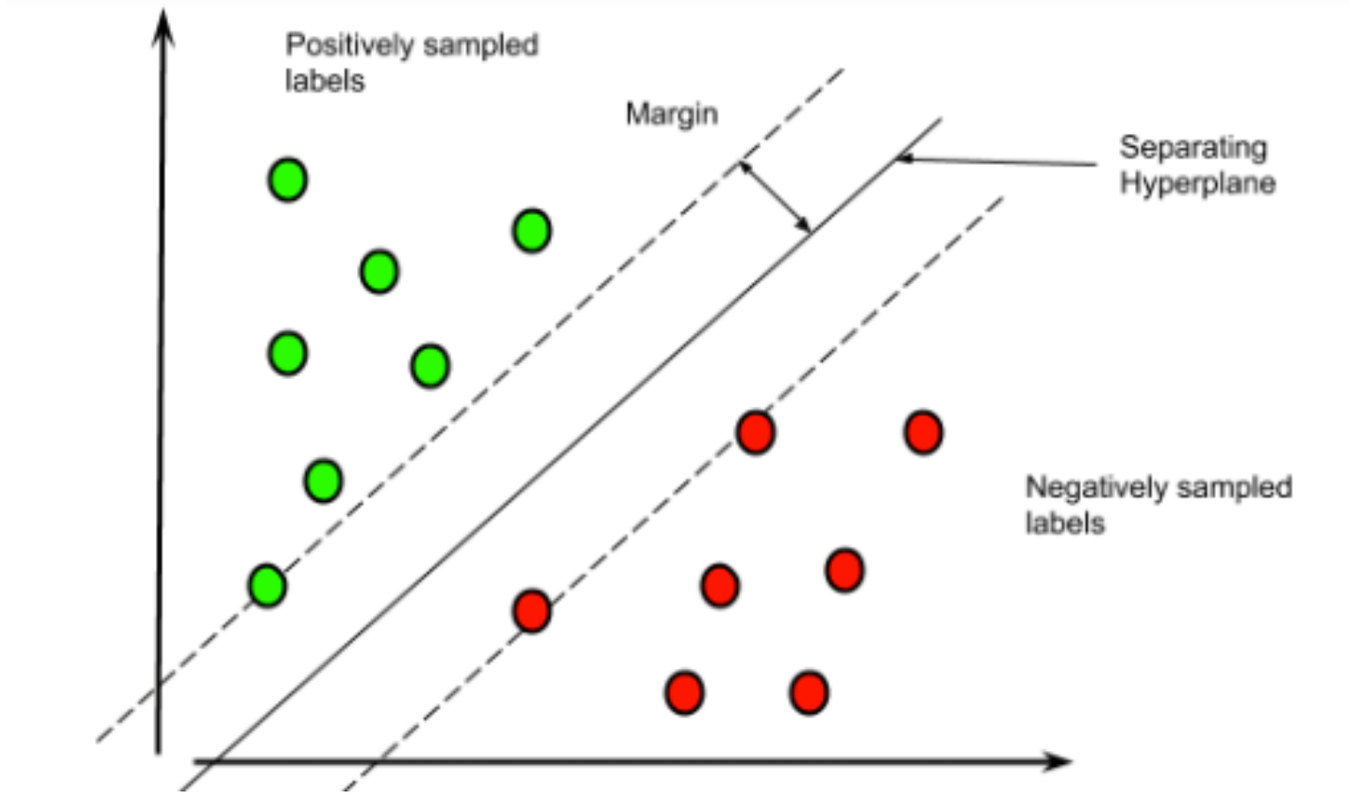


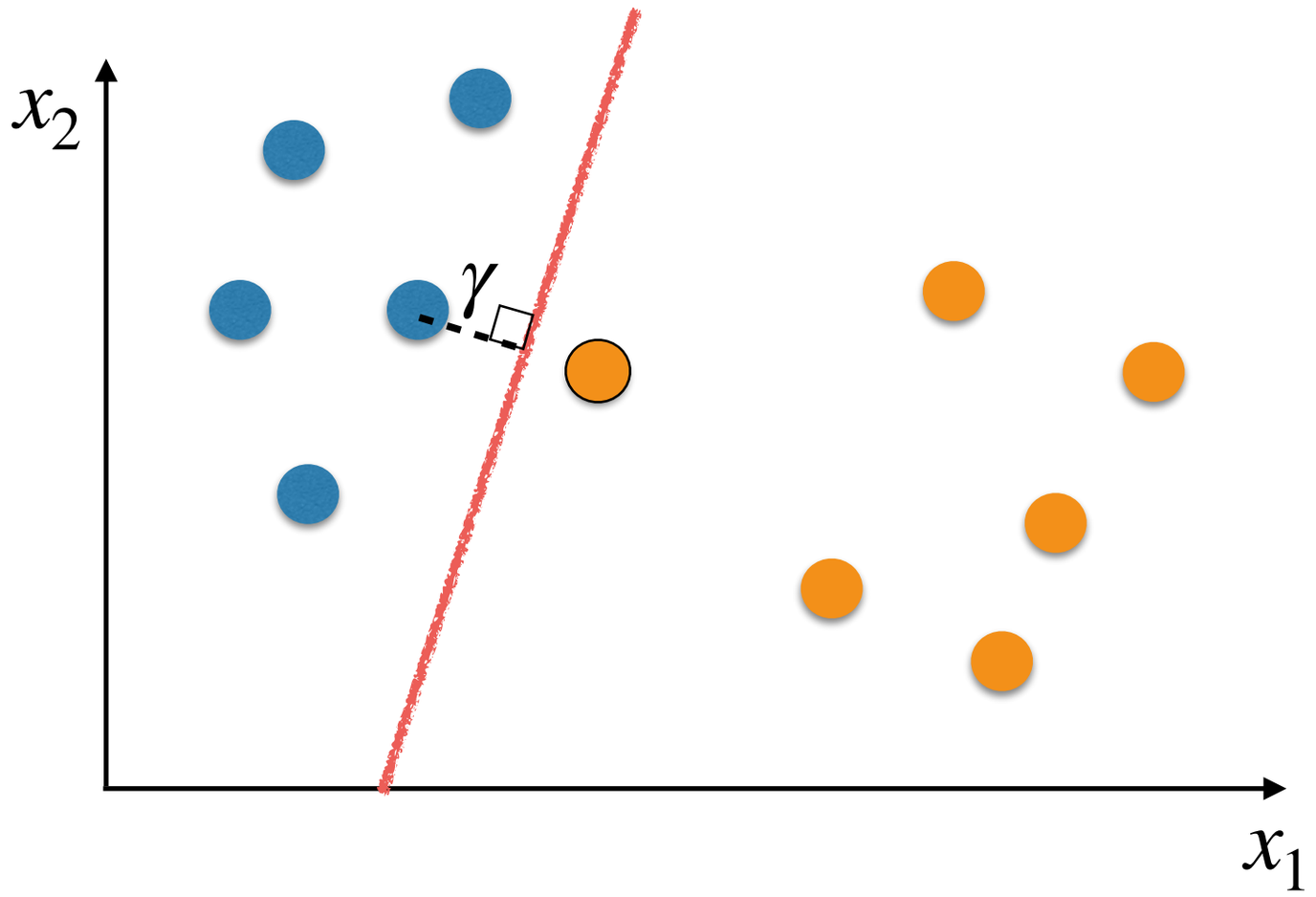
Support Vector Machines (SVMs): Maximum Margin Classifiers

