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

Last lecture

- Discriminative / Generative Models
- Logistic regression (binary classification)
  - Cross entropy
  - Formulation, sigmoid function
  - Training—gradient descent
- More measures for binary classification (AUC, AUPR)
- Class imbalance
- Multi-class logistic regression

Today’s lecture

- 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

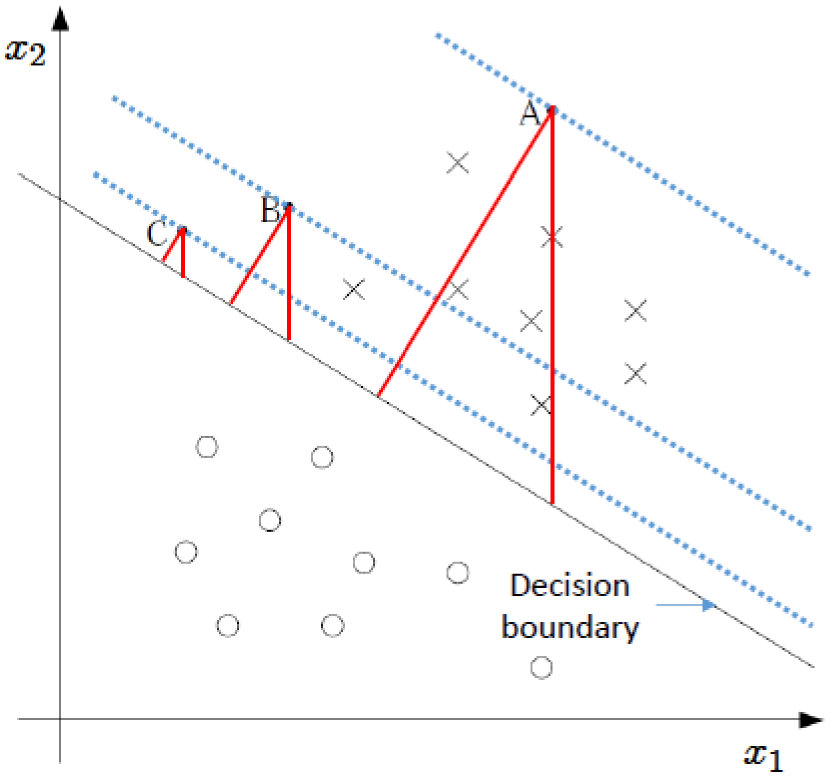
- Let $s(x) = \theta_0 + \theta_1 x_2 + \theta_2 x_2$, so the probability in logistic regression is defined as $p_\theta(y = 1|x) = \frac{1}{1+e^{-s(x)}}$

- 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 $\theta_0 + \theta_1 x_2 + \theta_2 x_2 = 0$

Illustration of logistic regression

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



$$s(x^A) = \theta_0 + \theta_1 x_1^A + \theta_2 x_2^A = 7$$

$$s(x^B) = \theta_0 + \theta_1 x_1^B + \theta_2 x_2^B = 3$$

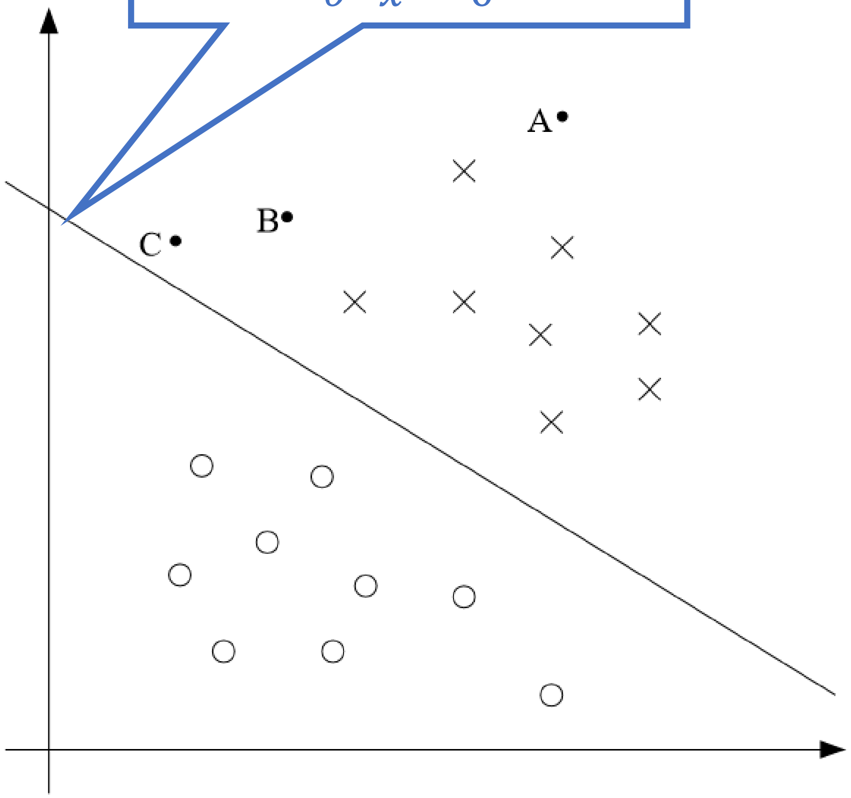
$$s(x^C) = \theta_0 + \theta_1 x_1^C + \theta_2 x_2^C = 1$$

$$s(x) = 0$$

Intuition

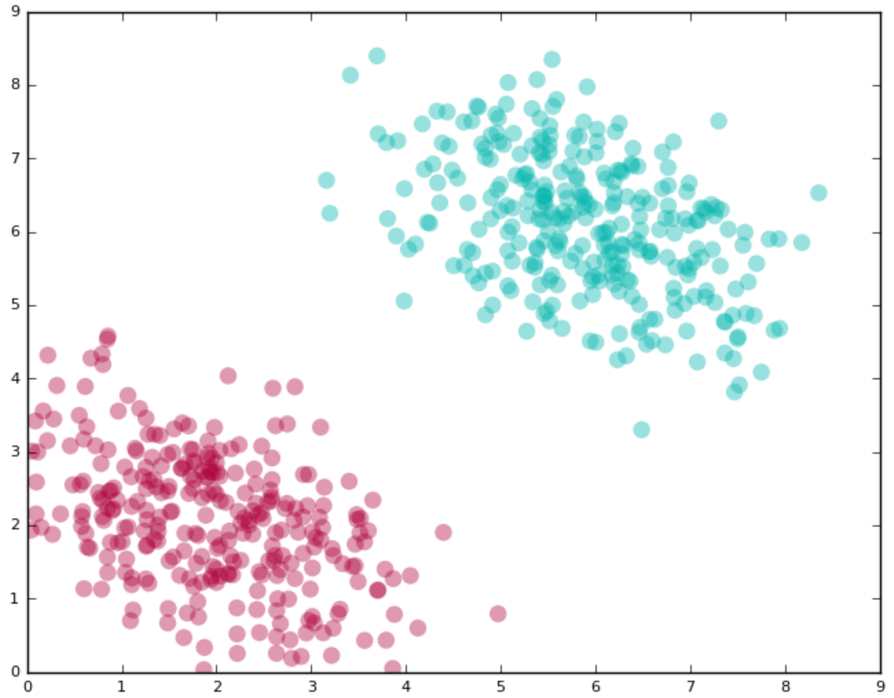
- Positive when
  $$p_\theta(y = 1|x) = h_\theta(x) = \sigma(\theta^\top x) \geq 0.5$$
  or
  $$\theta^\top x \geq 0$$
- Point A
  - Far from decision boundary
  - More confident to predict the label 1
- Point C
  - Near decision boundary
  - A small change to the decision boundary could cause prediction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

