# Support Vector Machines

DCS310

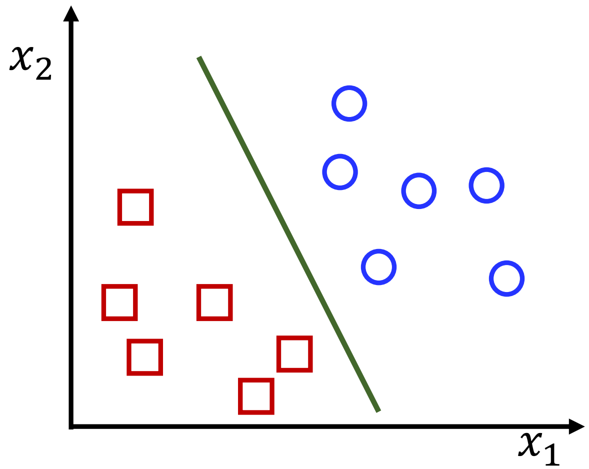
Sun Yat-sen University

# 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

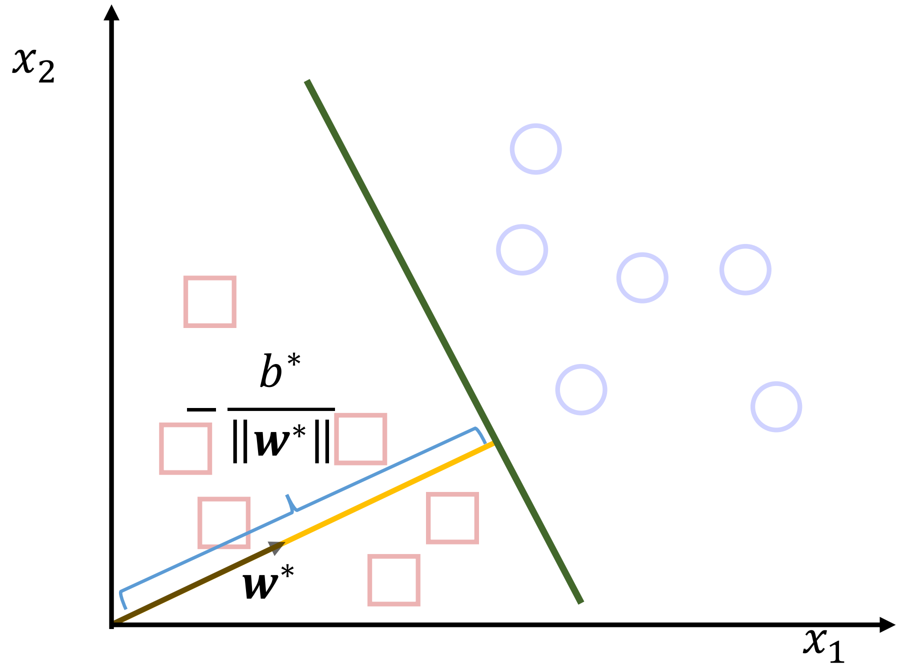



- Logistic regression
  - The decision-boundary hyperplane is found by minimizing the cross-entropy loss

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

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

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



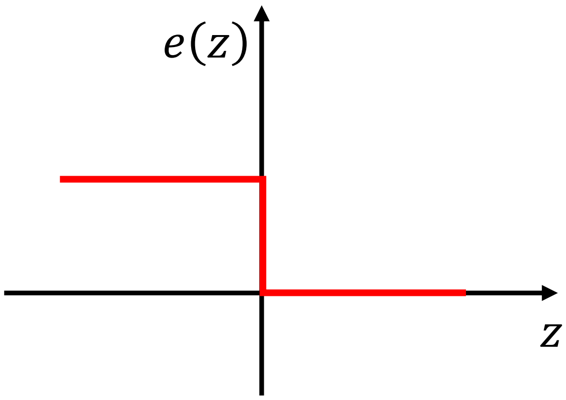

1) The hyperplane is *perpendicular* to the vector $\boldsymbol{w}^*$
2) The distance from the original point to the hyperplane is $-\frac{b^*}{\|\boldsymbol{w}^*\|}$