Leandro L. Minku

Overview

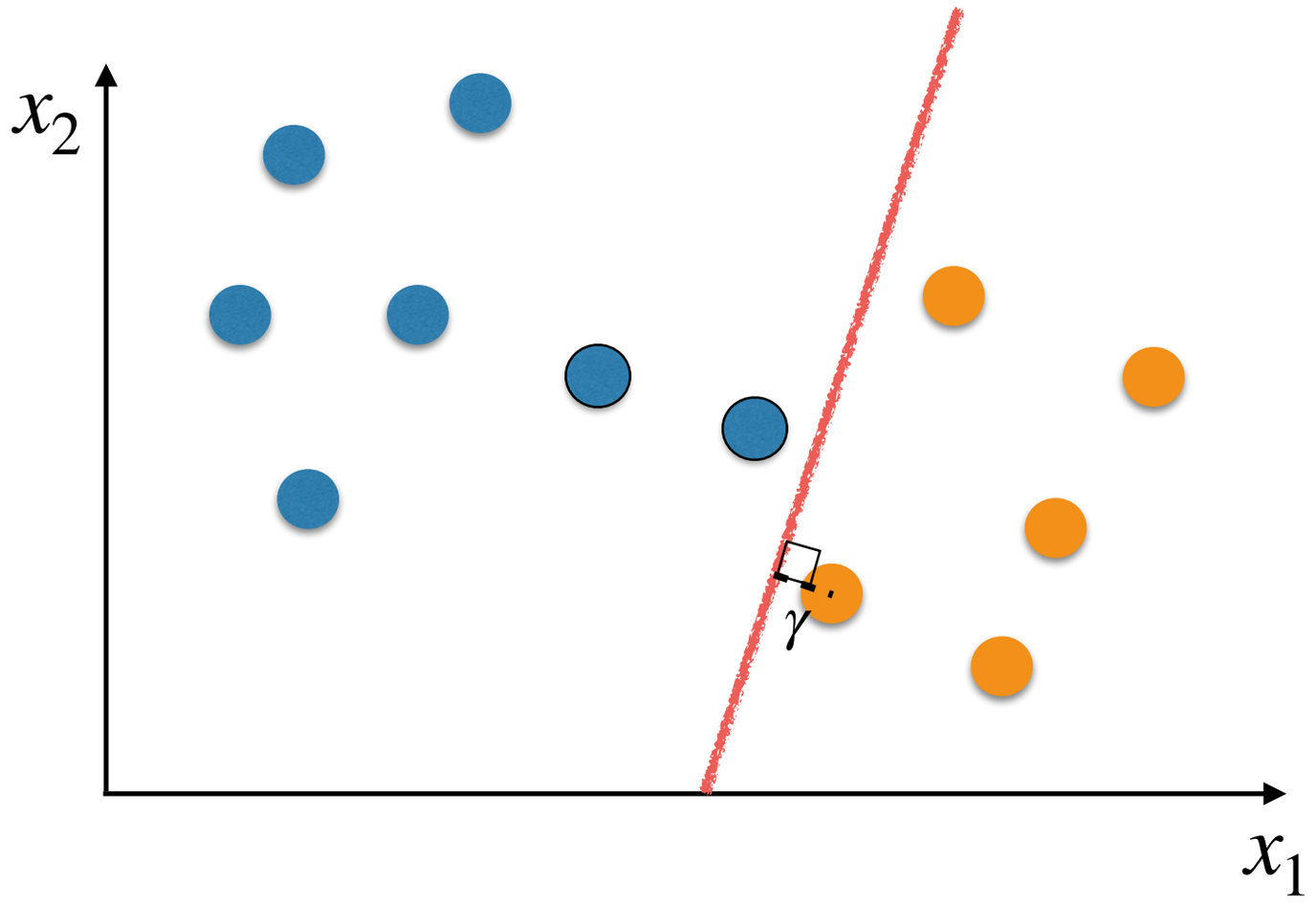
- Support vector machines
  - Hypothesis Set
  - Optimisation Problem

General Idea



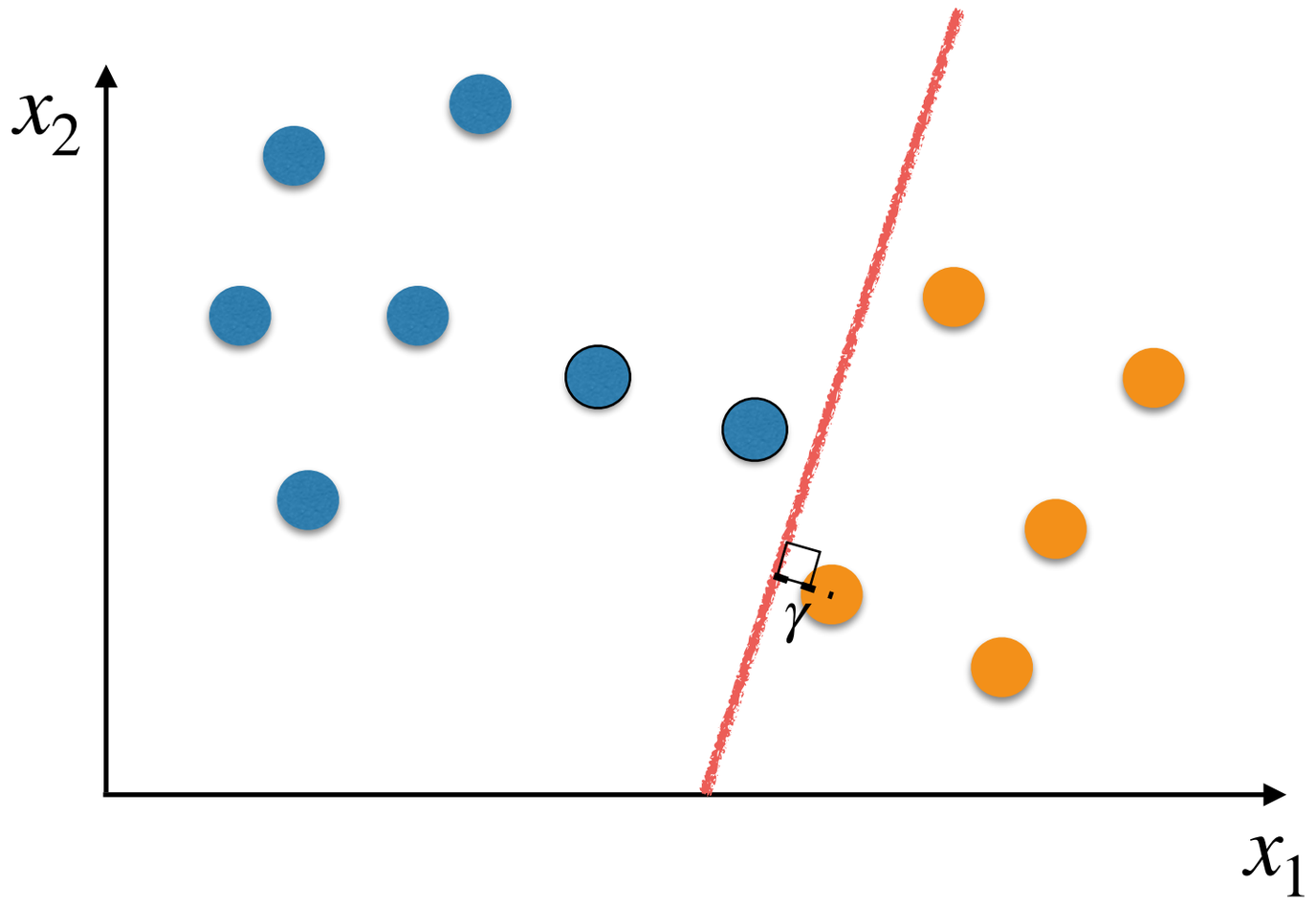
The perpendicular distance γ between the decision boundary and the closest of the training examples on the left is too small.

