



# Support Vector Machines

Qinliang Su (苏勤亮)

Sun Yat-sen University

[suqliang@mail.sysu.edu.cn](mailto:suqliang@mail.sysu.edu.cn)

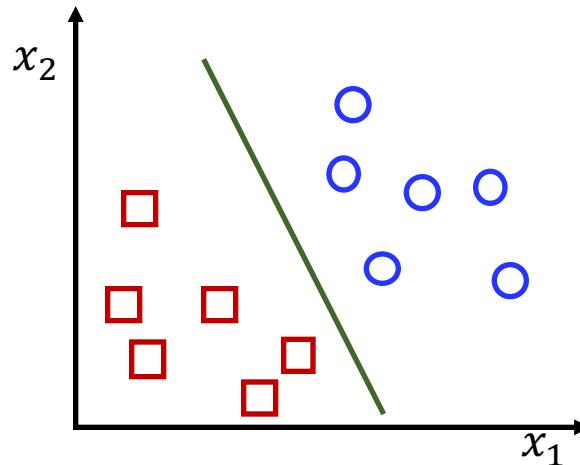
# Outline

---

- Decision Boundaries of Linear Classifiers
- Maximum-Margin Classifier
- Soft Maximum-Margin Classifier
- Support Vector Machine
- Relation to Logistic Regression

# Decision Boundaries in Linear Classifiers

- In linear classifiers, the decision boundary is always a hyperplane. The goal is to find the hyperplane that can separate different types of samples

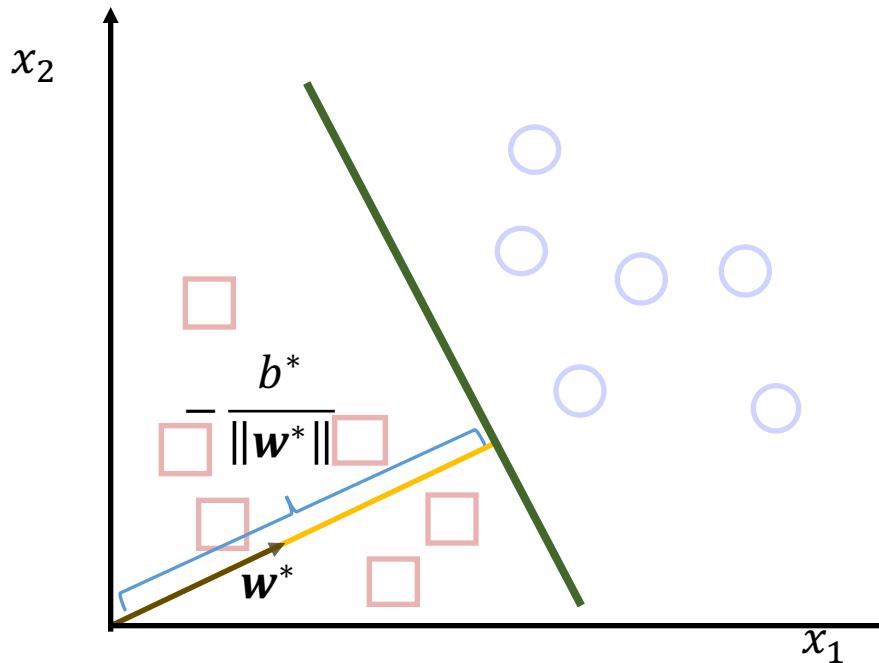


- Logistic regression
  - The decision-boundary hyperplane is found by minimizing the cross-entropy loss

$$L(\mathbf{w}, b) = -y \log(\sigma(\mathbf{w}^T \mathbf{x} + b)) - (1 - y) \log(1 - \sigma(\mathbf{w}^T \mathbf{x} + b))$$

- With the optimal  $w^*$  and  $b^*$ , the hyperplane is composed of  $x$  in

$$\{x | w^{*T}x + b^* = 0\}$$



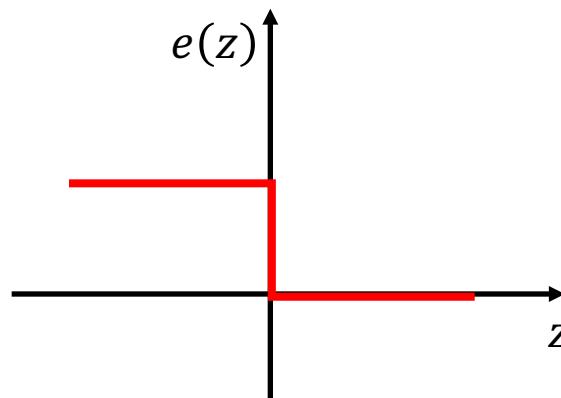
- 1) The hyperplane is *perpendicular* to the vector  $w^*$
- 2) The distance from the original point to the plane is  $-\frac{b^*}{\|w^*\|}$

- Ideal classifier
  - The hyperplane is determined by minimizing the loss

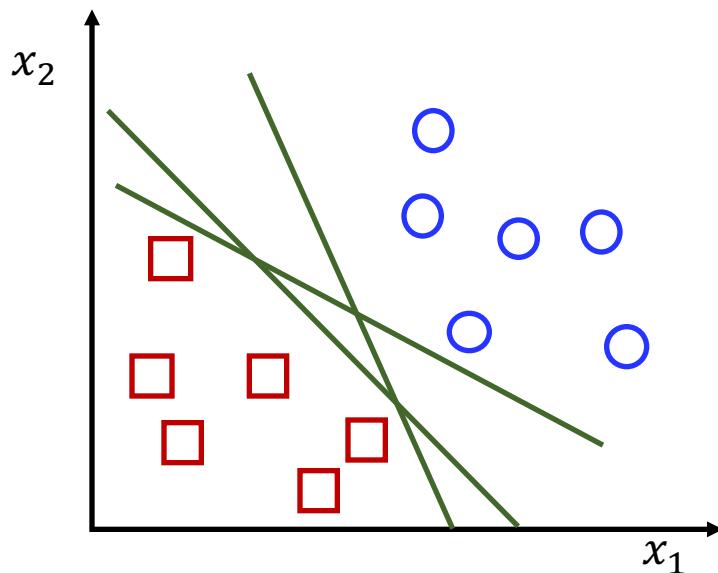
$$L(\mathbf{w}, b) = \sum_{\ell=1}^N e \left( y^{(\ell)} (\mathbf{w}^T \mathbf{x}^{(\ell)} + b) \right)$$

