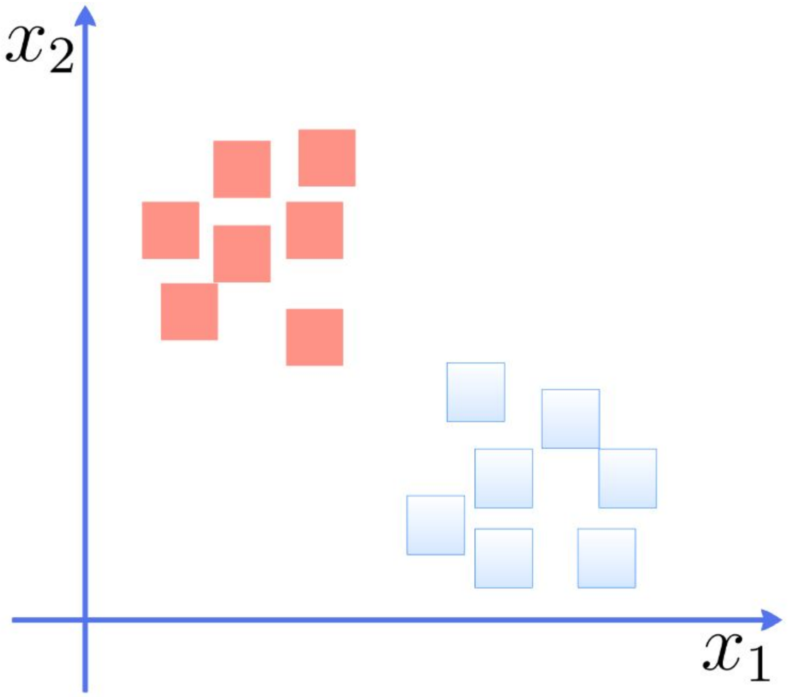
K6312 Information Mining & Analysis

Chen Zhenghua & Zhao Rui

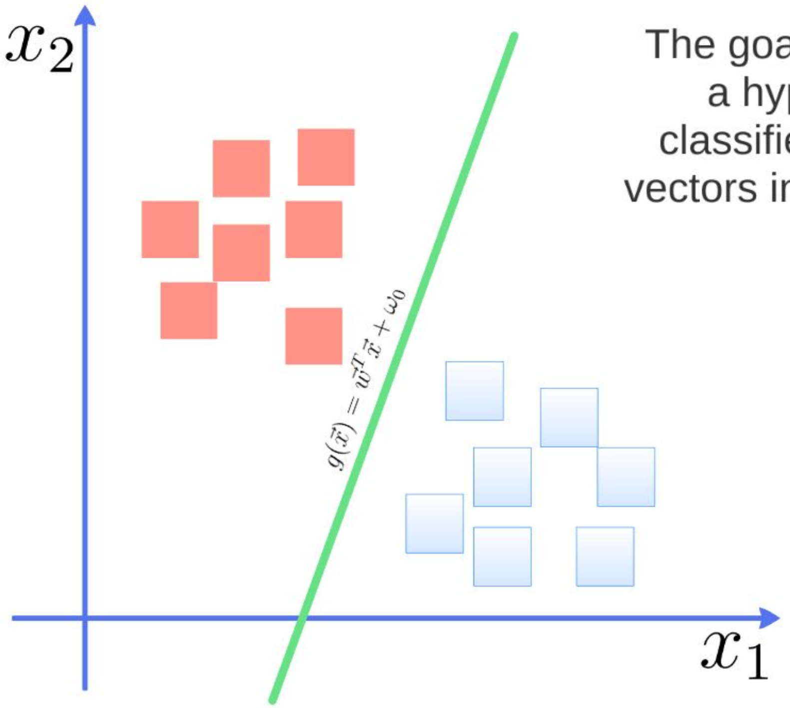
Support Vector Machine

Linearly separable binary sets



Source: https://anchor.fm/tkorting

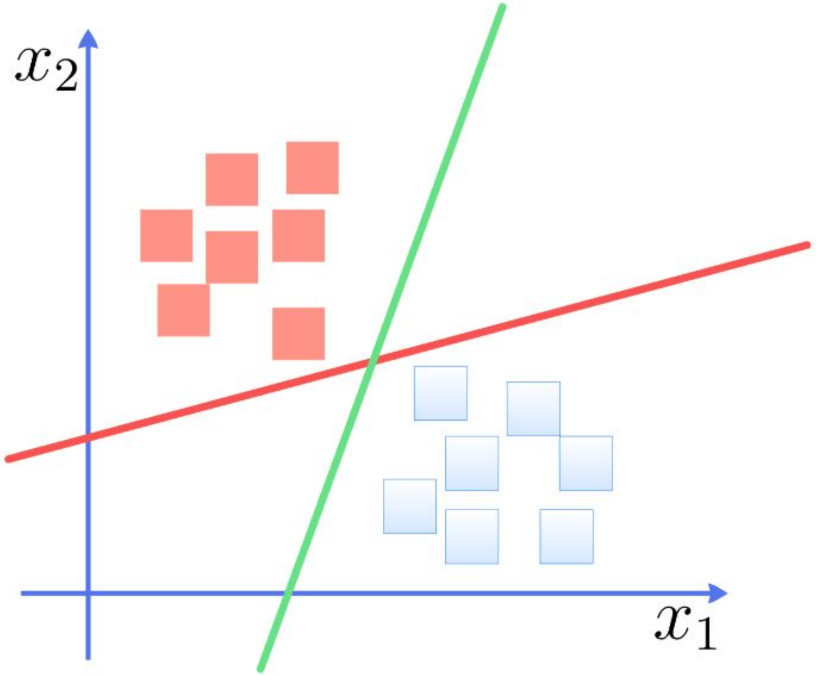
Linearly separable binary sets



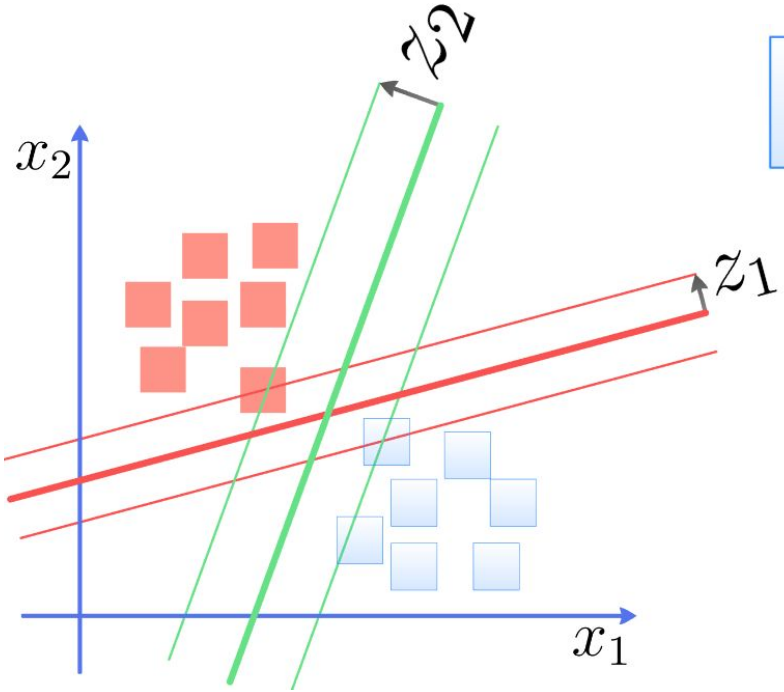
The goal is to design a hyperplane that classifies all training vectors in two classes

