

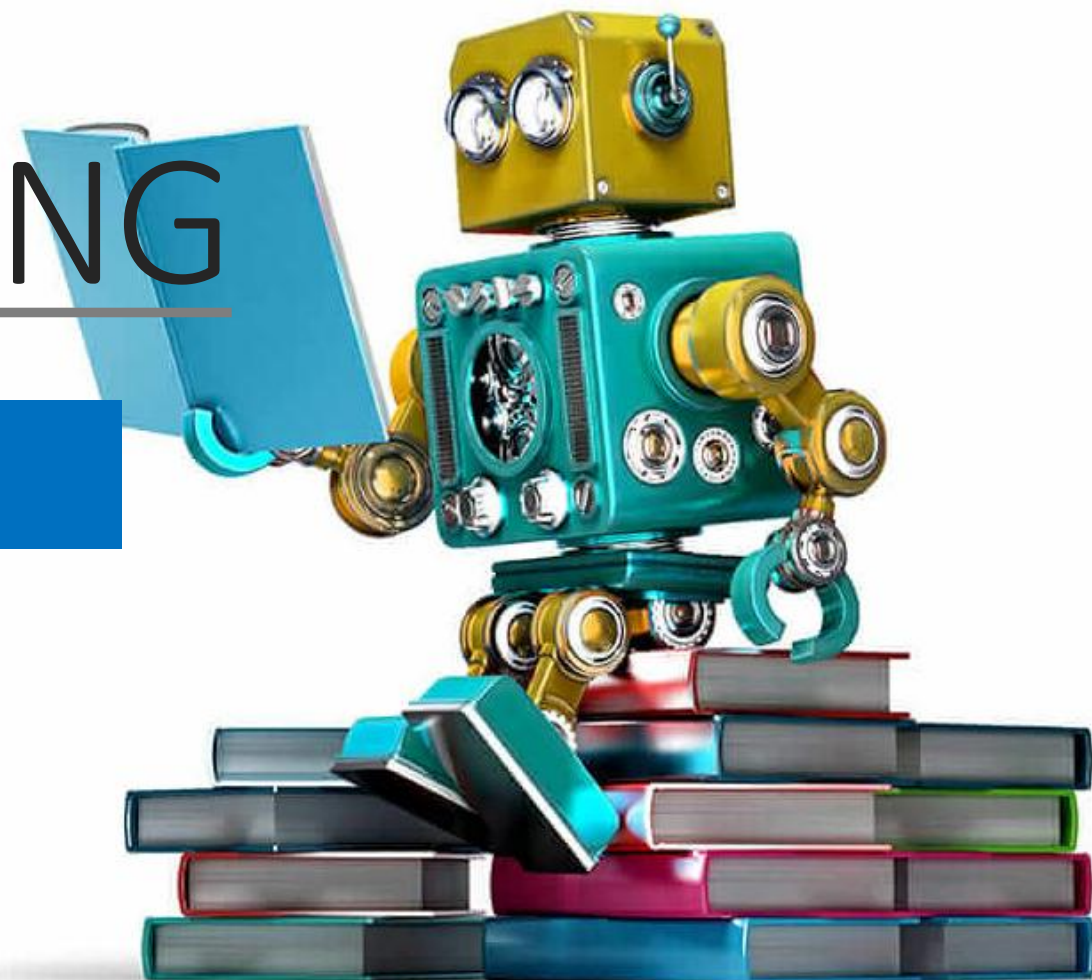
# MACHINE LEARNING

---

## LAB10 SVM

贾艳红 Jana

Email: [jiayh@mail.sustech.edu.cn](mailto:jiayh@mail.sustech.edu.cn)





- Intro. to Linear separability and Perceptron
- Intro. to Support Vector Machine (svm) classifier
- LAB Assignment



# Binary Classification

Given training data  $(\mathbf{x}_i, y_i)$  for  $i = 1 \dots N$ , with  $\mathbf{x}_i \in \mathbb{R}^d$  and  $y_i \in \{-1, 1\}$ , learn a classifier  $f(\mathbf{x})$  such that

$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

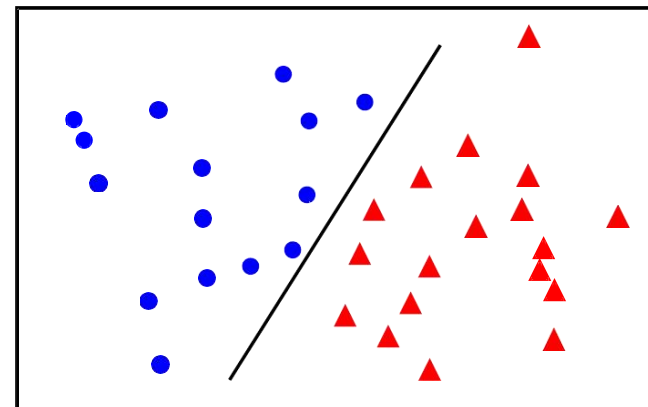
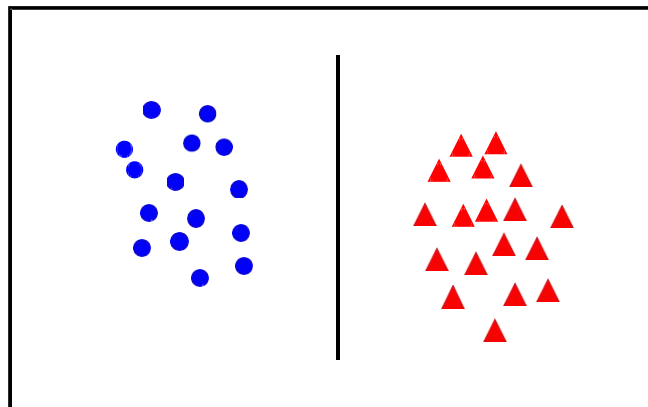
i.e.  $y_i f(\mathbf{x}_i) > 0$  for a correct classification.



