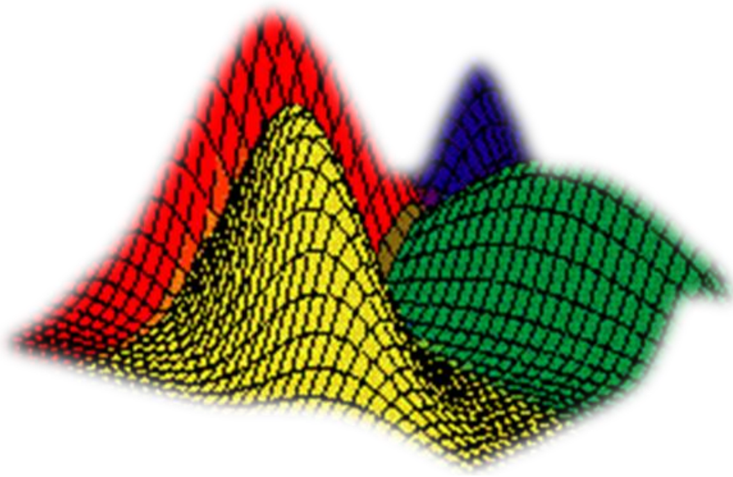
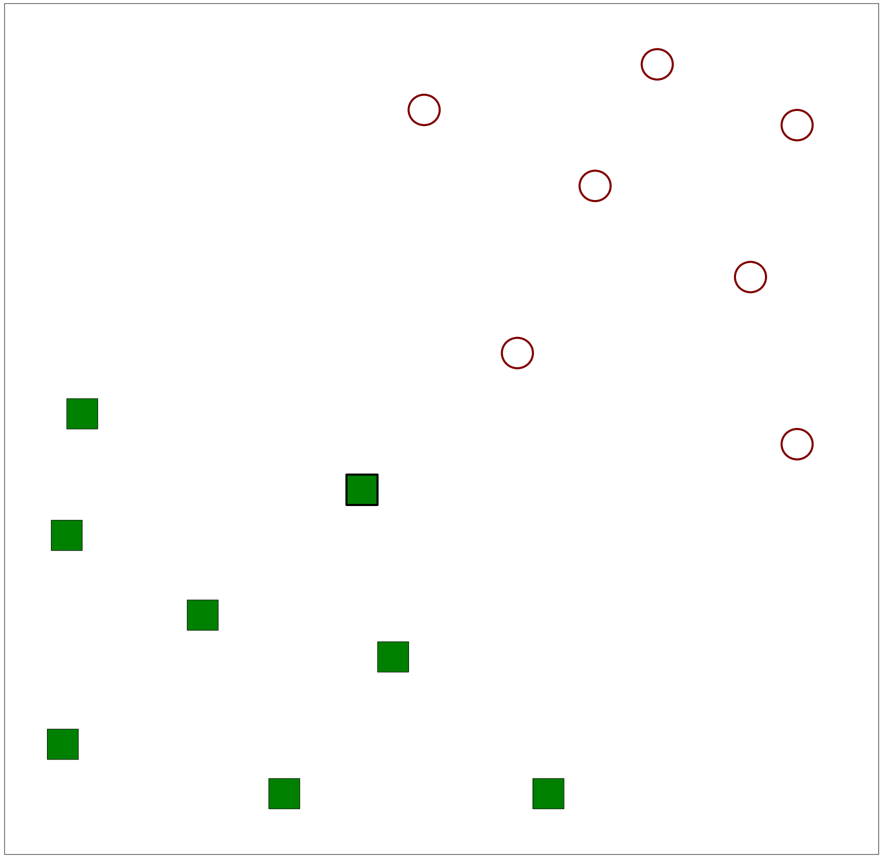


CSE 473
Pattern Recognition

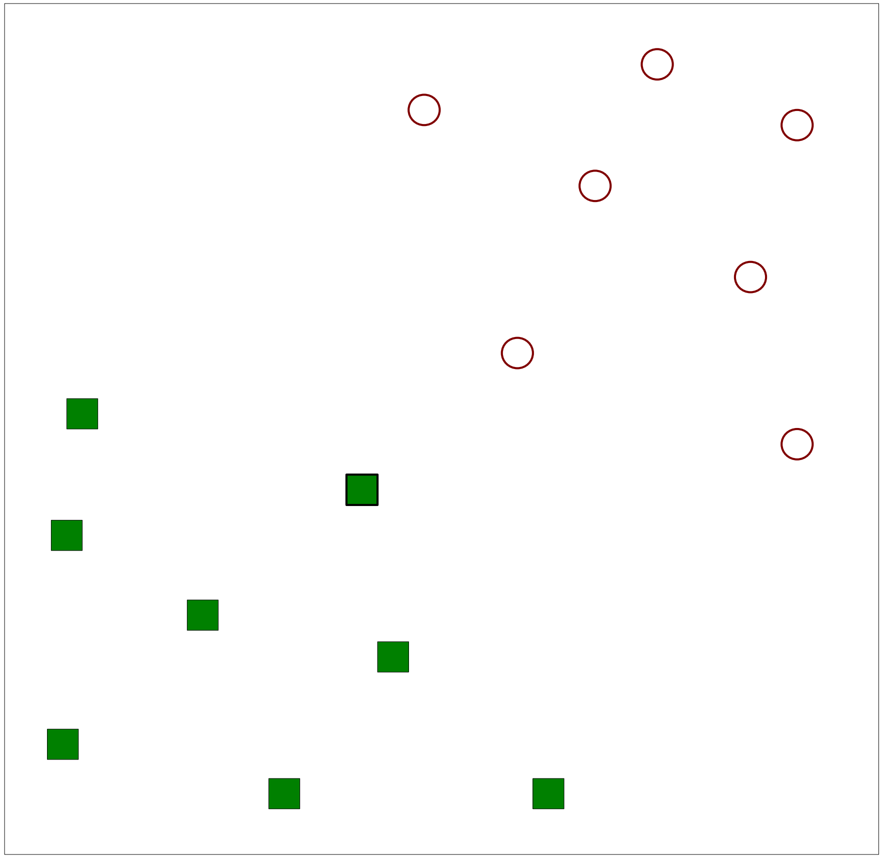
Two-Class case *again*

Let, we have N training samples



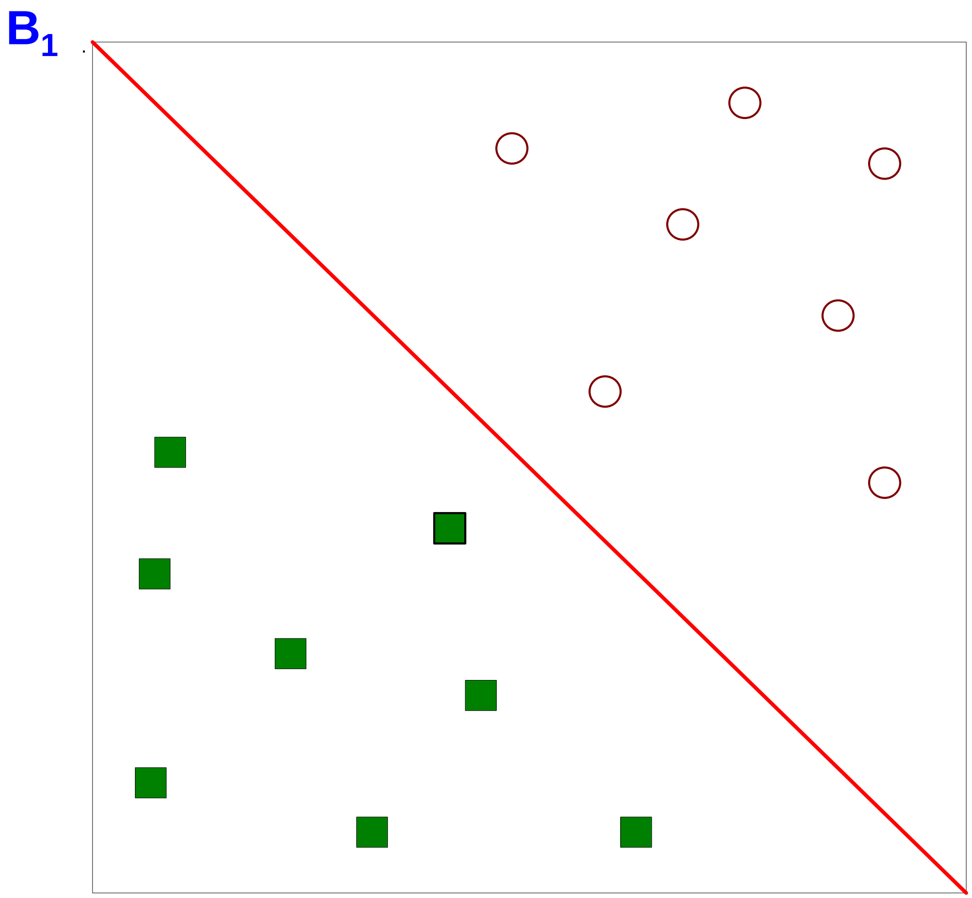
Two-Class case *again*

Let, we have N training samples



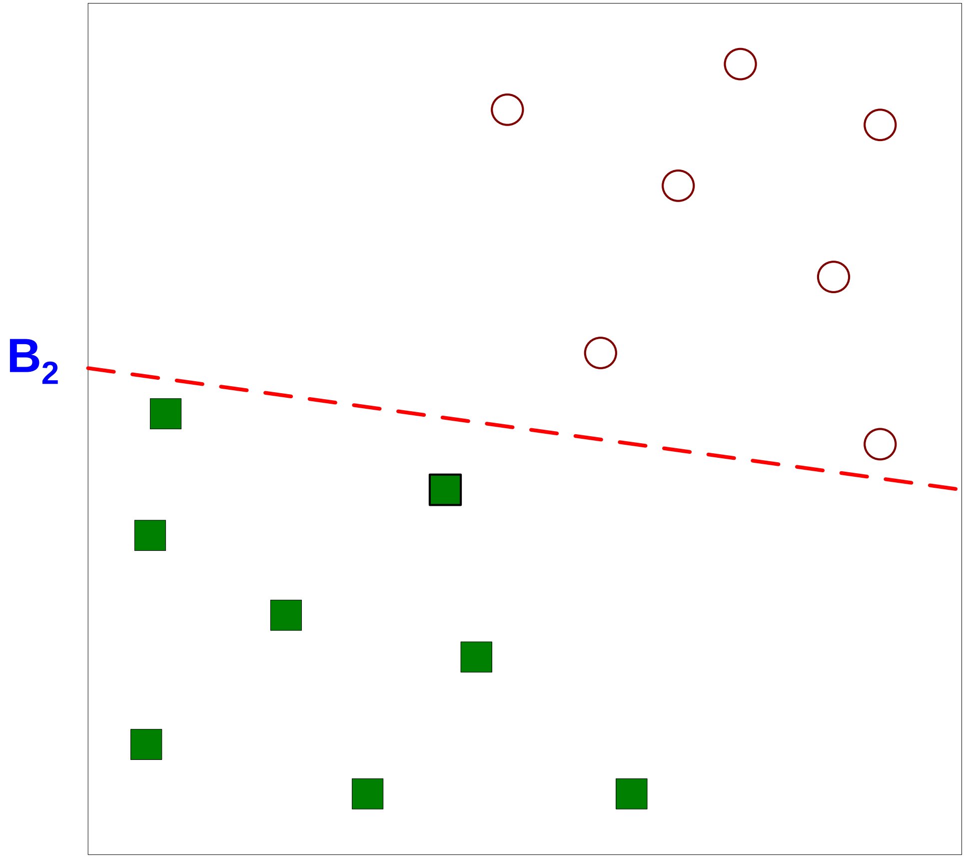
- Find a linear hyperplane (decision boundary) that will separate the data

