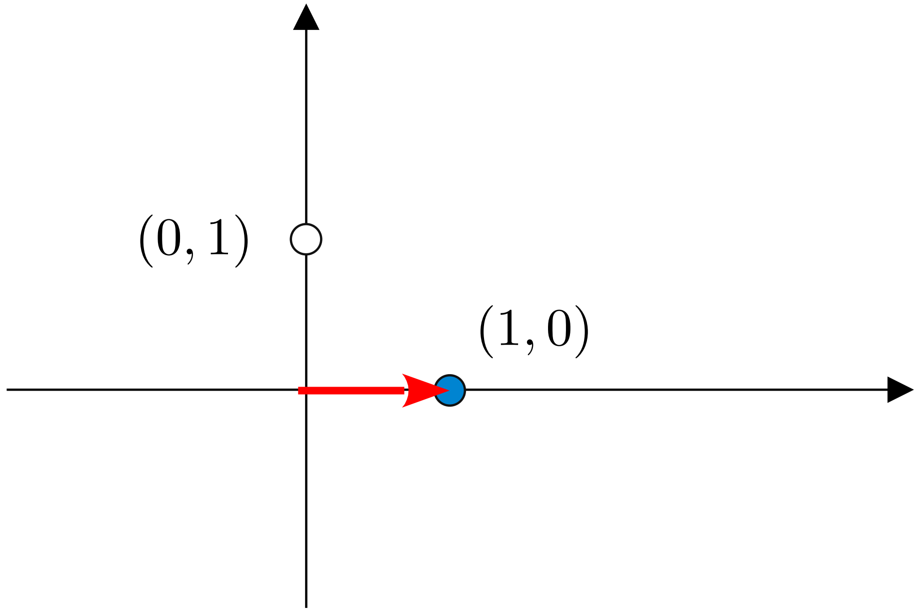
An illustrating example



An illustrating example

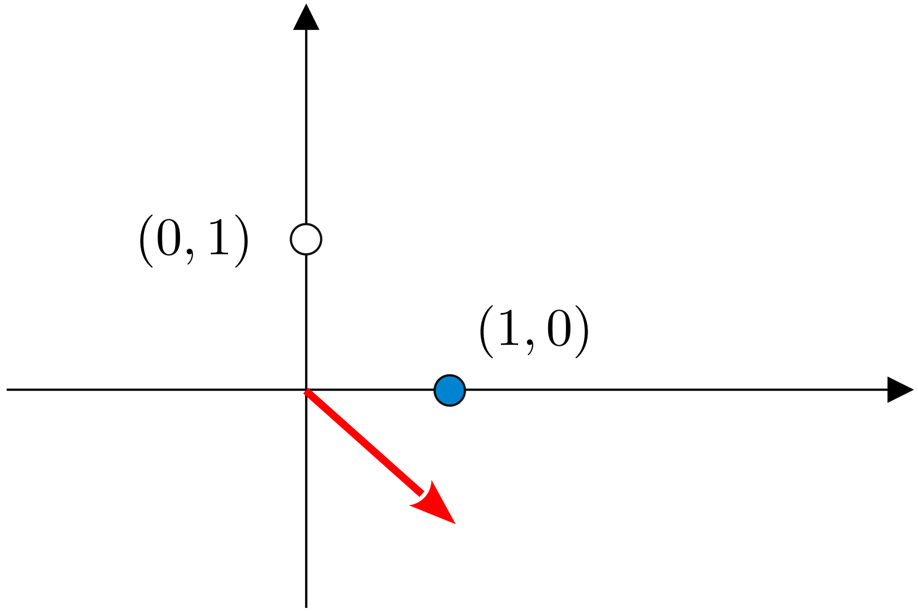


An illustrating example



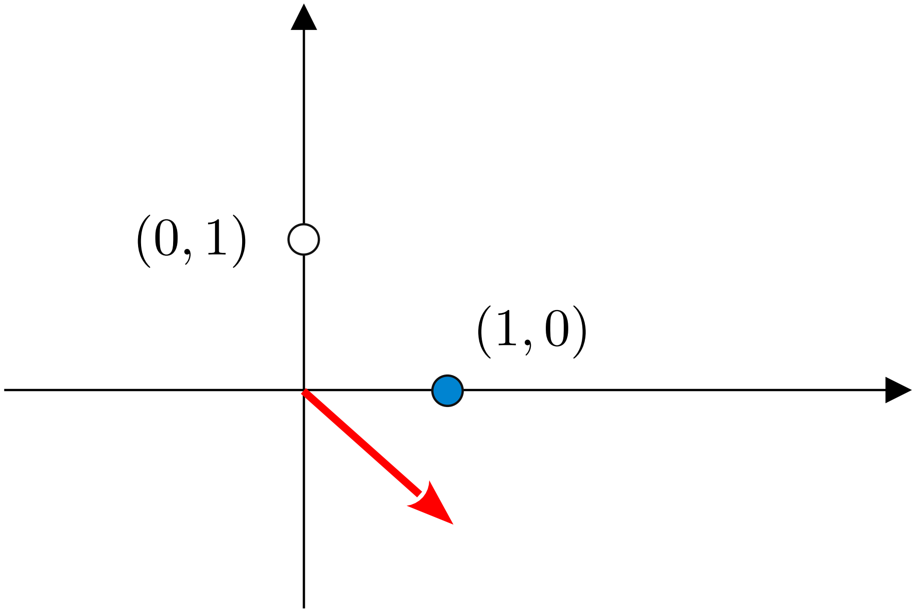
$$w = (0,0) + (1,0) = (1,0)$$