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

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

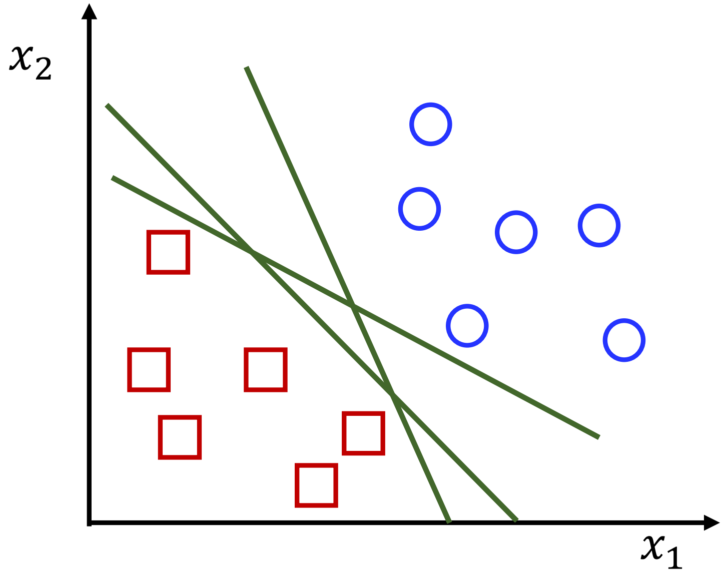
$L(\boldsymbol{w}, b)$ represents the number of misclassified samples

- $y \in \{-1, 1\}$
- $e(z)$ is a jump function, i.e., $e(z) = 0$ *if* $z \geq 0$; $e[z] = 1$ *otherwise*



- If the samples are linearly separable, there will be numerous ideal classifiers, which are determined by $\boldsymbol{w}^*$ and $b^*$

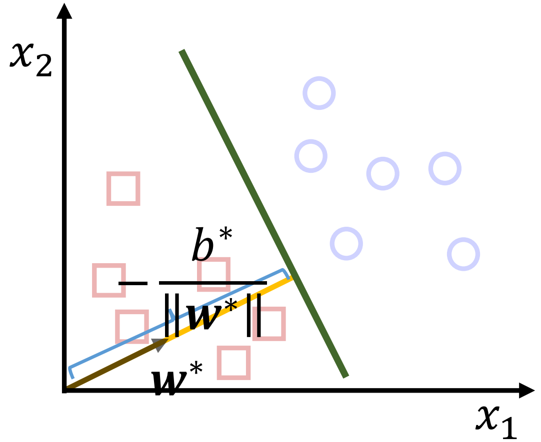
- Every $\boldsymbol{w}^*$ and $b^*$ corresponds to a hyperplane



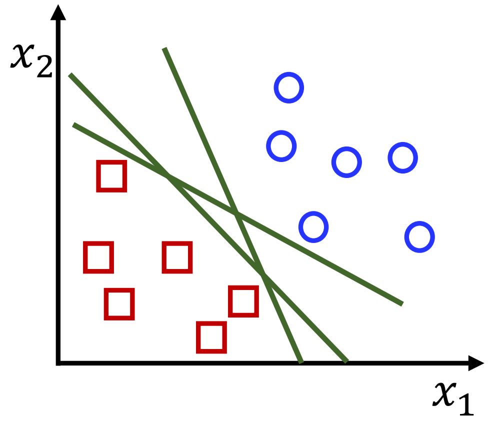

*All the hyperplanes can have the loss reduced to zero*

# Which Hyperplane is the Best?

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




- All the hyperplanes in the ideal classifier are optimal from the perspective of *minimizing the number of misclassified samples*