Linearly separable binary sets

The best choice will be the hyperplane that leaves the maximum margin from both classes



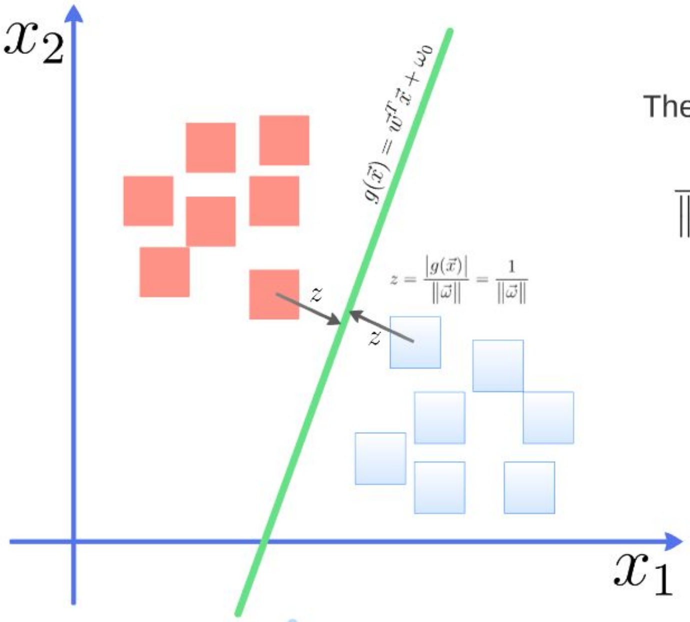
Linearly separable binary sets



$$z_2 > z_1$$

SVM for linearly separable binary sets

$$g(\vec{x}) \geqslant 1, \quad \forall \vec{x} \in \text{class } 1$$
$$g(\vec{x}) \leqslant -1, \quad \forall \vec{x} \in \text{class } 2$$

