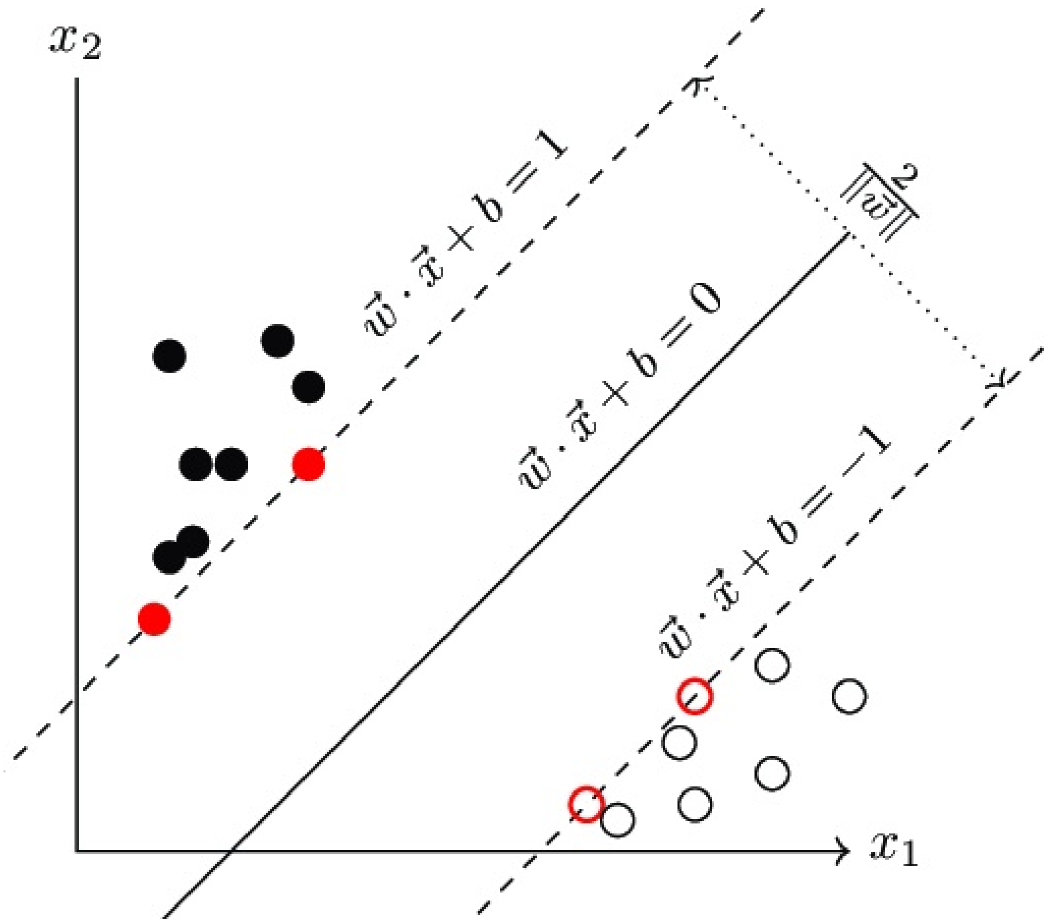
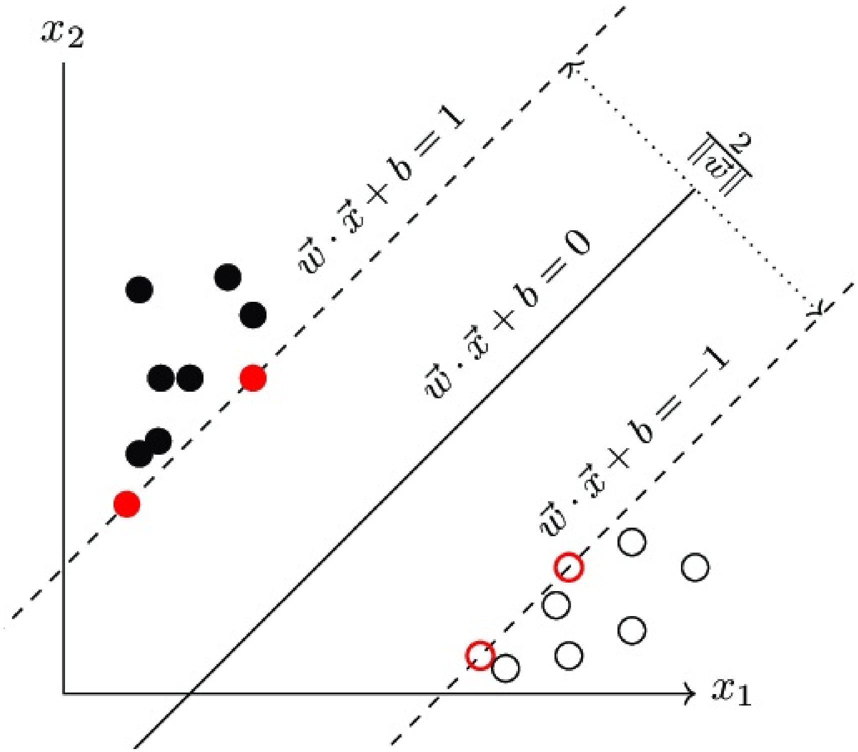