An illustrating example



$$w = (1,0) - (0,1) = (1,-1)$$

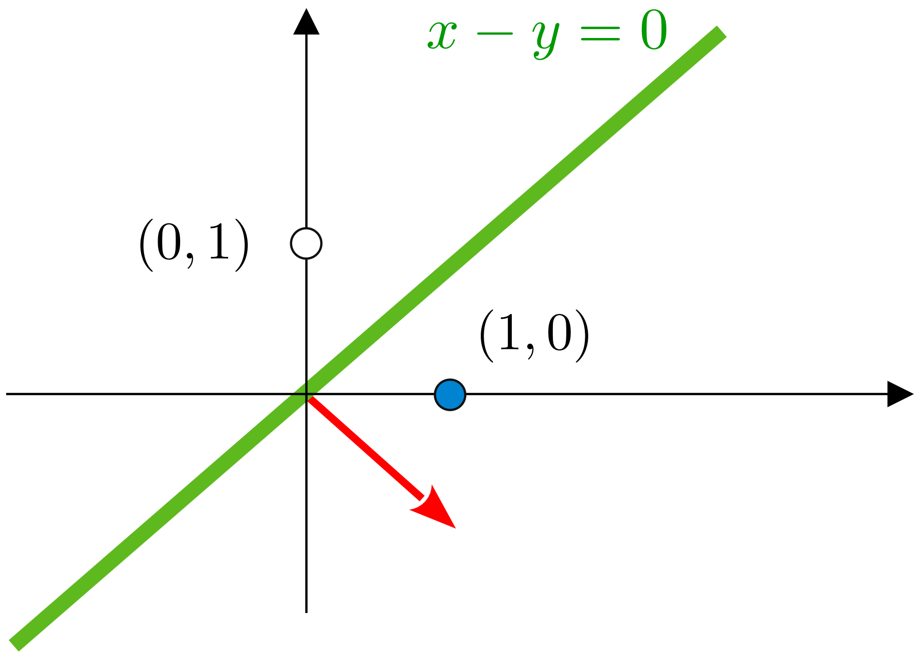
An illustrating example



Check:

$$\text{For } (1,0): \quad (1,-1)\cdot(1,0) = 1 > 0$$
$$\text{For } (0,1): \quad (1,-1)\cdot(0,1) = -1 < 0$$

