# CZ4041/CE4041: Machine Learning

## Lesson 8a: Support Vector Machines

Kelly KE

School of Computer Science and Engineering, NTU, Singapore

Acknowledgements: slides are adapted from the lecture notes of the books "Introduction to Machine Learning" (Chap. 13) and "Introduction to Data Mining" (Chap. 5). Slides are modified from the version prepared by Dr. Sinno Pan.

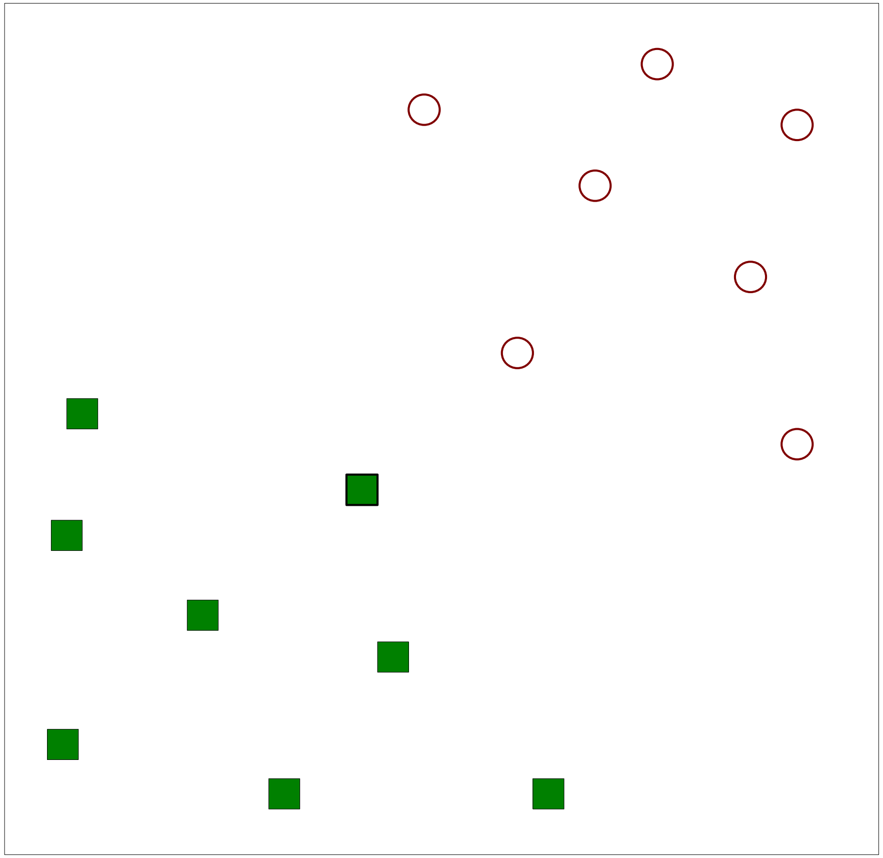
# Support Vector Machines (SVMs)

- SVMs have shown promising empirical results in many practical applications, such as computer vision, sensor networks and text mining
- The motivation behind SVMs is from the geometry perspective of linear algebra
- The objective of SVMs is to learn a maximum margin hyperplane
  - Based on statistical learning theory

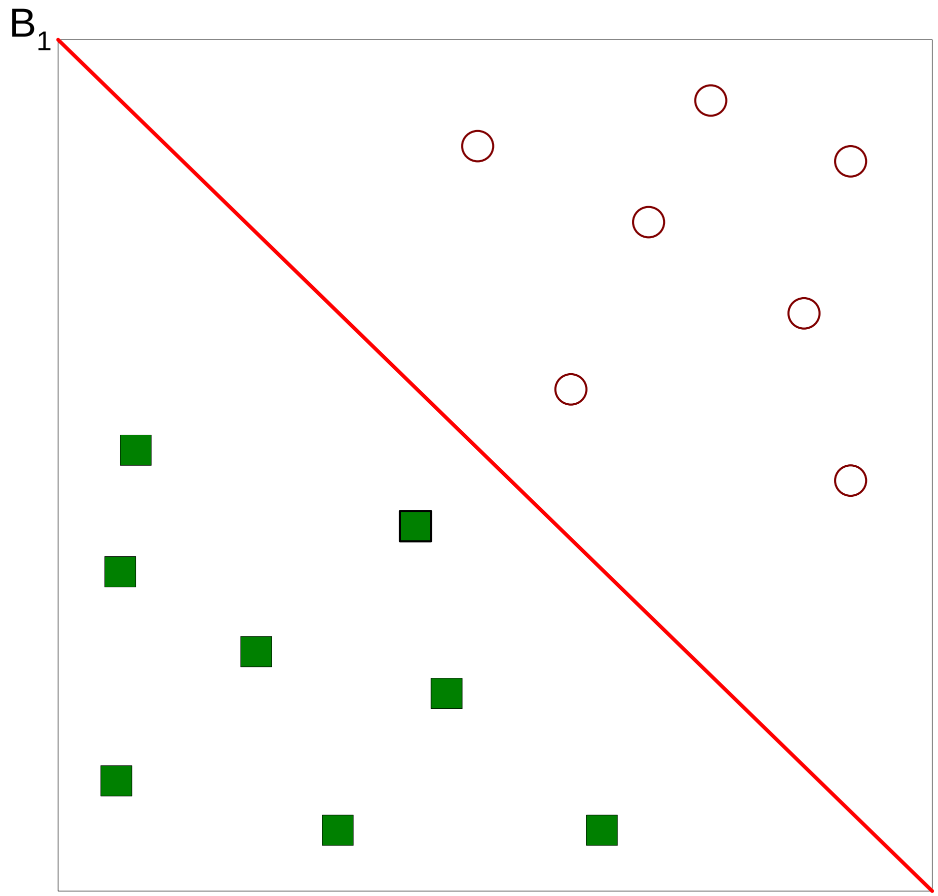
# Maximum Margin

- To learn a binary classifier

- To find a hyperplane (linear decision boundary) so that all the squares reside on one side of the hyperplane and all the circles reside on the other

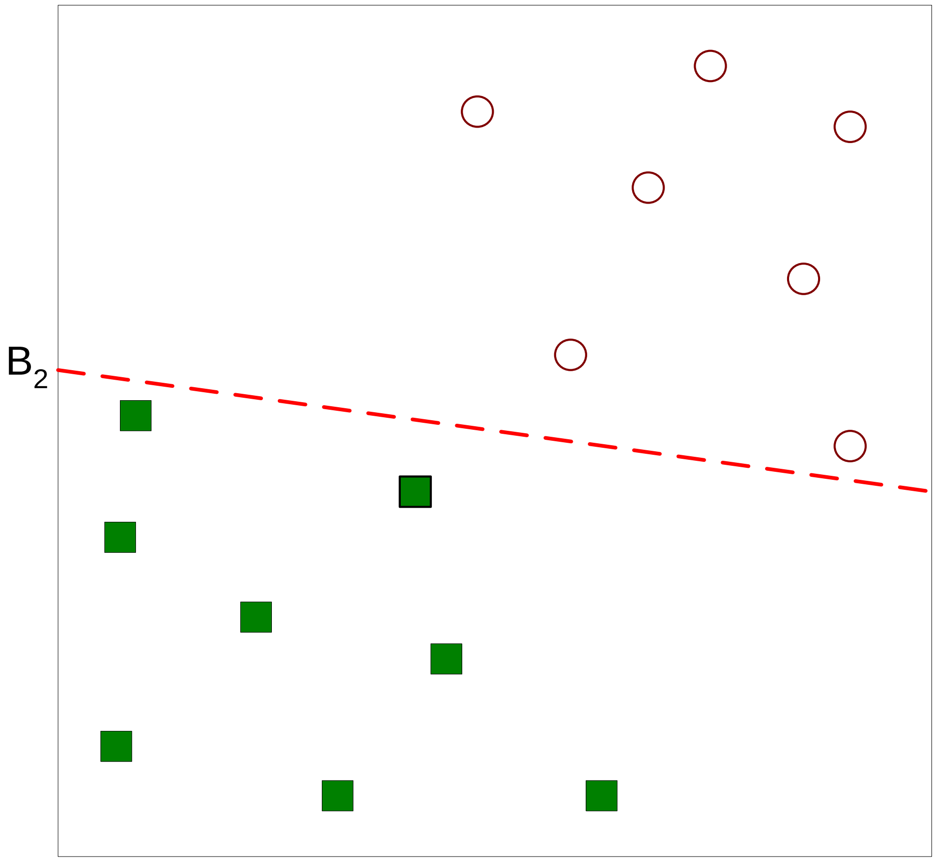


# Maximum Margin (cont.)




One Possible Solution

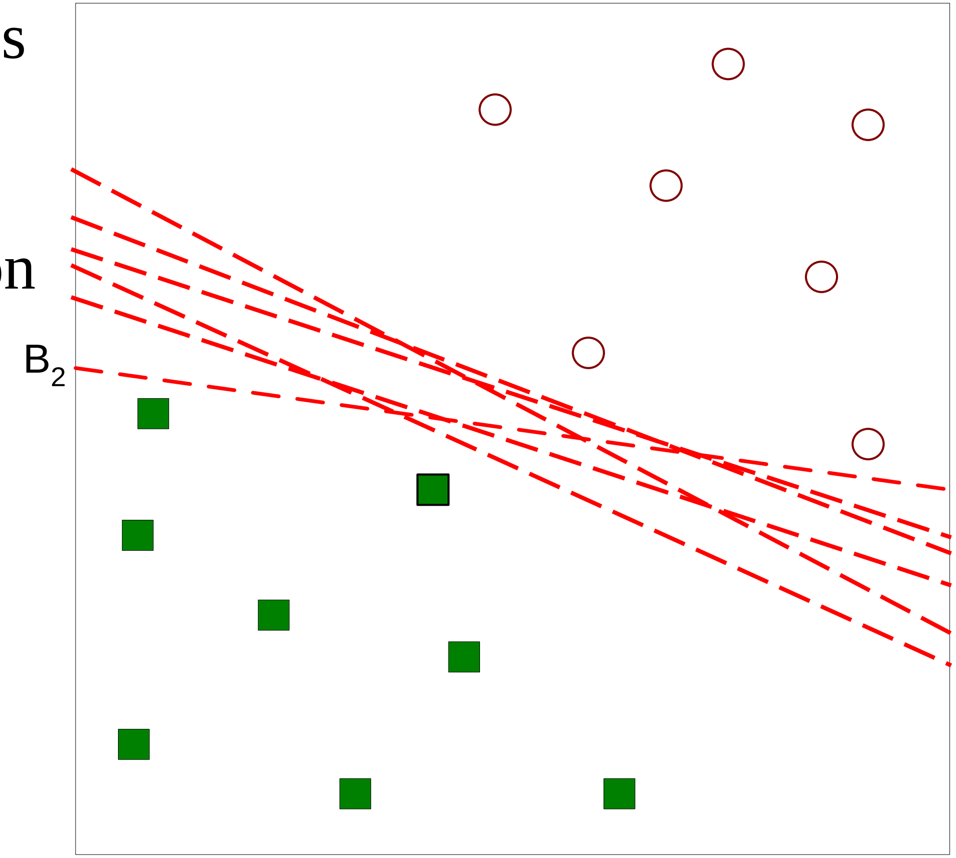
# Maximum Margin (cont.)