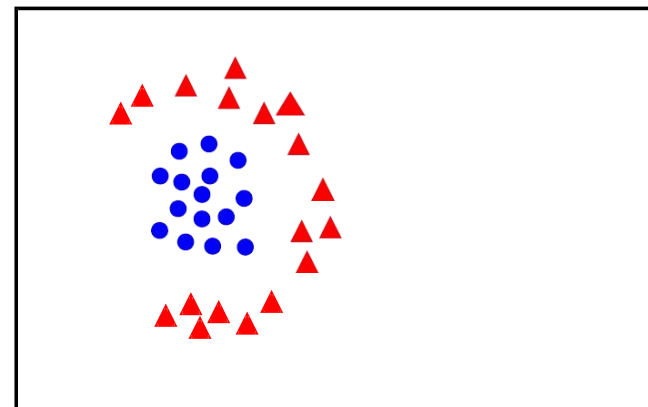
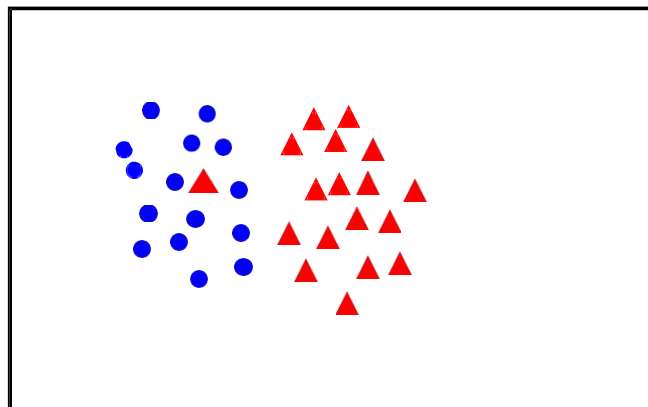
# Linear separability



linearly  
separable



not  
linearly  
separable

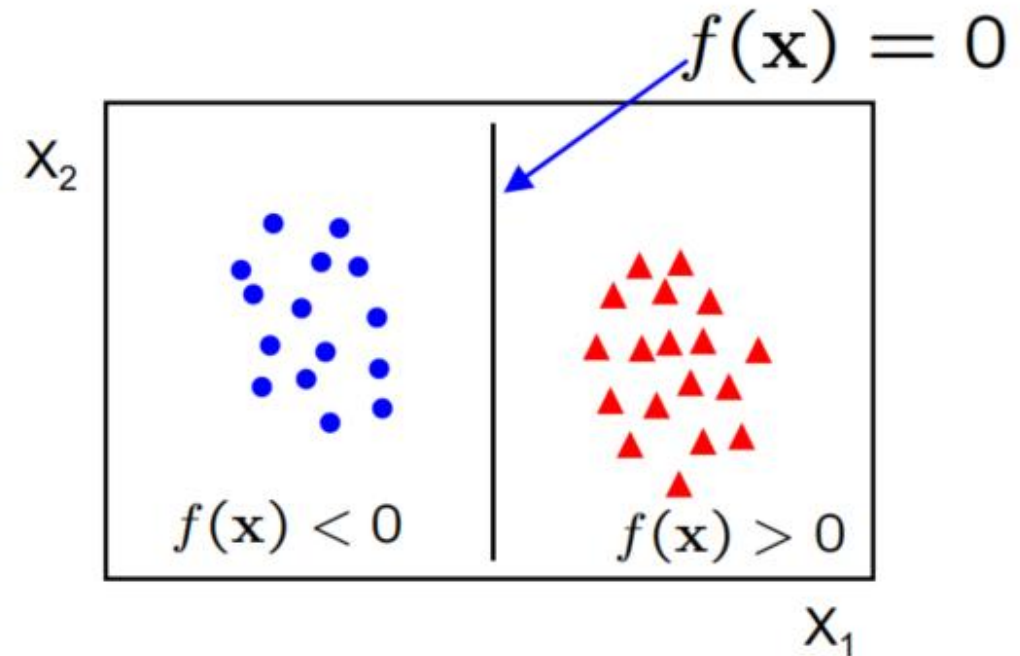




# Linear classifiers

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



- in 2D the discriminant is a line
- $\mathbf{W}$  is the **normal** to the line, and  $b$  the **bias**
- $\mathbf{W}$  is known as the **weight vector**

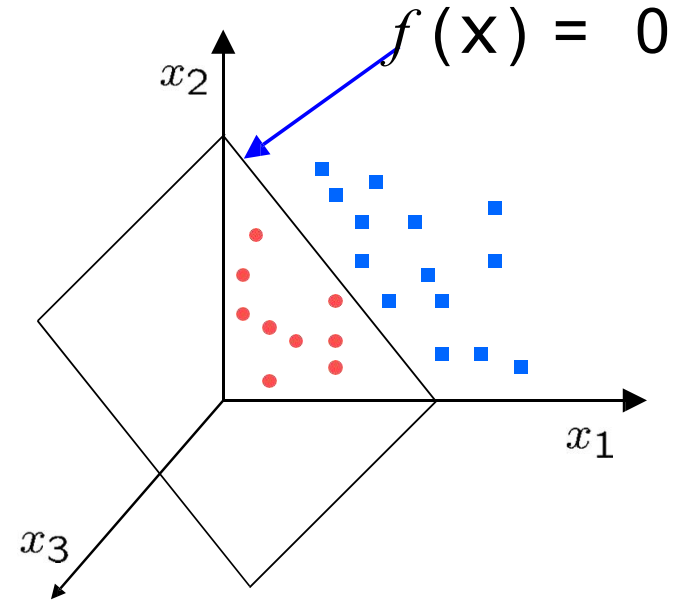


# Linear classifiers



A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



- in 3D the discriminant is a plane, and in nD it is a hyperplane
- For a linear classifier, the training data is used to learn  $\mathbf{w}$  and then discarded
- Only  $\mathbf{w}$  is needed for classifying new data





# The Perceptron Classifier

Given linearly separable data  $\mathbf{x}_i$  labelled into two categories  $y_i = \{-1, 1\}$ , find a weight vector  $\mathbf{w}$  such that the discriminant function

$$f(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + b$$

separates the categories for  $i = 1, \dots, N$

- how can we find this separating hyperplane ? [The Perceptron Algorithm](#)

Write classifier as  $f(\mathbf{x}_i) = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_i + w_0 = \mathbf{w}^\top \mathbf{x}_i$

where  $\mathbf{w} = (\tilde{\mathbf{w}}, w_0)$ ,  $\mathbf{x}_i = (\tilde{\mathbf{x}}_i, 1)$