General Idea



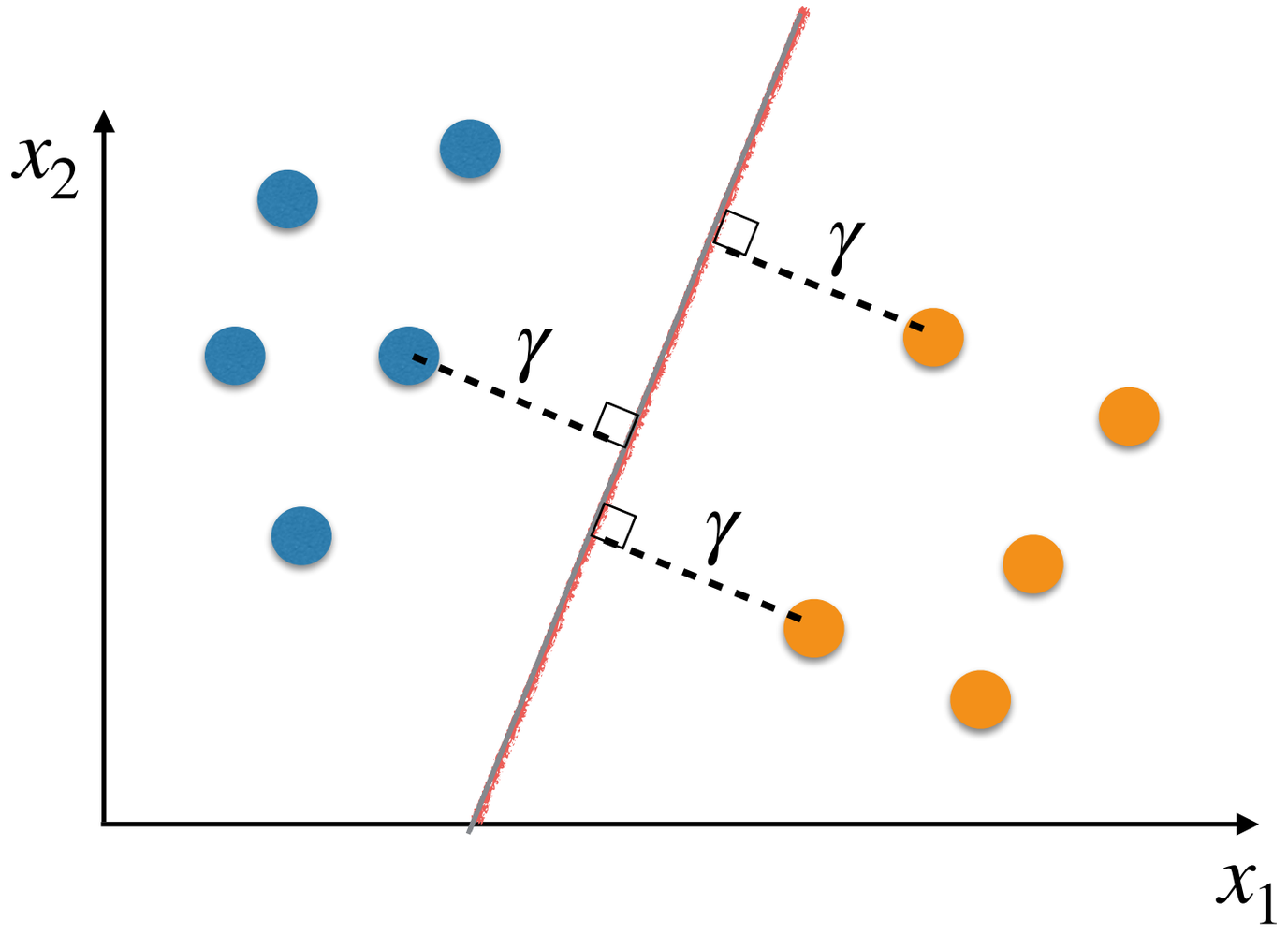
The perpendicular distance γ between the decision boundary and the closest of the training examples on the right is too small.

Margin



The perpendicular distance γ between the decision boundary and the closest training example is called the **margin**.

Where Would Be The Best Place To Put The Decision Boundary?



The decision boundary can be chosen so as to maximise the margin, which can help to avoid overfitting.

Training examples that are exactly on the margin are called **support vectors**.

Supervised Learning Problem

- Given a set of training examples

$$\mathscr{T} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

where $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathscr{X} \times \mathscr{Y}$ are drawn i.i.d. (independently and identically distributed) from a fixed albeit unknown joint probability distribution $P(\mathbf{x}, y) = P(y \,|\, \mathbf{x})P(\mathbf{x})$.

- Goal: to learn a function g able to generalise to unseen (test) examples of the same probability distribution $P(\mathbf{x}, y)$.
  - $g : \mathscr{X} \rightarrow \mathscr{Y}$, mapping input space to output space.
  - g as a probability distribution approximating $P(y \,|\, \mathbf{x})$.

Supervised Learning Problem

- Given a set of training examples

$$\mathcal{T} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

where $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ are drawn i.i.d. (independently and identically distributed) from a fixed albeit unknown joint probability distribution $P(\mathbf{x}, y) = P(y \,|\, \mathbf{x})P(\mathbf{x})$.

- Goal: to learn a function g able to generalise to unseen (test) examples of the same probability distribution $P(\mathbf{x}, y)$.
  - $g : \mathbb{R}^d \rightarrow \{-1, +1\}$, mapping input space to output space.
  - g as a probability distribution approximating $P(y \,|\, \mathbf{x})$.

Linear Classifier

$$w_1 x_1 + w_2 x_2 + w_0 = 0 \quad \text{Equation of a line}$$

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 = 0 \quad \text{Equation of a plane}$$

$$w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + w_0 = 0 \quad \text{Equation of a hyperplane}$$

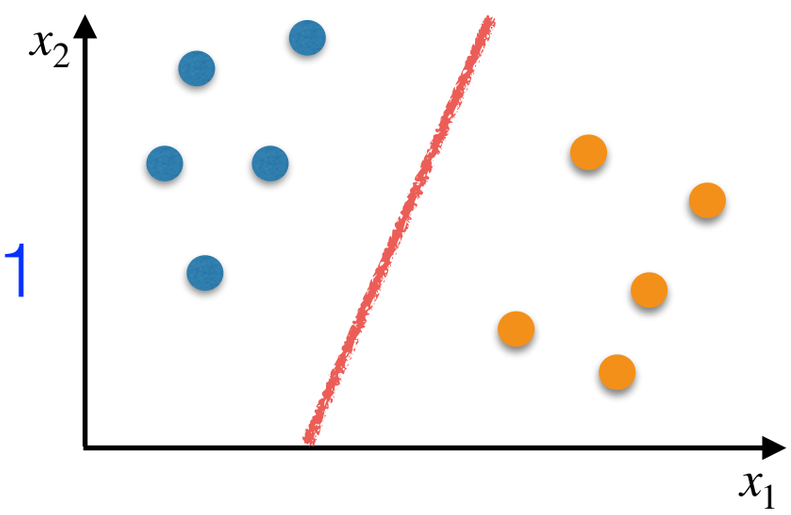
$$\mathbf{w}^T \mathbf{x} + w_0$$

$$\mathbf{w}^T \mathbf{x} + b$$

bias

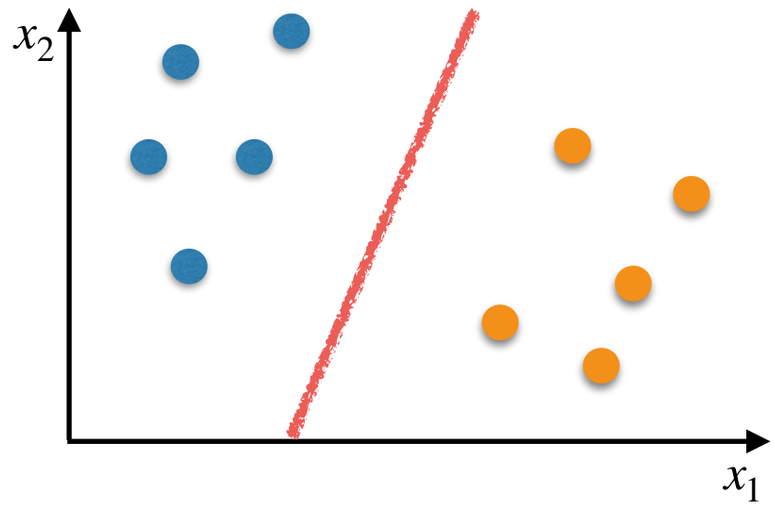
$\mathbf{w}^T \mathbf{x} + b > 0 \rightarrow$ class +1

