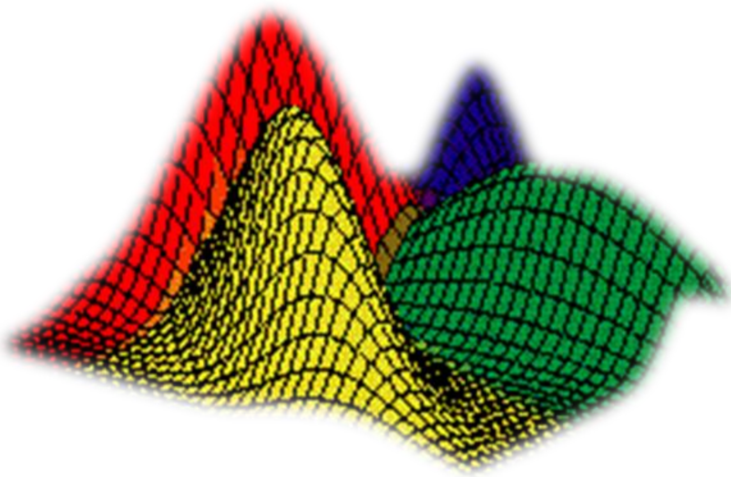
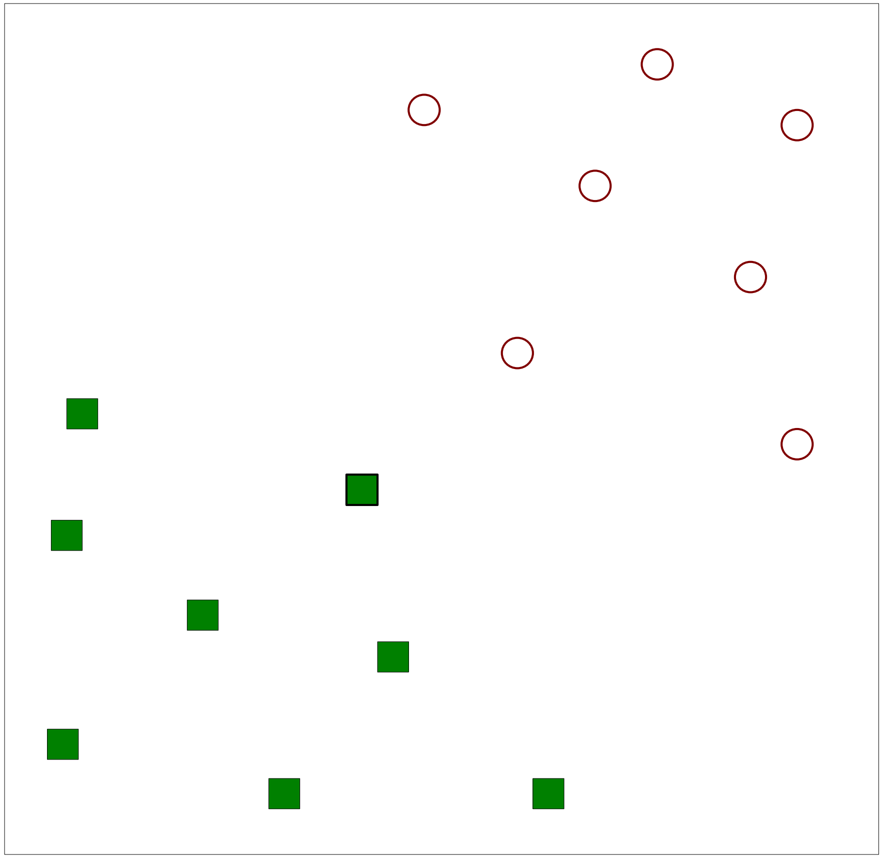