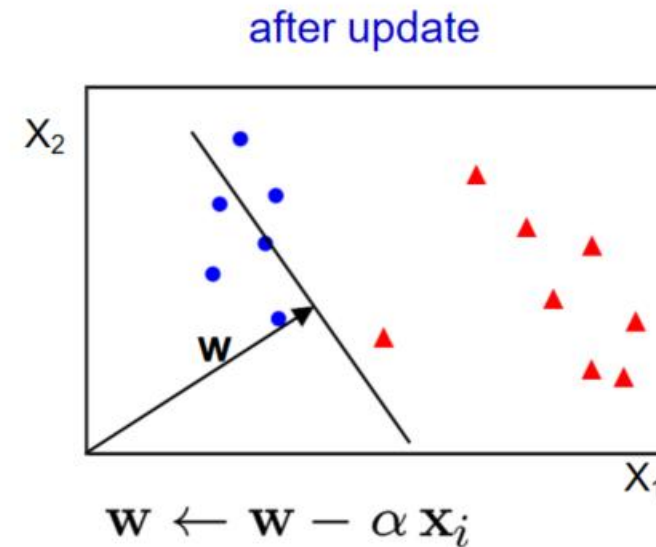
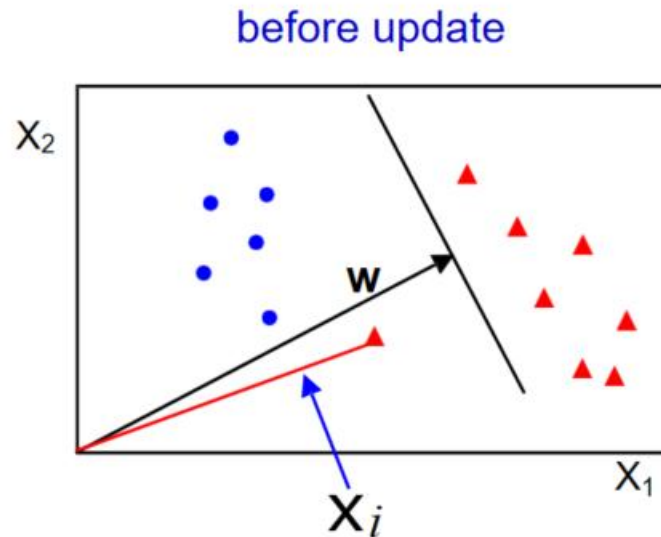
- Initialize  $\mathbf{w} = 0$
- Cycle through the data points  $\{\mathbf{x}_i, y_i\}$ 
  - if  $\mathbf{x}_i$  is misclassified then  $\mathbf{w} \leftarrow \mathbf{w} + \alpha \text{sign}(f(\mathbf{x}_i)) \mathbf{x}_i$
- Until all the data is correctly classified



# For example in 2D



- Initialize  $\mathbf{w} = 0$
- Cycle through the data points  $\{\mathbf{x}_i, y_i\}$ 
  - if  $\mathbf{x}_i$  is misclassified then  $\mathbf{w} \leftarrow \mathbf{w} + \alpha \text{sign}(f(\mathbf{x}_i)) \mathbf{x}_i$
- Until all the data is correctly classified



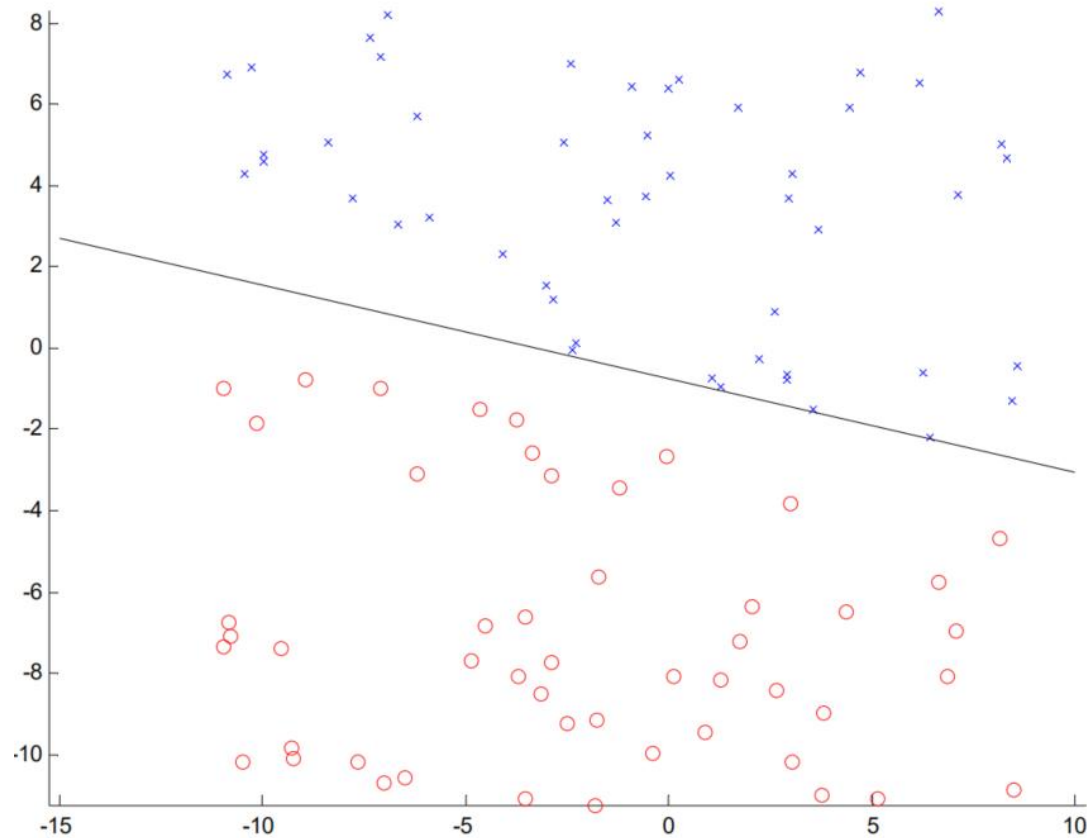
NB after convergence  $\mathbf{w} = \sum_i^N \alpha_i \mathbf{x}_i$