Two-Class case again



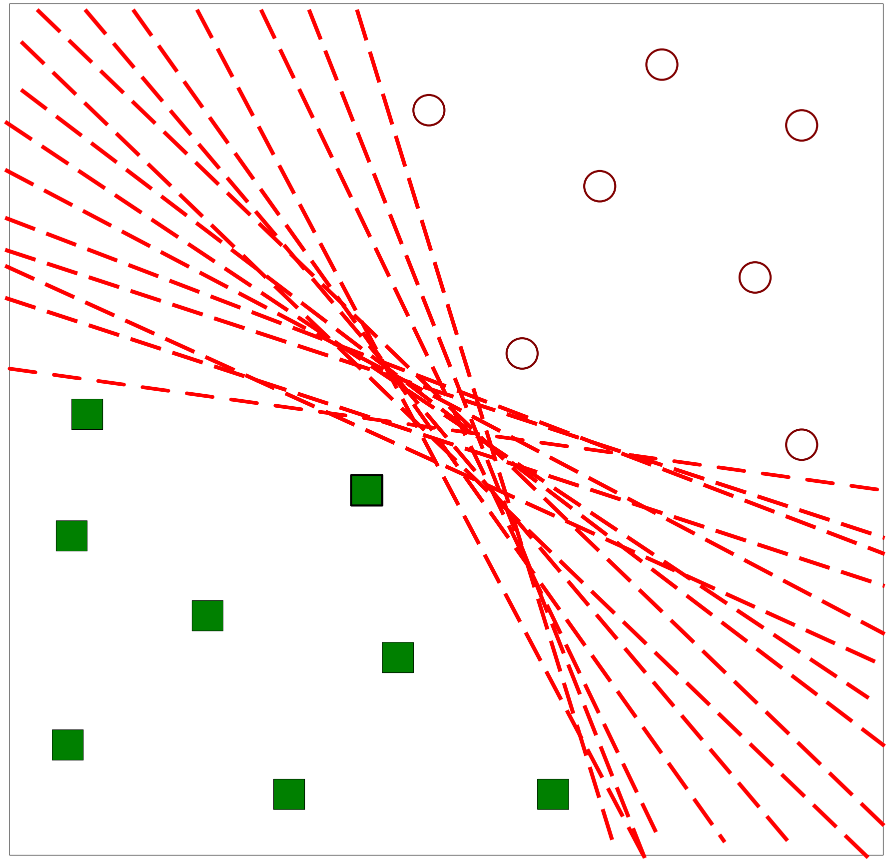
- One Possible Solution

Two-Class case again



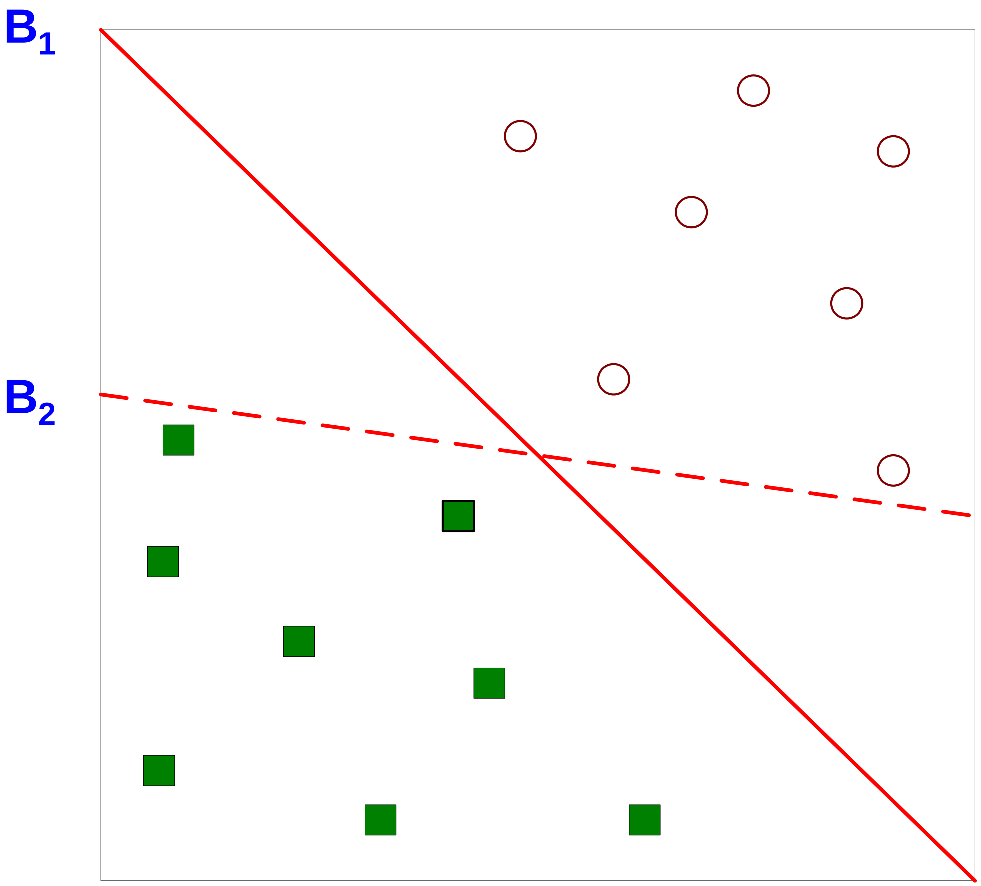
- Another possible solution

Two-Class case again



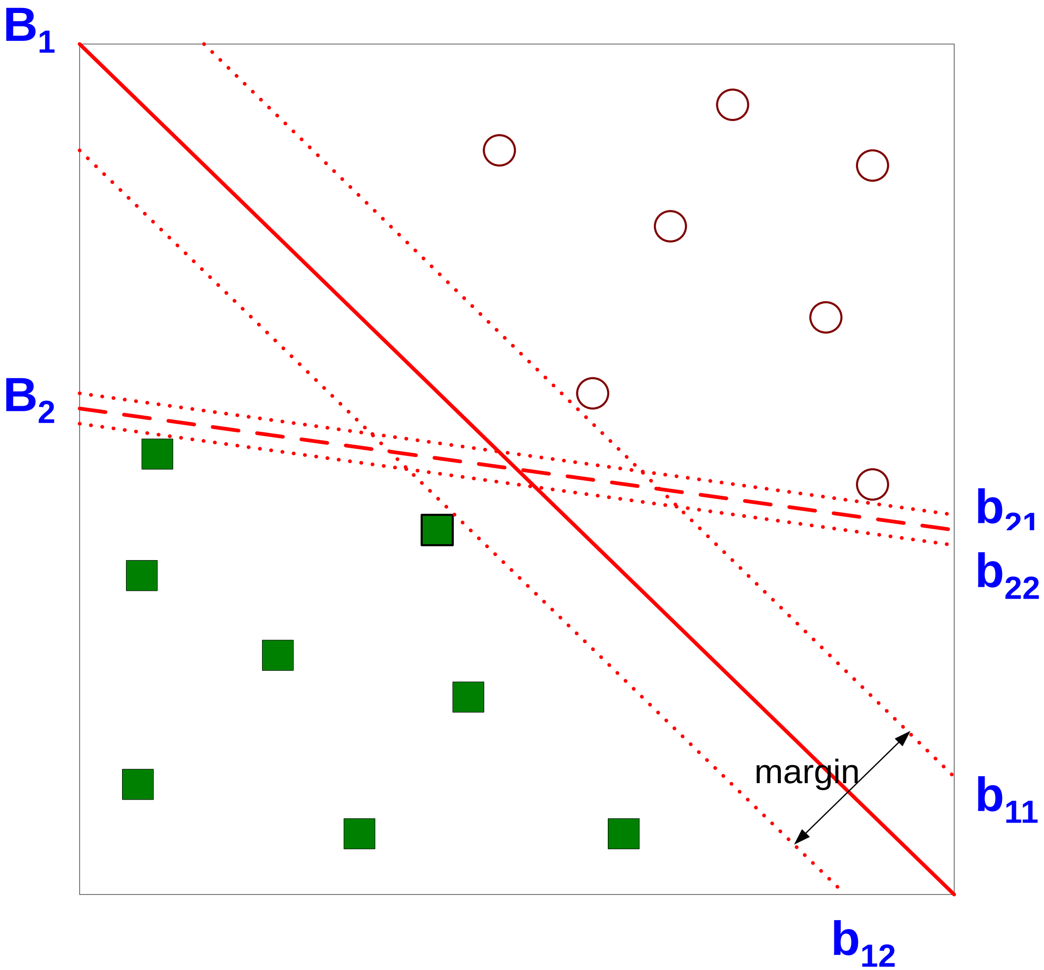
- Other possible solutions

Two-Class case again



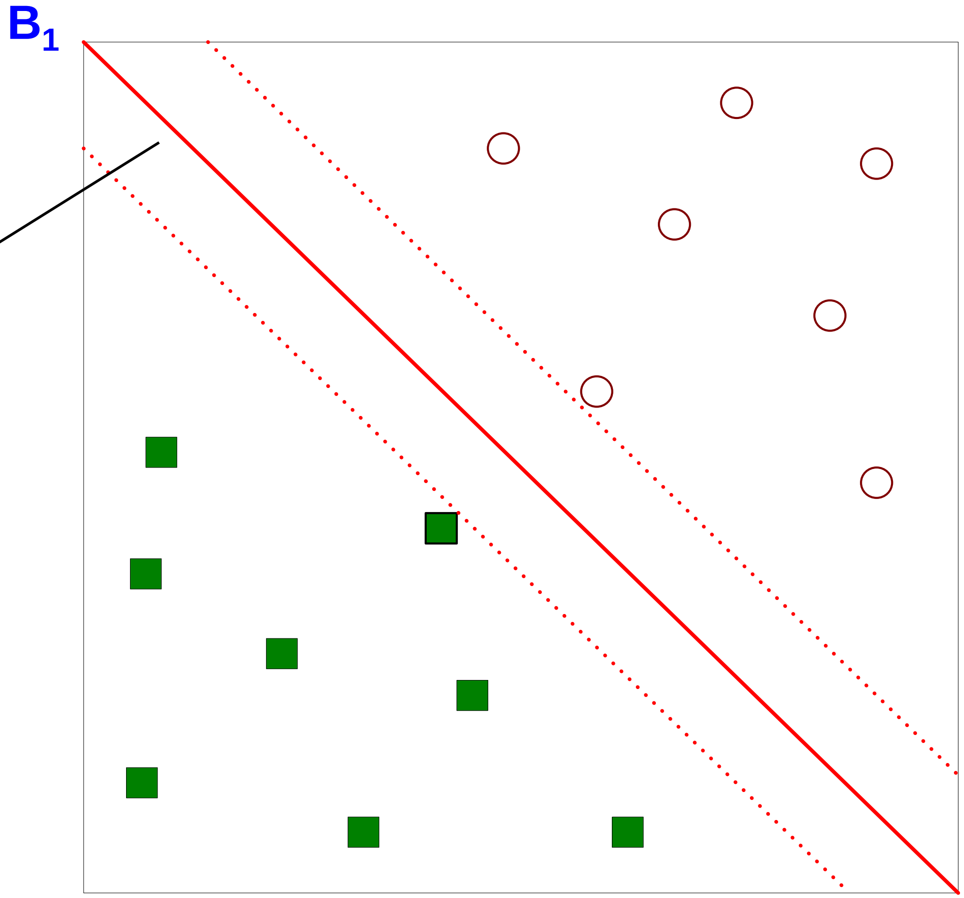
- Which one is better? B_1 or B_2?
- How do you define better?

Two-Class case again



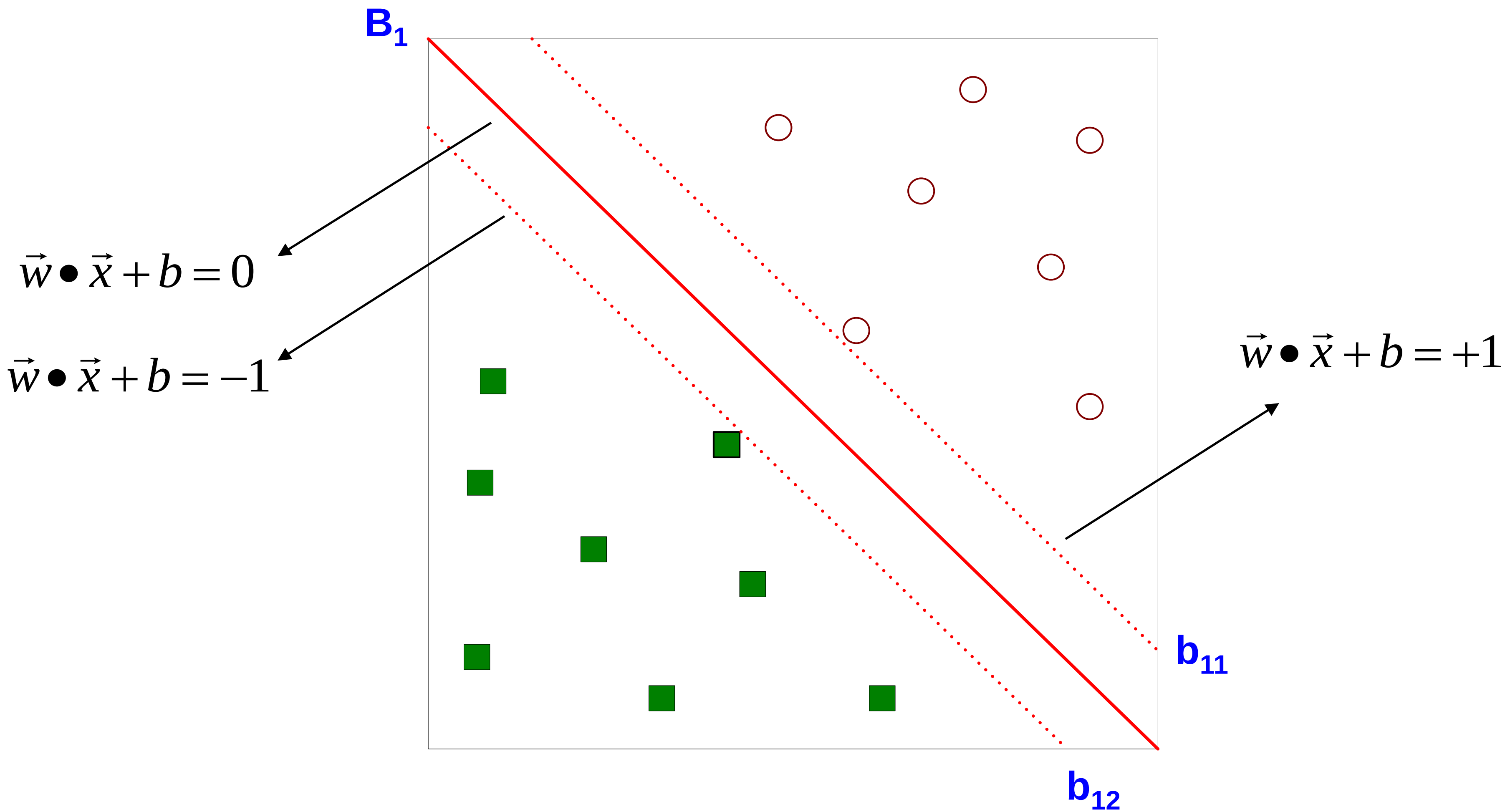
- Find hyperplane maximizes the margin => B1 is better than B2