An illustrating example



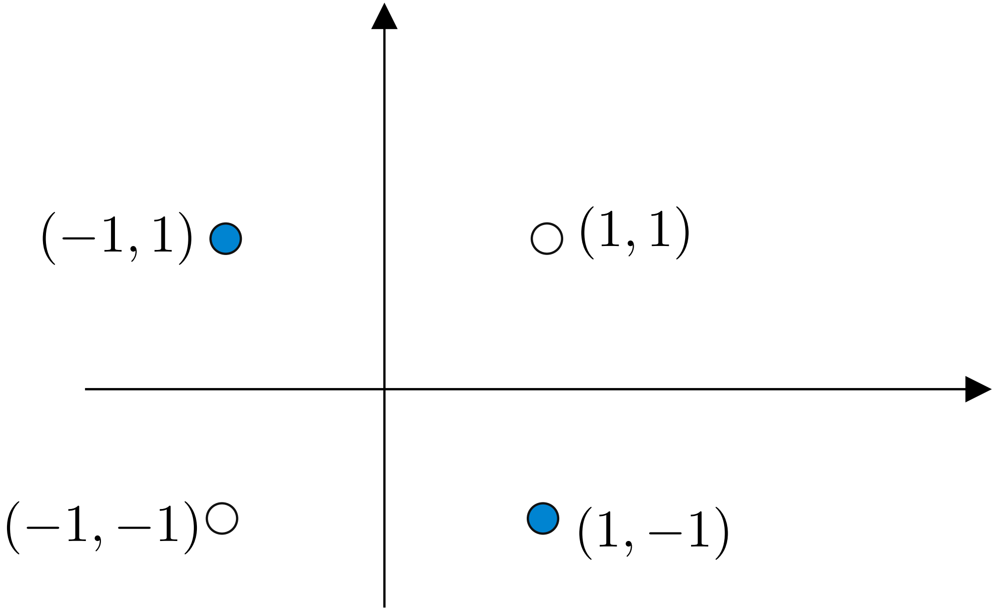
Check:

$$\text{For } (1, 0): \quad (1, -1) \cdot (1, 0) = 1 > 0$$

$$\text{For } (0, 1): \quad (1, -1) \cdot (0, 1) = -1 < 0$$

An Exercise

- Try to work on this example:



Theorem

- If there exists a vector w^* such that

$$\mathbf{x}^\top w^* \geq 1 \text{ for all positive } \mathbf{x} \in S$$
$$\mathbf{x}^\top w^* \leq -1 \text{ for all negative } \mathbf{x} \in S$$

then the number of updates of the Perceptron algorithm is at most $R^2|w^*|^2$, where $R = \max_{\mathbf{x} \in S} |\mathbf{x}|$.

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- For each update,
  - if $\mathbf{x}$ is a positive example, $(w + \mathbf{x})^\top w^* = w^\top w^* + \mathbf{x}^\top w^*$
  - if $\mathbf{x}$ is a negative example, $(w - \mathbf{x})^\top w^* = w^\top w^* - \mathbf{x}^\top w^*$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- For each update,
  - if $\mathbf{x}$ is a positive example, $(w + \mathbf{x})^\top w^* = w^\top w^* + \mathbf{x}^\top w^* \geq w^\top w^* + 1.$
  - if $\mathbf{x}$ is a negative example, $(w - \mathbf{x})^\top w^* = w^\top w^* - \mathbf{x}^\top w^* \geq w^\top w^* + 1.$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- For each update,
  - if $\mathbf{x}$ is a positive example, $(w+\mathbf{x})^\top(w+\mathbf{x}) = |w|^2 + 2\mathbf{x}^\top w + |\mathbf{x}|^2$
  - if $\mathbf{x}$ is a negative example, $(w-\mathbf{x})^\top(w-\mathbf{x}) = |w|^2 - 2\mathbf{x}^\top w + |\mathbf{x}|^2$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- For each **update**,
  - if $\mathbf{x}$ is a positive example, $(w+\mathbf{x})^\top(w+\mathbf{x}) = |w|^2 + 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + |\mathbf{x}|^2.$
  - if $\mathbf{x}$ is a negative example, $(w-\mathbf{x})^\top(w-\mathbf{x}) = |w|^2 - 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + |\mathbf{x}|^2.$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- For each **update**,
  - if $\mathbf{x}$ is a positive example, $(w+\mathbf{x})^\top(w+\mathbf{x}) = |w|^2 + 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$
  - if $\mathbf{x}$ is a negative example, $(w-\mathbf{x})^\top(w-\mathbf{x}) = |w|^2 - 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of
$$w^\top w^* \text{ and } |w|^2.$$
- If we make M updates,
$$w^\top w^* \geq M \cdot 1 = M.$$
$$|w|^2 \leq MR^2.$$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- If we make M updates,

