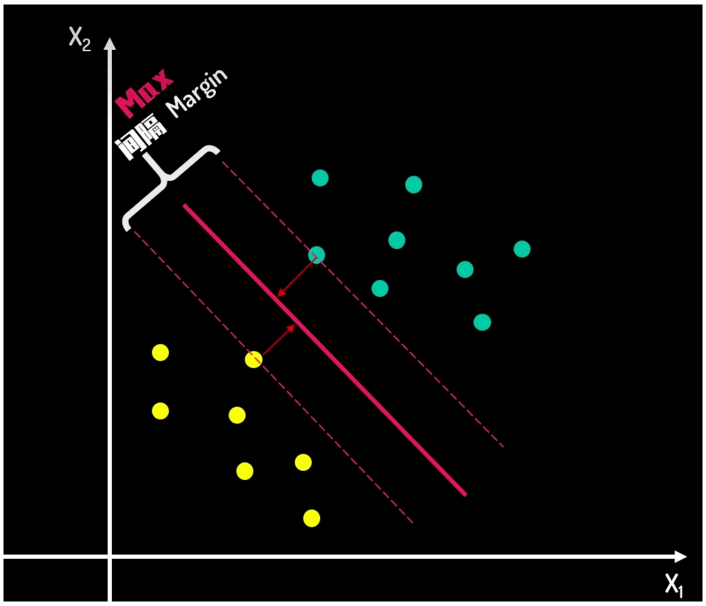
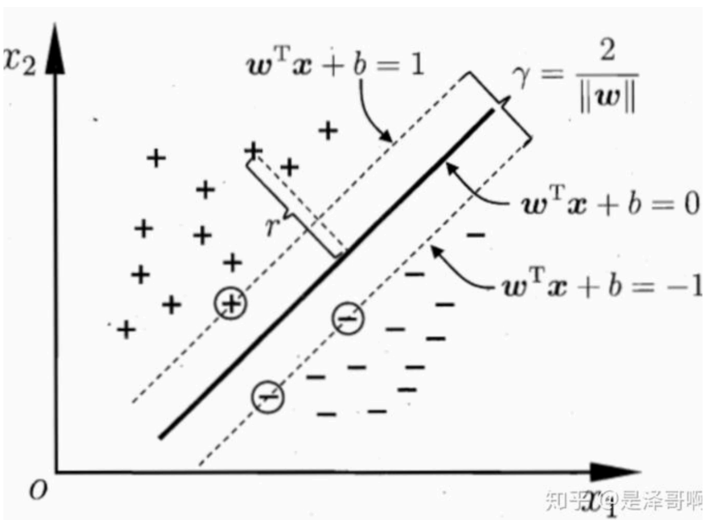
I. Goal — Maximize Margin



Distance: $x(x_1 \cdots x_n)$ to $w^T x + b$ is $\dfrac{|w^T x + b|}{\|w\|}$

We have

$$\begin{cases} \dfrac{w^T x + b}{\|w\|} \geq d & y = 1 \\ \dfrac{w^T x + b}{\|w\|} \leq -d & y = -1 \end{cases} \Rightarrow \begin{cases} \dfrac{w^T x + b}{\|w\| d} \geq 1 & y = 1 \\ \dfrac{w^T x + b}{\|w\| d} \leq -1 & y = -1 \end{cases} \xRightarrow{\text{let } \|w\| d = 1} y(w^T x + b) \geq 1$$



Hence, in order to maximize γ, we want to minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) \geq 1$, $i = 1, 2 \cdots$

II. Lagrange Mutiplier

In order to use this method, we need to first convert inequality constrain into equality constrain, i.e.

Let $1 - y_i(w^T x_i + b) = -a_i^2 \leq 0$, let $g_i(w) = 1 - y_i(w^T x_i + b)$

$$\Rightarrow L(w,\lambda_i,p_i) = \frac{\|\vec{w}\|^2}{2} + \sum_{i=1}^{S} \lambda_i \cdot [g_i(w) + a_i^2]$$

$$\begin{cases} \frac{\partial L}{\partial w} = \vec{w} - \sum_{i=1}^{S} \lambda_i y_i \vec{x_i} = 0 \quad \sum_{i=1}^{S} \lambda_i (1 - y_i(w^T x_i + b) + a_i^2) \\ \frac{\partial L}{\lambda_i} = 1 - y_i(w^T x_i + b) + a_i^2 = 0 \\ \frac{\partial L}{\partial a_i} = \lambda_i a_i = 0 \quad \cdot\!\times \\ \lambda_i \geq 0 \text{ (KKT)} \end{cases}$$