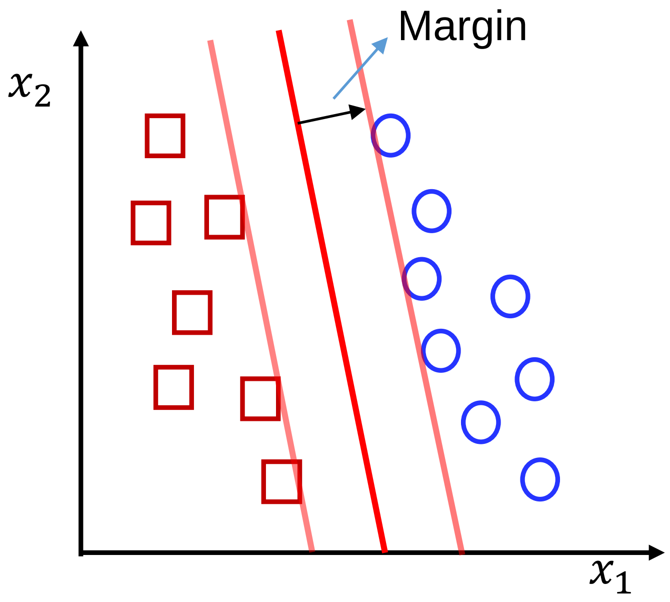


# Outline

- 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

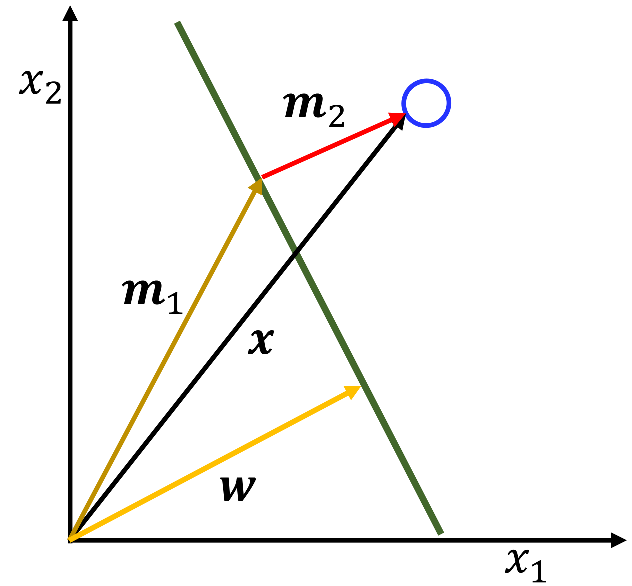



Thanks to the large margin, *it can be expected that an unseen sample is more likely to be classified correctly under such a decision boundary*

# How to Represent the Margin?

- The distance from the sample $\boldsymbol{x}$ to the hyperplane $\mathcal{H}$
  - Every sample $\boldsymbol{x}$ can be decomposed as

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




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

  - Thus, we have

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

$$= \boldsymbol{w}^T\boldsymbol{m}_2$$

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

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

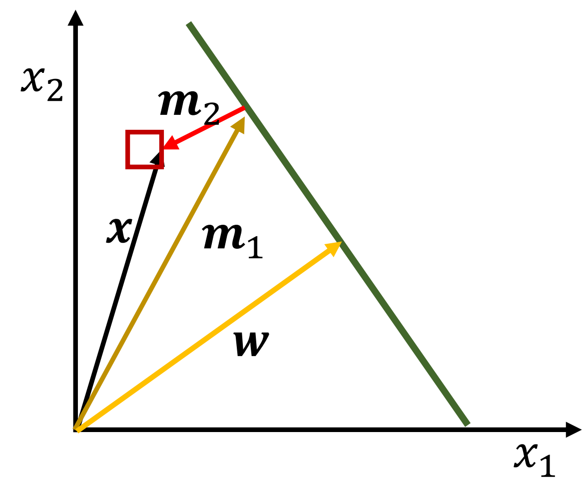
with $|\gamma|$ *representing the length of* $\boldsymbol{m}_2$

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

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

- The distance of a sample on the other side of the hyperplane (i.e. of negative class, $h(\boldsymbol{x}) < \boldsymbol{0}$) is

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

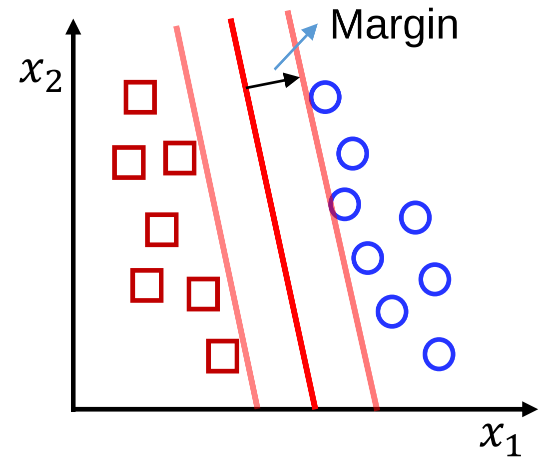



- The distance of a sample $(\boldsymbol{x}, y)$ to the hyperplane is given by

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

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

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




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

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

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

*But how to optimize is unknown*

# The Transformed Objective Function

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

$$y^{(\ell)}\cdot\left(\kappa\boldsymbol{w}^{*T}\boldsymbol{x}^{(\ell)}+\kappa b^*\right)\geq 1 \ \text{ for all } \ell=1,2,\cdots,n$$

Among all $\boldsymbol{w},b$ that satisfy $y^{(\ell)}\cdot\left(\boldsymbol{w}^T\boldsymbol{x}^{(\ell)}+b\right)\geq 1$ for all $\ell$, *the $\boldsymbol{w}$ and $b$ with the smallest $\|\boldsymbol{w}\|^2$ must be the optima $\boldsymbol{w}^*$* and *$b^*$ that maximizes*