$$w^\top w^* \geq M \cdot 1 = M.$$

$$|w|^2 \leq MR^2.$$

Note: $\dfrac{w^\top w^*}{|w^*|} \leq |w|.$

Proof

- Fix some w^* satisfying the conditions.
- Keep track of

$$w^\top w^* \text{ and } |w|^2.$$

- If we make *M* updates,

$$w^\top w^* \geq M \cdot 1 = M.$$

$$|w|^2 \leq MR^2.$$

Note: $\dfrac{w^\top w^*}{|w^*|} \leq |w|.$

$$\therefore\ M \leq R^2 |w|^2.$$

Online learning scenario

- Remove the assumption that the data is sampled from a fixed distribution.

- The data points come on by one at time.

Online learning scenario

- Remove the assumption that the data is sampled from a fixed distribution.
- The data points come on by one at time.
- The Perceptron algorithm can be adjusted in the online setting.

The Perceptron algorithm (online)

- Start with $w = \mathbf{0}$. For t = 1, 2, …, do:
- Given example $\mathbf{x}_t$, predict sign($\mathbf{x}_t^\top w$).
- If the prediction was a mistake, then update:
  - If $\mathbf{x}$ is a positive example, let $w \leftarrow w + \mathbf{x}_t$.
  - If $\mathbf{x}$ is a negative example, let $w \leftarrow w - \mathbf{x}_t$.

Theorem

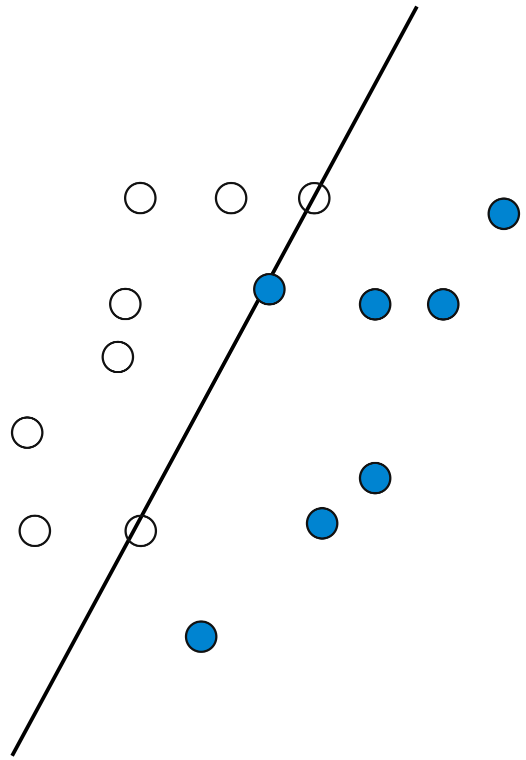
- Given any online sequence $\mathbf{x}_1$, $\mathbf{x}_2$, …
- If there exists a vector w^* such that

$$\mathbf{x}_t^\top w^* \geq 1 \text{ for all positive } \mathbf{x}_t$$
$$\mathbf{x}_t^\top w^* \leq -1 \text{ for all negative } \mathbf{x}_t$$

then the number of updates of the online Perceptron algorithm is at most $R^2|w^*|^2$, where $R = \max_t |\mathbf{x}_t|$.