# For example in 2D

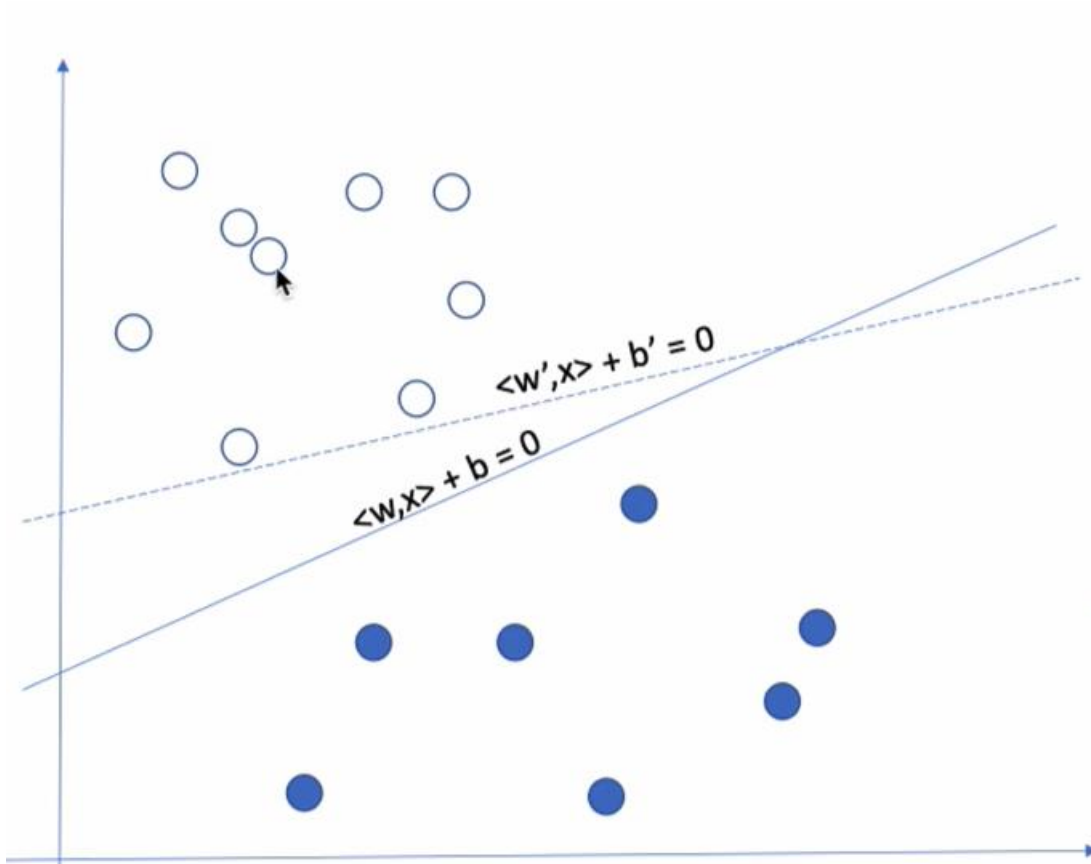
## Perceptron example



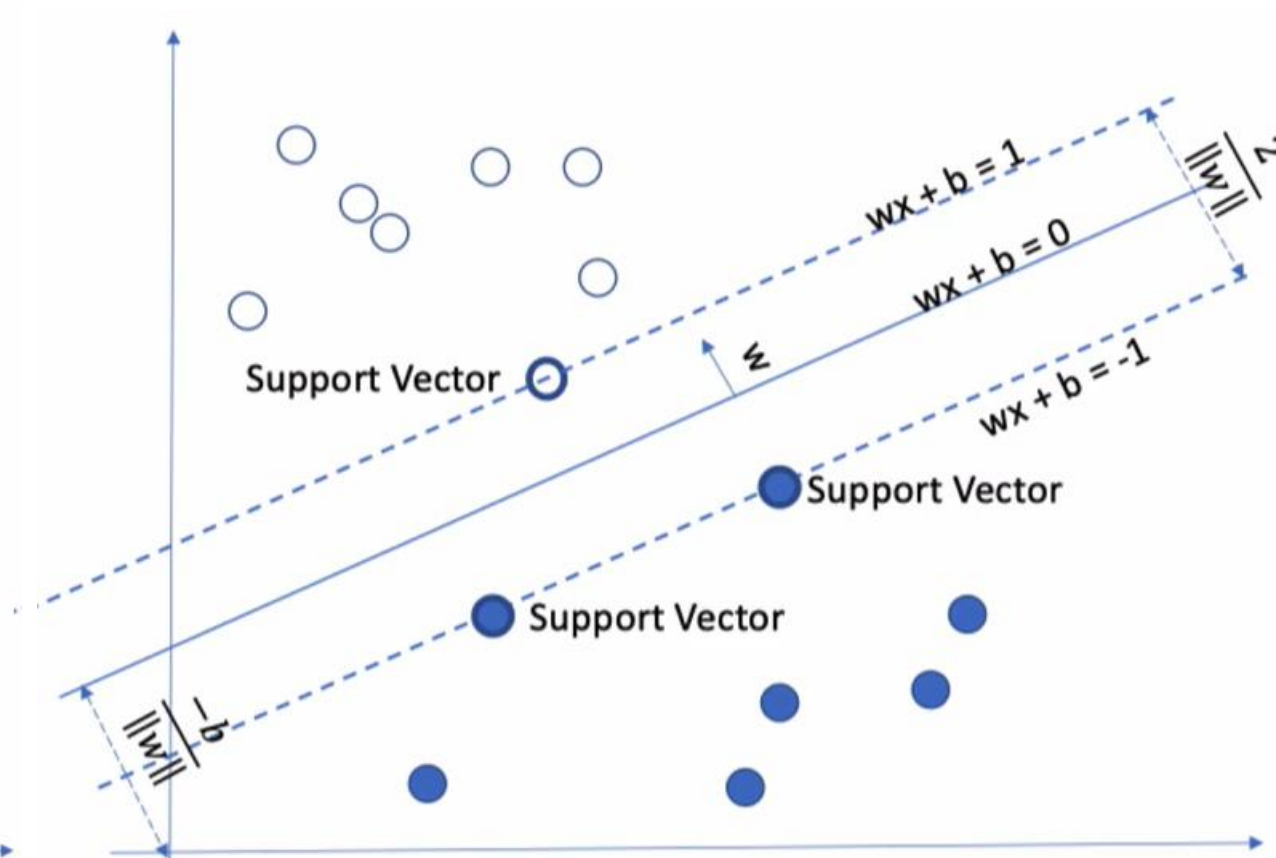
- if the data is linearly separable, then the algorithm will converge
- convergence can be slow ...
- separating line close to training data
- we would prefer a larger **margin** for **generalization**