$L(\mathbf{w}, b)$  represents the number of misclassified samples

- $y \in \{-1, 1\}$
- $e(z) = 0 \text{ if } z \geq 0; E[z] = 1 \text{ otherwise}$



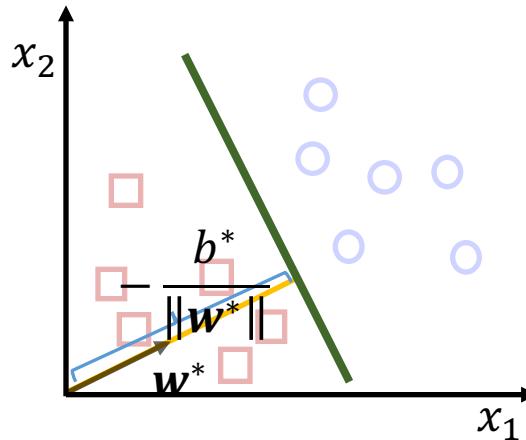
- If the samples are linearly separable, there will be numerous ideal classifiers, which are determined by  $w^*$  and  $b^*$
- Every  $w^*$  and  $b^*$  corresponds to a hyperplane



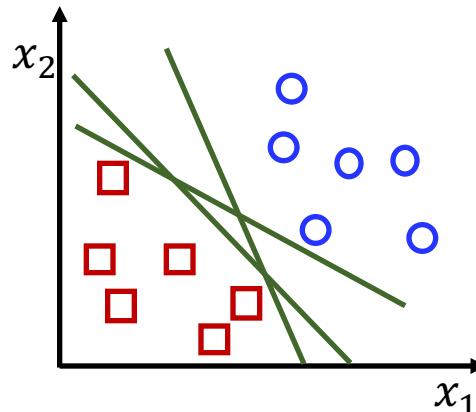
*All the hyperplane can have the loss reduced to zero*

# Which Hyperplane is the Best?

- The hyperplane in logistic regression is optimal from the perspective of *minimizing the cross-entropy loss*

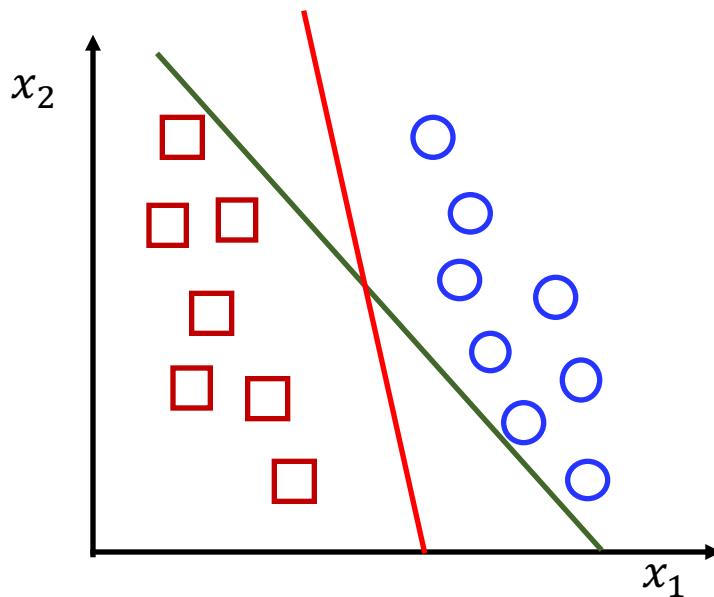


- The hyperplanes in the ideal classifier is optimal from the perspective of *minimizing the number of misclassified samples*



- These hyperplanes are all evaluated on the *training samples*
- But what we need is the performance on the *(unseen) test data*

According to our intuitions, which of the hyperplanes below would perform better on the test data?



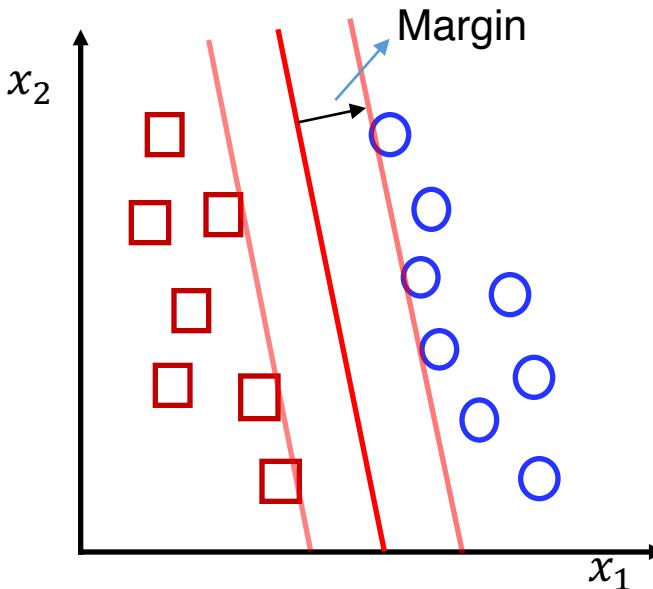
# Outline

---

- Decision Boundaries of Linear Classifiers
- Maximum-Margin Classifier
- Soft Maximum-Margin Classifier
- Support Vector Machine
- Relation to Logistic Regression