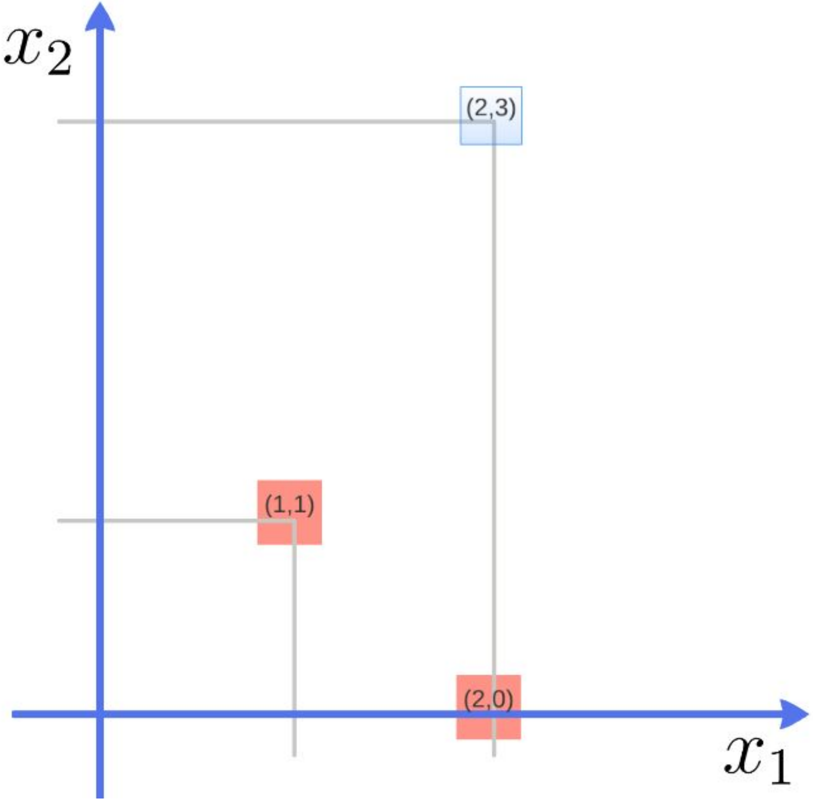


The total margin is computed by

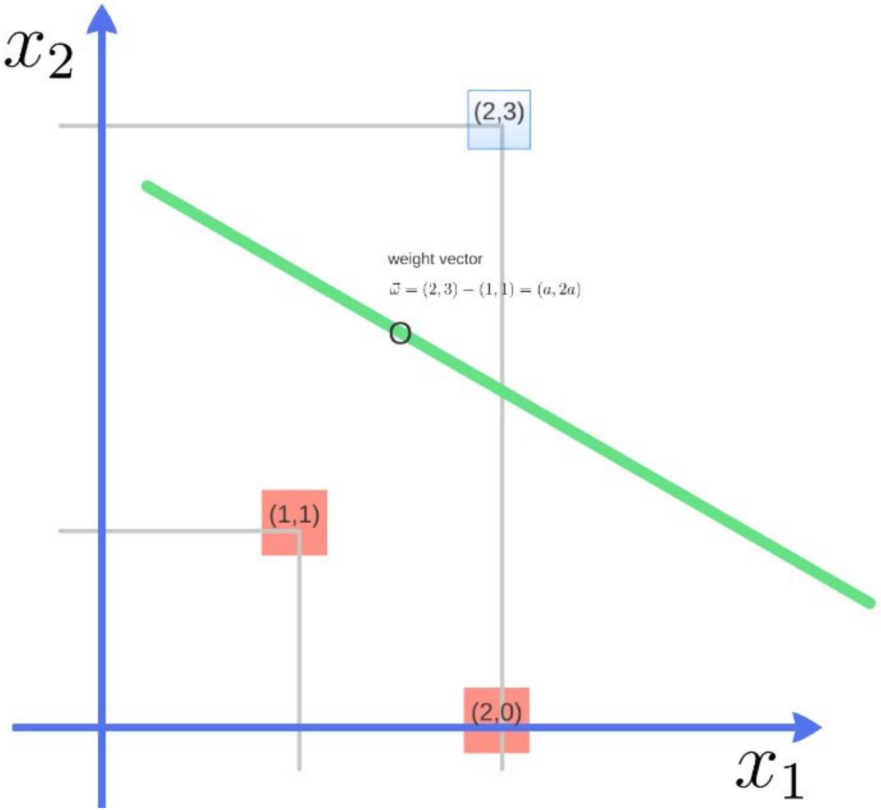
$$\frac{1}{\|\vec{\omega}\|} + \frac{1}{\|\vec{\omega}\|} = \frac{2}{\|\vec{\omega}\|}$$

