

# Lecture 5: Support Vector Machine

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

<https://shuaili8.github.io>

<https://shuaili8.github.io/Teaching/VE445/index.html>



# Outline

- Linear classifiers and the margins
- Objective of the SVM
- Lagrangian method in convex optimization
- Solve SVM by Lagrangian duality
- Regularization
- Kernel method
- SMO algorithm to solve the Lagrangian multipliers

References: <http://cs229.stanford.edu/notes/cs229-notes3.pdf>

# Review: Label decision of logistic regression

- Logistic regression provides the probability

$$p_{\theta}(y = 1|x) = \sigma(\theta^{\top} x) = \frac{1}{1 + e^{-\theta^{\top} x}}$$

$$p_{\theta}(y = 0|x) = \frac{e^{-\theta^{\top} x}}{1 + e^{-\theta^{\top} x}}$$

- The final label of an instance is decided by setting a threshold  $h$

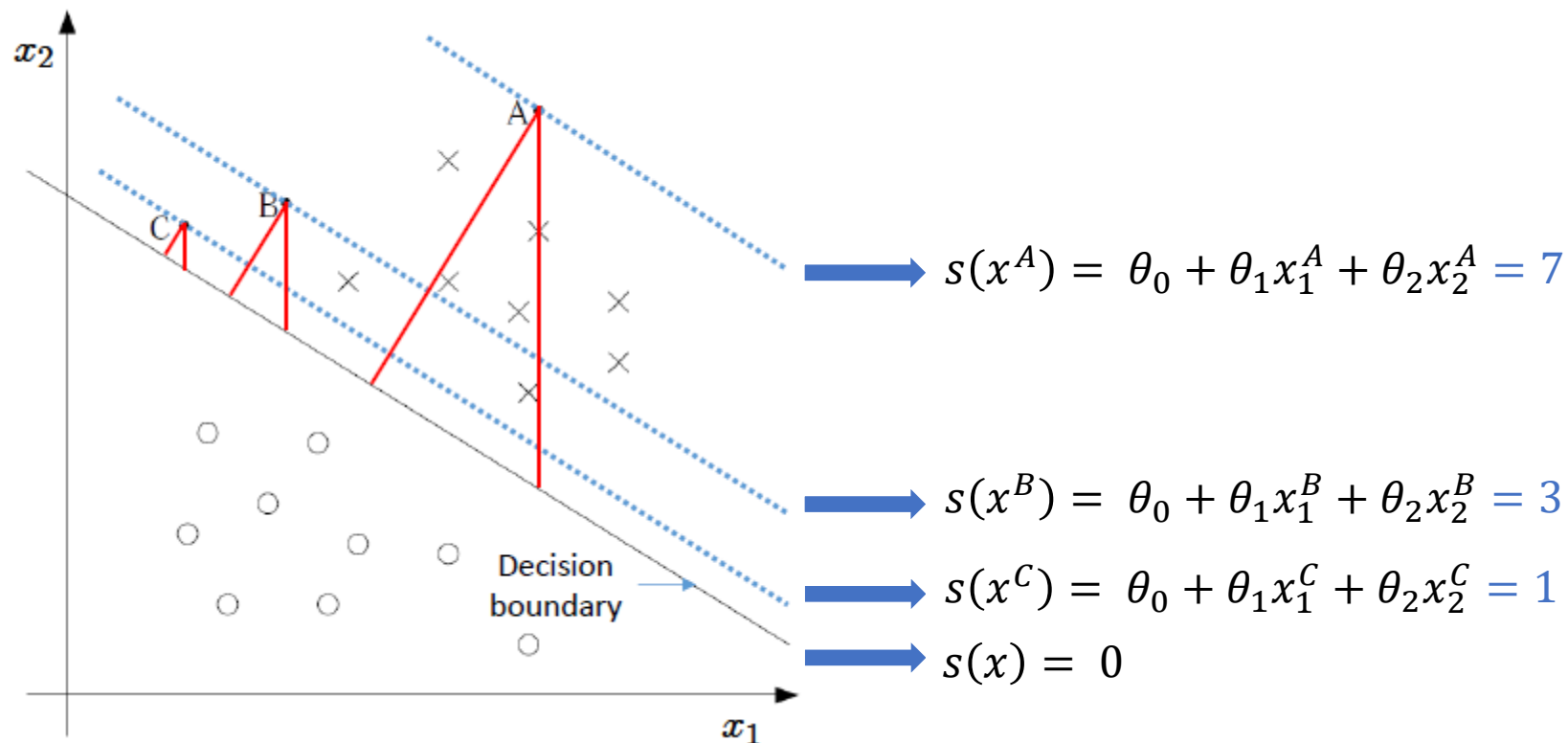
$$\hat{y} = \begin{cases} 1, & p_{\theta}(y = 1|x) > h \\ 0, & \text{otherwise} \end{cases}$$

# Scores of logistic regression

- Let  $s(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ , so the probability in logistic regression is defined as  $p_{\theta}(y = 1|x) = \frac{1}{1+e^{-s(x)}}$
- Positive prediction means positive scores
- Negative prediction means negative scores
- The absolute value of the score  $s(x)$  is **proportional** to the distance  $x$  to the decision boundary  $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

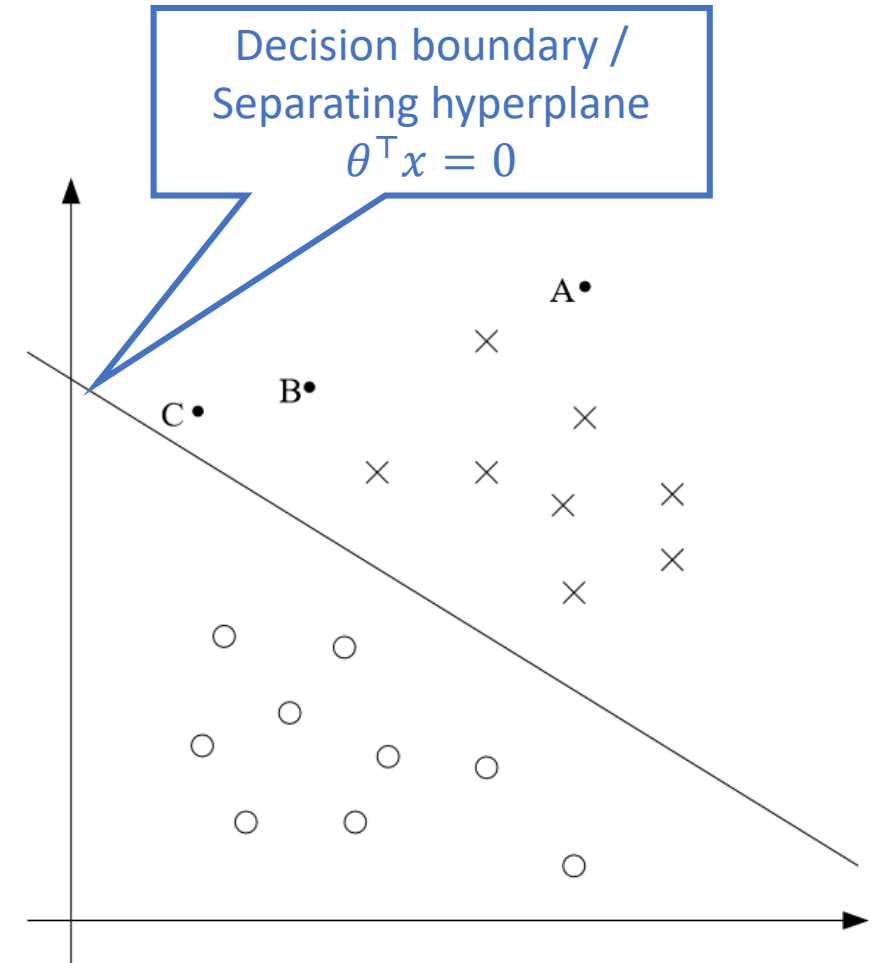
# Illustration of logistic regression

- The higher score, the larger distance to the decision boundary, the higher confidence. E.g.



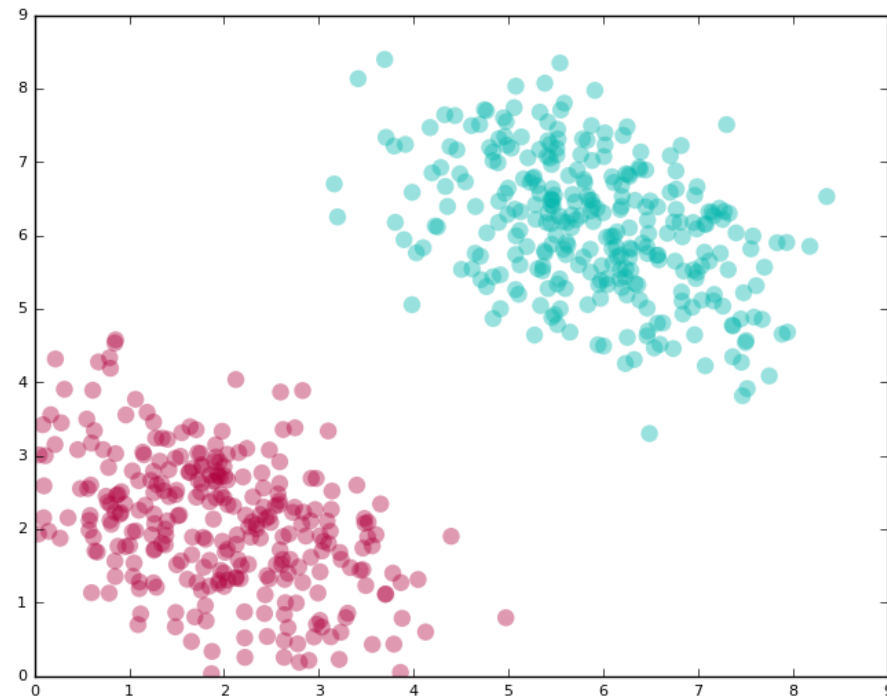
# Intuition

- Positive when
$$p_{\theta}(y = 1|x) = h_{\theta}(x) = \sigma(\theta^{\top}x) \geq 0.5$$
or
$$\theta^{\top}x \geq 0$$
- Point *A*
  - Far from decision boundary
  - More confident to predict the label 1
- Point *C*
  - Near decision boundary
  - A small change to the decision boundary could cause prediction to be  $y = 0$