$\mathbf{w}^T \mathbf{x} + b < 0 \rightarrow$ class -1



Hypothesis Set

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x} + b > 0 \\ -1 & \text{if } \mathbf{w}^T\mathbf{x} + b < 0 \end{cases}, \quad \forall \mathbf{w} \in \mathbb{R}^d, \forall b \in \mathbb{R}$$

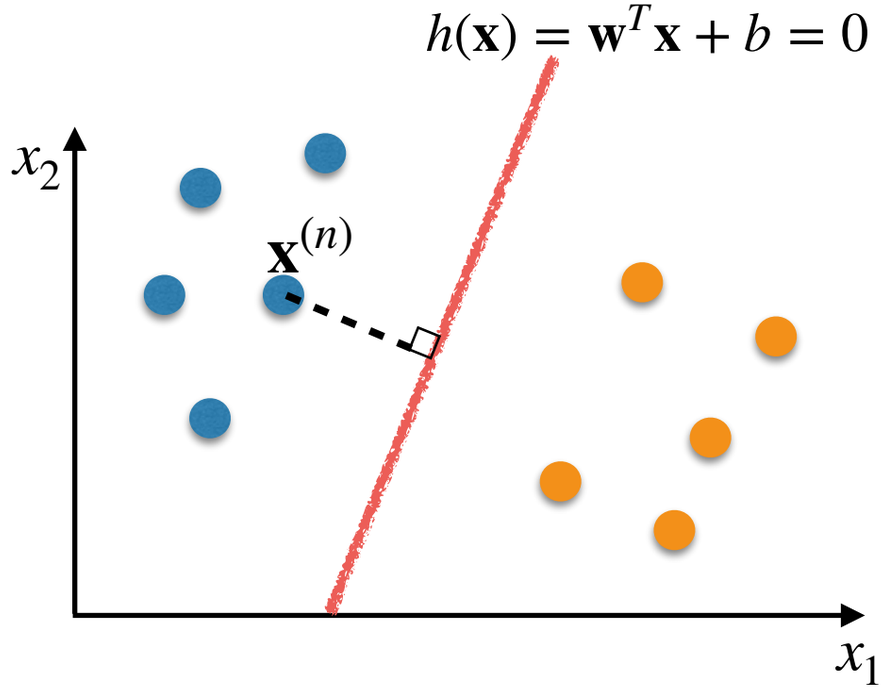


For simplicity, we will use the notation $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$.

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



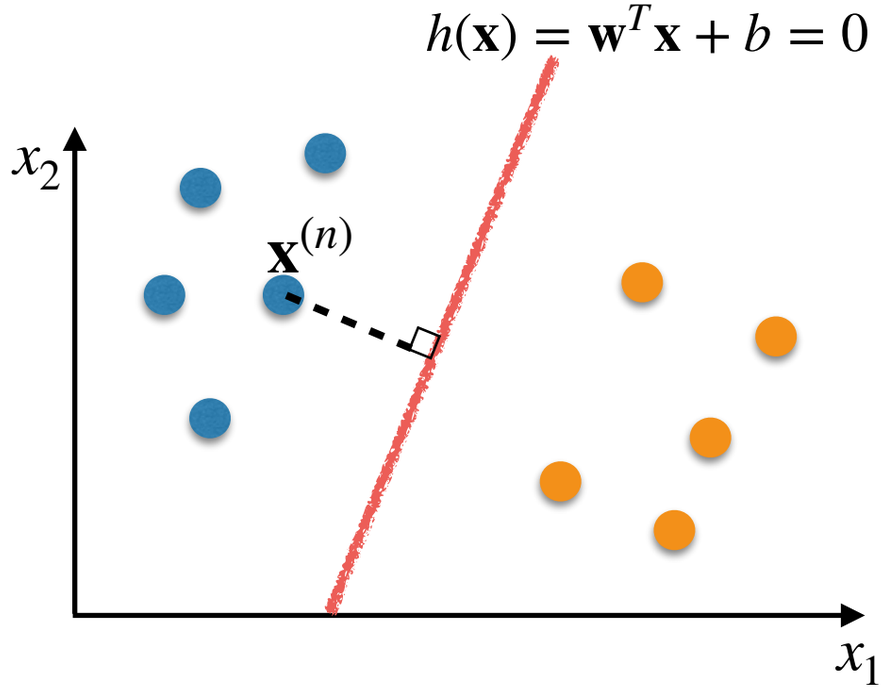
Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

$$\min_n \text{dist}(h, \mathbf{x}^{(n)})$$

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



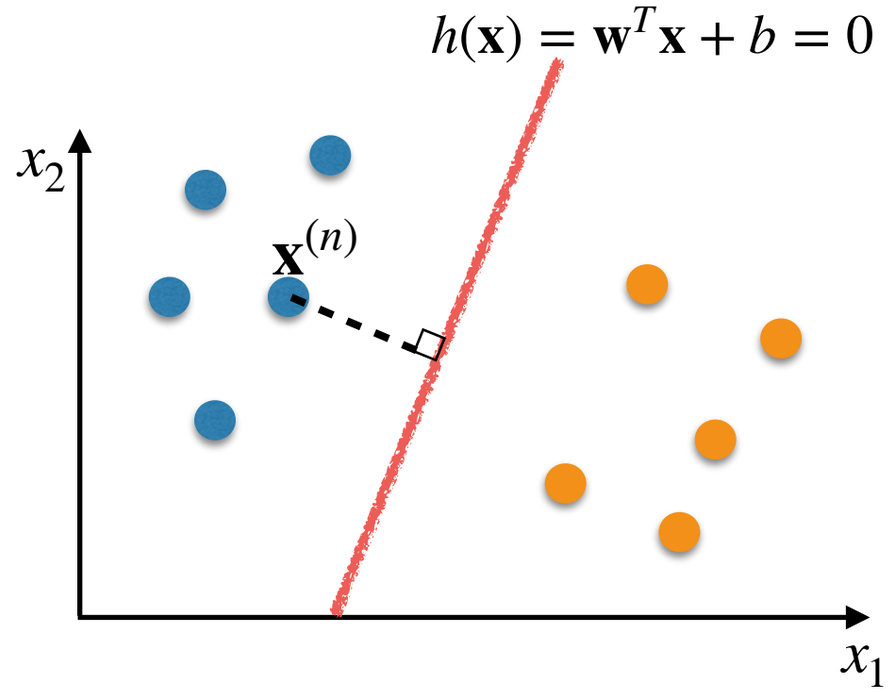
Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \min_{n} \text{dist}(h, \mathbf{x}^{(n)}) \right\}$$

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

Constraint: all training examples must be correctly classified.