Inseparable data

- What if the best w^* is not perfect?



Inseparable data

- What if the best w^* is not perfect?

$$\mathbf{x}_t^\top \mathbf{w}^* \text{ is just ``a bit wrong'' for some } \mathbf{x}_t$$

Hinge-loss

- The hinge-loss of w^* on a positive $\mathbf{x}_t$

$$\max(0, 1 - \mathbf{x}_t^\top w^*).$$

- The hinge-loss of w^* on a negative $\mathbf{x}_t$

$$\max(0, 1 + \mathbf{x}_t^\top w^*).$$

- Define the total hinge-loss $L_{\text{hinge}}(w^*, S)$

$$L_{\text{hinge}}(w^*, S): \quad \text{the sum of hinge-losses of } w^* \text{ on all } \mathbf{x}_t.$$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}\left(R^2|w^*|^2 + 2L(w^*, S)\right)$$

mistakes, where $R = \max_t |\mathbf{x}_t|.$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}\left(R^2|w^*|^2 + 2L(w^*, S)\right)$$

mistakes, where $R = \max_t |\mathbf{x}_t|$.

Proof:

For each **update**,

if $\mathbf{x}$ is a positive example, $(w+\mathbf{x})^\top(w+\mathbf{x}) = |w|^2 + 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$

if $\mathbf{x}$ is a negative example, $(w-\mathbf{x})^\top(w-\mathbf{x}) = |w|^2 - 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most
$$\min_{w^*}(R^2|w^*|+2L(w^*,S))$$
mistakes, where $R = \max_t |\mathbf{x}_t|$.

Proof:

For each **update**,

if $\mathbf{x}$ is a positive example, $(w+\mathbf{x})^\top(w+\mathbf{x}) = |w|^2 + 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$

if $\mathbf{x}$ is a negative example, $(w-\mathbf{x})^\top(w-\mathbf{x}) = |w|^2 - 2\mathbf{x}^\top w + |\mathbf{x}|^2 \leq |w|^2 + R^2.$

This part is just the same as before.

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|+2L(w^*, S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|.$

Proof:

- For each **update**, we increase $w^\top w^*$ by a $\mathbf{x}_t^\top w^*$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots$, the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*| + 2L(w^*, S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|$.

Proof:

- For each **update**, we increase $w^\top w^*$ by a $\mathbf{x}_t{}^\top w^*$

$$(w + \mathbf{x}_t)^\top w^* = w^\top w^* + \mathbf{x}_t^\top w^*.$$
$$(w - \mathbf{x}_t)^\top w^* = w^\top w^* - \mathbf{x}_t^\top w^*.$$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|+2L(w^*,S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|.$

Proof:

- For each **update**, we increase $w^\top w^*$ by a $\mathbf{x}_t{}^\top w^*$

$$(w+\mathbf{x}_t)^\top w^* = w^\top w^* + \mathbf{x}_t^\top w^*,$$
$$(w-\mathbf{x}_t)^\top w^* = w^\top w^* - \mathbf{x}_t^\top w^*)$$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots$, the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*| + 2L(w^*, S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|$.

Proof:

- For each **update**, we increase $w^\top w^*$ by $\mathbf{x}_t{}^\top w^*$ or $-\mathbf{x}_t{}^\top w^*$

$$(w + \mathbf{x}_t)^\top w^* = w^\top w^* + \mathbf{x}_t^\top w^*$$
$$(w - \mathbf{x}_t)^\top w^* = w^\top w^* - \mathbf{x}_t^\top w^*$$

$$\geq 1 - L_{\text{hinge}}(w^*, \mathbf{x}_t)$$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|+2L(w^*,S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|$.