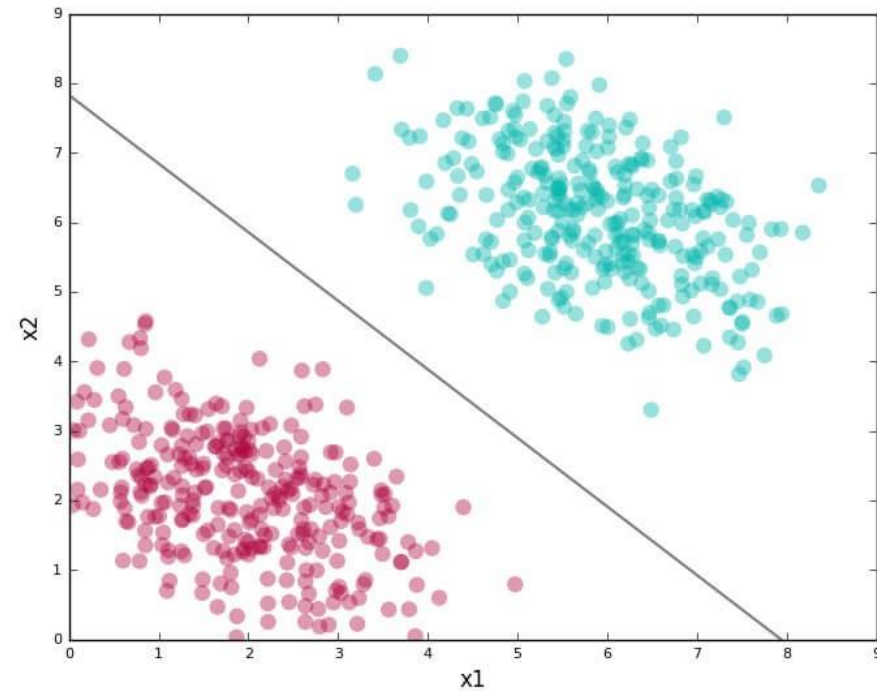
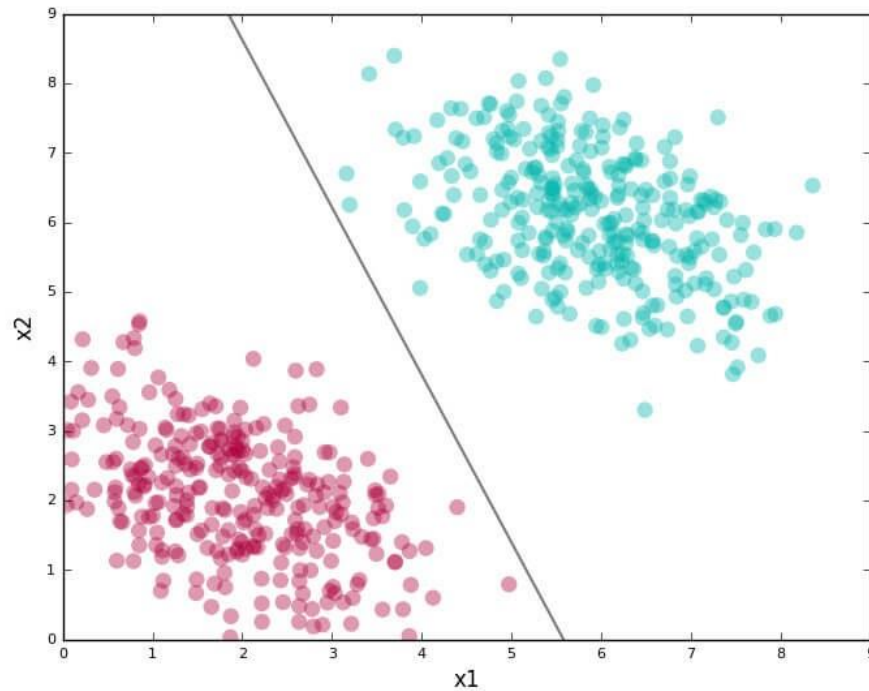
# Example

- Given a dataset of two classes, how to find a line to separate them?



## Example (cont.)

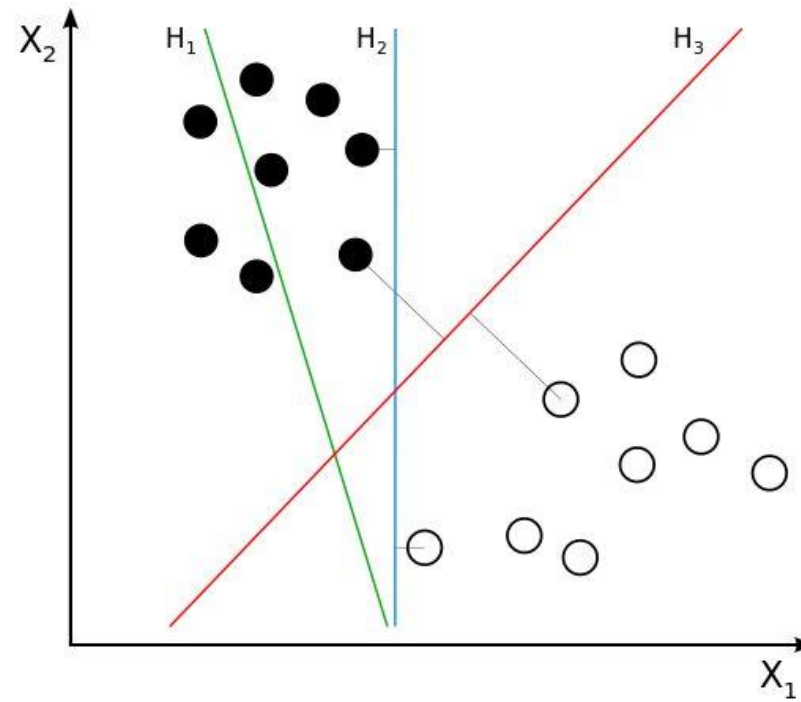
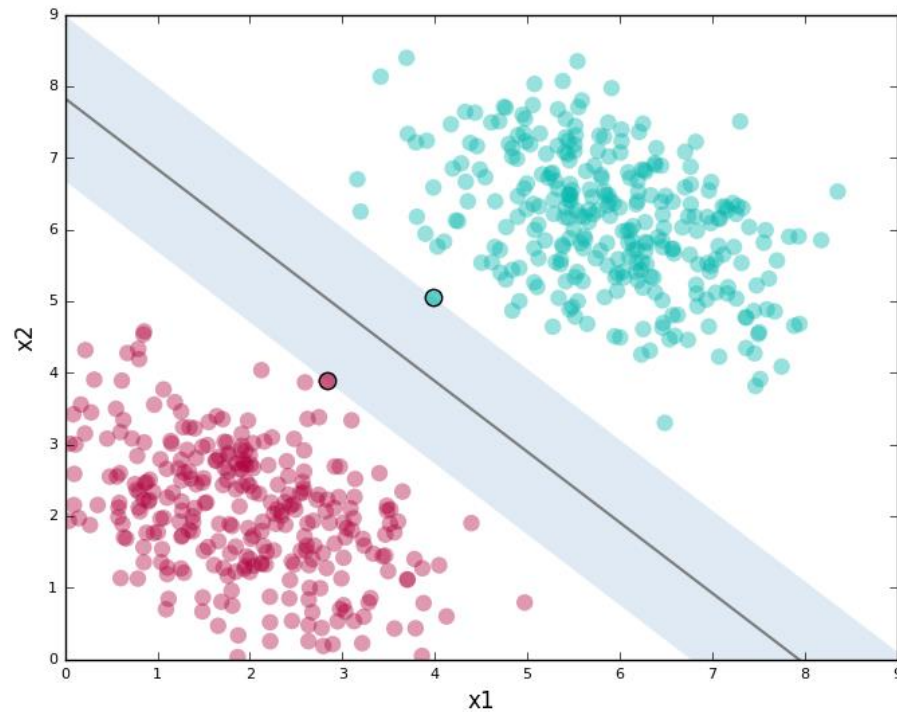
- Both the two solutions can separate the data perfectly, but we prefer the one on the right, why?





# Example (cont.)

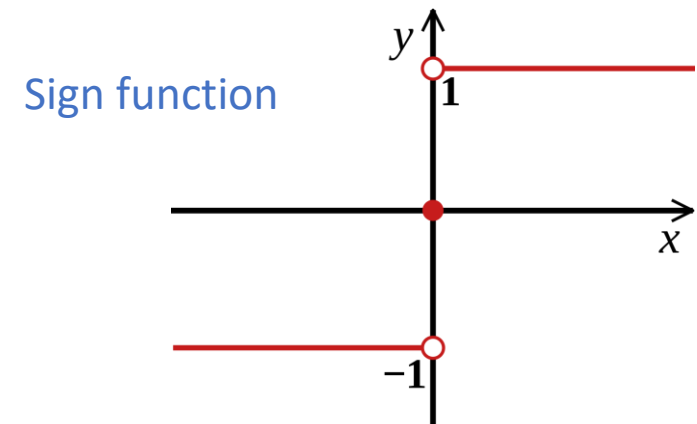
- It makes us feel safe because it provides the most margin!



- These are the support vectors, and the model is called support vector machine.

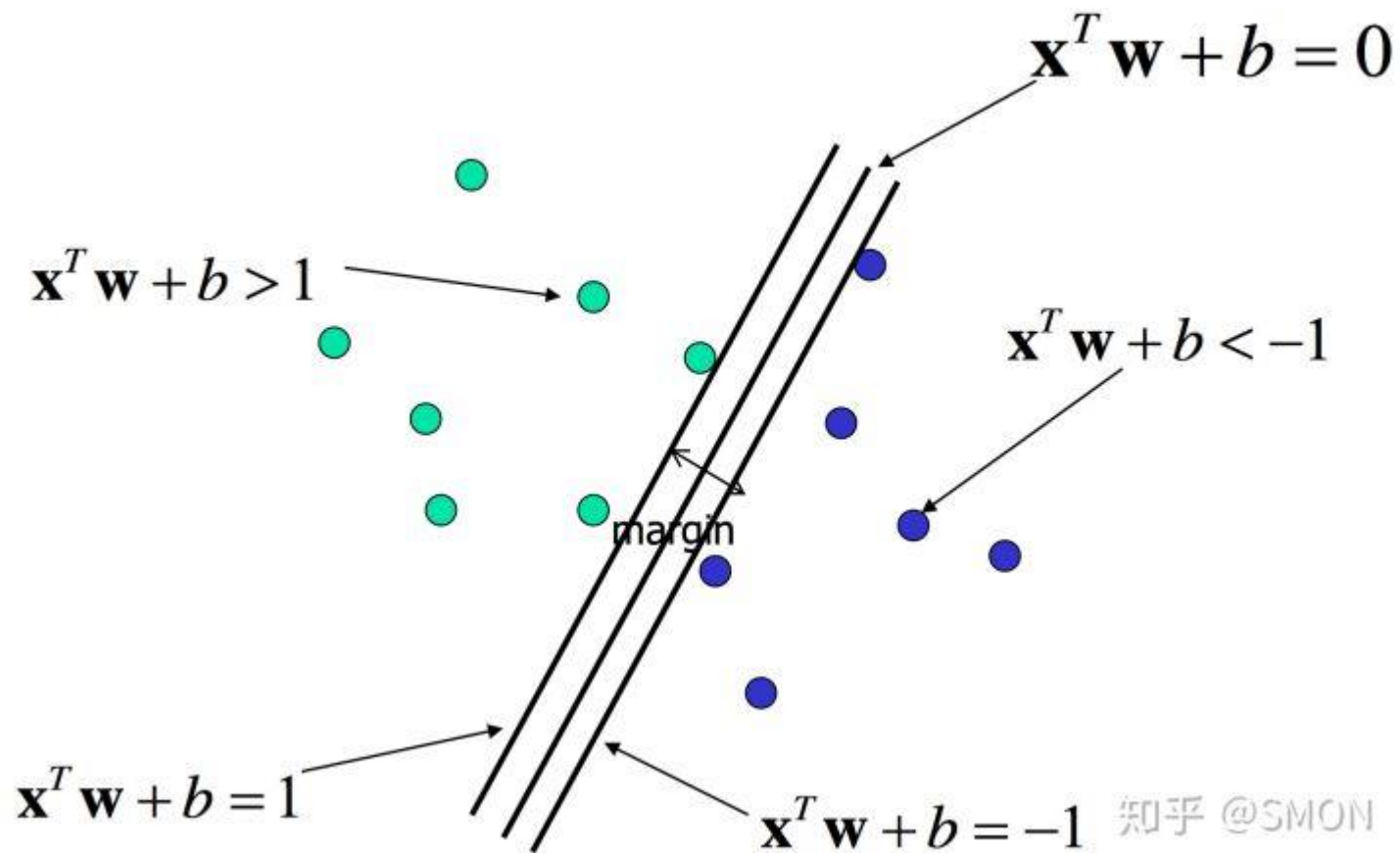
# Notations for SVM

- Feature vector  $x$
- Class label  $y \in \{-1, 1\}$ 
  - Instead of  $\{0, 1\}$
- Parameters
  - Intercept  $b$ 
    - We also drop the convention we had previously of letting  $x_0 = 1$  be an extra coordinate in the input feature vector
  - Feature weight vector  $w$
- Label prediction
  - $h_{w,b}(x) = g(w^\top x + b)$
  - $g(z) = \begin{cases} +1 & z \geq 0 \\ -1 & \text{otherwise} \end{cases}$
  - Directly output the label
    - Without estimating probability first (compared with logistic regression)