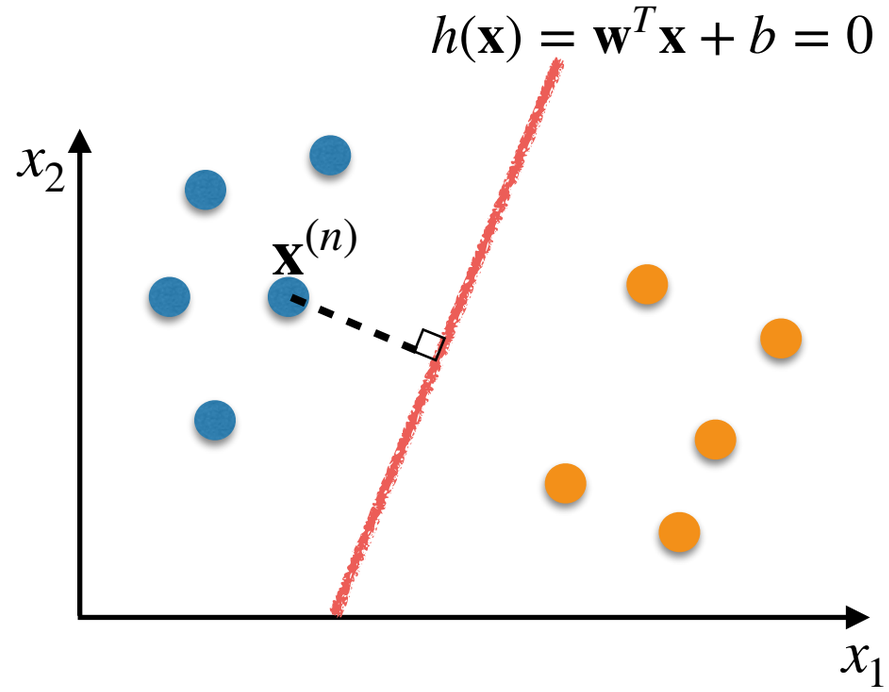
Subject to $y^{(n)}h(\mathbf{x}^{(n)}) > 0,\ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

$$y^{(n)} = +1 \ \ h(\mathbf{x}^{(n)}) > 0$$

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

Constraint: all training examples must be correctly classified.

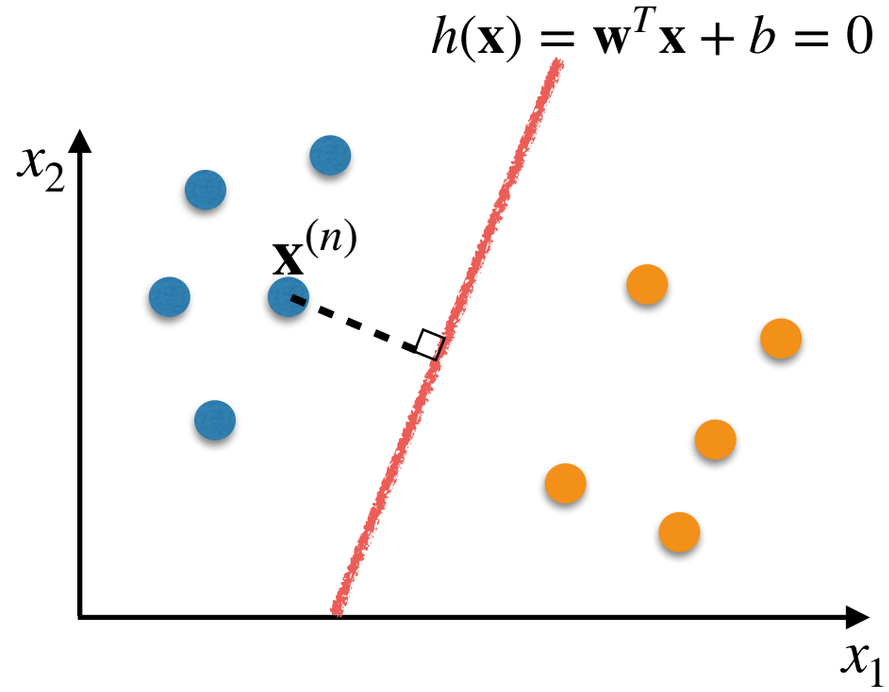
Subject to $y^{(n)}h(\mathbf{x}^{(n)}) > 0,\ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

$$y^{(n)} = -1 \quad h(\mathbf{x}^{(n)}) < 0$$

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

Constraint: all training examples must be correctly classified.

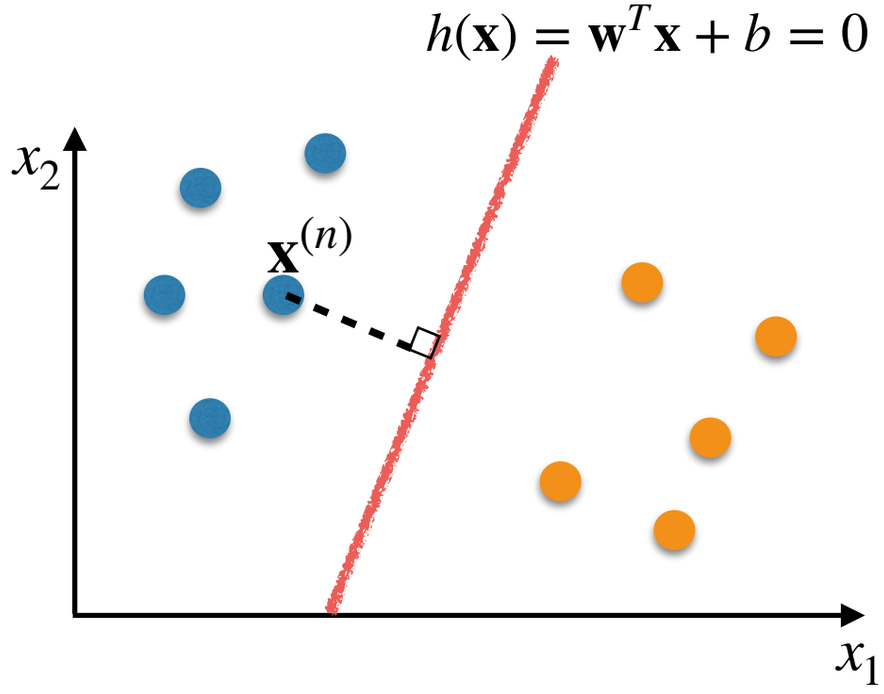
Subject to $y^{(n)}h(\mathbf{x}^{(n)}) > 0,\ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

$$y^{(n)} = +1 \qquad h(\mathbf{x}^{(n)}) < 0$$

Perpendicular Distance From a Point $\mathbf{x}^{(n)}$ to a Hyperplane $h(\mathbf{x}) = 0$

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm
(the length of the vector $\mathbf{w}$)



Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

Constraint: all training examples must be correctly classified.

Subject to $y^{(n)}h(\mathbf{x}^{(n)}) > 0,\ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

$$y^{(n)} = -1 \quad h(\mathbf{x}^{(n)}) > 0$$

Optimisation Problem

Find $\mathbf{w}$ and b that maximise the margin, i.e., the perpendicular distance between the hyperplane and the closest training example.

$$\underset{\mathbf{w},b}{\operatorname{argmax}} \left\{ \min_{n} \operatorname{dist}(h, \mathbf{x}^{(n)}) \right\}$$

Constraint: all training examples must be correctly classified.