Two-Class case again

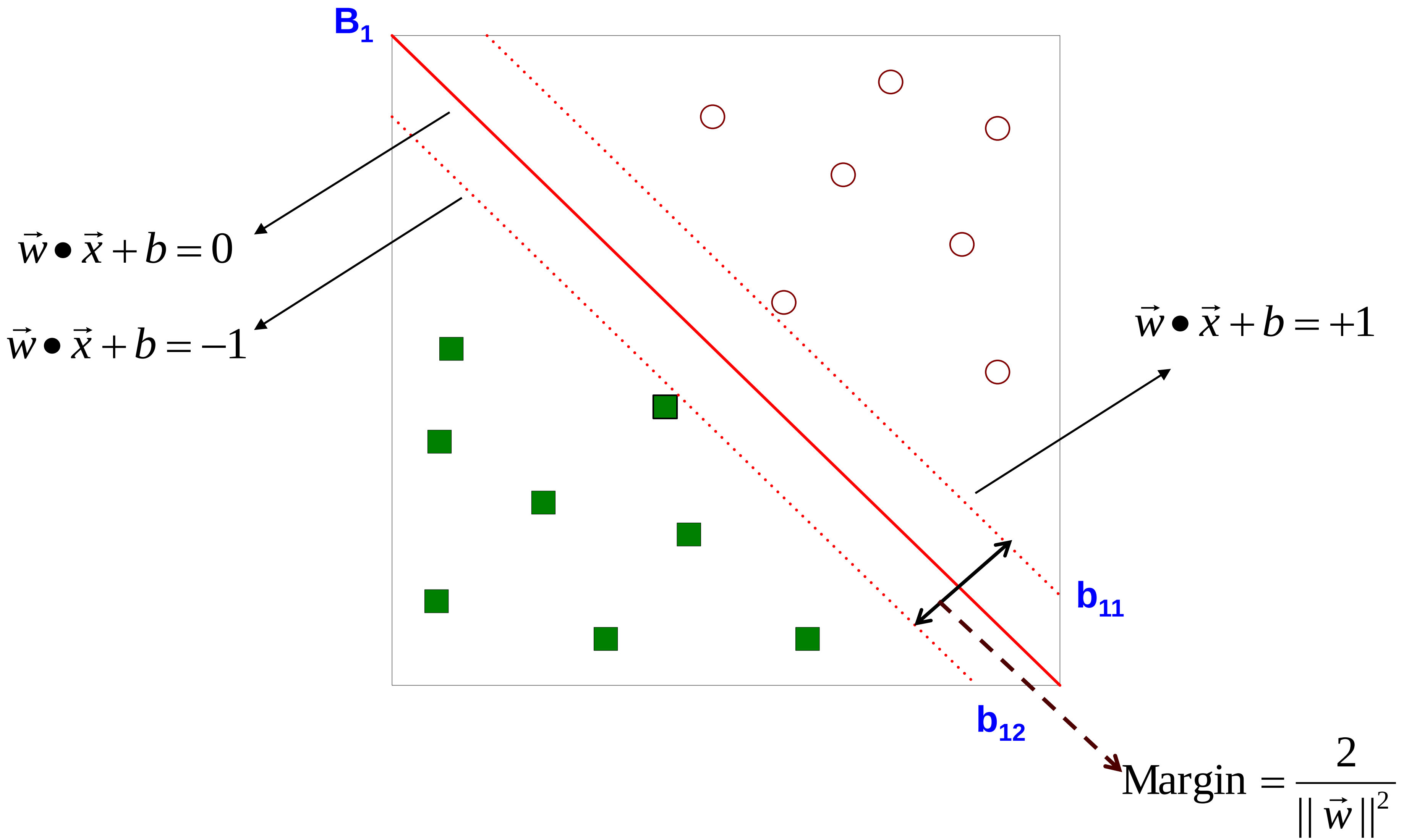


$$\vec{w} \bullet \vec{x} + b = 0$$

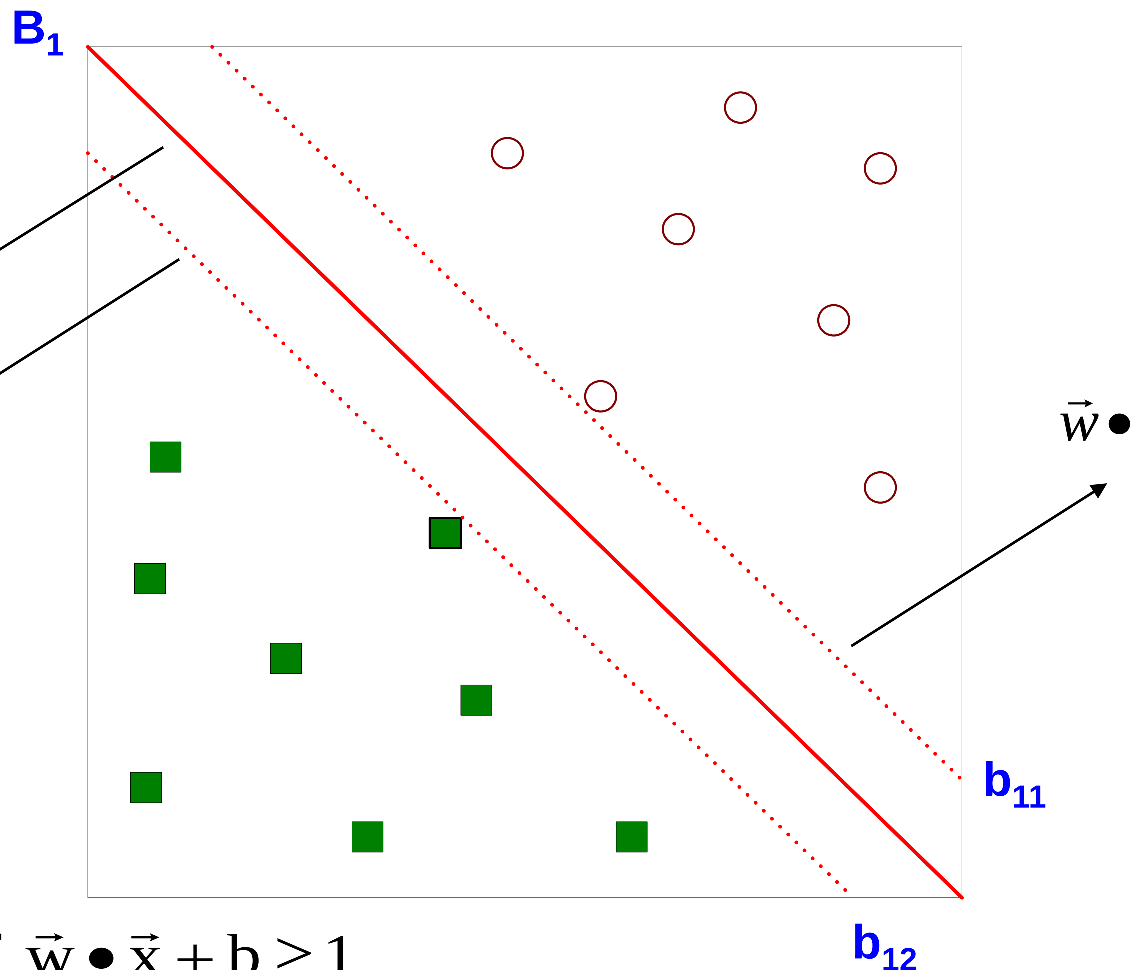
Two-Class case again



Two-Class case again



Two-Class case again



$$\vec{w} \bullet \vec{x} + b = 0$$

$$\vec{w} \bullet \vec{x} + b = -1$$

$$\vec{w} \bullet \vec{x} + b = +1$$

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$

Support Vector Machine

- We want to maximize: $$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$
  - subject to the following constraints:

$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$$

Support Vector Machine

- We want to maximize: $$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$
  - subject to the following constraints:

$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases} \quad \forall_i$$

Support Vector Machine

- We want to maximize: $\text{Margin} = \dfrac{2}{\|\vec{w}\|^2}$
  - Which is equivalent to minimizing: $L(w) = \dfrac{\|\vec{w}\|^2}{2}$
  - subject to the following constraints:
    - $y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$

Support Vector Machine

The Expression

$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$$

can be written as

$$y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) \geq 1$$

Support Vector Machine

The Expression

$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$$

can be written as

$$y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) \geq 1$$

- We can say :
  - minimize:

$$L(w) = \frac{\|\vec{w}\|^2}{2}$$

  - Subject to:

$$y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) \geq 1 \quad \text{or} \quad y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) - 1 \geq 0$$

Support Vector Machine

- $L(w) = \frac{||\vec{w}||^2}{2}$ is a quadratic equation

- Solving for ***w*** and ***b*** is not easy

Support Vector Machine

- $L(w) = \frac{||\vec{w}||^2}{2}$ is a quadratic equation
- Solving for ***w*** and ***b*** is not easy
- What happens if ***w*** =0?

Support Vector Machine

- $L(w) = \frac{\|\vec{w}\|^2}{2}$ is a quadratic equation
- Solving for ***w*** and ***b*** is not easy
- What happens if ***w*** =0?

Some of $y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) - 1 \geq 0$ may be infeasible

Support Vector Machines

- minimize: $$L(w) = \frac{\|\vec{w}\|^2}{2}$$
- Subject to: $$y_i(\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b}) \geq 1 \qquad \forall_i$$

- Use Lagrange function:

$$L_p = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (w.x_i + b) - 1 \right)$$

Support Vector Machines

- Lagrange function:

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w} . \vec{x}_i + b) - 1 \right)$$

- New constraints are:

$$\frac{\partial L_p}{\partial \vec{w}} = 0$$

$$\frac{\partial L_p}{\partial b} = 0$$

Support Vector Machines

- Lagrange function:

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

- New constraints are:

$$\frac{\partial L_p}{\partial \vec{w}} = 0 \quad \Rightarrow \quad \vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \lambda_i y_i = 0$$

Support Vector Machines

- Lagrange function:

$$L_p = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

- constraints are:

$$\vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

Still not solvable, many variables

Support Vector Machines

- Use Lagrange function:

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

- constraints are:

From Karush-Kuhn_Tucker Transform,

$$\vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\lambda_i \geq 0$$

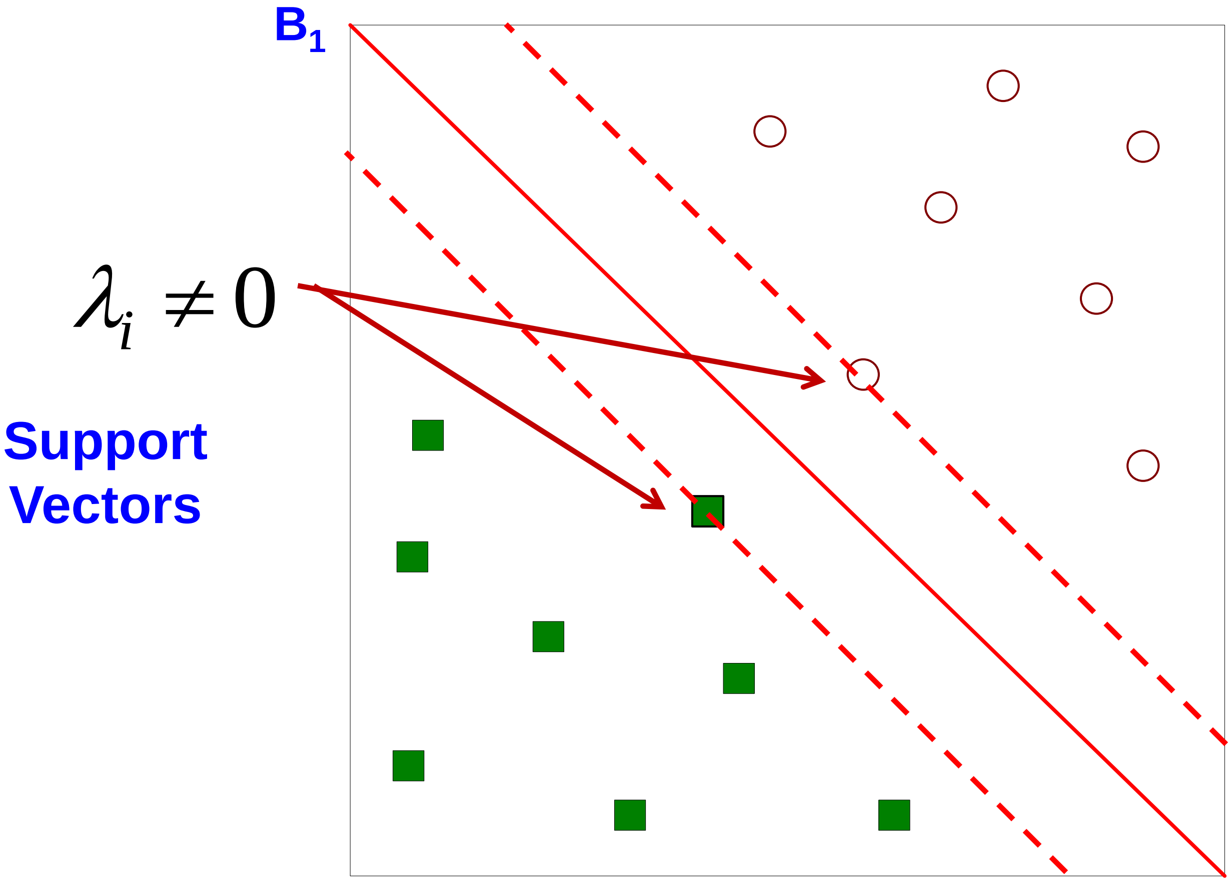
$$\lambda_i \left[y_i (\vec{w}.\vec{x}_i + b) - 1 \right] = 0$$

Support Vector Machines

$$\lambda_i \geq 0 \quad : \text{non-negative}$$

$$\lambda_i \left[y_i (\vec{w}.\vec{x}_i + b) - 1 \right] = 0$$

Support Vector Machines



$$\lambda_i \geq 0$$

$$\lambda_i\left[y_i(\vec{w}.\vec{x}_i + b) - 1\right] = 0$$

Support Vector Machines

- Replace w with λ's in L_p :

put $\vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i$ and $\sum_{i=1}^{N} \lambda_i y_i = 0$

in $L_p = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \big(y_i (\vec{w}.\vec{x}_i + b) - 1 \big)$$

$$= \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$
\begin{aligned}
L_p &= \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \Big(y_i (\vec{w}.\vec{x}_i + b) - 1 \Big) \\
&= \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i \\
&= \frac{\|\vec{w}\|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i
\end{aligned}
$$

$$L_p = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

$$= \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\|\vec{w}\|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\|\vec{w}\|^2}{2} - \vec{w}.\vec{w} - b \times 0 + \sum_{i=1}^{N} \lambda_i$$

$$\vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

$$
\begin{aligned}
L_p &= \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right) \\
&= \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i \\
&= \frac{\|\vec{w}\|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i \\
&= \frac{\|\vec{w}\|^2}{2} - \vec{w}.\vec{w} - b \times 0 + \sum_{i=1}^{N} \lambda_i \\
&= \sum_{i=1}^{N} \lambda_i + \frac{\vec{w}.\vec{w}}{2} - \vec{w}.\vec{w}
\end{aligned}
$$

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

$$= \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\| \vec{w} \|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\| \vec{w} \|^2}{2} - \vec{w}.\vec{w} - b \times 0 + \sum_{i=1}^{N} \lambda_i$$

$$= \sum_{i=1}^{N} \lambda_i + \frac{\vec{w}.\vec{w}}{2} - \vec{w}.\vec{w}$$

$$= \sum_{i=1}^{N} \lambda_i - \frac{\vec{w}.\vec{w}}{2}$$

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

$$= \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\| \vec{w} \|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b\sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

.

.

$$= \sum_{i=1}^{N} \lambda_i - \frac{\vec{w}.\vec{w}}{2}$$

$$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i \cdot \sum_{j=1}^{N} \lambda_j y_j \vec{x}_j$$

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

$$= \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i y_i \vec{w}.\vec{x}_i - \sum_{i=1}^{N} \lambda_i y_i b + \sum_{i=1}^{N} \lambda_i$$

$$= \frac{\| \vec{w} \|^2}{2} - \vec{w}.\sum_{i=1}^{N} \lambda_i y_i \vec{x}_i - b \sum_{i=1}^{N} \lambda_i y_i + \sum_{i=1}^{N} \lambda_i$$

.

.

$$= \sum_{i=1}^{N} \lambda_i - \frac{\vec{w}.\vec{w}}{2}$$

$$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i \cdot \sum_{j=1}^{N} \lambda_j y_j \vec{x}_j$$

$$= \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

Support Vector Machines

- Replace w with λ's in L_p :

$$L_p = \frac{\| \vec{w} \|^2}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i (\vec{w}.\vec{x}_i + b) - 1 \right)$$

- The dual to be maximized:

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$$

Support Vector Machines

- After solving λ's :
  - Find ***w*** and b:

$$\frac{\partial L_p}{\partial w} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i}$$

$$\vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} = 1$$

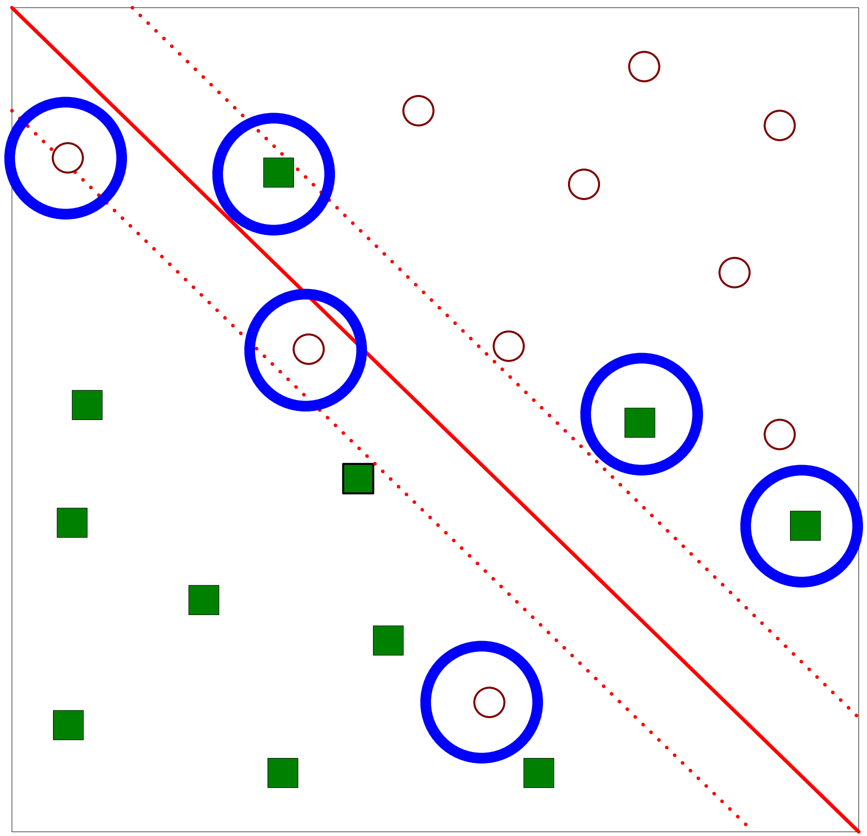
Support Vector Machines

- Classify an unknown example ***z***:

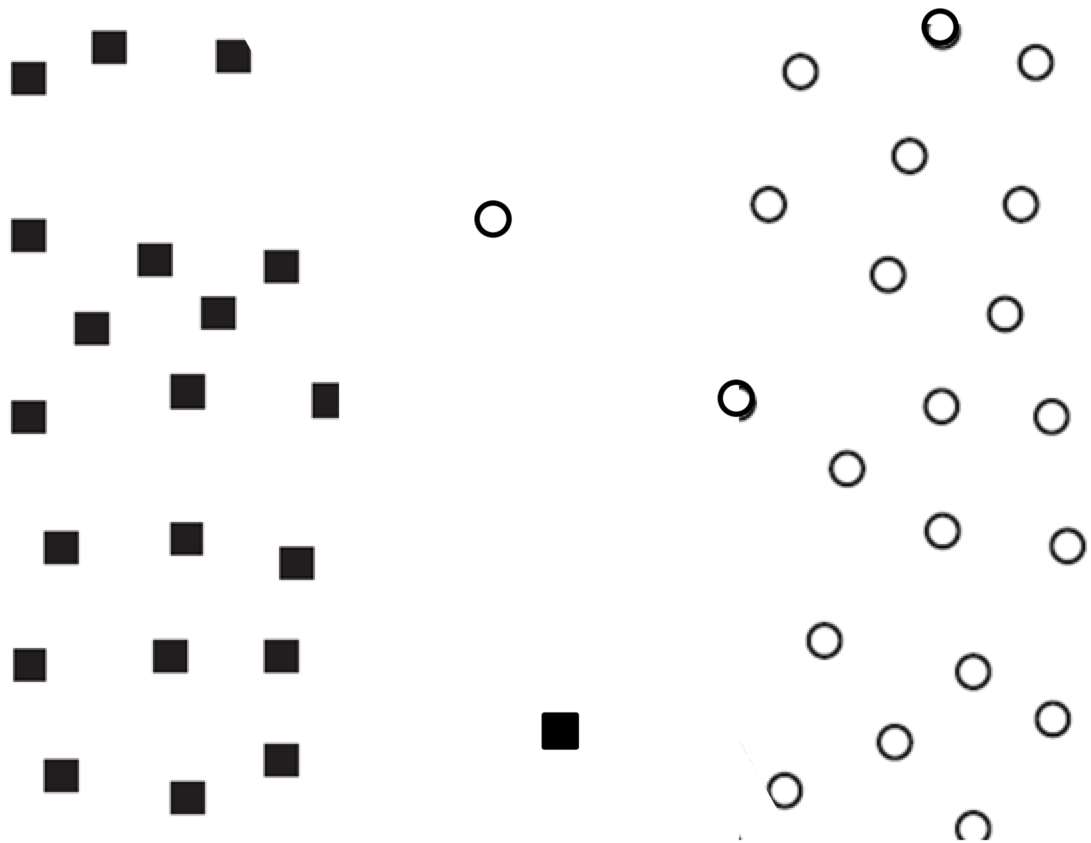
$$f(\mathbf{z}) = sign(\mathbf{w} \cdot \mathbf{z} + b)$$

Support Vector Machines

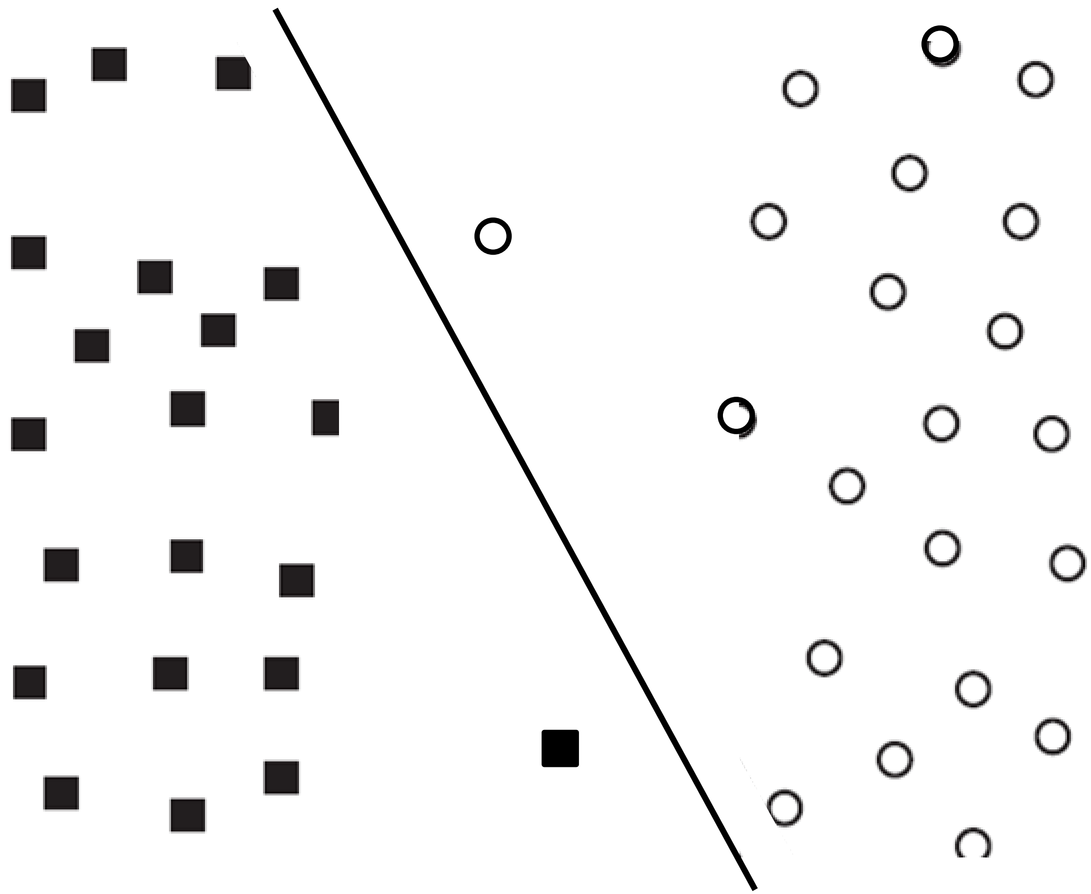
- What happens, if the problem is not linearly separable?



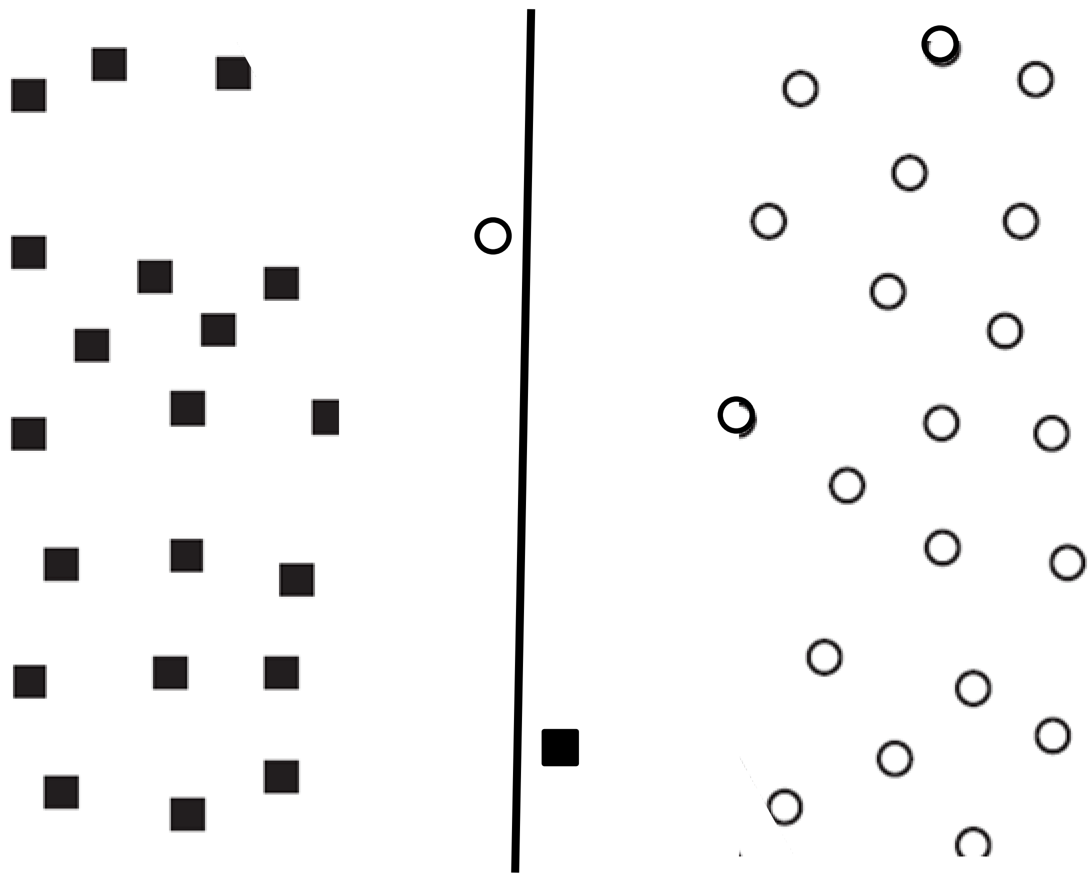
Linear SVM for Non-separable Case



Linear SVM for Non-separable Case

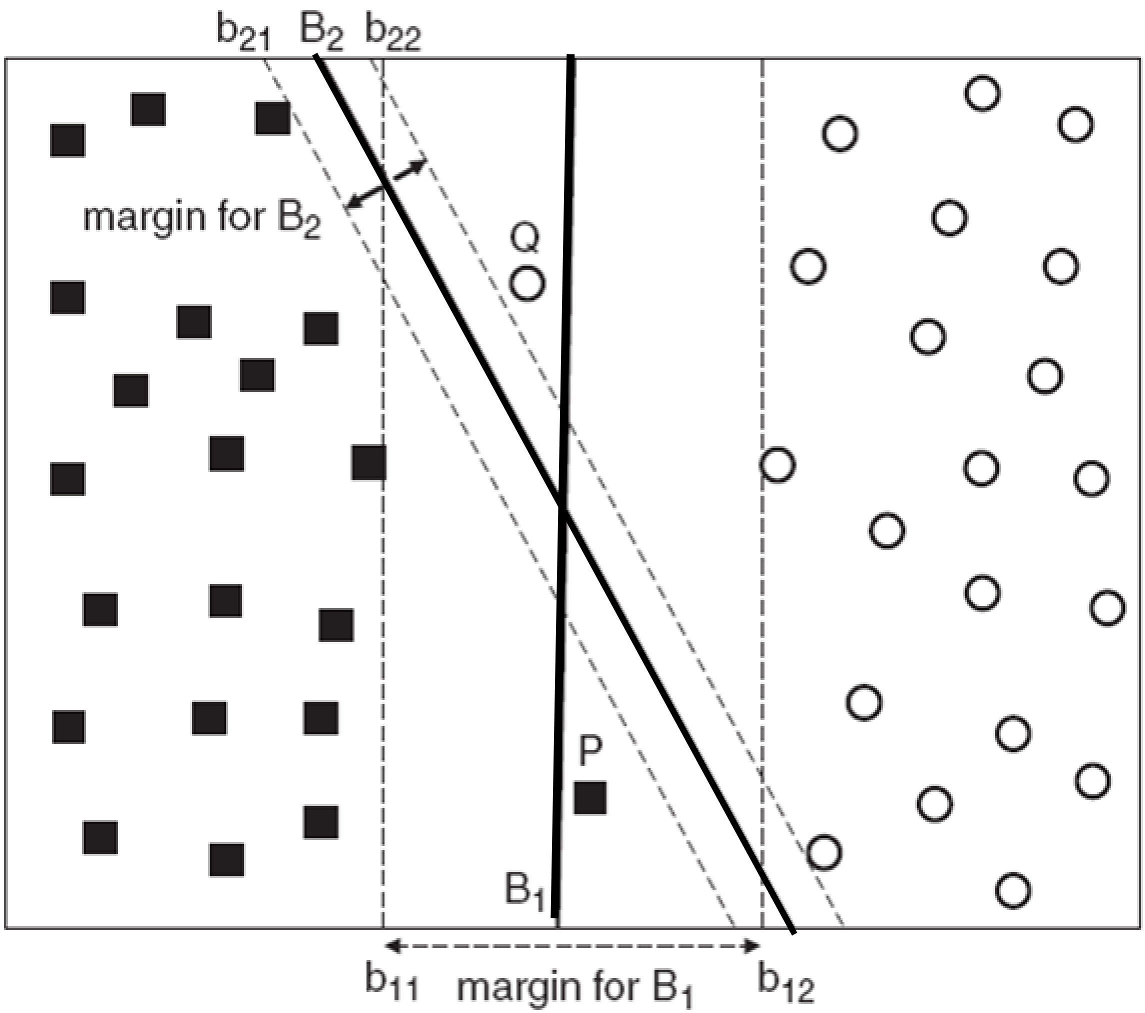


Linear SVM for Non-separable Case



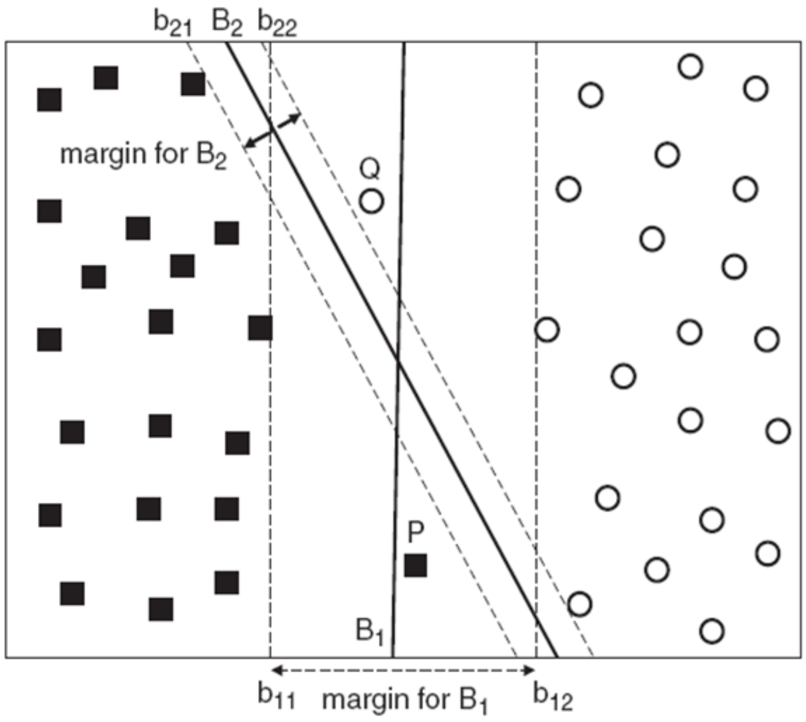
Linear SVM for Non-separable Case

- Which one is **better**: B_1 or B_2?



