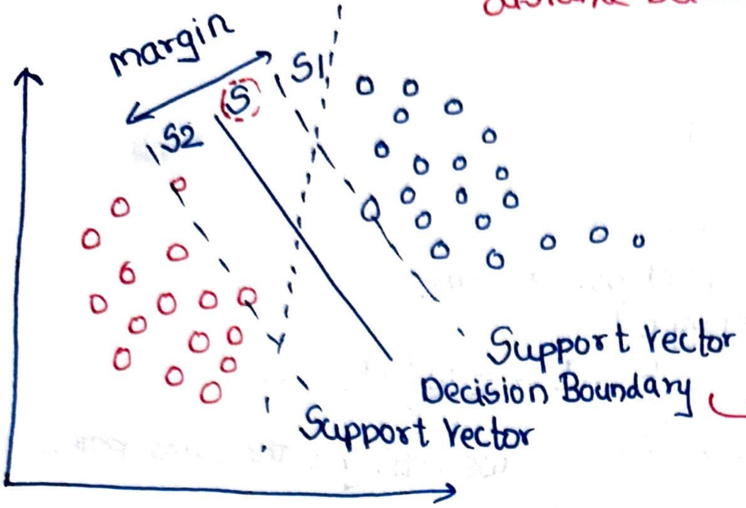
Chap 5 Svm / SVC (Support Vector Classifiers)

Inductive Bias of SVCS

S1 (band) which is more narrow
distance between Classes is smaller.



→ center of the band (most fair)

Support vector Classifiers.

binary, linear
(use line, hyperplane to separate).

S has same slope parallel lines,

Given a slope, if we look at the band nearby

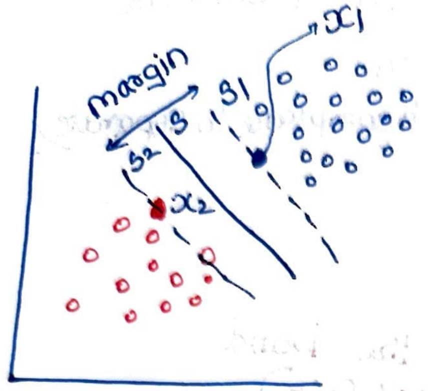
Given a slope → we look at bands → choose the one in middle.

Choose slope: (that produces) maximum margin (not narrow)

have a little chance of error (space).

2) Mini Review of Basic math.

Parallel Hyperplanes and their equations



$x w^T + b = 0$

$x_1 w^T + b = c$

↳ point not on line.

multiply by c^{-1}

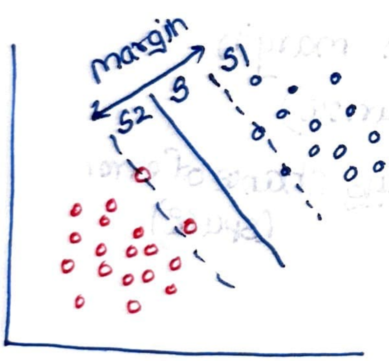
$\frac{x_1 w^T}{c} + \frac{b}{c} = 1 \rightarrow \frac{x w^T + b}{c} = 0$

we still get the same point

$x_1 w^T + b = 1$ | $x w^T + b - 1 = 0 \rightarrow S1$

$x_2 w^T + b = -1$ | $x w^T + b + 1 = 0 \rightarrow S_2$

Distance between two hyperplanes



$S_1 : X w^T + b - 1 = 0$

distance $(S_1, 0) \rightarrow$ origin $= \frac{|b-1|}{\|w\|_2}$ → intercept, → 2norm of weights

distance $(S_2, 0) = \frac{|b+1|}{\|w\|_2}$ → intercept, → norm of weights

Subtract them $d(S_1, S_2) = \frac{|b+1 - (b-1)|}{\|w\|_2} = \frac{2}{\|w\|_2}$

Inner Dot Product as a similarity measure:-

$x_1, x_2 \quad \|x_1 - x_2\|_2^2 = \underset{\text{row}}{(x_1 - x_2)} \underset{\text{column}}{(x_1 - x_2)^T}$ inner product of vector

$= \underbrace{\|x_1\|_2^2}_{\text{1st}} + \underbrace{\|x_2\|^2}_{\text{2nd}} - \underbrace{2 X_1^T X_2}_{\text{(inner product)}}$

euclidean$^2 (X_1, X_2)$

$(x_1 x_2)^T \longrightarrow$ becomes similarity
dropping -ve sign
↳ euclidean similarity

* Similarity measures related to engineered features.

good → linear regression
↳ perceptron
} non linear classification

KNN

Similarity → $K(x_1, x_2) = \exp(-\|x_1 - x_2\|_2)$

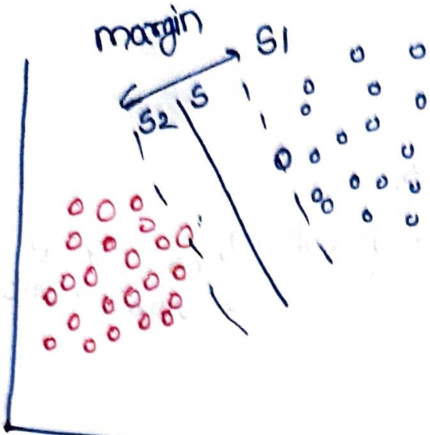
Similarity Measure

euclidean similarity

This case. $K(x_1, x_2) = \varphi(x_1) \cdot \varphi(x_2^T)$ ↳ after transformation.