Subject to $y^{(n)} h(\mathbf{x}^{(n)}) > 0, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \min_{n} \text{dist}(h, \mathbf{x}^{(n)}) \right\}$$

Subject to $y^{(n)} h(\mathbf{x}^{(n)}) > 0, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

$$\text{dist}(h, \mathbf{x}^{(n)}) = \frac{|\, h(\mathbf{x}^{(n)}) \,|}{\|\mathbf{w}\|} = \frac{y^{(n)} h(\mathbf{x}^{(n)})}{\|\mathbf{w}\|}$$

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \min_{n} \left(\frac{y^{(n)} h(\mathbf{x}^{(n)})}{\|\mathbf{w}\|} \right) \right\}$$

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{n} \left(y^{(n)} h(\mathbf{x}^{(n)}) \right) \right\}$$

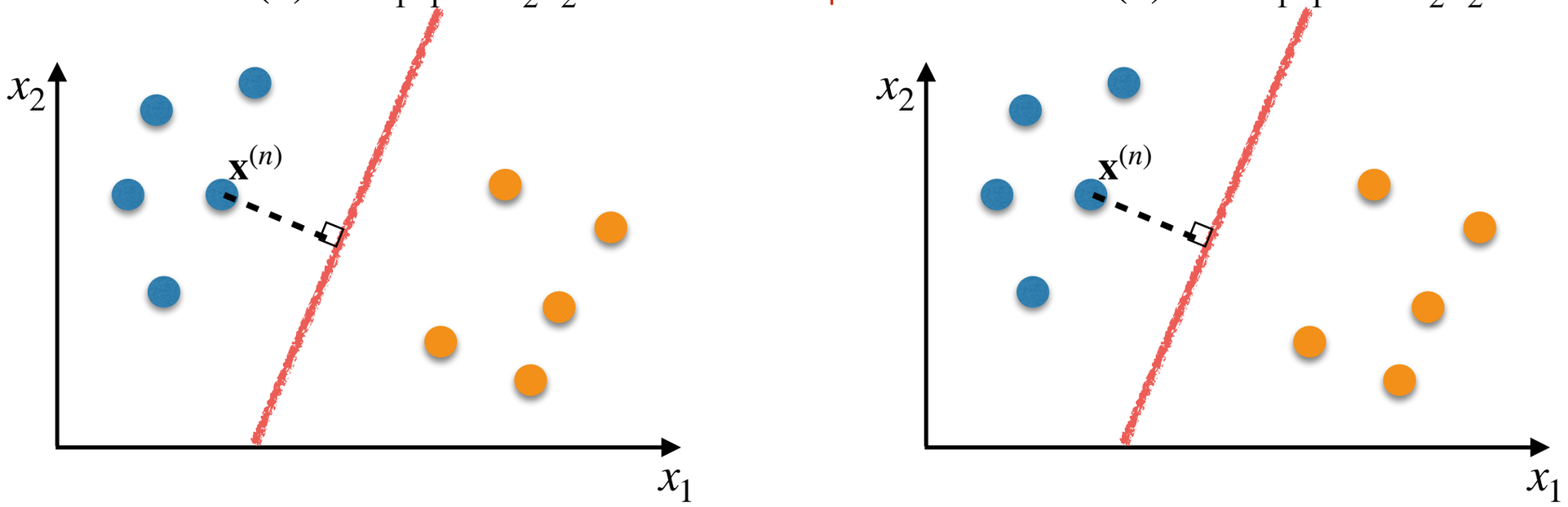
Subject to $y^{(n)} h(\mathbf{x}^{(n)}) > 0, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Subject to $\min_{n} y^{(n)} h(\mathbf{x}^{(n)}) = 1, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Why Are These Two Constraints Equivalent?

Rescaling $\mathbf{w}$ and b does not change the position of the hyperplane, nor the distances of the training examples to it.

$h(\mathbf{x}) = w_1 x_1 + w_2 x_2 + b = 0$ equivalent to $h(\mathbf{x}) = \kappa w_1 x_1 + \kappa w_2 x_2 + \kappa b = 0$



Given a hyperplane that can separate the training examples, this is equivalent to a hyperplane that rescales $\mathbf{w}$ and b by dividing them by $\min_n \; y^{(n)} h(\mathbf{x}^{(n)})$, such that $y^{(n)} h(\mathbf{x}^{(n)}) = 1$ for the closest example.

So, the optimal solution satisfying $y^{(n)} h(\mathbf{x}^{(n)}) > 0$ is equivalent to the optimal solution satisfying $\min_n y^{(n)} h(\mathbf{x}^{(n)}) = 1$.

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{n} \left(y^{(n)} h(\mathbf{x}^{(n)}) \right) \right\}$$

Subject to $\min_{n} y^{(n)} h(\mathbf{x}^{(n)}) = 1, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \underbrace{\min_{n} \left(y^{(n)} h(\mathbf{x}^{(n)}) \right)}_{=1} \right\}$$

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \right\}$$

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \|\mathbf{w}\| \right\}$$

Subject to $\min_{n} y^{(n)} h(\mathbf{x}^{(n)}) = 1$, $\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \|\mathbf{w}\| \right\}$$

Subject to $\min_n y^{(n)} h(\mathbf{x}^{(n)}) = 1, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$ stricter

Subject to $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1, \ \forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$ looser

The optimal solution will satisfy the equality in $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1$ for at least one training example, i.e., these two constraints are equivalent for the optimal solution.

We are trying to minimise $\|\mathbf{w}\|$ and the smallest possible value for $\|\mathbf{w}\|$ will happen when the constraint $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1$ is 1 for a training example, as $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$.

Rewriting the Problem in a Simpler Format Easier To Solve

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \|\mathbf{w}\| \right\}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm (the length of the vector $\mathbf{w}$)

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to $y^{(n)}h(\mathbf{x}^{(n)}) \geq 1,\ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Quadratic Programming Problem

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

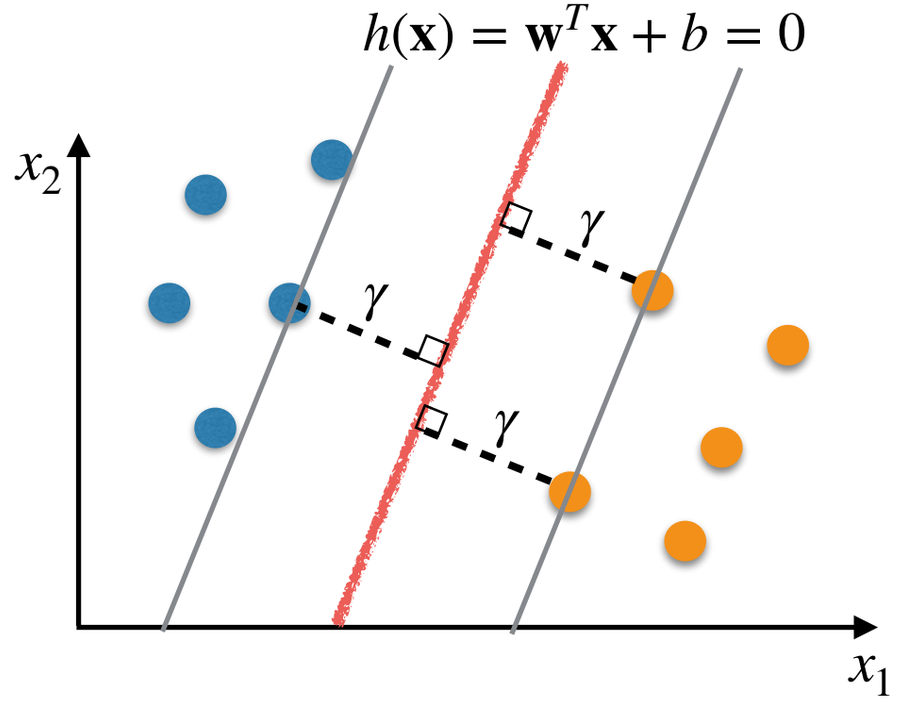
Subject to $y^{(n)}h(\mathbf{x}^{(n)}) \geq 1$ for $\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

Maximum Margin Classifier

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to $y^{(n)}h(\mathbf{x}^{(n)}) \geq 1$

for $\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.