# What is the best $w$ ?



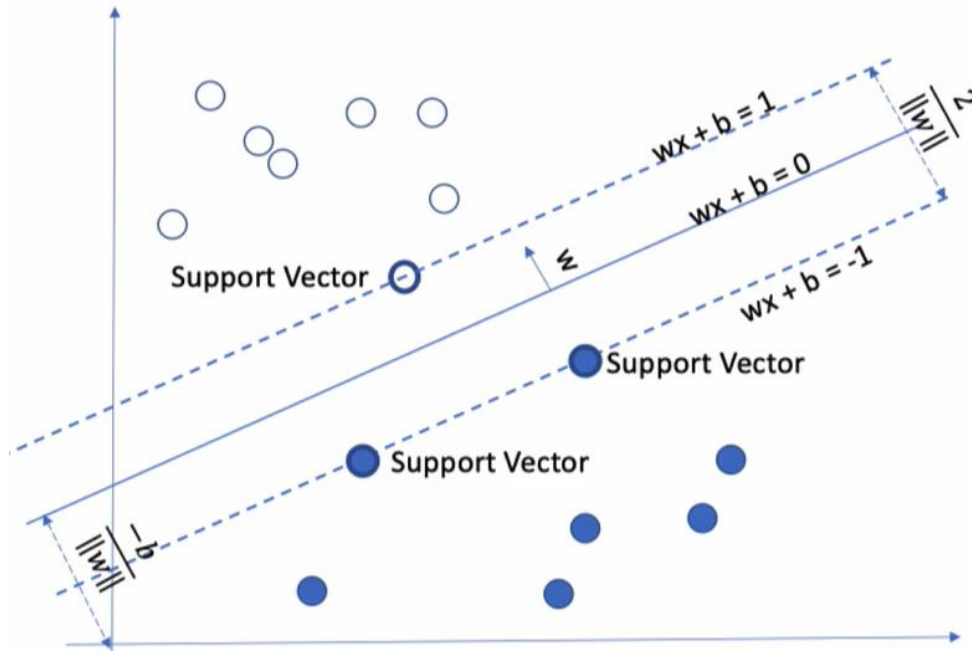
• Linear classifier



svm



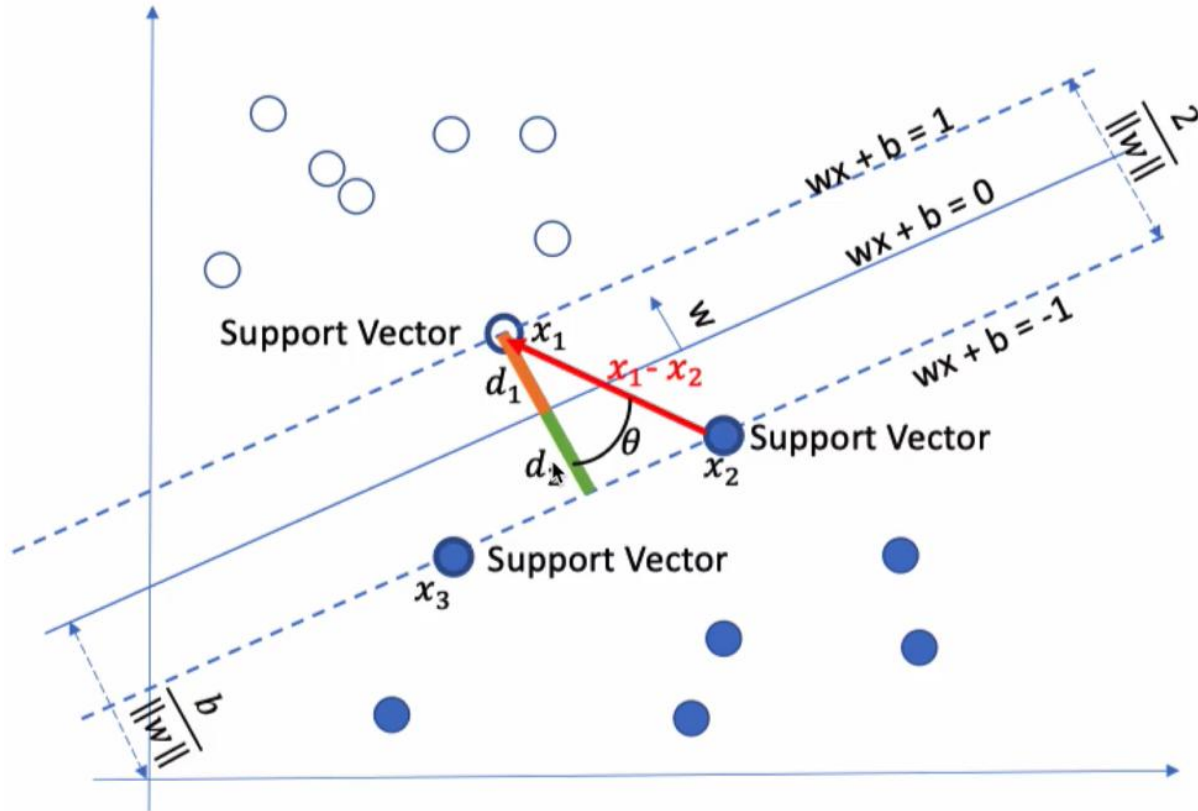
# SVM – sketch derivation



- Since  $\mathbf{w}^\top \mathbf{x} + b = 0$  and  $c(\mathbf{w}^\top \mathbf{x} + b) = 0$  define the same plane, we have the freedom to choose the normalization of  $\mathbf{w}$
- Choose normalization such that  $\mathbf{w}^\top \mathbf{x}_+ + b = +1$  and  $\mathbf{w}^\top \mathbf{x}_- + b = -1$  for the positive and negative support vectors respectively