# The Maximum-Margin Objective

- To perform well on unseen data, the intuition is to **find a hyperplane that makes the margin as large as possible**



Thanks to the large margin, *it can be expected that an unseen sample is more likely to be classified correctly under such a decision boundary*

# How to Represent the Margin?

- The distance from the sample  $x$  to the hyperplane  $\mathcal{H}$

- Every  $x$  can be decomposed as

$$x = \mathbf{m}_1 + \mathbf{m}_2$$

- $\mathbf{m}_1$  is on the  $\mathcal{H}$ , i.e.,  $\mathbf{w}^T \mathbf{m}_1 + b = 0$
  - $\mathbf{m}_2 \perp \mathcal{H}$  and  $\mathbf{m}_2 \parallel \mathbf{w}$

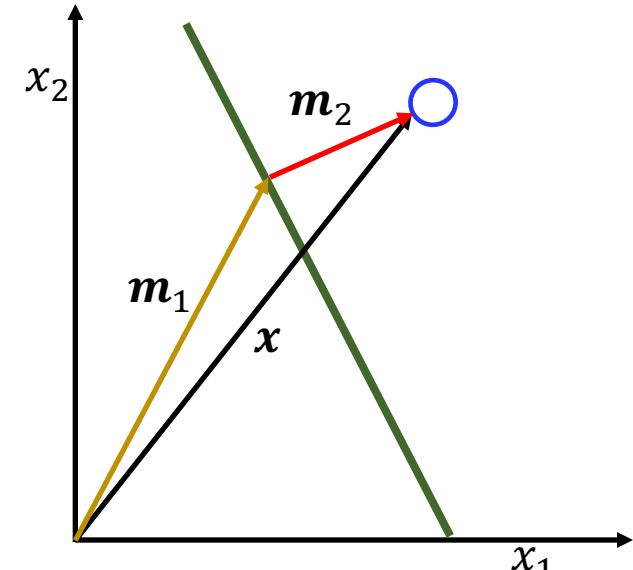
- Thus, we have

$$\begin{aligned} h(x) &\triangleq \mathbf{w}^T x + b = \mathbf{w}^T(\mathbf{m}_1 + \mathbf{m}_2) + b \\ &= \mathbf{w}^T \mathbf{m}_2 \end{aligned}$$

- Due to  $\mathbf{m}_2 \parallel \mathbf{w}$ , we can write

$$\mathbf{m}_2 = \gamma \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|},$$

with  $|\gamma|$  representing the length of  $\mathbf{m}_2$

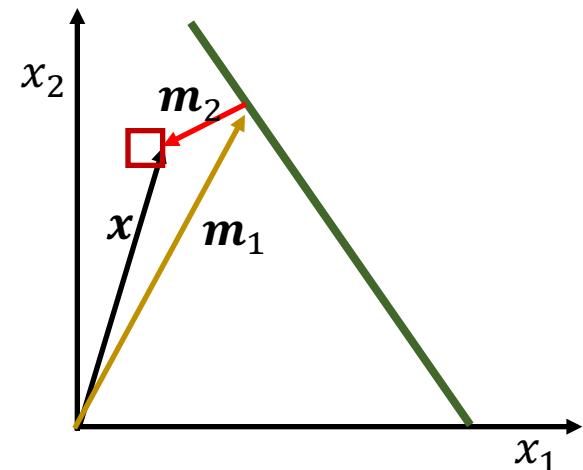


- Substituting  $\mathbf{m}_2 = \gamma \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}$  into  $h(\mathbf{x}) = \mathbf{w}^T \mathbf{m}_2$  gives

$$h(\mathbf{x}) = \gamma \cdot \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|} \quad \Rightarrow \quad \gamma = \frac{h(\mathbf{x})}{\|\mathbf{w}\|}$$

- The distance of a sample on the other side of the hyperplane is

$$\gamma = -\frac{h(\mathbf{x})}{\|\mathbf{w}\|}$$



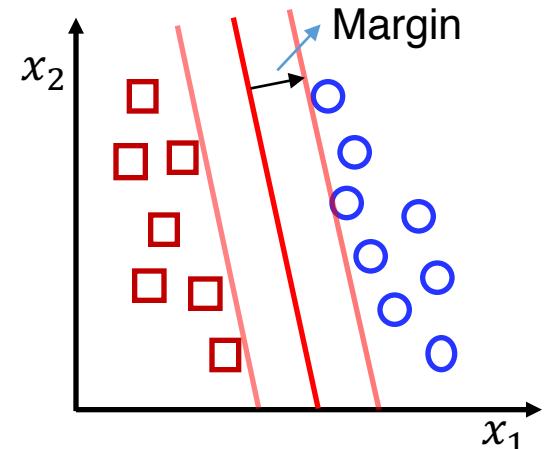
- The distance of a sample  $(\mathbf{x}, y)$  to the hyperplane is given by

$$\frac{y \cdot h(\mathbf{x})}{\|\mathbf{w}\|} = \frac{y \cdot (\mathbf{w}^T \mathbf{x} + b)}{\|\mathbf{w}\|}$$

where  $y \in \{-1, 1\}$

- The margin of a hyperplane under a dataset is given by the minimum distance, i.e.,

$$\text{Margin} = \min_{\ell} \frac{y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)}{\|\mathbf{w}\|}$$



- Thus, the maximum-margin classifier is to find the  $\mathbf{w}^*$  and  $b^*$  that maximize the margin, i.e.,

$$\mathbf{w}^*, b^* = \arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)] \right\}$$