Pattern Recognition

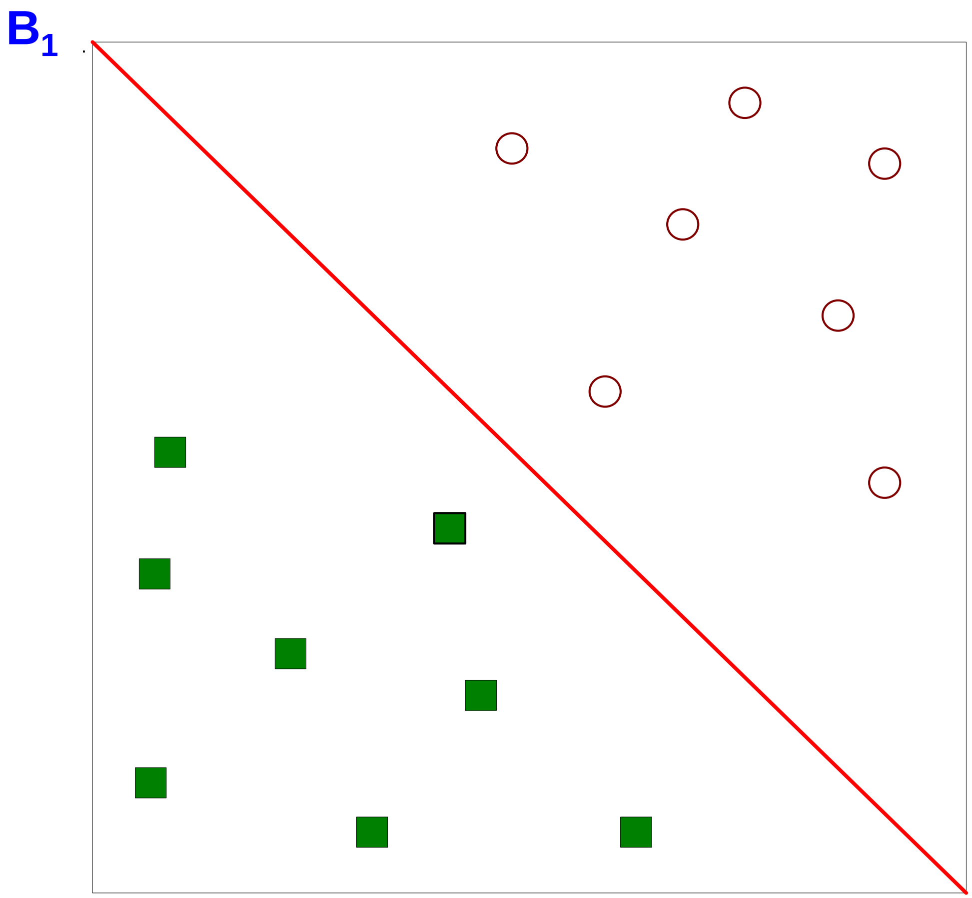
Linear Classifier

Two-Class case again



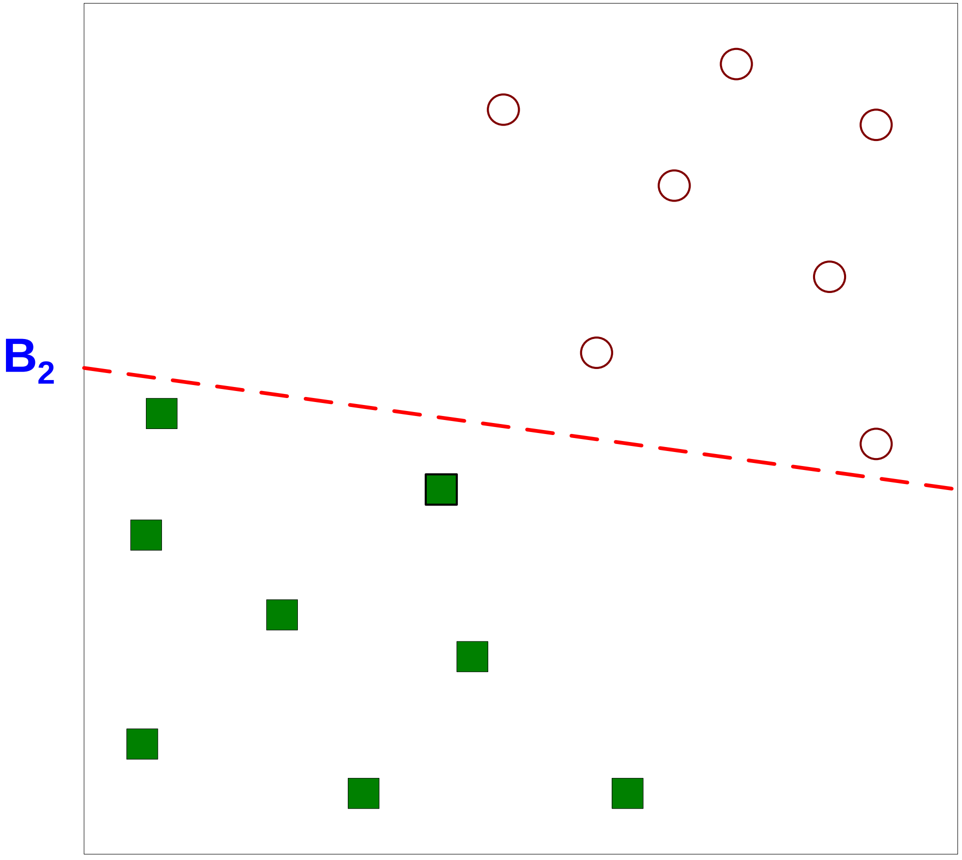
- Find a linear hyperplane (decision boundary) that will separate the data