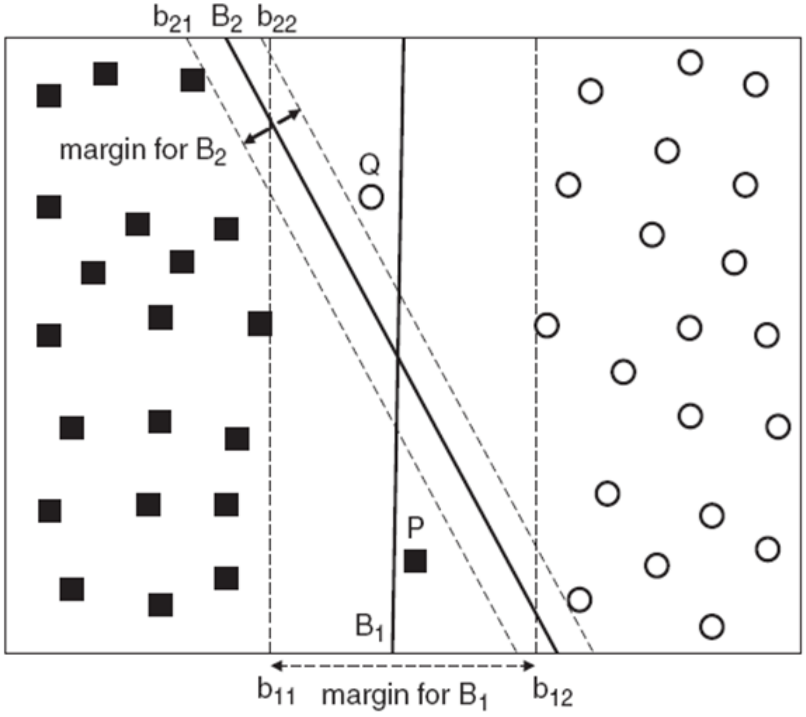
Linear SVM for Non-separable Case

- Target: use **Linear SVM** to separable non-separable samples
- How: use soft margin
- **Tolerable to** small training **error**
- Need a **trade-off** between **margin** and **training errors**



Linear SVM for Non-separable Case

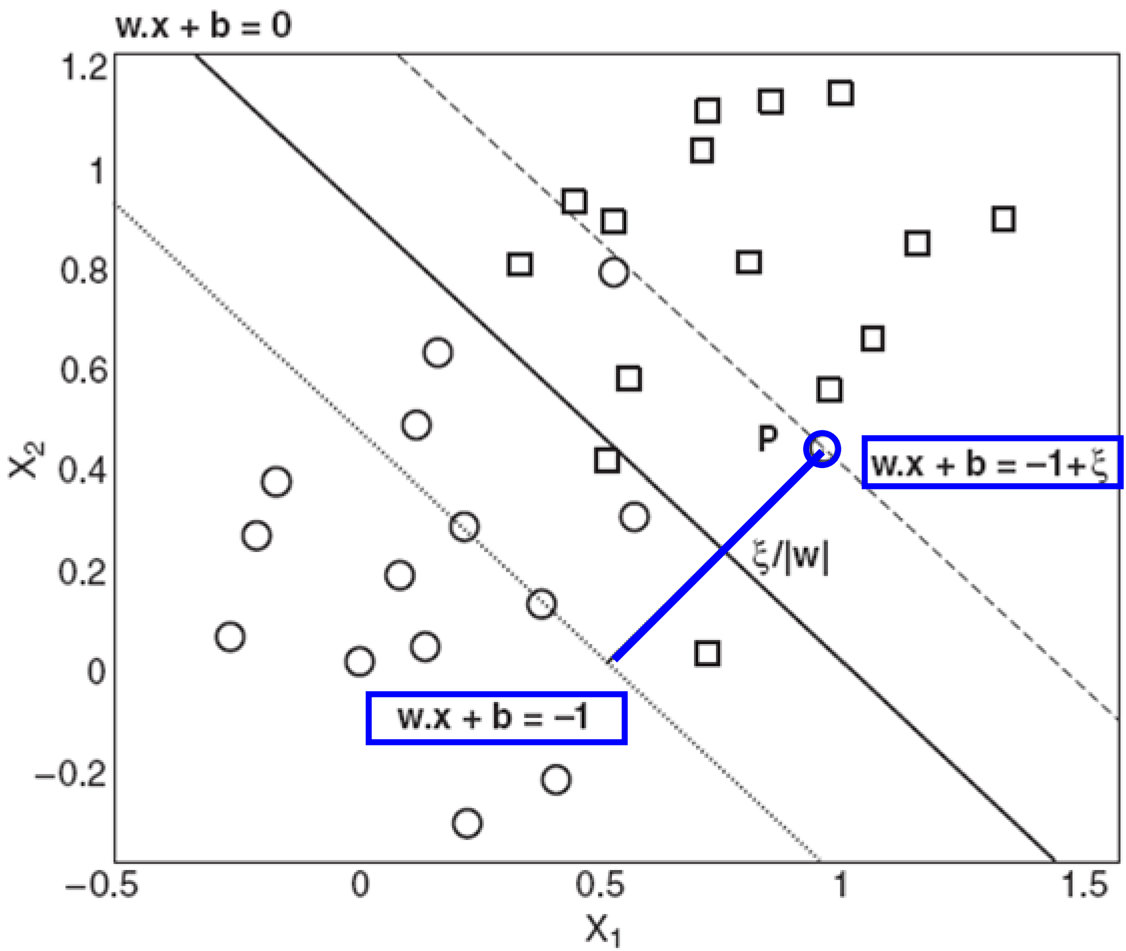
- P and Q no longer satisfy previous constraints
- Introduce slack variables (ξ) to relax the constraints



$$\vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \text{ if } y_i = 1,$$
$$\vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \text{ if } y_i = -1$$
$$\text{where, } \forall_i : \xi_i > 0$$

Linear SVM for Non-separable Case

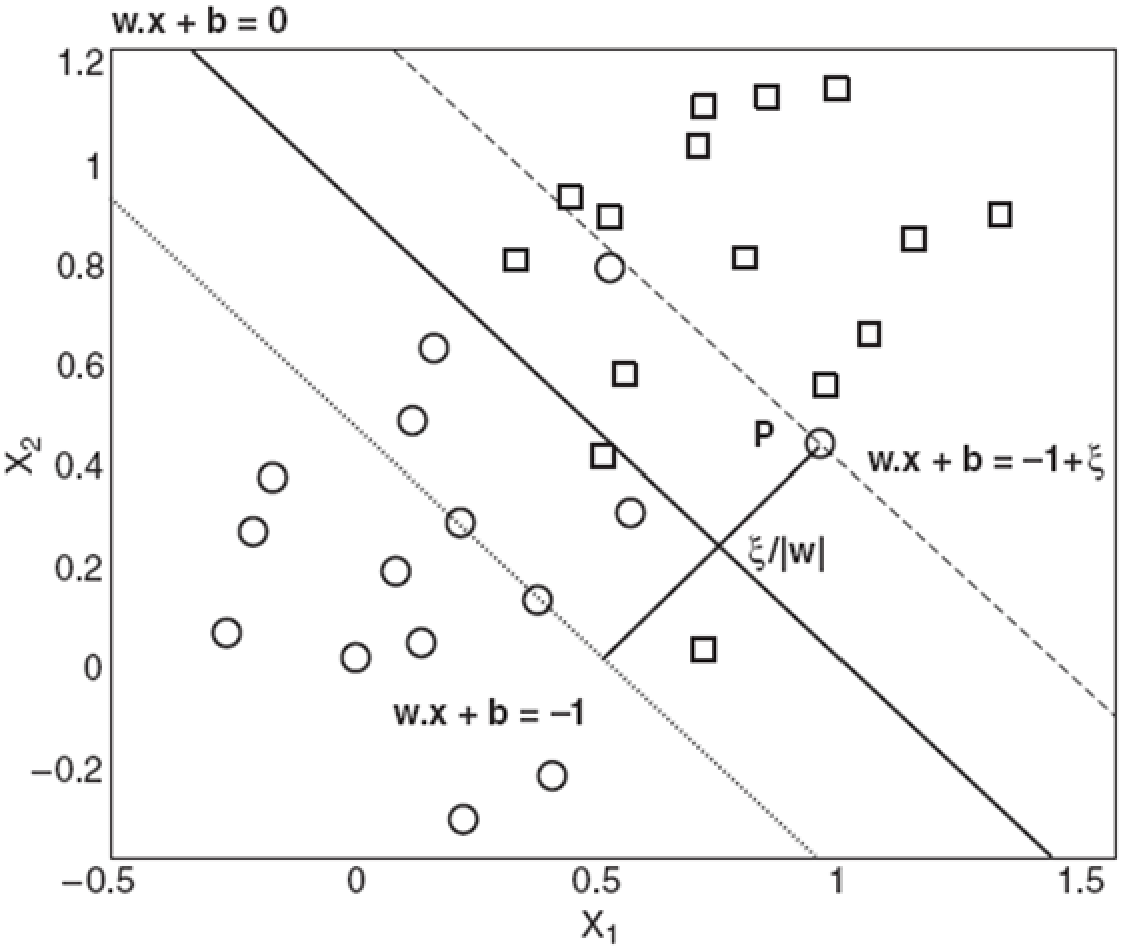
- What ξ indicates?



Linear SVM for Non-separable Case

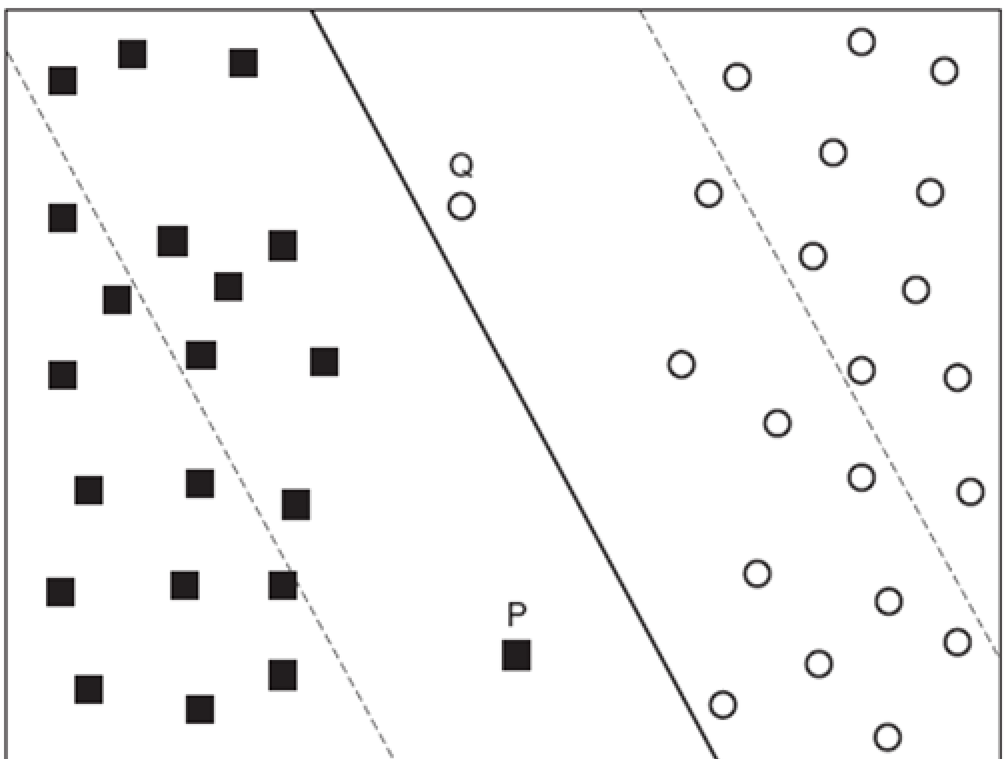
$$\vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \text{ if } y_i = 1,$$
$$\vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \text{ if } y_i = -1$$
$$\text{where, } \forall_i : \xi_i > 0$$

- Slack variables (ξ) estimates the error of decision boundary



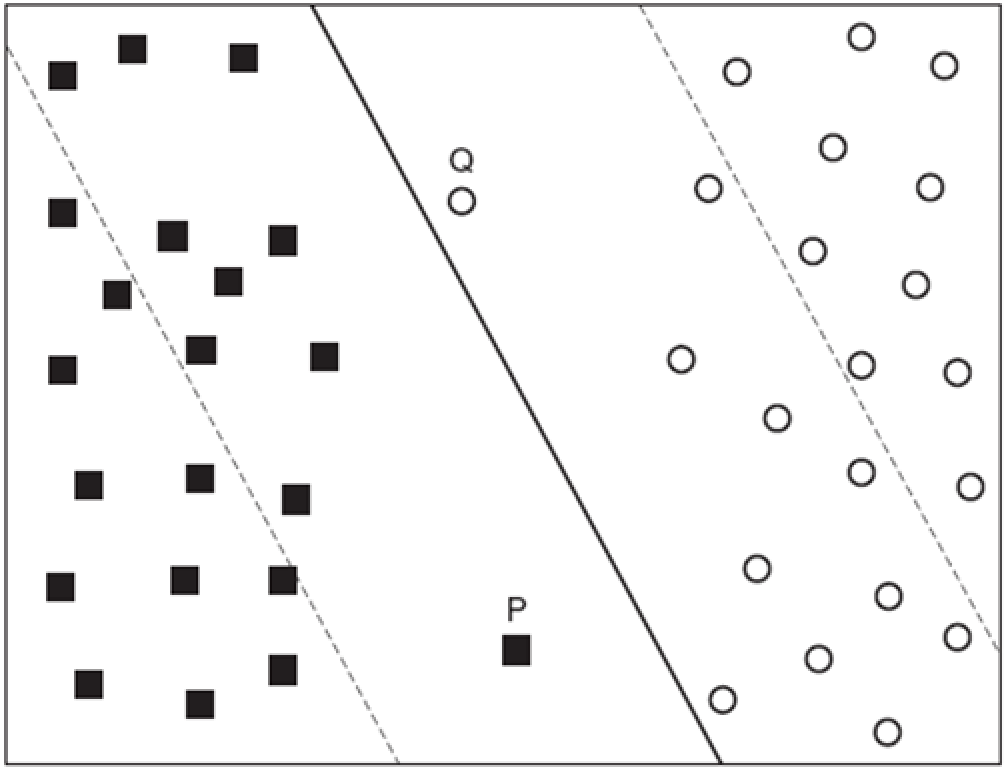
Linear SVM for Non-separable Case

- trade off:
  - too large margin
  - too many misclassification



Linear SVM for Non-separable Case

- Use Slack variables (ξ) in objective function



Linear SVM for Non-separable Case

– The new objective function is

$$f(w) = \frac{\|\vec{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k$$

– Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 - \xi_{\mathrm{i}} \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 + \xi_{\mathrm{i}} \end{cases}$$

Linear SVM for Non-separable Case

- The Lagrange primal using new objective function is

$$L_P = \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\{y_i(\vec{w}\cdot\vec{x}+b)-1+\xi_i\} - \sum_{i=1}^{N}\mu_i\xi_i$$

- Subject to:

$$\xi_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0$$

Linear SVM for Non-separable Case

– The Lagrange primal using new objective function is

$$L_P = \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\{y_i(\vec{w}\cdot\vec{x}+b)-1+\xi_i\} - \sum_{i=1}^{N}\mu_i\xi_i$$

– Subject to:

$$\xi_i \geq 0, \lambda_i \geq 0, \mu_i \geq 0$$

$$\lambda_i\{y_i(\vec{w}.\vec{x}_i+b)-1+\xi_i\}=0$$

$$\mu_i\xi_i = 0$$

Linear SVM for Non-separable Case

– The Lagrange primal

$$L_P = \frac{\| \vec{w} \|^2}{2} + C\sum_{i=1}^{N} \xi_i \ - \sum_{i=1}^{N} \lambda_i \{ y_i (\vec{w} \cdot \vec{x} + b) - 1 + \xi_i \ \} - \sum_{i=1}^{N} \mu_i \xi_i$$

– First derivative w. r. to variables:

$$\frac{\partial L_p}{\partial \vec{w}} = 0 \quad \Rightarrow \quad \vec{w} = \sum_{i=1}^{N} \lambda_i y_i \vec{x}_i \qquad \frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\frac{\partial L_p}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \lambda_i - \mu_i = 0 \quad \Rightarrow \quad C = \lambda_i + \mu_i$$

Linear SVM for Non-separable Case

– The Lagrange primal

$$L_P = \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\{y_i(\vec{w}\cdot\vec{x}+b)-1+\xi_i\} - \sum_{i=1}^{N}\mu_i\xi_i$$

– The dual is

$$L_D = \boxed{\frac{\|\vec{w}\|^2}{2}} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\{y_i(\boxed{\vec{w}\cdot\vec{x}}+b)-\boxed{1}+\xi_i\} - \sum_{i=1}^{N}(C-\lambda_i)\xi_i$$

$$= \sum_{i=1}^{N}\lambda_i - \frac{\|\vec{w}\|^2}{2} = \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \vec{x}_i\cdot\vec{x}_j$$

What is the Difference?