*But how to optimize is unknown*

# The Transformed Objective Function

- Optimizing another objective function that shares the same optima as the original problem
  - If  $\mathbf{w}^*$  and  $b^*$  is the optima to  $\max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)] \right\}$ , then  $\kappa \mathbf{w}^*$  and  $\kappa b^*$  is also the optimal for all  $\kappa \in \mathbb{R}$
  - There must exist optima  $\kappa \mathbf{w}^*$  and  $\kappa b^*$  that satisfy the constraints

$$y^{(\ell)} \cdot (\kappa \mathbf{w}^{*T} \mathbf{x}^{(\ell)} + \kappa b^*) \geq 1$$

- Among all  $\mathbf{w}, b$  satisfying  $y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b) \geq 1$ , *the one with the smallest  $\|\mathbf{w}\|^2$  must be one of the optima*

Otherwise, another  $\mathbf{w}, b$  that correspond to larger  $\frac{1}{\|\mathbf{w}\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)]$  can be found

- Therefore, the maximum-margin hyperplane can be found by solving the optimization problem below

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$s.t.: y^{(\ell)} \cdot (w^T x^{(\ell)} + b) \geq 1, \quad \text{for } \ell = 1, 2, \dots, N$$

- This is a quadratic optimization problem. Its optimal solution can be found by *numerical methods* efficiently
- With the optimal  $w^*$  and  $b^*$ , an unseen data  $x$  can be classified as

$$\hat{y}(x) = \text{sign}(w^{*T} x + b^*)$$

# The Equivalent Dual Formulation

---

- Every convex optimization problem corresponds to an equivalent dual formulation

All contents in this section are extracted from the subject of  
**convex optimization**

- The **Lagrangian function** of the original optimization problem

$$\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{\ell=1}^N \mathbf{a}_\ell (y^{(\ell)} (\mathbf{w}^T \mathbf{x}^{(\ell)} + b) - 1),$$

where the Lagrange multiplier  $\mathbf{a}_\ell$  is required to satisfy  $\mathbf{a}_\ell \geq 0$

- The **Lagrange dual function**

$$g(\mathbf{a}) = \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \mathbf{a})$$

- The **dual formulation** of the original optimization problem

$$\begin{aligned} & \max_{\mathbf{a}} g(\mathbf{a}) \\ & \text{s.t.: } \mathbf{a} \geq \mathbf{0} \end{aligned}$$

- Deriving the close-form expression of function  $g(\mathbf{a})$
- Setting the gradient  $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$  and  $\frac{\partial \mathcal{L}}{\partial b} = 0$ , which gives

$$\mathbf{w} = \sum_{\ell=1}^N a_\ell y^{(\ell)} \mathbf{x}^{(\ell)} \quad \sum_{\ell=1}^N a_\ell y^{(\ell)} = 0$$

- Substituting them into  $\mathcal{L}(\mathbf{w}, b, \mathbf{a})$  gives  $g(\mathbf{a}) = \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \mathbf{a})$  as

$$g(\mathbf{a}) = \sum_{\ell=1}^N a_\ell - \frac{1}{2} \sum_{\ell=1}^N \sum_{j=1}^N a_\ell a_j y^{(\ell)} y^{(j)} \mathbf{x}^{(\ell)T} \mathbf{x}^{(j)}$$

- Then, the dual optimization becomes

$$\begin{aligned} & \max_{\boldsymbol{a}} g(\boldsymbol{a}) \\ & \text{s.t. : } \boldsymbol{a} \geq \mathbf{0} \text{ and } \sum_{\ell=1}^N a_{\ell} y^{(\ell)} = 0 \end{aligned}$$

where  $g(\boldsymbol{a}) = \sum_{\ell=1}^N a_{\ell} - \frac{1}{2} \sum_{\ell=1}^N \sum_{j=1}^N a_{\ell} a_j y^{(\ell)} y^{(j)} \mathbf{x}^{(\ell)T} \mathbf{x}^{(j)}$

It is a quadratic optimization, and can be solved by *numerical methods*

- Relation between optima  $\mathbf{w}^*$ ,  $b^*$  and optima  $\mathbf{a}^*$ 
  - With the optima  $\mathbf{a}^*$ , according to  $\mathbf{w} = \sum_{\ell=1}^N a_\ell y^{(\ell)} \mathbf{x}^{(\ell)}$ , the optimal  $\mathbf{w}^*$  is equal to

$$\mathbf{w}^* = \sum_{\ell=1}^N a_\ell^* y^{(\ell)} \mathbf{x}^{(\ell)}$$

- Due to  $y^{(\ell)}(\mathbf{w}^{*T} \mathbf{x}^{(\ell)} + b) = 1$  for all samples  $(\mathbf{x}^{(\ell)}, y^{(\ell)})$  that are on the margin, we can derive that

$$b^* = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left( y^{(n)} - \sum_m a_m^* y^{(m)} \mathbf{x}^{(n)T} \mathbf{x}^{(m)} \right)$$

- Maximum-margin classifiers

- Primal version

$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \mathbf{x} + b^*)$$

- Dual version

Substituting  $\mathbf{w}^* = \sum_{n=1}^N a_n^* y^{(n)} \mathbf{x}^{(n)}$  into the primal version gives

$$\hat{y}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^N a_n^* y^{(n)} \mathbf{x}^{(n)T} \mathbf{x} + b^*\right)$$

The two classifiers are *equivalent*

$$\hat{y}(\mathbf{x}) = \text{sign} \left( \sum_{n=1}^N \left( a_n^* (\mathbf{x}^{(n)T} \mathbf{x}) \right) \cdot y^{(n)} + b^* \right)$$