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

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

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$s.t.: y^{(\ell)} \cdot \left(\boldsymbol{w}^T \boldsymbol{x}^{(\ell)} + b\right) \geq 1, \quad \text{for } \ell = 1, 2, \cdots, N$$

- This is a convex quadratic optimization problem. Its optimal solution can be found by *numerical methods* efficiently

- With the optimal $\boldsymbol{w}^*$ and $b^*$, an unseen data $\boldsymbol{x}$ can be classified as

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

# 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

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

where the Lagrange multiplier $a_\ell$ is required to satisfy $a_\ell \geq 0$

- The *Lagrange dual function*

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

- The dual formulation of the original optimization problem

$$\boxed{\begin{aligned} &\max_{\boldsymbol{a}} \; g(\boldsymbol{a}) \\ &\quad s.t.: \boldsymbol{a} \geq \boldsymbol{0} \end{aligned}}$$

- Deriving the close-form expression (a.k.a. analytical solution) of function $g(\boldsymbol{a})$, i.e. removing minimization w.r.t. $\boldsymbol{w}$ and $b$
  - Setting the gradient $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \mathbf{0}$ and $\frac{\partial \mathcal{L}}{\partial b} = 0$, which gives (classroom work)

$$\frac{\partial\left(\frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{\ell=1}^{N} a_\ell\left(y^{(\ell)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(\ell)} + b\right) - 1\right)\right)}{\partial \boldsymbol{w}} = \mathbf{0} \Rightarrow \boldsymbol{w} = \sum_{\ell=1}^{N} a_\ell y^{(\ell)} \boldsymbol{x}^{(\ell)}$$

$$\frac{\partial\left(\frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{\ell=1}^{N} a_\ell\left(y^{(\ell)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(\ell)} + b\right) - 1\right)\right)}{\partial b} = \mathbf{0} \Rightarrow \sum_{\ell=1}^{N} a_\ell y^{(\ell)} = 0$$

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

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

- Then, the dual optimization becomes

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

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

Attention: $g(\boldsymbol{a})$ no longer contains minimization w.r.t. $\boldsymbol{w}$ and $b$

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

- Relation between optima $\boldsymbol{w}^*$, $b^*$ and optima $\boldsymbol{a}^*$

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

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

  - Due to $y^{(i)}\left(\boldsymbol{w}^{*T}\boldsymbol{x}^{(i)} + b^*\right) = 1$ for all samples $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$ that are on the margin (i.e. $i \in \mathcal{S}$), we can derive that

For each $i \in \mathcal{S}$

$$b^* = \begin{cases} 1 - \boldsymbol{w}^{*T}\boldsymbol{x}^{(i)} & \text{if } y^{(i)} = 1 \\ -1 - \boldsymbol{w}^{*T}\boldsymbol{x}^{(i)} & \text{if } y^{(i)} = -1 \end{cases}$$

$$= y^{(i)} - \sum_{\ell=1}^{N} a_\ell^* y^{(\ell)} \boldsymbol{x}^{(\ell)T} \boldsymbol{x}^{(i)}$$

Mean solution is more robust

$$b^* = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left( y^{(i)} - \sum_{\ell=1}^{N} a_\ell^* y^{(\ell)} \boldsymbol{x}^{(\ell)T} \boldsymbol{x}^{(i)} \right)$$

- Maximum-margin classifiers
  - Primal version

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

  - Dual version

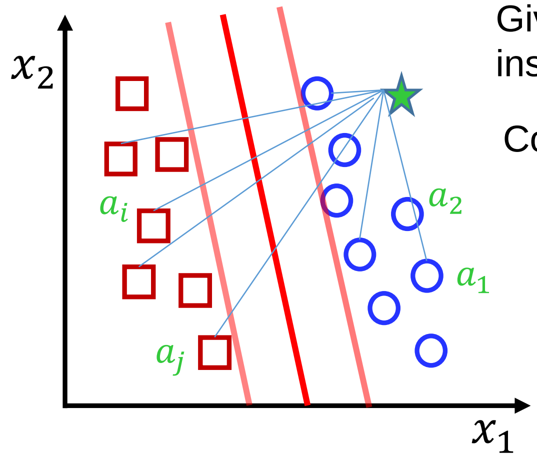
    Substituting $\boldsymbol{w}^* = \sum_{n=1}^{N} a_n^* y^{(n)} \boldsymbol{x}^{(n)}$ into the primal version gives

