

CZ4041/SC4000: Machine Learning

Lesson 8a: Support Vector Machines

LI Boyang, Albert

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)

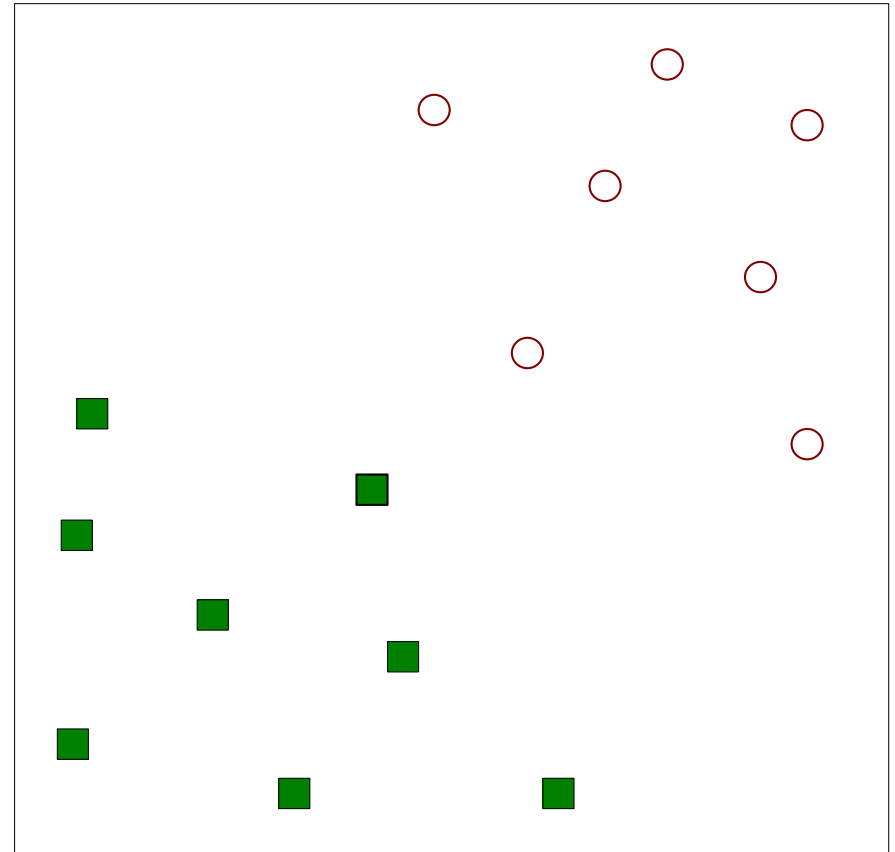
- SVMs have shown promising empirical results in many practical applications, such as computer vision, sensor networks and text mining
- The motivation behind SVMs is from the geometry perspective of linear algebra
- The objective of SVMs is to learn a maximum margin hyperplane
 - Based on statistical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

Separating Hyperplane

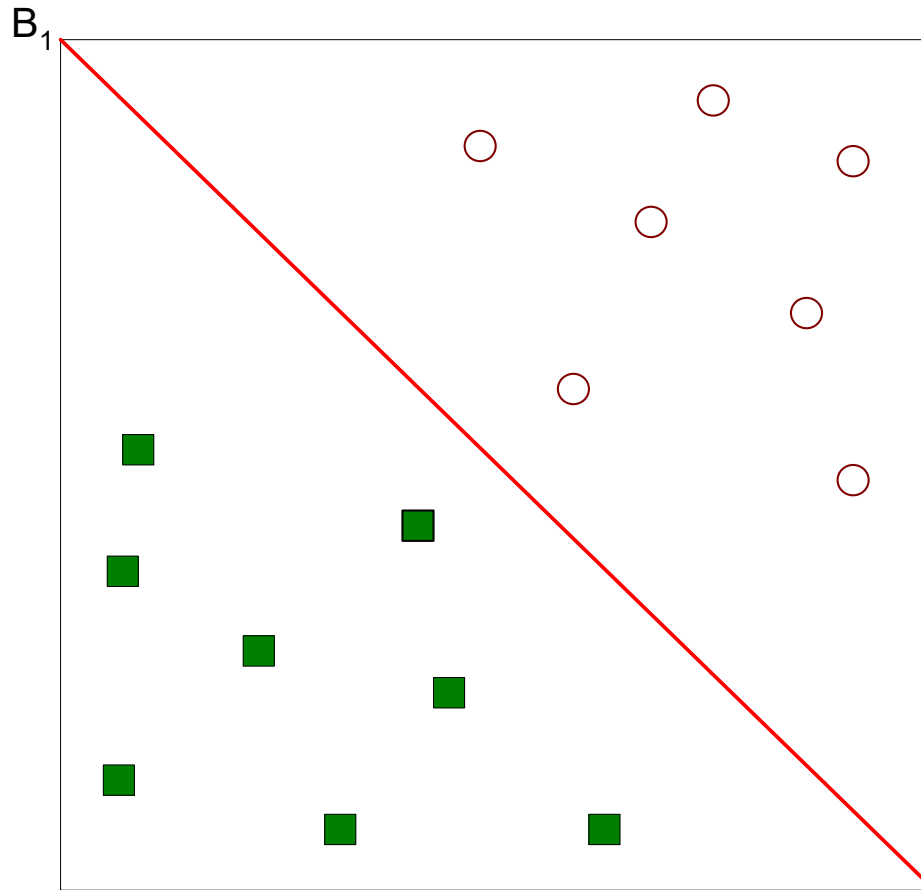
- To learn a binary classifier



- 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

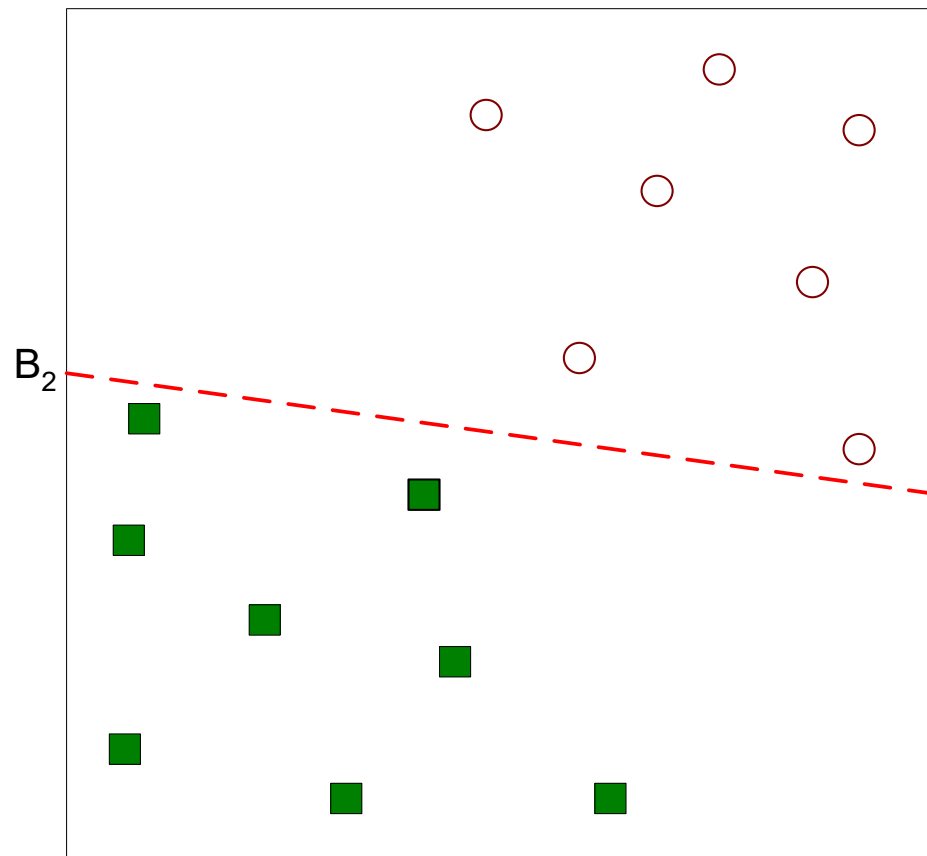


Separating Hyperplane (cont.)



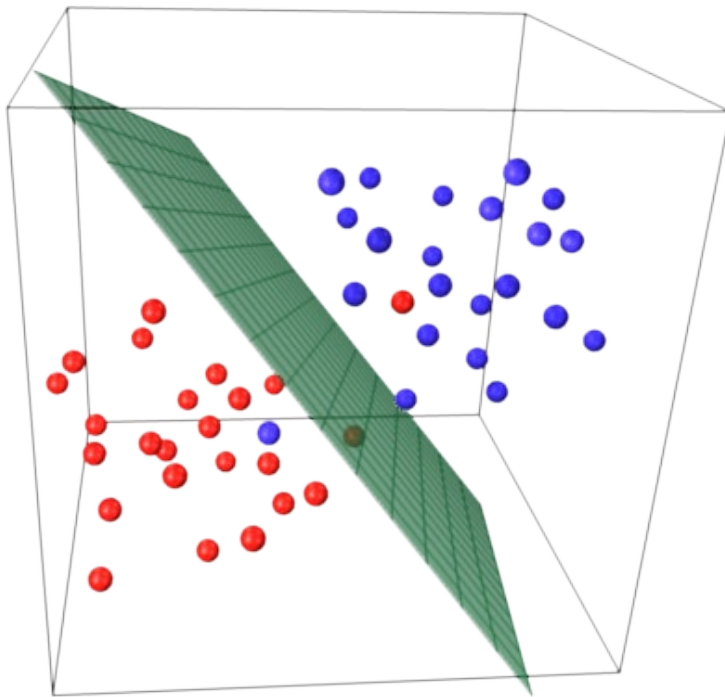
One Possible Solution

Separating Hyperplane (cont.)