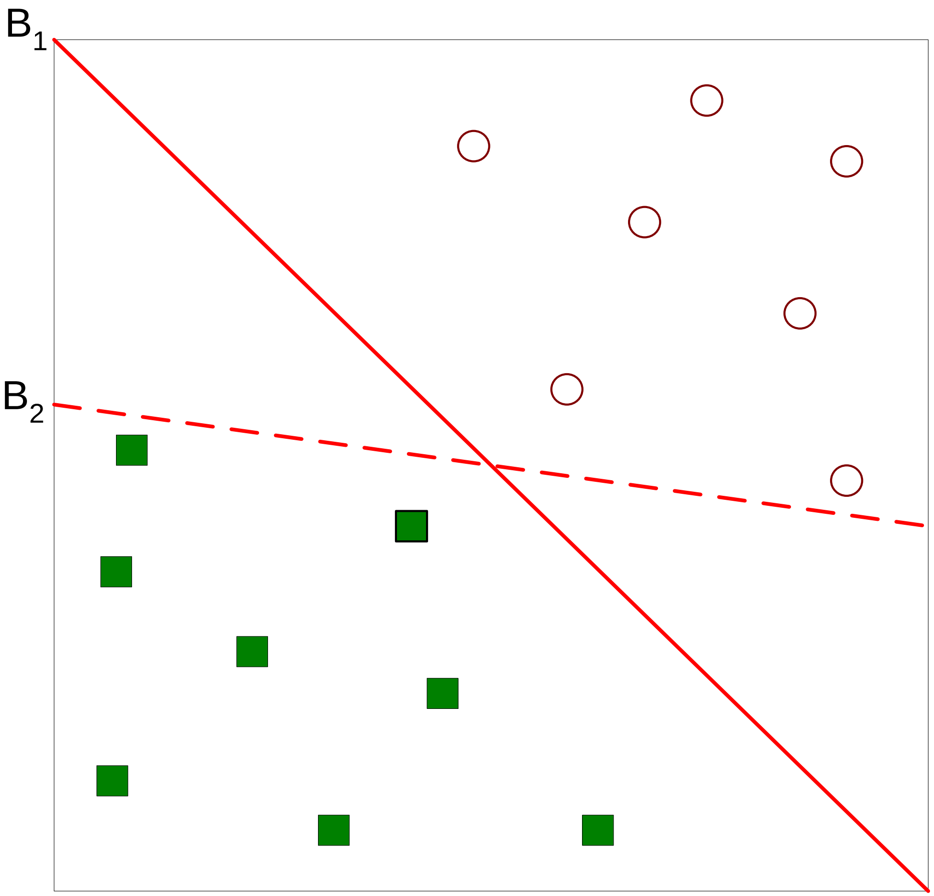

Another possible solution

# Maximum Margin (cont.)

- Although all the hyperplanes shown in the figure can separate training examples perfectly, their generalization errors may be different
- How to choose one of these hyperplanes to construct a classifier's decision boundary with small generalization errors?

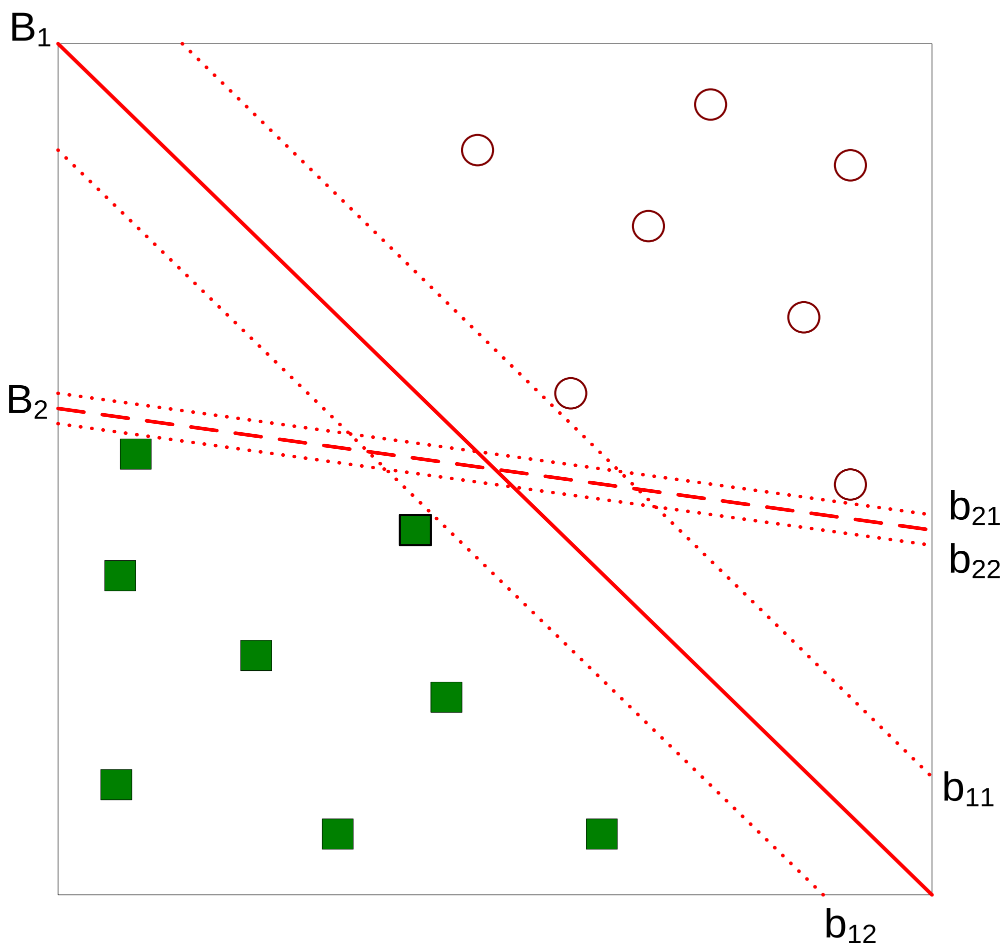


# Maximum Margin (cont.)



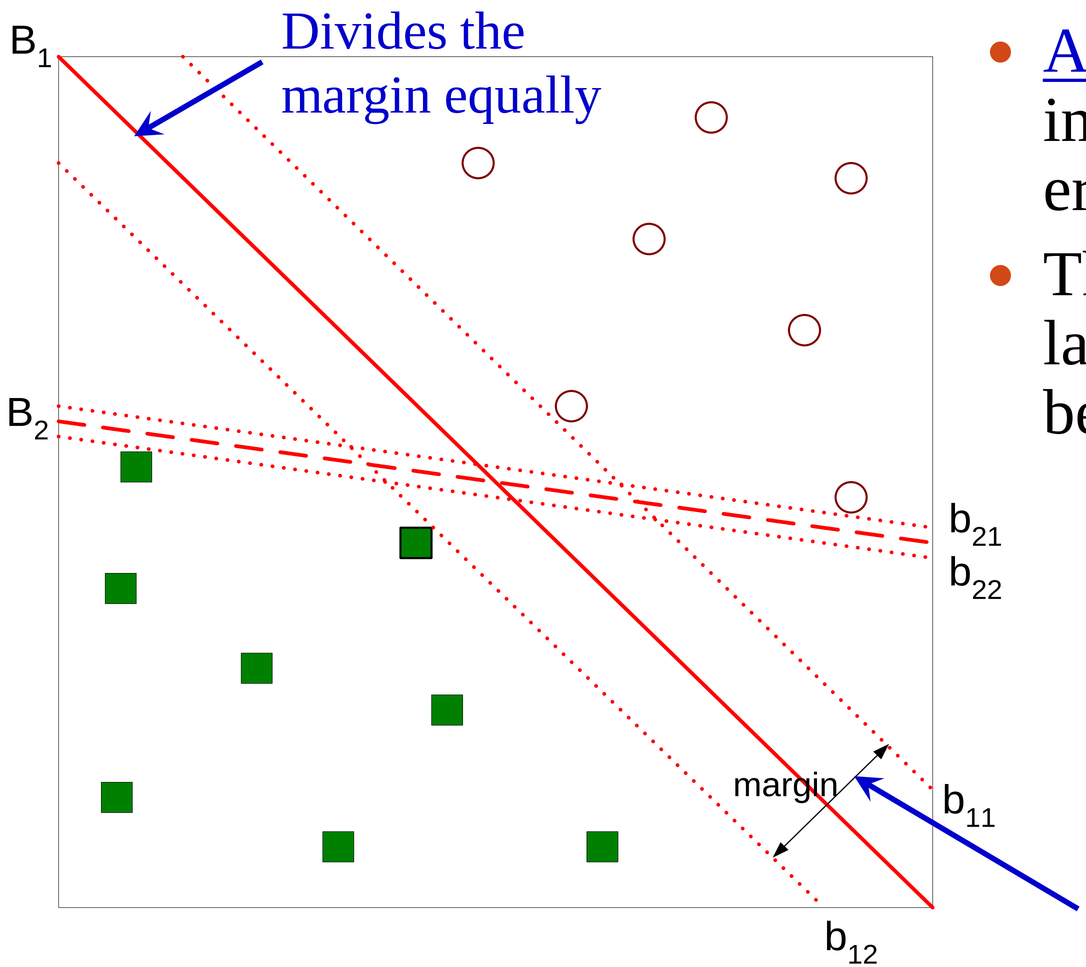

Which one is better? B1 or B2?