x_2
$\vec{w} \cdot \vec{x} + b = 1$
$\vec{w} \cdot \vec{x} + b = 0$
$\vec{w} \cdot \vec{x} + b = -1$
$\frac{2}{\|\vec{w}\|}$
x_1



$$\min_{w,b} \ \frac{1}{2}\|w\|_2^2$$

s.t.

$$y_i(w^T x_i + b) \geq 1 \qquad i=1,\dots,n$$

$$L(w,b,\lambda)$$

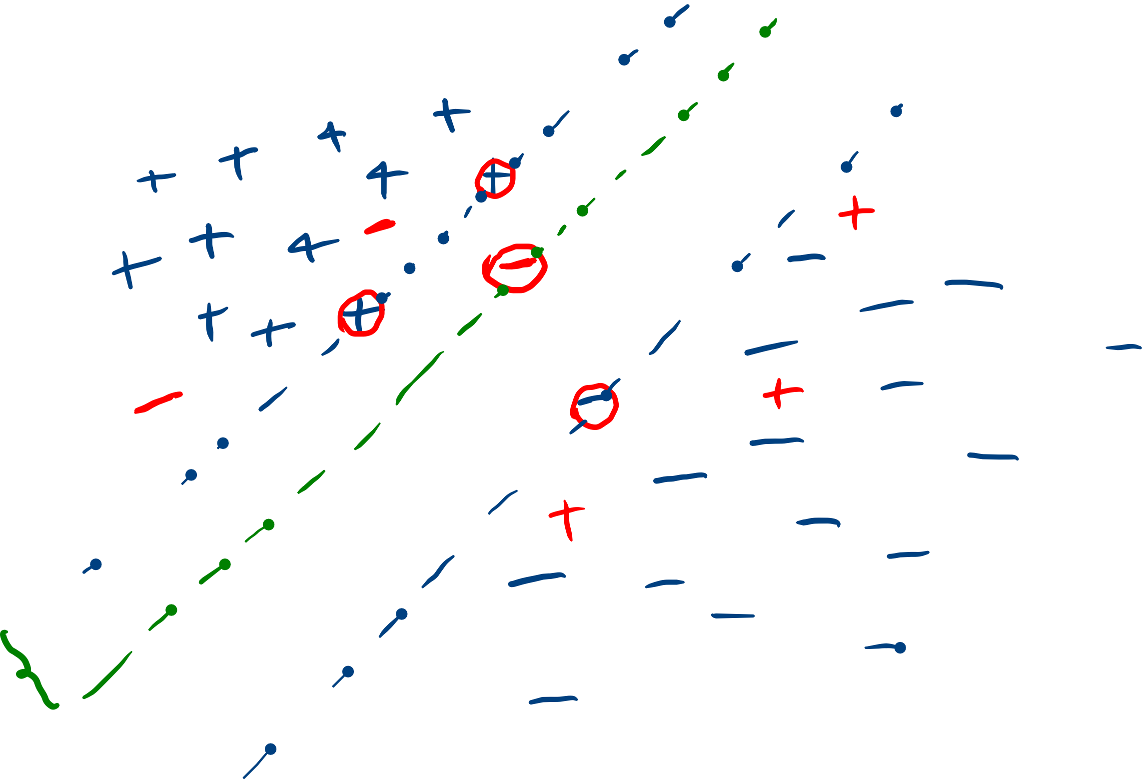
$$\nabla_w L = 0 \Rightarrow \boxed{w = \sum_{i=1}^{n} \lambda_i y_i x_i}$$