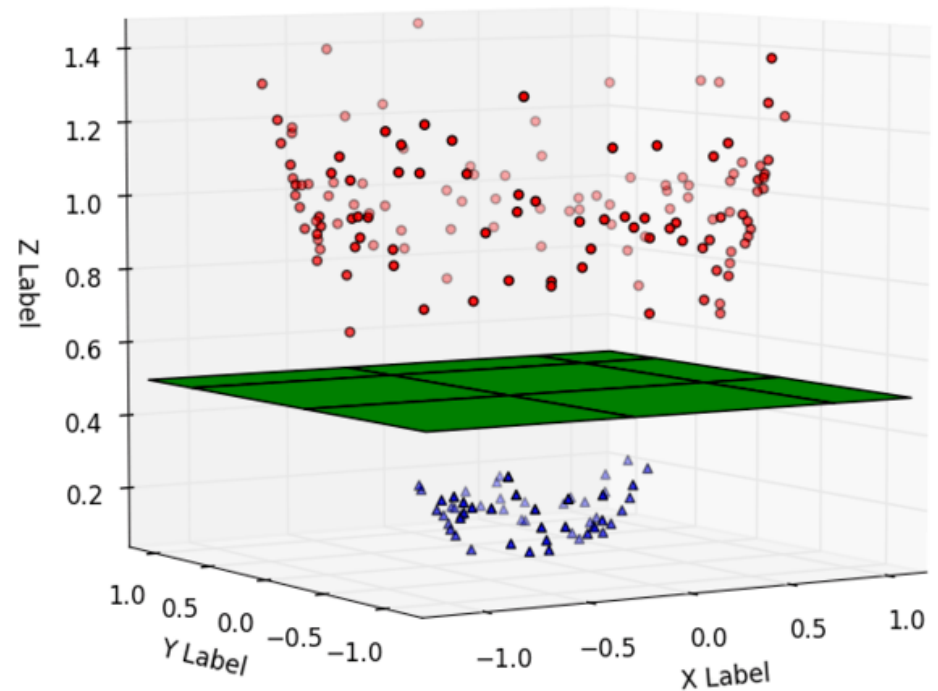


Another possible solution

Separating Hyperplane in 3D

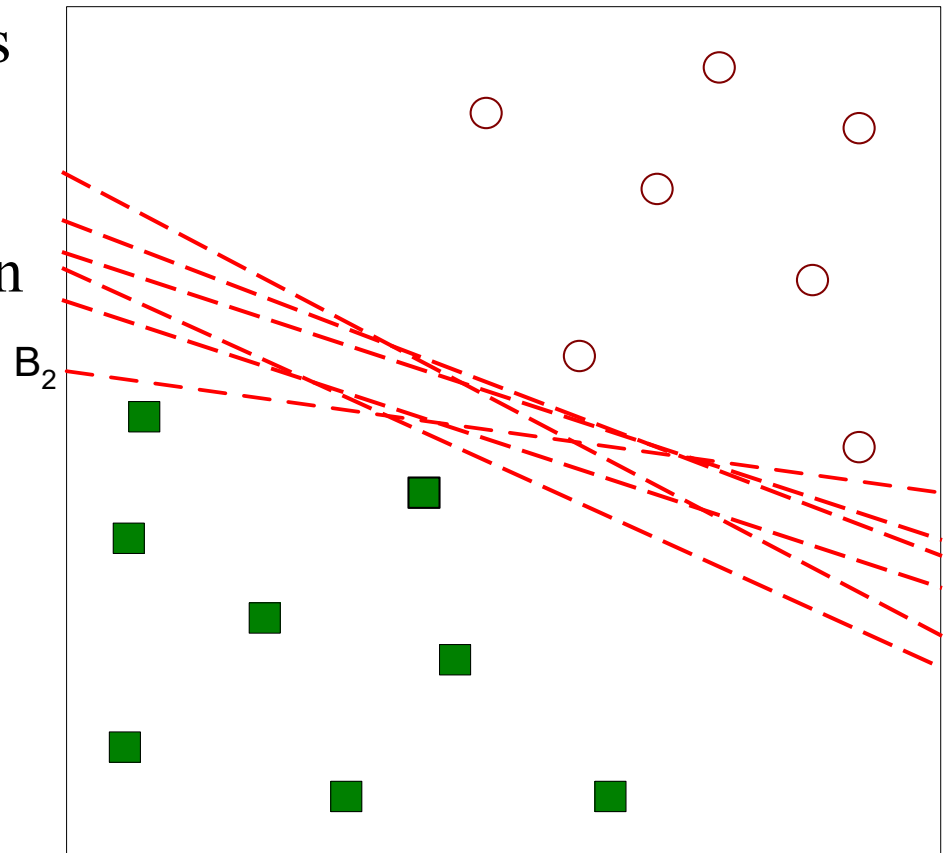


Data in \mathbb{R}^3 (separable w/ hyperplane)

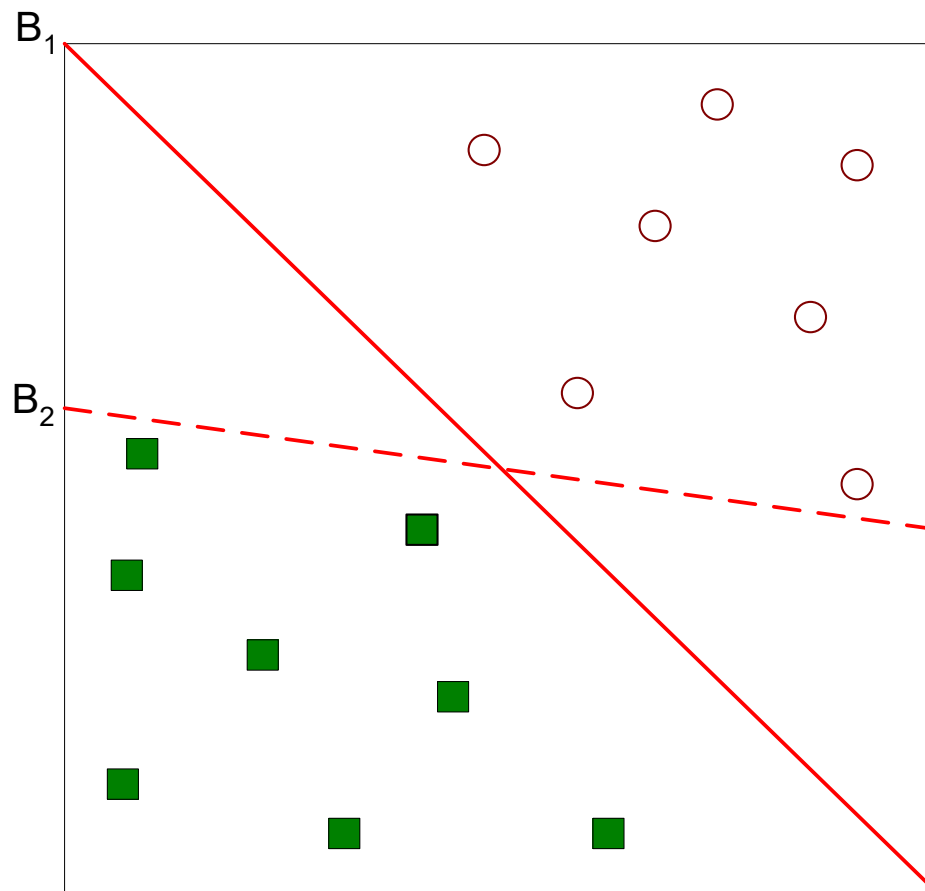


Maximum Margin

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

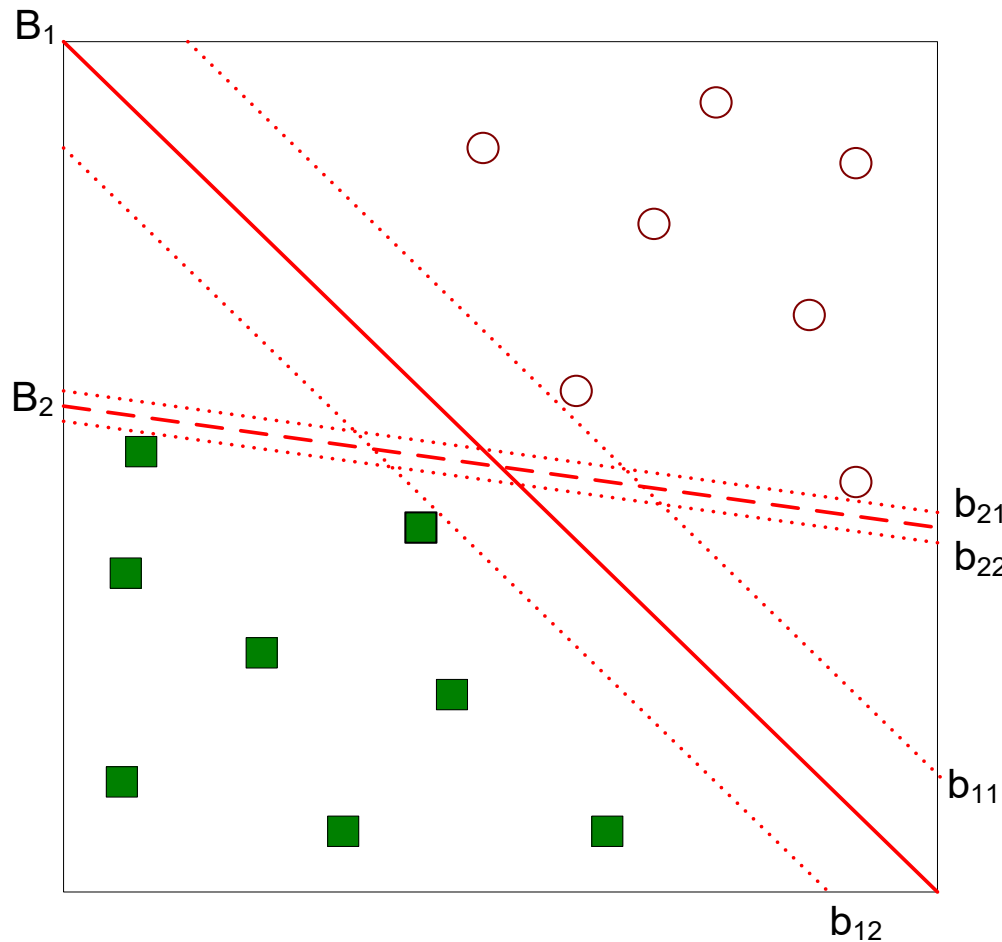


Maximum Margin (cont.)



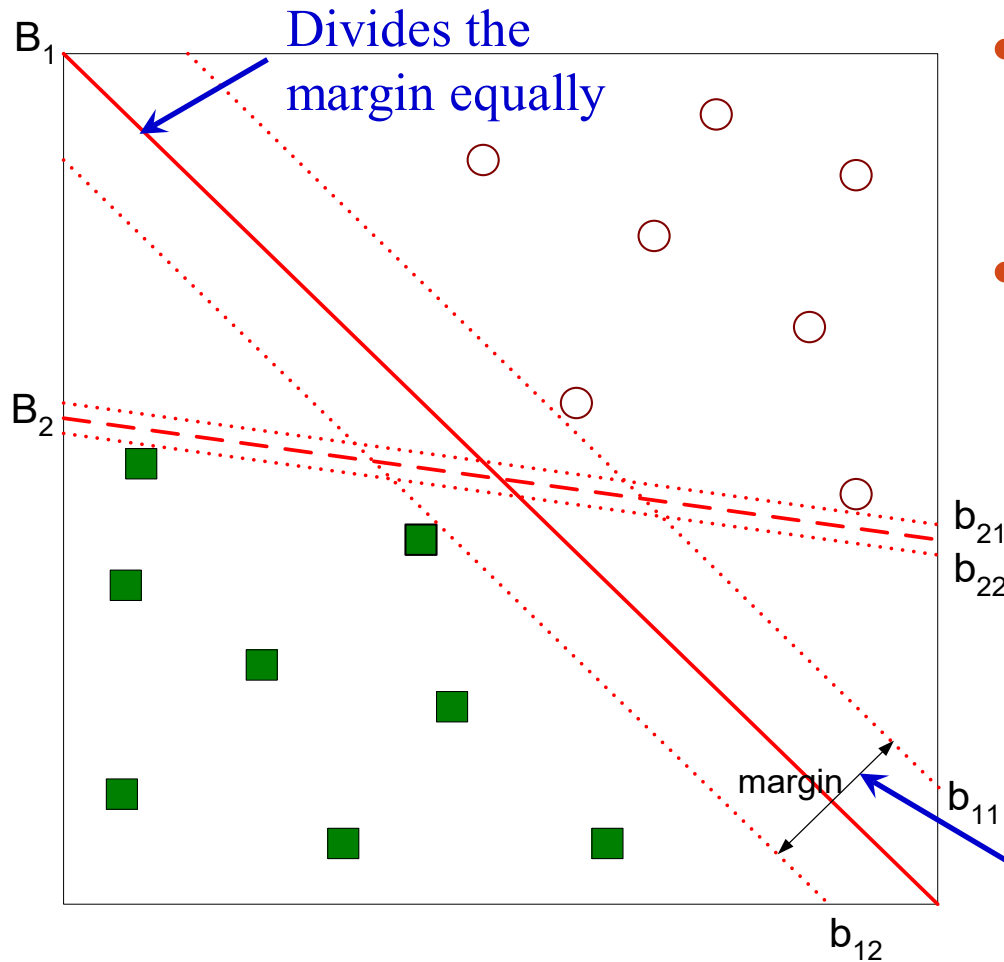
Which one is better? B_1 or B_2 ?