For $\lambda_i a_i = 0$

$$\left.\begin{array}{l} 1^\circ\ \lambda_i = 0 \Rightarrow \lambda_i g_i(w) = 0 \\ 2^\circ\ \lambda_i \neq 0 \Rightarrow a_i = 0 \Rightarrow g_i(w) = 0 \end{array}\right\} \Rightarrow \lambda_i g_i(w) = 0$$

$$\Rightarrow \begin{cases} \vec{w} - \sum_{i=1}^{S} \lambda_i y_i \vec{x_i} = 0 \\ \lambda_i g_i(w) = 0 \\ g_i(w) \leq 0 \\ \lambda_i \geq 0 \end{cases}$$

Since $a_i^2 \geq 0$ and can be arbitrary number

$$\min \frac{1}{2}\|\vec{w}\|^2 + \sum_{i=1}^{S} \lambda_i [g_i(w) + a_i^2] \Rightarrow \min \frac{1}{2}\|\vec{w}\|^2 + \sum_{i=1}^{S} \lambda_i g_i(w) = L(w,b,\lambda)$$

Suppose the min of $\frac{1}{2}\|\vec{w}\|^2 = p$, Since $\lambda_i g_i(w) \leq 0$, so

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

We want to find λ s.t. $L(w,b,\lambda)$ is as close to p as possible.

So the original optimization can be stated as

$$\min_{w,b} \max_{\lambda} L(w,b,\lambda) \quad \text{s.t. } \lambda_i \geq 0$$

Since KKT $\Leftrightarrow$ Strong Duality, it's equivalent of

$$\max_{\lambda} \min_{w,b} L(w,b,\lambda) \quad s.t. \quad \lambda_i \geq 0$$

III. After Duality Conversion

$$\max_{\lambda} \min_{w,b} L(w,b,\lambda) = \frac{1}{2}\|\vec{w}\|^2 + \sum_{i=1}^{n} \lambda_i(1 - y_i(w^T x_i + b))$$

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \lambda_i \vec{x_i} y_i \\ \frac{\partial L}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \end{cases}$$

plug them into $L(w,b,\lambda)$

$$L(w,b,\lambda) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (\vec{x_i} \cdot \vec{x_j}) + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n} \lambda_i y_i \left(\sum_{j=1}^{n} \lambda_j y_j (x_i \cdot x_j) + b\right)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i^T x_j) + \sum_{i=1}^{n} \lambda_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i y_i \lambda_j y_j (x_i^T x_j) - \sum_{i=1}^{n} \lambda_i y_i b$$

$$= \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i^T \cdot x_j)$$

$\Rightarrow$ now we want to maximize

$$\max_{\lambda} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j (x_i^T \cdot x_j), \quad s.t. \begin{cases} \sum_{i=1}^{n} \lambda_i y_i = 0 \\ \lambda_i \geq 0 \end{cases}$$

IV. Sequential Minimal Optimization (vary one at a time, fix others)

Suppose we want to optimize λ_1, λ_2, then we let

$$\lambda_1 y_1 + \lambda_2 y_2 = c = -\sum_{k \neq 1,2} \lambda_k y_k$$

So $\to$ can be simplified as

$$L(\lambda_1, \lambda_2) = \lambda_1 + \lambda_2 + \sum_{i=3}^{n} \lambda_i - \frac{1}{2}\Big(\lambda_1\lambda_1 y_1 y_1 (x_1^T \cdot x_1) + \lambda_1\lambda_2 y_1 y_2 (x_1^T x_2) +$$

$$\lambda_2\lambda_1 y_2 y_1 (x_2^T x_1) + \lambda_2\lambda_2 y_2 y_2 (x_2^T x_2) + 2\sum_{j=3}^{n} \lambda_1 \lambda_j y_1 y_j (x_1^T \cdot x_j)$$

$$+ 2\sum_{j=3}^{n} \lambda_2 \lambda_j y_2 y_j (x_2^T \cdot x_j) + \sum_{i=3}^{n}\sum_{j=3}^{n} \lambda_i \lambda_j y_i y_j (x_i^T \cdot x_j)\Big)$$

($i=1, j\geq 3$ and $j=1, i\geq 3$)

($i=2, j\geq 3$ and $j=2, i\geq 3$) ($i\geq 3, j\geq 3$)

Since $\lambda_2 = \frac{c - \lambda_1 y_1}{y_2}$, by subbing in

$$L(\lambda_1)=\lambda_1+\frac{C-\lambda_1 y_1}{y_2}+\sum_{i=3}^{n}\lambda_i-\frac{1}{2}\Big[\lambda_1^2 \underset{1}{y_1^2}\|x_1\|^2+2\lambda_1\frac{C-\lambda_1 y_1}{y_2}y_1y_2\langle x_1,x_2\rangle$$

$$+\left(\frac{C-\lambda_1 y_1}{y_2}\right)^2 \underset{1}{y_2^2}\|x_2\|^2+2\sum_{j=3}^{n}\lambda_1\lambda_j y_1 y_j\langle x_1,x_j\rangle+2\sum_{j=2}^{n}\frac{C-\lambda_1 y_1}{y_2}\lambda_j y_2 y_j\langle x_2,x_j\rangle$$

$$+\sum_{i=3}^{n}\sum_{j=3}^{n}\lambda_i\lambda_j y_i y_j\langle x_i,x_j\rangle\Big]$$

which is a quadratic function w.r.t λ_1.