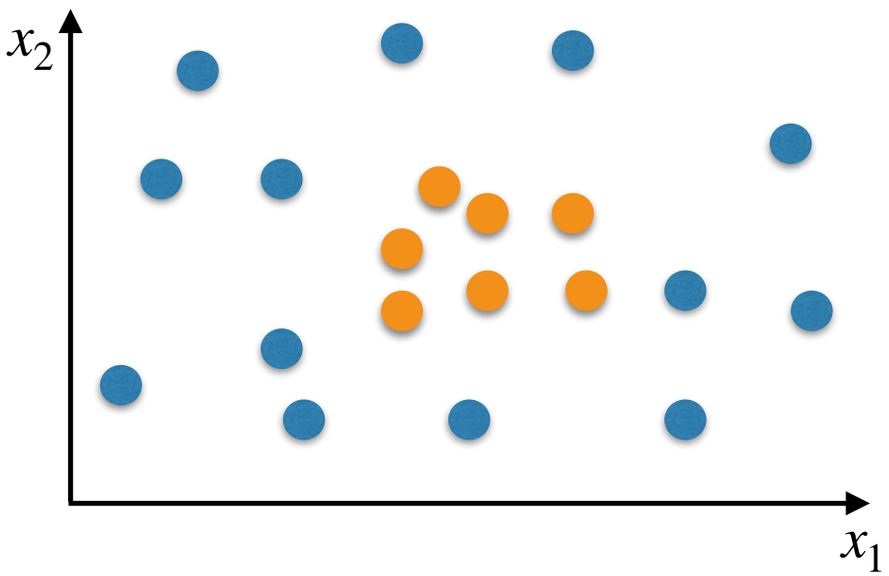


Maximum Margin Classifier for Nonlinear Problems

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to $y^{(n)}h(\mathbf{x}^{(n)}) \geq 1$

for $\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$.



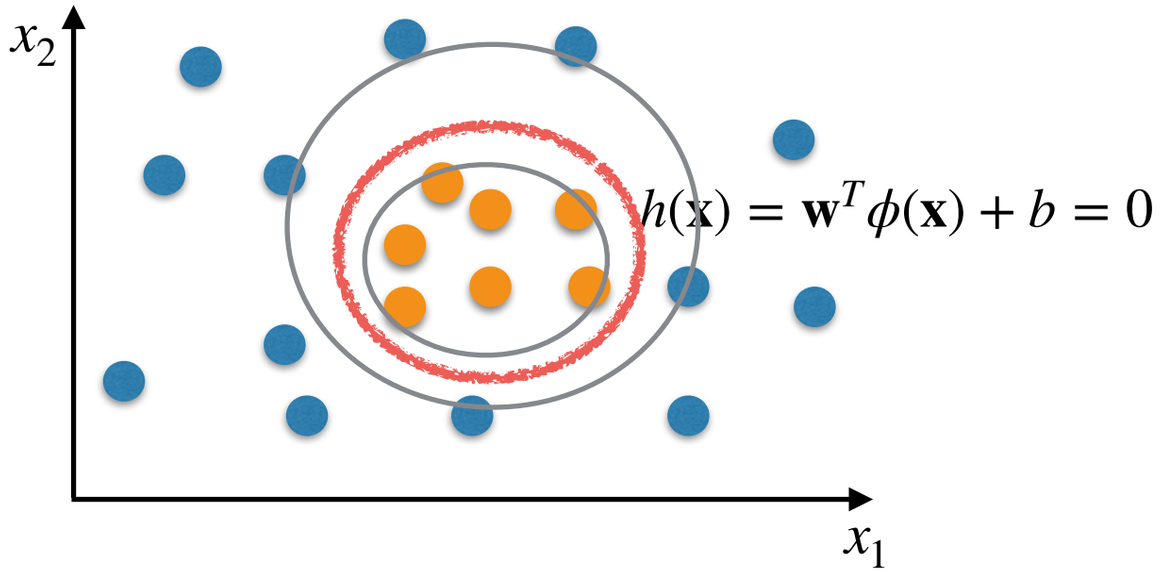
Maximum Margin Classifier for Nonlinear Problems

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to

$$y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) \geq 1,$$

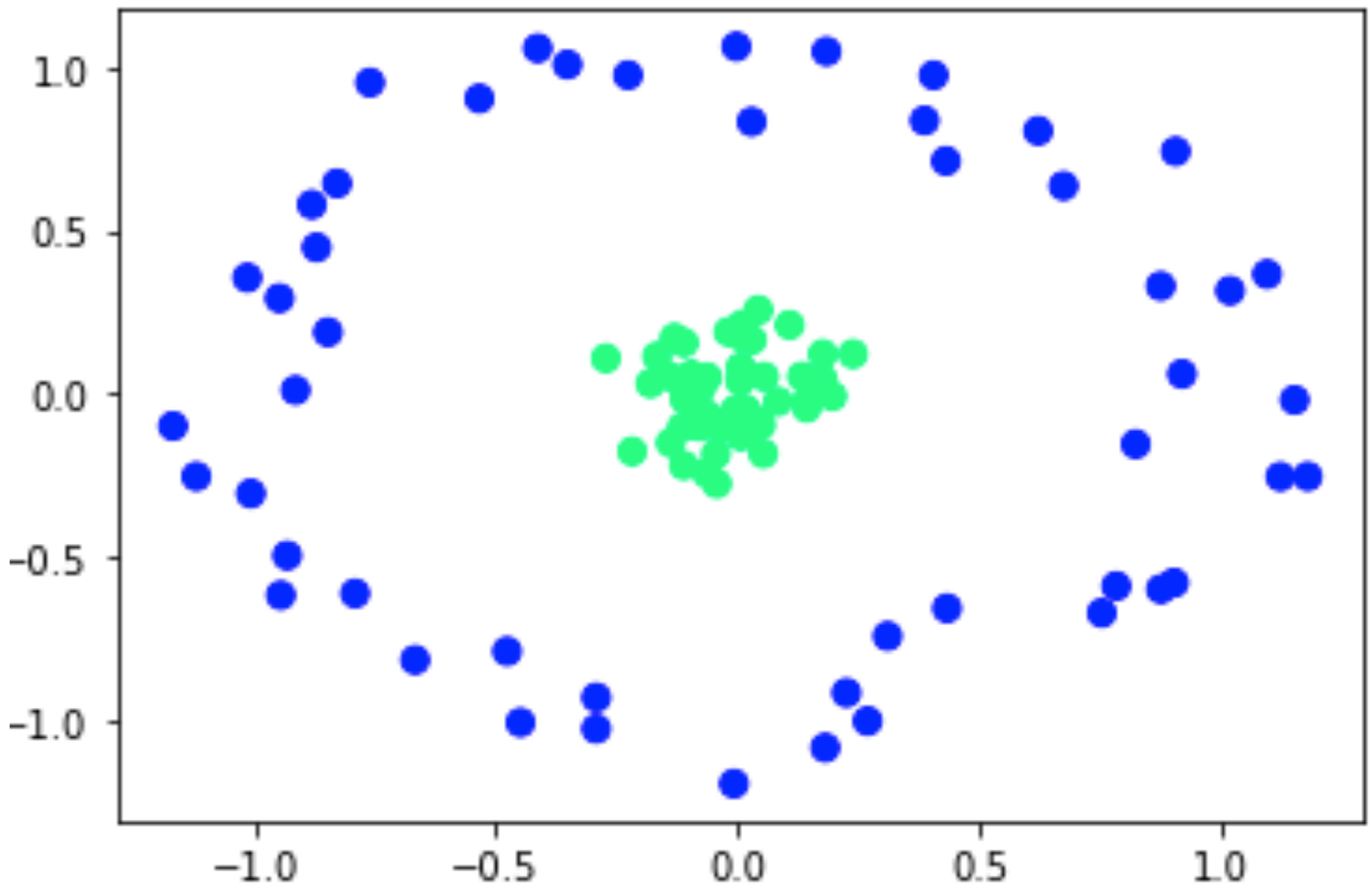
$$\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$$



It is possible to use $\phi(\mathbf{x}) = \mathbf{x}$ if we wish.

We refer to SVM as a linear SVM when using $\phi(\mathbf{x}) = \mathbf{x}$.