Two-Class case again



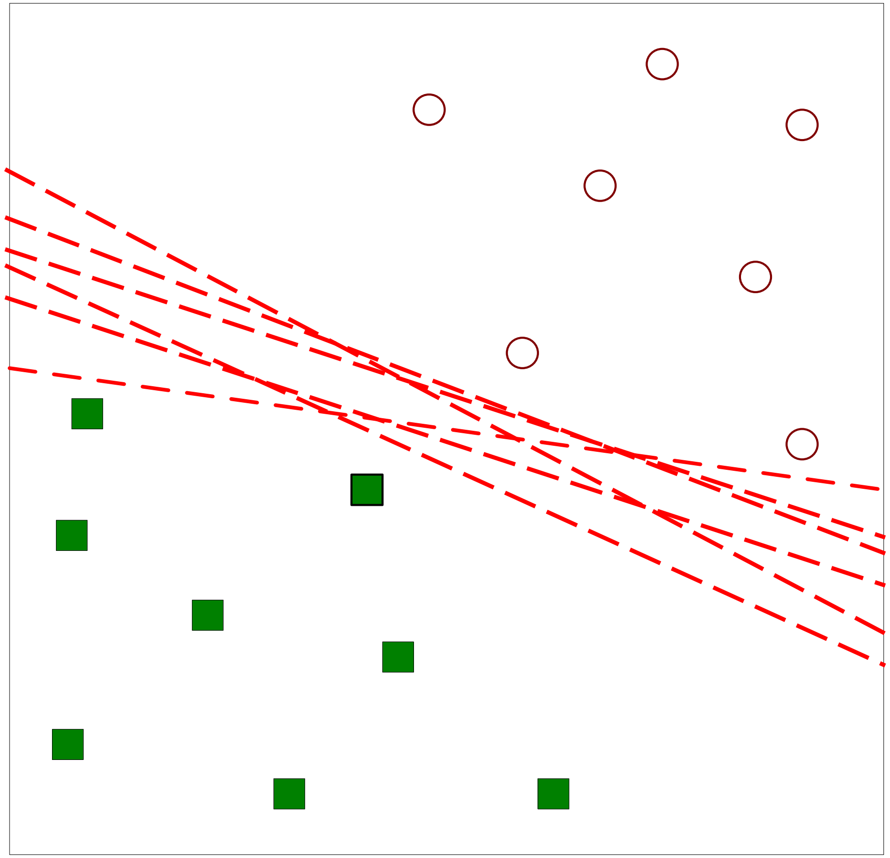
- One Possible Solution

Two-Class case again



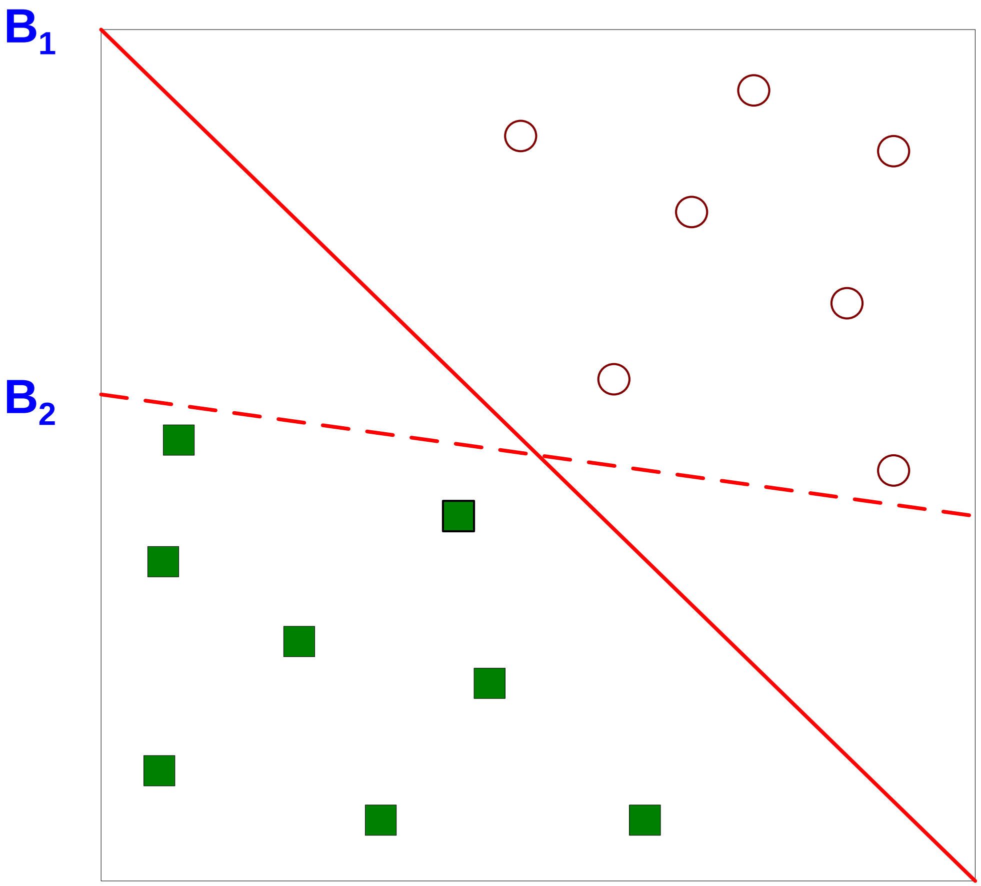
- Another possible solution

Two-Class case again



