



# Support Vector Machines

Qinliang Su (苏勤亮)

Sun Yat-sen University

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

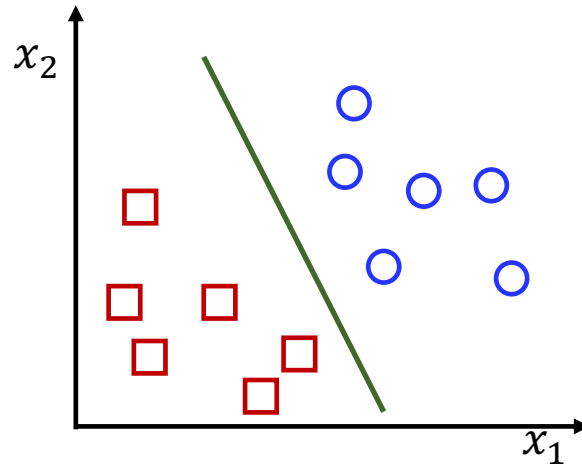
# Outline

---

- Decision Boundaries of Linear Classifiers
- Linear Maximum-Margin Classifier
- Soft Linear 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

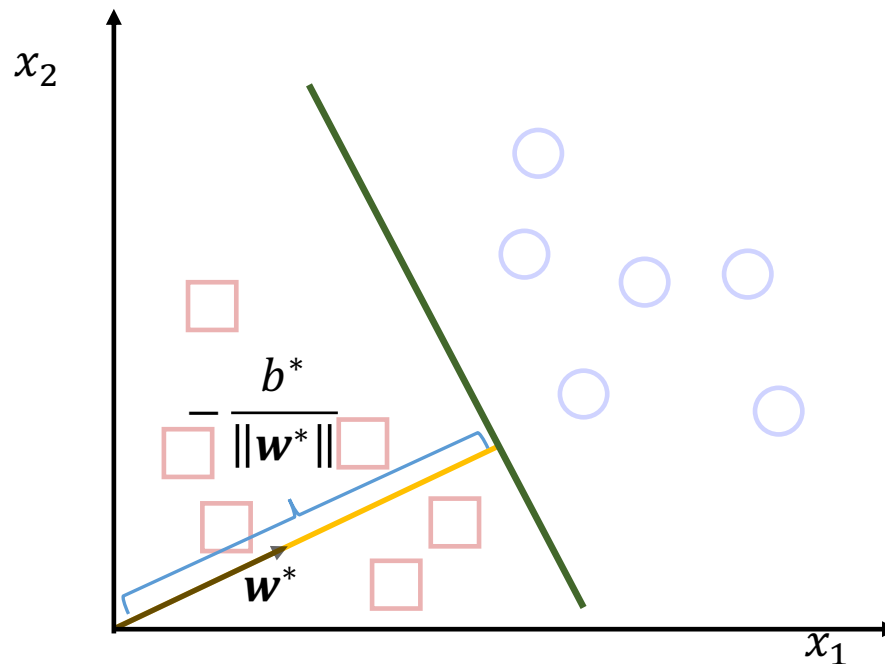


- Logistic regression
  - The decision-boundary hyperplane is found by minimizing the cross-entropy loss