Decision boundary / Separating hyperplane $\theta^\top x = 0$



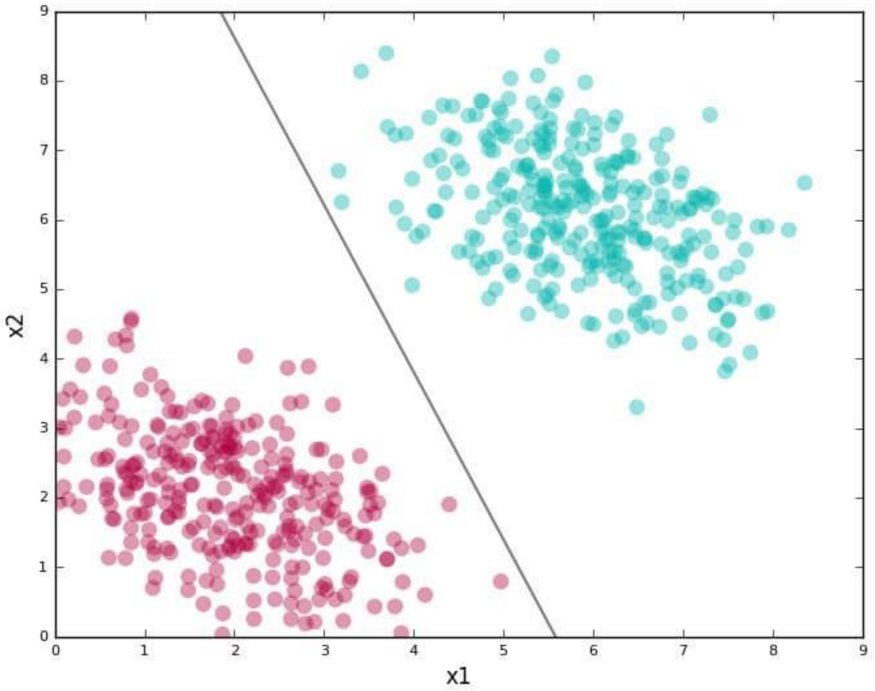
Example

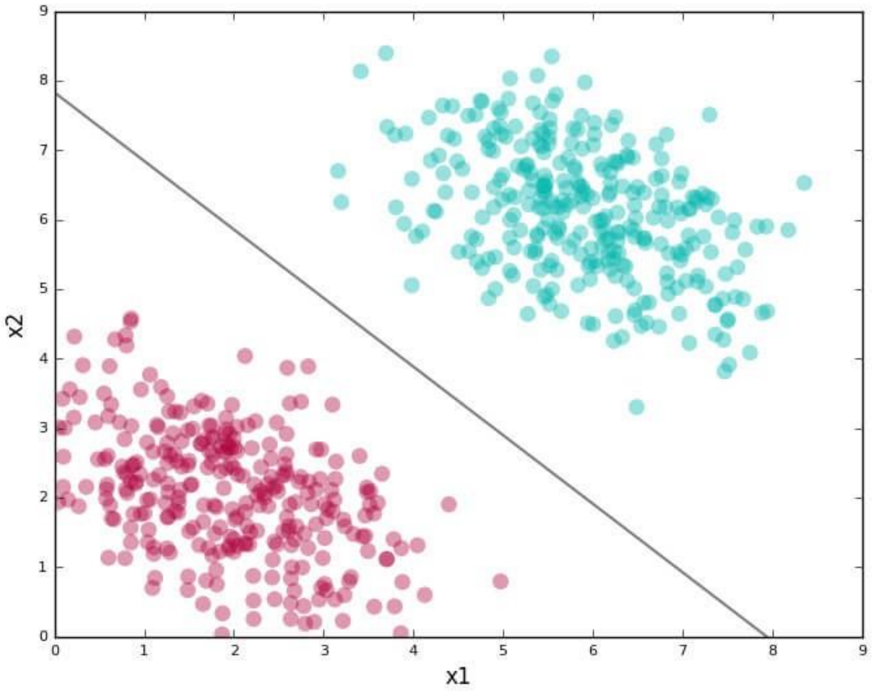
- Given a dataset of two classes, how to find a line to separate them?



Example (cont.)

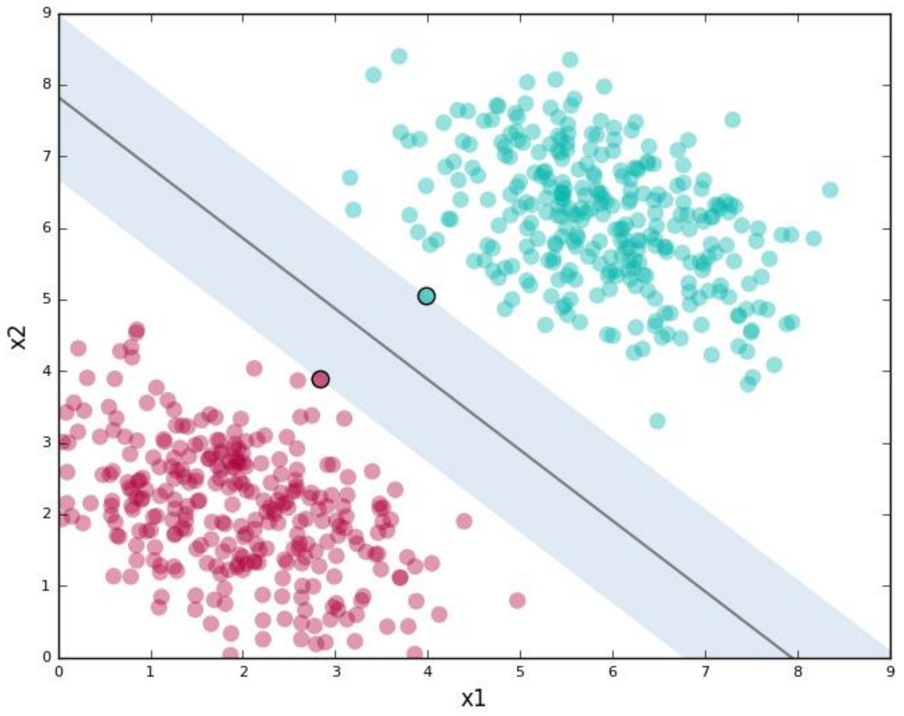
- Both the two solutions can separate the data perfectly, but we prefer the one on the right, why?

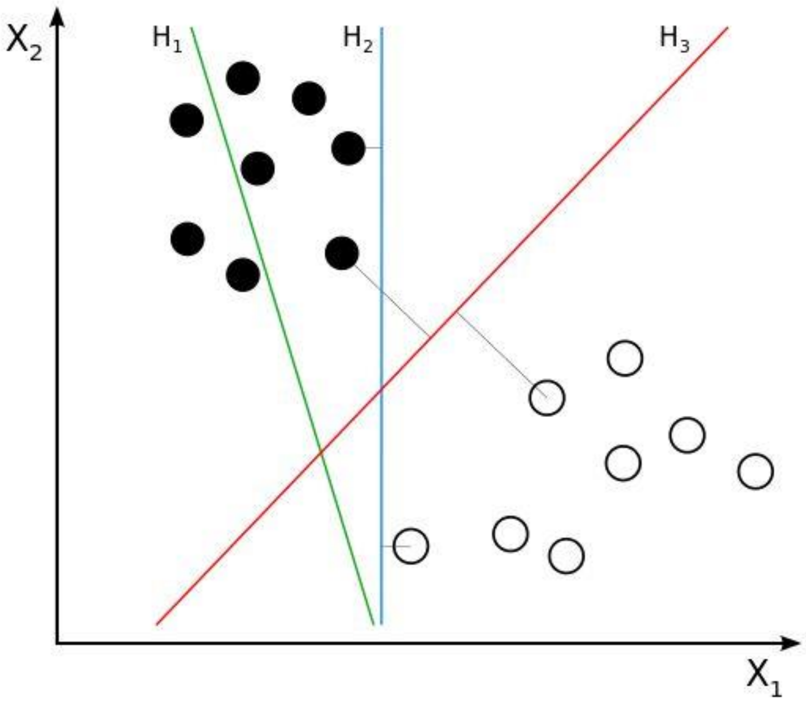




Example (cont.)

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



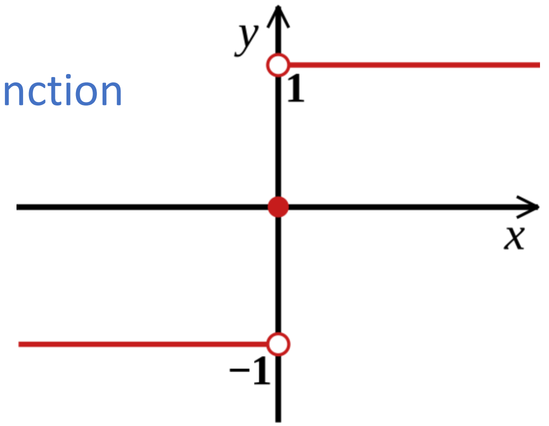


- These are the support vectors, and the model is called support vector machine.

Notations for SVM

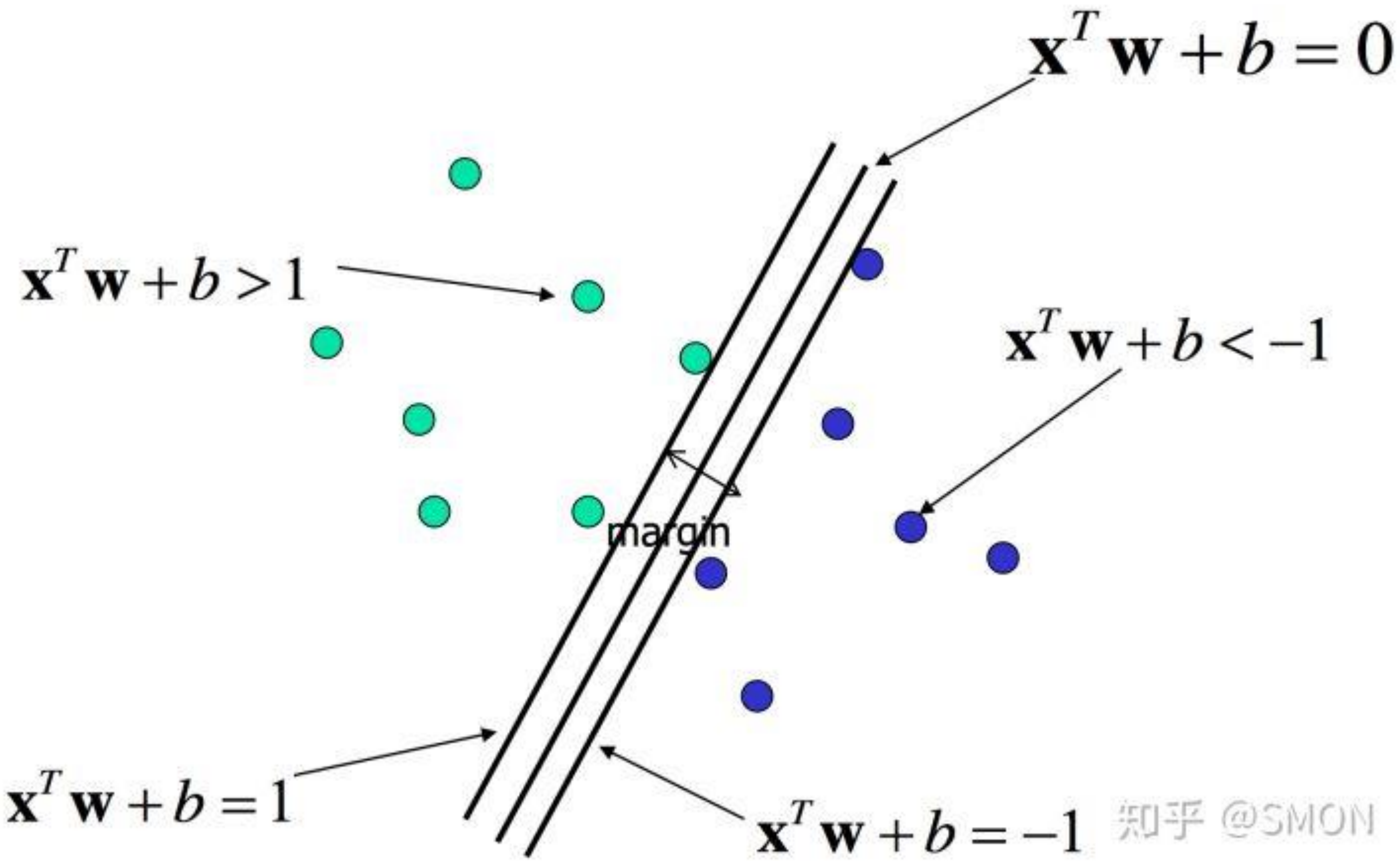
- Feature vector x
- Class label $y \in \{-1, 1\}$
  - Instead of $\{0,1\}$
- Parameters
  - Intercept b
    - We also drop the convention we had previously of letting $x_0 = 1$ be an extra coordinate in the input feature vector
  - Feature weight vector w
- Label prediction
  - $h_{w,b}(x) = g(w^\top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