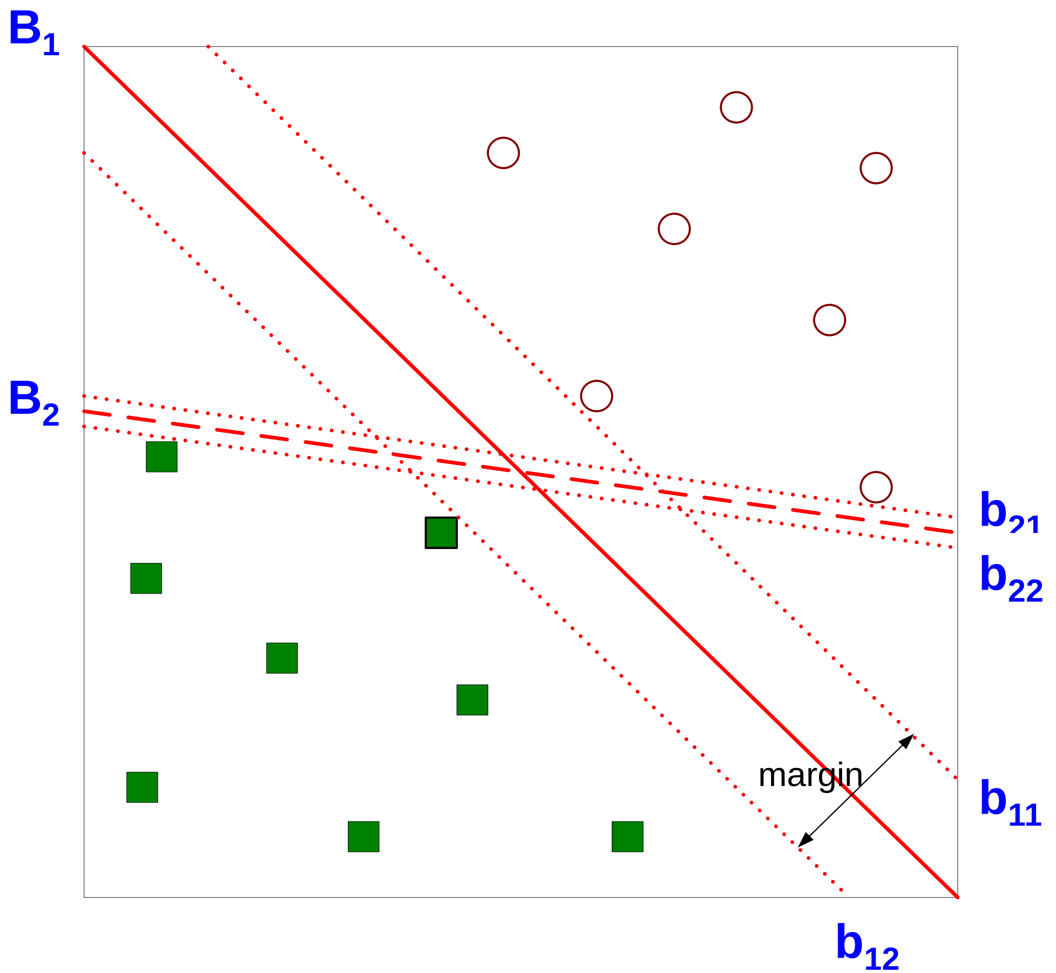
- Other possible solutions

Two-Class case again



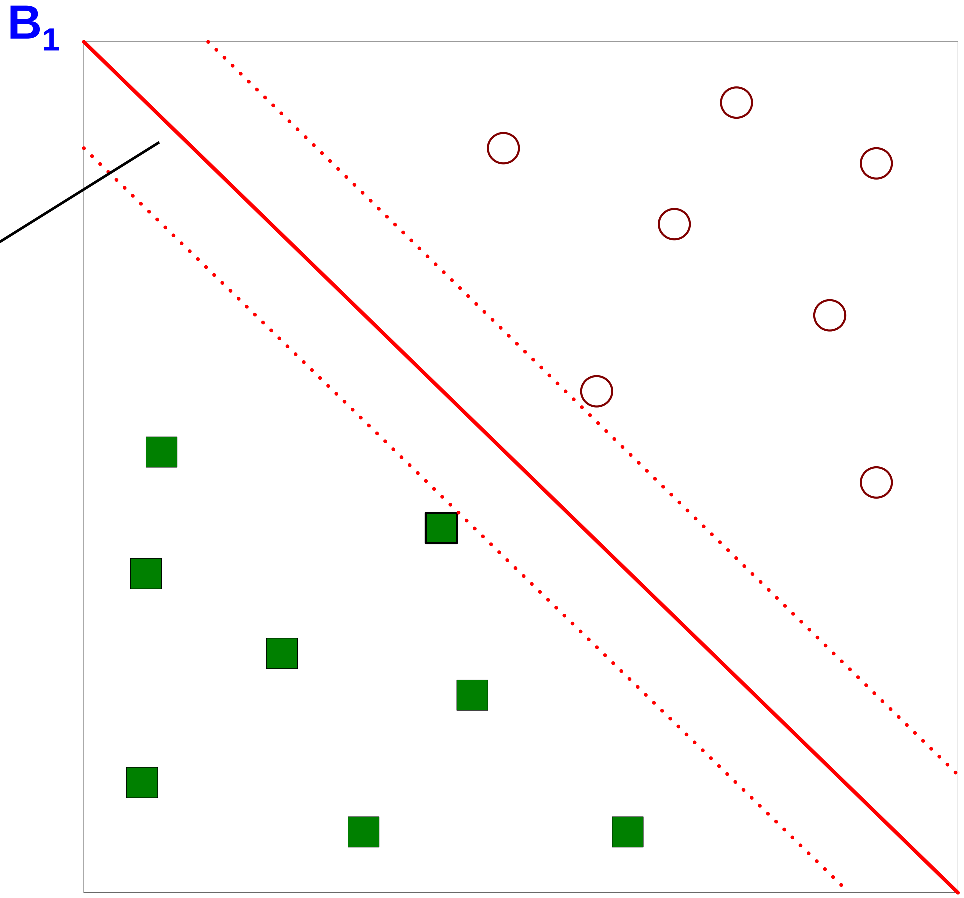
- Which one is better? B_1 or B_2?
- How do you define better?