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

The two classifiers are *equivalent*

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

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




Giving weights to different data instance $a_i$

Computing the similarity $\boldsymbol{x}^{(n)T}\boldsymbol{x}$

# Comparisons on the Primal and Dual Results

- Optimization (training) complexity

**Primal**

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

$$s.t.: y^{(\ell)} \cdot \left(\boldsymbol{w}^T \boldsymbol{x}^{(\ell)} + b\right) \geq 1,$$

$$\text{for } \ell = 1, 2, \cdots, N$$

# of parameter to optimize:
dimension of features

**Dual**

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

$$s.t.: \quad \boldsymbol{a} \geq \boldsymbol{0}$$

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

# of parameter to optimize:
# of training samples

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

- Testing complexity

| Primal | Dual |
|---|---|
| $\hat{y}(\boldsymbol{x}) = sign(\boldsymbol{w}^{*T}\boldsymbol{x} + b^*)$ | $\hat{y}(\boldsymbol{x}) = sign\left(\sum_{n=1}^{N} a_n^* y^{(n)} \boldsymbol{x}^{(n)T}\boldsymbol{x} + b^*\right)$ |
| Just need one inner-product $\boldsymbol{w}^{*T}\boldsymbol{x}$ | Need $N$ inner-products $\boldsymbol{x}^{(n)T}\boldsymbol{x}$ for $n = 1, 2, \cdots, N$ |

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

$$a_n^* \geq 0$$

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

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

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 $y^{(n)}\left(\boldsymbol{w}^{*T}\boldsymbol{x}^{(n)} + b^*\right) = 1$

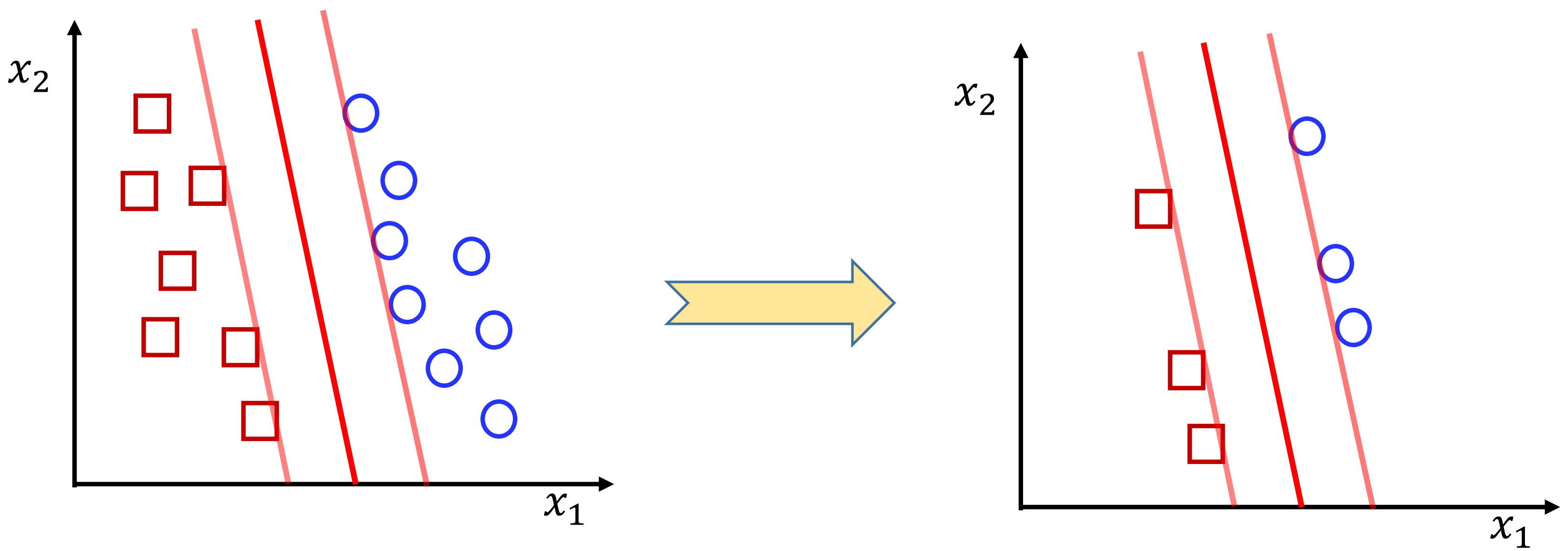
- If $\boldsymbol{x}^{(n)}$ satisfies $y^{(n)}\left(\boldsymbol{w}^{*T}\boldsymbol{x}^{(n)} + b^*\right) = 1$, it means that it lies on the margin