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.)



- Assumption: larger margins imply better generalization errors
- The margin of B_1 is much larger than that of B_2 . Therefore, B_1 is better than B_2

The distance between these two hyperplanes is known as the margin of the classifier

Decision Boundary

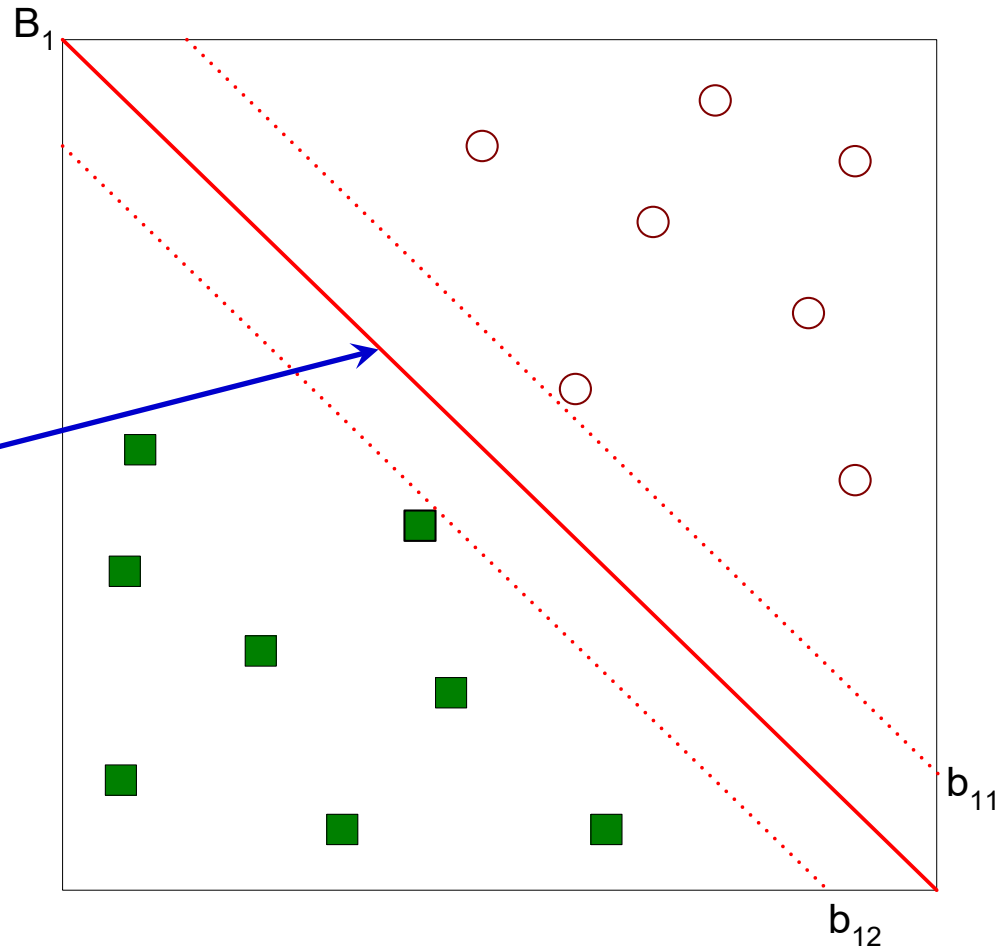
- Given a binary classification task, denote $y_i = +1$ the circle class, and $y_i = -1$ the square class

Decision boundary:

$$w_1x_1 + w_2x_2 + b = 0$$

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

Inner product: $\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^D w_i x_i$



Review: Inner Products of Vectors

- We use bold letters to denote vectors, such as \mathbf{a} and \mathbf{b} .
- A vector can have many dimensions: $\mathbf{a} = (a_1, a_2, \dots, a_D)$
- The inner product of two D -dimensional vectors (D -vectors), \mathbf{a} and \mathbf{b} is defined as

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \dots + a_D b_D = \sum_i a_i b_i$$

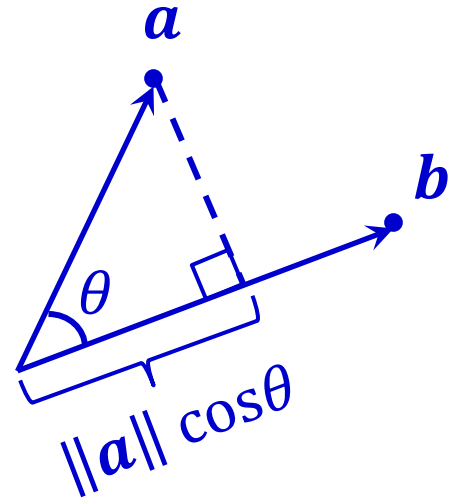
- Also, $\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$, where θ is the angle between \mathbf{a} and \mathbf{b}
- $\|\mathbf{a}\|$ is the Euclidean norm of \mathbf{a}

$$\|\mathbf{a}\| = \sqrt{a_1 a_1 + a_2 a_2 + \dots + a_D a_D} = \sqrt{\mathbf{a} \cdot \mathbf{a}}$$

Review: Geometry of Inner Products

$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$, where θ is the angle between \mathbf{a} and \mathbf{b}

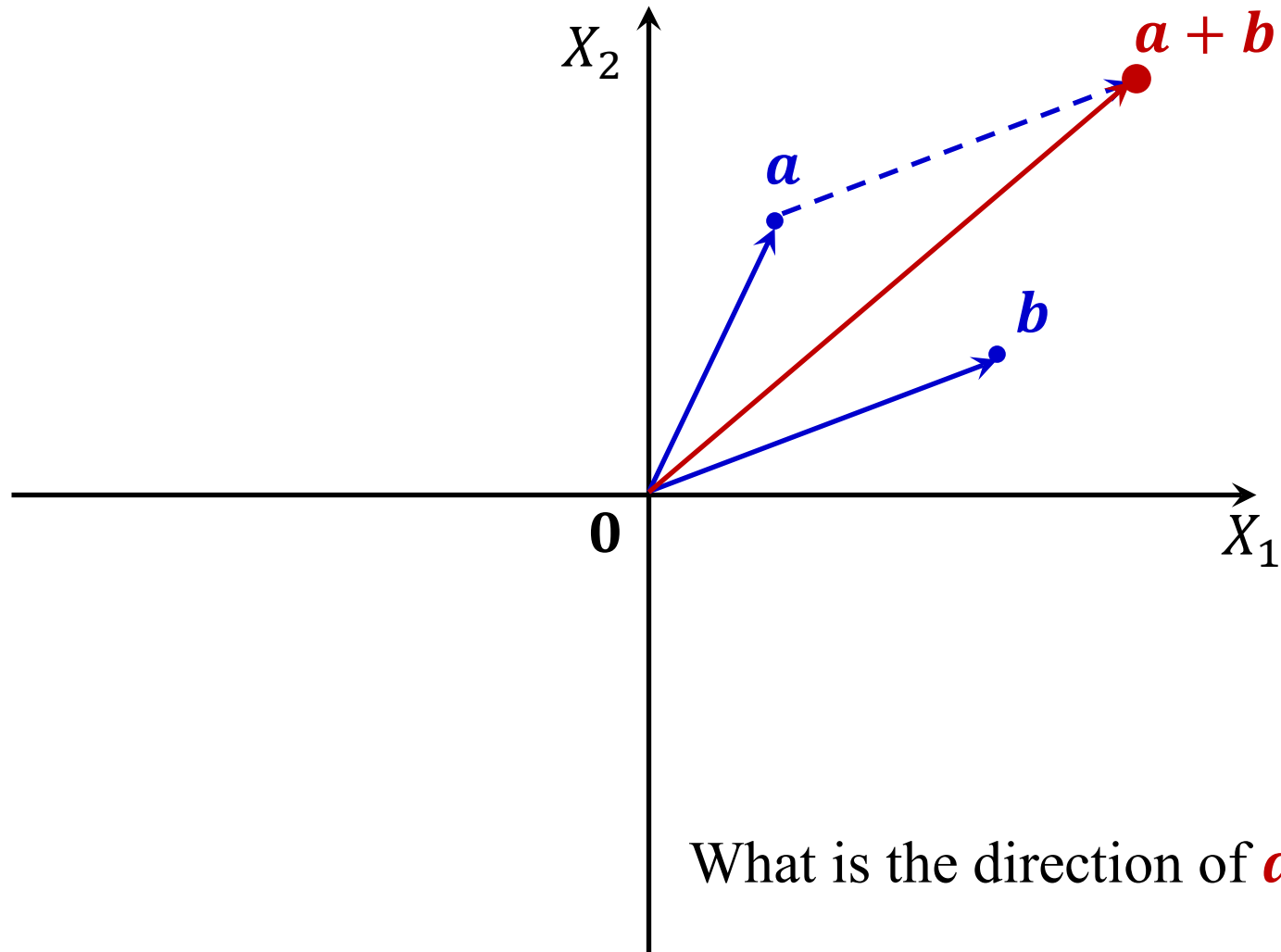
$\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|} = \|\mathbf{a}\| \cos \theta$ is the length of the projection of \mathbf{a} on (or onto) \mathbf{b}



What if \mathbf{a} and \mathbf{b} form an obtuse angle?