Two-Class case again



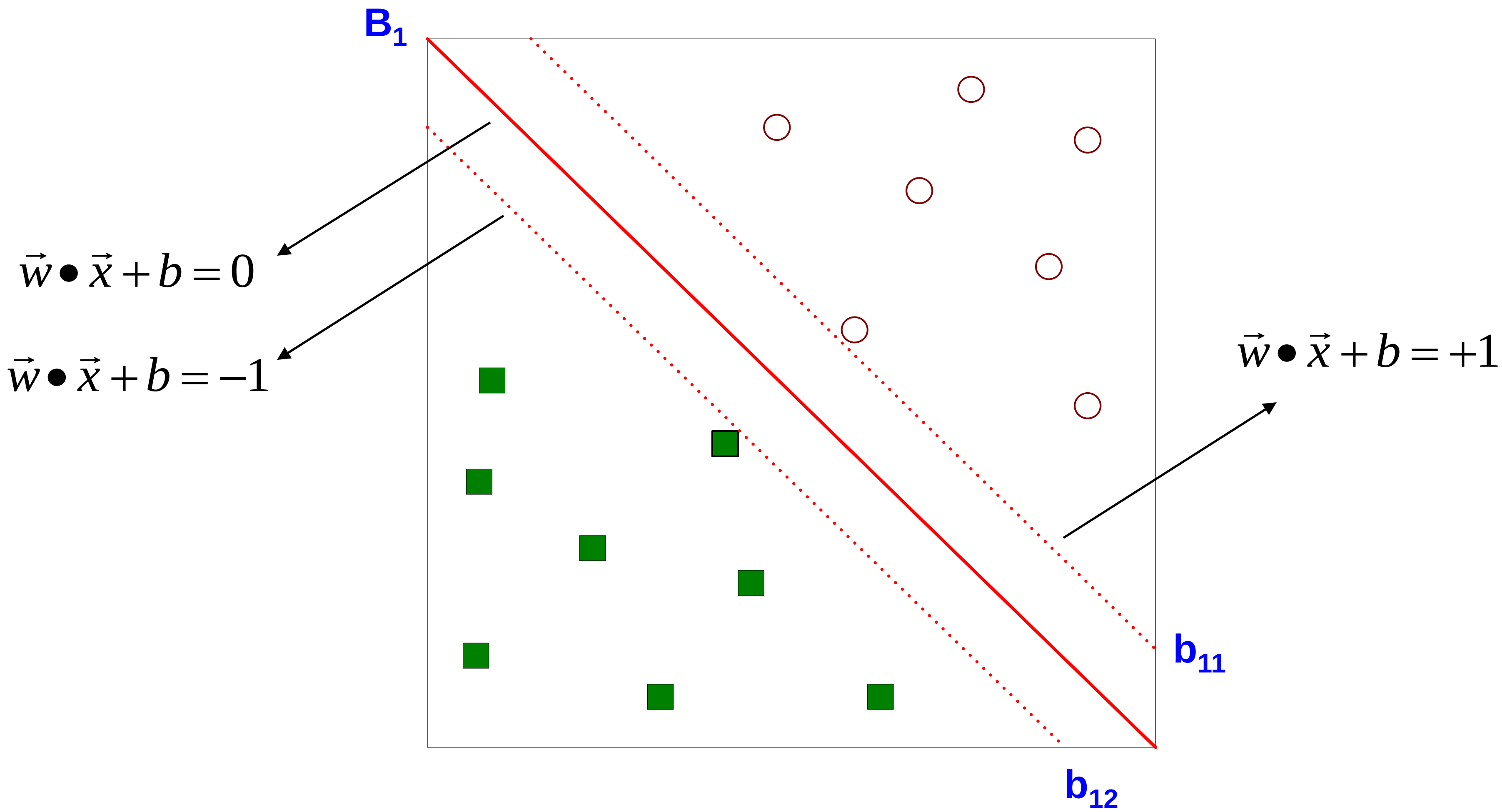
- Find hyperplane maximizes the margin => B1 is better than B2