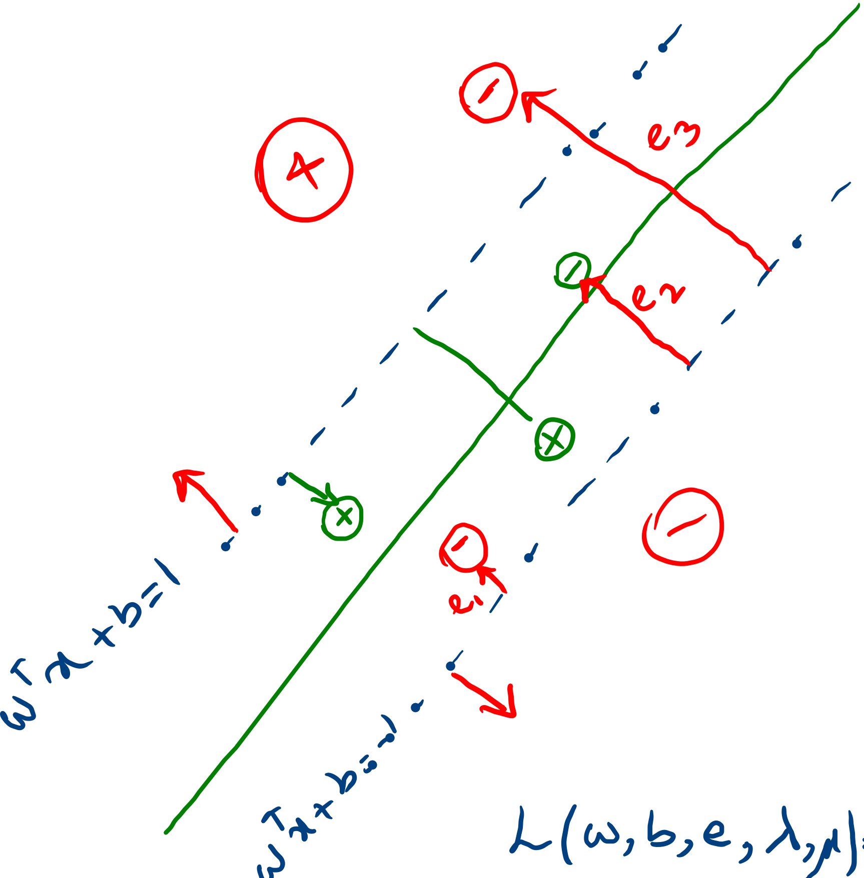
$$\nabla_b L = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\max_{\lambda} \; -\frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \lambda_i$$

Quadratic Programing

$$\lambda_i \geq 0 \;, \quad \sum \lambda_i y_i = 0$$





Soft Margin SVM

$$\min_{w,b,e} \ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} e_i$$

s.t.