Sign function



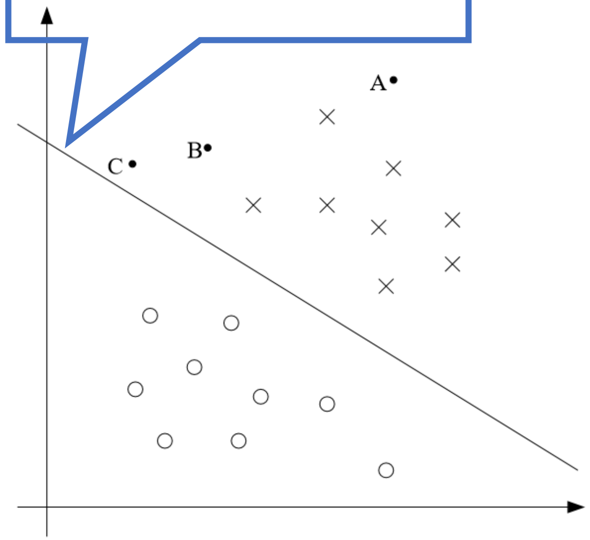
Hyperplane and margin

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



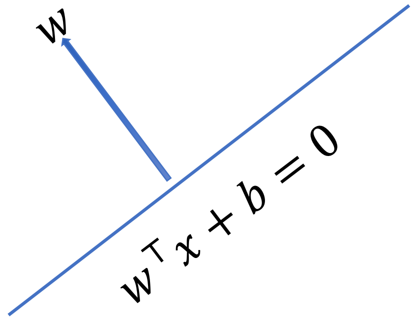
Functional margin

Decision boundary
$\theta^\top x = w^T x + b = 0$



- 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 $w^\top x + b = 0$

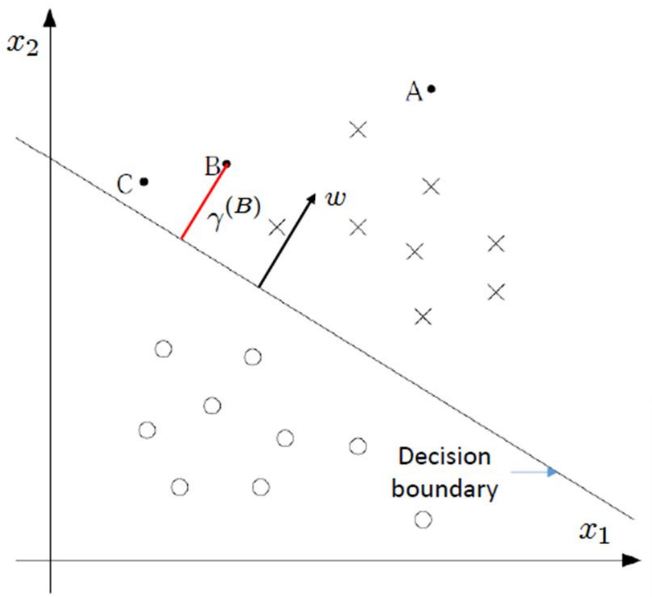
Geometric margin



- 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
  - $w^\top \left(x - \gamma \frac{w}{\|w\|}\right) + b = 0$
  - Solve it, get

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

  - In general, $\gamma = y(w^\top x + b)$ 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

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

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

$$\|w\| = 1$$

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

which is equivalent to

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

Non-convex objective

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

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

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

$$\max_{w,b} \frac{1}{\|w\|}$$
$$s.t. \quad y^i(w^\top x^i + b) \geq 1, \qquad i = 1, \dots, N$$

or equivalently

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

Can be efficiently solved by quadratic programming (QP)

Lagrange Duality

Lagrangian for convex optimization

- Given a convex optimization problem

$$\min_{w} \quad f(w)$$
$$\text{s.t.} \quad h_i(w) = 0, \quad i = 1, \ldots, l$$

- The Lagranigan of this problem is defined as

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

Lagrangian multipliers

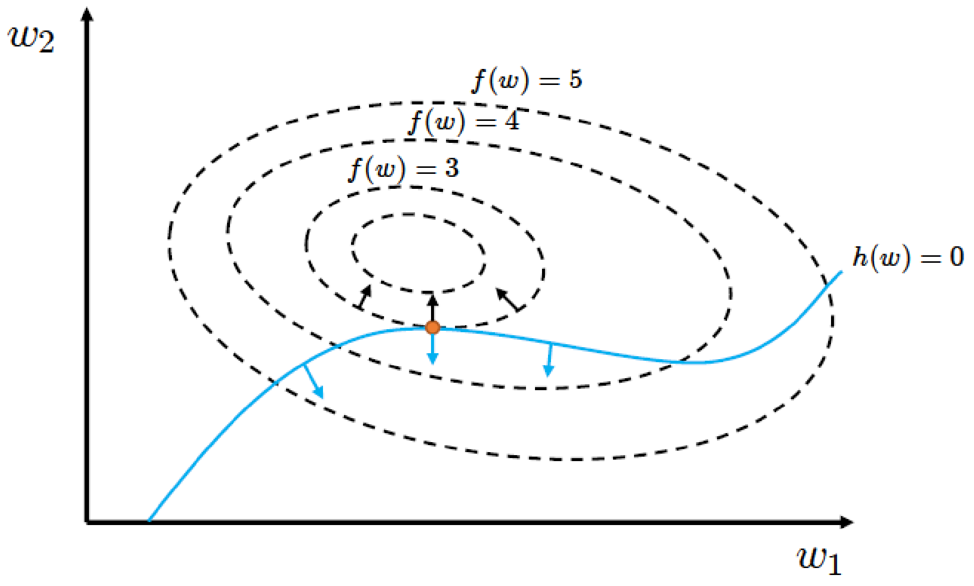
- Solving

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

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?

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

- 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

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

- 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

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

  - 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

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

- Consider quantity

primal

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

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

- 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

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

- 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

- 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^*, α^*, β^*
  - w^* is the solution of the primal problem
  - α^*, β^* are the solution of the dual problem
  - And the values of the two problems are equal

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

- w^*, α^*, β^* satisfy the KKT conditions:

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

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

KKT dual complementarity condition

$$\alpha_i^* g_i(w^*) = 0, \; i = 1, \ldots, k$$

$$g_i(w^*) \leq 0, \; i = 1, \ldots, k$$

$$\alpha^* \geq 0, \; i = 1, \ldots, k$$

  - 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

$$\min_{w,b} \ \frac{1}{2}\|w\|^2$$
$$s.t. \quad y^i\left(w^\top x^i + b\right) \geq 1, \qquad i = 1, \dots, N$$

- Rewrite the constraints as

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

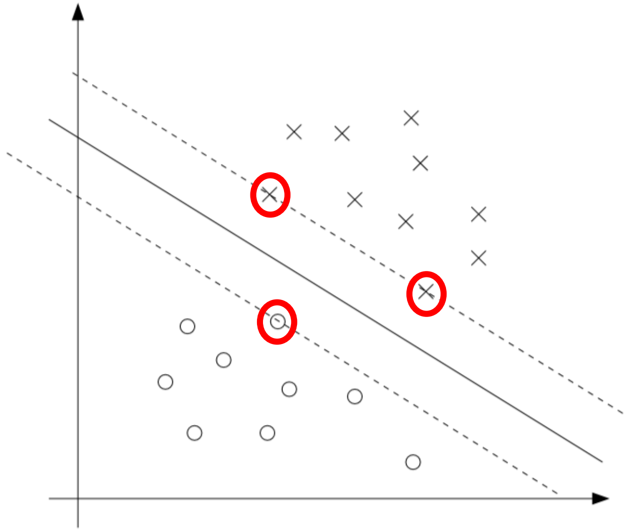
- Note that from the KKT dual complementarity condition, $\alpha_i > 0$ is only possible for training samples with $g_i(w) = 0$