Two-Class case again

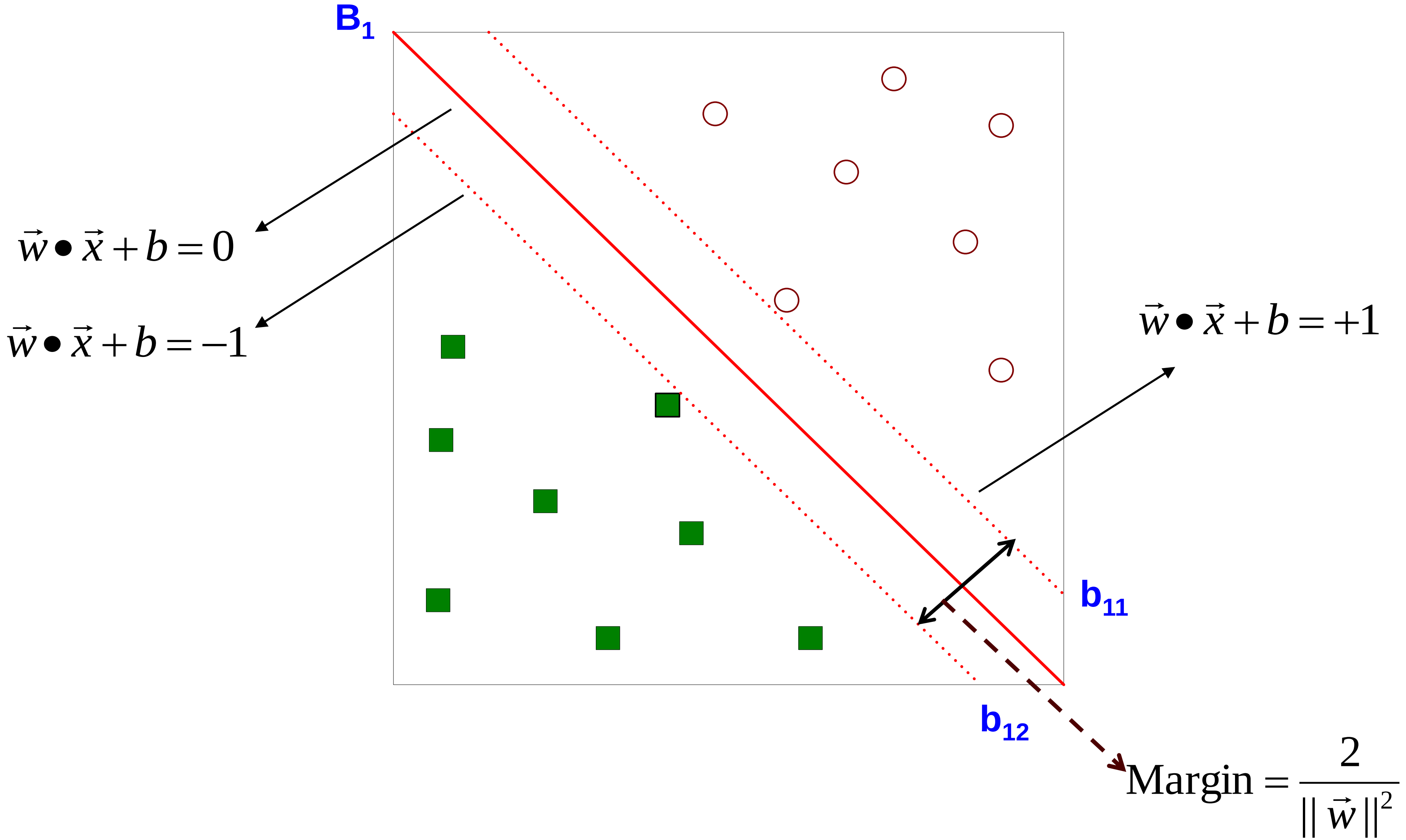


$$\vec{w} \bullet \vec{x} + b = 0$$

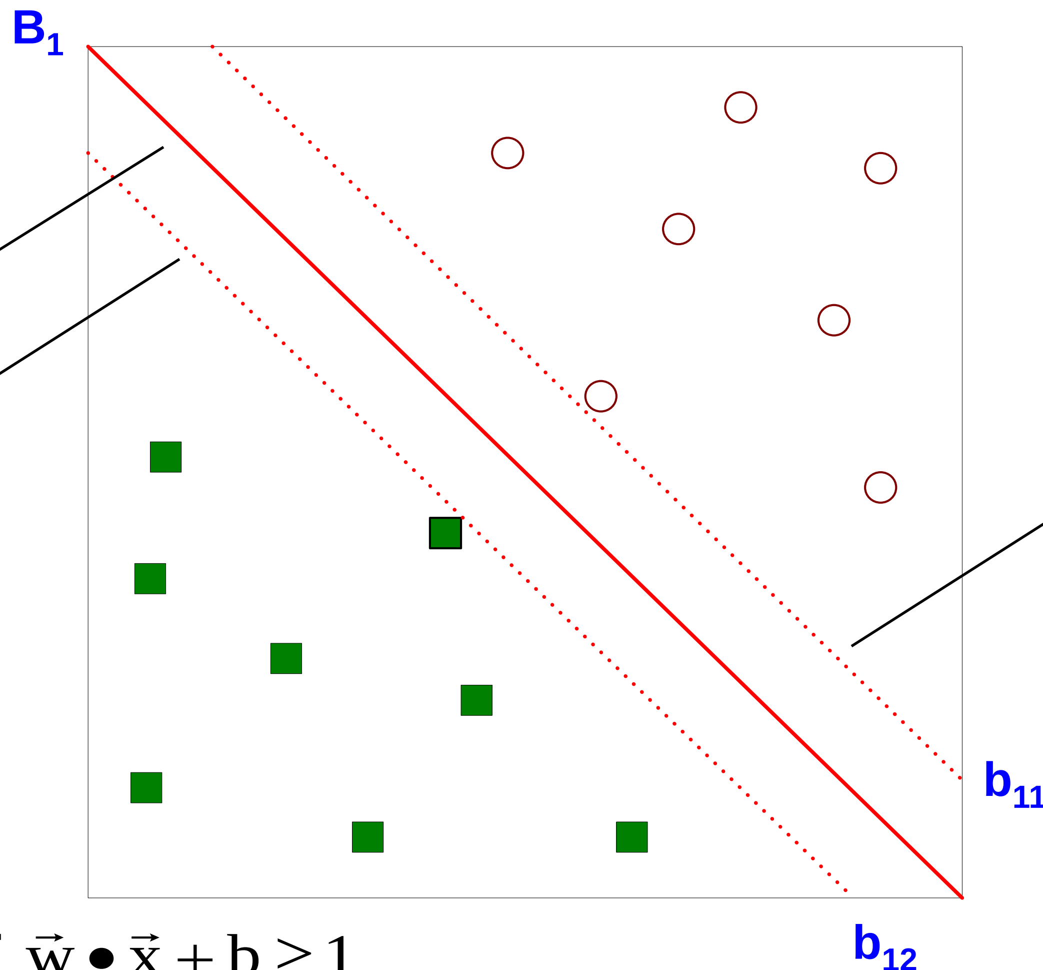
Two-Class case again



Two-Class case again



Two-Class case again



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

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

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

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

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

Support Vector Machines

- We want to maximize: $$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$
  - But subjected 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 Machines