- How to understand the dual maximum-margin classifier?
  - For a test  $\mathbf{x}$ , computing its similarity with all the training samples  $\mathbf{x}^{(n)}$  for  $n = 1, \dots, N$  by  $\mathbf{x}^{(n)T} \mathbf{x}$
  - Summing all the labels  $y^{(n)}$  weighted by the sample similarity  $\mathbf{x}^{(n)T} \mathbf{x}$  and the multiplier  $a_n^*$

# Comparisons on the Primal and Dual Results

- Optimization complexity

Primal

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

$$s.t.: y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b) \geq 1, \\ \text{for } \ell = 1, 2, \dots, N$$

# of parameter to optimize:  
**dimension of features**

Dual

$$\max_{\mathbf{a}} g(\mathbf{a})$$

$$s.t.: \mathbf{a} \geq \mathbf{0}$$

$$\sum_{\ell=1}^N a_\ell y^{(\ell)} = 0$$

# of parameter to optimize:  
**# of training samples**

In *high-dimensional feature case*, solving the dual problem is more efficient

- Testing complexity

Primal

$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \mathbf{x} + b^*)$$

Just need **one** inner-product  $\mathbf{w}^{*T} \mathbf{x}$

Dual

$$\hat{y}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^N a_n^* y^{(n)} \mathbf{x}^{(n)T} \mathbf{x} + b^*\right)$$

Need  **$N$**  inner-products  $\mathbf{x}^{(n)T} \mathbf{x}$  for  $n = 1, 2, \dots, N$

At the first glance, the dual classifier looks much more expensive than the primal one

- But it can be shown that **most of  $a_n^*$  are 0**

# Sparsity in the Lagrange Multiplier $a^*$

- For any convex optimization problem, the optima satisfies the *KKT conditions*, which, for our problem, are

$$a_n^* \geq 0$$

$$y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) - 1 \geq 0$$

$$a_n^* [y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) - 1] = 0$$

The first two conditions come from the original primal and dual problems

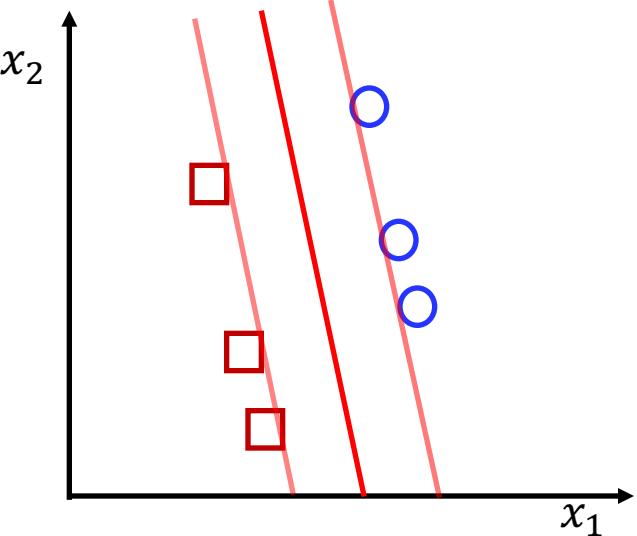
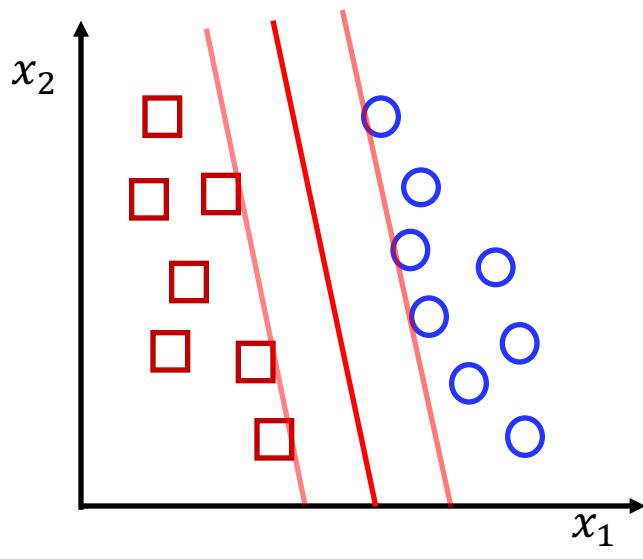
- From the last condition, we can see that  $a_n^* \neq 0$  only when  $y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) = 1$
- If  $\mathbf{x}^{(n)}$  satisfies  $y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) = 1$ , it means that it lies on the margin

This kind of samples are called *support vectors*

- Thus, when we classify an unseen sample  $x$  as

$$\hat{y}(x) = \text{sign} \left( \sum_n a_n^* y^{(n)} \mathbf{x}^{(n)T} \mathbf{x} + b^* \right),$$

we only need to evaluate the similarity  $\mathbf{x}^{(n)T} \mathbf{x}$  between  $x$  and the support vectors (samples)



# Outline

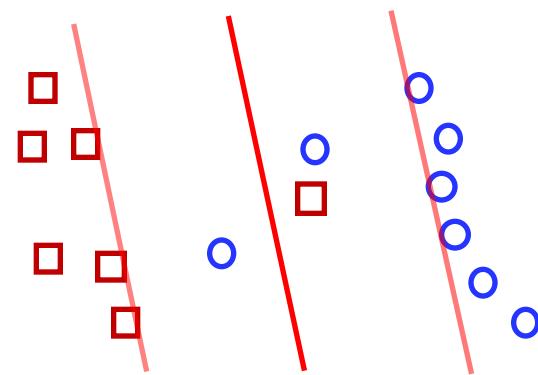
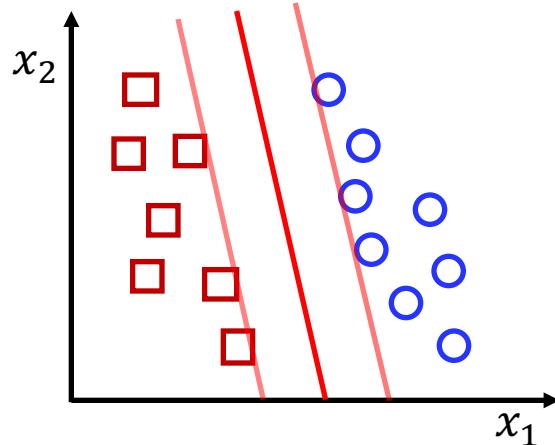
---