Minimizing this term will maximize the separability

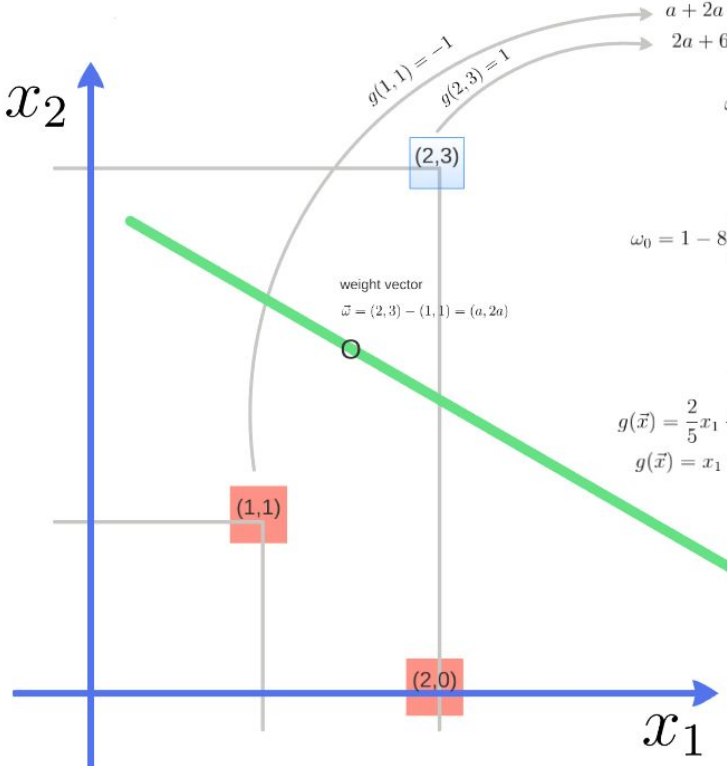
Example



Example



Example



weight vector $\vec{w} = (a, 2a)$

$a + 2a + \omega_0 = -1,$ using point (1,1)

$2a + 6a + \omega_0 = 1,$ using point (2,3)

...

$\omega_0 = 1 - 8a \qquad 3a + 1 - 8a = -1$

$\vdots \qquad 5a = 2$

$a = \frac{2}{5}$

$$\omega_0 = 1 - 8\frac{2}{5} = \frac{5-16}{5}$$

$$\omega_0 = -\frac{11}{5}$$

...

$$\vec{w} = (\frac{2}{5}, \frac{4}{5})$$

$$g(\vec{x}) = \frac{2}{5}x_1 + \frac{4}{5}x_2 - \frac{11}{5}$$

$$g(\vec{x}) = x_1 + 2x_2 - 5.5$$

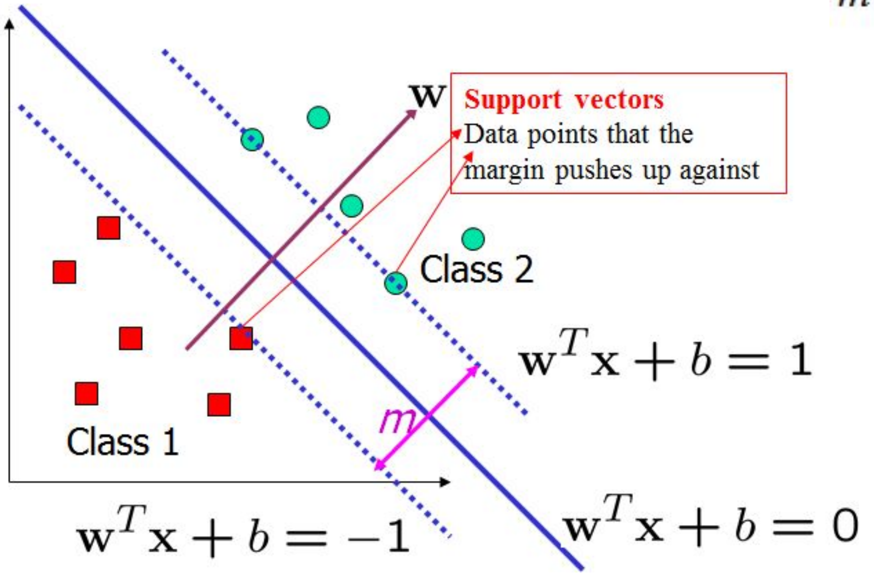
This is called the Support Vector Machine