- We want to maximize: $$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$
  - Which is equivalent to minimizing: $$L(w) = \frac{\|\vec{w}\|^2}{2}$$
  - But subjected 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 Machines

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 Machines

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:

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

Support Vector Machines

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

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

- What happens if ***w*** =0?

Support Vector Machines

- minimize:

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

- Subject to:

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

- 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,

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

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

$$\lambda_i \geq 0$$

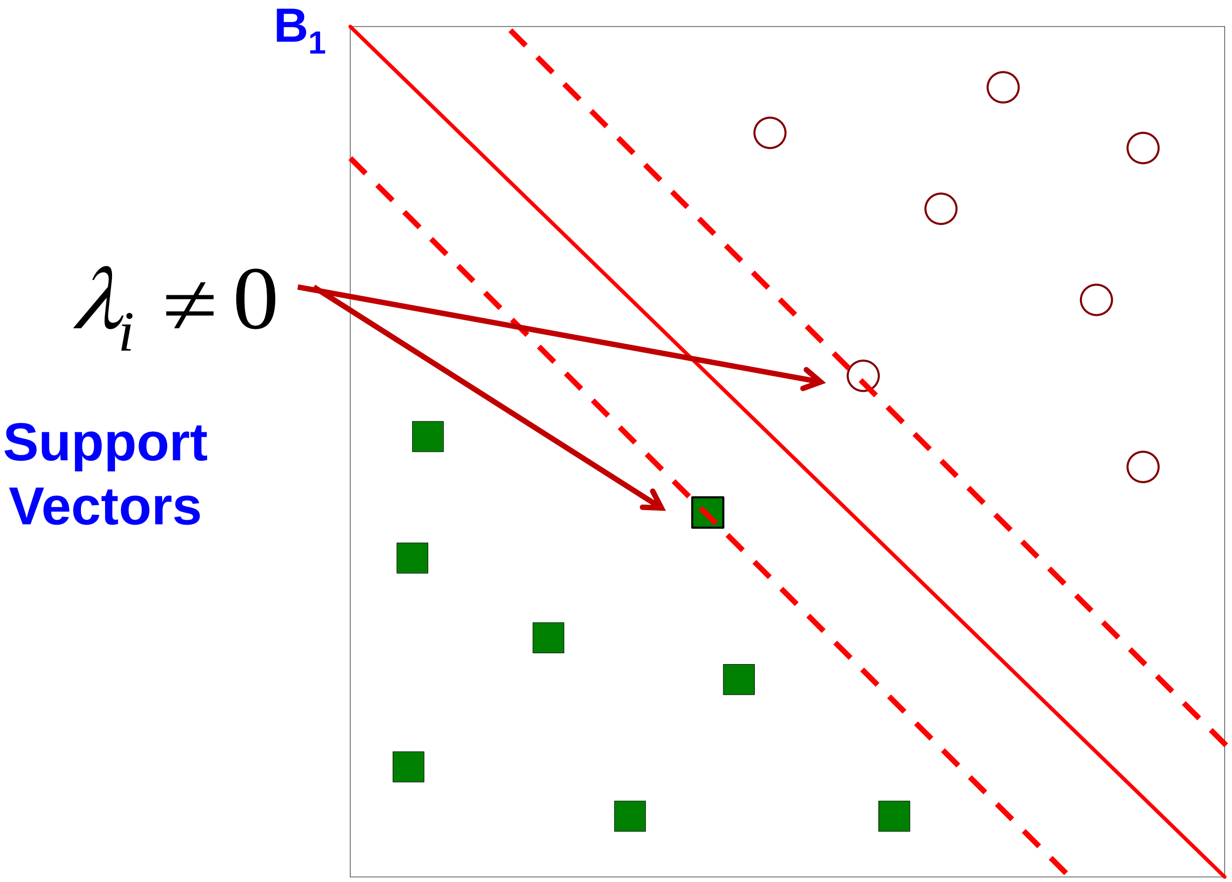
$$\lambda_i \left[y_i (\vec{w}.\vec{x}_i + b) - 1 \right] = 0$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

.

.

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

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

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

.

.