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

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

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



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

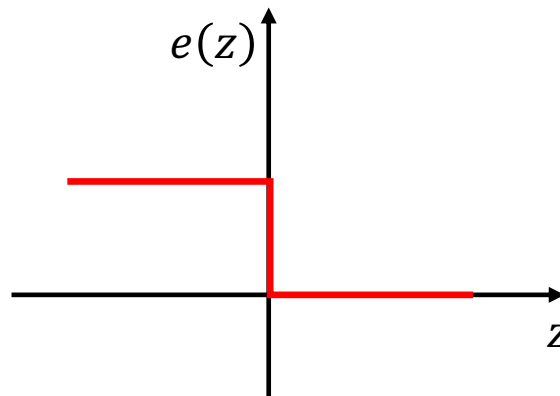
- Ideal classifier

➤ The hyperplane is determined by minimizing the loss

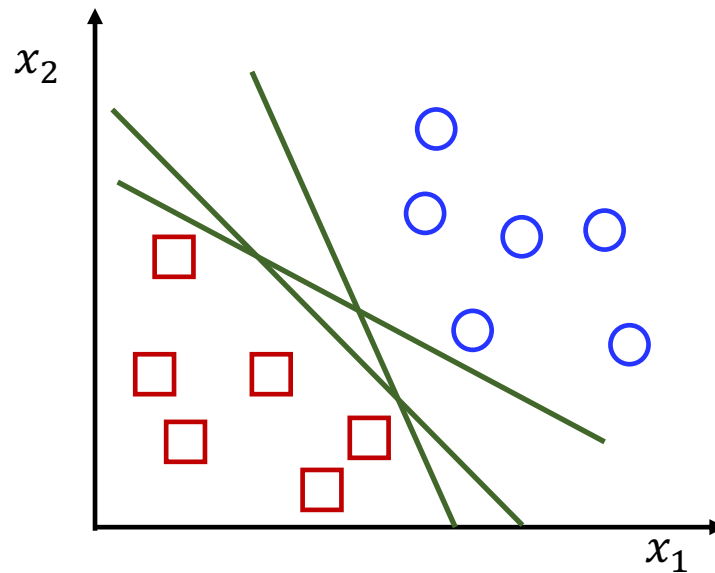
$$L(\mathbf{w}, b) = \sum_{\ell=1}^N e\left(y^{(\ell)}(\mathbf{w}^T \mathbf{x}^{(\ell)} + b)\right)$$

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

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



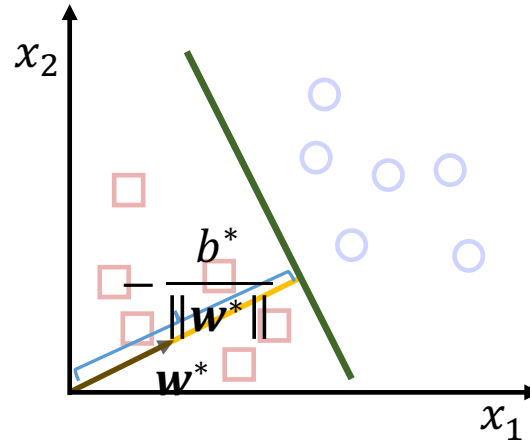
- If the samples are linearly separable, there will be numerous ideal classifiers, which are determined by  $\mathbf{w}^*$  and  $b^*$
- Every  $\mathbf{w}^*$  and  $b^*$  corresponds to a hyperplane



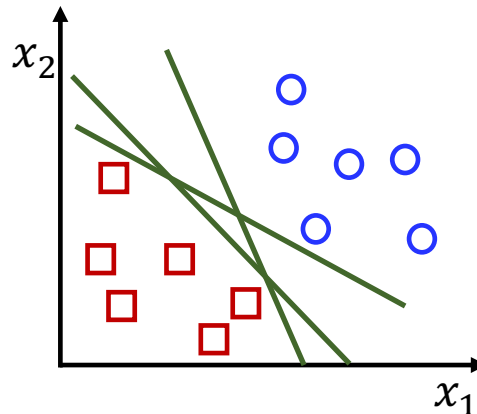
*All of hyperplanes above can have the loss reduced to zero*

# Which Hyperplane is the Best?

- The hyperplane below is optimal from the perspective of *minimizing the cross-entropy loss*

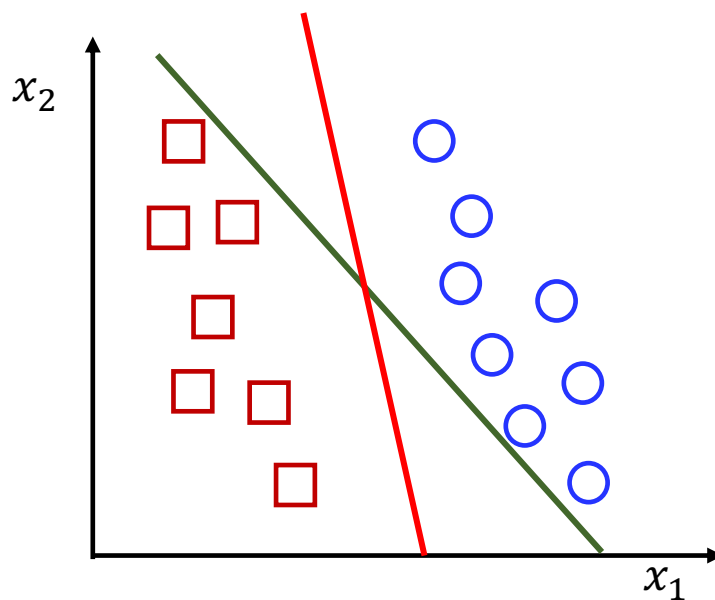


- The hyperplanes below are optimal from the perspective of *minimizing the number of misclassified samples*



- 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 hyperplane below would more likely produce better results on test data?