Math Warning

Summary

The decision boundary should be as far away from the data of both classes as possible

$$m = \frac{2}{\sqrt{w.w}} \qquad m = \frac{2}{||\mathbf{w}||}$$



We should maximize the margin, *m*

Linear SVM

The Optimization Problem

Let $\{x_1, ..., x_n\}$ be our data set and let y_i {1,-1} be the class label of x_i

The decision boundary should classify all points correctly

A constrained optimization problem

$$m = \frac{2}{||\mathbf{w}||}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

Karush-Kuhn-Tucker (KKT) conditions

KKT:

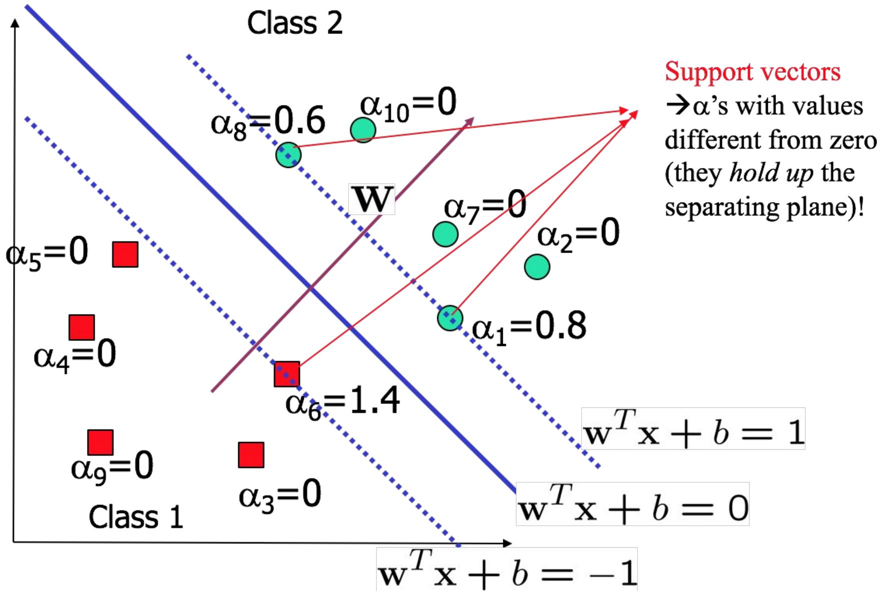
$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Lagrangian multipliers

Decision Hyperplane:

$$g(x) = w^T x + w_0 = \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T x + w_0 = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + w_0$$

Note that data only appears as dot products



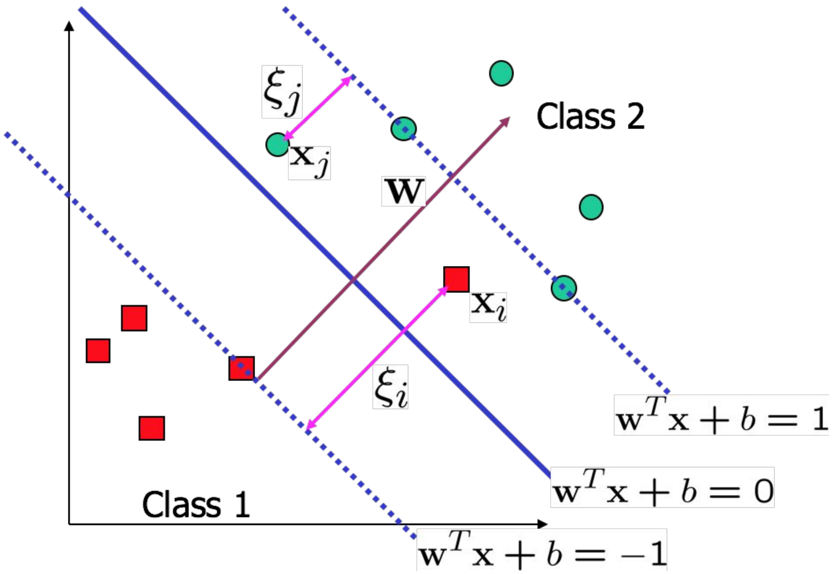
The dot product of two vectors $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_n]$ is defined as:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