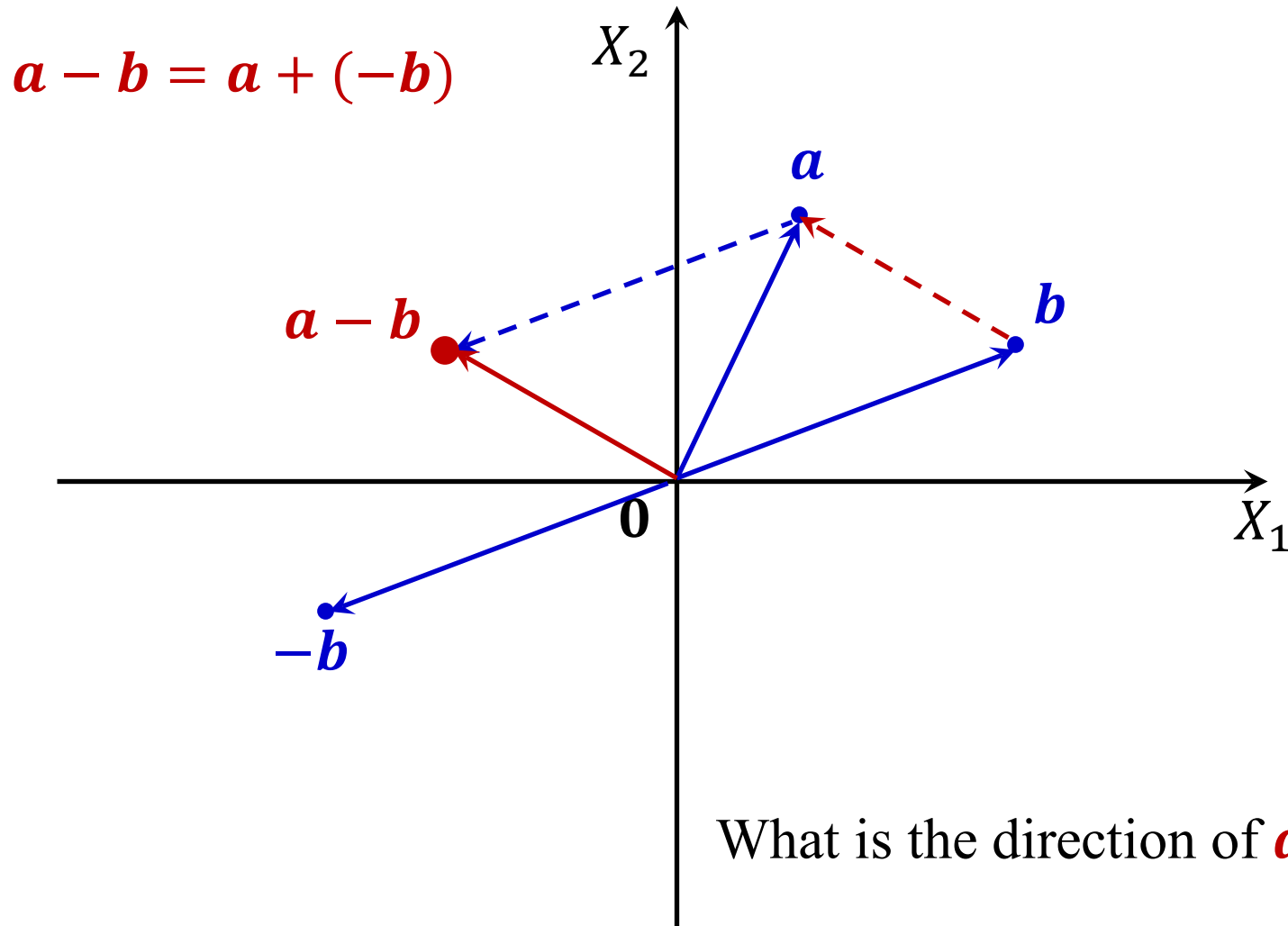
If \mathbf{a} and \mathbf{b} are orthogonal,
 $\theta = 90^\circ, \cos \theta = 0, \mathbf{a} \cdot \mathbf{b} = 0$

Review: Vector Addition

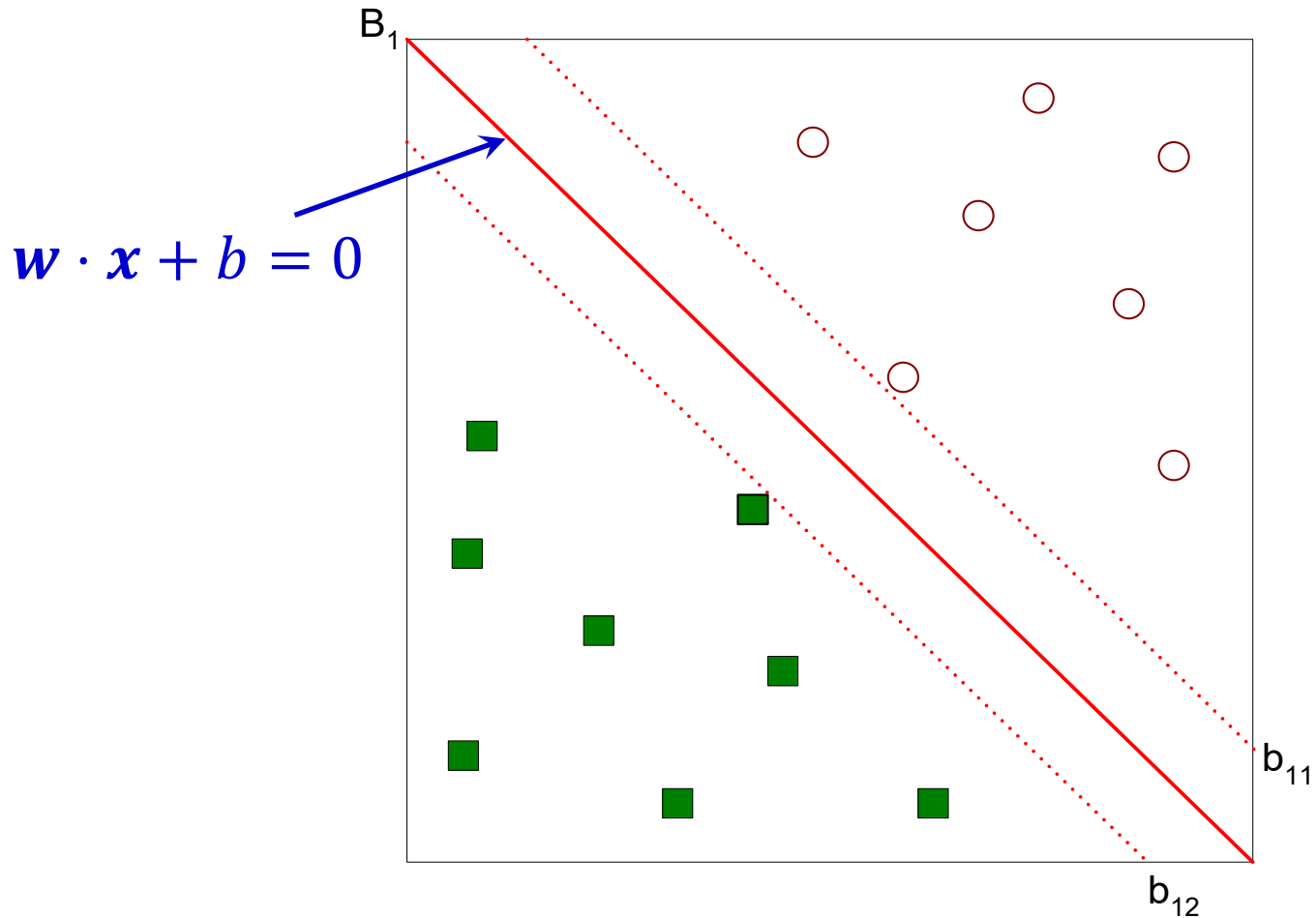


What is the direction of $a + b$?

Review: Vector Subtraction



Making Predictions



For any test example \mathbf{x}^* :

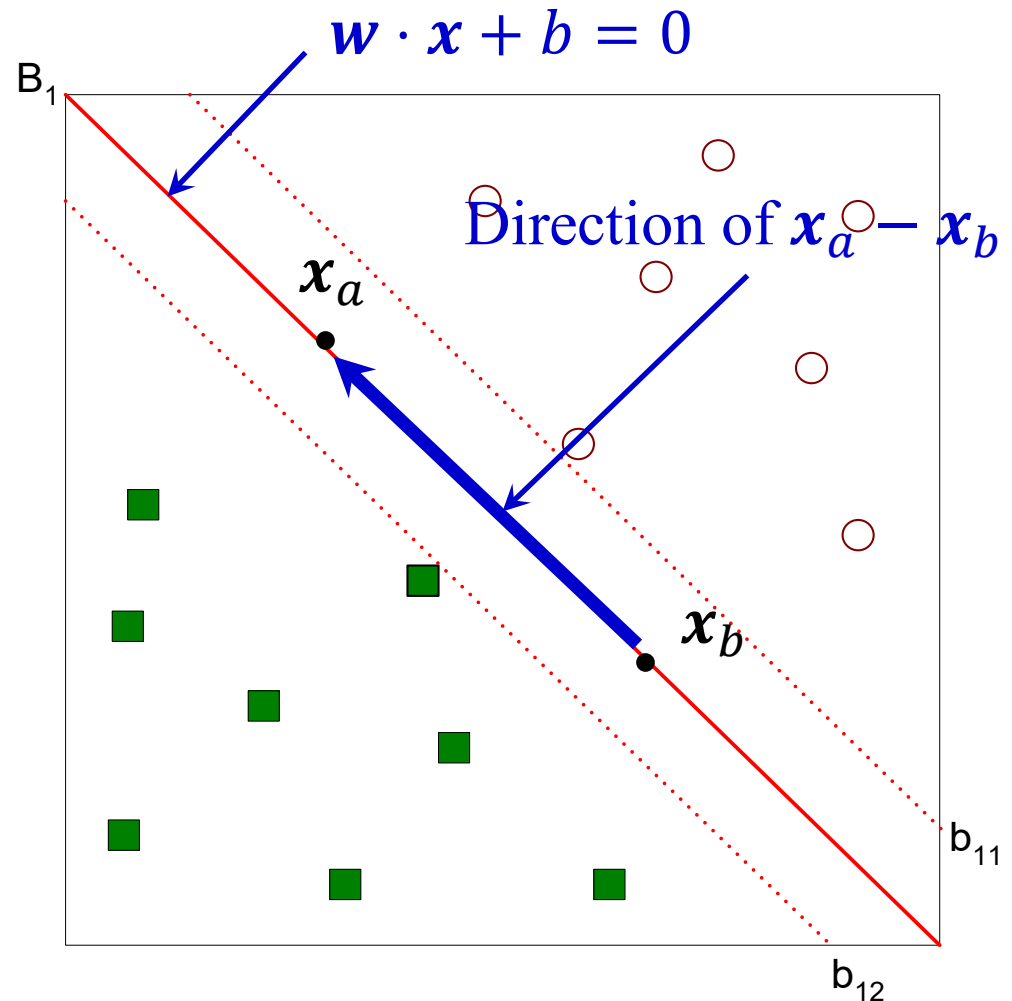
$$\begin{cases} f(\mathbf{x}^*) = +1, & \text{if } \mathbf{w} \cdot \mathbf{x}^* + b \geq 0 \\ f(\mathbf{x}^*) = -1, & \text{if } \mathbf{w} \cdot \mathbf{x}^* + b < 0 \end{cases}$$

Margin – Induction

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

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

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



Margin – Induction (cont.)