$$y_i(w^T x_i + b) \geq 1 - e_i$$

$$e_i \geq 0$$

$$L(w,b,e,\underbrace{\lambda,\mu}_{\text{dual var}}) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} \lambda_i\left(1 - e_i - y_i(w^T x_i + b)\right) + \sum_{i=1}^{n} \mu_i(-e_i)$$

$$\nabla_{e_j} L = C - \lambda_j - \mu_j = 0 \Rightarrow \boxed{\mu_j = C - \lambda_j} \geq 0 \Rightarrow 0 \leq \lambda_j \leq C$$

$$\max_{\lambda} \; -\frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j \underbrace{x_i^T x_j} + \sum_i \lambda_i$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0 , \qquad \underbrace{0 \leq \lambda_i \leq C}$$

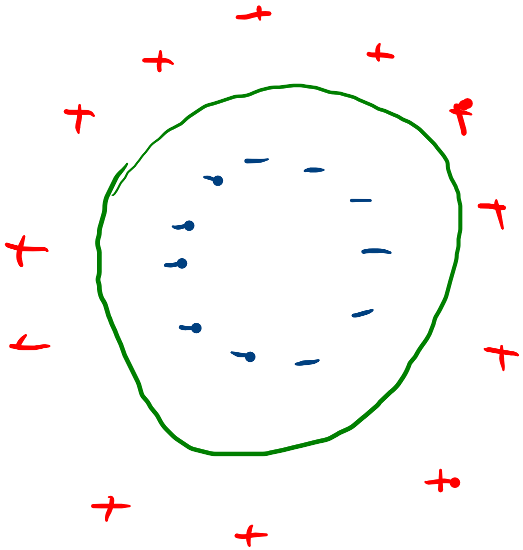
λ^*

$$\boxed{K(x_i, x_j) = \phi(x_i)^T \phi(x_j)}$$

$$w = \sum_{i=1}^{n} \lambda_i y_i \phi(x_i)$$

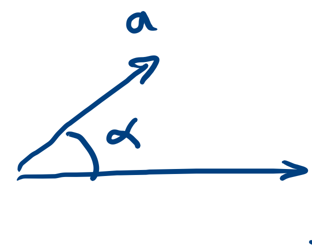
$$w^T \phi(x) + b$$

$$g(x) = \sum_{i=1}^{n} \lambda_i y_i \underbrace{\phi(x_i)^T \phi(x)}_{K(x_i, x)} + b \; \begin{matrix} + \\ \gtrless \\ - \end{matrix} \; 0$$



r

θ



$|a|\,|b|\,\underbrace{\cos\alpha}_{1}$

$x_1, \ldots, x_n$

Φ

$\Phi(x_1), \ldots, \Phi(x_n)$

Kernel Trick

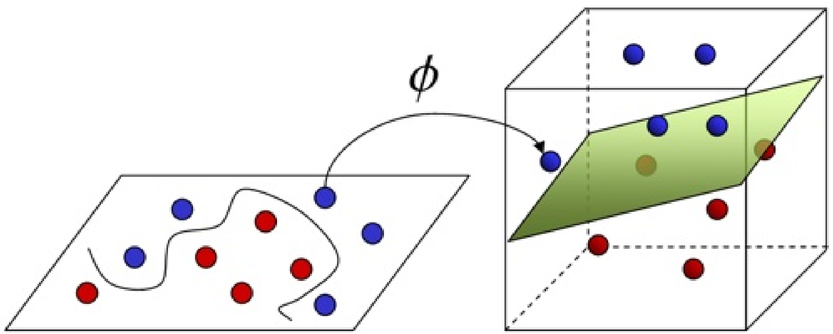
$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$$