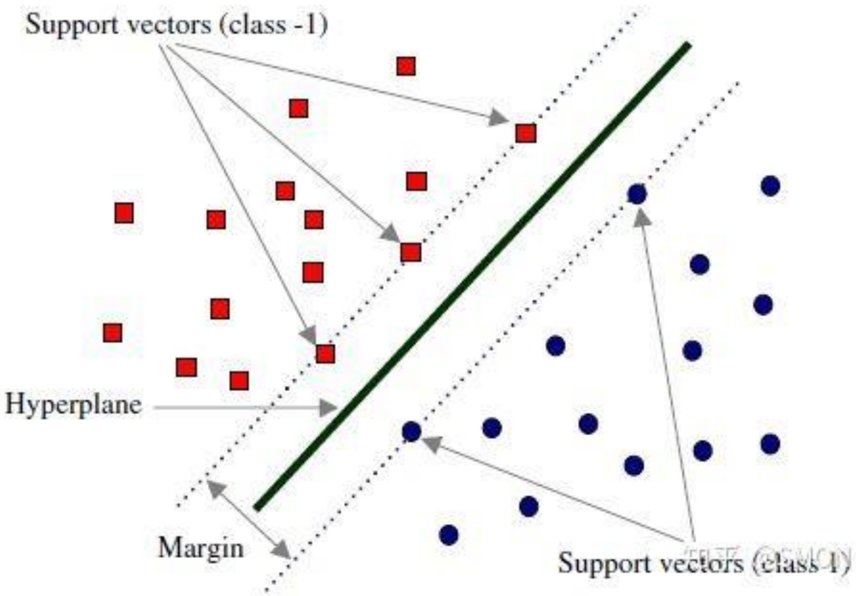
Support vectors

- 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:

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

- Lagrangian

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

Solving it

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

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

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

Dual problem

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

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

$$s.t. \quad \alpha_i \geq 0, i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0$$

- Then solve α^* by SMO

Can check the KKT conditions hold

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

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

KKT dual complementarity condition $\longrightarrow$ $\alpha_i^* g_i(w^*) = 0, \quad i = 1, \dots, k$

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

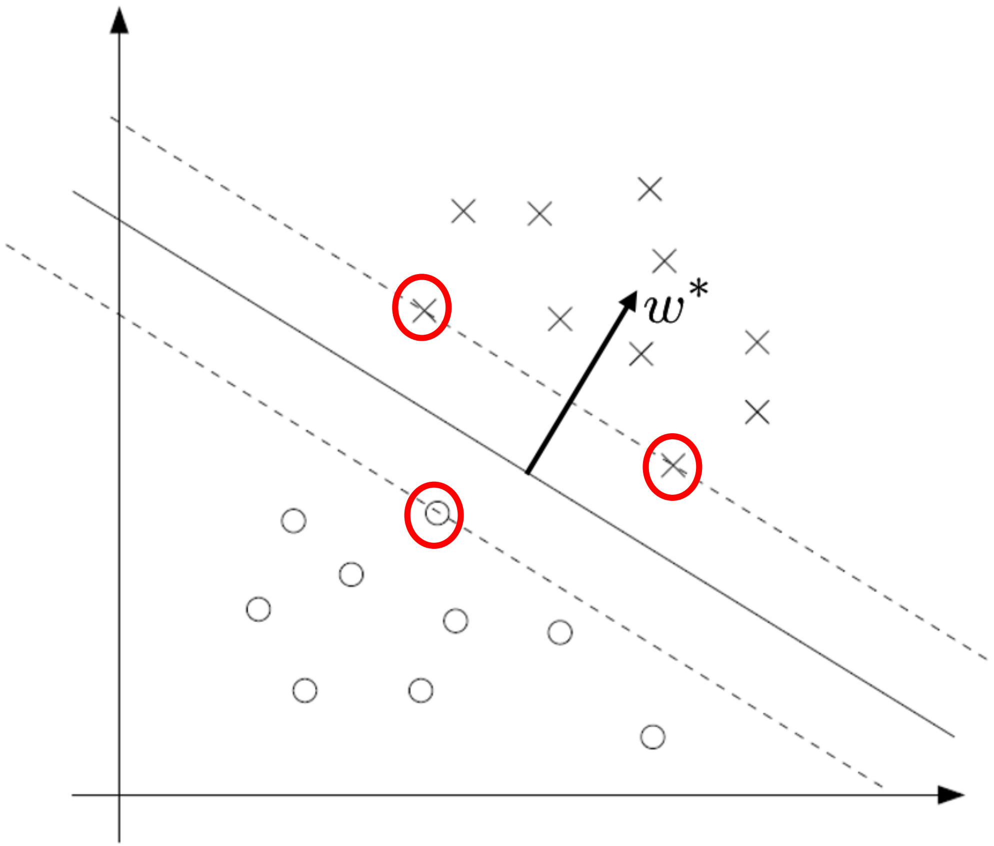
$$\alpha^* \geq 0, \quad i = 1, \dots, k$$

Solve w^* and b^*

- With α^*

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

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

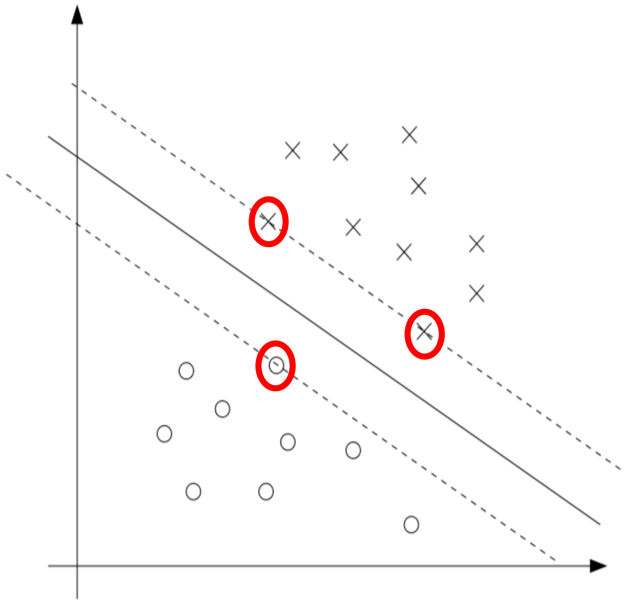


- Then

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

Check it!

Predicting values



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

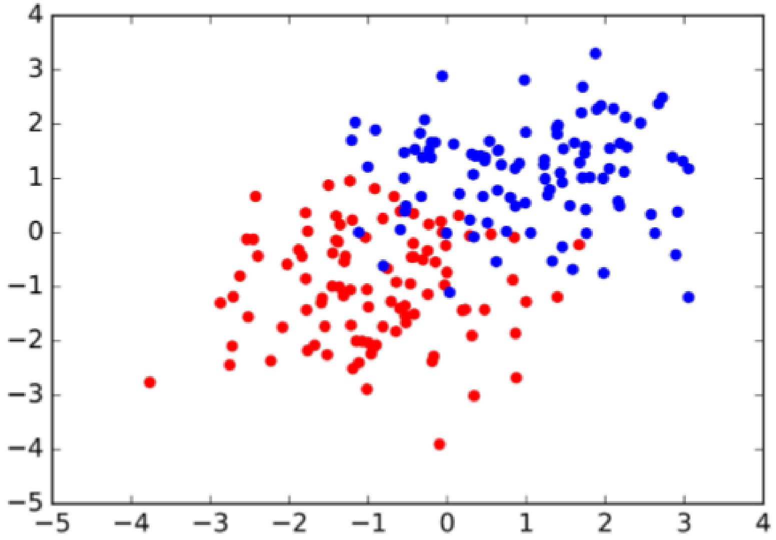
- Only need to calculate the inner product of x with support vectors

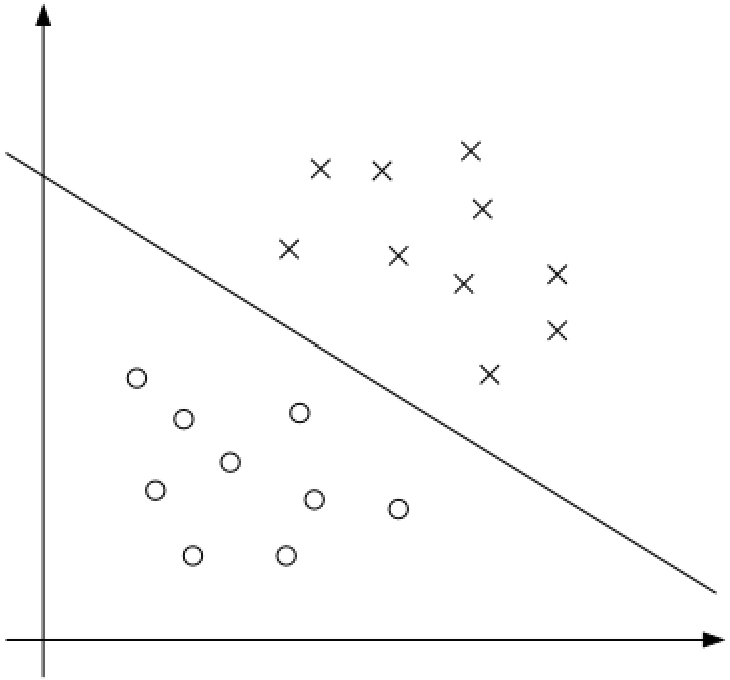
- Prediction is
$$y = \mathrm{Sign}(w^\top x + b)$$