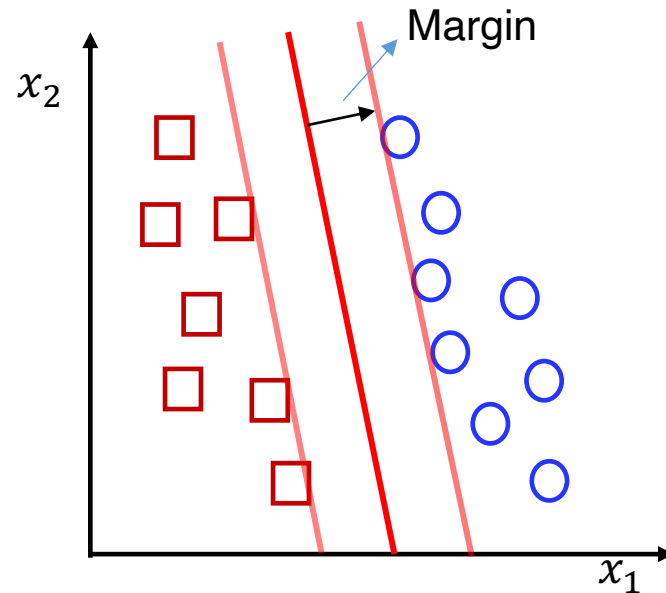
# Outline

---

- Decision Boundaries of Linear Classifiers
- **Linear Maximum-Margin Classifier**
- Soft Linear 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 enlarge the margin as much as possible**



When the margin is large, we can expect that *an unseen sample has a higher chance to be categorized correctly*

# How to Represent the Margin?

- The distance from the sample  $x$  to the hyperplane  $\mathcal{H}$ . Denote  $\mathbf{w}^T \mathbf{x} + b = h(\mathbf{x})$

- Every  $x$  can be decomposed as

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

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

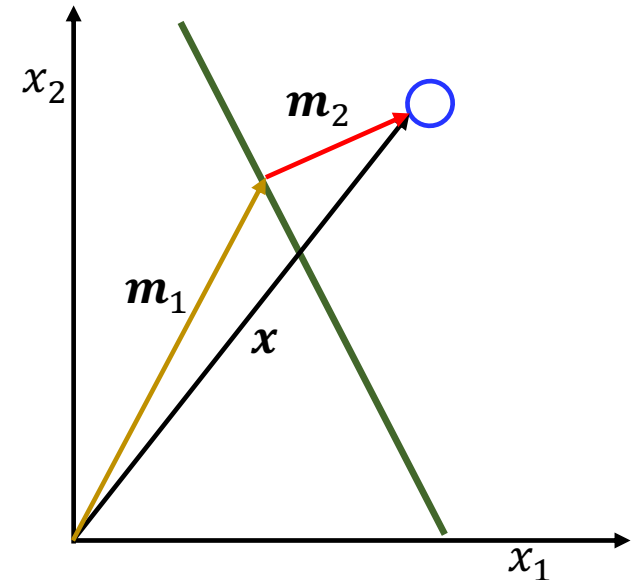
- Thus, we have

$$\mathbf{w}^T \mathbf{x} + b = \mathbf{w}^T (\mathbf{m}_1 + \mathbf{m}_2) + b = \mathbf{w}^T \mathbf{m}_2 = h(\mathbf{x})$$

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

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

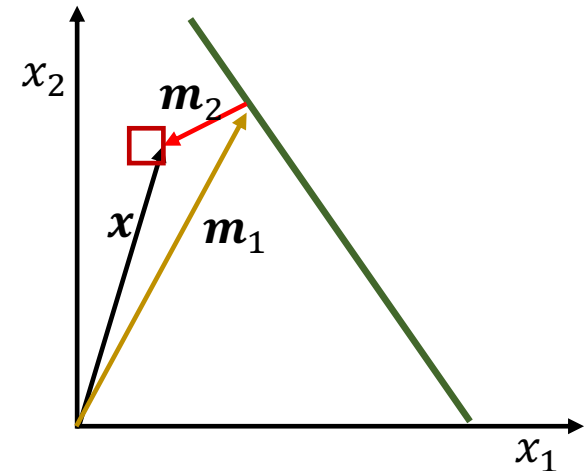
with  $|\gamma|$  representing the length of  $\mathbf{m}_2$



- 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  $\mathbf{x}$  on the upper side to the hyperplane is  $\frac{h(\mathbf{x})}{\|\mathbf{w}\|}$
- The distance of a sample  $\mathbf{x}$  on the lower side to the hyperplane is  $-\frac{h(\mathbf{x})}{\|\mathbf{w}\|}$



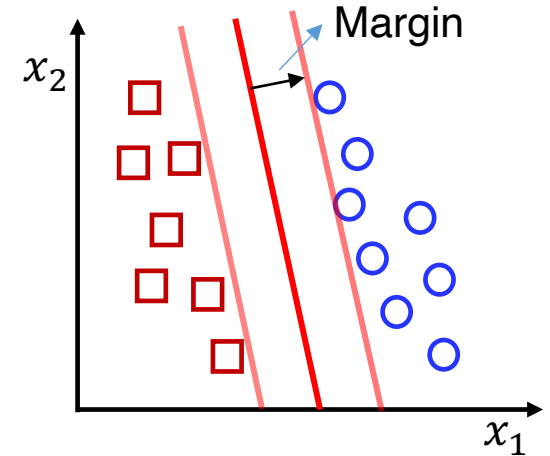
- The distance of a sample  $(\mathbf{x}, y)$  to the hyperplane is given by