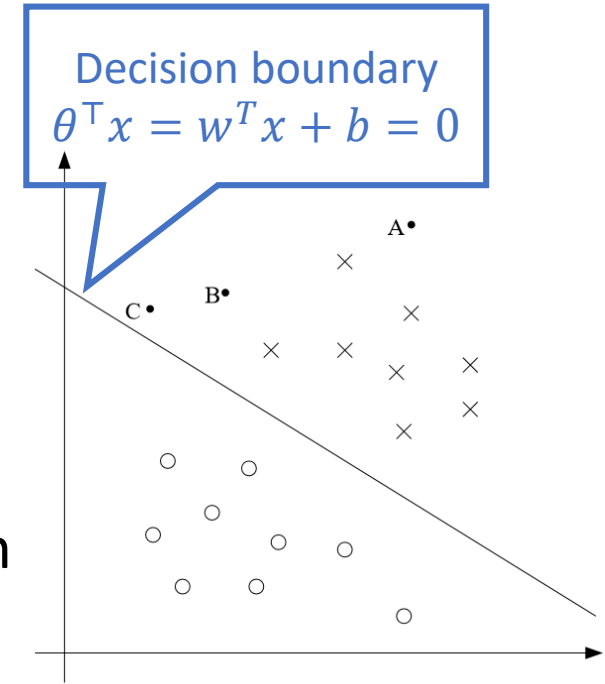
# Hyperplane and margin

- Idea of using  $y \in \{-1, 1\}$

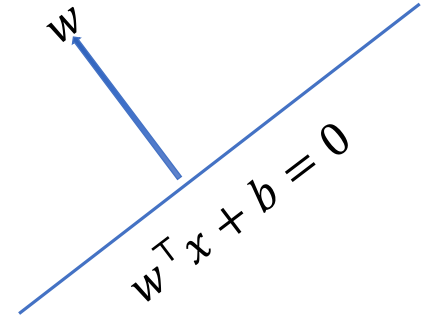


# Functional margin

- **Functional margin** of  $(w, b)$  with respect to  $(x, y)$  is  $\gamma = y(w^\top x + b)$ 
  - $w^\top x + b$  is the **score** of  $x$
  - When  $y = 1$ , **large** positive  $w^\top x + b$  value would give a high confidence
  - When  $y = -1$ , **large** negative  $w^\top x + b$  value would give a high confidence
  - $y(w^\top x + b) > 0$  means the prediction is correct
  - But changing  $(w, b)$  to  $(2w, 2b)$  would **increase** the functional margin
    - **Without** changing the decision boundary  $w^\top x + b = 0$



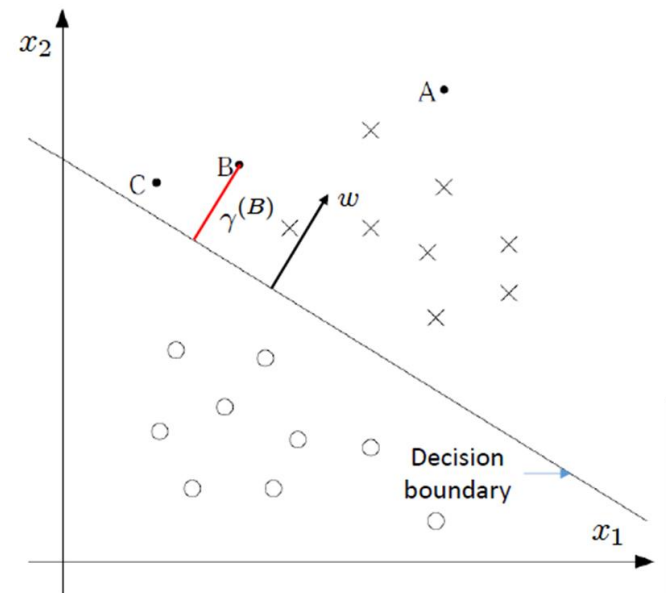
# Geometric margin



- $w$  vector is orthogonal to the decision boundary
- **Geometric margin** is the distance of the point to the decision boundary
  - For **positive** prediction points
  - $x - \gamma \frac{w}{\|w\|}$  lies on the decision boundary
  - $w^T \left( x - \gamma \frac{w}{\|w\|} \right) + b = 0$
  - Solve it, get

$$\gamma = \frac{w^T x + b}{\|w\|}$$

- In general,  $\gamma = y(w^T x + b)$  with  $\|w\| = 1$



# Objective of an SVM

- Given a training set

$$S = \{(x_i, y_i)\}, i = 1, \dots, N$$

margin is the smallest geometric margin

$$\gamma = \min_{i=1, \dots, n} \gamma^i$$

- Objective: maximize the margin

$$\max_{\gamma, w, b} \gamma$$

$$s. t. \quad y^i (w^\top x^i + b) \geq \gamma, \quad i = 1, \dots, N$$

$$\|w\| = 1$$

Non-convex constraint  
 $\|w\| \leq 1$  is convex

which is equivalent to

$$\max_{\gamma, w, b} \frac{\gamma}{\|w\|}$$

Non-convex objective

$$s. t. \quad y^i (w^\top x^i + b) \geq \gamma, \quad i = 1, \dots, N$$

Scaling  $(\gamma, w)$  as  
 $(c\gamma, cw)$  doesn't  
change the problem

# Objective of an SVM (cont.)

- Functional margin scales w.r.t.  $(w, b)$  without changing the decision boundary
- Fix the functional margin as 1, that is let  $\gamma = 1$

- Then the objective is

$$\begin{array}{ll} \max_{w,b} & \frac{1}{\|w\|} \\ \text{s.t.} & y^i(w^\top x^i + b) \geq 1, \quad i = 1, \dots, N \end{array}$$

or equivalently

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

Can be efficiently solved by quadratic programming (QP)

# Lagrange Duality




# Lagrangian for convex optimization

- Given a convex optimization problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & h_i(w) = 0, \quad i = 1, \dots, l \end{aligned}$$

- The Lagrangian of this problem is defined as

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

 Lagrangian multipliers

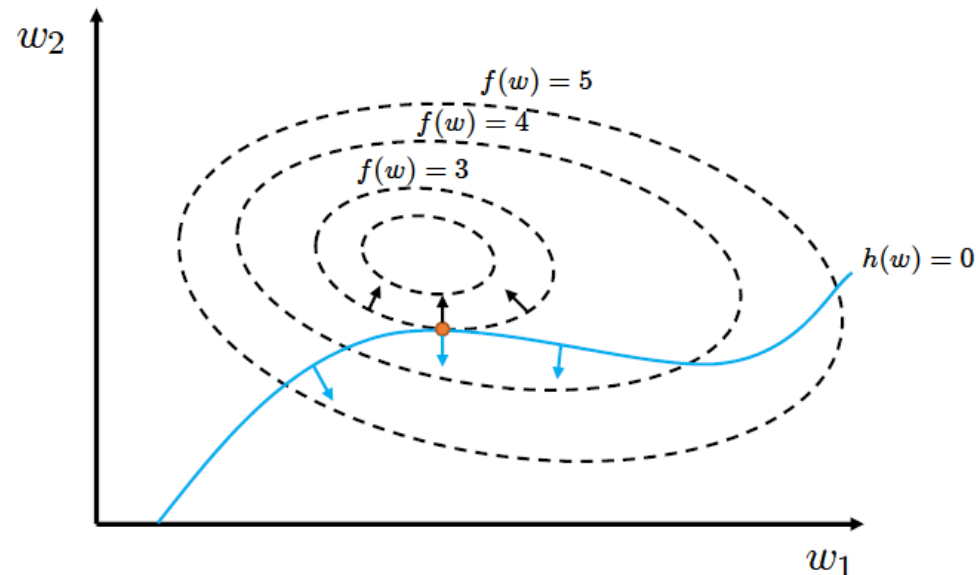
- Solving

$$\frac{\partial \mathcal{L}(w, \beta)}{\partial w} = 0 \quad \frac{\partial \mathcal{L}(w, \beta)}{\partial \beta} = 0$$

yields the solution of the original optimization problem

# Geometric interpretation

- With only one constraint



$$\mathcal{L}(w, \beta) = f(w) + \beta h(w)$$
$$\frac{\partial \mathcal{L}(w, \beta)}{\partial w} = \frac{\partial f(w)}{\partial w} + \beta \frac{\partial h(w)}{\partial w} = 0$$

The two gradients are on the same line but with different direction

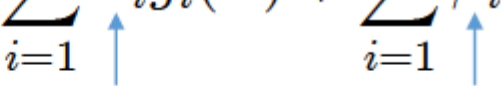
# With inequality constraints

- What if there are inequality constraint?

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, l \end{aligned}$$

- The Lagrangian of this problem is defined as:

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^l \beta_i h_i(w)$$

  
Lagrangian multipliers

# More on primal problem

- Primal problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, l \end{aligned}$$

- Generalized Lagrangian

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^l \beta_i h_i(w)$$

- Consider quantity

primal  $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$

- If a given  $w$  violates any constraints, i.e.  $g_i(w) > 0$  or  $h_i(w) \neq 0$ , then

$$\theta_{\mathcal{P}}(w) = +\infty$$

- If all constraints are satisfied for  $w$ , then

$$\theta_{\mathcal{P}}(w) = f(w)$$

# More on primal problem (cont.)

- Primal problem

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, l \end{aligned}$$

- Consider quantity

primal  $\theta_{\mathcal{P}}(w) = \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ +\infty & \text{otherwise} \end{cases}$$

- Generalized Lagrangian

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^l \beta_i h_i(w)$$

# More on primal problem (cont.)

- The minimization problem

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$$

is the same with the original problem


$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \leq 0, \quad i = 1, \dots, k \\ & h_i(w) = 0, \quad i = 1, \dots, l \end{aligned}$$

- The value of the primal problem

$$p^* = \min_w \theta_{\mathcal{P}}(w)$$

# Dual problem

- $\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta)$

- Define   $\theta_{\mathcal{D}}(\alpha, \beta) = \min_w \mathcal{L}(w, \alpha, \beta)$

- Dual optimization problem

$$\max_{\alpha, \beta: \alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta) = \max_{\alpha, \beta: \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta)$$

with the value

$$d^* = \max_{\alpha, \beta: \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta)$$

# Primal problem vs. dual problem

$$d^* = \max_{\alpha, \beta: \alpha_i \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \min_w \max_{\alpha, \beta: \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta) = p^*$$