From Wiki

Non-linearly Separable Problems

We allow "error" in classification; it is based on the output of the discriminant function wx+b



New objective function:

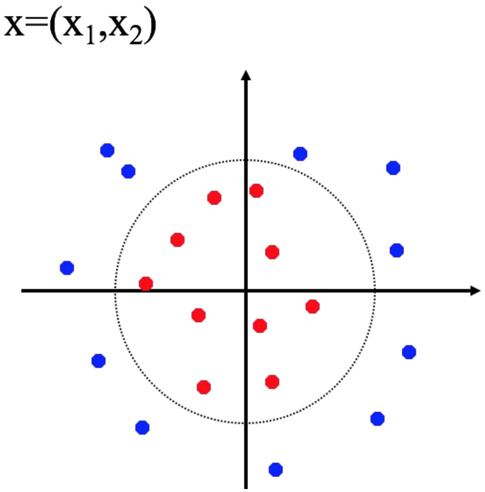
$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

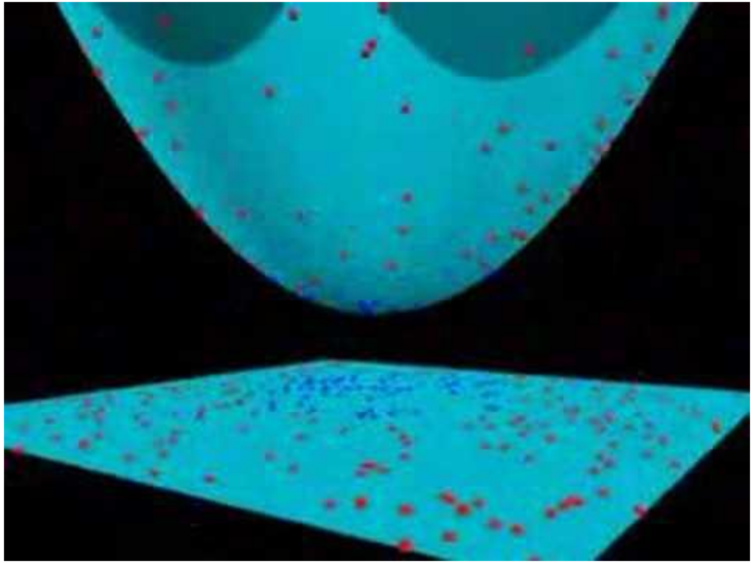
C : tradeoff parameter between error and margin;
chosen by the user;
large C means a higher penalty to errors

Extension to Non-linear SVM (Kernel Method)