This kind of samples are called *support vectors*

- Thus, when we classify an unseen sample $\boldsymbol{x}$ as

$$\hat{y}(\boldsymbol{x}) = sign\left(\sum_{n \in \mathcal{S}} a_n^* y^{(n)} \boldsymbol{x}^{(n)T} \boldsymbol{x} + b^*\right),$$

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



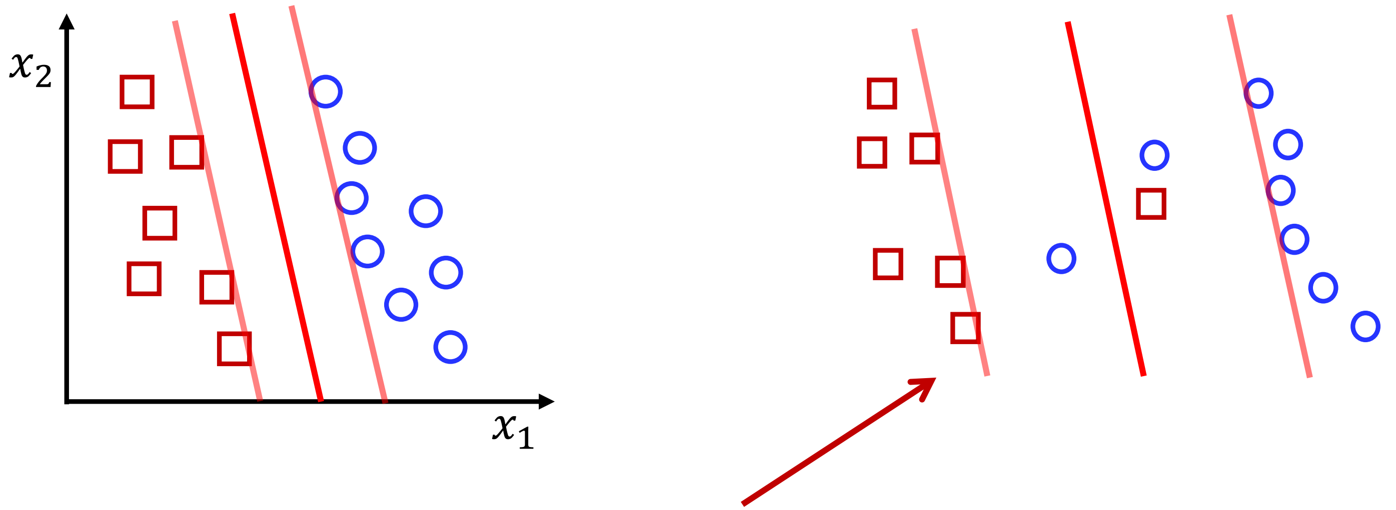
# Outline

- 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

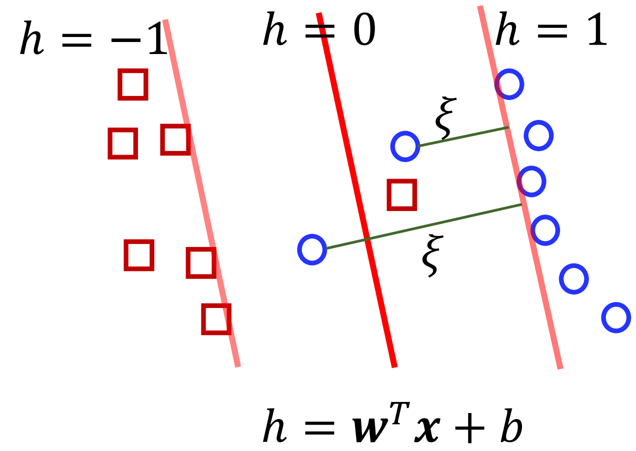
$$\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2$$

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

# Soft Maximum Margin

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

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




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

松弛变量

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

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

where $C$ is used to control the relative importance

- The optimization problem now becomes

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

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

$$\xi_n \geq 0, \qquad \text{for } n = 1,2,\cdots,N$$

- Using the same method as before (refer to slice 17), the *dual formulation* can be derived as (homework)