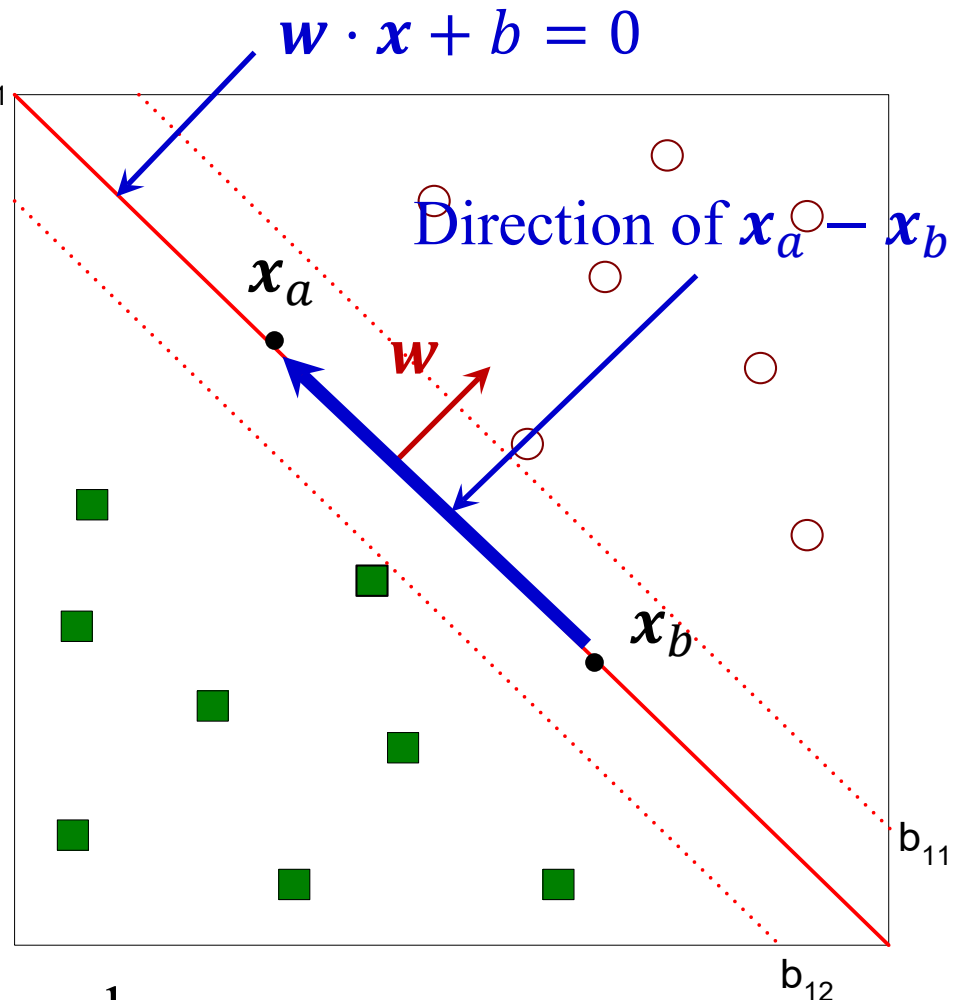
- Suppose \mathbf{x}_a and \mathbf{x}_b are two points located on the decision boundary,

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

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

Based on definition
of inner product

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



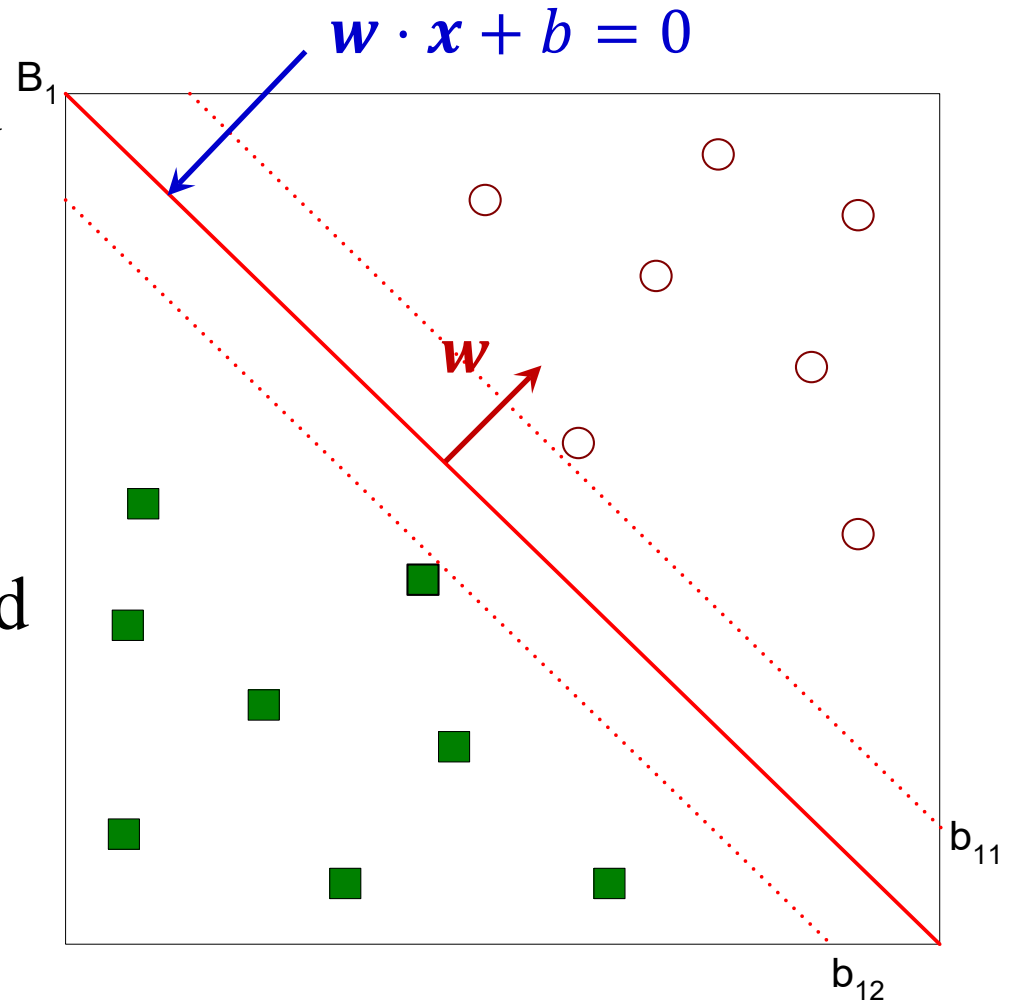
Margin – Induction (cont.)

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

$$\mathbf{w} \cdot \mathbf{x}_c + b \geq k, \\ \text{where } k > 0$$

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

$$\mathbf{w} \cdot \mathbf{x}_s + b \leq k', \\ \text{where } k' < 0$$



Margin – Induction (cont.)

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

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

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

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

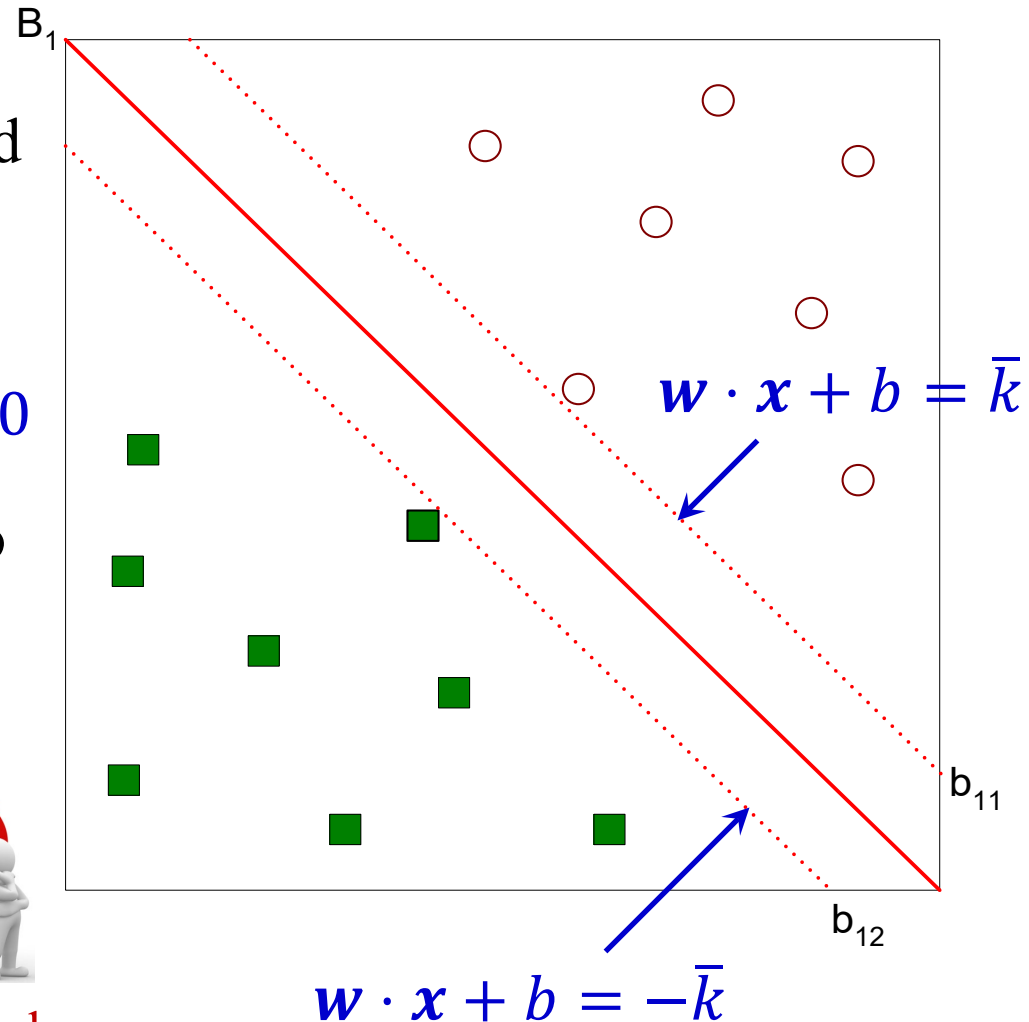
$$\mathbf{w} \cdot \mathbf{x} + \bar{b} = \bar{k}$$

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

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



Tutorial



Margin – Induction (cont.)

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

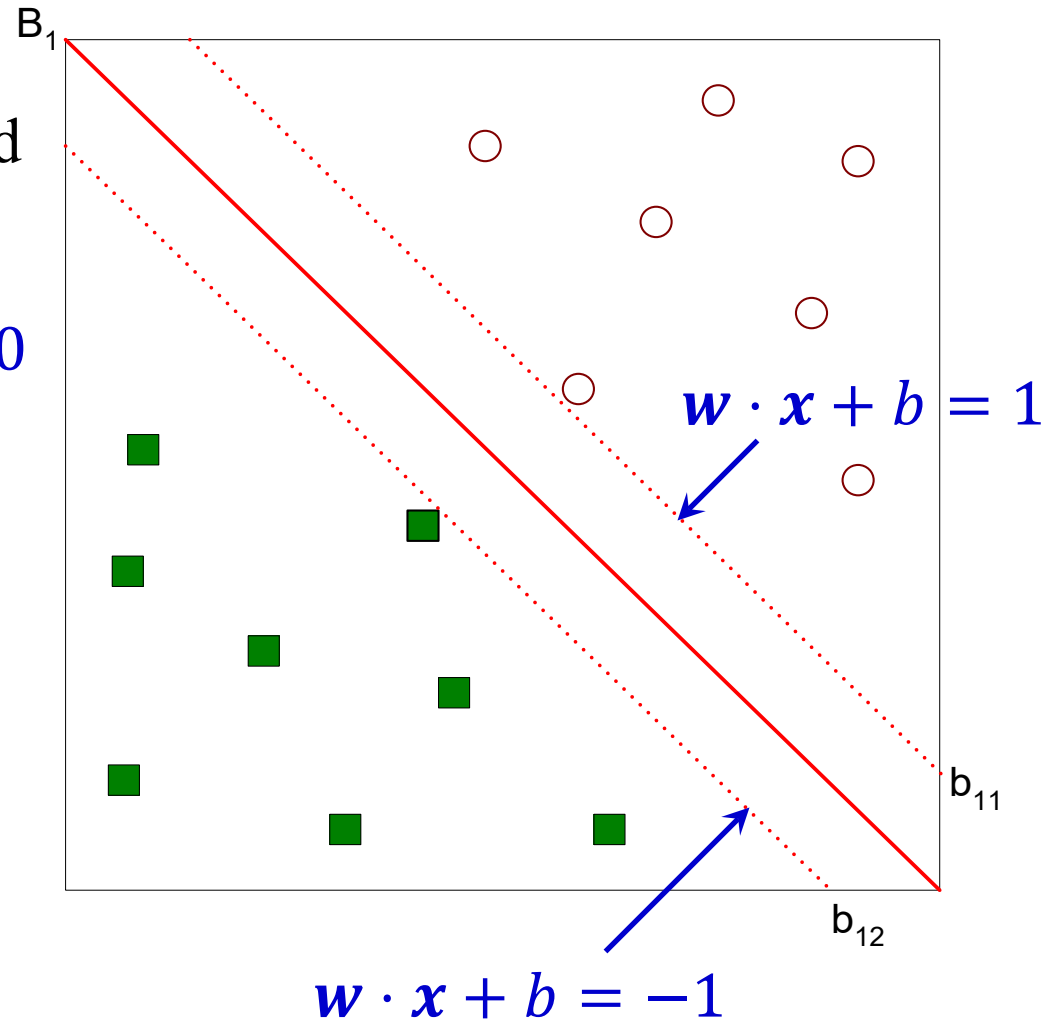
$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} + \bar{b} &= \bar{k} \\ \mathbf{w} \cdot \mathbf{x} + \bar{b} &= -\bar{k} \end{aligned} \quad \text{where } \bar{k} > 0$$