How to deal the non-linear data

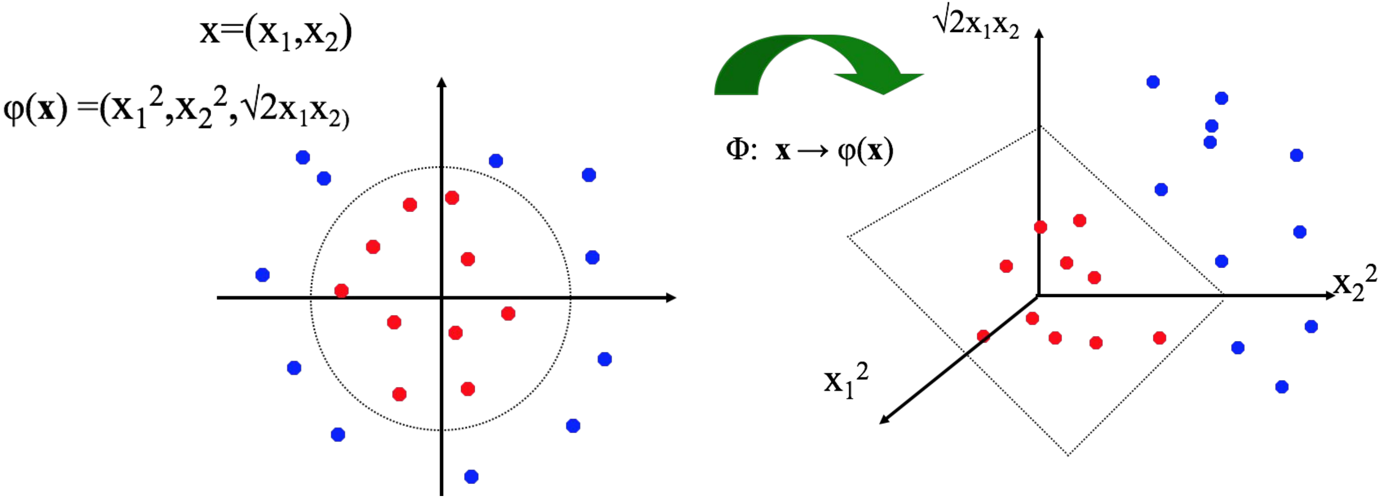
Linear decision boundary cannot work





Non-linear SVMs: Feature Space

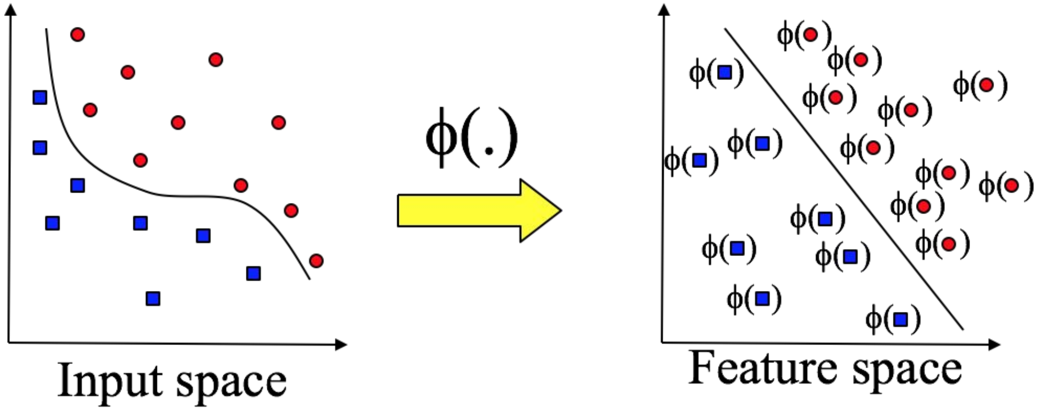
General idea: the original input space (x) can be mapped to some higher-dimensional feature space ($\varphi(\mathbf{x})$)where the training set is separable:



If data are mapped into higher a space of sufficiently high dimension, then they will in general be linearly separable; N data points are in general separable in a space of N-1 dimensions or more!!!

Transformation to Feature Space

- Possible problem of the transformation
  - High computation burden due to high-dimensionality and hard to get a good estimate
- SVM solves these two issues simultaneously
  - "Kernel tricks" for efficient computation
  - Minimize $||\mathbf{w}||^2$ can lead to a "good" classifier



Solution

Decision Hyperplane:

$$g(x) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + w_0 = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + w_0$$

Introduce a Kernel Function (*) *K* such that:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

The inner product can be computed by *K* without going through the map $\phi(\cdot)$ explicitly!!!

Example Transformation

Consider the following transformation

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$

Define the kernel function *K* (**x**,**y**) as

$$\begin{aligned} \langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) \rangle &= (1 + x_1y_1 + x_2y_2)^2 \\ &= K(\mathbf{x}, \mathbf{y}) \end{aligned}$$

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

Examples of Kernel Functions

- Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel

$$K(\mathbf{x}, \mathbf{y}) = \text{exp}(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

- Hyperbolic tangent kernel

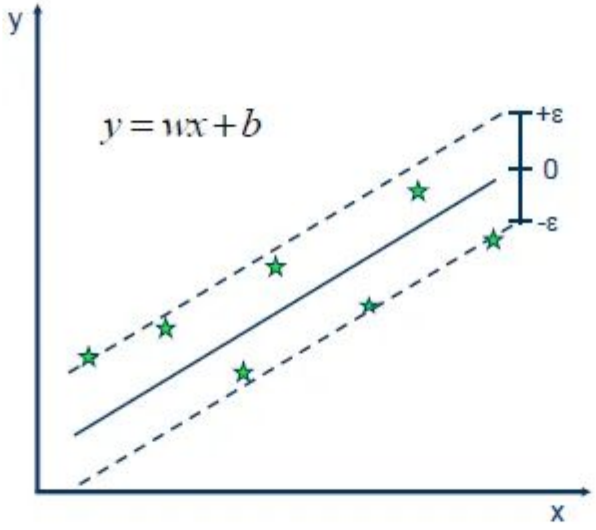
$$K(\mathbf{x}, \mathbf{y}) = \text{tanh}(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

- Research on different kernel functions in different applications is very active

Support Vector Regression

Formulation

SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data.