# Maximum Margin (cont.)




- Each decision boundary $B_i$ is associated with a pair of parallel hyperplanes: $b_{i1}$ and $b_{i2}$
- $b_{i1}$ is obtained by moving the hyperplane until it touches the closest circle(s)
- $b_{i2}$ is obtained by moving a hyperplane away from the decision boundary until it touches the closest square(s)
- The distances from $b_{i1}$ and $b_{i2}$ to $B_i$ are the same

# Maximum Margin (cont.)

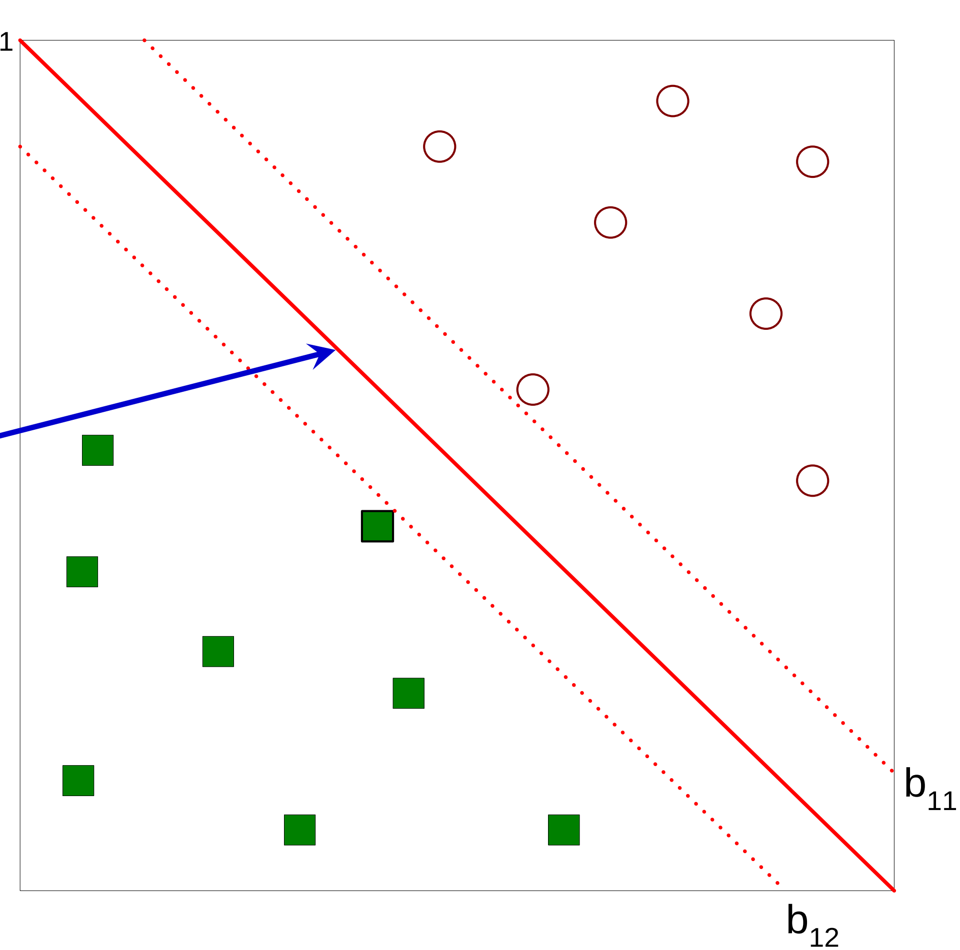



- <u>Assumption:</u> larger margins imply better generalization errors
- The margin of $B_1$ is much larger than that of $B_2$. $B_1$ is better than $B_2$

The distance between these two hyperplanes is known as the <u>margin</u> of the classifier

# Decision Boundary

- Given a binary classification task, denote $y_i = +1$ the circle class, and $y_i = -1$ the square class



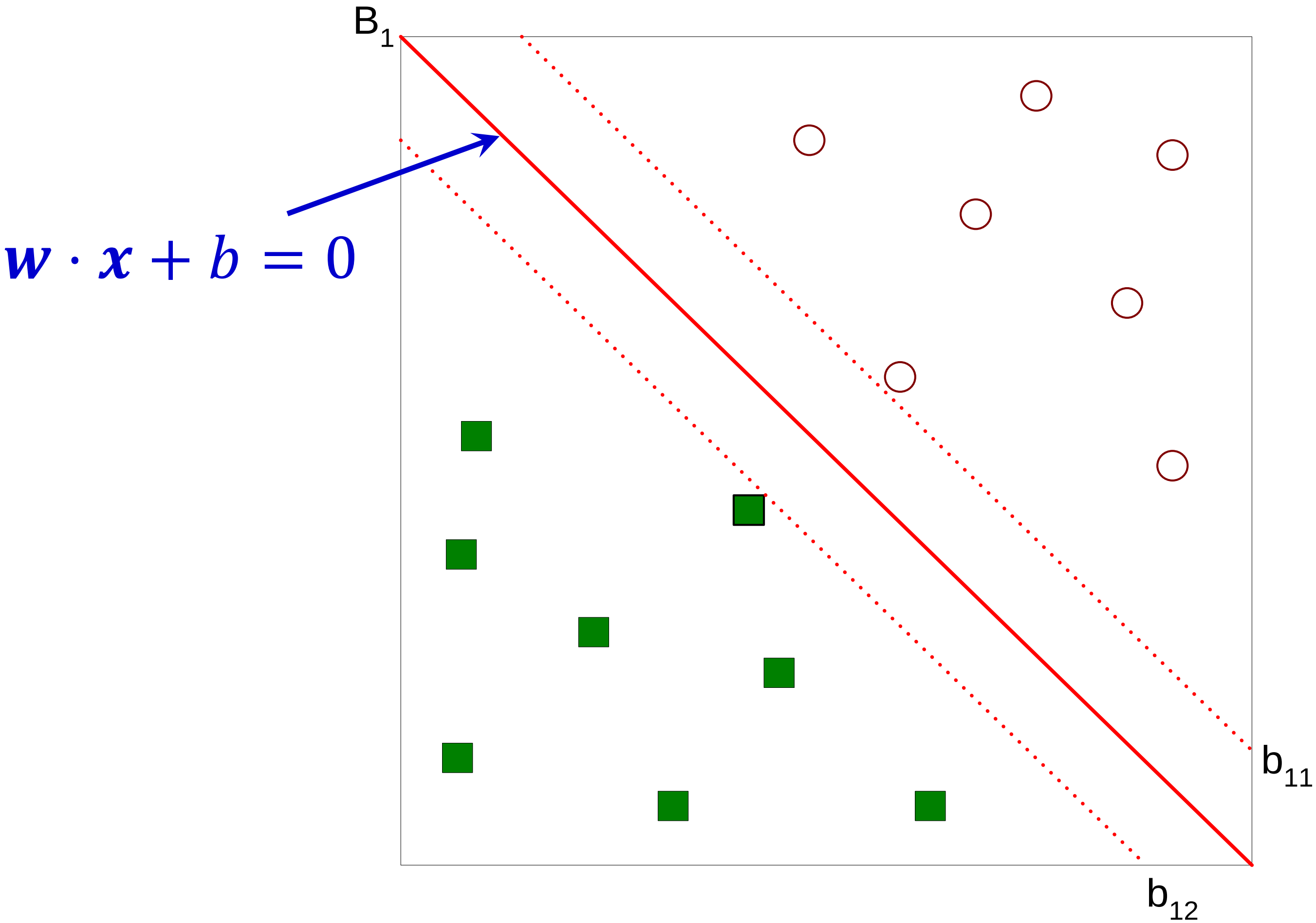

Decision boundary:

$$w_1 x_1 + w_2 x_2 + b = 0$$

General form: $\boldsymbol{w} \boxed{\cdot} \boldsymbol{x} + b = 0$

Inner product: $\boldsymbol{w} \cdot \boldsymbol{x} = \sum_{i=1}^{d} (w_i \times x_i)$

# Making Predictions




For any test example $\boldsymbol{x}^*$: $\begin{cases} f(\boldsymbol{x}^*) = +1, \text{ if } \boldsymbol{w} \cdot \boldsymbol{x}^* + b \geq 0 \\ f(\boldsymbol{x}^*) = -1, \text{ if } \boldsymbol{w} \cdot \boldsymbol{x}^* + b < 0 \end{cases}$

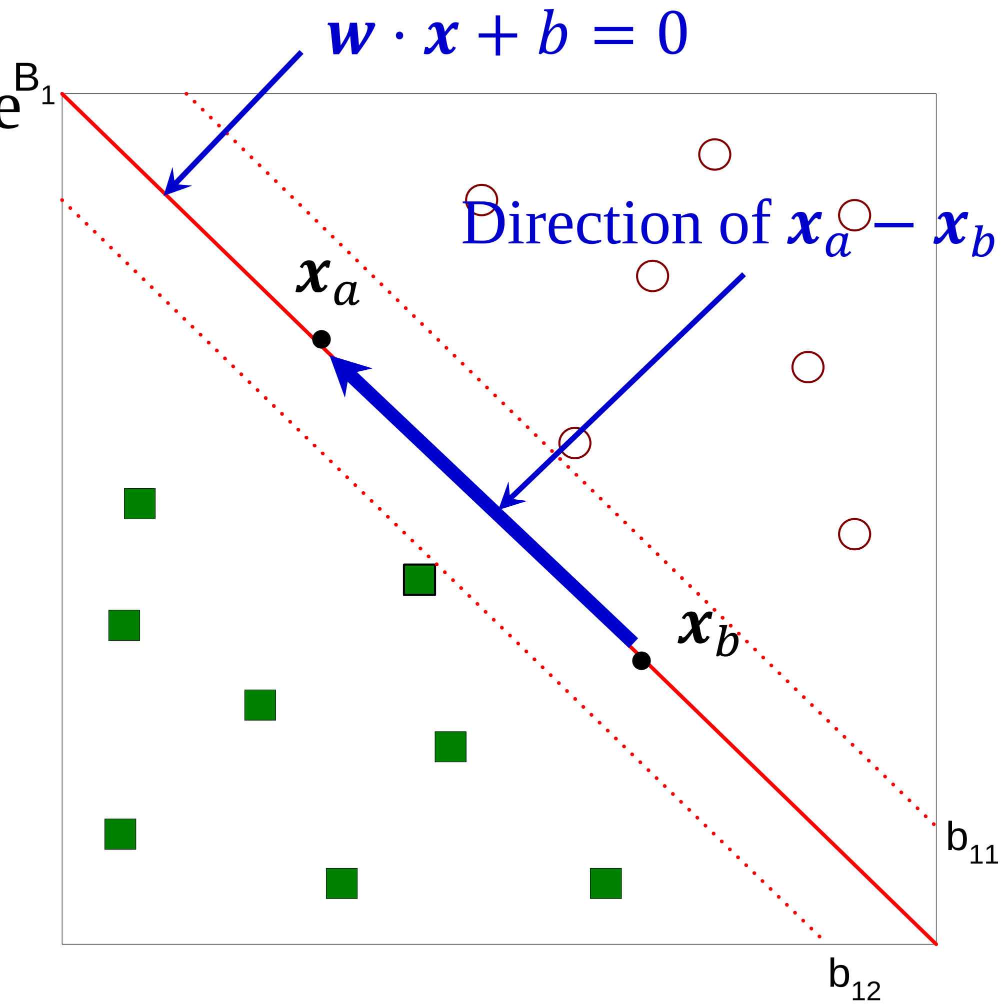
# Margin – Induction

- Suppose $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are two points located on the decision boundary