Proof:

- For each **update**, we increase $w^\top w^*$ by $\mathbf{x}_t^\top w^*$ or $-\mathbf{x}_t^\top w^*$

$$(w+\mathbf{x}_t)^\top w^* = w^\top w^* + \mathbf{x}_t^\top w^*$$
$$(w-\mathbf{x}_t)^\top w^* = w^\top w^* - \mathbf{x}_t^\top w^*$$

$\geq 1 - L_{\text{hinge}}(w^*, \mathbf{x}_t)$

$$\text{Sum over all mistakes} \Rightarrow w^\top w^* \geq M - L_{\text{hinge}}(w^*, S).$$

Theorem

- On any sequence of examples $S = \mathbf{x}_1, \mathbf{x}_2, \ldots,$ the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|+2L(w^*,S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|.$

Proof: Sum over all mistakes $\Rightarrow$ $w^\top w^* \geq M - L_{\text{hinge}}(w^*,S) := M - L.$

$$\begin{aligned}
w^\top w^*/|w^*| &\leq |w| \\
(M-L)^2 \leq (w^\top w^*)^2 &\leq |w|^2|w^*|^2 \\
(M-L)^2 &\leq MR^2|w|^2 \\
M^2 - 2ML + L^2 &\leq MR^2|w^*|^2 \\
M - 2L + L^2/M &\leq R^2|w^*|^2.
\end{aligned}$$

Theorem

- On any sequence of examples S = $\mathbf{x}_1$, $\mathbf{x}_2$, …, the Perceptron algorithm makes at most

$$\min_{w^*}(R^2|w^*|+2L(w^*,S))$$

mistakes, where $R = \max_t |\mathbf{x}_t|$.

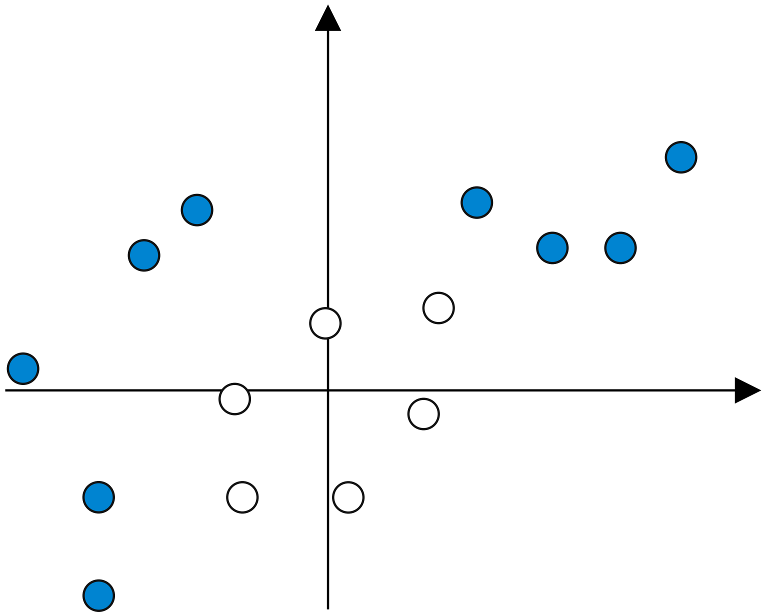
Proof: Sum over all mistakes $\Rightarrow$ $w^\top w^* \geq M - L_{\text{hinge}}(w^*, S) := M - L$.

$$\begin{aligned}
w^\top w^*/|w^*| &\leq |w| \\
(M-L)^2 \leq (w^\top w^*)^2 &\leq |w|^2|w^*|^2 \\
(M-L)^2 &\leq MR^2|w^*|^2 \\
M^2 - 2ML + L^2 &\leq MR^2|w^*|^2 \\
M - 2L + L^2/M &\leq R^2|w^*|^2.
\end{aligned}$$

$$\begin{aligned}
\therefore M &\leq R^2|w^*|^2 + 2L - L^2/M \\
&\leq R^2|w^*|^2 + 2L.
\end{aligned}$$