$$\mathbf{w} = \frac{\mathbf{w}}{\bar{k}} \quad b = \frac{\bar{b}}{\bar{k}}$$

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

$$\mathbf{w} \cdot \mathbf{x} + b = 1$$

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



Margin – Induction (cont.)

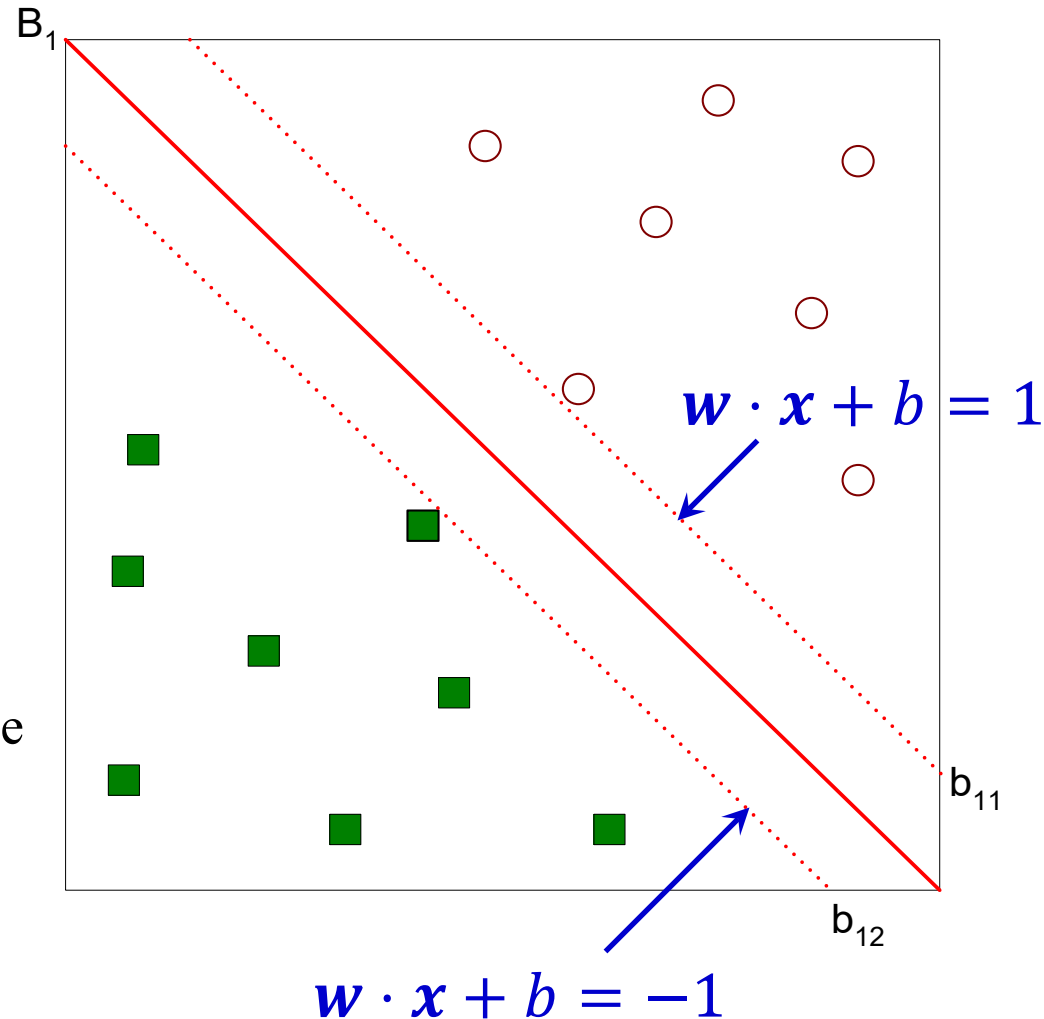
$$\mathbf{w} = \frac{\mathbf{w}}{\bar{k}} \quad b = \frac{\bar{b}}{\bar{k}}$$

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

$$\mathbf{w} \cdot \mathbf{x} + b = 1$$

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

We should use different letters for the rescaled \mathbf{w} and b , but we abuse notations for simplicity.



Margin – Induction (cont.)

$$b_{11}: \mathbf{w} \cdot \mathbf{x}_1 + b = 1$$

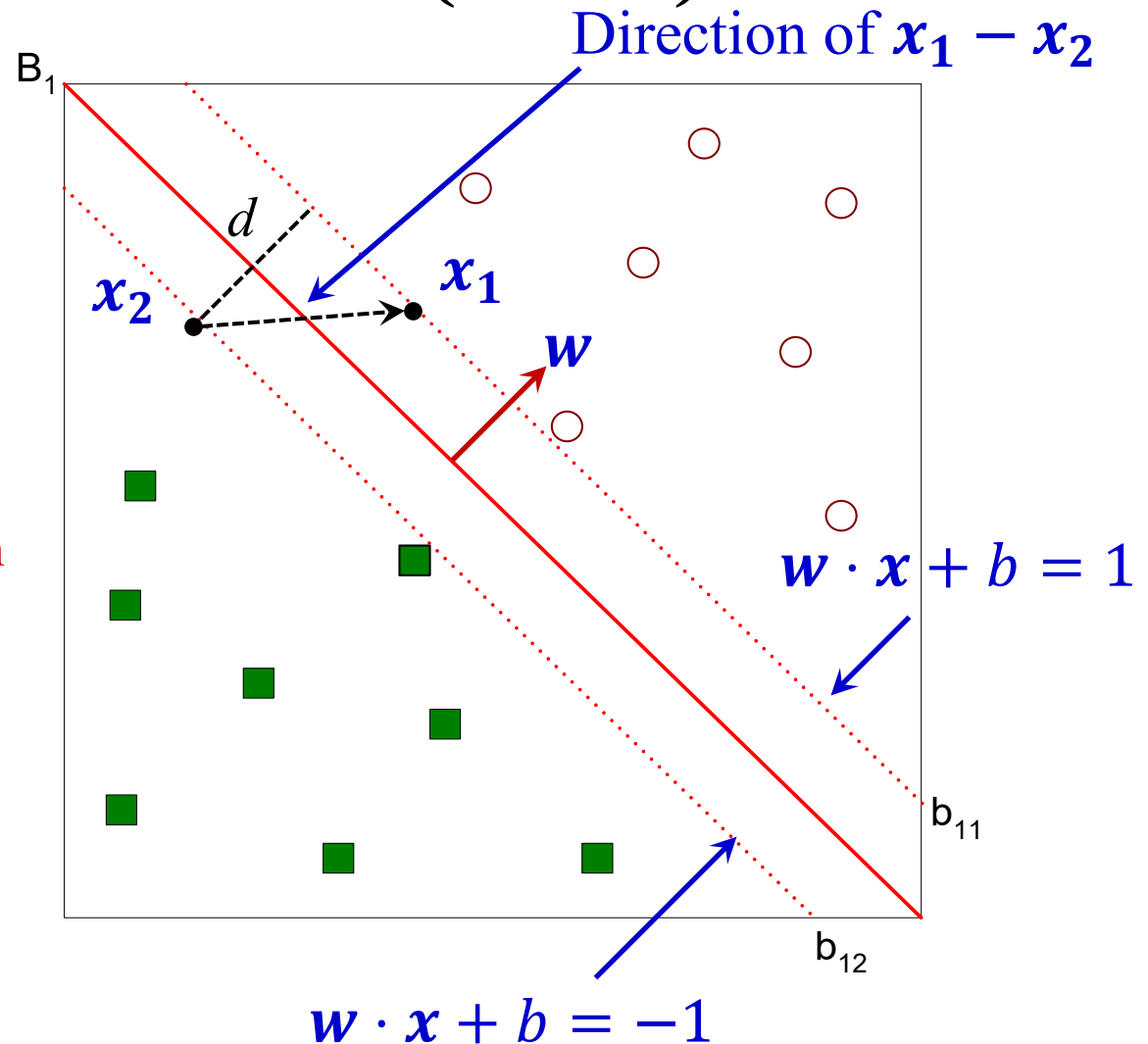
$$b_{12}: \mathbf{w} \cdot \mathbf{x}_2 + b = -1$$

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

Based on definition
of inner product

$$\|\mathbf{w}\|_2 \boxed{d} = 2$$

Margin



Margin – Induction (cont.)

