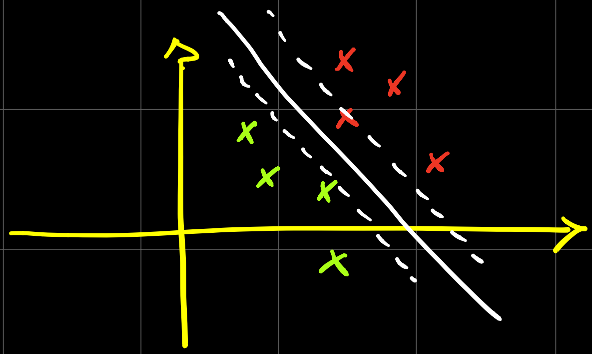
Support Vector Machines



Steps:

1) Set constraints

$$y_i(\vec{x}_i^T \vec{\omega} + b) - 1 \geq 0$$

2) Distance to separating hyperplane:

Minimize $d = \dfrac{y_i(\vec{x}_i^T \vec{\omega} + b)}{\|\omega\|} \geq \dfrac{1}{\|\omega\|}$

3) Set Lagrangian:

$$\text{Min } L_P = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{k} \alpha_i y_i (\vec{x}_i^T \vec{\omega} + b) + \sum^{k} \alpha_i$$

4) Transform into dual:

$$\frac{\partial L_P}{\partial \vec{\omega}} = 0 \qquad \frac{\partial L_P}{\partial b} = 0$$

$$\frac{\partial L_P}{\partial \vec{\omega}} = \vec{\omega} - \sum \alpha_i y_i \vec{x}_i = 0$$

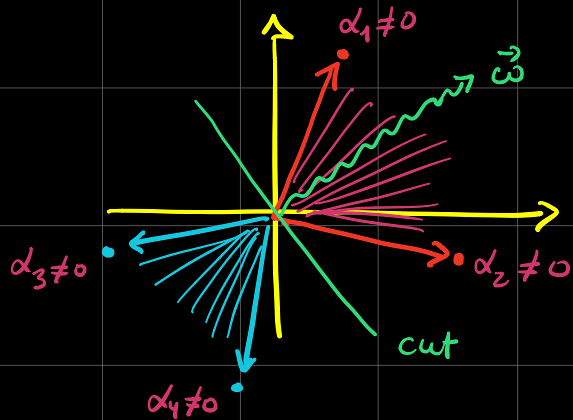
$\partial \vec{\omega}$

$$\Rightarrow \boxed{\vec{\omega} = \sum \alpha_i \vec{x}_i y_i}$$

$$\frac{\partial L_P}{\partial b} = \underbrace{\sum \alpha_i y_i}_{} = 0$$

sum of positive weights
equal in magnitude to sum
of negative weights

Visualization in the case $b=0$



$\vec{\omega}$ is a linear combination of the vectors closer to the cut.

5) Substituting into L_P

$$\text{Max} \quad L_D = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i^T x_j$$

$$\text{Max} \quad L_D = \sum \alpha_i - \frac{1}{2} \sum_i \alpha_i y_i \left(C(x_i) - \cancel{b}\right)$$

If $y_i C(x_i) > 0 \Rightarrow$ good $\quad \alpha_i = 0$

$y_i c(x_i) < 0 \Rightarrow$ bad $\quad \alpha_i > 0$

$$c(\vec{x}_i) = \sum \alpha_i y_i \vec{x}_i^{\,T} \cdot \vec{x}_j + b$$

6) The dual problem is only a function of the α's. Solve for α.

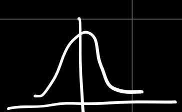
For kernels

$$c(x_i) = \sum \alpha_j y_j \underbrace{K(x_i, x_j)}_{\text{instead of inner product}} + b$$