- Proof

$$\min_w \mathcal{L}(w, \alpha, \beta) \leq \mathcal{L}(w, \alpha, \beta), \forall w, \alpha \geq 0, \beta$$

$$\Rightarrow \max_{\alpha, \beta: \alpha \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \max_{\alpha, \beta: \alpha \geq 0} \mathcal{L}(w, \alpha, \beta), \forall w$$

$$\Rightarrow \max_{\alpha, \beta: \alpha \geq 0} \min_w \mathcal{L}(w, \alpha, \beta) \leq \min_w \max_{\alpha, \beta: \alpha \geq 0} \mathcal{L}(w, \alpha, \beta)$$

- But under certain condition  $d^* = p^*$



# Karush-Kuhn-Tucker (KKT) Conditions

- Suppose

- $f$  and  $g_i$ 's are convex
- $h_i$ 's are affine
- $g_i$ 's are all strictly feasible
  - There exists  $w$  such that  $g_i(w) < 0$  for all  $i$

- Then there must exist  $w^*, \alpha^*, \beta^*$

- $w^*$  is the solution of the primal problem
- $\alpha^*, \beta^*$  are the solution of the dual problem
- And the values of the two problems are equal

$$p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$$

- $w^*, \alpha^*, \beta^*$  satisfy the KKT conditions:

$$\frac{\partial}{\partial w_i} \mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, n$$

$$\frac{\partial}{\partial \beta_i} \mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, l$$

KKT dual

complementarity

condition

→  $\alpha_i^* g_i(w^*) = 0, \quad i = 1, \dots, k$

$$g_i(w^*) \leq 0, \quad i = 1, \dots, k$$

$$\alpha^* \geq 0, \quad i = 1, \dots, k$$

- If  $\alpha_i^* > 0$ , then  $g_i(w^*) = 0$

- The converse is also true

- If some  $w, a, b$  satisfy the KKT conditions, then it is also a solution to the primal and dual problems
- More details can be found in Boyd's book "Convex optimization"

Back to SVM

# Rewrite the SVM objective

- The objective of SVM is

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

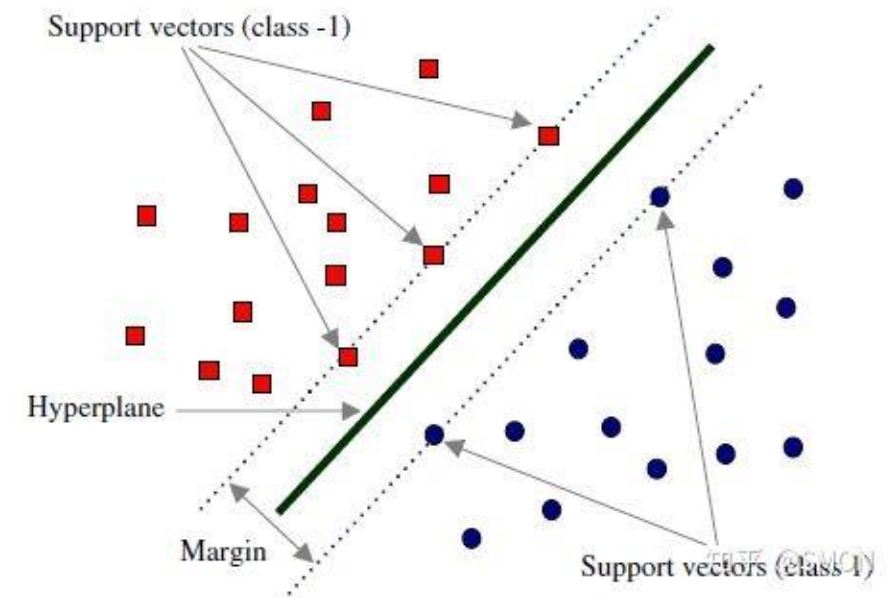
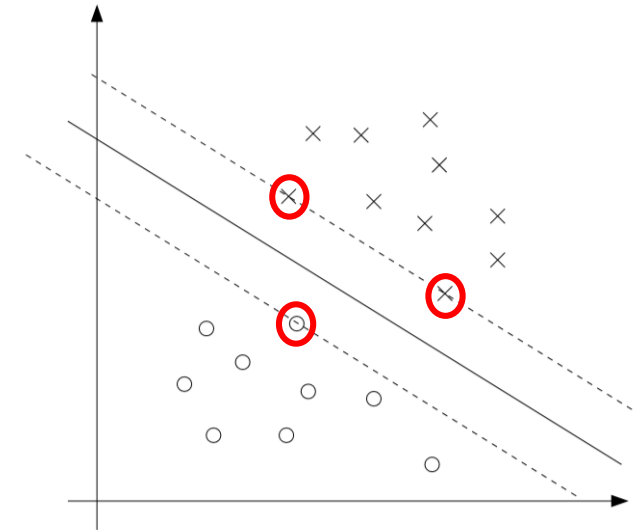
- Rewrite the constraints as

$$g_i(w) = -y^i (w^\top x^i + b) + 1 \leq 0$$

- It is equivalent to solve the dual problem
- Note that from the KKT dual complementarity condition,  $\alpha_i > 0$  is only possible for training samples with  $g_i(w) = 0$

# Support vectors

- The points with smallest margins
- $g_i(w) = 0$ 
  - $-y^i(w^\top x^i + b) + 1 = 0$
  - Positive support vectors
    - $w^\top x + b = 1$
  - Negative support vectors
    - $w^\top x + b = -1$
- Only support vectors decide the decision boundary
  - Moving or deleting non-support points doesn't change the decision boundary



# Lagrangian of SVM

- SVM objective:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$
$$g_i(w) = -y^i(w^\top x^i + b) + 1 \leq 0, i = 1, \dots, N$$

- Lagrangian

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^N \alpha_i [1 - y^i(w^\top x^i + b)]$$

# Solve the dual

- $L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^N \alpha_i [1 - y^i (w^\top x^i + b)]$
- Let the partial derivative to be zero:
  - $\frac{\partial L(w, b; \alpha)}{\partial w} = w - \sum_{i=1}^N \alpha_i y^i x^i = 0$
  - $\frac{\partial L(w, b; \alpha)}{\partial b} = -\sum_{i=1}^N \alpha_i y^i = 0$
- Then substitute them back to  $L$ :
  - $L(w, b, \alpha)$ 
$$= \frac{1}{2} \left\| \sum_{i=1}^N \alpha_i y^i x^i \right\|^2 + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i y^i \left( \sum_{j=1}^N \alpha_j y^j x^j \right)^\top x^i + b \sum_{i=1}^N \alpha_i y^i$$
$$= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^{j^\top} x^i$$

# Solve the dual (cont.)

- $\max_{\alpha \geq 0} \theta_D(\alpha) = \max_{\alpha \geq 0} \min_{w, b} L(w, b, \alpha)$
- Dual problem

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^\top x^i \\ \text{s. t.} \quad & \alpha_i \geq 0, i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \end{aligned}$$

Can check the KKT conditions hold

- Then solve  $\alpha^*$  by SMO

$$\begin{aligned} \frac{\partial}{\partial w_i} \mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \quad i = 1, \dots, n \\ \frac{\partial}{\partial \beta_i} \mathcal{L}(w^*, \alpha^*, \beta^*) &= 0, \quad i = 1, \dots, l \\ \text{KKT dual complementarity condition} &\longrightarrow \alpha_i^* g_i(w^*) = 0, \quad i = 1, \dots, k \\ &g_i(w^*) \leq 0, \quad i = 1, \dots, k \\ &\alpha^* \geq 0, \quad i = 1, \dots, k \end{aligned}$$

# Solve $w^*$ and $b^*$

- With  $\alpha^*$

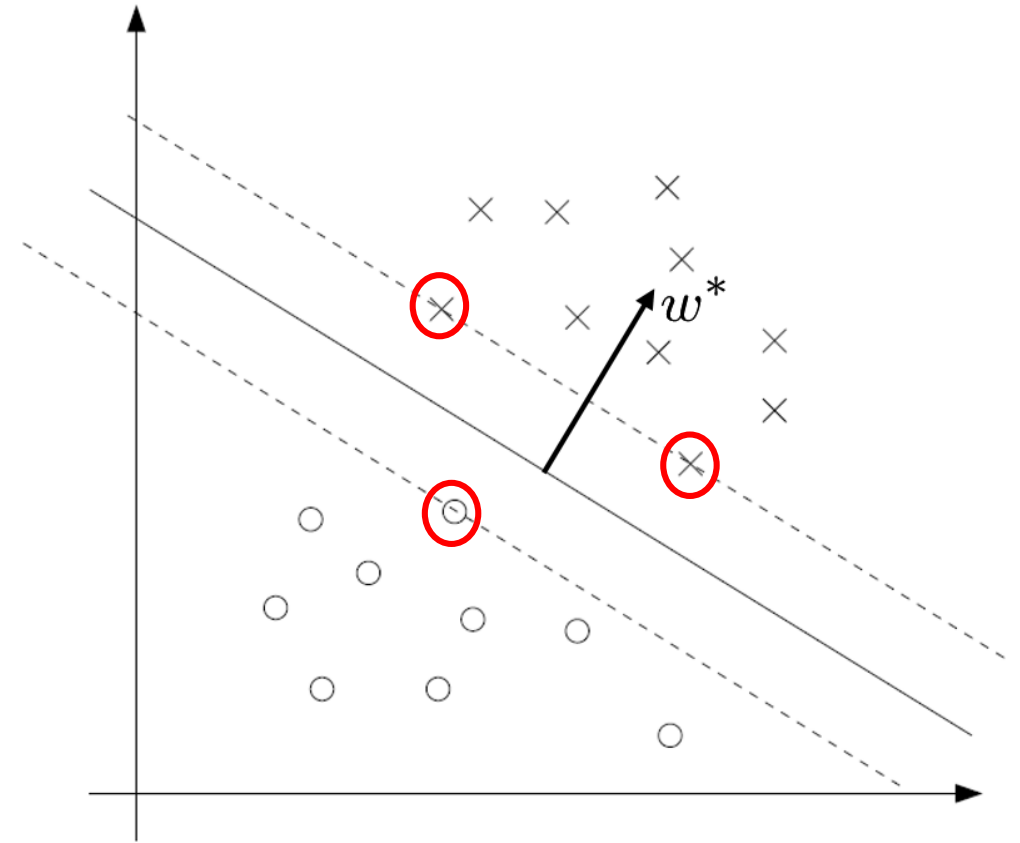
$$w = \sum_{i=1}^N \alpha_i y^i x^i$$

- $\alpha_i > 0$  only holds on support vectors

- Then

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*\top} x^{(i)} + \min_{i:y^{(i)}=1} w^{*\top} x^{(i)}}{2}$$

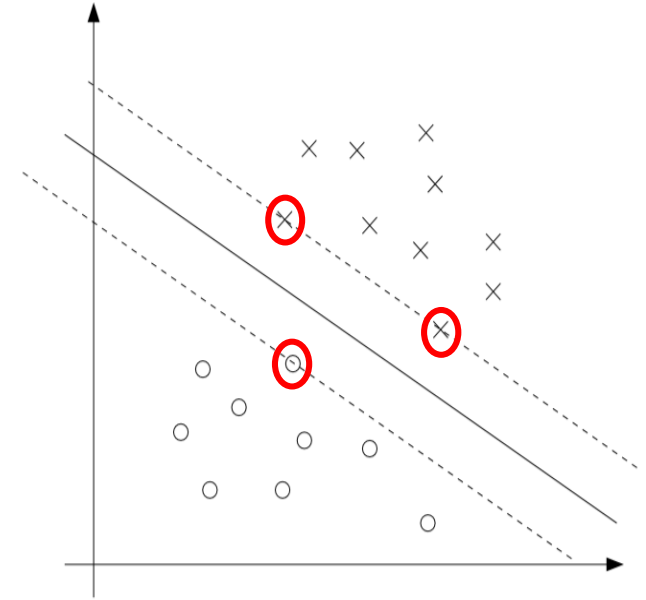
Check it!