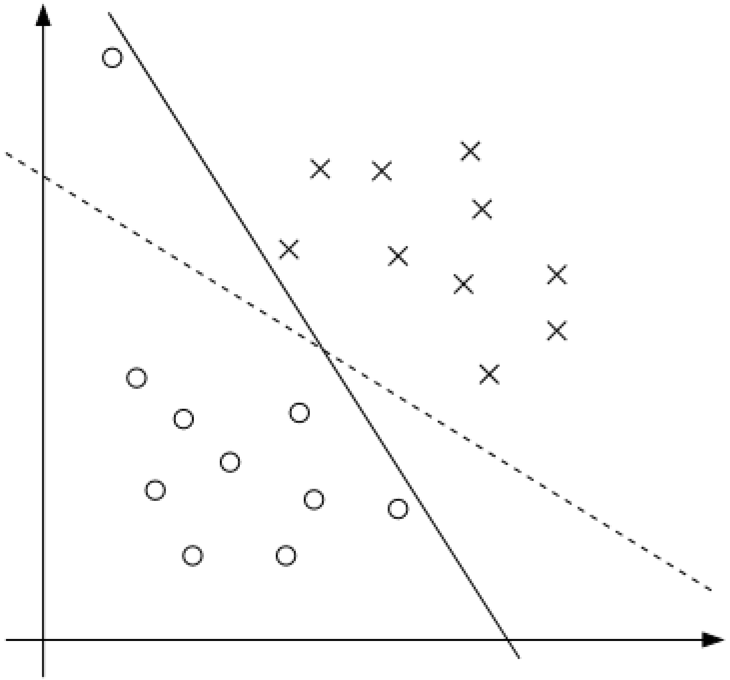
Regularization and the Non-Separable Case

Motivation

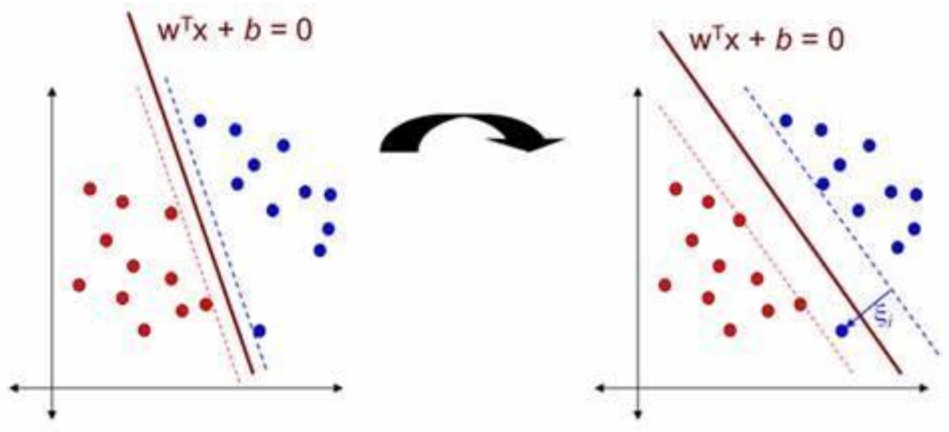
- SVM assumes data is linearly separable
  - But some data is linearly non-separable
  - SVM is susceptible to outliers







Solution – Soft margin

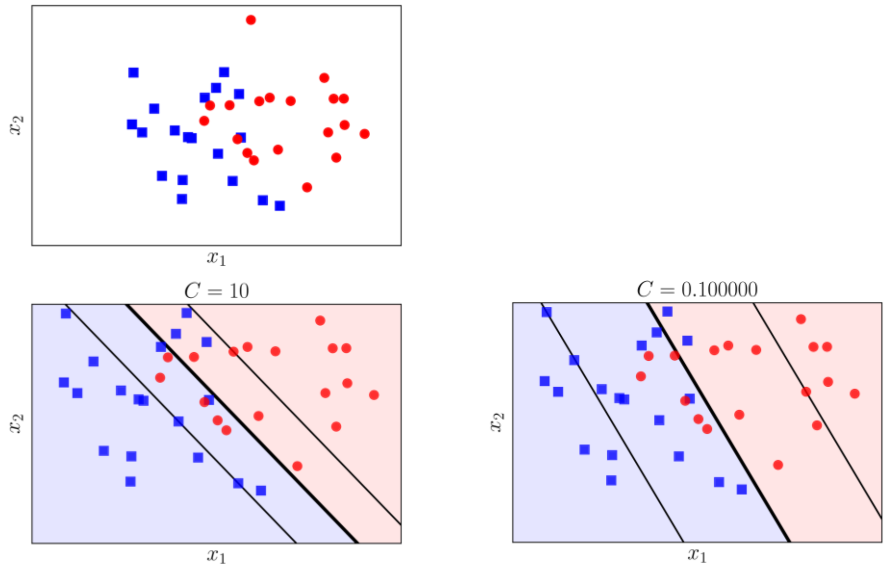


- To make the algorithm work for non-linearly separable datasets as well as be less sensitive to outliers
- Add slack variables

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

L^1 regularization

$$s.t. \quad y^i(w^\top x^i + b) \geq 1 - \xi_i, \qquad i = 1, \dots, N$$
$$\xi_i \geq 0, \qquad i = 1, \dots, N$$



Example

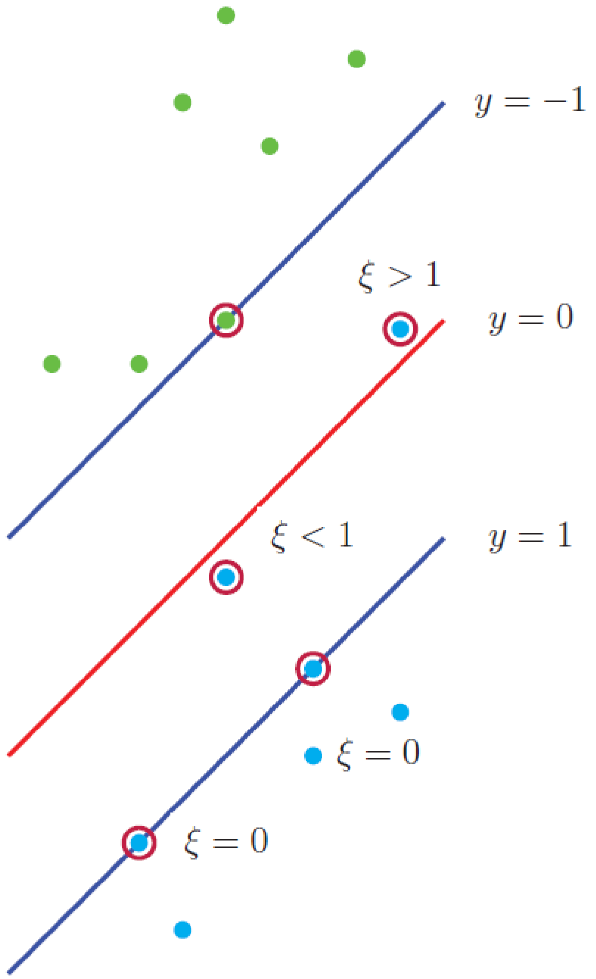
- Correctly classified points beyond the support line with $\xi = 0$

- 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 $\xi > 2$



Lagrangian

- Lagrangian

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

- 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 ξ term disappear

- Then substitute them back to L:
  - $L(w,b,\alpha)$

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

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

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

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

$$s.t. \quad \alpha_i \geq 0, r_i \geq 0, \qquad i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0$$

$$C - \alpha_i - r_i = 0, \qquad i = 1, \dots, N$$

Dual problem (cont.)

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

Surprisingly, this is the only change

Efficiently solved by SMO algorithm

- When α is solved, w and b can be solved

Revisit the regularized objective

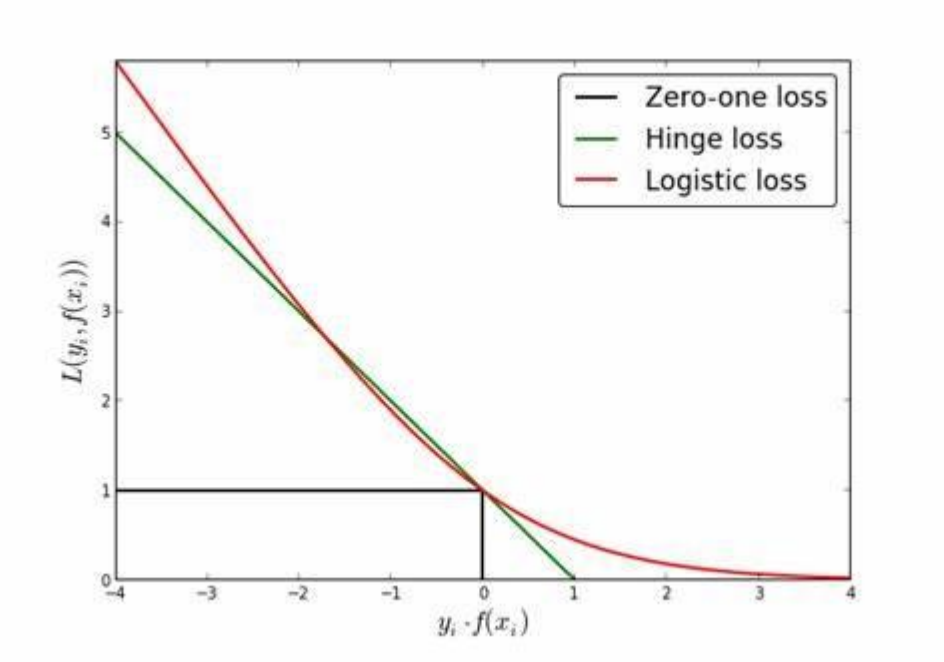
- $$\min_{w,b} \; \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$
  $$\begin{aligned} s.t. \quad & y^i\left(w^\top x^i + b\right) \geq 1 - \xi_i, \qquad i = 1, \dots, N \\ & \xi_i \geq 0, \qquad i = 1, \dots, N \end{aligned}$$

- $$\min_{w,b} \; \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \underbrace{\max\{0, 1 - y^i\left(w^\top x^i + b\right)\}}_{\text{SVM hinge loss}}$$