We repeat this until we find all λ_i, then

$$w=\sum_{i=1}^{n}\lambda_i y_i \vec{x}_i$$

Take a random point on supporting vector, noted as $\vec{x_s}$, y_s, then

$$y_s(wx_s+b)=1$$

$$\Rightarrow\ y_s^2(wx_s+b)=y_s$$

$$\Rightarrow\ b=y_s-wx_s$$

We find w, b, hence w^Tx+b

V. Kernel Function

$\rightarrow$ it will be $\max\left[\sum_{i=1}^{n}\lambda_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j\langle T(\vec{x_i}),T(\vec{x_j})\rangle\right]$

subject to $\lambda_i\geq 0$

We want to find a function $k(\vec{x},\vec{y})=\langle T(\vec{x}),T(\vec{y})\rangle$, and it's called kernel function, which can hugely reduce memory cost. For example

$$k(\vec{x},\vec{y})=(\vec{x}\cdot\vec{y}+1)^2=(x_1^2,x_2^2\cdots x_n^2,\sqrt{2}x_1,\cdots\sqrt{2}x_n,1)\cdot(y_1^2,\cdots y_n^2,\sqrt{2}y_1,\cdots\sqrt{2}y_n,1)$$

Common kernel functions are

polynomial: $K(\vec{x},\vec{y})=(C+\vec{x}\cdot\vec{y})^d$

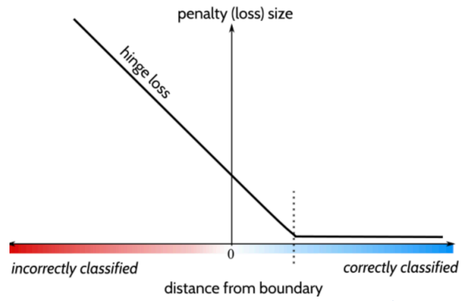
Gaussian (RBF) $K(\vec{x},\vec{y})=e^{\left(-\frac{\|\vec{x}-\vec{y}\|^2}{2\delta^2}\right)}$

↳ it can represent a mapping to inf dimension

Let $\delta = 1$

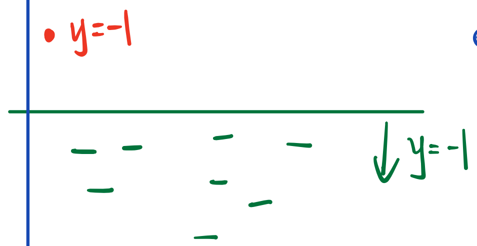
$$K(\vec{x},\vec{y}) = e^{\left(-\frac{\|\vec{x}-\vec{y}\|^2}{2}\right)} = e^{-\frac{1}{2}(\|\vec{x}\|^2+\|\vec{y}\|^2-2\vec{x}\cdot\vec{y})} = e^{-\frac{1}{2}(\|\vec{x}\|^2+\|\vec{y}\|^2)} \cdot e^{\vec{x}\cdot\vec{y}} = C\cdot e^{\vec{x}\cdot\vec{y}}$$

$$= C\cdot\sum_{n=0}^{\infty}\frac{(\vec{x_i}\cdot\vec{x_j})^n}{n!}$$



VI. Soft Margin

Allow some points land in between the gap $(1-y_i(w^Tx_i+b)\le 0)$



Suppose we have a point $y=-1$. then lost ε_i can be expressed as $\max(0, 1-y_i(w^Tx_i+b))$

We want to minize $\frac{\|\vec{w}\|^2}{2} + C\cdot\sum_{i=1}^{n}\varepsilon_i$,

after using Lagrange

We want to $\min\limits_{w,b,\varepsilon}\max\limits_{\lambda,\mu} \frac{1}{2}\|\vec{w}\|^2 + C\cdot\sum_{i=1}^{n}\varepsilon_i + \sum_{i=1}^{n}\lambda_i[1-\varepsilon_i-y_i(w^Tx_i+b)] - \sum_{i=1}^{n}\mu_i\varepsilon_i$

using duality $\Rightarrow$ $\max\limits_{\lambda,\mu}\min\limits_{w,b,\varepsilon} L(w,b,\varepsilon,\lambda,\mu)$

↳ it will become $\max\limits_{\lambda}\left[\sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \langle\vec{x_i},\vec{x_j}\rangle\right]$

subject to $\lambda_i \ge 0$, $\sum_{i=1}^{n}\lambda_i y_i = 0$, $C = \lambda_i + \mu_i$ (additional constrain)

Follow the same SMO algo in hard margin case.