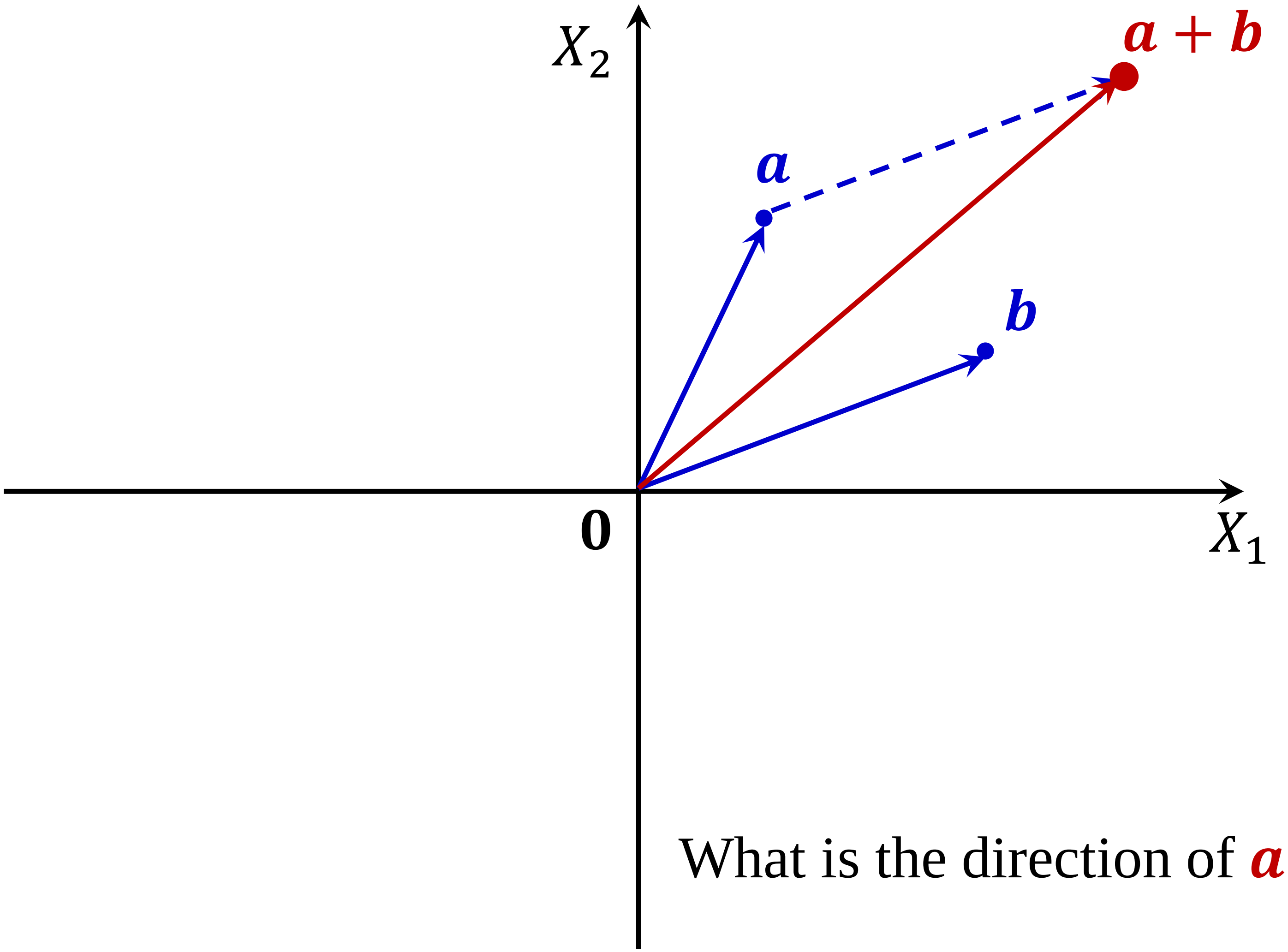
$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_a + b = 0 \\ \boldsymbol{w} \cdot \boldsymbol{x}_b + b = 0 \end{cases}$$

$$\Downarrow$$

$$\boldsymbol{w} \cdot (\boldsymbol{x}_a - \boldsymbol{x}_b) = 0$$



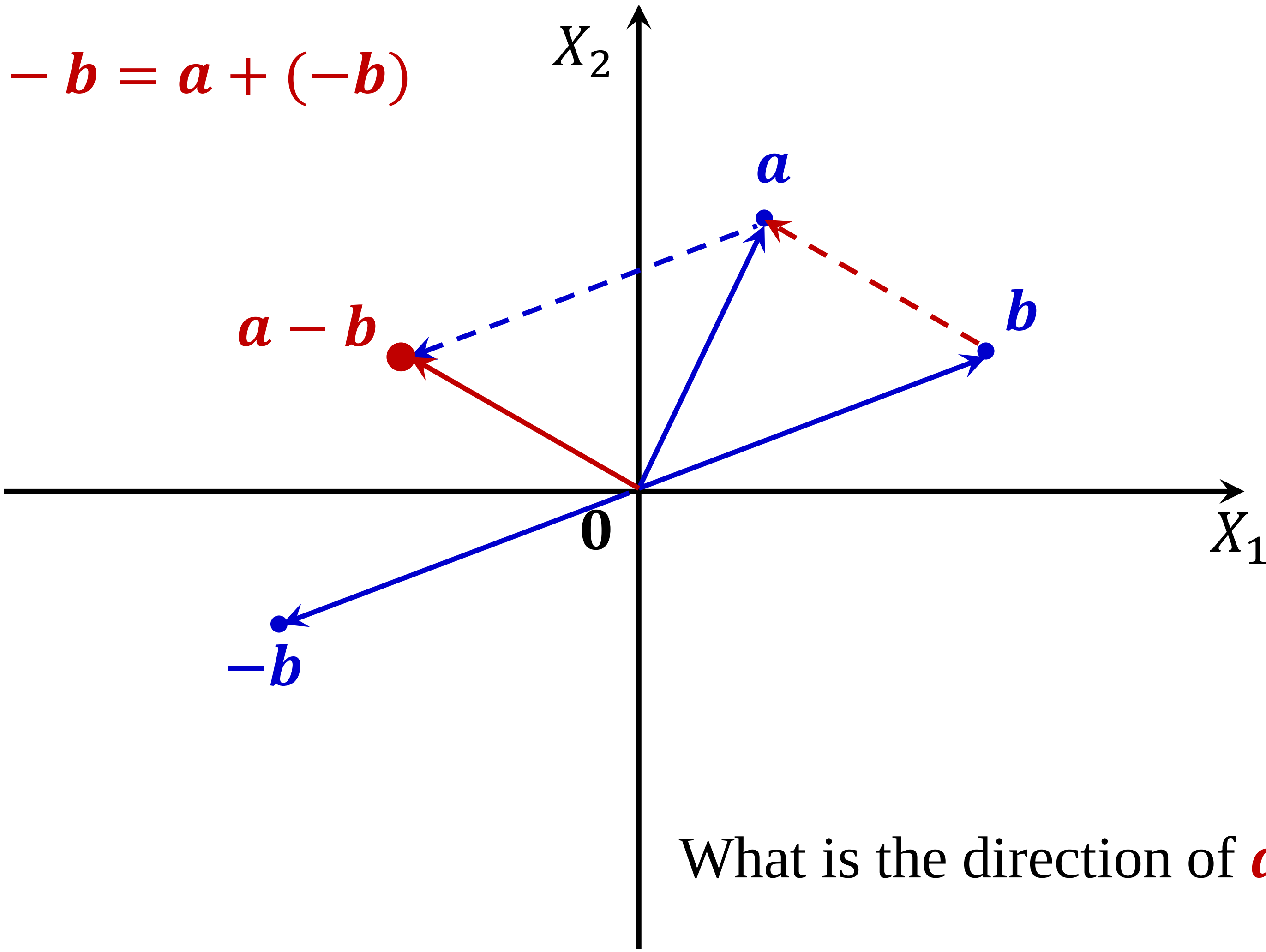
# Review: Direction of Vectors




What is the direction of $\boldsymbol{a} + \boldsymbol{b}$?

# Review: Direction of Vectors (cont.)

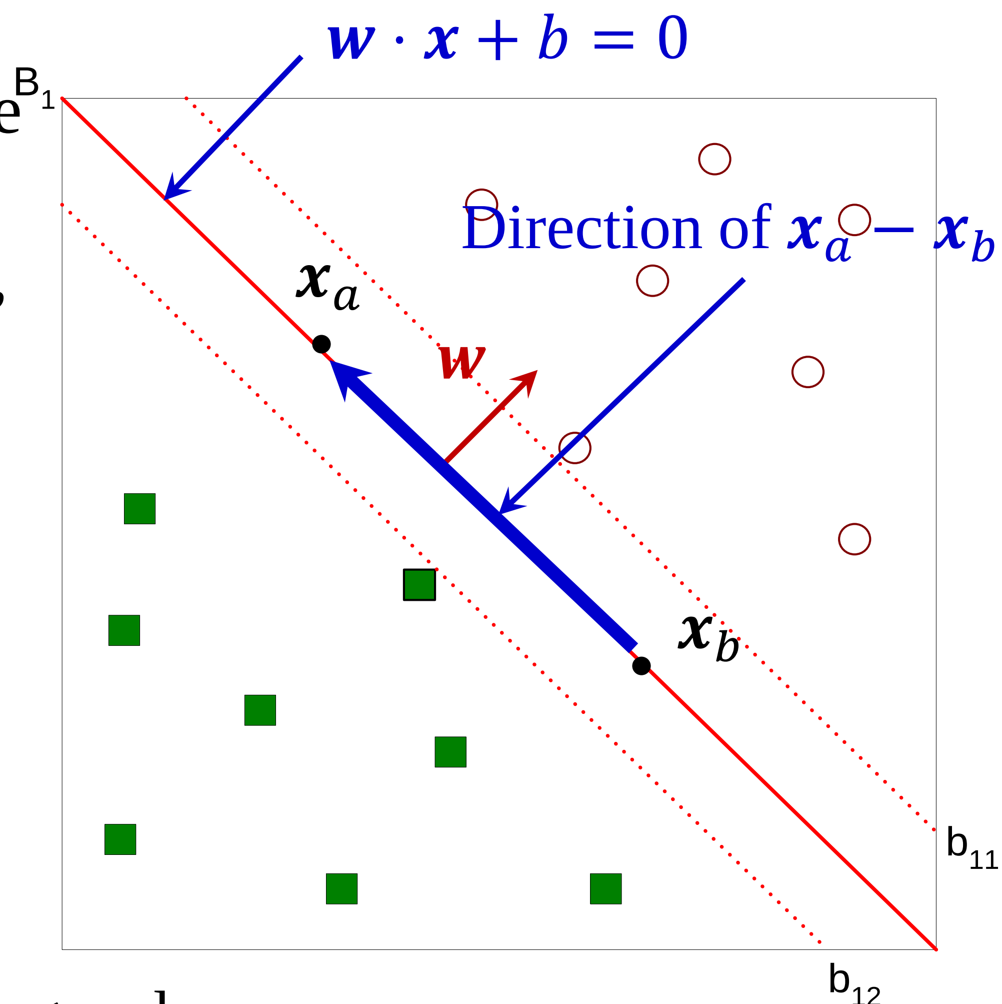
$\boldsymbol{a} - \boldsymbol{b} = \boldsymbol{a} + (-\boldsymbol{b})$




What is the direction of $\boldsymbol{a} - \boldsymbol{b}$?