SVM hinge loss vs. logistic loss

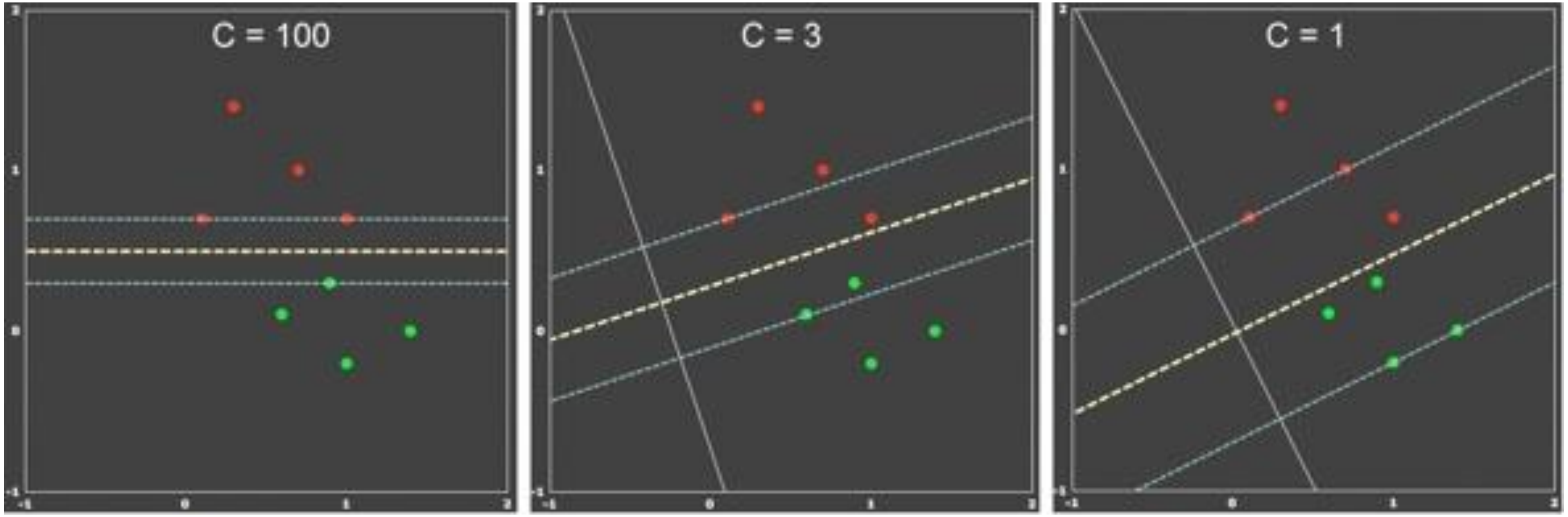
- SVM hinge loss
  - $L\big(y, f(x)\big) = \max\{0, 1 - yf(x)\}$
- For $y = 1$

- Logistic loss
  - $L\big(y, f(x)\big) = -y\ \log \sigma\big(f(x)\big) - (1 - \mathrm{y}) \log\Big(1 - \sigma\big(f(x)\big)\Big)$



The effect of penalty coefficient

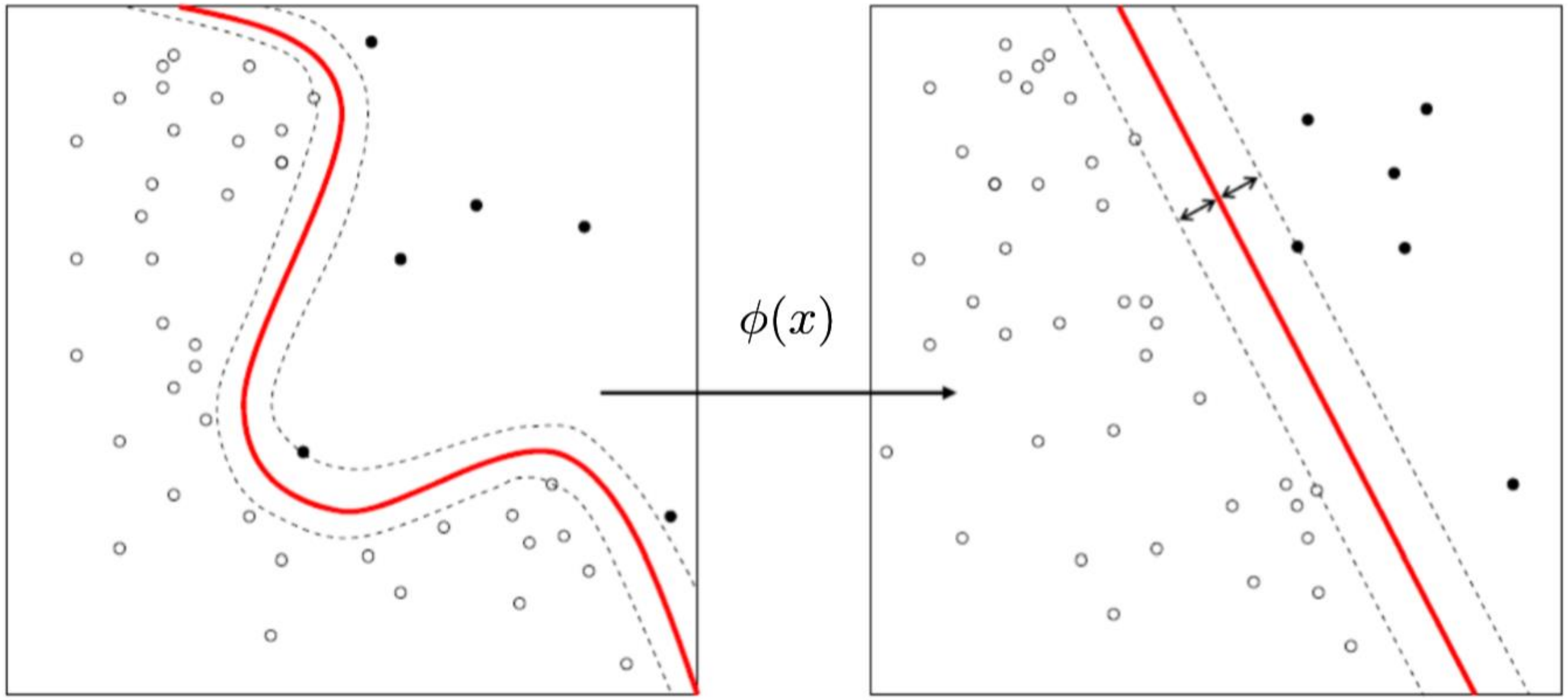
- $\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$
- Large C will result in narrow margin

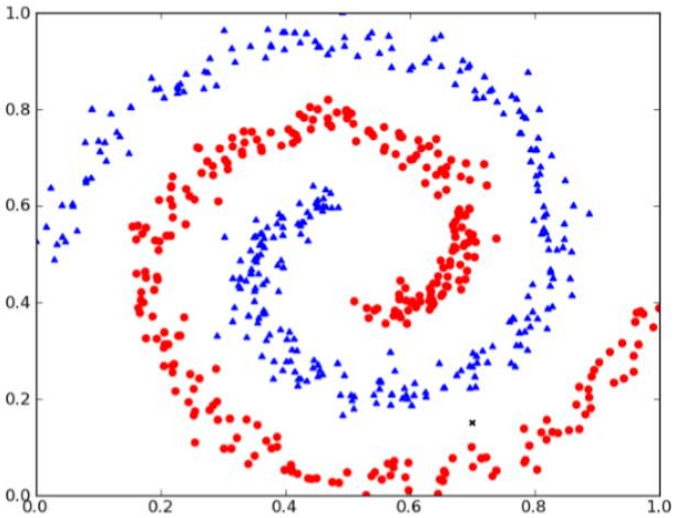


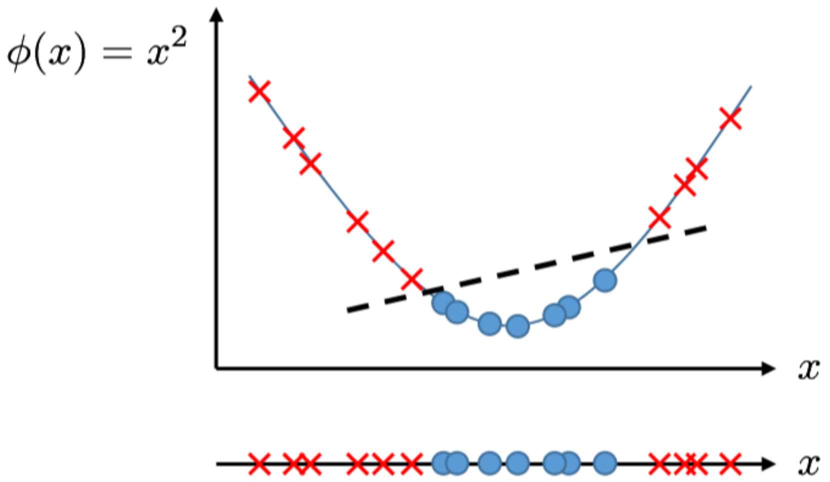
Kernels

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

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

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

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

Property

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

- 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 Φ? (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:

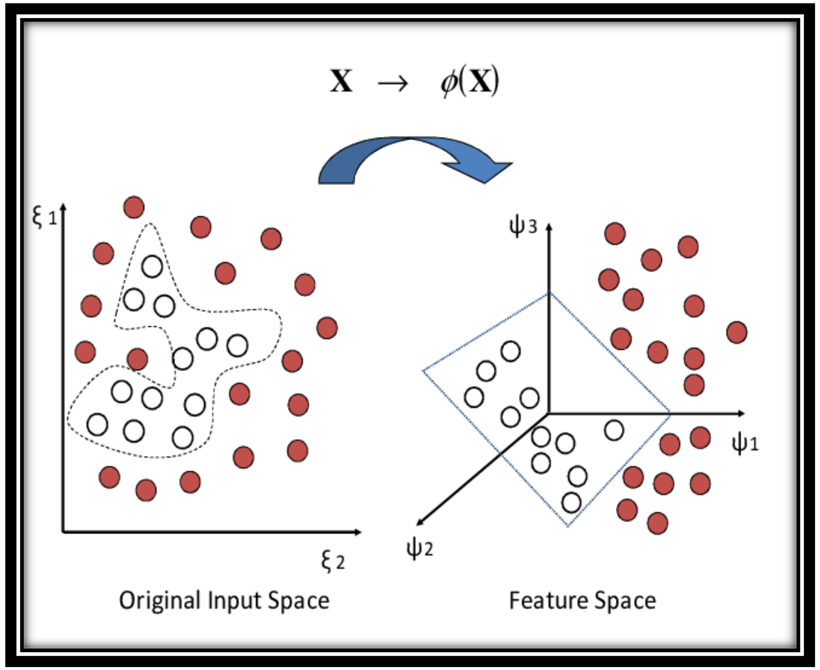
$$(\mathbf{x}_i^T \mathbf{x}_j / a + b)^d$$

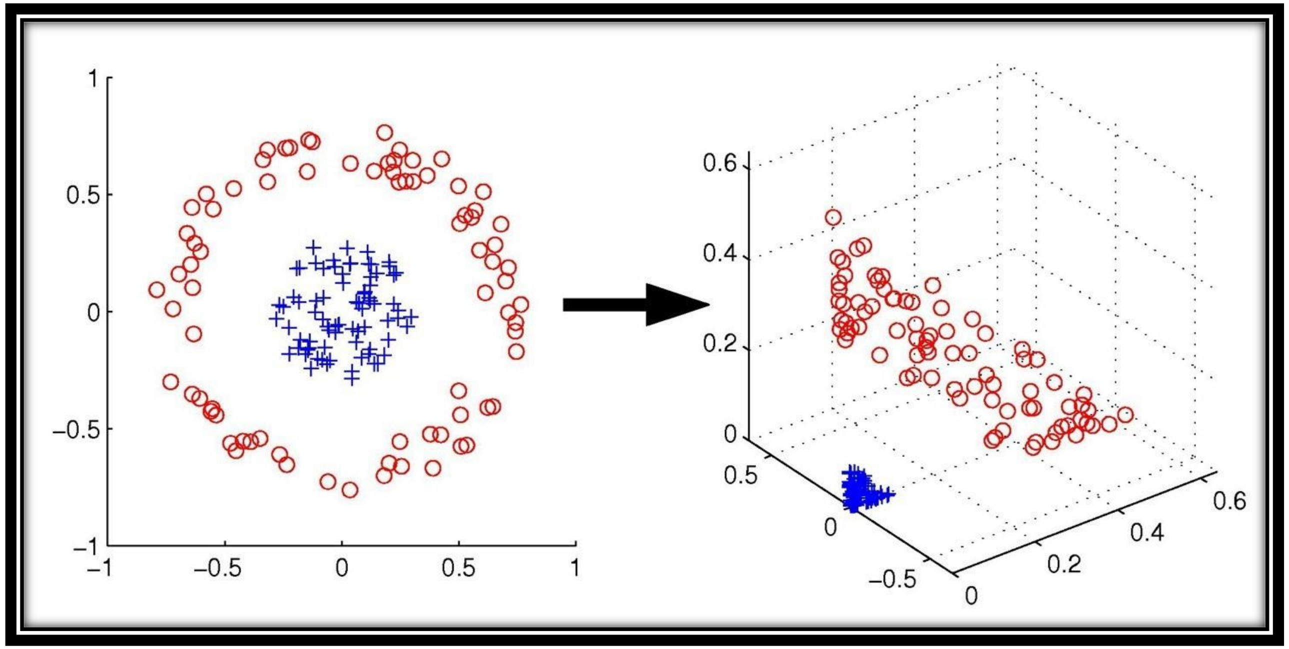
  Numerical difficulties: $(< 1)^d \to 0, (> 1)^d \to \infty$

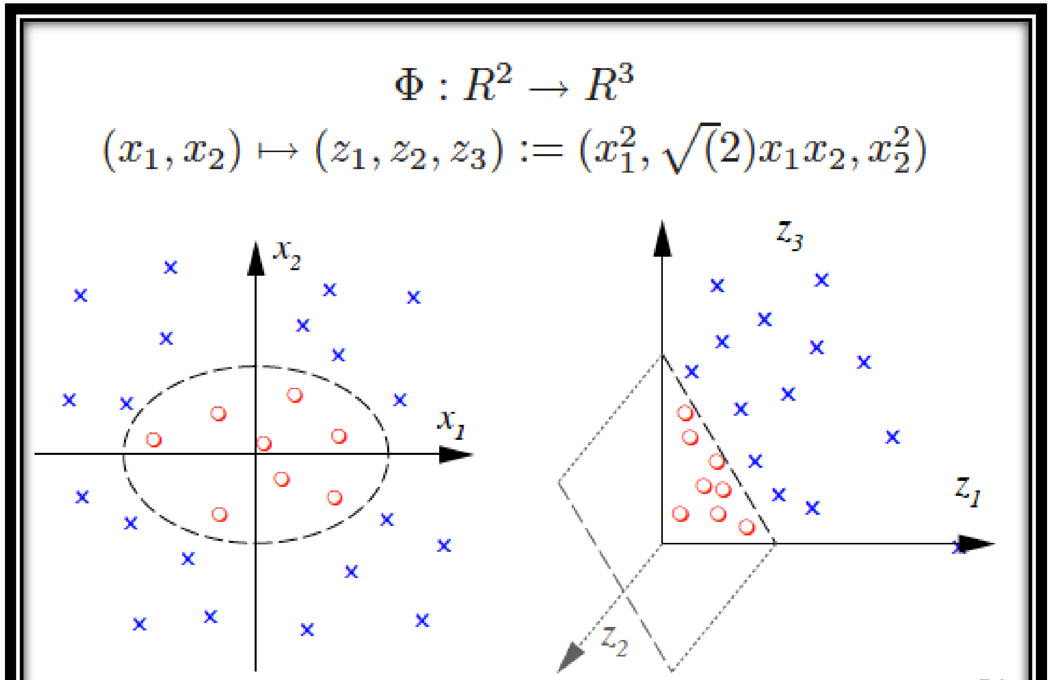
  More parameters than RBF

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

Examples

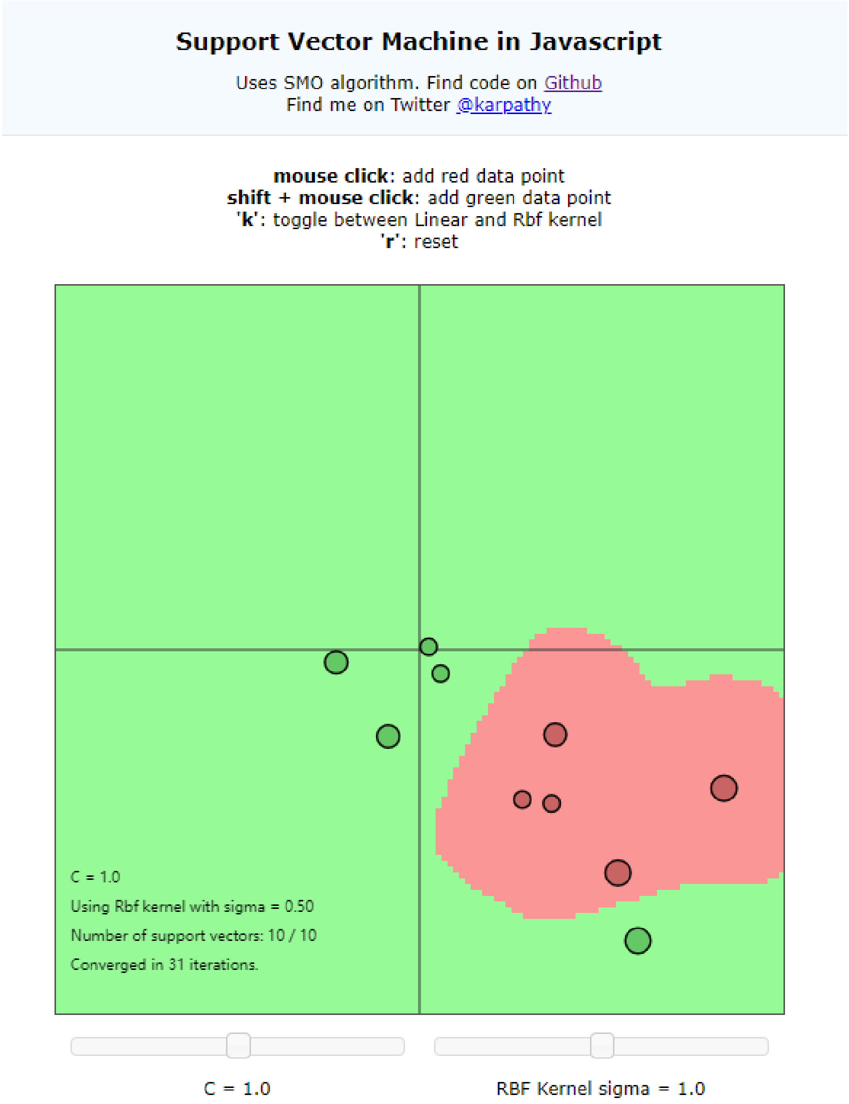

$\mathbf{X} \rightarrow \phi(\mathbf{X})$
ξ 1
ξ 2
ψ3
ψ1
ψ2
Original Input Space
Feature Space