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

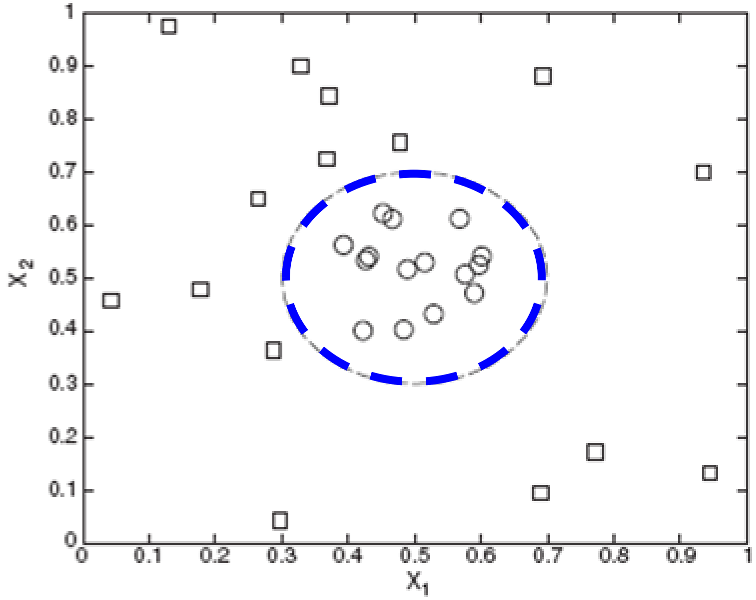
- In linearly **separable** cases:
  - This means λ's are unbounded

$$\lambda_i \geq 0$$

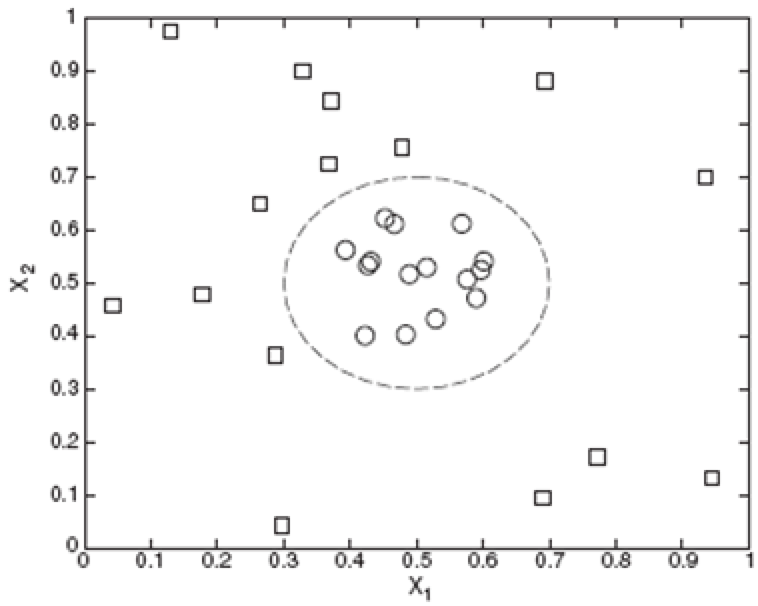
- In linearly **non-separable** cases:

$$C = \lambda_i + \mu_i \;\;\Rightarrow\;\; 0 \leq \lambda_i \leq C$$

What’s Next?



- How can we separate these examples?
  - Non-linear SVM
  - Neural Network



Nonlinear Classifier

Review of Perceptron's Capability

Recall the AND or OR functions

x_1	x_2	AND		OR	
0	0	0		0	
0	1	0		1	
1	0	0		1	
1	1	1		1	

Review of Perceptron's Capability

Recall the AND or OR functions

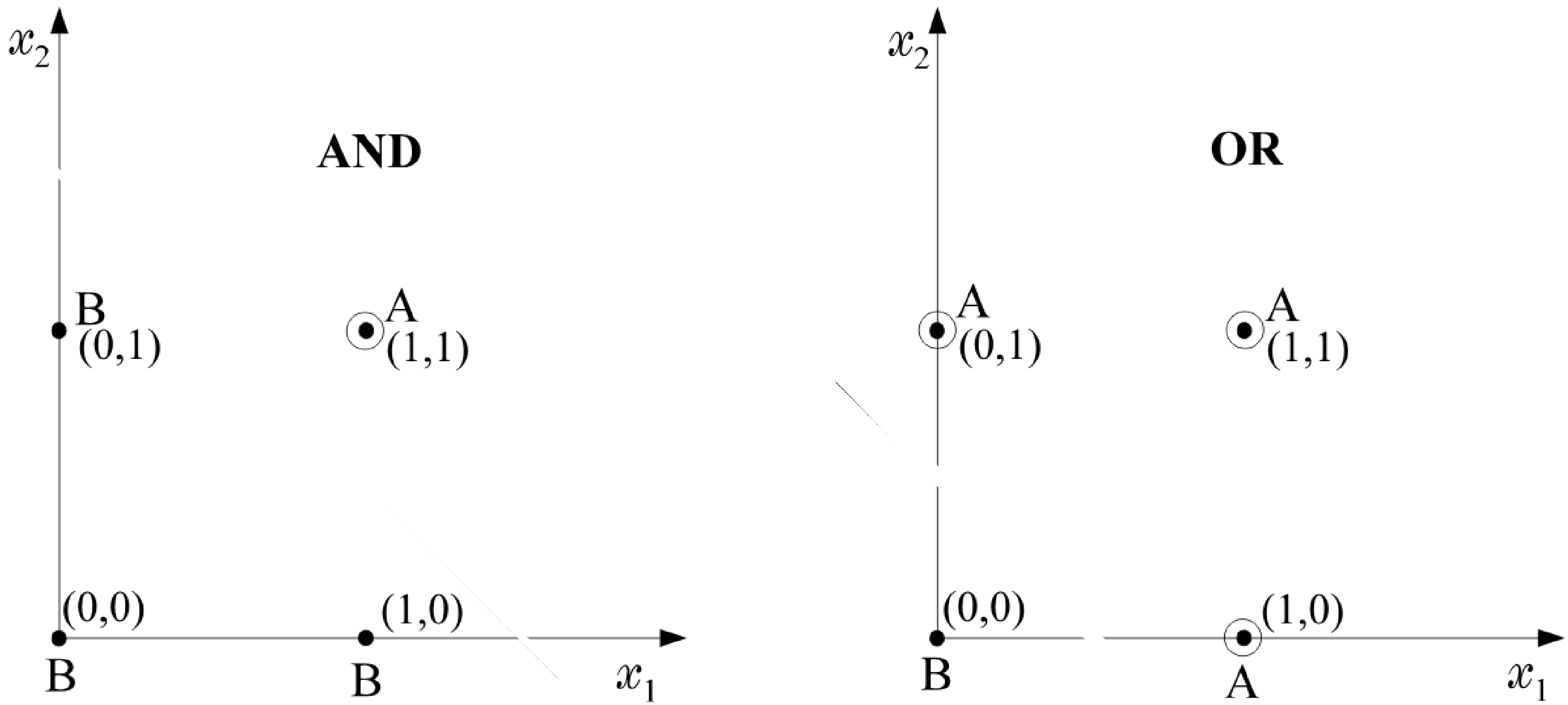
x_1	x_2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

Review of Perceptron's Capability

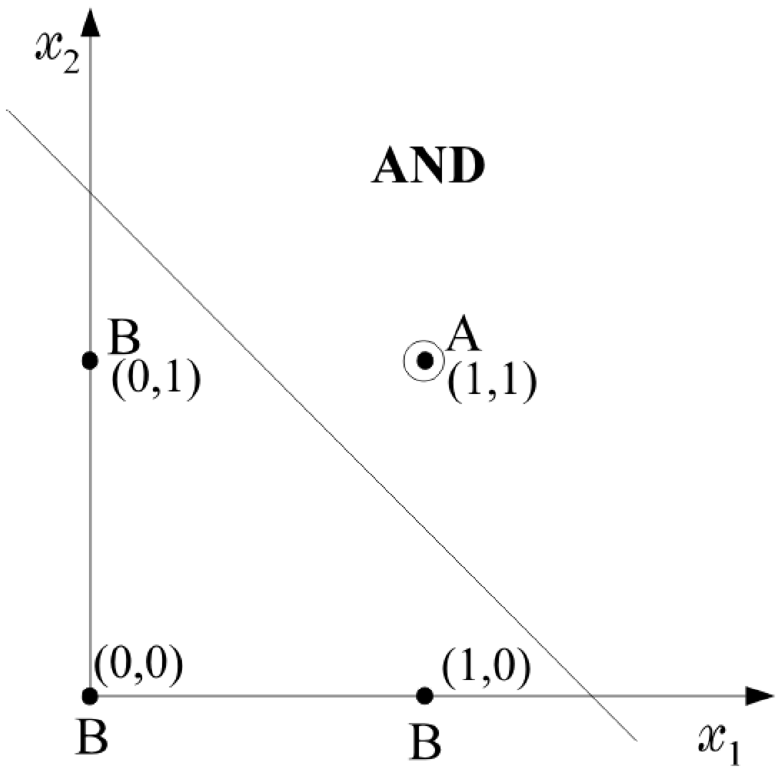
Can you remember the perceptron's capability to separate them?

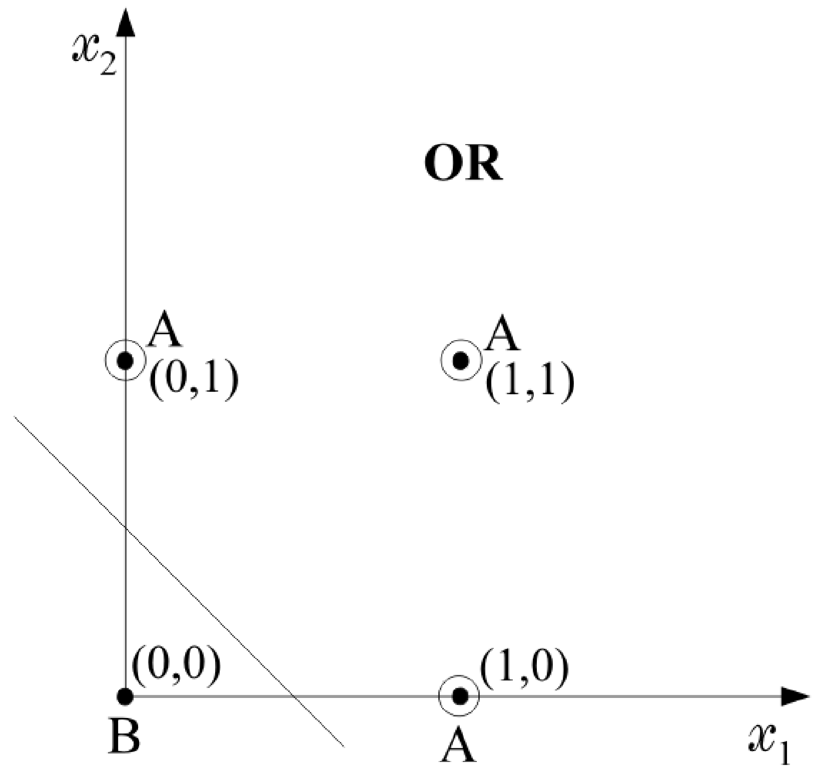
x_1	x_2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