Kernel-trick

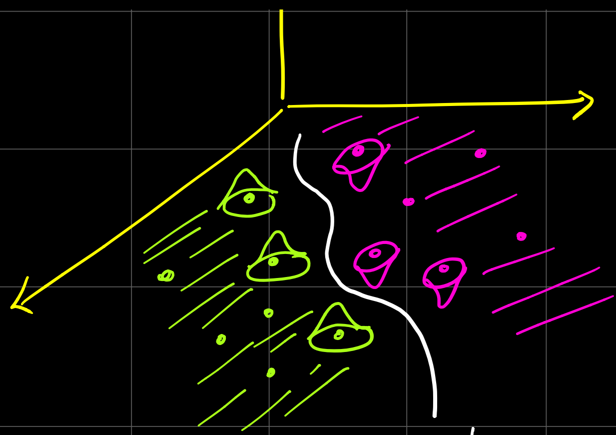
$n >> 3$

$e^{-\|x_i - x_j\|^2}$



there exists a transformation to a high dimensional space where the inner product is identical to the kernel function

Intuition



decision boundary

$$\sum \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i y_i \alpha_j y_j K(\vec{x}_i, \vec{x}_j)$$

$$\text{Max} \; \boxed{\sum \alpha_i - \frac{1}{2} \sum_{i} \alpha_i y_i \, C(x_i)}$$

we want to make the α_i's big

but increasing α_i can increase this negative term

put the most weight on points closer to the cut (wrong $C(x_i)$ or very small)

— · —

Numerics

	$\vec{x}_1$	$\vec{x}_2$	$\vec{x}_3$	...	$\vec{x}_\ell$
$\vec{x}_1$	K_{11}	K_{12}		...	$K_{1\ell}$
$\vec{x}_2$	K_{21}				

$$\vec{x}_\ell \; \begin{array}{|c|} K_{\ell 1} \; K_{\ell 2} \; \cdots \; K_{\ell \ell} \\ \hline \end{array}$$

The kernel functions are computed once

It is the generalization of the covariance matrix.

$$\Sigma = X^T X = X X^T = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_\ell \end{pmatrix} (\vec{x}_1 \; \vec{x}_2 \ldots \vec{x}_\ell)$$

The dual problem is

$$\underbrace{\text{Max} \quad \sum \alpha_i - \frac{1}{2} \vec{\alpha}^T \Sigma \vec{\alpha} \qquad \alpha_i \geq 0 \qquad \text{where} \quad \vec{\alpha} = \begin{pmatrix} \alpha_1 y_1 \\ \alpha_2 y_2 \\ \vdots \end{pmatrix}}_{\text{Quadratic with constraints}}$$

Algorithm

If constraints cannot be kept, we cap the α's

$$0 \leq \alpha_i \leq C$$

We then solve

$$\text{Max} \quad L_D = \sum \alpha_i - \frac{1}{2} \sum \alpha_i y_i \, C(\vec{x}_i)$$

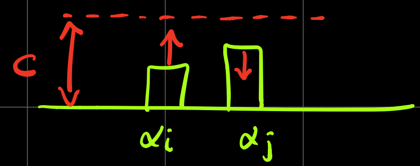
with $0 \le \alpha_i \le C$

Procedure

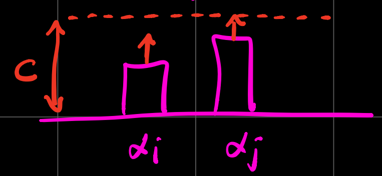
Look for a wrongly classified x_i with $\alpha_i < C$. Select another x_j

Adjust α_i and α_j

If same class

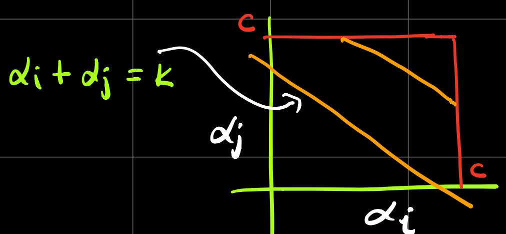


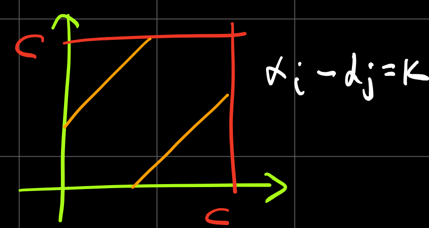
If different clas



In $\sum \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i y_i \alpha_j y_j \vec{x}_i^T \vec{x}_j$ } Eq.

all α's are fixed, except the two selected ones. Optimize the quadratic.





SMO algorithm