# SVM – sketch derivation



SVM are also called max-Margin Classifier

$$w^T x_1 + b = 1$$

$$w^T x_2 + b = -1$$

$$(w^T x_1 + b) - (w^T x_2 + b) = 2$$

$$w^T (x_1 - x_2) = 2$$

$$w^T (x_1 - x_2) = \|w\|_2 \|x_1 - x_2\|_2 \cos \theta = 2$$

$$\|x_1 - x_2\|_2 \cos \theta = \frac{2}{\|w\|_2}$$

$$d_1 = d_2 = \frac{\|x_1 - x_2\|_2 \cos \theta}{2} = \frac{\frac{2}{\|w\|_2}}{2} = \frac{1}{\|w\|_2}$$

$$d_1 + d_2 = \frac{2}{\|w\|_2}$$

# SVM – Optimization



- Learning the SVM can be formulated as an optimization:

$$\begin{aligned} \max_{w,b} \quad & \frac{2}{\|w\|_2} \\ \text{s.t.} \quad & y^{(i)}(w^T * x^{(i)} + b) \geq 1, i = 1, 2, \dots, n \end{aligned}$$

- Or equivalently

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

- This is a quadratic optimization problem subject to linear constraints and there is a unique minimum



# The Optimization Problem Solution



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

- The solution involves constructing a *dual problem* where a *Lagrange multiplier*  $\alpha_i$  is associated with every constraint in the primary problem:

Find  $\alpha_1 \dots \alpha_N$  such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$  is maximized and

(1)  $\sum \alpha_i y_i = 0$

(2)  $\alpha_i \geq 0$  for all  $\alpha_i$



# The Optimization Problem Solution



- The solution has the form:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i \quad b = y_k - \mathbf{w}^T \mathbf{x}_k \text{ for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0$$

- Each non-zero  $\alpha_i$  indicates that corresponding  $\mathbf{x}_i$  is a support vector.
- Then the classifying function will have the form:

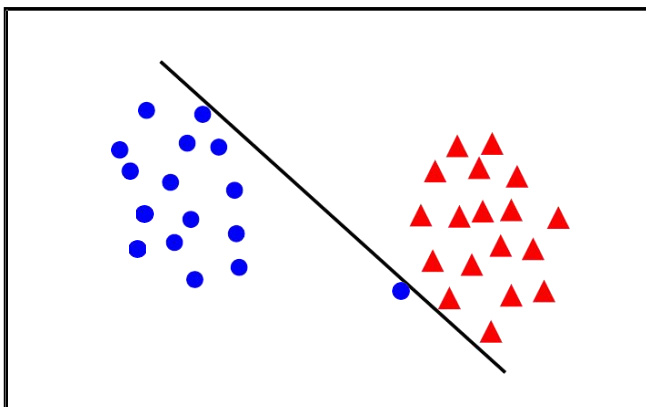
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

- Also keep in mind that solving the optimization problem involved computing the inner products  $\mathbf{x}_i^T \mathbf{x}_j$  between all pairs of training points.

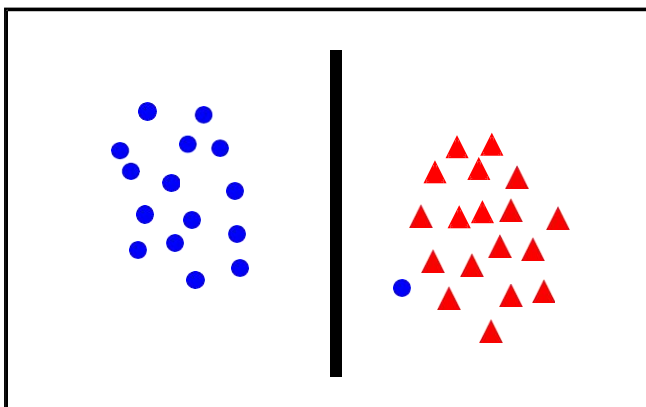




# Linear separability again: What is the best $w$ ?



- the points can be linearly separated but there is a very narrow margin



- but possibly the large margin solution is better, even though one constraint is violated

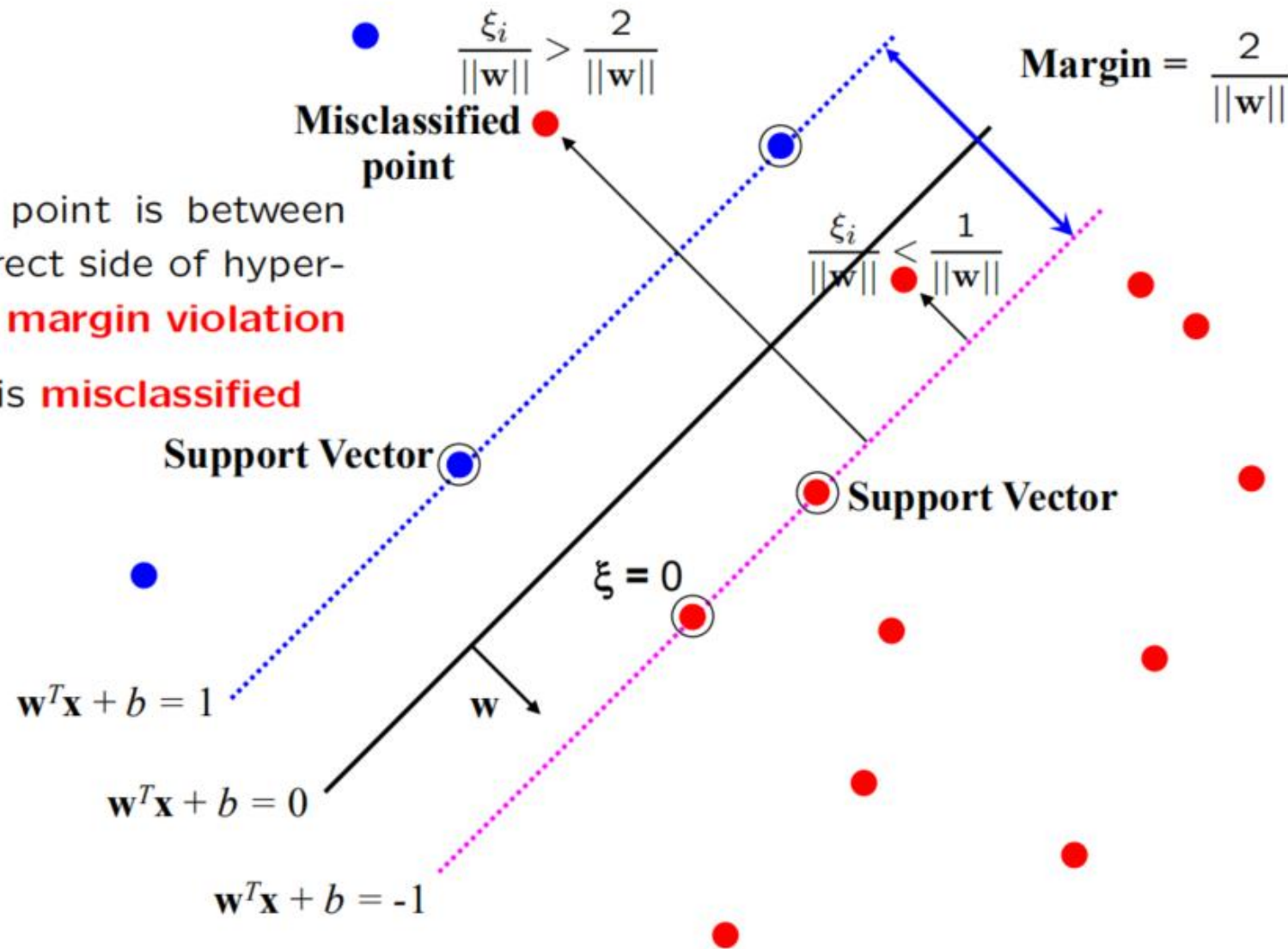
In general there is a trade off between the margin and the number of mistakes on the training data