- Solution:

$$\min \frac{1}{2}\|w\|^2$$

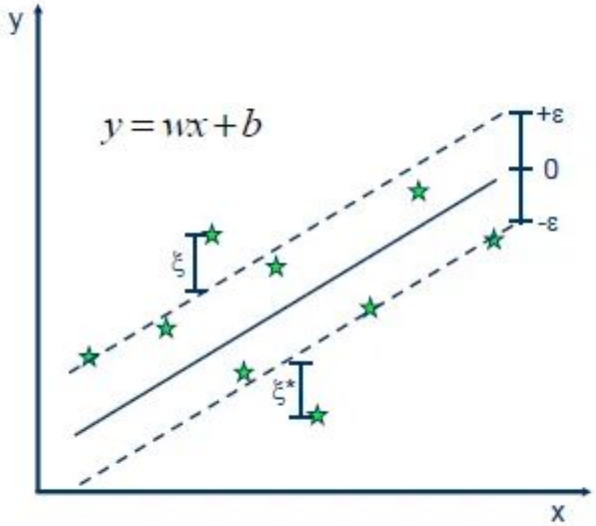
- Constraints:

$$y_i - wx_i - b \le \varepsilon$$
$$wx_i + b - y_i \le \varepsilon$$

Linear SVR

hyperplane:

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



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)$$

- Constraints:

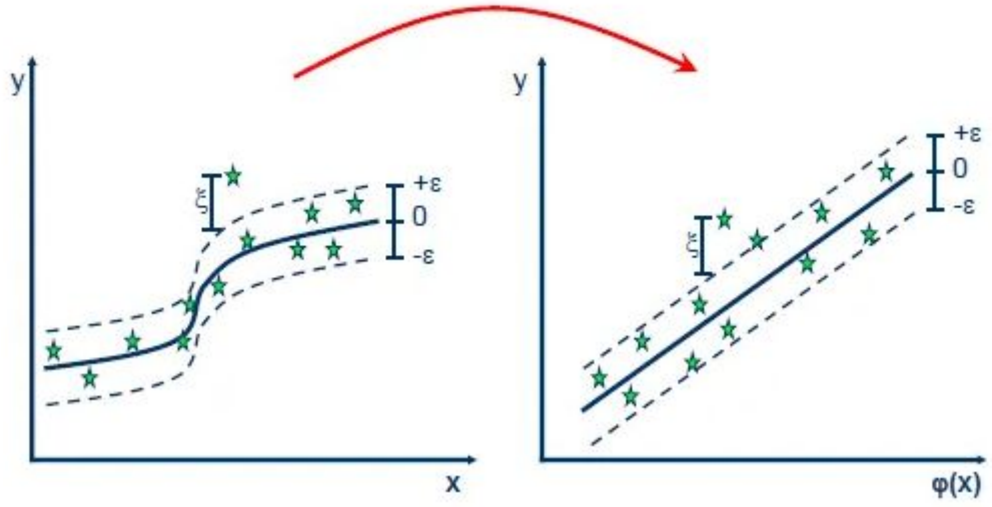
$$y_i - wx_i - b \le \varepsilon + \xi_i$$

$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

Non-Linear SVR

Kernel Track:



$$y = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) \cdot \left\langle \varphi(x_i), \varphi(x) \right\rangle + b$$

$$y = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) \cdot K(x_i, x) + b$$

Connections between SVM & SVR

Both cases result in the following problem:

$$\min \frac{1}{2} w^2$$

Under the condition that:

- All samples are classified correctly (Classification)
- The value y of all samples deviates less than ϵ from f(x). (Regression)

Weaknesses

- Training (and Testing) is quite slow compared to ANN
    - Because of Constrained Quadratic Programming
- Essentially a binary classifier
    - However, there are some tricks to evade this.
- Very sensitive to noise
    - A few off data points can completely throw off the algorithm
- Biggest Drawback: The choice of Kernel function
    - There is no “set-in-stone” theory for choosing a kernel function for any given problem (still in research...)

Strengths

- Training is relatively easy
  - We don’t have to deal with local minimum like in ANN.
  - SVM solution is always global and unique.
- Less prone to overfitting
- Simple, easy to understand geometric interpretation
  - No large networks to mess around with.