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

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

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

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

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

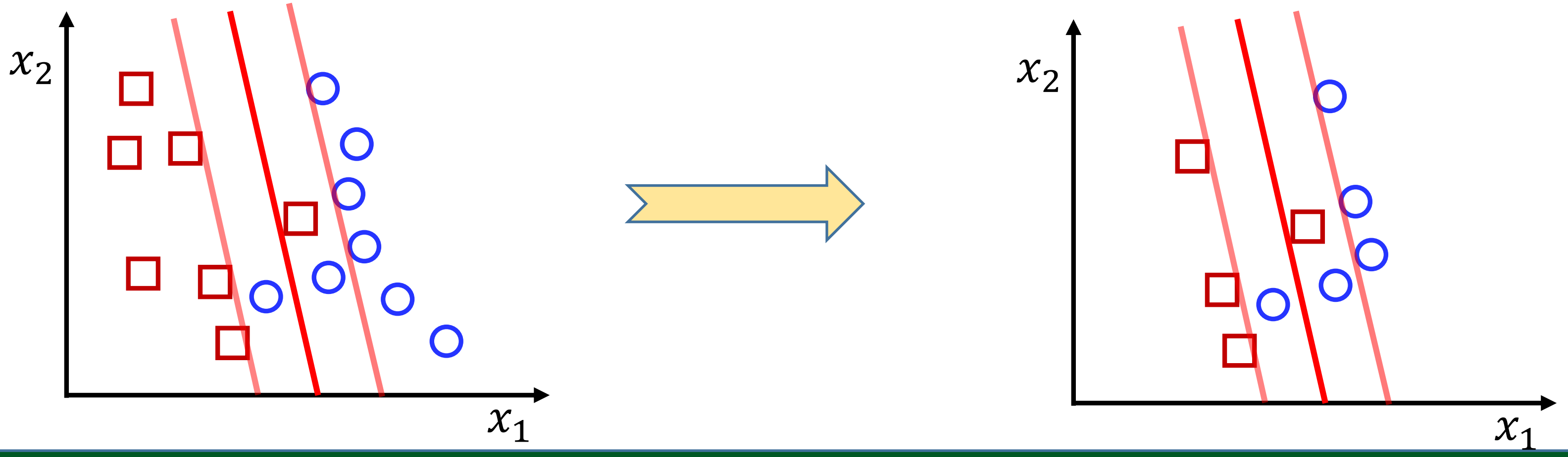
$$\hat{y}(\boldsymbol{x}) = sign(\boldsymbol{w}^{*T}\boldsymbol{x} + b^*)$$

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

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

Also, the two classifiers above are *equivalent*

- The optima $\boldsymbol{a}^*$ 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 non-linearizing the model, we can transform the original feature $\boldsymbol{x}$ to another space via the basis function

$$\boldsymbol{\phi}\colon \boldsymbol{x} \longrightarrow \boldsymbol{\phi}(\boldsymbol{x})$$

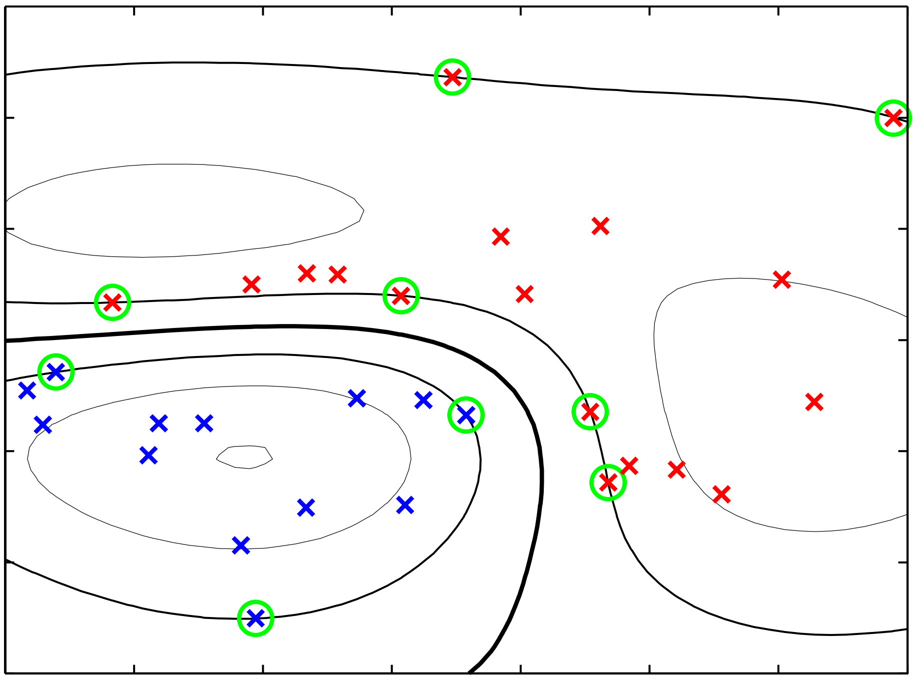
- The primal maximum-margin optimization problem becomes