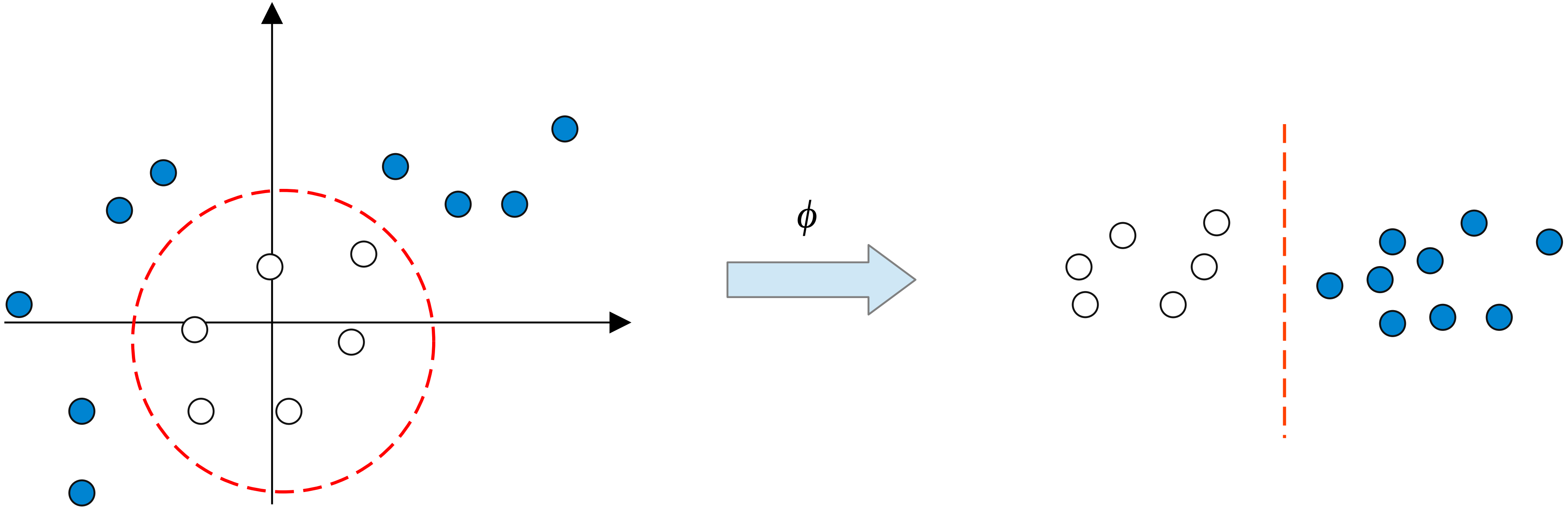
How to generalize a linear separator?

- Solution: **Kernel Functions**.



How to generalize a linear separator?

- Solution: **Kernel Functions**.



Kernel functions

- Replace the dot product in the high dimensional space.
- Over pairs of data points such that for some function $\phi : R^d \mapsto R^N$, we have $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$.

Kernel functions

- Replace the dot product in the high dimensional space.
- Over pairs of data points such that for some function $\phi : R^d \mapsto R^N$, we have $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$.

- For example,

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^k, \text{ for some } k \geq 1.$$

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x}^\top \mathbf{x}')^2 = (1 + x_1 x_1' + x_2 x_2')^2 \\ &= 1 + 2x_1 x_1' + 2x_2 x_2' + x_1^2 {x_1'}^2 + 2x_1 x_2 x_1' x_2' + x_2^2 {x_2'}^2 \\ &= \phi(\mathbf{x})^\top \phi(\mathbf{x}'). \end{aligned}$$

Kernel functions

- Replace the dot product in the high dimensional space.
- Over pairs of data points such that for some function $\phi : R^d \mapsto R^N$,

  we have $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$.