# Predicting values

- $w^\top x + b = \left( \sum_{i=1}^N \alpha_i y^i x^i \right)^\top x + b$   
 $= \sum_{i=1}^N \alpha_i y^i \langle x^i, x \rangle + b$



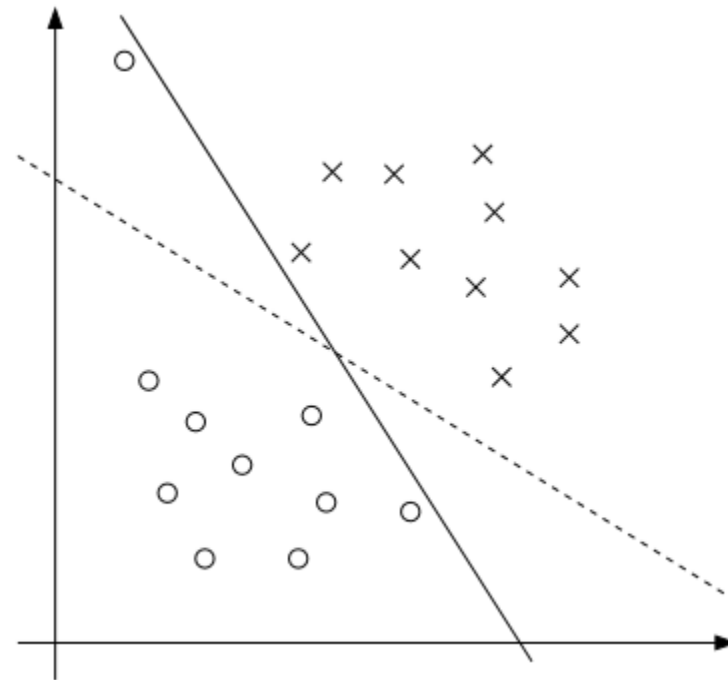
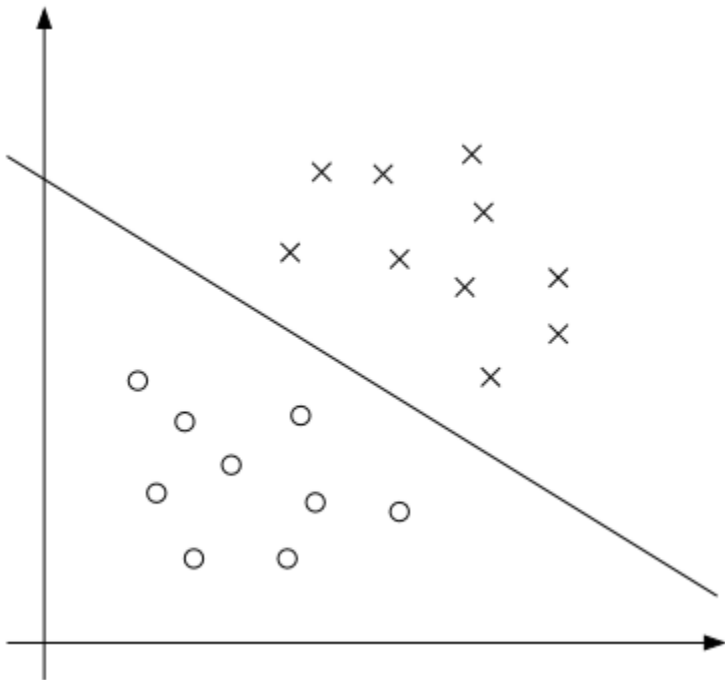
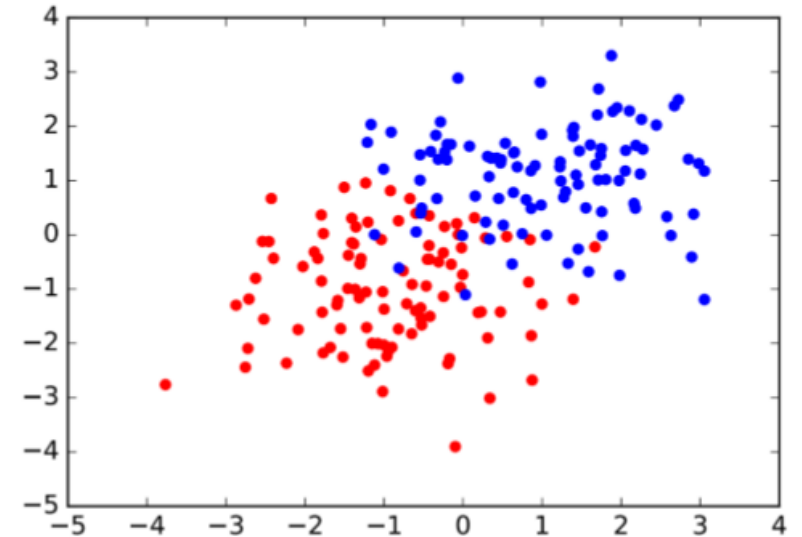
- Only need to calculate the inner product of  $x$  with support vectors
- Prediction is

$$y = \text{Sign}(w^\top x + b)$$

# Regularization and the Non-Separable Case

# Motivation

- SVM assumes data is linearly separable
  - But some data is linearly non-separable
  - SVM is susceptible to outliers



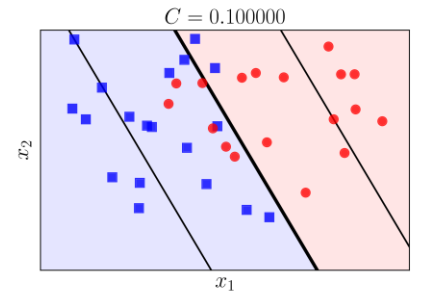
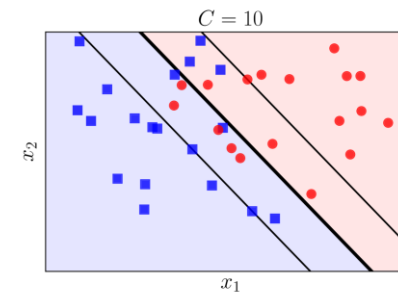
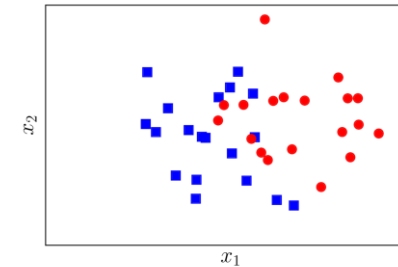
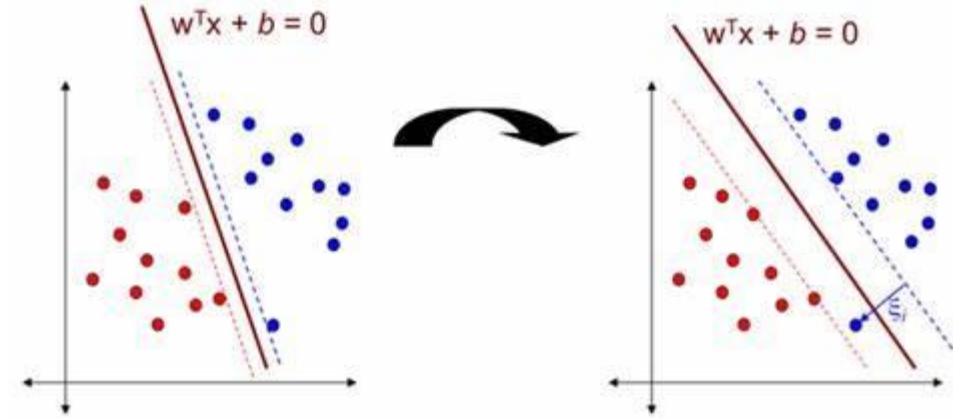
# Solution – Soft margin

- To make the algorithm work for non-linearly separable datasets as well as be less sensitive to outliers
- Add slack variables

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad \leftarrow L^1 \text{ regularization}$$

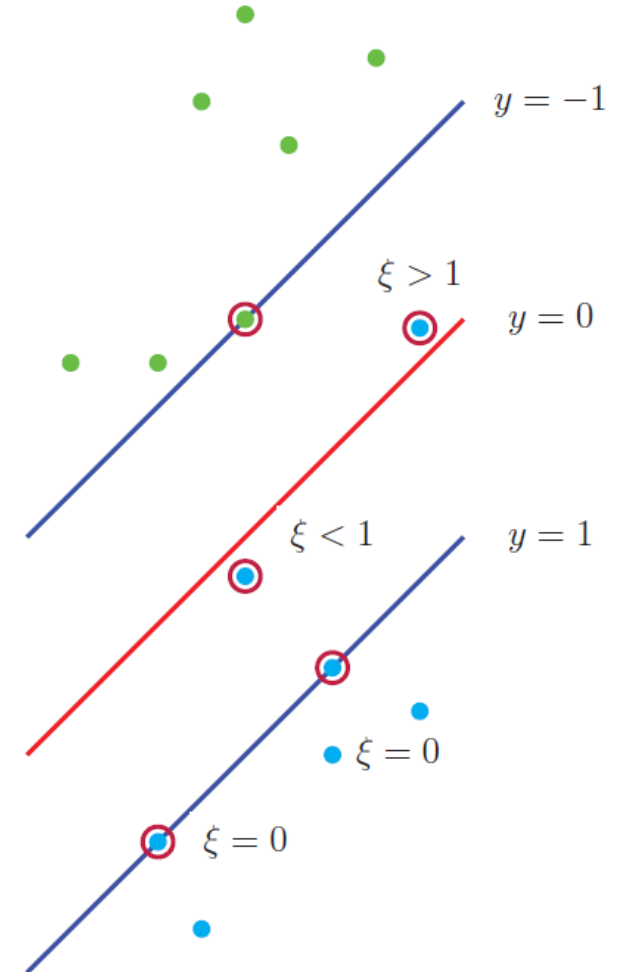
$$\text{s.t.} \quad y^i (w^\top x^i + b) \geq 1 - \xi_i, \quad i = 1, \dots, N$$

$$\xi_i \geq 0, \quad i = 1, \dots, N$$



# Example

- Correctly classified points beyond the support line with  $\xi = 0$
- Correctly classified points on the support line (support vectors) with  $\xi = 0$
- Correctly classified points inside the margin with  $0 < \xi < 1$
- The misclassified points inside the margin with slack  $1 < \xi < 2$
- The misclassified points outside the margin with slack  $\xi > 2$



# Solve the Lagrangian dual problem

- Lagrangian

$$L(w, b, \xi, \alpha, r) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i + \sum_{i=1}^N \alpha_i [1 - \xi_i - y^i (w^\top x^i + b)] - \sum_{i=1}^N r_i \xi_i$$

- Let the partial derivative to be zero:

- $\frac{\partial L(w, b, \xi; \alpha, r)}{\partial w} = w - \sum_{i=1}^N \alpha_i y^i x^i = 0$

- $\frac{\partial L(w, b, \xi; \alpha, r)}{\partial b} = -\sum_{i=1}^N \alpha_i y^i = 0$

- $\frac{\partial L(w, b, \xi; \alpha, r)}{\partial \xi_i} = C - \alpha_i - r_i = 0$

Make  $\xi$  term disappear

- Then substitute them back to  $L$ :