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

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 a  $\mathbf{w}^*$  and  $b^*$  such that it maximizes the margin, *i.e.*,

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

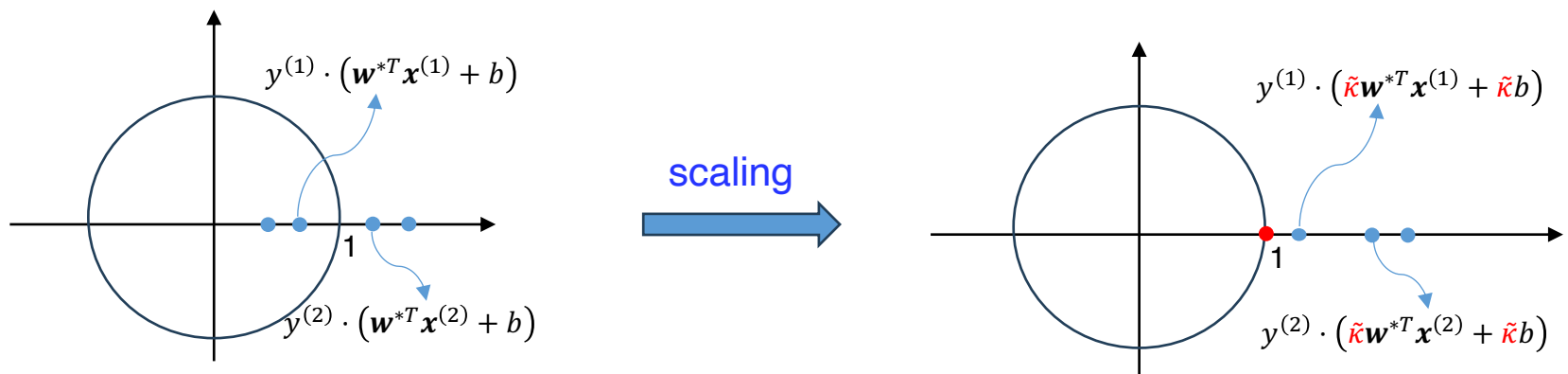
*But how to optimize is unknown*

# The Transformed Objective Function

- **Basic idea:** Optimizing an objective function that shares the same optima as the original problem
  - Suppose  $\mathbf{w}^*$  and  $b^*$  is an optima of  $\frac{1}{\|\mathbf{w}\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)]$ . Then,  $\kappa \mathbf{w}^*$  and  $\kappa b^*$  must also be an optima for all  $\kappa \neq 0$
  - Moreover, there always exists a specific  $\tilde{\kappa}$  such that

$$y^{(\ell)} \cdot (\tilde{\kappa} \mathbf{w}^{*T} \mathbf{x}^{(\ell)} + \tilde{\kappa} b^*) \geq 1 \text{ for all } \ell = 1, 2, \dots, n$$

and there exists at least one '=' holding



- Therefore, the maxima of  $\frac{1}{\|\mathbf{w}\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b)]$  can be found by solving the **constraint-free** optimization problem

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

or by solving the **constrained** optimization problem

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

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

there exists at least one '=' holding

- The optimal solution  $\mathbf{w}^*$  and  $\tilde{\mathbf{w}}^*$  may not be identical, but their induced values  $\frac{1}{\|\mathbf{w}^*\|} \min_{\ell} [y^{(\ell)} \cdot (\mathbf{w}^{*T} \mathbf{x}^{(\ell)} + b^*)]$  and  $\frac{1}{\|\tilde{\mathbf{w}}^*\|} \min_{\ell} [y^{(\ell)} \cdot (\tilde{\mathbf{w}}^{*T} \mathbf{x}^{(\ell)} + \tilde{b}^*)]$  must be equal

- In the second optimization problem, since  $y^{(\ell)} \cdot (\tilde{\mathbf{w}}^T \mathbf{x}^{(\ell)} + \tilde{b}) \geq 1$  for all  $\ell = 1, 2, \dots, n$  and **there exists at least one '=' holding**, we can easily have

$$\min_{\ell} [y^{(\ell)} \cdot (\tilde{\mathbf{w}}^T \mathbf{x}^{(\ell)} + \tilde{b})] = 1$$

Thus, the optimization objective can be reduced to  $\frac{1}{\|\tilde{\mathbf{w}}\|}$