Review of Perceptron's Capability



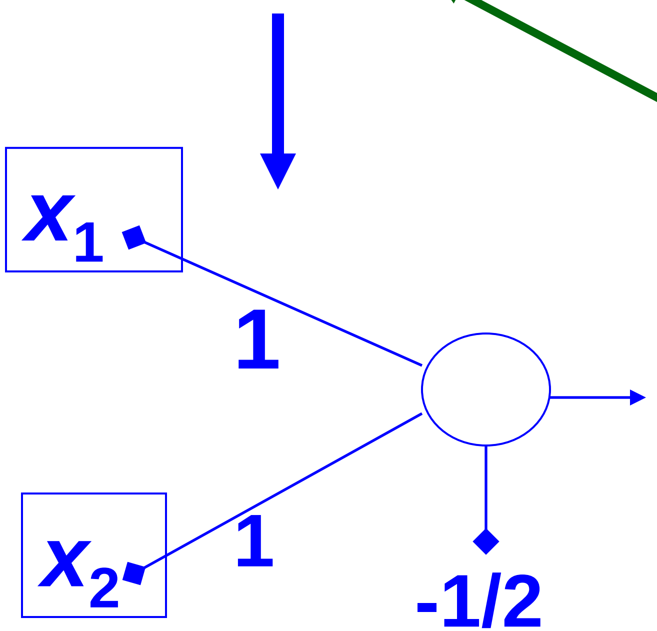
Review of Perceptron's Capability

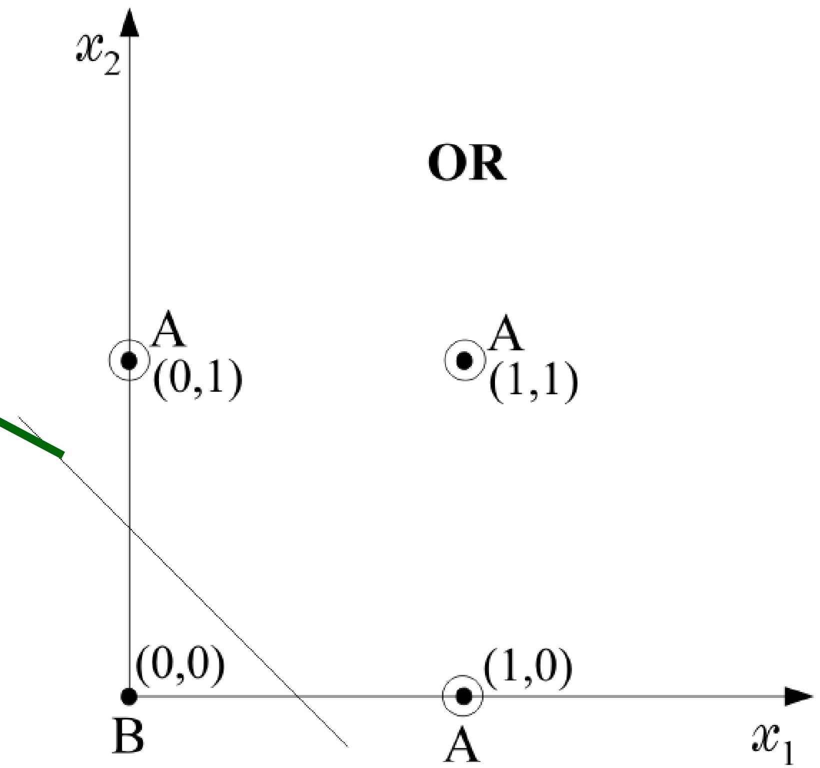




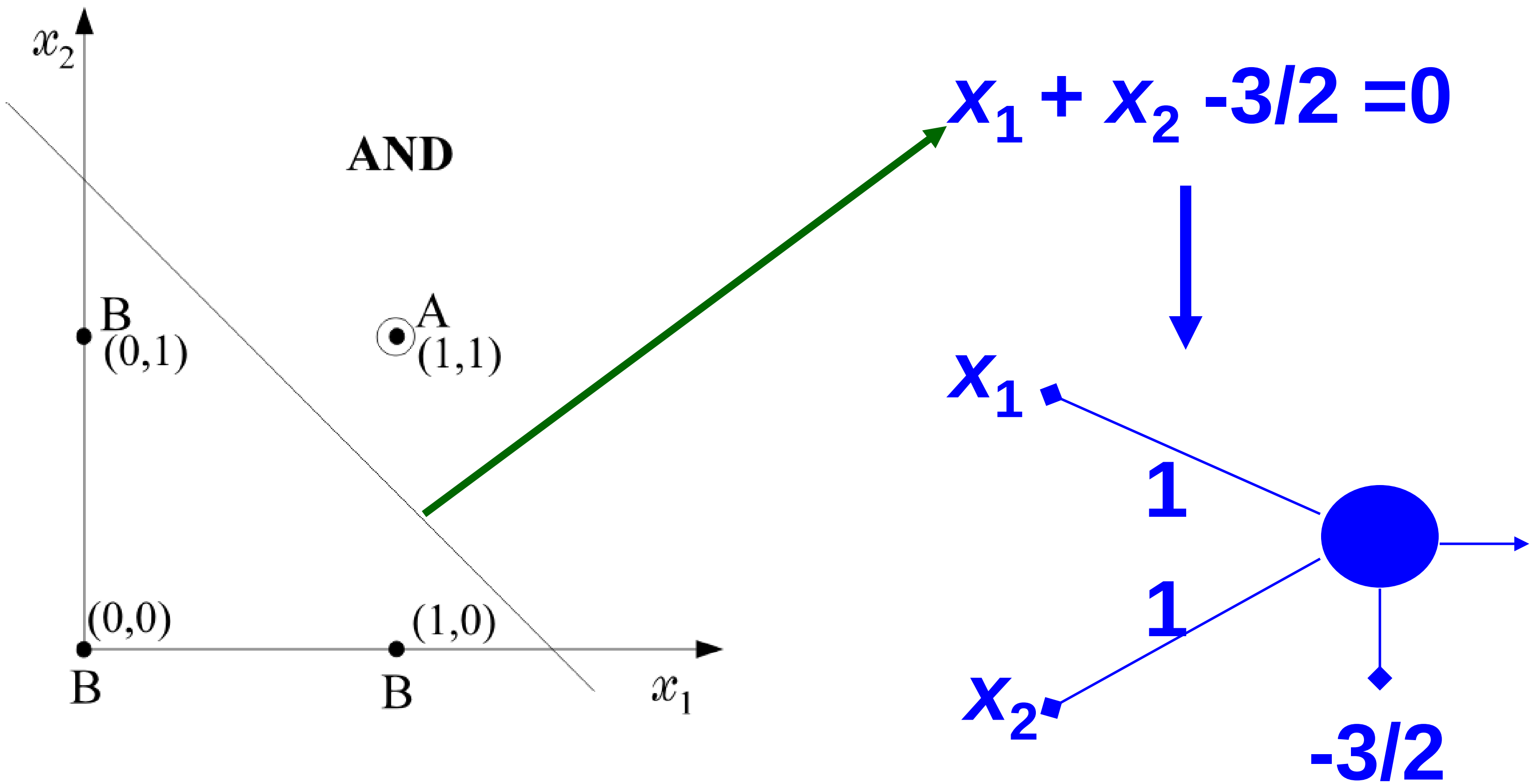
Review of Perceptron’s Capability

$$x_1 + x_2 - 1/2 = 0$$





Review of Perceptron's Capability



Review of Perceptron's Capability

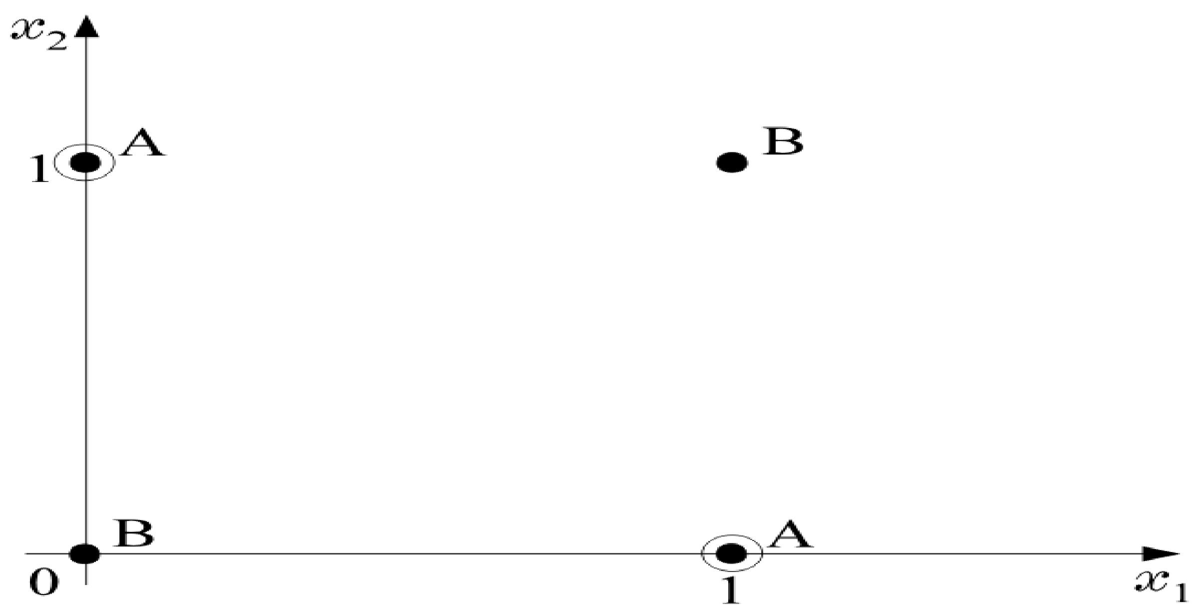
Now recall the XOR function

x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B

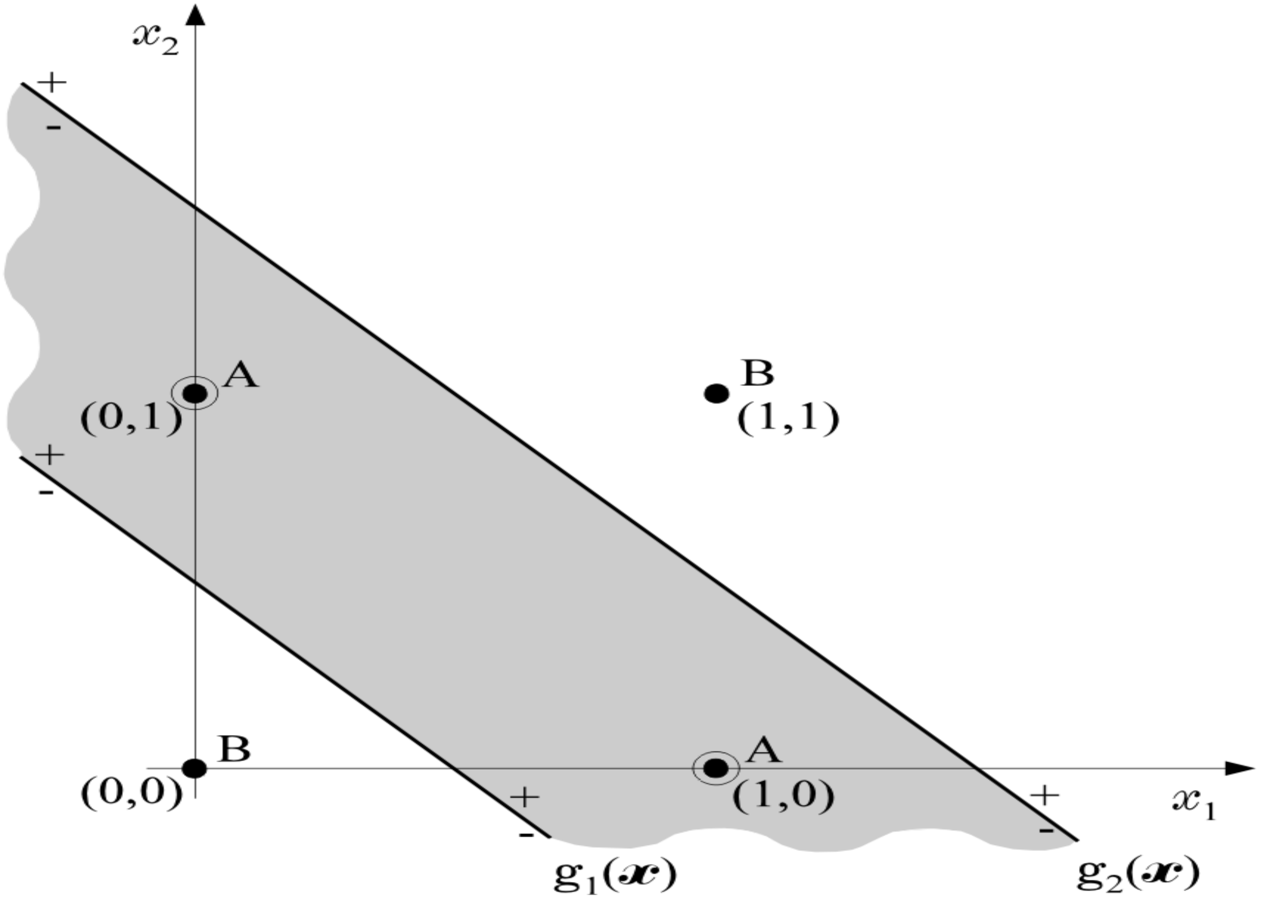
Review of Perceptron's Capability

Now recall the XOR function

x_1	x_2	XOR	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B

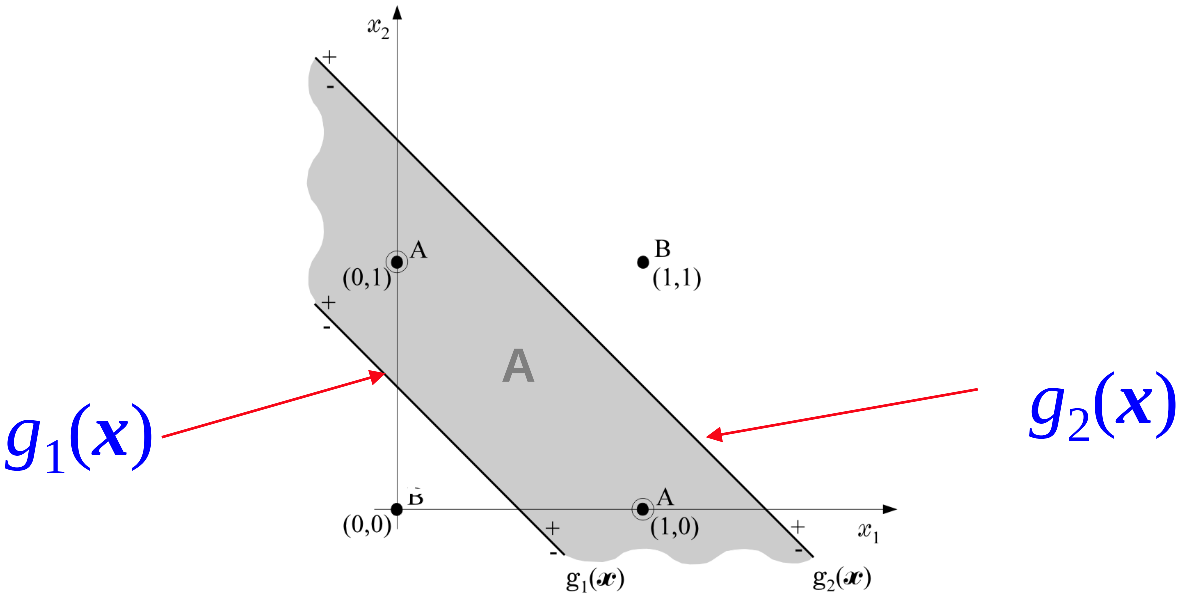


Review of Perceptron's Capability



- For the XOR problem, draw two lines instead of one

Review of Perceptron's Capability



- Each of them is realized by a perceptron.

$$y_i = f(g_i(\underline{x})) = \begin{cases} 0 \\ 1 \end{cases} \quad i = 1, 2$$

- Find the position of $\underline{x}$ *w.r.t.* both lines, based on the values of y_1, y_2.

Review of Perceptron's Capability

1st phase			
x_1	x_2	y_1	y_2
0	0	-	-
0	1	+	-
1	0	+	-
1	1	+	+

- Equivalently: The computations of the first phase perform a mapping $\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$

Review of Perceptron's Capability

1st phase			
x_1	x_2	y_1	y_2
0	0	0(-)	0(-)
0	1	1(+)	0(-)
1	0	1(+)	0(-)
1	1	1(+)	1(+)

- Equivalently: The computations of the first phase perform a mapping $\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$

Review of Perceptron's Capability

1^{st} phase			
x_1	x_2	y_1	y_2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

- Equivalently: The computations of the first phase perform a mapping $\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$

Review of Perceptron's Capability

1st phase				2nd phase
x_1	x_2	y_1	y_2	
0	0	0	0	B(0)
0	1	1	0	A(1)
1	0	1	0	A(1)
1	1	1	1	B(0)

- Equivalently: The computations of the first phase perform a mapping $\underline{x} \rightarrow \underline{y} = [y_1, y_2]^T$

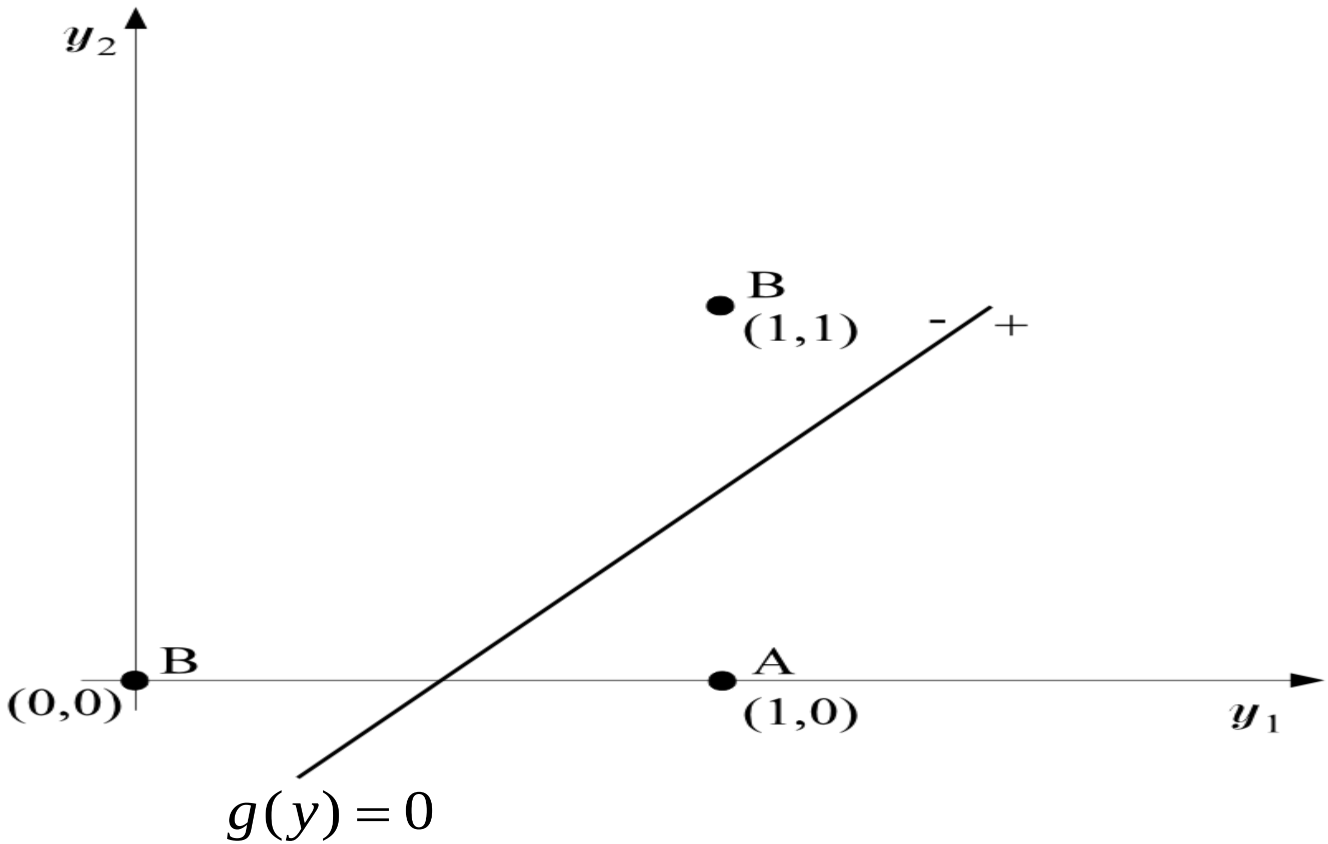
Review of Perceptron's Capability

1st phase				2nd phase
x_1	x_2	y_1	y_2	
0	0	0	0	B(0)
0	1	1	0	A(1)
1	0	1	0	A(1)
1	1	1	1	B(0)

Now classify based on $[y_1, y_2]$

Review of Perceptron's Capability

The decision is now performed on the transformed $\underline{y}$ data.