- $L(w, b, \alpha)$

$$\begin{aligned} &= \frac{1}{2} \left\| \sum_{i=1}^N \alpha_i y^i x^i \right\|^2 + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i y^i \left( \sum_{j=1}^N \alpha_j y^j x^j \right)^\top x^i + b \sum_{i=1}^N \alpha_i y^i \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^{j^\top} x^i \end{aligned}$$

Same as before

# Dual problem

- $\max_{\alpha \geq 0, r \geq 0} \theta_{\mathcal{D}}(\alpha) = \max_{\alpha \geq 0, r \geq 0} \min_{w, b} L(w, b, \alpha)$
- Dual problem

$$\begin{aligned} \max_{\alpha, r} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^{\top} x^i \\ \text{s. t.} \quad & \alpha_i \geq 0, r_i \geq 0, \quad i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \\ & C - \alpha_i - r_i = 0, \quad i = 1, \dots, N \end{aligned}$$

# Dual problem (cont.)

- $\max_{\alpha} W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^{j\top} x^i$   
s.t.  $0 \leq \alpha_i \leq C, \quad i = 1, \dots, N$   
 $\sum_{i=1}^N \alpha_i y^i = 0$

Efficiently solved by SMO algorithm

Surprisingly, this is the only change


- When  $\alpha$  is solved,  $w$  and  $b$  can be solved



# Revisit the regularized objective

- $$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$
$$s. t. \quad y^i (w^\top x^i + b) \geq 1 - \xi_i, \quad i = 1, \dots, N$$
$$\xi_i \geq 0, \quad i = 1, \dots, N$$

- $$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max\{0, 1 - y^i (w^\top x^i + b)\}$$



SVM hinge loss

# SVM hinge loss vs. logistic loss

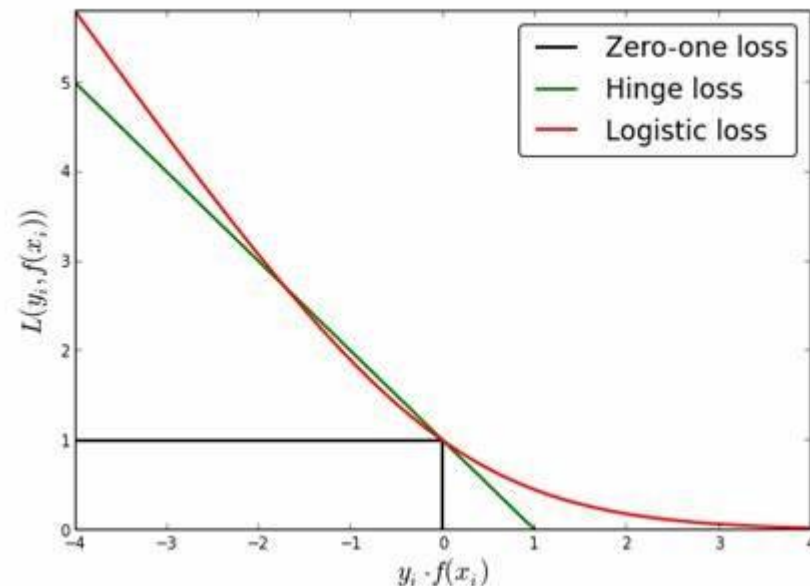
- SVM hinge loss

- $L(y, f(x)) = \max\{0, 1 - yf(x)\}$

- For  $y = 1$

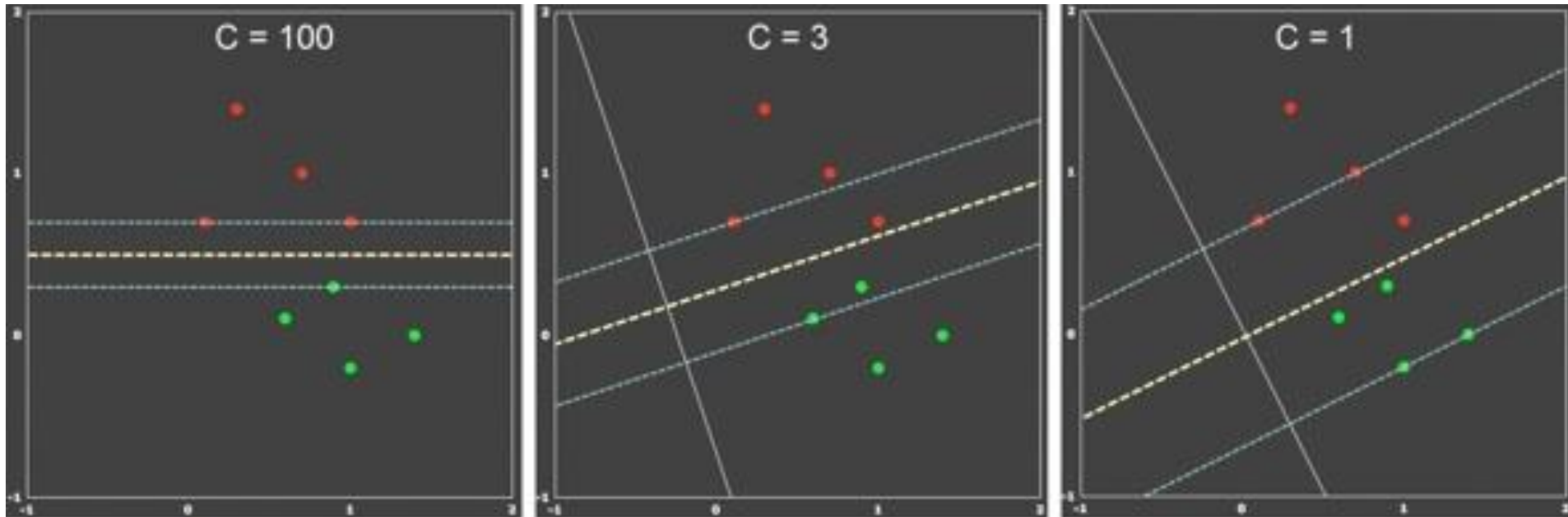
- Logistic loss

- $L(y, f(x)) = -y \log \sigma(f(x)) - (1 - y) \log (1 - \sigma(f(x)))$



# The effect of penalty coefficient

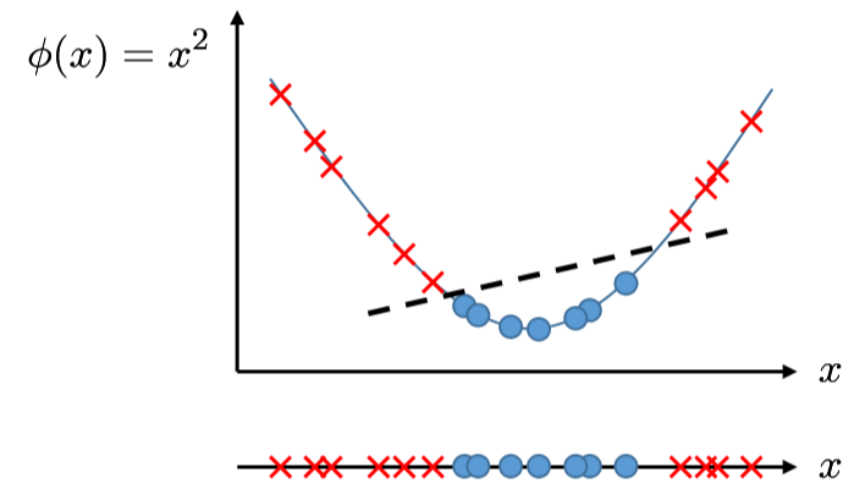
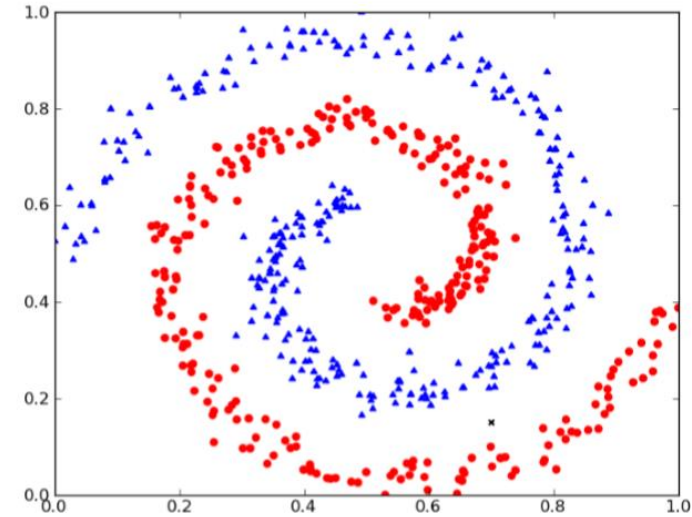
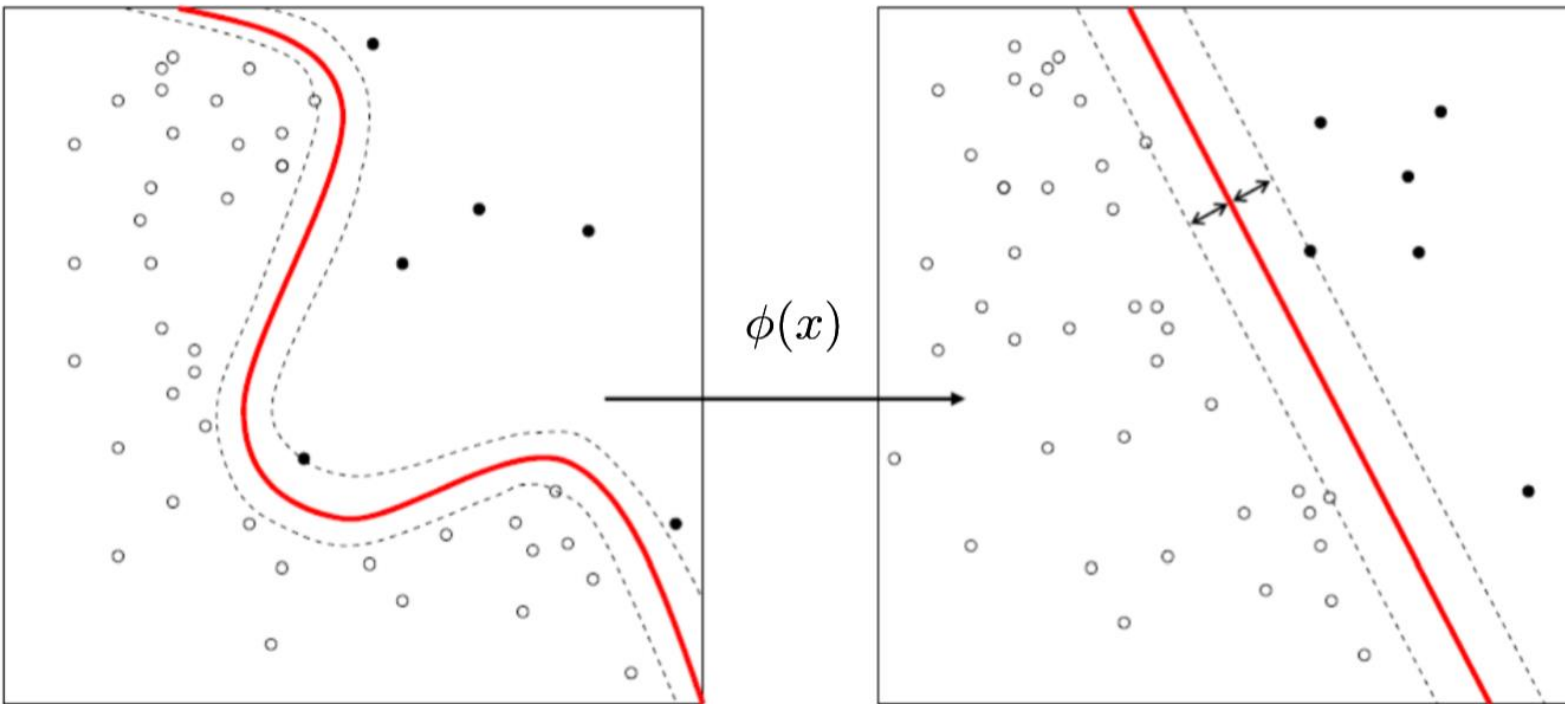
- $\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$
- Large  $C$  will result in narrow margin