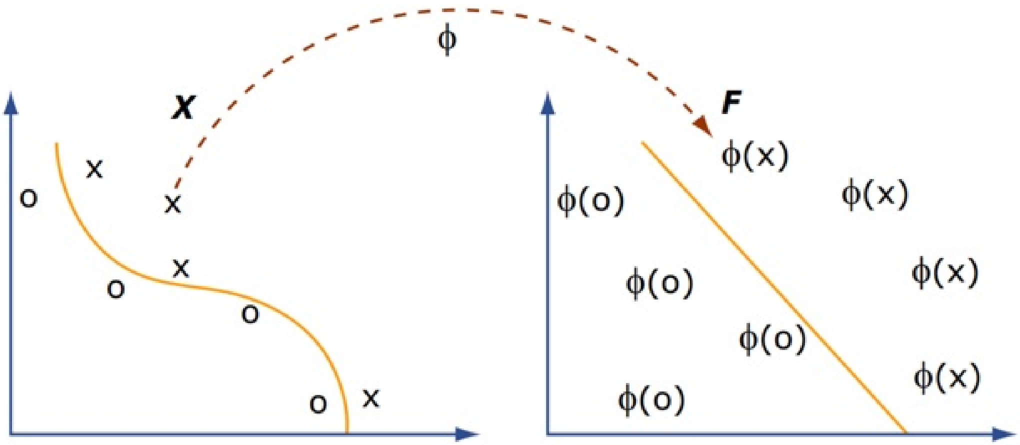
$$K(x_i, x_j) = \cancel{x_i x_j} + \cos(x_i + x_j)$$

Mercer's Theorem:

$$\iint$$

Transforming the Data





- **Feature space** is of **higher dimension** than the **input space** in practice
- Computation in the feature space can be **costly** because it is high dimensional
- The **kernel trick** comes to rescue

kernel trick

- Suppose ϕ(.) is given as follows

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

- An **inner product** in the feature space is

$$\langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle = (1 + x_1y_1 + x_2y_2)^2$$

- So, if we define the kernel function as follows, there is no need to **carry out ϕ(.) explicitly**

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

- This use of kernel function to **avoid carrying out** ϕ(.) **explicitly** is known as the kernel trick

Examples of Kernel Functions

- **Polynomial** kernel with **degree** d

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

- **Gaussian kernel** with width σ

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

RBF Φ

  - Closely related to **radial basis function neural networks**
  - The feature space is **infinite-dimensional** (it still be written as a dot product in a new feature space k(**x**, **x**$_0$) = Φ(**x**) ∗ Φ(**x**$_0$), only with an **infinite number of dimensions**)

- **Sigmoid** with parameter κ and θ

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

  - It does not satisfy the **Mercer condition** on all κ and θ

$$K(x,y) = e^{-\|x-y\|^2} = e^{-(x-y)^T(x-y)} = \underbrace{e^{-x^Tx}}\,\underbrace{e^{-y^Ty}}\,\underbrace{e^{+2x^Ty}}$$

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots$$

$$\phi(x)^T\phi(y)$$

Mercer Theorem

$$\iint g(x)K(x,y)g(y)\,dx\,dy \geq 0$$

Techniques for Constructing New Kernels.

Chapter 6

Bishop

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following new kernels will also be valid:

$c > 0$

$$k(\mathbf{x}, \mathbf{x}') = ck_1(\mathbf{x}, \mathbf{x}') \quad (6.13)$$
$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})\underline{k_1(\mathbf{x}, \mathbf{x}')}f(\mathbf{x}') \quad (6.14)$$
$$k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}')) \quad (6.15)$$
$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}')) \quad (6.16)$$
$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \quad (6.17)$$
$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}') \quad (6.18)$$
$$k(\mathbf{x}, \mathbf{x}') = k_3(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')) \quad (6.19)$$
$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}' \quad (6.20)$$
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (6.21)$$
$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a)k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (6.22)$$