- Decision Boundaries of Linear Classifiers
- Maximum-Margin Classifier
- Soft Maximum-Margin Classifier
- Support Vector Machine
- Relation to Logistic Regression

# Non-separable Classes

- The assumption used in the previous maximum-margin classifier

The training samples are linearly separable!!!



- However, *such a hyperplane may not exist*. That is, there is no feasible solution to the optimization problem

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

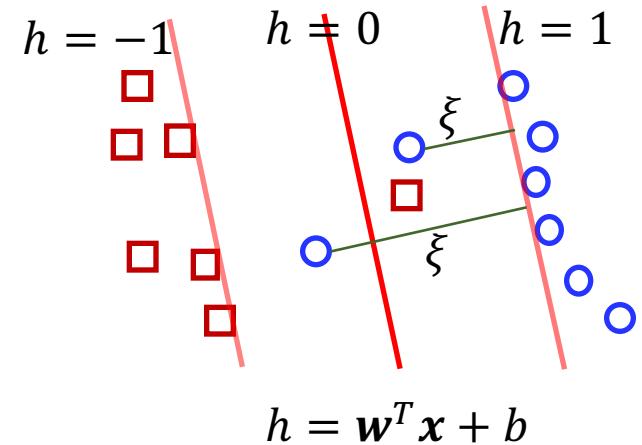
$$s. t.: y^{(n)} \cdot (\mathbf{w}^T \mathbf{x}^{(n)} + b) \geq 1, \quad \text{for } n = 1, 2, \dots, N$$

# Soft Maximum Margin

- To address the issue, instead of requiring  $y^{(n)} \cdot (\mathbf{w}^T \mathbf{x}^{(n)} + b) \geq 1$  for all  $n = 1, \dots, N$ , we only require

$$y^{(n)} \cdot (\mathbf{w}^T \mathbf{x}^{(n)} + b) \geq 1 - \xi_n$$

where  $\xi_n$  is slack variable and  $\xi_n \geq 0$



- The objective is not just to minimize  $\frac{1}{2} \|\mathbf{w}\|^2$ , but also need to minimize the sum of  $\xi_n$ , which leads to the objective

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

where  $C$  is used to control the relative importance

- The optimization problem now becomes

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n$$

$$s.t.: y^{(n)} \cdot (\mathbf{w}^T \mathbf{x}^{(n)} + b) \geq 1 - \xi_n,$$

$$\xi_n \geq 0, \quad \text{for } n = 1, 2, \dots, N$$

- Using the same method as before, the dual formulation can be derived as

$$\max_{\mathbf{a}} g(\mathbf{a})$$

$$s.t.: a_n \geq 0, \quad a_n \leq C$$

$$\sum_{n=1}^N a_n y^{(n)} = 0$$

where  $g(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m y^{(n)} y^{(m)} \mathbf{x}^{(n)T} \mathbf{x}^{(m)}$

- With the optima  $w^*$  and  $b^*$ , a sample  $x$  is classified as

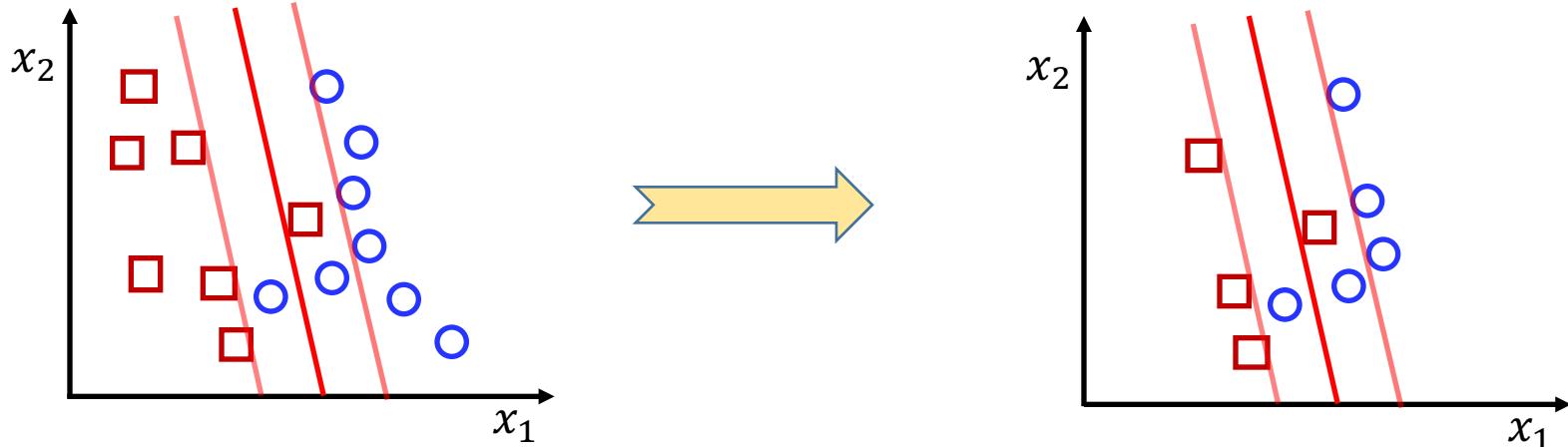
$$\hat{y}(x) = \text{sign}(w^{*T}x + b^*)$$