- Maximizing  $\frac{1}{\|\tilde{\mathbf{w}}\|}$  can be replaced by minimizing  $\|\tilde{\mathbf{w}}\|^2$ . Hence, the problem can be equivalently written as

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

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

**there exists at least one '=' holding**

- When minimizing  $\|\tilde{\mathbf{w}}\|^2$ , the constraint '**at least one '=' holds**' will be satisfied automatically (**why??**). Thus, it can be dropped without influencing the result



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

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y^{(\ell)} \cdot (\mathbf{w}^T \mathbf{x}^{(\ell)} + b) \geq 1, \quad \text{for } \ell = 1, 2, \dots, N \end{aligned}$$

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

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

# The Equivalent Dual Formulation

---

- Every convex optimization problem corresponds to an equivalent dual formulation

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

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

- Deriving the close-form expression of function  $g(\mathbf{a})$

➤ Setting the gradient  $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$  and  $\frac{\partial \mathcal{L}}{\partial b} = 0$  gives

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

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

➤ 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

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

where  $[\mathbf{M}]_{\ell j} \triangleq y^{(\ell)} y^{(j)} \mathbf{x}^{(\ell)T} \mathbf{x}^{(j)}$

- Then, the dual optimization becomes

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

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

It is also a quadratic optimization, which can be efficiently solved by *numerical methods*

- Relation between optima  $\mathbf{w}^*$ ,  $b^*$  and optima  $\mathbf{a}^*$ 
  - Given the optima  $\mathbf{a}^*$ , according to  $\mathbf{w} = \sum_{\ell=1}^N a_{\ell} y^{(\ell)} \mathbf{x}^{(\ell)}$ , the optimal  $\mathbf{w}^*$  can be equivalently represented as

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

- 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

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

where  $\mathcal{S}$  denotes the set of samples on the margin

- Maximum-margin classifiers

- Primal version

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

- Dual version

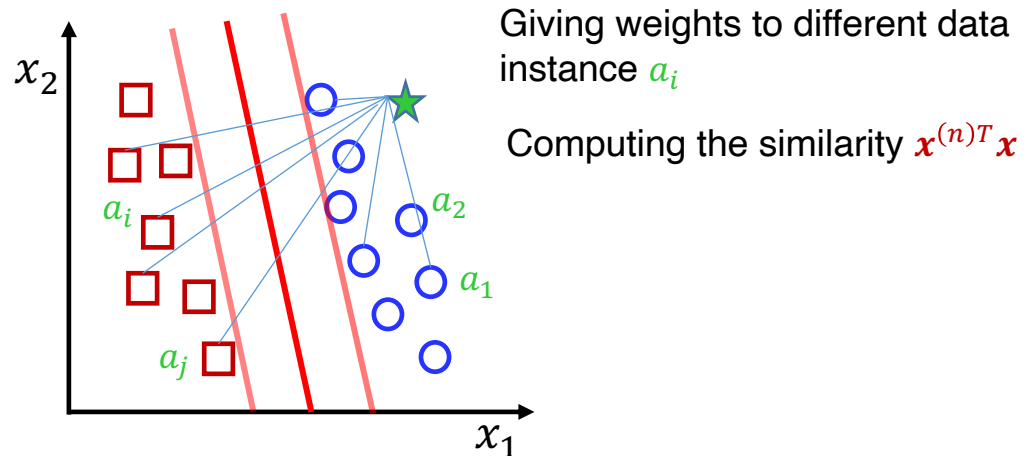
Substituting  $\mathbf{w}^* = \sum_{n=1}^N a_n^* y^{(n)} \mathbf{x}^{(n)}$  into the primal version gives

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

The two classifiers are *equivalent*

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

- 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 Problems

- Optimization complexity

## Primal

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

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

# of parameters to optimize:  
dimension of features

## Dual

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

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

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

# of parameters 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 operation  $\mathbf{w}^{*T} \mathbf{x}$

### Dual

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

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

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

- Fortunately, it can be proved 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