k_1

k_2

$K(x, y)$

$K(\phi(x), \phi(y))$

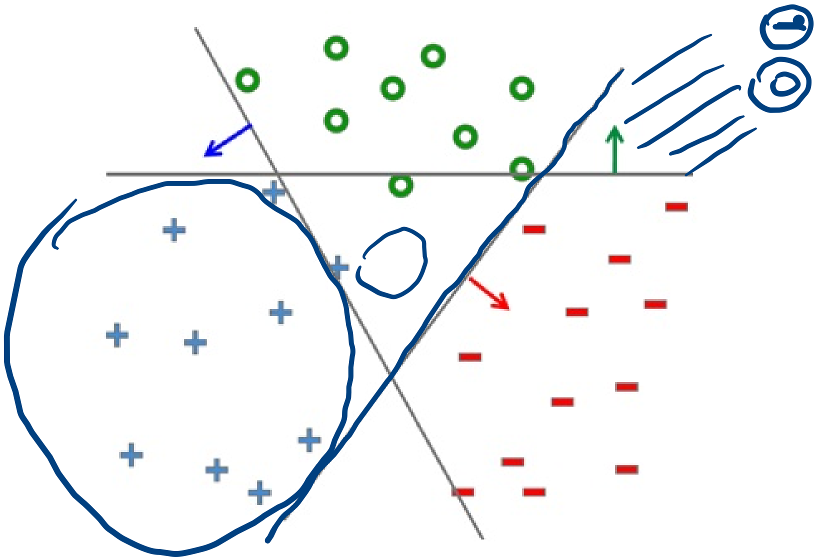
where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, $\boldsymbol{\phi}(\mathbf{x})$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, $k_3(\cdot, \cdot)$ is a valid kernel in $\mathbb{R}^M$, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\mathbf{x}_a$ and $\mathbf{x}_b$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, and k_a and k_b are valid kernel functions over their respective spaces.

$$K(x,y) = x^T A y = x^T B^T B y = \underbrace{(Bx)^T}_{\Phi(x)^T} \underbrace{(By)}_{\Phi(y)}$$