# Kernels

# Non-linearly separable case

- Feature mapping



# From inner product to kernel function

- SVM

$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^\top x^i$$

- Kernel

$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j K(x^i, x^j)$$

- $K(x^i, x^j) = \langle \Phi(x^i), \Phi(x^j) \rangle$

- Kernel trick:

- For many cases, only  $K(x^i, x^j)$  are needed, so we can only define these  $K_{ij}$  without explicitly defining  $\Phi$

# Property

- If  $K$  is a valid kernel (that is, is defined by some feature mapping  $\Phi$ ), then the kernel matrix  $K = (K_{ij})_{ij} \in \mathbb{R}^{N \times N}$  is symmetric positive semi-definite
- Symmetric
  - $K_{ij} = K(x^i, x^j) = \langle \Phi(x^i), \Phi(x^j) \rangle = \langle \Phi(x^j), \Phi(x^i) \rangle = K(x^j, x^i) = K_{ji}$

- Positive semi-definite

$$\begin{aligned} z^T K z &= \sum_i \sum_j z_i K_{ij} z_j \\ &= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\ &= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \left( \sum_i z_i \phi_k(x^{(i)}) \right)^2 \\ &\geq 0. \end{aligned}$$

# Examples on kernels

- Gaussian kernel

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

- Radial basis function (RBF) kernel
  - What is the feature mapping  $\Phi$ ? (Hint: by using Taylor series)

- Simple polynomial kernel  $K(x, z) = (x^\top z)^d$

- Cosine similarity kernel  $K(x, z) = \frac{x^\top z}{\|x\| \cdot \|z\|}$

- Sigmoid kernel

$$K(x, z) = \tanh(\alpha x^\top z + c)$$

$$\tanh(b) = \frac{1 - e^{-2b}}{1 + e^{-2b}}$$



# Which kernel to select?

- RBF, polynomial, or others?
- For beginners, use RBF first
- Linear kernel: special case of RBF  
Accuracy of linear the **same** as RBF under certain parameters (Keerthi and Lin, 2003)
- Polynomial kernel:

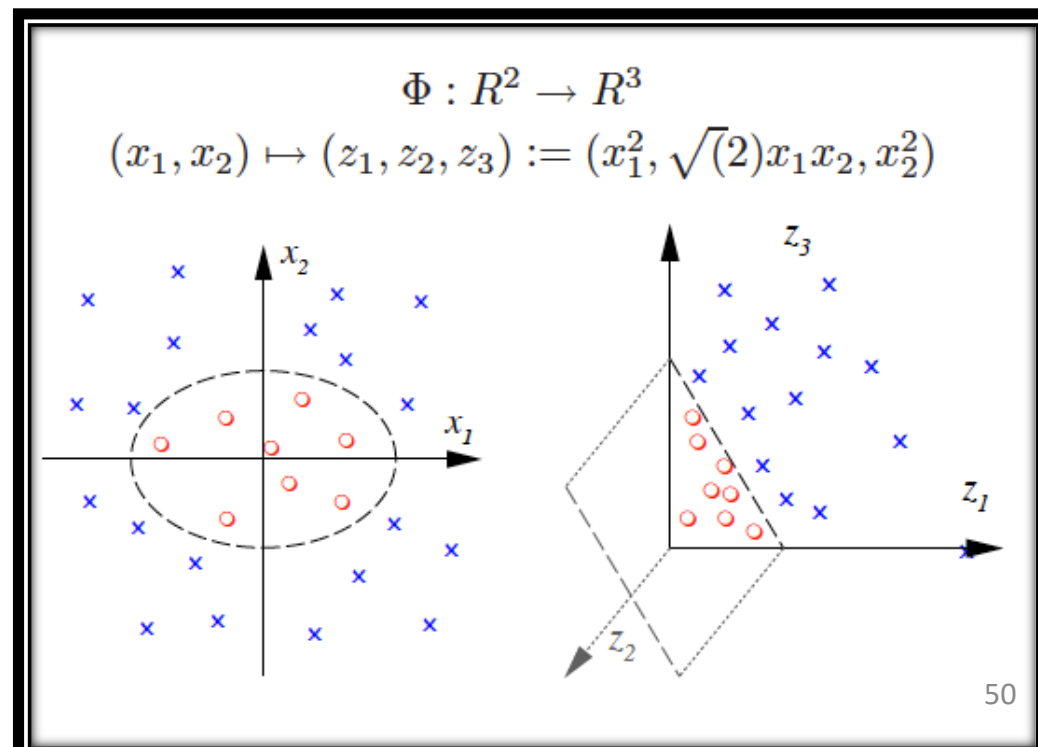
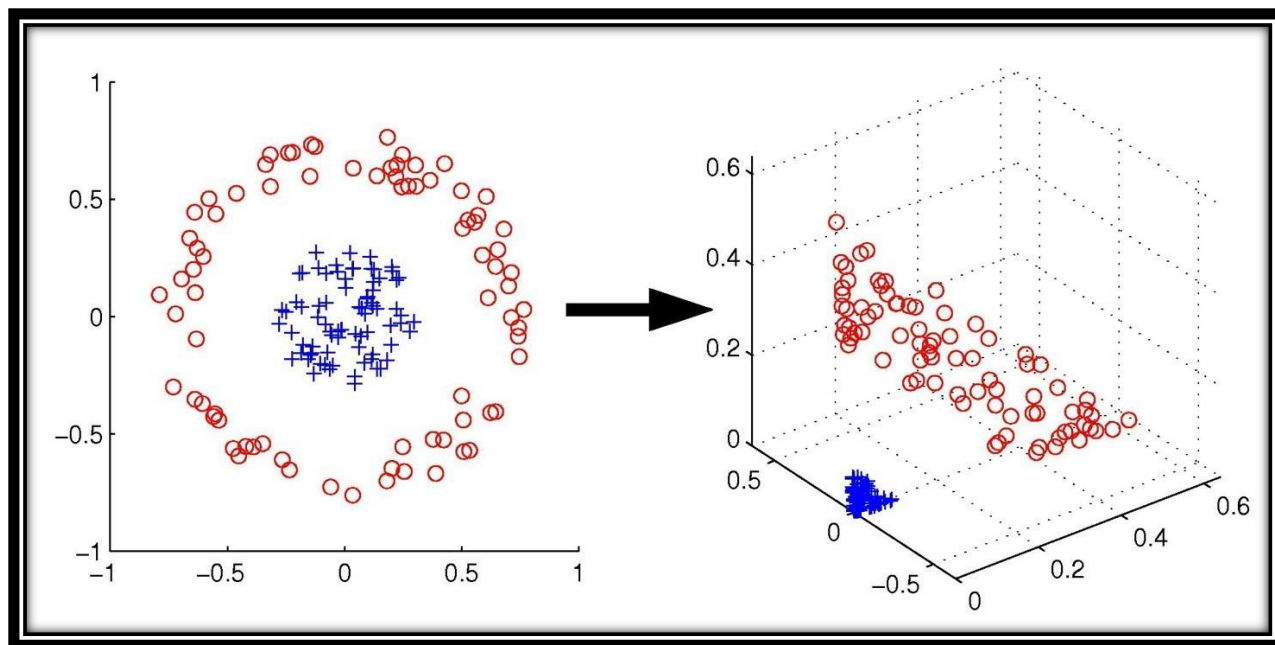
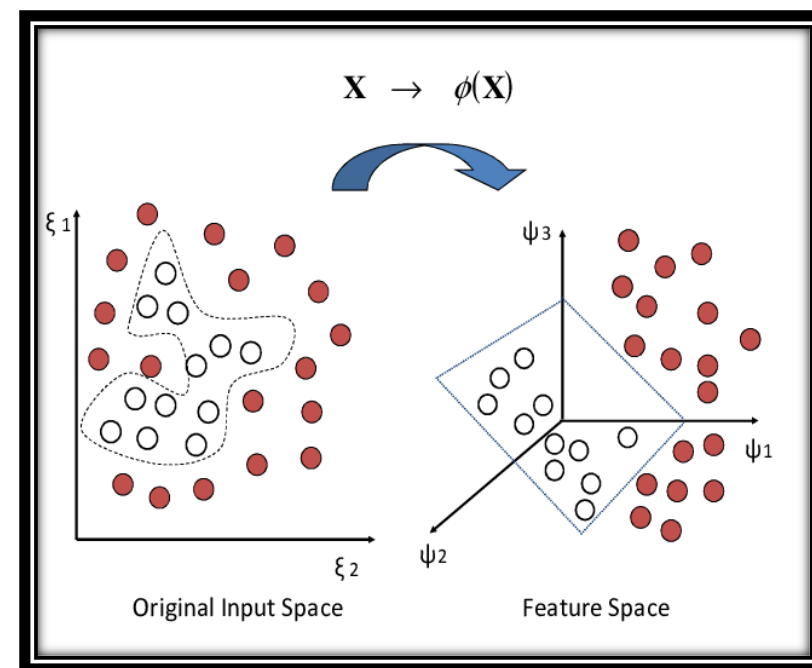
$$(\mathbf{x}_i^T \mathbf{x}_j / a + b)^d$$

Numerical difficulties:  $(< 1)^d \rightarrow 0, (> 1)^d \rightarrow \infty$

More parameters than RBF

- Commonly used kernels are Gaussian (RBF), polynomial, and linear
- But in different areas, special kernels have been developed. Examples
  1.  $\chi^2$  kernel is popular in computer vision
  2. String kernel is useful in some domains

# Examples



# Demo time

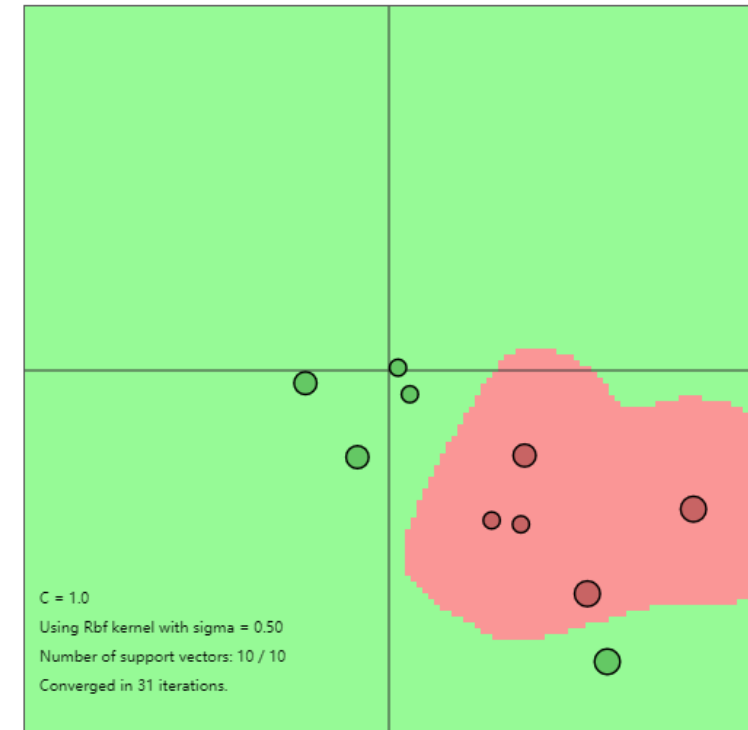
- Before we learn how to solve the optimization problem, let's have some relax and see the online demo of SVM