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

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

$$\xi_n \geq 0, \qquad \text{for } n = 1, 2, \cdots, N$$

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



- Intuitively, data is easier to be separated in high-dimensional space

- *To achieve better performance, we prefer the dimension of the transformed feature space* $\boldsymbol{\phi}\left(\boldsymbol{x}^{(n)}\right)$ *to be as high as possible*

- However, the dimension of basis function $\boldsymbol{\phi}(\boldsymbol{x})$ *cannot be too high*, otherwise the optimization would be very expensive

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

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

$$\xi_n \geq 0, \qquad \text{for } n = 1, 2, \cdots, N$$

- The problem can be solved via its dual form

$$\max_{\boldsymbol{a}} \; g(\boldsymbol{a})$$

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

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

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

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

- The dimension of $\boldsymbol{a}$ 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

$$\boldsymbol{\phi}\left(\boldsymbol{x}^{(n)}\right)^{T}\boldsymbol{\phi}(\boldsymbol{x}),$$

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 $k(\boldsymbol{x}, \boldsymbol{x}')$ that can be expressed as an inner production of some function $\boldsymbol{\phi}(\cdot)$

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

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

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

对称正定

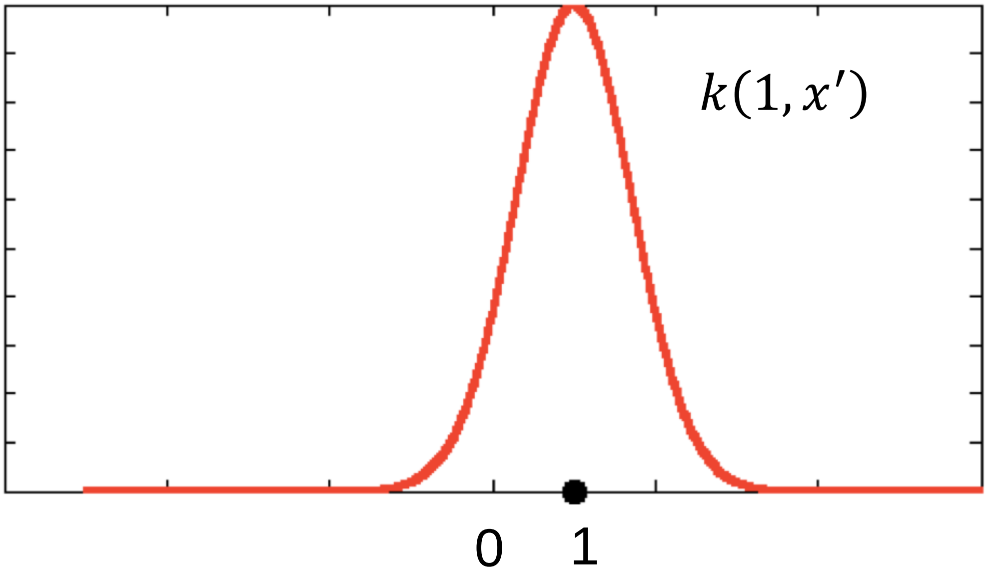
$$\int\int g(\boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{y})g(\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y} \geq 0, \qquad \forall\, g(\cdot) \in L^2,$$

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

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

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




➢ The function $\boldsymbol{\phi}(\cdot)$ of Gaussian kernel has infinite dimensions

# Kernel Trick

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

$$\max_{\boldsymbol{a}} \; g(\boldsymbol{a})$$

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

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

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

- The induced classifier

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

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

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

$$L(\boldsymbol{w}, b) = -\sum_{n=1}^{N} \left[y^{(n)} \log \sigma\left(h^{(n)}\right) + \left(1 - y^{(n)}\right) \log\left(1 - \sigma\left(h^{(n)}\right)\right)\right] + \lambda \|\boldsymbol{w}\|^2$$

$$= \sum_{n=1}^{N} \log\left(1 + \exp\left(-y^{(n)} h^{(n)}\right)\right) + \lambda \|\boldsymbol{w}\|^2$$

$$= \sum_{n=1}^{N} E_{LR}\left(y^{(n)} h^{(n)}\right) + \lambda \|\boldsymbol{w}\|^2$$

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

- In the ideal classifier, we minimize the loss

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

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

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

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

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

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

$$L(\boldsymbol{w}, b) = C \sum_{n=1}^{N} E_{SV}\left(y^{(n)} h^{(n)}\right) + \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$= \sum_{n=1}^{N} E_{SV}\left(y^{(n)} h^{(n)}\right) + \lambda \|\boldsymbol{w}\|^2$$

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 three error functions