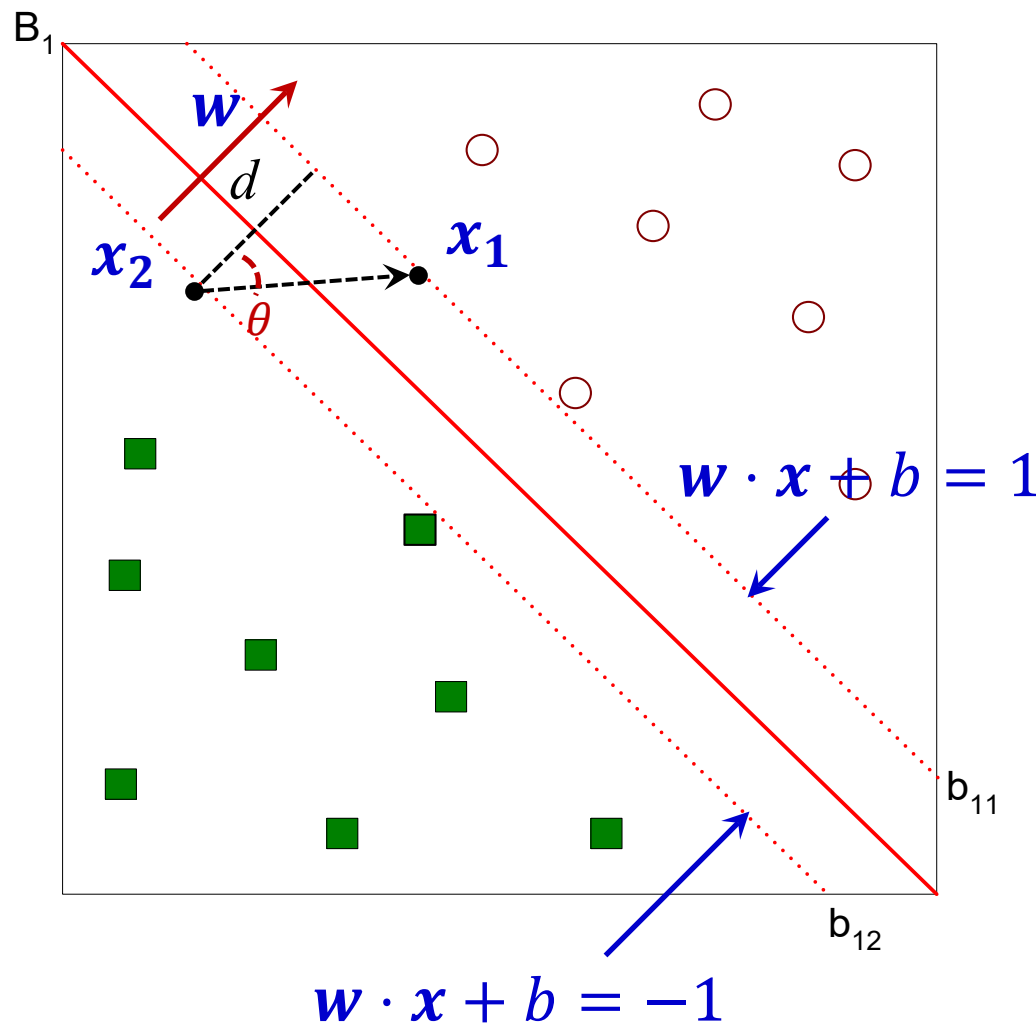
$$w \cdot (x_1 - x_2) = 2$$

Based on definition
of inner product

$$\|w\|_2 \times \|x_1 - x_2\|_2 \times \cos(\theta) = 2$$

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


$$\|w\|_2 \times d = 2$$



Important: this holds only
for the rescaled w .

Margin Maximization

$$\|\mathbf{w}\|_2 \times d = 2 \implies d = \frac{2}{\|\mathbf{w}\|_2} \quad \|\mathbf{w}\|_2^2 = \sum_{i=1}^d (w_i \times w_i)$$

$$\text{Maximize margin } d = \frac{2}{\|\mathbf{w}\|_2} \implies \text{Minimize } \frac{\|\mathbf{w}\|_2^2}{2}$$


For convenience
in computation

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

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

OR

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

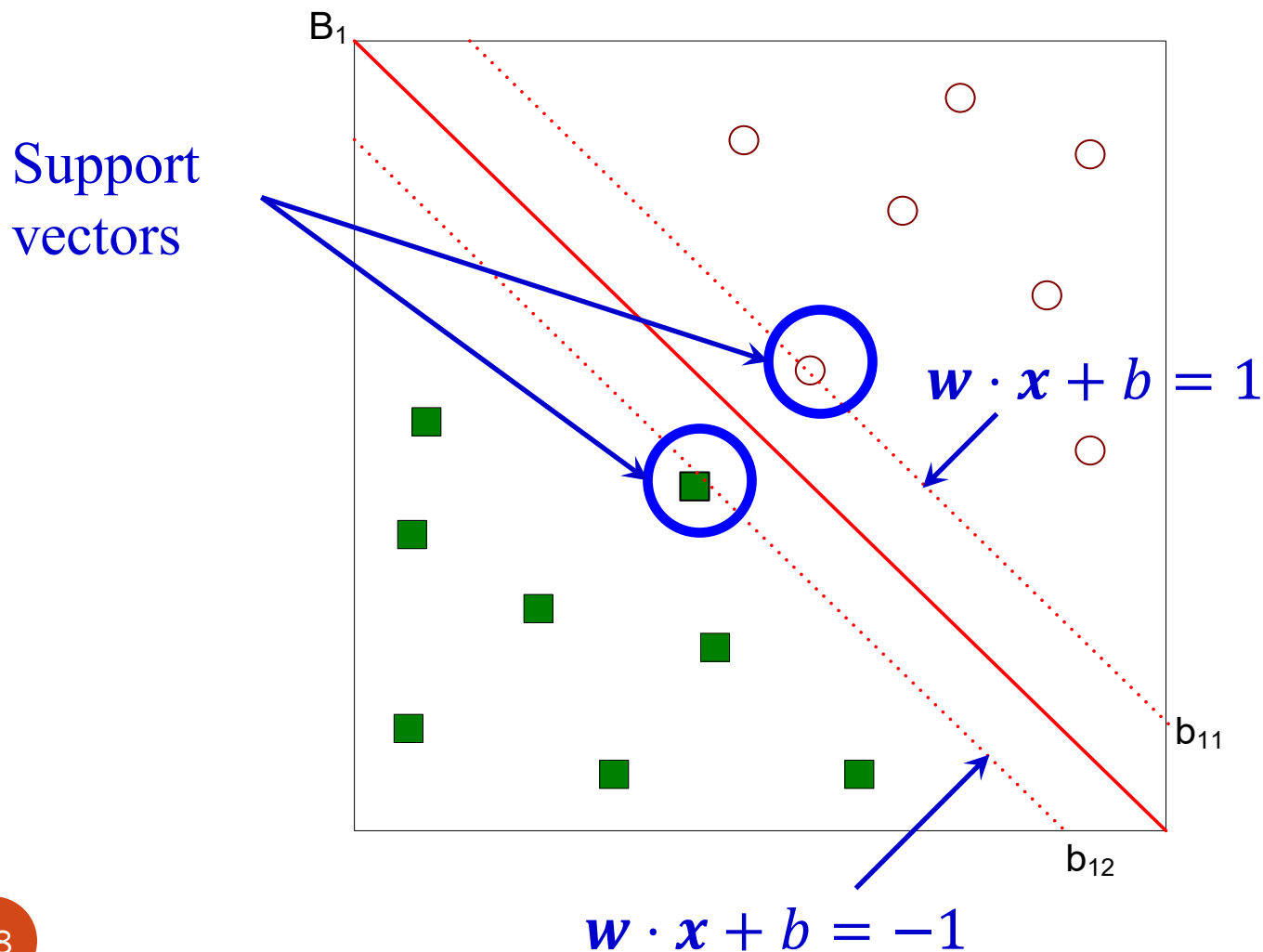
Optimization Problem for SVMs

- Optimization problem of linear SVMs

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

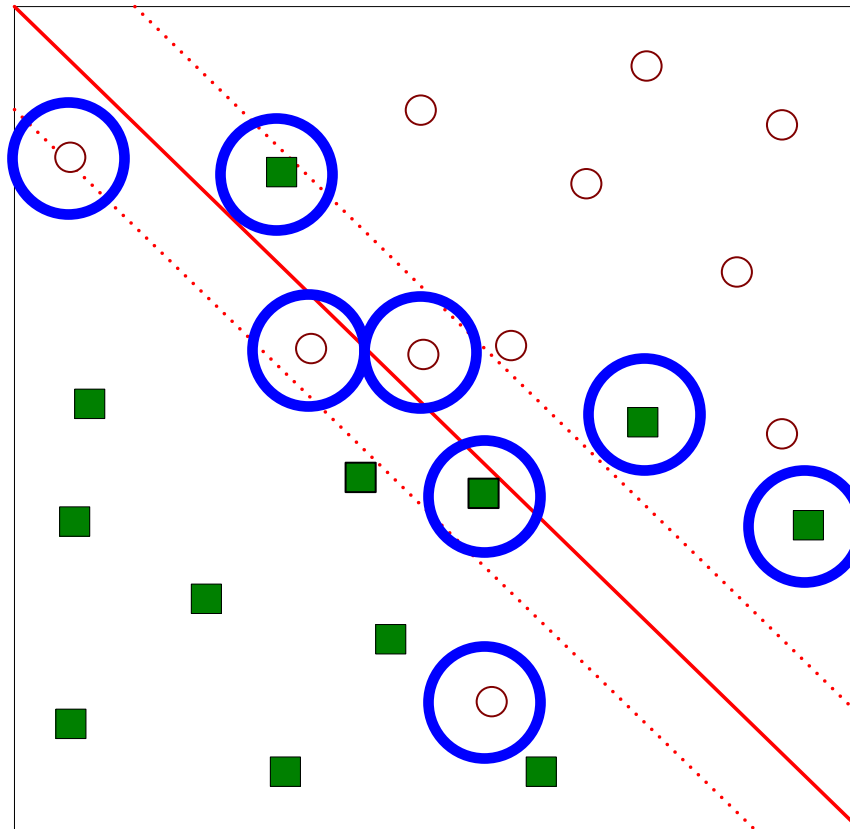
- The optimization is convex
 - Many numerical approaches can be applied to find the solution
 - Convex is easy. Non-convex is hard.
 - The exact optimization algorithms are beyond the scope of this course.

Support Vectors



Non-separable Case

- 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
```

...

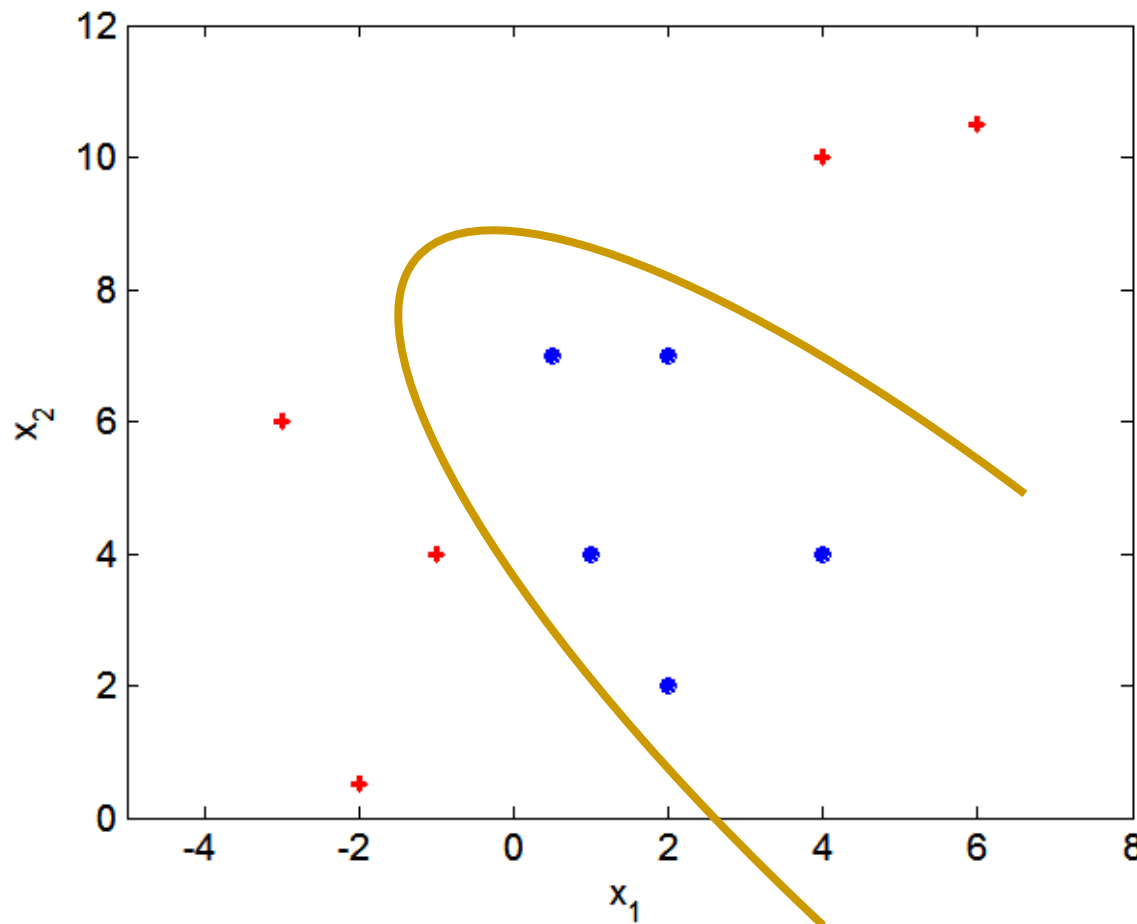
```
>>> svmC = svm.LinearSVC()
```