- For example,

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^k, \text{ for some } k \geq 1.$$

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') &= (1 + \mathbf{x}^\top \mathbf{x}')^2 = (1 + x_1 x_1' + x_2 x_2')^2 \\ &= 1 + 2x_1 x_1' + 2x_2 x_2' + x_1^2 {x_1'}^2 + 2x_1 x_2 x_1' x_2' + x_2^2 {x_2'}^2 \\ &= \phi(\mathbf{x})^\top \phi(\mathbf{x}'). \qquad \text{for } \phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2) \end{aligned}$$

Theorem (Kernel functions' legality)

- Suppose K_1 and K_2 are kernel functions. Then,
- For any constant $c \geq 0$, cK_1 is a legal kernel.
- For any scalar function f, $K_3 = f(\mathbf{x})\, f(\mathbf{x}')K_1(\mathbf{x}, \mathbf{x}')$ is a legal kernel.
- The sum $K_1 + K_2$ is a legal kernel.
- The product K_1K_2 is a legal kernel.

Popular kernel functions

- Polynomial

$$K(\mathbf{x}, \mathbf{x}') = (c + \mathbf{x}^\top \mathbf{x}')^k, \text{ for some } k \geq 1 \text{ and constant } c \geq 0.$$

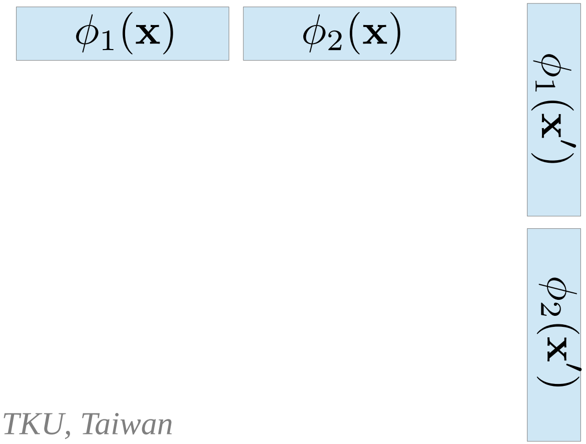
- Gaussian (Radial Basis Function; RBF)

$$K(\mathbf{x}, \mathbf{x}') = e^{-c|\mathbf{x}-\mathbf{x}'|^2}.$$

Theorem (Kernel functions' legality)

- Suppose K_1 and K_2 are kernel functions. Then,
- For any constant $c \geq 0$, cK_1 is a legal kernel.
- For any scalar function f, $K_3 = f(\mathbf{x})\, f(\mathbf{x}')K_1(\mathbf{x}, \mathbf{x}')$ is a legal kernel.
- The sum $K_1 + K_2$ is a legal kernel.
- The product K_1K_2 is a legal kernel.

$$K_1 + K_2 = \phi_1(\mathbf{x})^\top \phi_1(\mathbf{x}') + \phi_2(\mathbf{x})^\top \phi_2(\mathbf{x}')$$



Popular kernel functions

- Polynomial

$$K(\mathbf{x},\mathbf{x}') = (c + \mathbf{x}^\top \mathbf{x}')^k, \ \text{for some } k \geq 1 \text{ and constant } c \geq 0.$$

- Gaussian (Radial Basis Function; RBF)

$$K(\mathbf{x},\mathbf{x}') = e^{-c|\mathbf{x}-\mathbf{x}'|^2}.$$

Why are they legal?

Popular kernel functions

- Polynomial

$$K(\mathbf{x}, \mathbf{x}') = (c + \mathbf{x}^\top \mathbf{x}')^k, \text{ for some } k \geq 1 \text{ and constant } c \geq 0.$$

- Gaussian (Radial Basis Function; RBF)

$$K(\mathbf{x}, \mathbf{x}') = e^{-c|\mathbf{x}-\mathbf{x}'|^2}.$$

Why are they legal?

$$\text{Gaussion: } f(\mathbf{x})f(\mathbf{x}')e^{2c\mathbf{x}^\top \mathbf{x}'} \text{ for } f(\mathbf{x}) = e^{-c|\mathbf{x}|^2}.$$