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

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***:

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

Table 1: Training Data

A_1	A_2	y	λ_i
0.38	0.47	+	65.52
0.49	0.61	-	65.52
0.92	0.41	-	0
0.74	0.89	-	0
0.18	0.58	+	0
0.41	0.35	+	0
0.93	0.81	-	0
0.21	0.10	+	0

Illustration : Linear SVM

- Consider the case of a binary classification starting with a training data of 8 tuples as shown in Table 1.
- Using quadratic programming, we can solve the KKT constraints to obtain the Lagrange multipliers λ_i for each training tuple, which is shown in Table 1.
- Note that only the first two tuples are support vectors in this case.
- Let $W = (w_1, w_2)$ and b denote the parameter to be determined now. We can solve for w_1 and w_2 as follows:

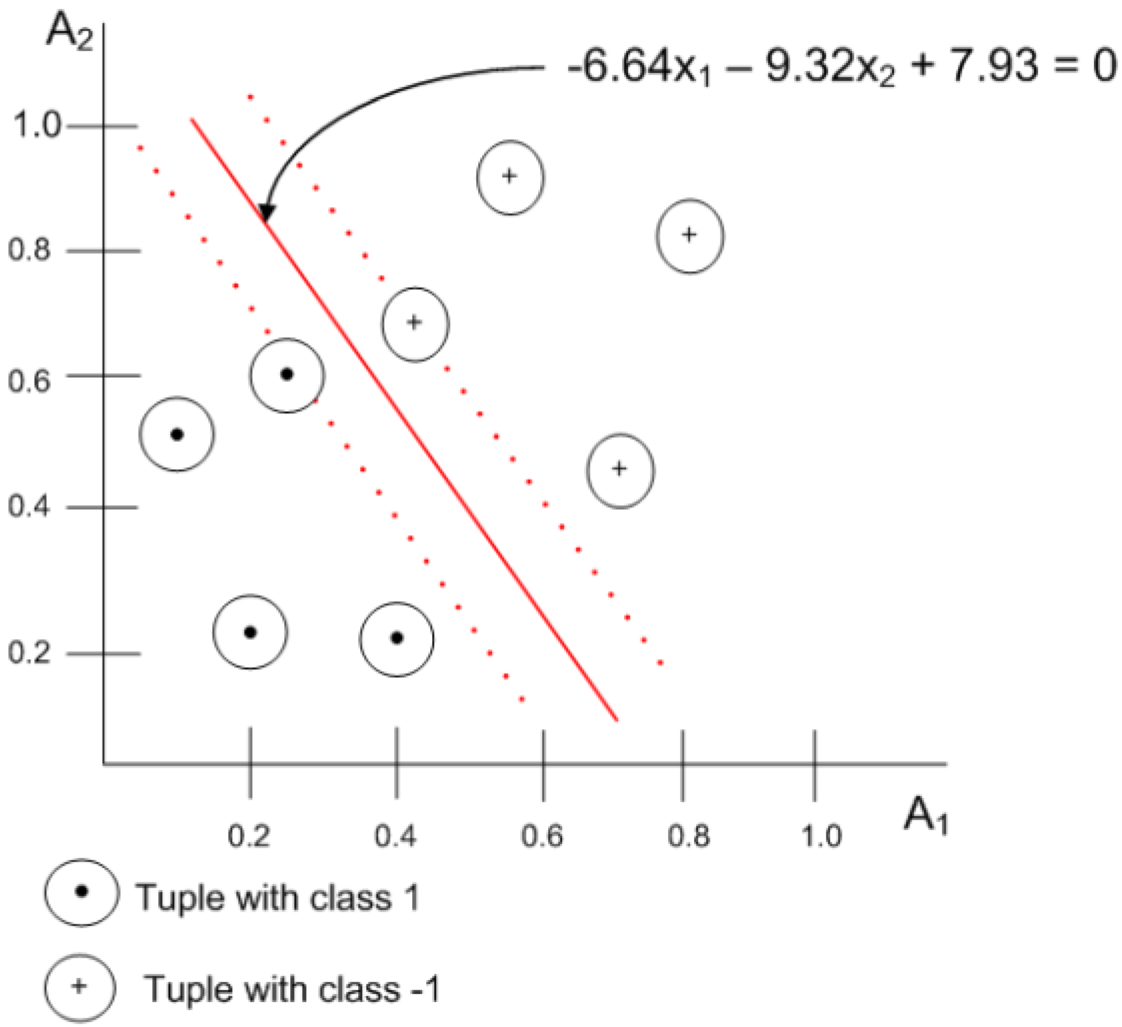
$$w_1 = \sum_i \lambda_i . y_i . x_{i1} = 65.52 \times 1 \times 0.38 + 65.52 \times -1 \times 0.49 = -6.64 \tag{22}$$

$$w_2 = \sum_i \lambda_i . y_i . x_{i2} = 65.52 \times 1 \times 0.47 + 65.52 \times -1 \times 0.61 = -9.32$$

- The parameter b can be calculated for each support vector as follows
  $b_1 = 1 - W.x_1$ // for support vector x_1
  $= 1 - (-6.64) \times 0.38 - (-9.32) \times 0.47$ //using dot product
  $= 7.93$
  $b_2 = 1 - W.x_2$ // for support vector x_2
  $= 1 - (-6.64) \times 0.48 - (-9.32) \times 0.611$ //using dot product
  $= 7.93$

- Averaging these values of b_1 and b_2, we get $b = 7.93$.

Figure 6: Linear SVM example.



- Thus, the MMH is $-6.64x_1 - 9.32x_2 + 7.93 = 0$ (also see Fig. 6).
- Suppose, test data is $X = (0.5, 0.5)$. Therefore,

$$\delta(X) = W.X + b$$
$$= -6.64 \times 0.5 - 9.32 \times 0.5 + 7.93$$
$$= -0.05$$
$$= -ve$$

- This implies that the test data falls on or below the MMH and SVM classifies that X belongs to class label -.

Support Vector Machines

- What if the problem is not linearly separable?