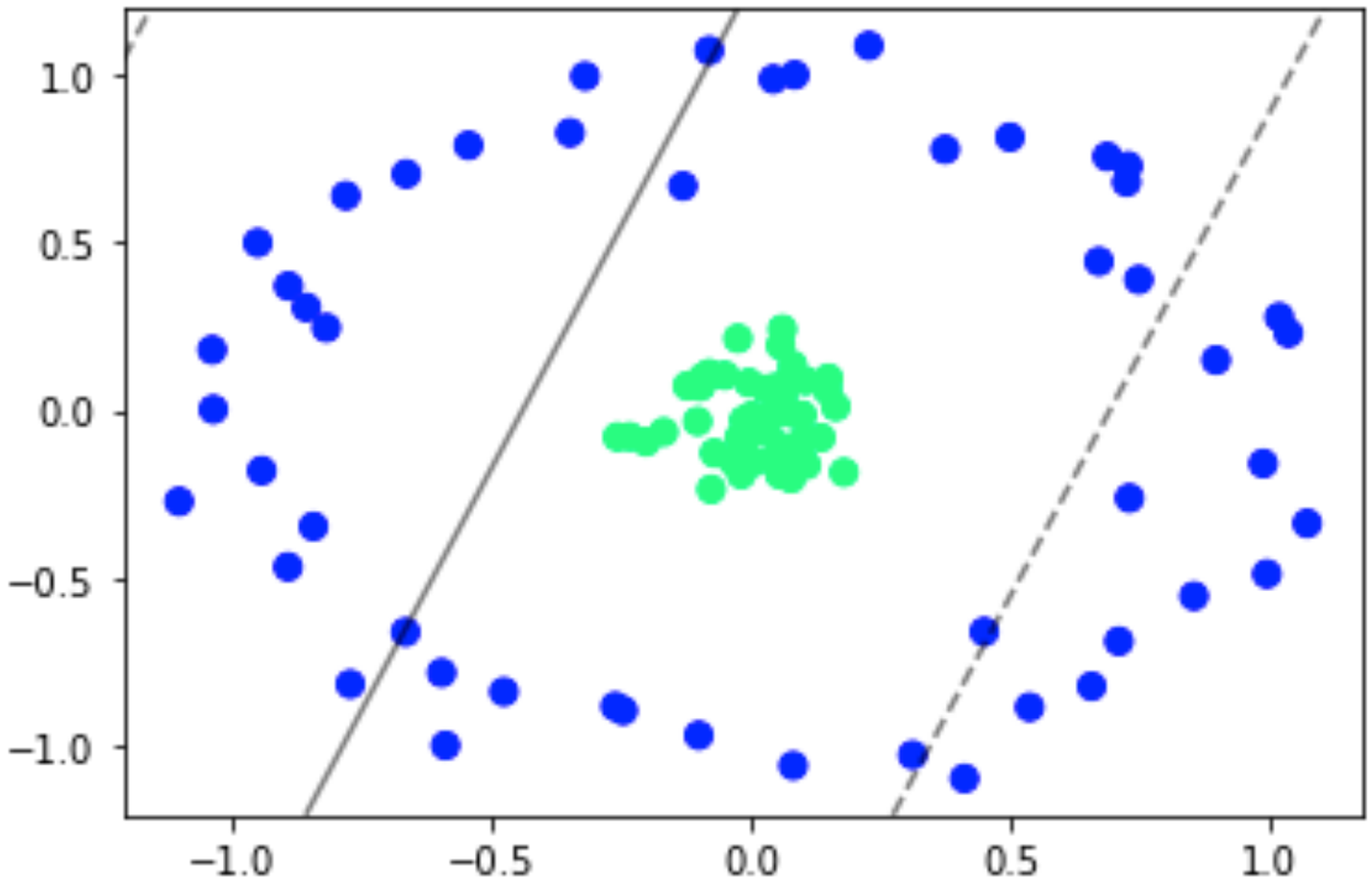
Example



Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

Example

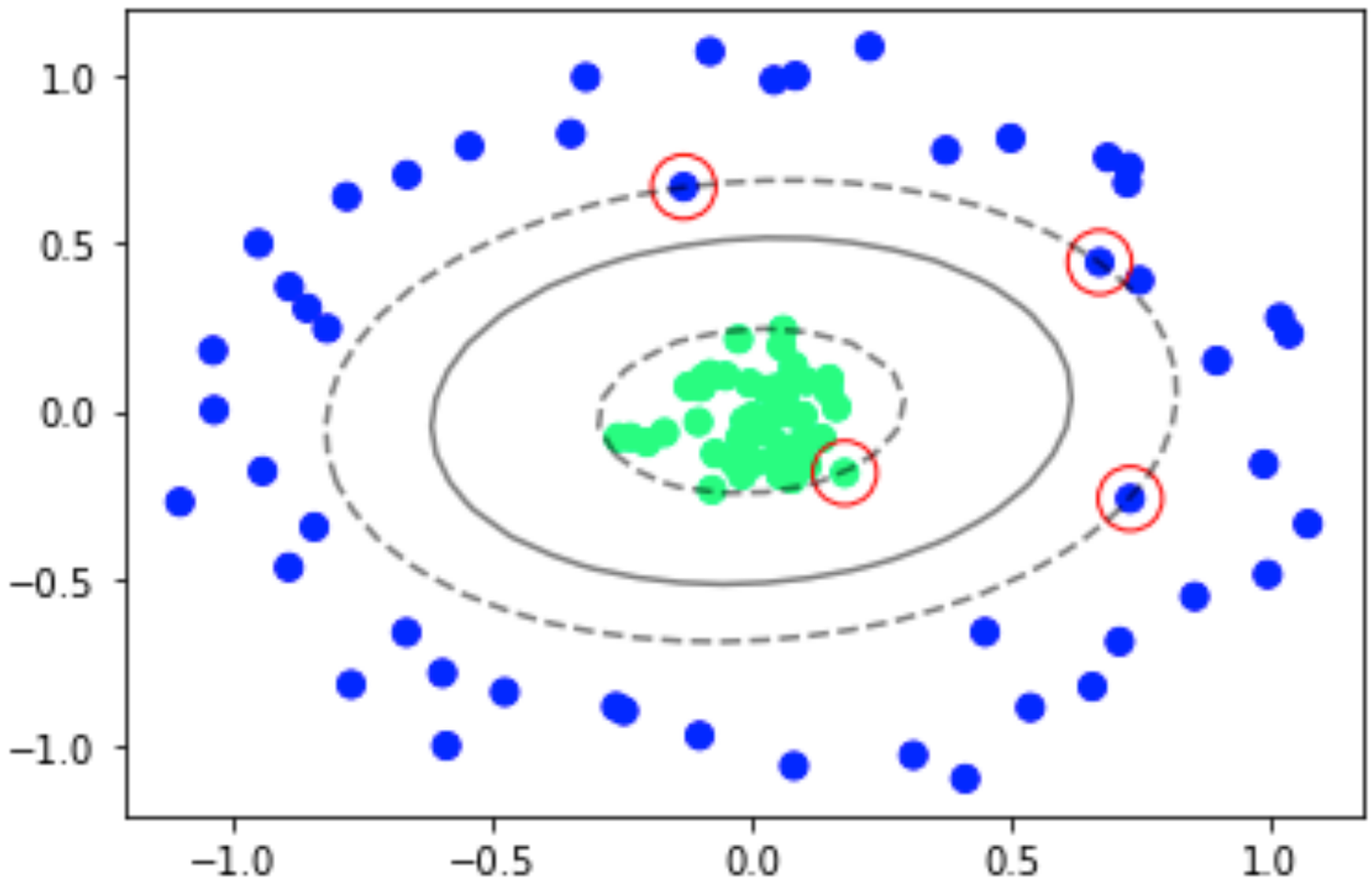
Using $\phi(\mathbf{x}) = \mathbf{x}$



Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

Example

Using polynomial embedding of degree 2



Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* Red circles represent the support vectors

Summary

- Support vector machines are maximum margin classifiers.
- The problem of maximising the margin can be converted into a quadratic programming problem for being solved efficiently.
- Support vector machines are linear classifiers.
- Nonlinear transformations can be used to enable support vector machines to separate non-linear problems.