# Margin – Induction (cont.)

- Suppose $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are two points located on the decision boundary,




$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_a + b = 0 \\ \boldsymbol{w} \cdot \boldsymbol{x}_b + b = 0 \end{cases}$$

$$\Downarrow$$

$$\boldsymbol{w} \cdot (\boldsymbol{x}_a - \boldsymbol{x}_b) = 0$$

Based on definition of inner product

$$\Downarrow$$

The direction of $\boldsymbol{w}$ is orthogonal to the decision boundary

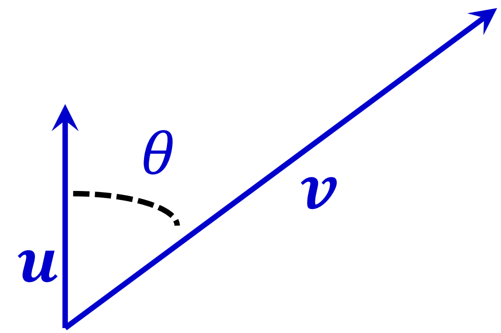
# Inner Product: Review

- Given two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of $d$ dimensions, the inner (or dot) product of $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as

$$\boldsymbol{u} \cdot \boldsymbol{v} = \sum_{i=1}^{d} (u_i \times v_i)$$

- From the geometry viewpoint:

$$\boldsymbol{u} \cdot \boldsymbol{v} = \boxed{\|\boldsymbol{u}\|_2} \times \|\boldsymbol{v}\|_2 \times \cos(\theta)$$




$$\|\boldsymbol{u}\|_2 = \sqrt{\boldsymbol{u} \cdot \boldsymbol{u}} = \sqrt{\sum_{i=1}^{d} (u_i \times u_i)}$$

Length of $\boldsymbol{u}$

When $\boldsymbol{u} \cdot \boldsymbol{v} = 0 \Rightarrow \cos(\theta) = 0$

$\Rightarrow \theta = 90°$

$\Rightarrow$ $\boldsymbol{u}$ and $\boldsymbol{v}$ are orthogonal

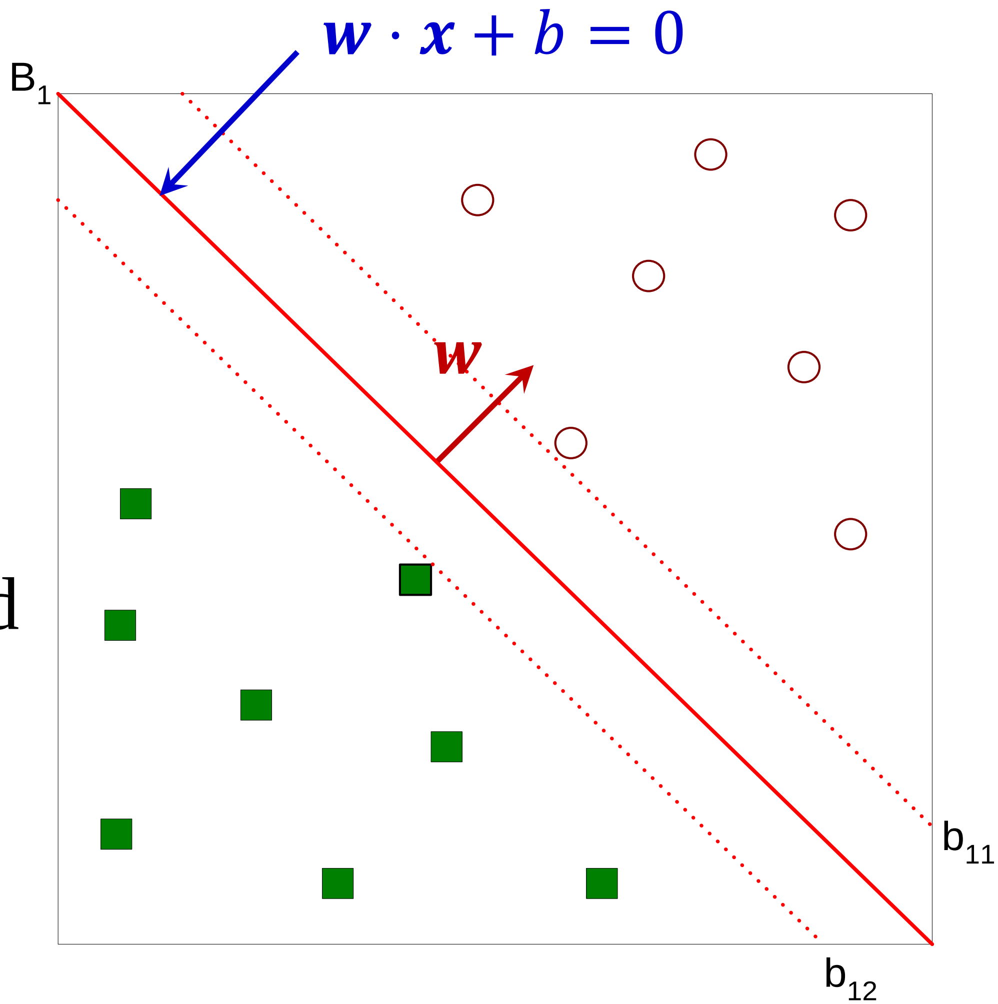
# Margin – Induction (cont.)

- For any circle $\boldsymbol{x}_c$ located above the decision boundary:

  $$\boldsymbol{w} \cdot \boldsymbol{x_c} + b \geq k,$$
  where $k > 0$

- For any square $\boldsymbol{x}_s$ located below the decision boundary:

  $$\boldsymbol{w} \cdot \boldsymbol{x_s} + b \leq k',$$
  where $k' < 0$



# Margin – Induction (cont.)

The two parallel hyperplanes passing the closest circle(s) and square(s) can be written as

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = k, \text{where } k > 0$$

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = k', \text{where } k' < 0$$

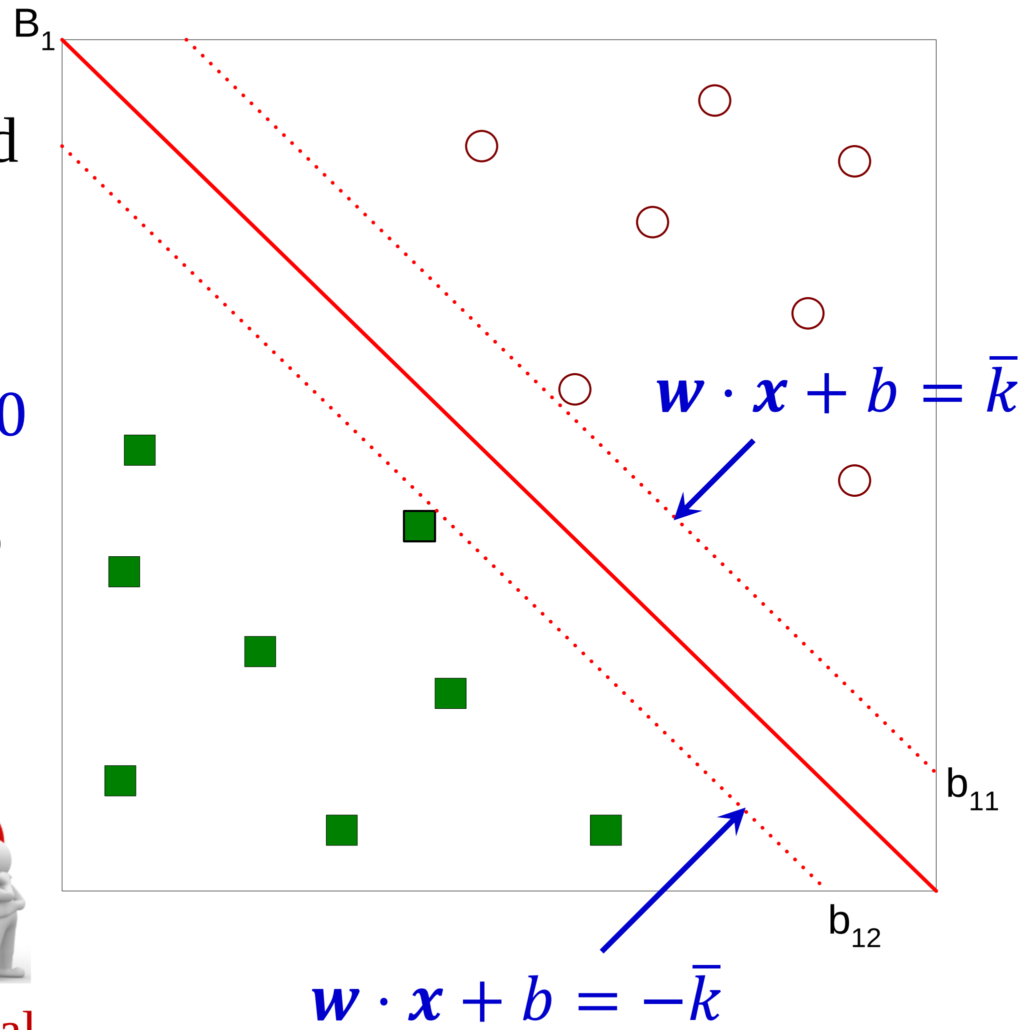
It can be shown that, these two parallel hyperplanes can be further rewritten as

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = \bar{k}$$

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = -\bar{k}$$

$$\text{where } \bar{k} > 0$$

Tutorial



# Margin – Induction (cont.)

The two parallel hyperplanes passing the closest circle(s) and square(s) can be written as

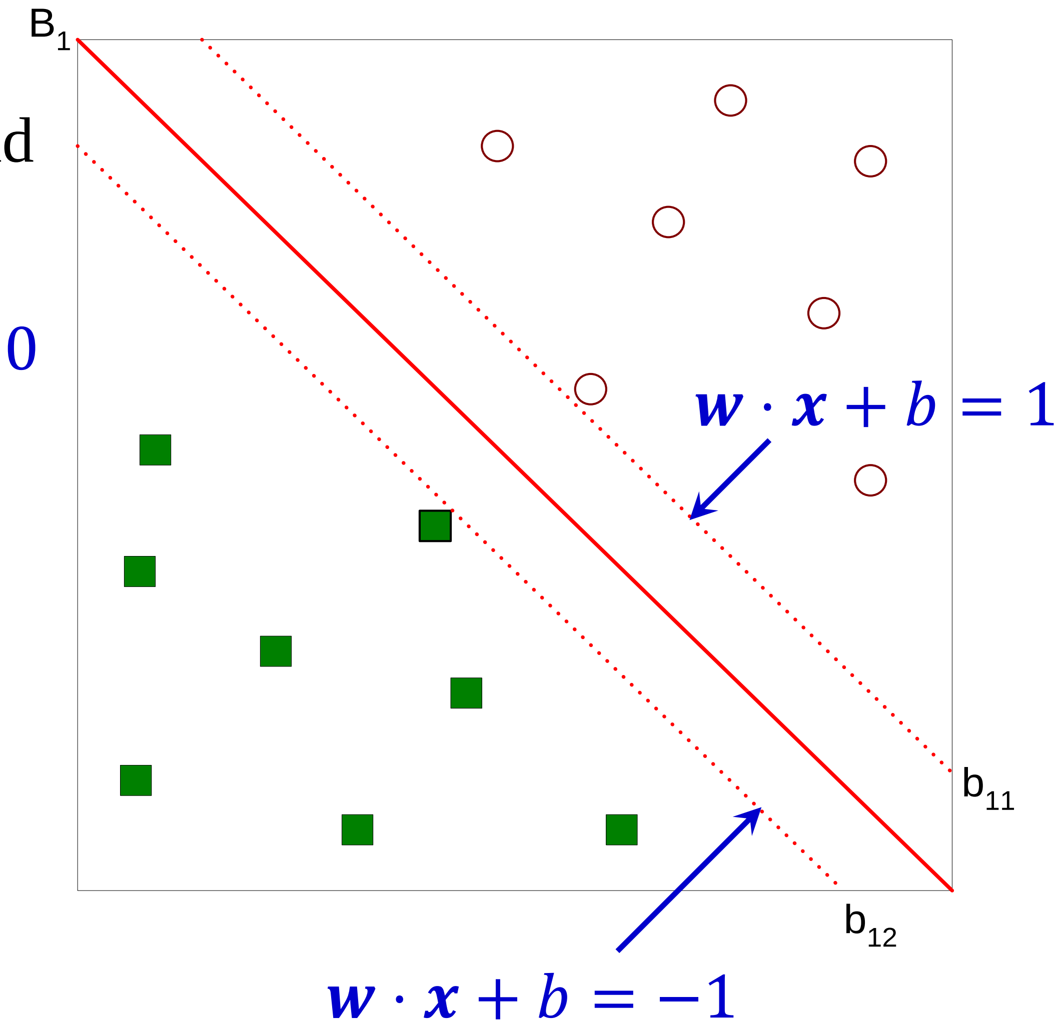
$$\boldsymbol{w} \cdot \boldsymbol{x} + b = \bar{k}$$
$$\boldsymbol{w} \cdot \boldsymbol{x} + b = -\bar{k}$$