- With the optima  $a^*$ , a sample  $x$  is classified as

$$\hat{y}(x) = \text{sign}\left(\sum_{n=1}^N a_n^* y^{(n)} x^{(n)T} x + b^*\right)$$

Also, the two classifiers above are *equivalent*

- The optima  $a^*$  is sparse, with only elements within the margin being nonzero



# Outline

---

- Decision Boundaries of Linear Classifiers
- Maximum-Margin Classifier
- Soft Maximum-Margin Classifier
- **Support Vector Machine**
- Relation to Logistic Regression

# Non-linearization

---

- The maximum-margin classifiers so far are **linear**
- To extend to the nonlinear scenarios, we can transform the original feature  $x$  to another space via the **basis function**

$$\phi: x \rightarrow \phi(x)$$

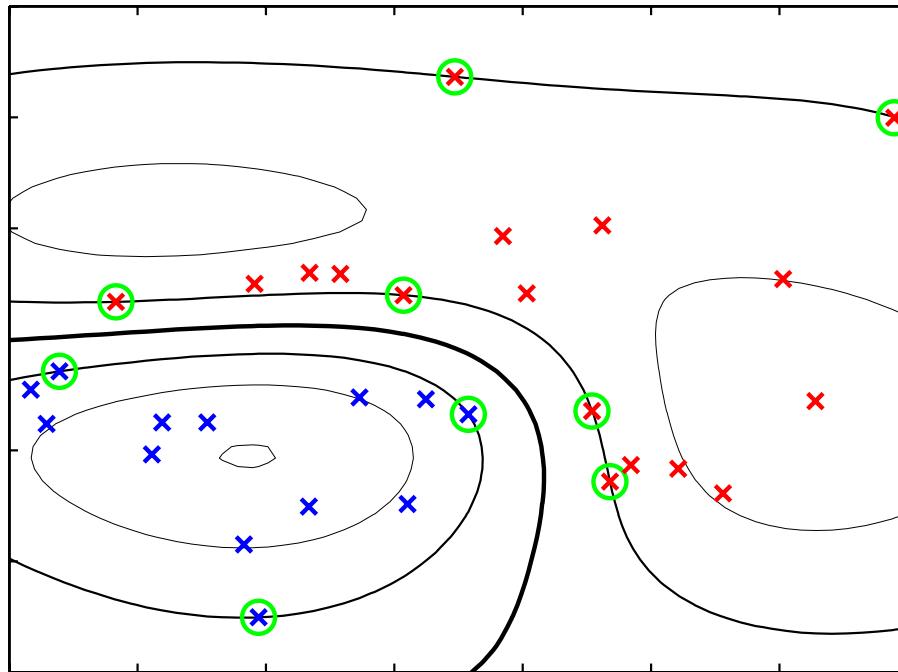
- The primal maximum-margin optimization problem becomes

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n$$

$$s.t.: y^{(n)} \cdot (w^T \phi(x^{(n)}) + b) \geq 1 - \xi_n,$$

$$\xi_n \geq 0, \quad \text{for } n = 1, 2, \dots, N$$

Classifier:  $\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \boldsymbol{\phi}(\mathbf{x}^{(n)}) + b^*)$



- Intuitively, data is easier to be separated in high-dimensional space
- *To achieve better performance, we prefer the dimension of the transformed feature space  $\phi(x^{(n)})$  to be as high as possible*
- However, The dimension of basis function  $\phi(x)$  *cannot be too high*, otherwise the optimization would be very expensive

$$\begin{aligned}
 & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \\
 & \text{s.t.: } y^{(n)} \cdot (\mathbf{w}^T \phi(x^{(n)}) + b) \geq 1 - \xi_n, \\
 & \quad \xi_n \geq 0, \quad \text{for } n = 1, 2, \dots, N
 \end{aligned}$$

- The problem can be solved via its dual form

$$\max_{\mathbf{a}} g(\mathbf{a})$$

$$s.t.: \quad a_n \geq 0, \quad a_n \leq C$$

$$\sum_{n=1}^N a_n y^{(n)} = 0$$

where  $g(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m y^{(n)} y^{(m)} \boldsymbol{\phi}(\mathbf{x}^{(n)})^T \boldsymbol{\phi}(\mathbf{x}^{(m)})$

Classifier:  $\hat{y}(\mathbf{x}) = \text{sign} \left( \sum_{n=1}^N a_n^* y^{(n)} \boldsymbol{\phi}(\mathbf{x}^{(n)})^T \boldsymbol{\phi}(\mathbf{x}) + b^* \right)$

- The dimension of  $\mathbf{a}$  is *independent of the dimension of  $\boldsymbol{\phi}(\cdot)$* , thus the dual form is able to work in very large feature space  $\boldsymbol{\phi}(\cdot)$

The dual formulation requires to evaluate the inner product  
$$\phi(x^{(n)})^T \phi(x),$$
 which is expensive in high-dimensional case

The issue can be addressed by using the *kernel trick*

# Kernel Function

---

- A kernel function is a two-variable function  $k(x, x')$  that can be expressed as an inner production of some function  $\phi(\cdot)$

$$k(x, x') = \phi(x)^T \phi(x')$$

Obviously,  $x^T x'$  and  $\phi(x)^T \phi(x')$  are kernel functions

- **Mercer Theorem:** If a function  $k(x, x')$  is symmetric positive definite, *i.e.*,