<https://cs.stanford.edu/~karpathy/svmjs/demo/>

## Support Vector Machine in Javascript

Uses SMO algorithm. Find code on [Github](#)  
Find me on Twitter [@karpathy](#)

**mouse click:** add red data point  
**shift + mouse click:** add green data point  
**'k':** toggle between Linear and Rbf kernel  
**'r':** reset



C = 1.0

RBF Kernel sigma = 1.0

# SMO Algorithm

# Solve $\alpha^*$

- Dual problem

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^\top x^i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \end{aligned}$$

- With  $\alpha^*$  solved,  $w$  and  $b$  are solved

# Coordinate Ascent (Descent)

- For the optimization problem

$$\max_{\alpha} W(\alpha_1, \alpha_2, \dots, \alpha_N)$$

- Coordinate ascent algorithm

Loop until convergence: {

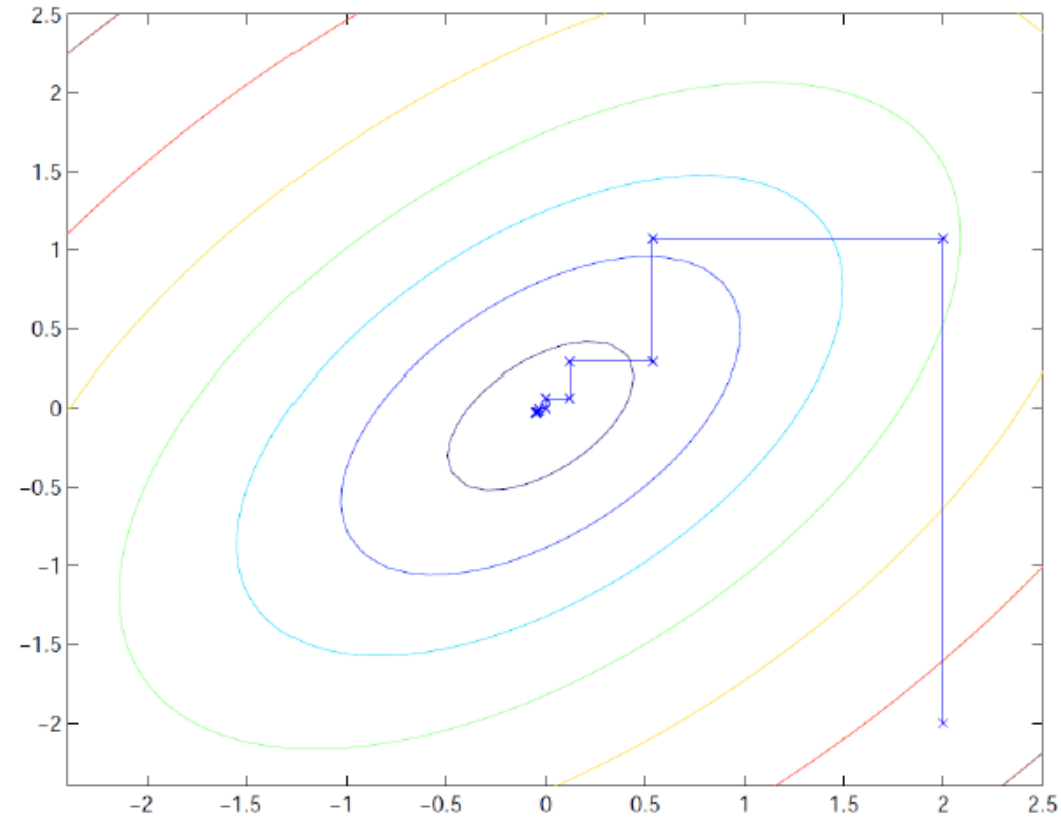
For  $i = 1, \dots, N$  {

$$\alpha_i := \arg \max_{\hat{\alpha}_i} W(\alpha_1, \dots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \dots, \alpha_N)$$

}

}

# Illustration



A two-dimensional coordinate ascent example

# Sequential minimal optimization (SMO)

- Recall the SVM optimization problem:

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^\top x^i \\ \text{s. t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \end{aligned}$$

- The coordinate ascent algorithm cannot be applied directly, because

$$\sum_{i=1}^N \alpha_i y^i = 0 \Rightarrow \alpha_i y^i = \sum_{j \neq i} \alpha_j y^j$$

- If we hold other  $\alpha_j$ , we can't make any changes to  $\alpha_i$



# Solution

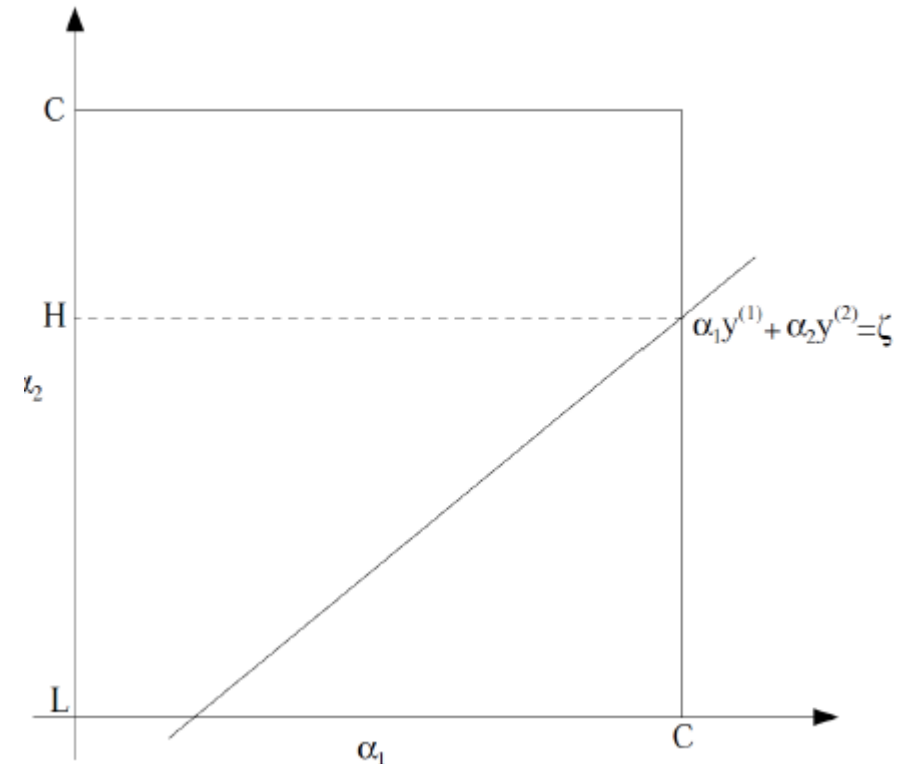
- Update **two variable** each time  
  Loop until convergence {
  1. Select some pair  $\alpha_i$  and  $\alpha_j$  to update next
  2. Re-optimize  $W(\alpha)$  w.r.t.  $\alpha_i$  and  $\alpha_j$}  
}
- Convergence test: whether the change of  $W(\alpha)$  is smaller than a predefined value (e.g. 0.01)
- Key advantage of SMO algorithm
  - The update of  $\alpha_i$  and  $\alpha_j$  (step 2) is efficient

# SMO (cont.)

- $$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^j{}^\top x^i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \end{aligned}$$

- Without loss of generality, hold  $\alpha_3 \dots \alpha_m$  and optimize  $w(\alpha)$  w.r.t  $\alpha_1$  and  $\alpha_2$

$$\begin{aligned} \alpha_1 y^1 + \alpha_2 y^2 &= - \sum_{i=3}^N \alpha_i y^i = \zeta \\ \Rightarrow \alpha_1 &= y^1 (\zeta - \alpha_2 y^2) \end{aligned}$$



## SMO (cont.)

- With  $\alpha_1 = (\zeta - \alpha_2 y^2) y^1$ , the objective is written as  
$$W(\alpha_1, \alpha_2, \dots, \alpha_N) = W((\zeta - \alpha_2 y^2) y^1, \alpha_2, \dots, \alpha_N)$$

- Thus the original optimization problem

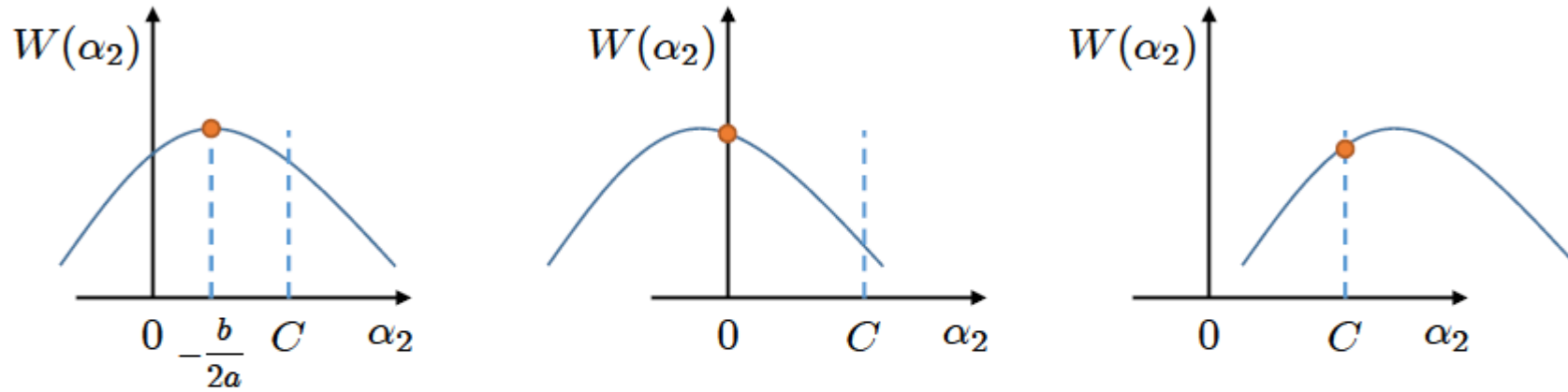
$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^i y^j x^{j\top} x^i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i y^i = 0 \end{aligned}$$

is transformed into a quadratic optimization problem w.r.t  $\alpha_2$

$$\begin{aligned} \max_{\alpha_2} \quad & W(\alpha_2) = a\alpha_2^2 + b\alpha_2 + c \\ \text{s.t.} \quad & 0 \leq \alpha_2 \leq C \end{aligned}$$

## SMO (cont.)

- Optimizing a quadratic function is much efficient



$$\begin{aligned} \max_{\alpha_2} \quad & W(\alpha_2) = a\alpha_2^2 + b\alpha_2 + c \\ \text{s.t.} \quad & 0 \leq \alpha_2 \leq C \end{aligned}$$

# Pros and cons of SVM

- Advantages:

- The solution, which is based on convex optimization, is globally optimal
- Can be applied to both linear/non-linear classification problems
- Can be applied to high-dimensional data
  - since the complexity of the data set mainly depends on the support vectors
- Complete theoretical guarantee
  - Compared with deep learning

- Disadvantages:

- The number of parameters  $\alpha$  is number of samples, thus hard to apply to large-scale problems
  - SMO can ease the problem a bit
- Mainly applies to binary classification problems
  - For multi-classification problems, can solve several binary classification problems, but might face the problem of imbalanced data