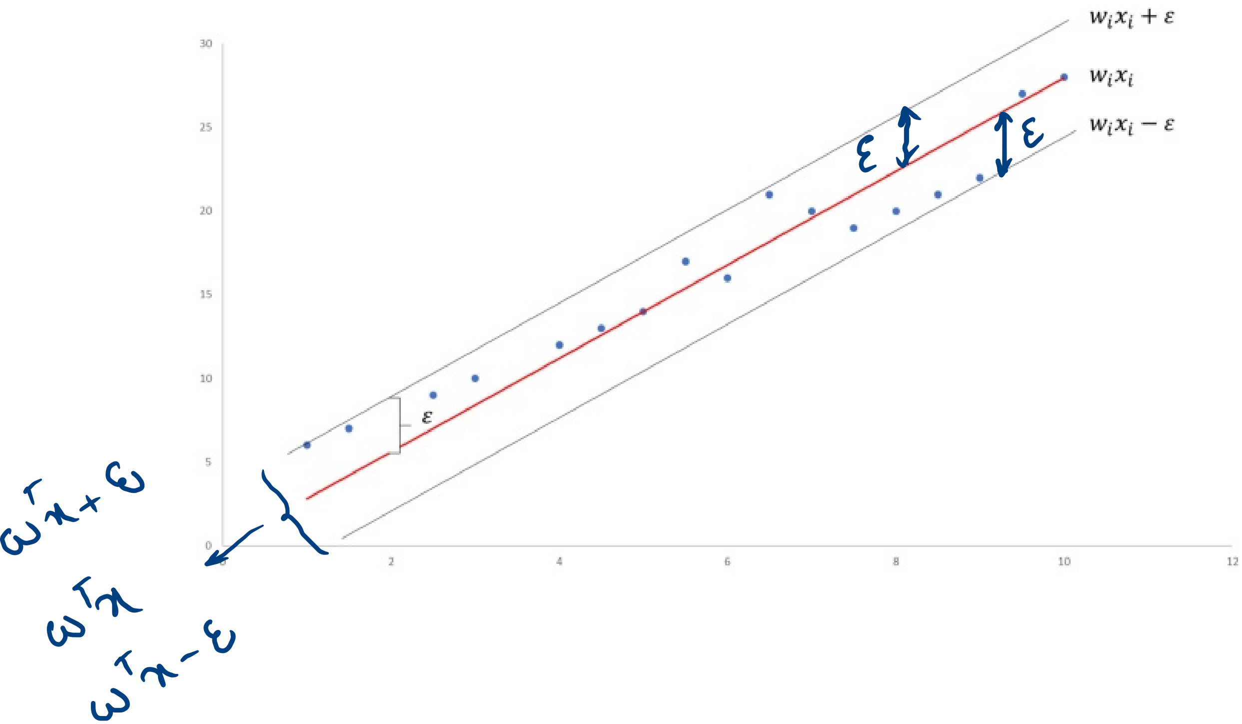
$$A = B^T B$$

What about multiple classes?

- One against all
  - **Learn 3 classifiers separately**:
    - Class k vs. rest
  - $(\mathbf{w}_k, b_k)_{k=1,2,3}$
  - $y = \arg\max_k \mathbf{w}_k x + b_k$
  - Disadvantages: ambiguous area
- In each step, **remove one class**:
  - Problem: sensitive to order
- One against one:
  - **Majority voting**



Support Vector Regression (SVR)



Minimize:

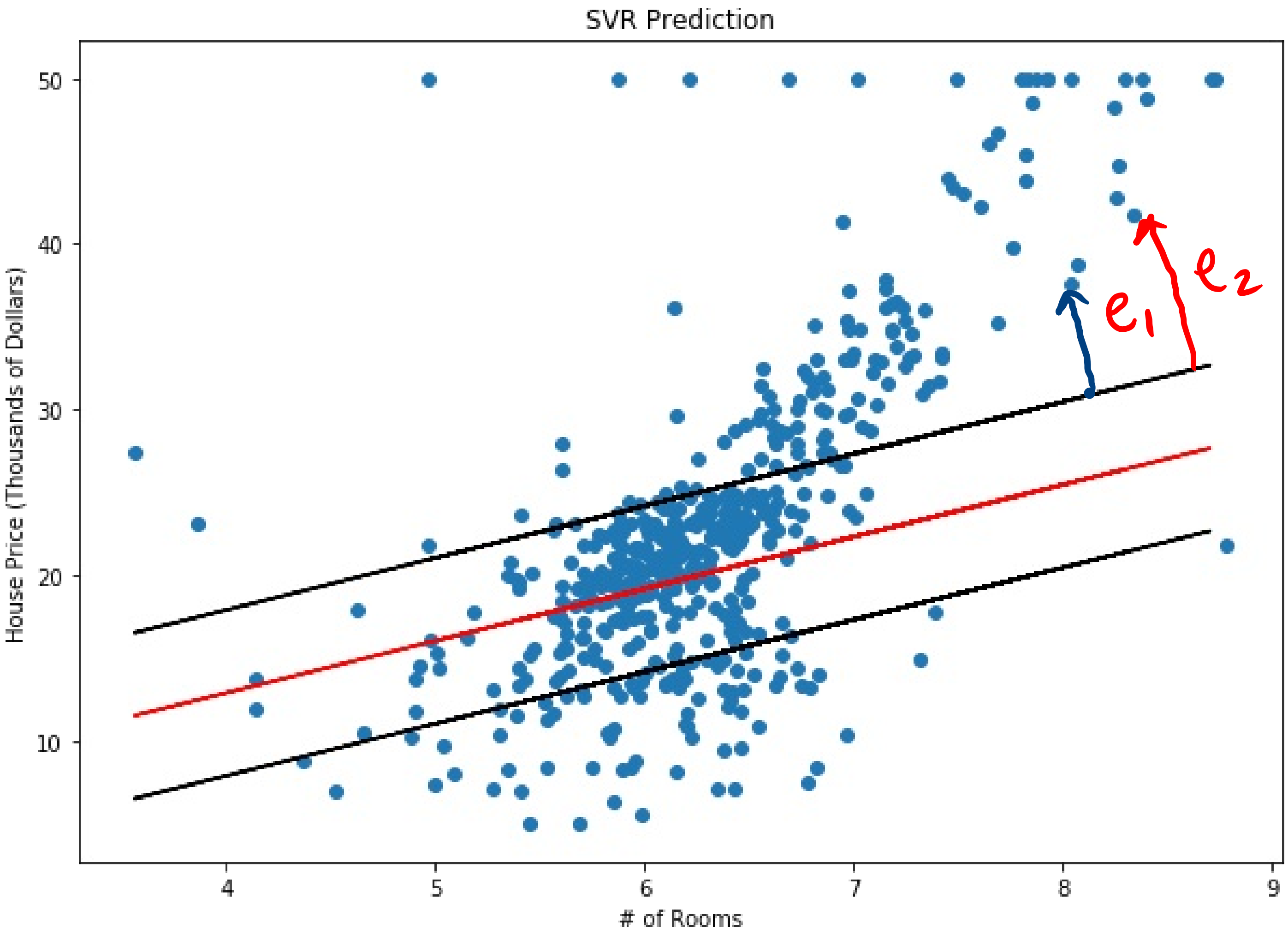
$$MIN \ \frac{1}{2}||\mathbf{w}||^2$$

Constraints:

$$|y_i - w_i x_i| \ \leq \ \varepsilon$$

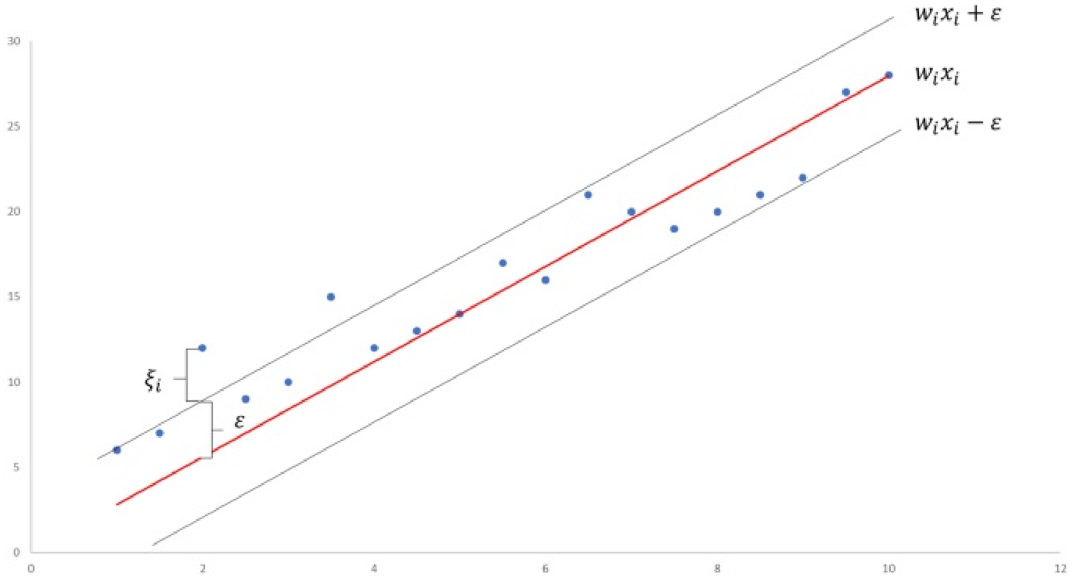
Ordinary Least Squers (OLS):

$$MIN \sum_{i=1}^{n} (y_i - w_i x_i)^2$$


SVR Prediction
House Price (Thousands of Dollars)
50
40
30
20
10
4
5
6
7
8
9
# of Rooms
e1
e2

e_i

Using Slack Variables



Minimize:

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

Constraints:

$$|y_i - w_i x_i| \leq \varepsilon + \underbrace{|\xi_i|}_{e_i}$$