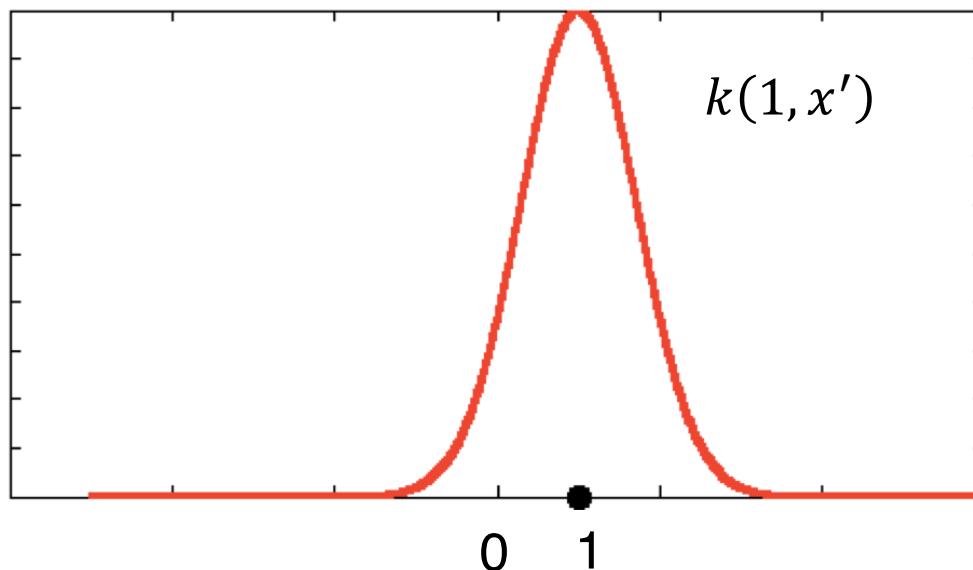
$$\int \int g(x)k(x, y)g(y)dx dy \geq 0, \quad \forall g(\cdot) \in L^2,$$

there must exist a function  $\phi(\cdot)$  such that  $k(x, x') = \phi(x)^T \phi(x')$

If a function  $k(x, x')$  satisfies the symmetric positive definite condition, it must be a kernel function

- One of the most widely used kernel is the Gaussian kernel, which takes the form

$$k(x, x') = \exp \left\{ -\frac{1}{2\sigma^2} \|x - x'\|^2 \right\}$$



- The function  $\phi(\cdot)$  of Gaussian kernel has infinite dimensions

# Kernel Trick

---

- With the kernel function, the dual maximum-margin classifier can be rewritten as

$$\max_{\mathbf{a}} g(\mathbf{a})$$

$$s.t.: \quad a_n \geq 0, \quad a_n \leq C$$

$$\sum_{n=1}^N a_n y^{(n)} = 0$$

$$\text{where } g(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m y^{(n)} y^{(m)} \mathbf{k}(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

- The induced classifier

$$\hat{y}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^N a_n^* y^{(n)} \mathbf{k}(\mathbf{x}^{(n)}, \mathbf{x}) + b^*\right)$$

**Kernel trick:** replacing the  $\phi(\mathbf{x})^T \phi(\mathbf{x}')$  with the kernel function  $k(\mathbf{x}, \mathbf{x}')$

- The conclusions can be summarized as
  - If  $k(x, x') = x^T x'$ , it is a linear maximum-margin classifier
  - If  $k(x, x') = \phi(x)^T \phi(x')$ , it is a *finite-dimensional* nonlinear maximum-margin classifier based on basis functions
  - If  $k(x, x') = \exp\left\{-\frac{1}{2\sigma^2}\|x - x'\|^2\right\}$ , it is a *infinite-dimensional* nonlinear maximum-margin classifier

# Outline

---

- Decision Boundaries of Linear Classifiers
- Maximum-Margin Classifier
- Soft Maximum-Margin Classifier
- Support Vector Machine
- Relation to Logistic Regression

- In the logistic regression, we minimize the loss

$$\begin{aligned}
 L(\mathbf{w}, b) &= - \sum_{n=1}^N \left[ y^{(n)} \log \sigma(h^{(n)}) + (1 - y^{(n)}) \left( 1 - \sigma(h^{(n)}) \right) \right] + \lambda \|\mathbf{w}\|^2 \\
 &= \sum_{n=1}^N \log(1 + \exp(-y^{(n)}h^{(n)})) + \lambda \|\mathbf{w}\|^2 \\
 &= \sum_{n=1}^N E_{LR}(y^{(n)}h^{(n)}) + \lambda \|\mathbf{w}\|^2
 \end{aligned}$$

where  $E_{LR}(z) = \log(1 + \exp(-z))$

- In the ideal classifier, we minimize the loss

$$L(\mathbf{w}, b) = \sum_{n=1}^N E_{Ideal}(y^{(n)}h^{(n)}) + \lambda \|\mathbf{w}\|^2$$

where  $E_{Ideal}(z) = 0 \text{ if } z \geq 0; 1 \text{ otherwise}$

- In the linear maximum-margin classifier, we are equivalently minimizing the loss

$$L(\mathbf{w}, b) = \sum_{n=1}^N E_\infty(y^{(n)} h^{(n)} - 1) + \frac{1}{2} \|\mathbf{w}\|^2$$

where  $E_\infty(z) = 0$  if  $z \geq 0$ ;  $+\infty$  otherwise

- In the soft linear maximum-margin classifier, we are equivalently minimizing the loss

$$\begin{aligned} L(\mathbf{w}, b) &= C \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)}) + \frac{1}{2} \|\mathbf{w}\|^2 \\ &= \sum_{n=1}^N E_{SV}(y^{(n)} h^{(n)}) + \lambda \|\mathbf{w}\|^2 \end{aligned}$$

where  $E_{SV}(z) = \max(0, 1 - z)$ , which is called the *hinge loss*

- So, we can see that the four classifiers can be formulated under the same framework, with the only difference coming from the chosen error function
- The plot of the four error functions