where $\bar{k} > 0$

$$\boldsymbol{w} = \frac{\boldsymbol{w}}{\bar{k}} \qquad b = \frac{b}{\bar{k}}$$

After rescaling $\boldsymbol{w}$ and $b$, the two parallel hyperplanes can be further rewritten as

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 1$$
$$\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$$



# Margin – Induction (cont.)

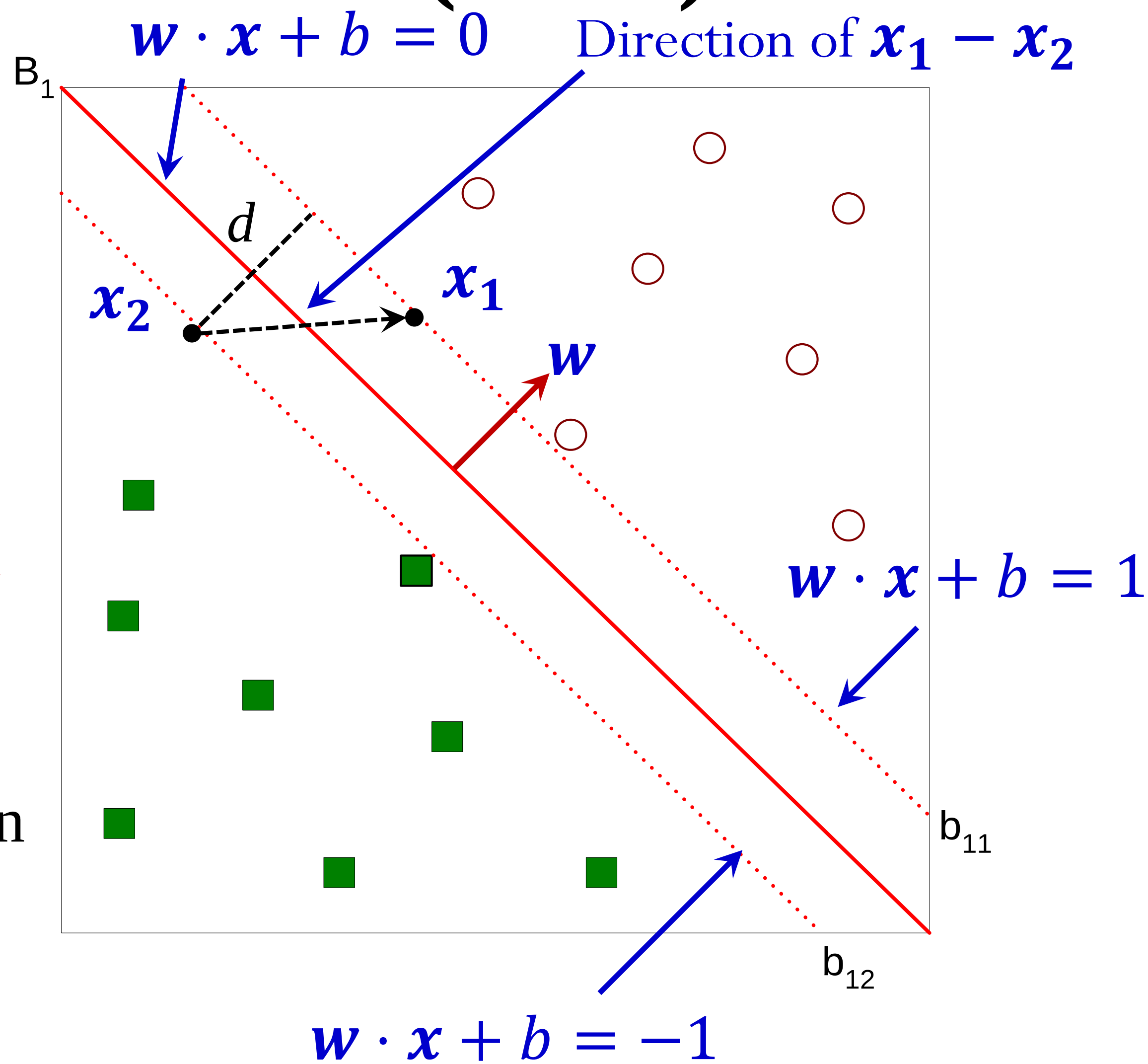
$\mathrm{b}_{11}$: $\boldsymbol{w}\cdot\boldsymbol{x}_1 + b = 1$

$\mathrm{b}_{12}$: $\boldsymbol{w}\cdot\boldsymbol{x}_2 + b = -1$

$$\boldsymbol{w}\cdot(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 2$$

Based on definition of inner product

$$\|\boldsymbol{w}\|_2 \times \underbrace{\boxed{d}}_{\text{Margin}} = 2$$




# Margin – Induction (cont.)

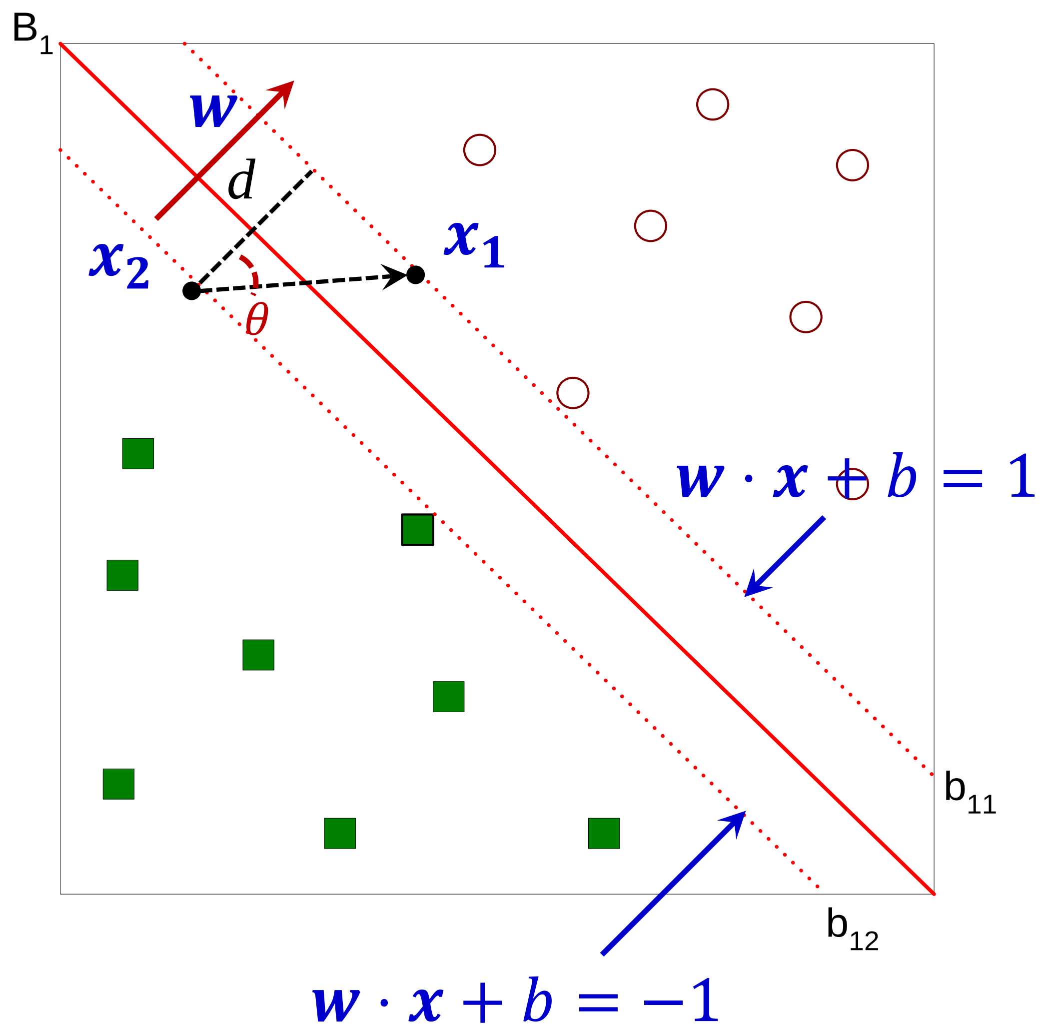
$$\boldsymbol{w} \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) = 2$$

Based on definition of inner product

$$\|\boldsymbol{w}\|_2 \times \boxed{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \times \cos(\theta)} = 2$$

Based on definition of $\cos(\cdot)$ $\quad = d$

$$\|\boldsymbol{w}\|_2 \times d = 2$$



# Margin Maximization

$$\|\boldsymbol{w}\|_2 \times d = 2 \Longrightarrow d = \frac{2}{\|\boldsymbol{w}\|_2}$$

$$\|\boldsymbol{w}\|_2^2 = \sum_{i=1}^{d} (w_i \times w_i)$$

Maximize margin $d = \frac{2}{\|\boldsymbol{w}\|_2}$ $\Longrightarrow$ Minimize $\frac{\|\boldsymbol{w}\|_2^{\boxed{2}}}{2}$

For convenience in computation

Constraints: $\boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 1, \text{if } y_i = 1$

$\boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq -1, \text{if } y_i = -1$