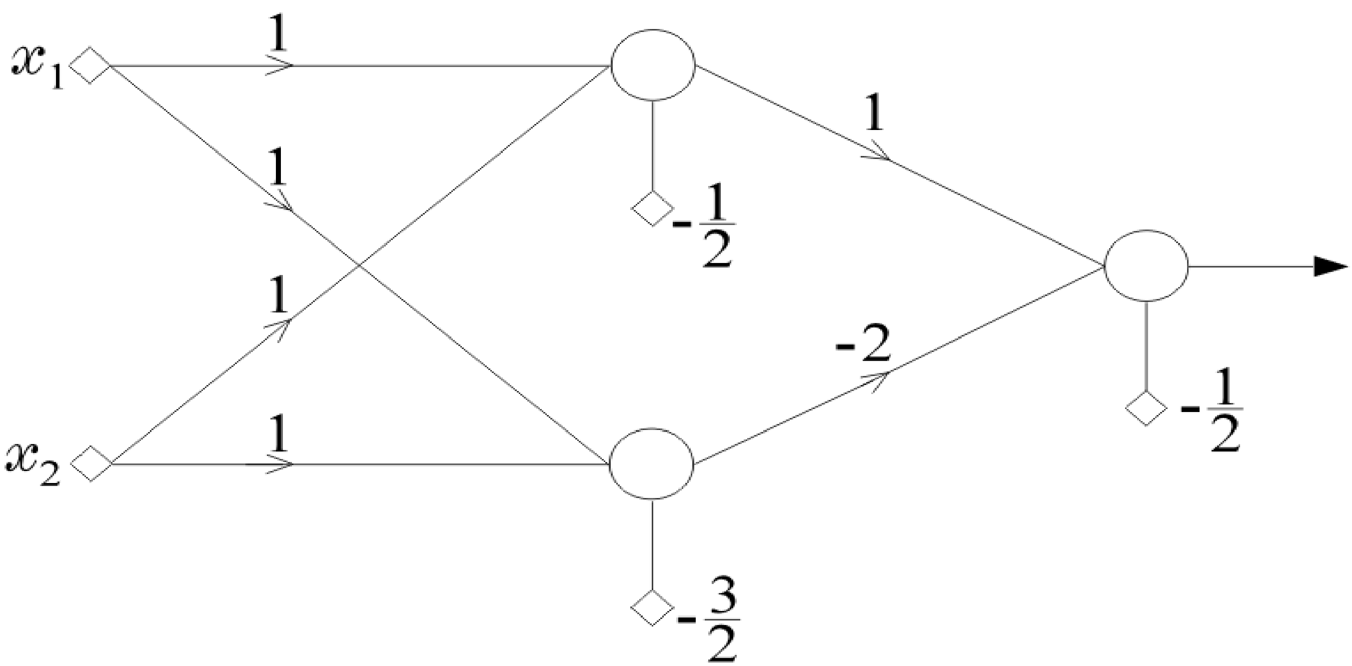


This can be performed via a second line, which can also be realized by a perceptron.

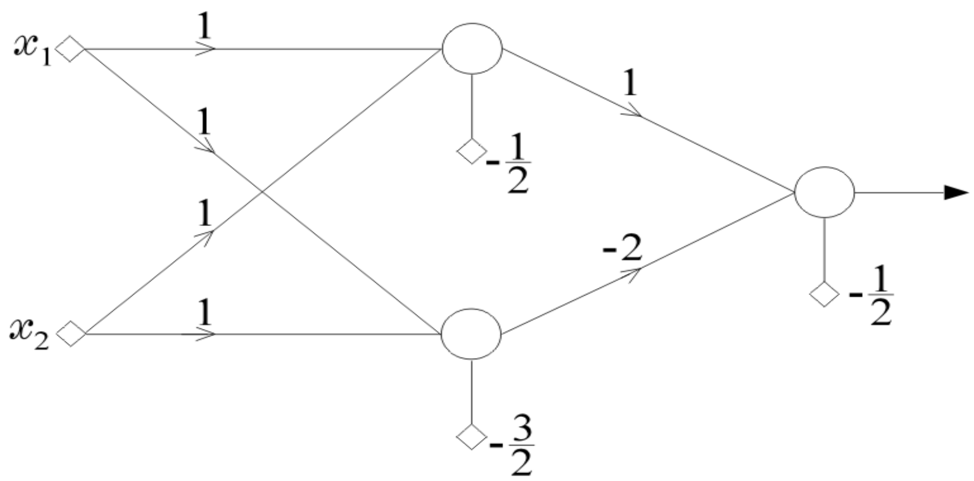
Two phases, Two Layers

- Computations of the first phase perform a mapping that transforms the nonlinearly separable problem to a linearly separable one.

- The architecture



Two Layer Perceptron



hidden layer

output layer

nodes realizes hyper planes:

$$g_1(\underline{x}) = x_1 + x_2 - \frac{1}{2} = 0$$

$$g_2(\underline{x}) = x_1 + x_2 - \frac{3}{2} = 0$$

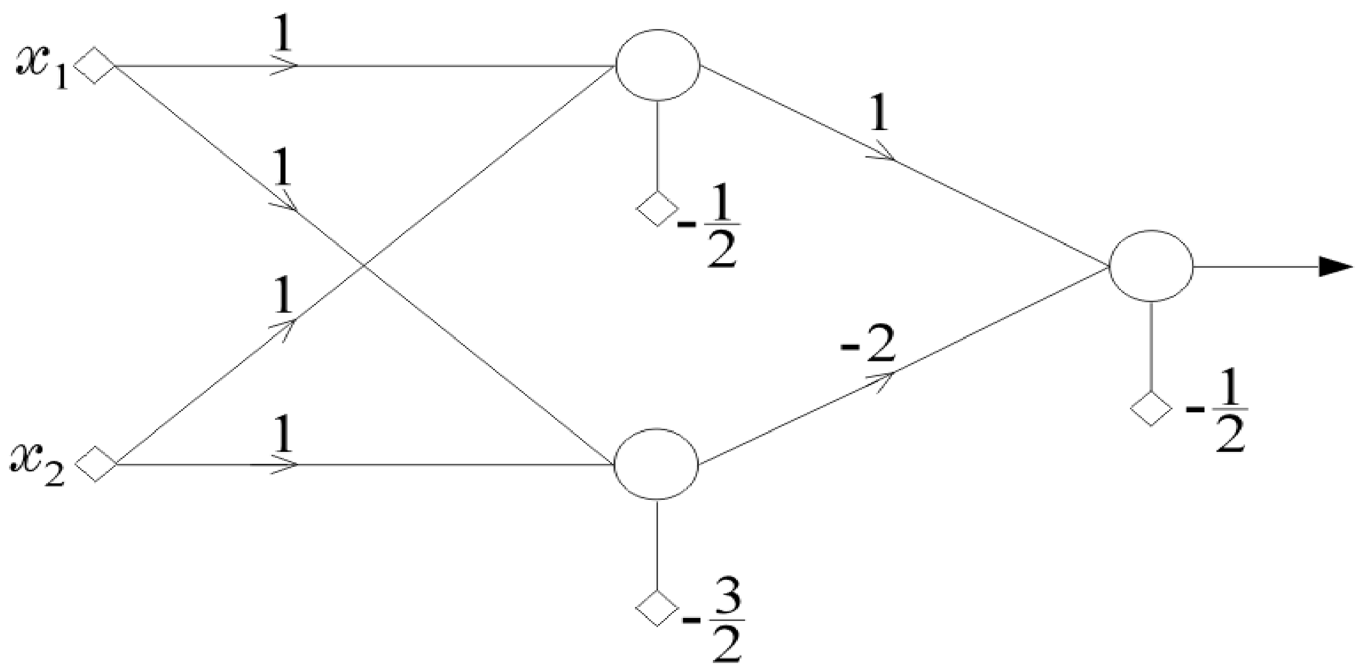
$$g(\underline{y}) = y_1 - 2y_2 - \frac{1}{2} = 0$$

Activation function:

$$f(.) = \begin{cases} 0 \\ 1 \end{cases}$$

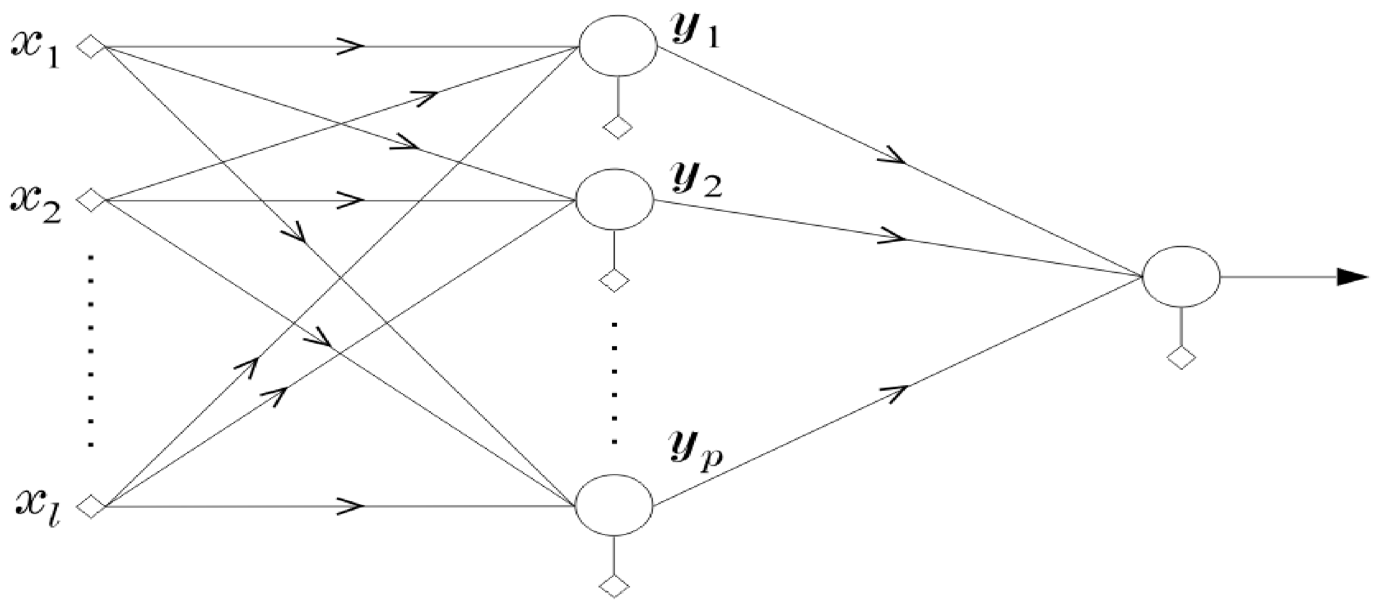
Classification Capabilities of Two Layer Perceptron

- The mapping performed by the first layer neurons is onto the vertices of the unit side square, e.g., (0, 0), (0, 1), (1, 0), (1, 1).



Classification Capabilities of Two Layer Perceptron

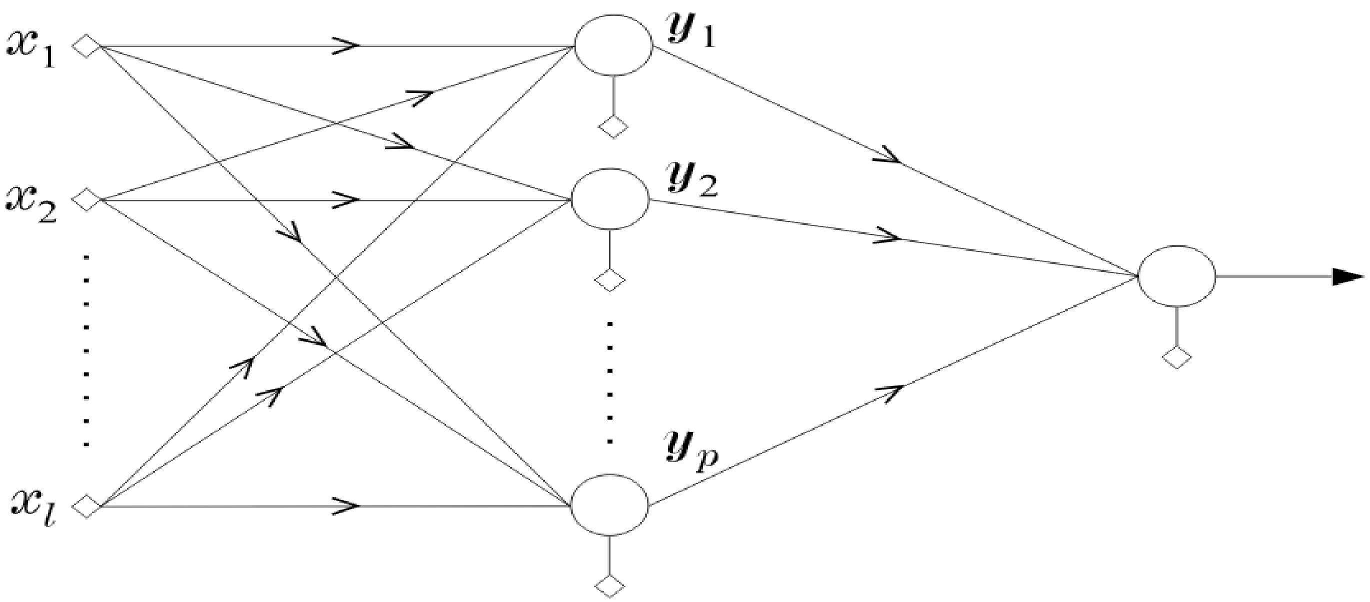
- Consider a more general case,



$$\underline{x} \in R^l$$

$$\underline{x} \rightarrow \underline{y} = [y_1, \ldots y_p]^T,\ y_i \in \{0, 1\}\ \ i = 1, 2, \ldots p$$

Classification Capabilities of Two Layer Perceptron



- maps a vector onto the vertices of the unit side hypercube, *Hp*
- mapping is through *p* neurons each realizing a hyper plane.
- The output of each of these neurons is 0 or 1