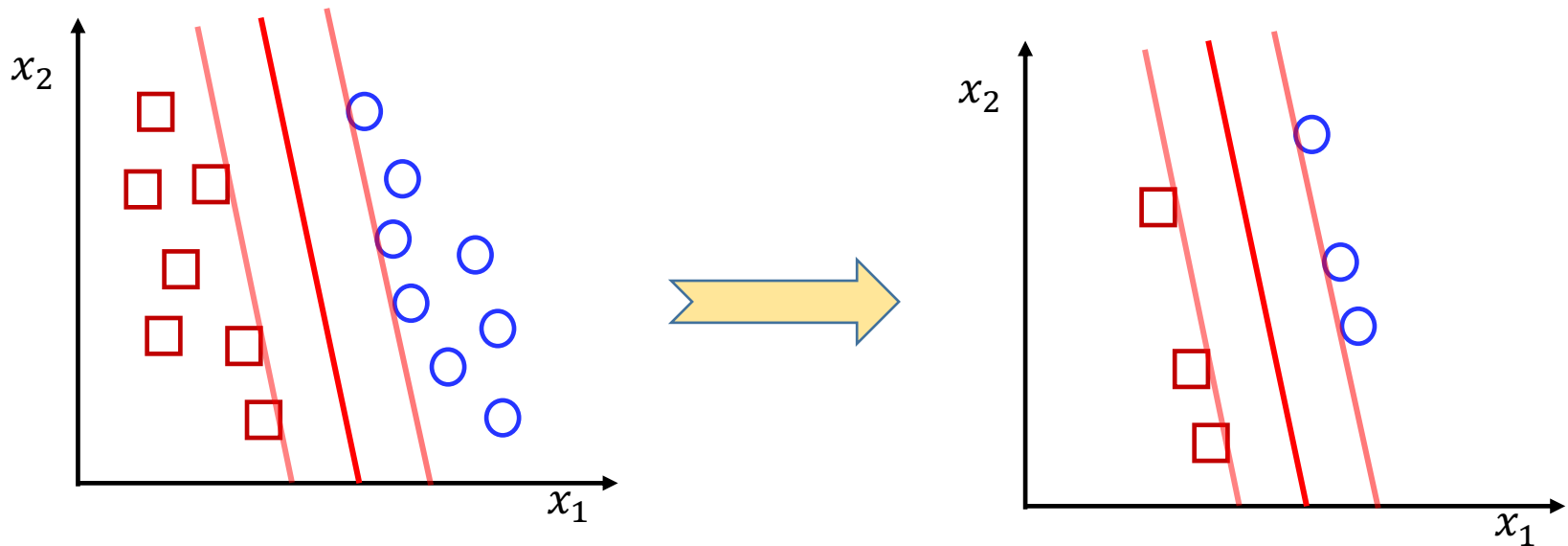
- From the last condition, we can see that  $a_n^* \neq 0$  only when  $y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) = 1$
- If  $\mathbf{x}^{(n)}$  satisfies  $y^{(n)}(\mathbf{w}^{*T} \mathbf{x}^{(n)} + b^*) = 1$ , it means that it lies on the margin

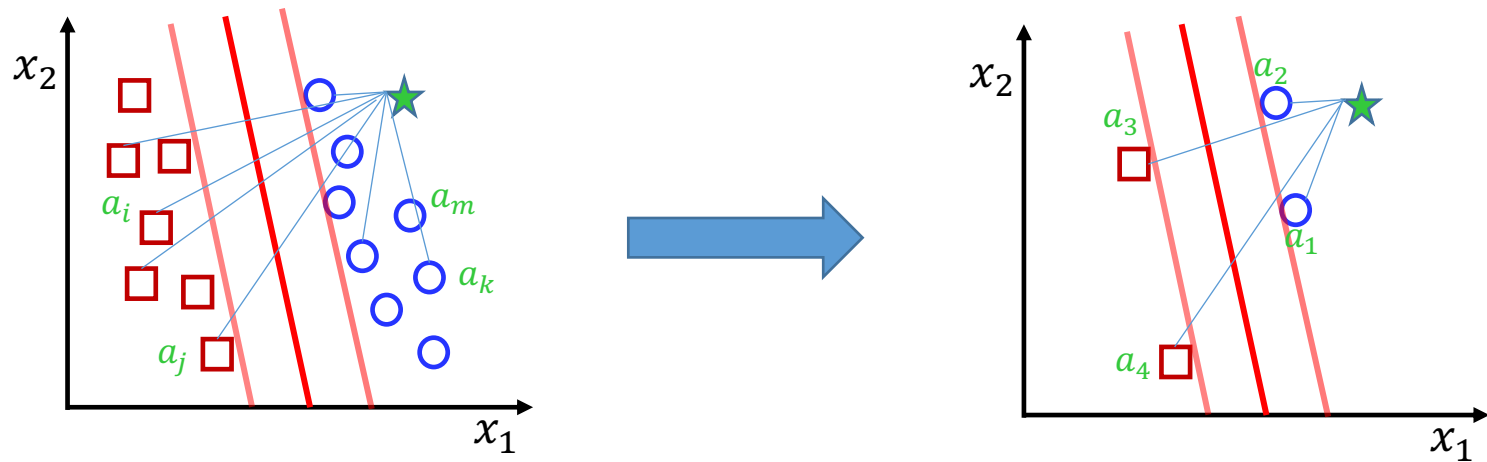
This kind of samples are called *support vectors*

- 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)





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

# Outline

---

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

# Non-separable Classes

- The implicit assumption in the previous maximum-margin classifier

The training samples are linearly separable!!!