```
>>> svmC.fit(X, y)
```

```
>>> pred= svmC.predict(X)
```

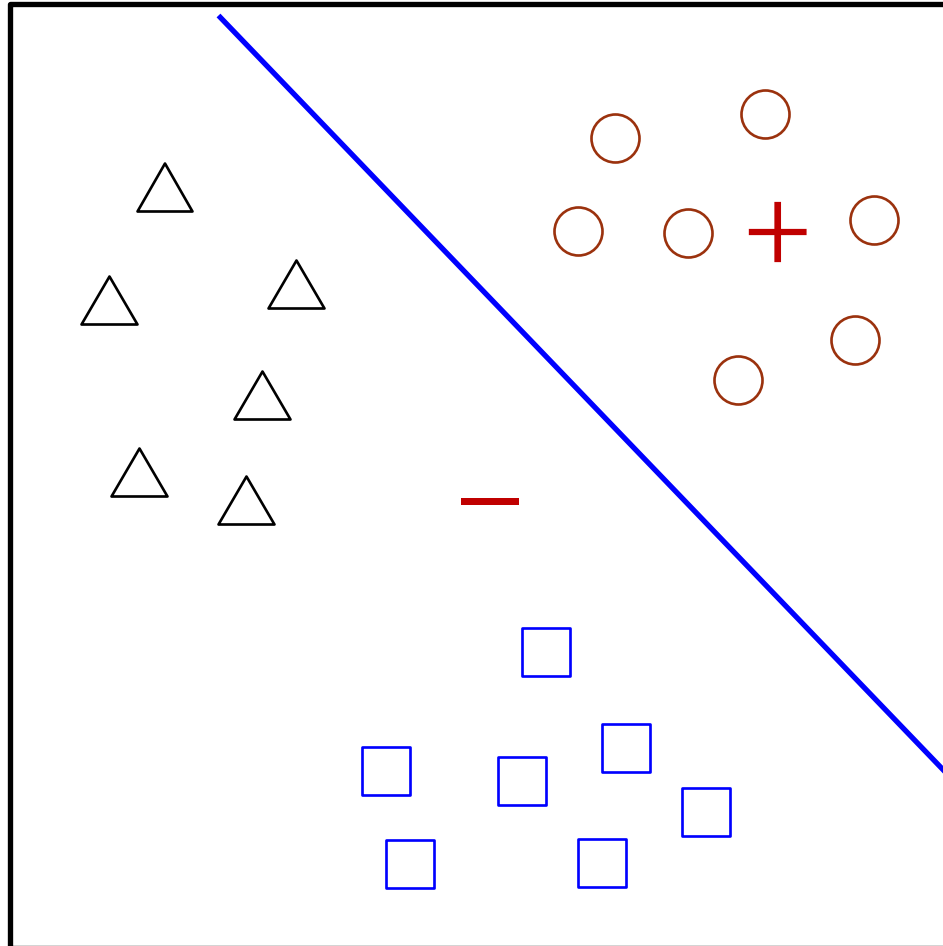
Nonlinear SVMs

- What if decision boundary is not linear?



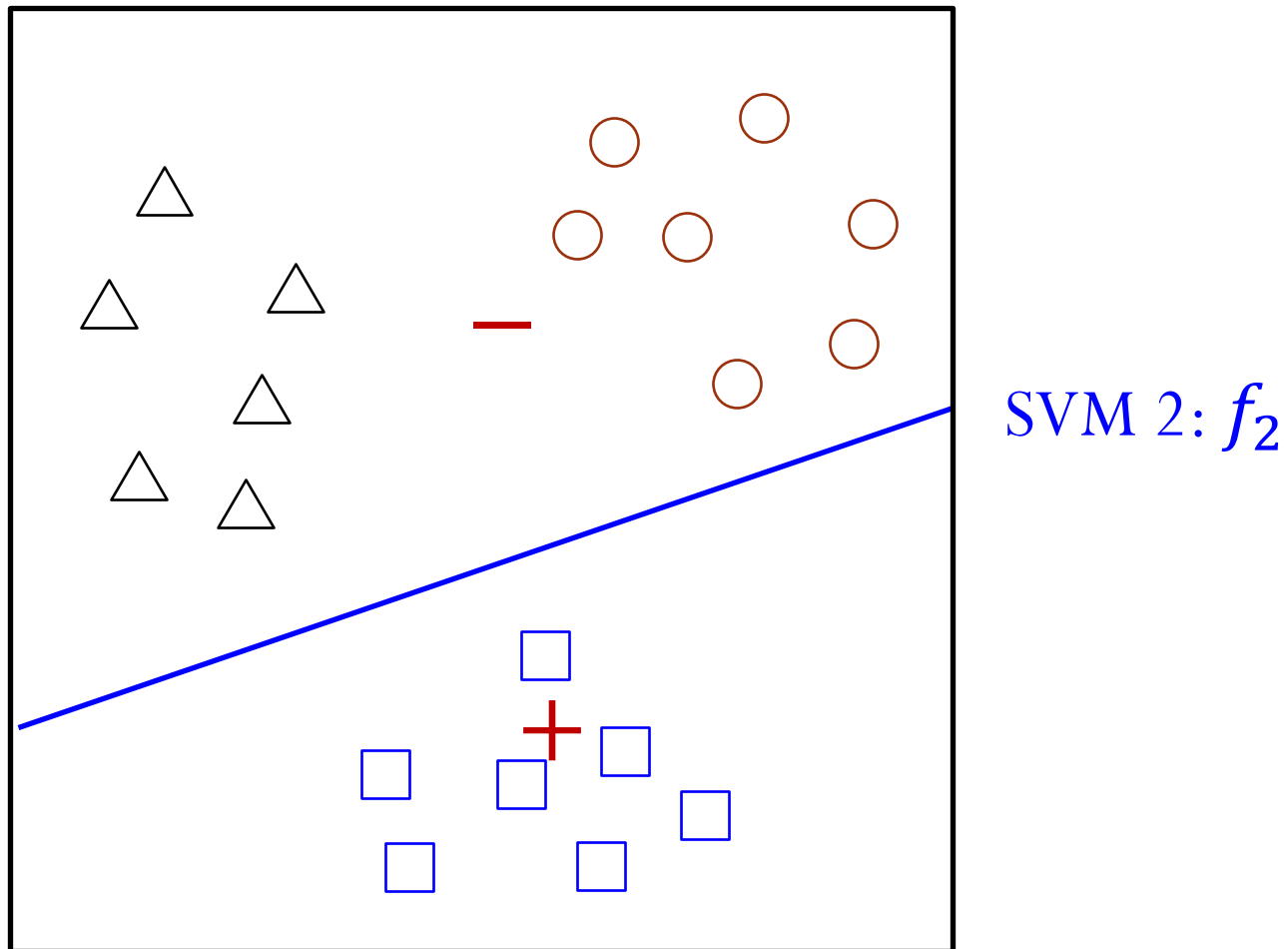
Kernel trick in
the dual form

Multi-Class Classification

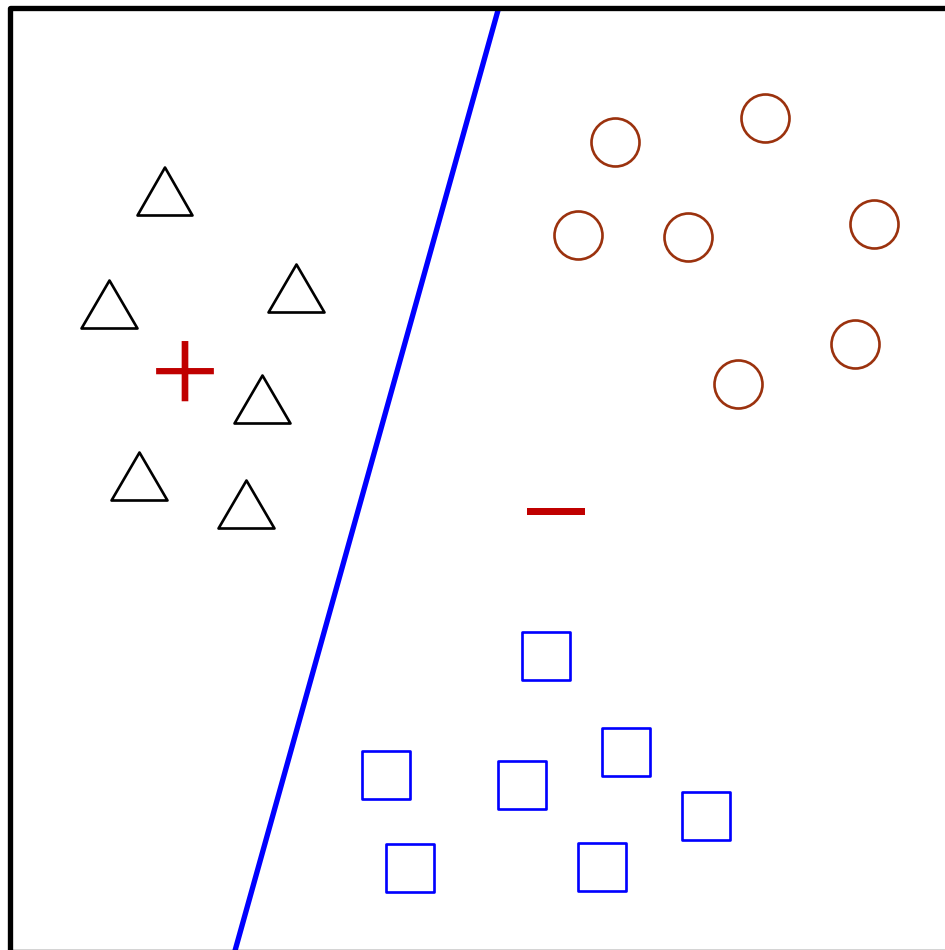


SVM 1: f_1

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 \mathbf{x}^* , apply binary classifier f_1 , f_2 , and f_3 to make predictions on \mathbf{x}^*
 - Combine predicted results of $f_1(\mathbf{x}^*)$, $f_2(\mathbf{x}^*)$, and $f_3(\mathbf{x}^*)$ 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 \mathbf{x}^* , suppose

$$f_1(\mathbf{x}^*) = -1$$

$$f_2(\mathbf{x}^*) = 1$$

$$f_3(\mathbf{x}^*) = -1$$

Total Votes:

C_1	C_2	C_3
0	1	1
0	1	0
1	1	0
1	3	1



Thank you!