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1
0.6
0.4
0.2
0
0.5
0
−0.5
0
0.2
0.4
0.6


$\Phi : R^2 \rightarrow R^3$
$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{(2)}x_1x_2, x_2^2)$
x_2
x_1
z_3
z_1
z_2

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/



SMO Algorithm

Solve α^*

- Dual problem

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

- With α^* solved, w and b are solved

Coordinate Ascent (Descent)

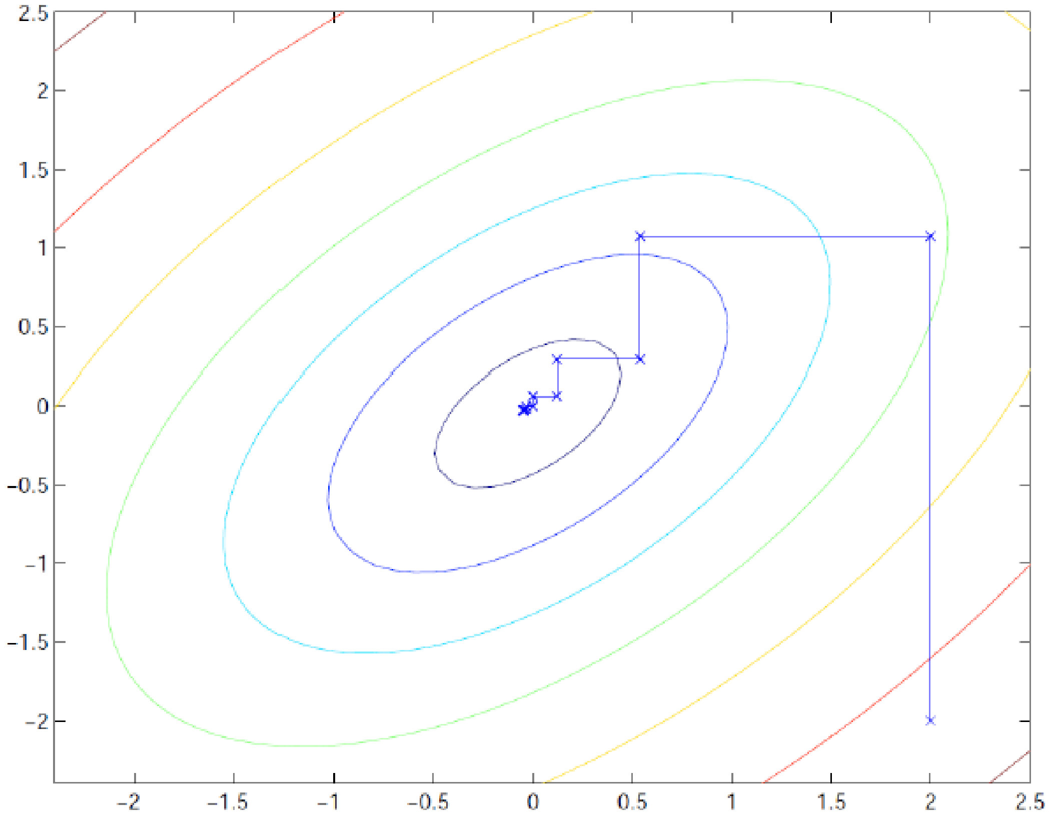
- For the optimization problem

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

- Coordinate ascent algorithm

$$
\begin{array}{l}
\text{Loop until convergence: } \{ \\
\quad \text{For } i = 1, \ldots, N \ \{ \\
\qquad \alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_N) \\
\quad \} \\
\}
\end{array}
$$

Illustration



A two-dimensional coordinate ascent example

Sequential minimal optimization (SMO)

- Recall the SVM optimization problem:

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

- 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 α_j, we can't make any changes to α_i

Solution

- Update two variable each time

  Loop until convergence {

  1. Select some pair α_i and α_j to update next
  2. Re-optimize $W(\alpha)$ w.r.t. α_i and α_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 α_i and α_j (step 2) is efficient

SMO (cont.)

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

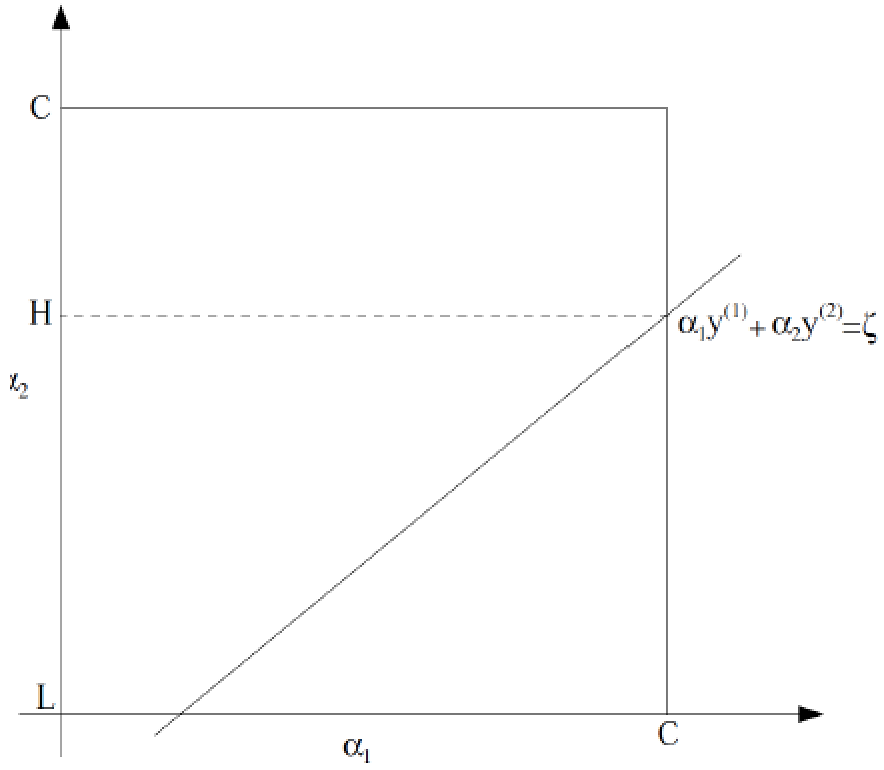
$$s.t. \quad 0 \leq \alpha_i \leq C, \qquad i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0$$

- Without loss of generality, hold $\alpha_3 \dots \alpha_m$ and optimize $w(\alpha)$ w.r.t α_1 and a_2

$$\alpha_1 y^1 + \alpha_2 y^2 = -\sum_{i=3}^{N} \alpha_i y^i = \zeta$$

$$\Rightarrow \alpha_1 = y^1(\zeta - \alpha_2 y^2)$$



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

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

$$s.t. \quad 0 \le \alpha_i \le C, \qquad i = 1, \dots, N$$

$$\sum_{i=1}^{N} \alpha_i y^i = 0$$

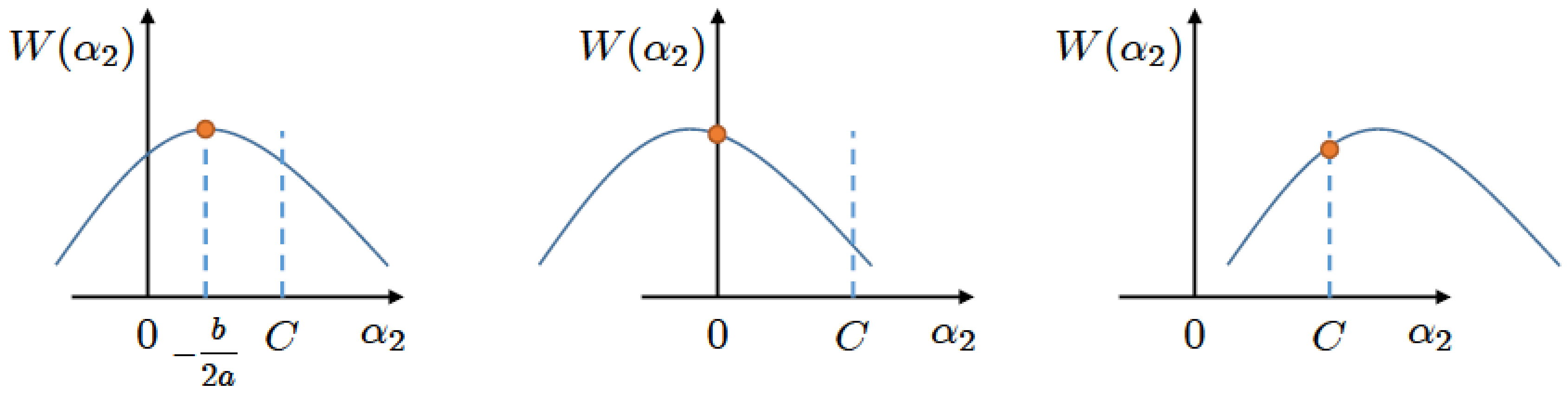
is transformed into a quadratic optimization problem w.r.t α_2

$$\max_{\alpha_2} \quad W(\alpha_2) = a\alpha_2^2 + b\alpha_2 + c$$

$$\text{s.t.} \quad 0 \le \alpha_2 \le C$$

SMO (cont.)

- Optimizing a quadratic function is much efficient



$$\max_{\alpha_2} \quad W(\alpha_2) = a\alpha_2^2 + b\alpha_2 + c$$

$$\text{s.t.} \quad 0 \leq \alpha_2 \leq C$$

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 α 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

Summary

- 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

Shuai Li

https://shuaili8.github.io

Questions?

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