**OR**

$$y_i \times (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1$$

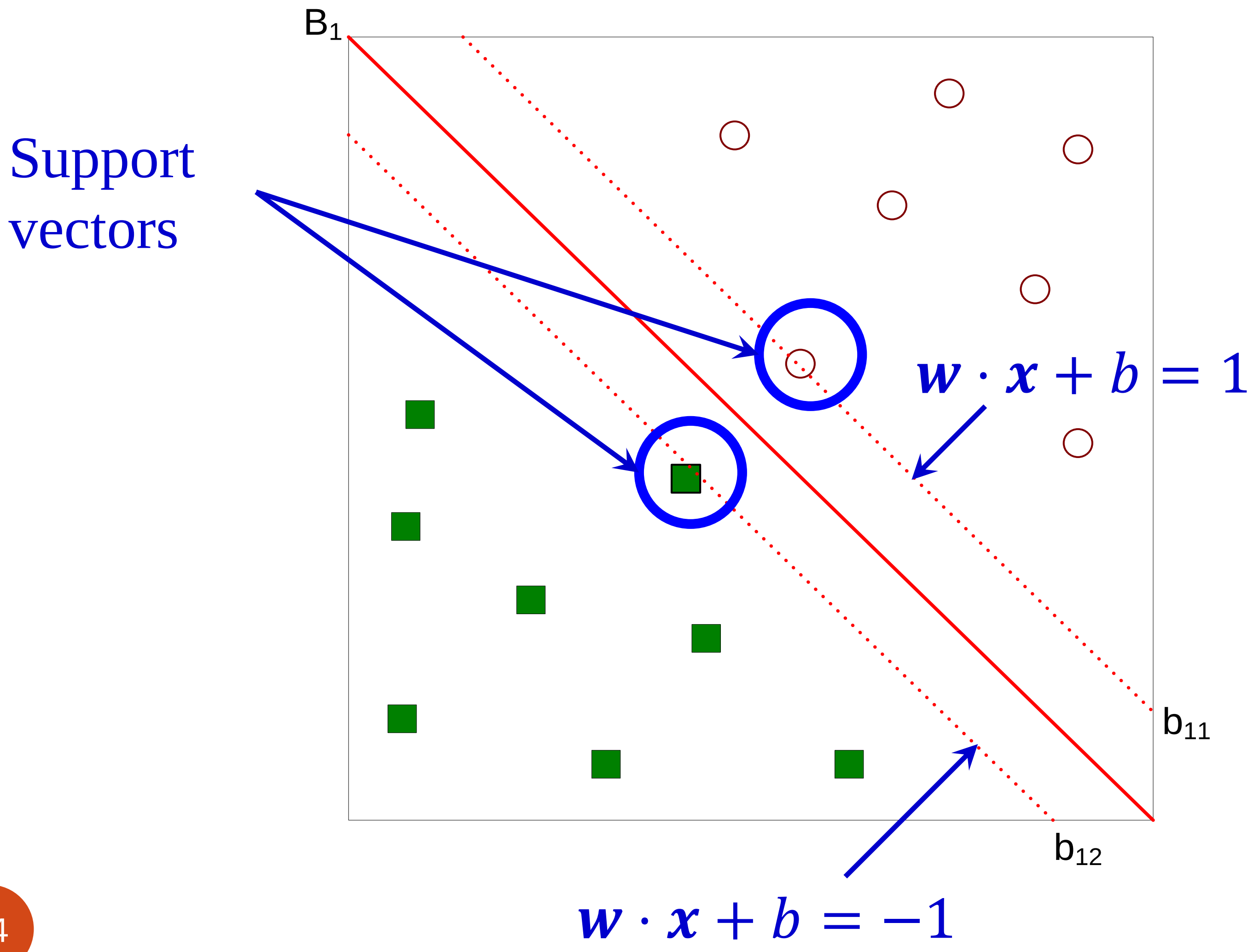
# Optimization Problem for SVMs

- Optimization problem of linear SVMs

$$\min_{\boldsymbol{w},b} \ \frac{\|\boldsymbol{w}\|_2^2}{2}$$
$$\text{s.t.} \quad y_i \times (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1,\ i = 1, \dots, N$$

- The optimization is convex
  - Many numerical approaches can be applied to solve it

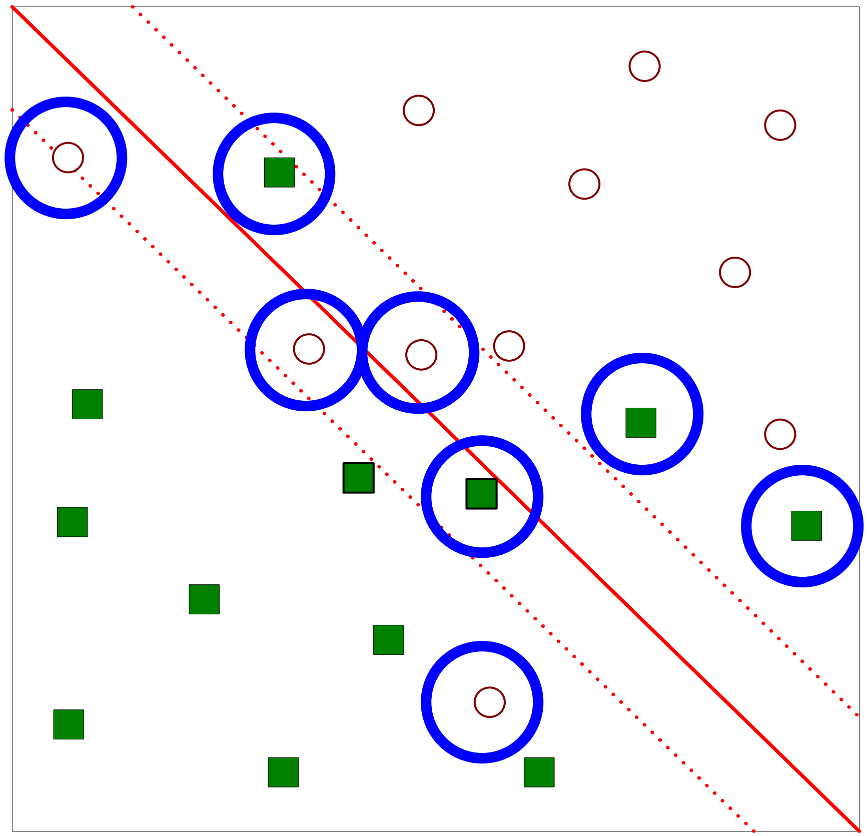
# Support Vectors



# Non-separable Case

**Additional Notes (Optional)**

- What if data of two classes cannot be perfectly separated?



Slack variables need to be introduced to absorb errors

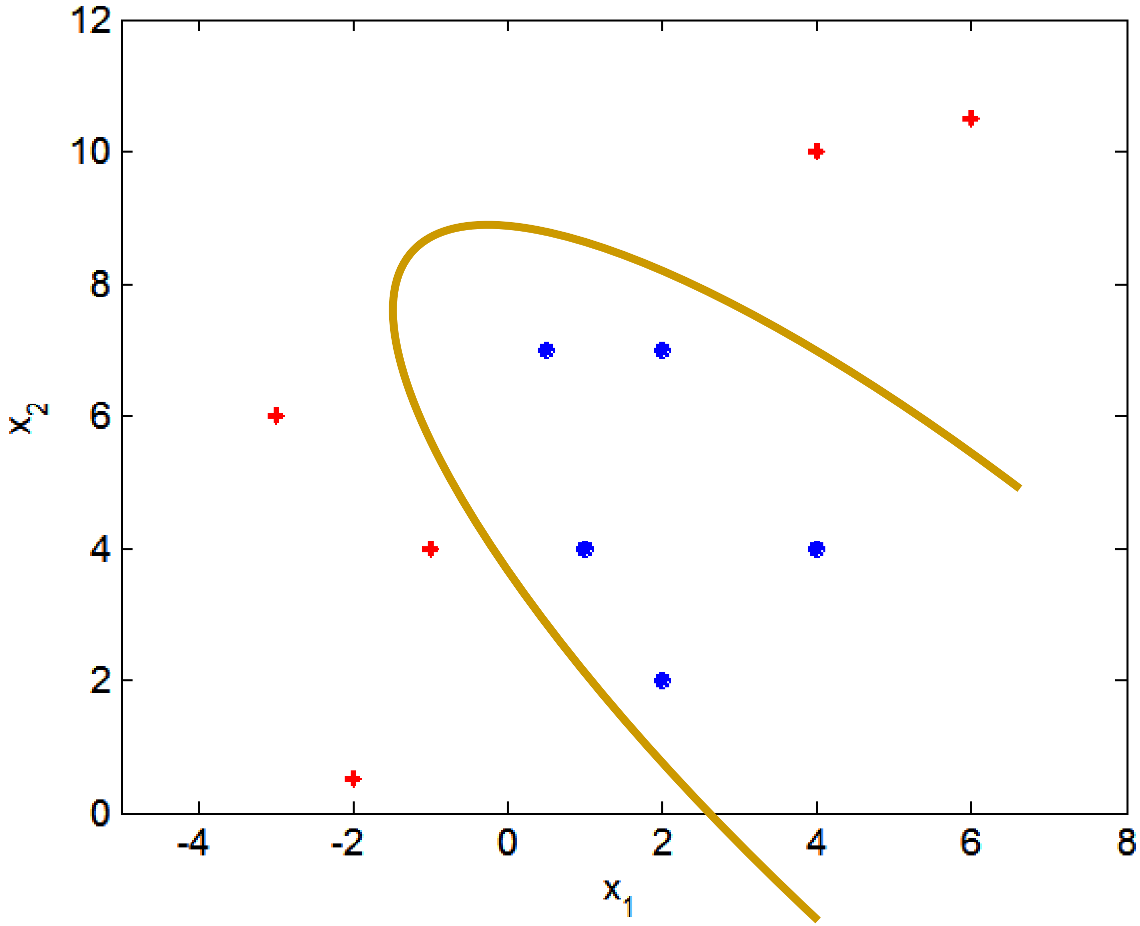
# Implementation Example

```
>>> from sklearn import svm

...

>>> svmC = svm.LinearSVC()
>>> svmC.fit(X, y)
>>> pred= svmC.predict(X)
```

Additional Notes (Optional)