- What will happen to the optimization problem under such circumstances?

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

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

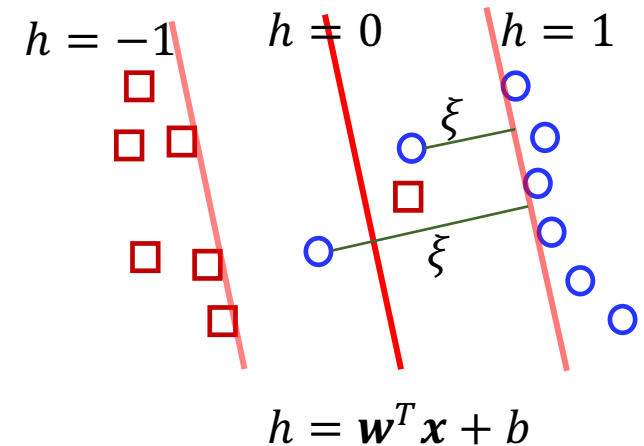
- There is no feasible solution to the optimization problem. That is, *no such hyperplane exists*

# 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* with  $\xi_n \geq 0$



- 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

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

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

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

When  $a_n > C$ , it can be shown that  $g(\mathbf{a}) = -\infty$

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

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

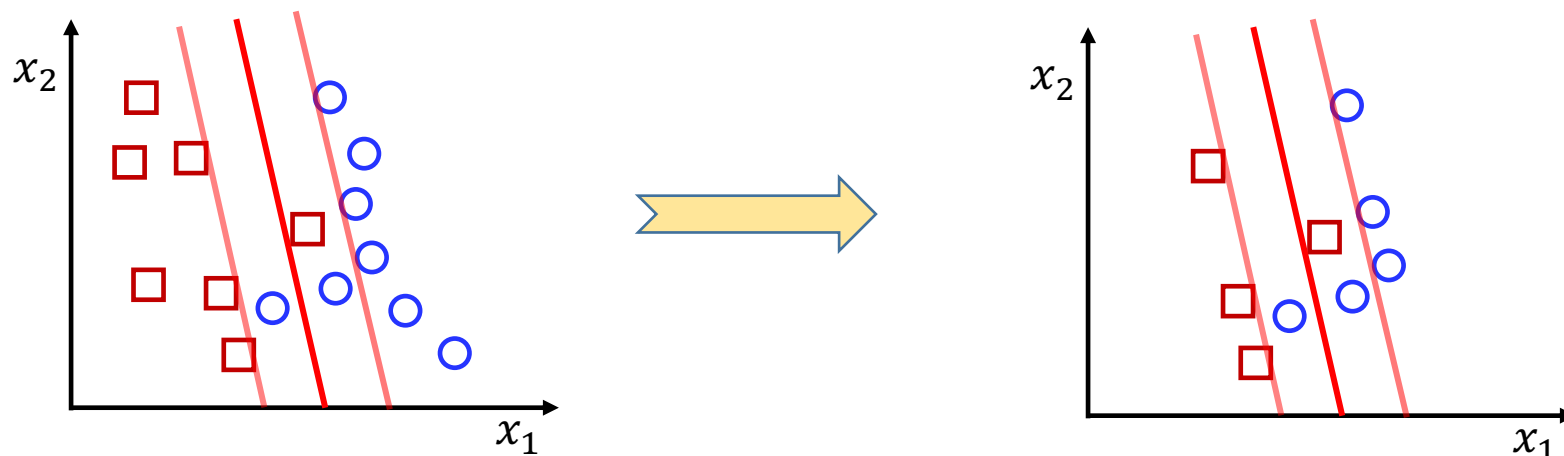
$$\hat{y}(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \mathbf{x} + b^*)$$

- With the optima  $\mathbf{a}^*$ , a sample  $\mathbf{x}$  is classified as

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

Also, the two classifiers above are *equivalent*

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



# Outline

---

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

# Non-linearization

---

- The maximum-margin classifiers so far are still **linear**
- To non-linearize the model, we can transform the original data  $x$  to the feature space via the **basis function**