# Introduce “slack” variables

$$\xi_i \geq 0$$

- for  $0 < \xi \leq 1$  point is between margin and correct side of hyper-plane. This is a **margin violation**
- for  $\xi > 1$  point is **misclassified**





# “Soft” margin solution



The optimization problem becomes

$$\begin{aligned} \min_{\mathbf{w}, b, \xi \geq 0} \quad & \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0 \end{aligned}$$

- Every constraint can be satisfied if  $\xi_i$  is sufficiently large
- $C$  is a **regularization** parameter:
  - small  $C$  allows constraints to be easily ignored  $\rightarrow$  large margin
  - large  $C$  makes constraints hard to ignore  $\rightarrow$  narrow margin
  - $C = \infty$  enforces all constraints: hard margin
- This is still a quadratic optimization problem and there is a unique minimum. Note, there is only one parameter,  $C$ .



# “Soft” margin solution



The optimization problem becomes

$$\begin{aligned} \min_{\mathbf{w}, b, \xi \geq 0} \quad & \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, n \\ & \xi_i \geq 0 \end{aligned}$$

The dual problem for soft margin classification:

Find  $\alpha_1 \dots \alpha_N$  such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$  is maximized and

(1)  $\sum \alpha_i y_i = 0$

(2)  $0 \leq \alpha_i \leq C$  for all  $\alpha_i$



# “Soft” margin solution

Solution to the dual problem is:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$$
$$b = y_k(1 - \xi_k) - \mathbf{w}^T \mathbf{x}_k \text{ where } k = \underset{k'}{\operatorname{argmax}} \alpha_k,$$

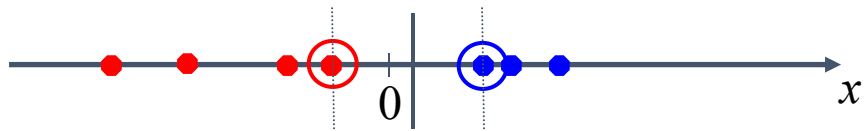
$\mathbf{w}$  is not needed explicitly  
for classification!

$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$



# Non-linear SVMs

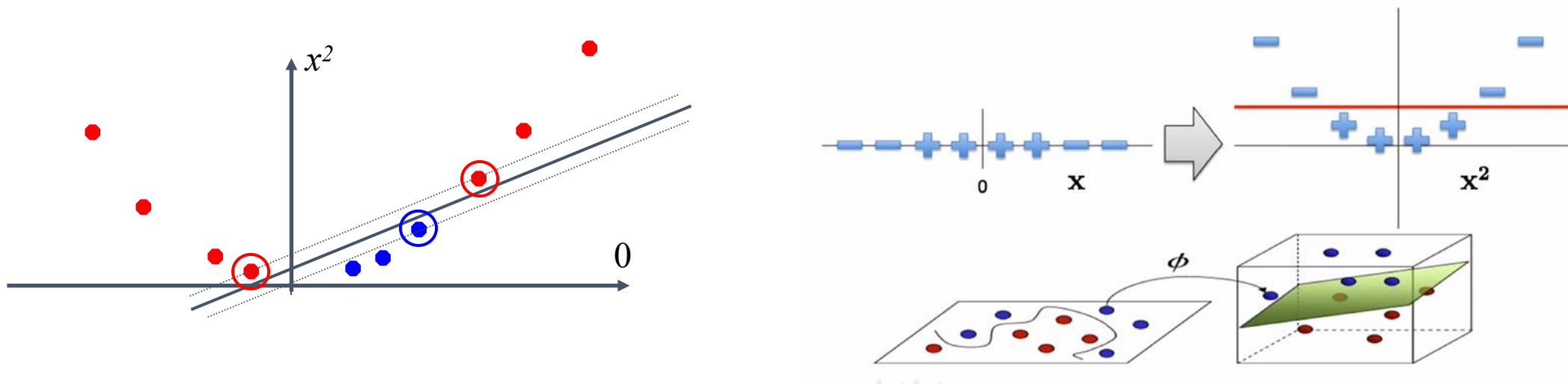
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



- How about... mapping data to a higher-dimensional space:

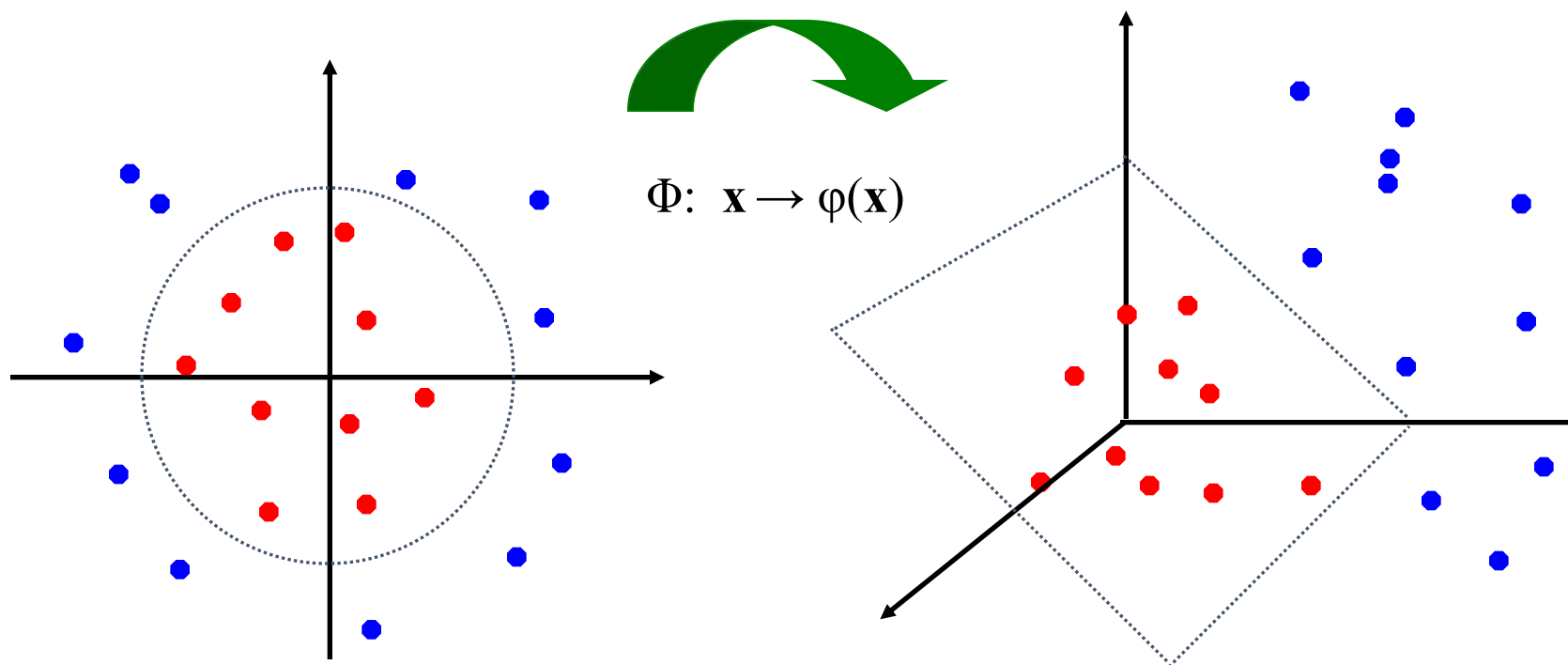




# Non-linear SVMs: Feature spaces



- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:







# The “Kernel Trick”



$$\begin{aligned} \max_{\alpha \geq 0, \lambda \geq 0} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \quad C - \alpha_i - \lambda_i = 0 \end{aligned}$$

两个数据点属于同一类别使值增加, 否则减小

衡量两个数据之间的相似性

不同数据点的权重不同, 不同的类别的权重一致

$$\begin{aligned} \max_{\alpha \geq 0} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_i \alpha_i y_i = 0, \quad \alpha_i \leq C, \quad i = 1, \dots, n \end{aligned}$$

两个数据点属于同一类别使值增加, 否则减小

衡量两个数据之间的相似性



# What Functions are Kernels?



- For some functions  $K(\mathbf{x}_i, \mathbf{x}_j)$  checking that

$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$  can be cumbersome.

- Mercer's theorem:

*Every semi-positive definite symmetric function is a kernel*

- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

$K =$

$K(\mathbf{x}_1, \mathbf{x}_1)$	$K(\mathbf{x}_1, \mathbf{x}_2)$	$K(\mathbf{x}_1, \mathbf{x}_3)$	$\dots$	$K(\mathbf{x}_1, \mathbf{x}_N)$
$K(\mathbf{x}_2, \mathbf{x}_1)$	$K(\mathbf{x}_2, \mathbf{x}_2)$	$K(\mathbf{x}_2, \mathbf{x}_3)$		$K(\mathbf{x}_2, \mathbf{x}_N)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$K(\mathbf{x}_N, \mathbf{x}_1)$	$K(\mathbf{x}_N, \mathbf{x}_2)$	$K(\mathbf{x}_N, \mathbf{x}_3)$	$\dots$	$K(\mathbf{x}_N, \mathbf{x}_N)$



# Examples of Kernel Functions



- Linear:  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polynomial of power  $p$ :  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
- Gaussian (radial-basis function network):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- Sigmoid:  $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$



# Non-linear SVMs Mathematically



- Dual problem formulation:

Find  $\alpha_1 \dots \alpha_N$  such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$  is maximized and

(1)  $\sum \alpha_i y_i = 0$

(2)  $\alpha_i \geq 0$  for all  $\alpha_i$

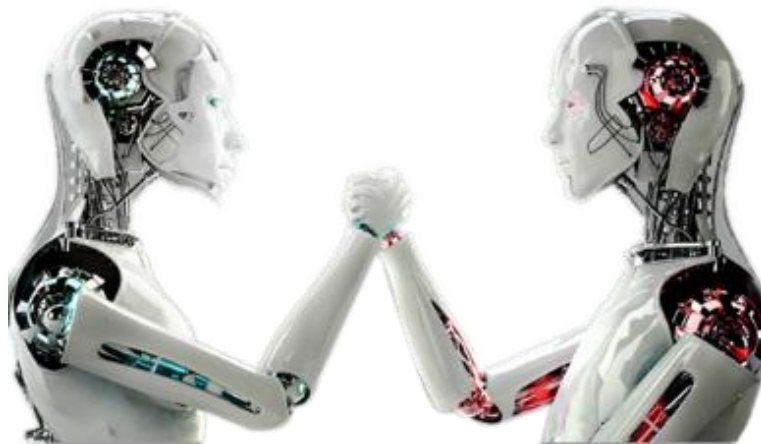
- The solution is:

$$f(\mathbf{x}) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

- Optimization techniques for finding  $\alpha_i$ 's remain the same!



# Lab Assignment





# Lab Task



Complete the exercises and questions in the `Lab10.svm.md`

# Thanks

贾艳红 Jana

Email: [jiayh@mail.sustech.edu.cn](mailto:jiayh@mail.sustech.edu.cn)