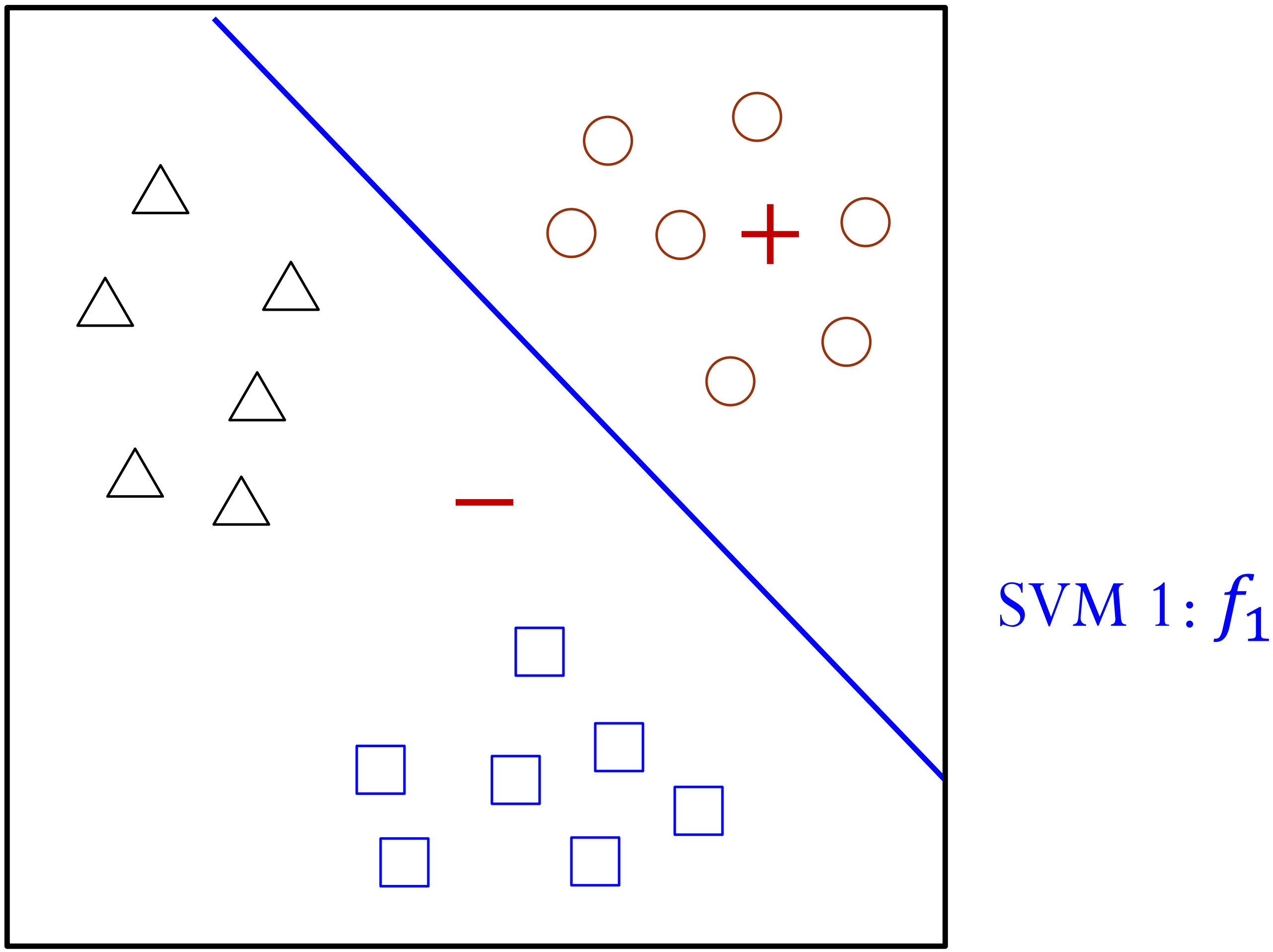
# Nonlinear SVMs

- What if decision boundary is not linear?

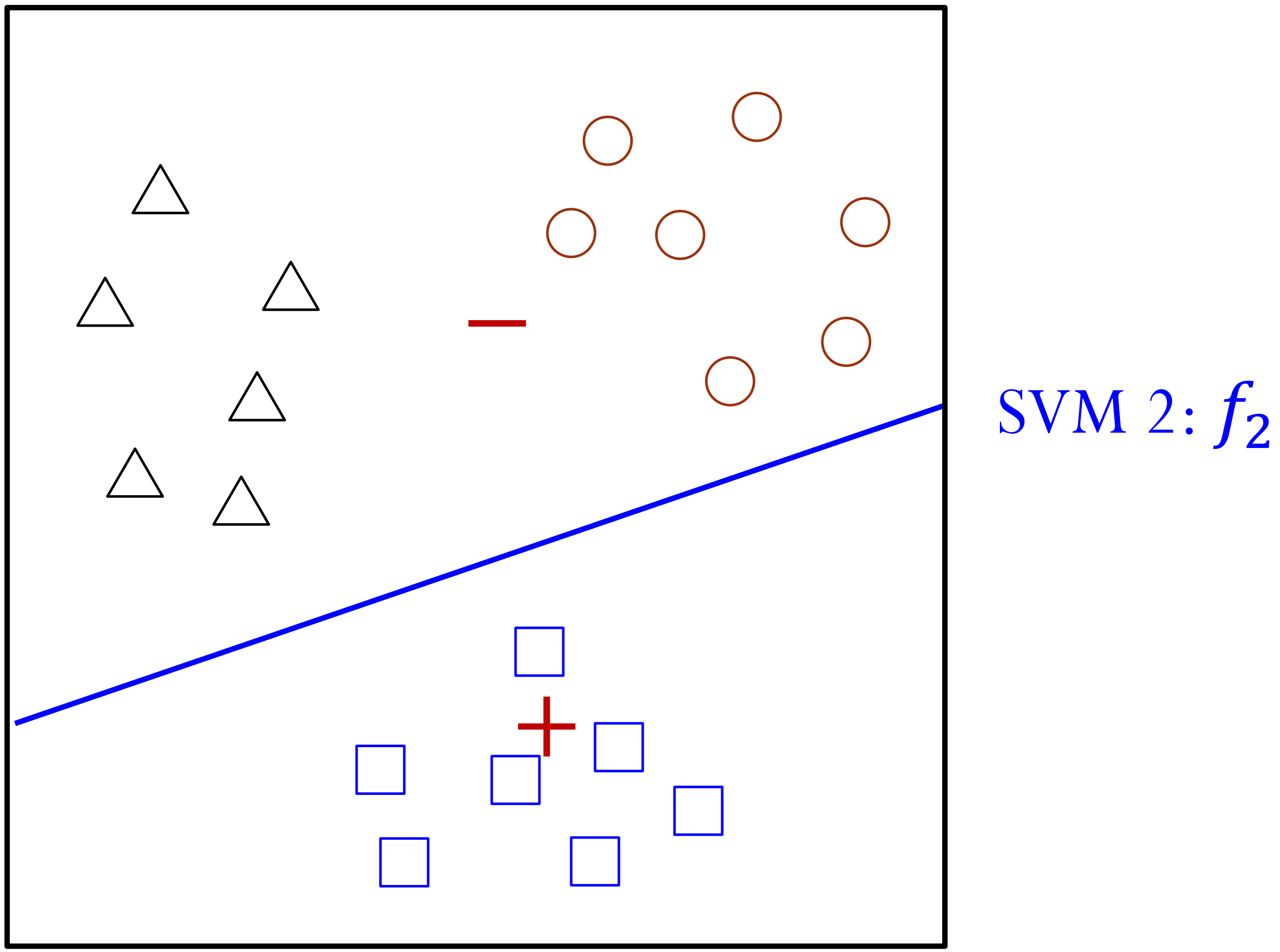



Kernel trick in the dual form

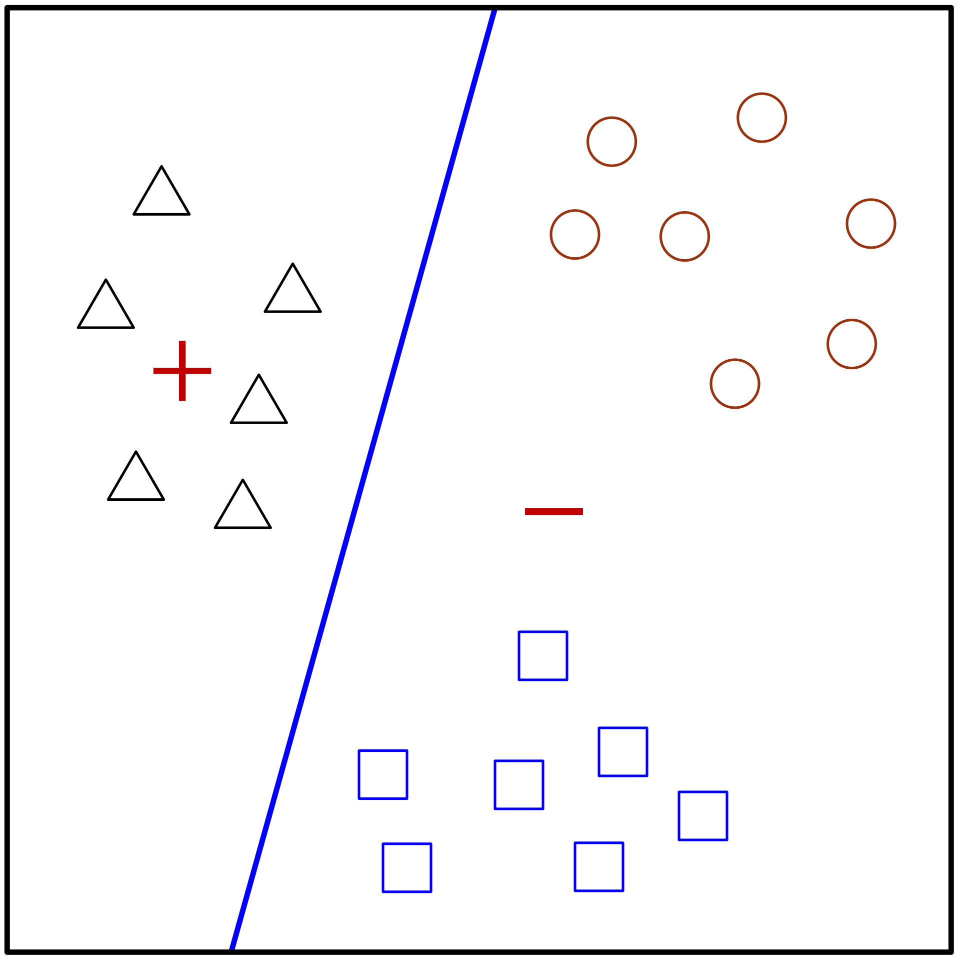
# Multi-Class Classification



# Multi-Class Classification (cont.)



# Multi-Class Classification (cont.)




SVM 3: $f_3$

# Multi-Class Classification (cont.)

- Give a 3-class classification problem: $C_1$, $C_2$ & $C_3$
- General approaches: 1 v.s. rest
  - Binary classification 1: positive ($C_1$) v.s. negative ($C_2$&$C_3$)
  - Binary classification 2: positive ($C_2$) v.s. negative ($C_1$&$C_3$)
  - Binary classification 3: positive ($C_3$) v.s. negative ($C_1$&$C_2$)
  - For a test instance $\boldsymbol{x}^*$, apply binary classifier $f_1$, $f_2$, and $f_3$ to make predictions on $\boldsymbol{x}^*$
  - Combine predicted results of $f_1(\boldsymbol{x}^*)$, $f_2(\boldsymbol{x}^*)$, and $f_3(\boldsymbol{x}^*)$ to make a final prediction

# Linear SVMs for Multi-Class

- $f_i$ only generates $-1/1$:
  1: belong to $C_i$, and $-1$: not belong to $C_i$
- Given a test data $\boldsymbol{x}^*$, suppose

| | $\boldsymbol{C_1}$ | $\boldsymbol{C_2}$ | $\boldsymbol{C_3}$ |
|---|---|---|---|
| $f_1(\boldsymbol{x}^*) = -1$ | 0 | 1 | 1 |
| $f_2(\boldsymbol{x}^*) = 1$ | 0 | 1 | 0 |
| $f_3(\boldsymbol{x}^*) = -1$ | 1 | 1 | 0 |
| **Total Votes:** | 1 | 3 | 1 |

# Thank you!