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

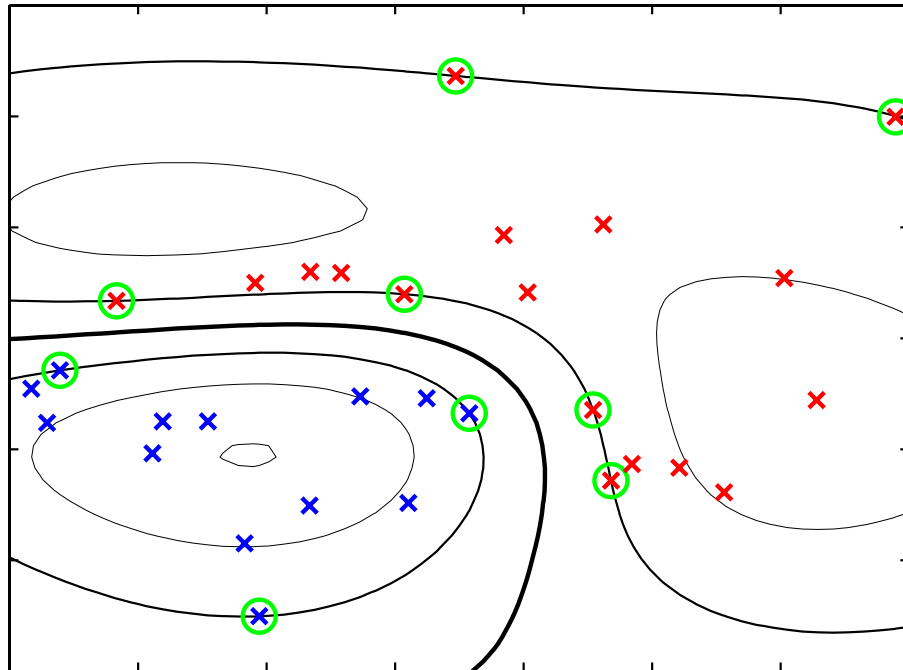
- The primal maximum-margin optimization problem becomes

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

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

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

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 yield better performance, we should set the dimension of transformed feature space  $\phi(\mathbf{x}^{(n)})$  *to be as high as possible*
- However, the dimension of basis function  $\phi(\mathbf{x})$  *cannot be set too high* since the primal problem would become very expensive

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

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

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

- The problem can be solved via its dual form

$$\begin{aligned} \max_{\mathbf{a}} \quad & g(\mathbf{a}) \\ \text{s.t.} \quad & a_n \geq 0, \quad a_n \leq C \\ & \sum_{n=1}^N a_n y^{(n)} = 0 \end{aligned}$$

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

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

- The dimension of  $\mathbf{a}$  is *independent of the dimension of  $\boldsymbol{\phi}(\cdot)$* , thus the dual form is able to work in a 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  $k(\mathbf{x}, \mathbf{x}')$  that can be expressed as an inner product of some function  $\boldsymbol{\phi}(\cdot)$

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$$

Obviously,  $\mathbf{x}^T \mathbf{x}'$  and  $\boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$  are kernel functions

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

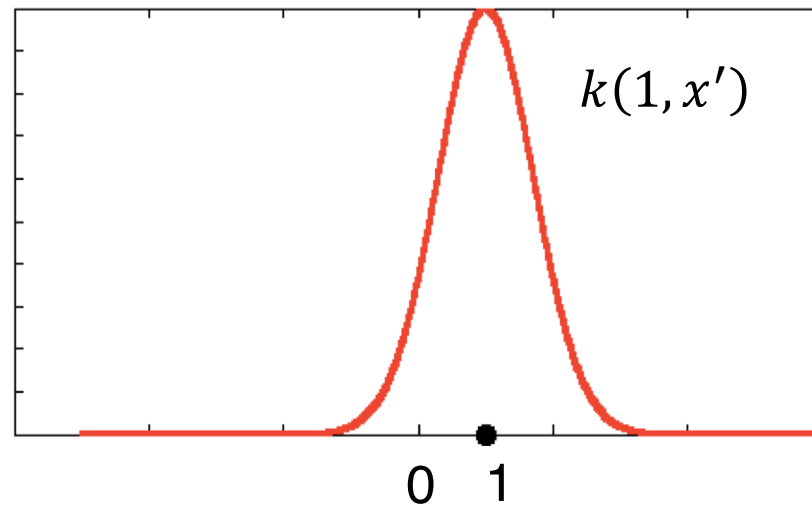
$$\int \int g(\mathbf{x}) k(\mathbf{x}, \mathbf{y}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0, \quad \forall g(\cdot) \in L^2,$$

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

If a function  $k(\mathbf{x}, \mathbf{x}')$  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

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



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

$$\phi(x) = e^{-x^2/2\sigma^2} \left[ 1, \sqrt{\frac{1}{1!\sigma^2}} x, \sqrt{\frac{1}{2!\sigma^4}} x^2, \sqrt{\frac{1}{3!\sigma^6}} x^3, \dots \right]^T$$

# Kernel Trick

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

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

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

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

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

- The induced classifier

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

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

# Outline

---

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

- In the logistic regression, we minimize the loss

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

Note:

$\tilde{y} \in \{0, 1\}, y \in \{-1, 1\}$

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  $E_{Ideal}(z) = 0$  if  $z \geq 0$ ; 1 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*

- 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