↳ adds engineered features

3} The objective in SVCs



$S_1 \to x\omega^T + b = 1$

$S_2 - s\omega^T + b = -1$

$$X = [\underbrace{X_1, X_2}_{\text{dataset divided in 2 parts}}]$$

$x' \in X_1 : x'\omega^T + b \geq 1$ }

all points in Blue should be.

for red

$x' \in X_2 : x'\omega^T + b \leq -1$

n constraints: Satisfied.

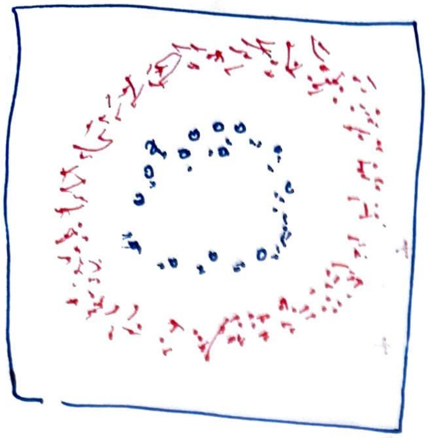
margin $= \dfrac{2}{\|\omega\|_2}$ } maximized.

minimize : $\dfrac{\|\omega\|_2}{2}$.] In general I hope I get small weights

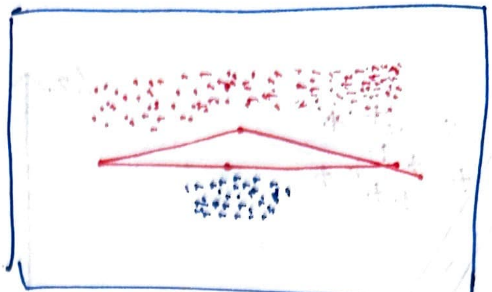
minimize weights

maximize margin.

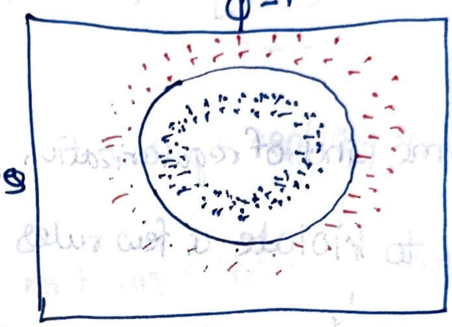
Optimization and Kernel Trick



$\xrightarrow{\phi}$



$\downarrow \phi^{-1}$



$x^1 \in X_1 \rightarrow x^1 w^T + b \geq 1$
$x^2 \in X_2 \rightarrow x^1 w^T + b \leq -1$] Linear inequalities

min $\|w\|_2$] quadratic function

Quadratic Programming } not so efficient

Kernel Matrix

$$\underline{n \times n} \quad Q_{i,j} = X_i \cdot X_j^T \rightarrow \text{same label.}$$
$$-X_i \cdot X_j^T \rightarrow \text{different label}$$

} even less efficient

kernel matrix

minimize $v^T Q v$

$\downarrow$ v (some vector)

→ Solve problem based on similarities (inner product).

example

$x_1 = +1 \quad x_2 = +1$

Similarity { $K(x_1, x_2) \sim 1$

$x_1 = +1 \quad x_2 = -1$

$K(x_1, x_2) \sim 0$

Kernel Trick.

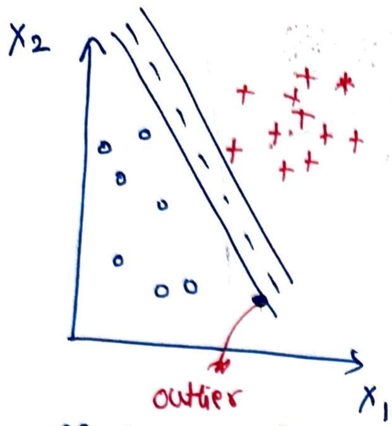
I can find my SVC → euclidean similary

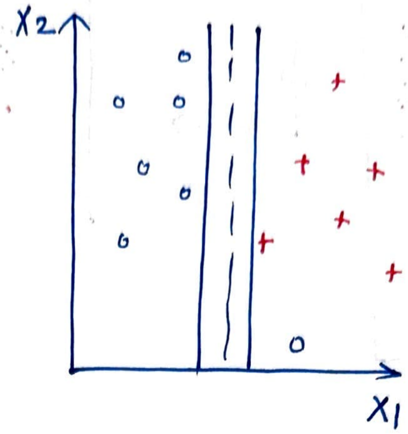
redefine the definition of similarity

SVC classifier becoms → $K_{i,j} = \exp(-\|X_i - X_j\|_2)$

non linear

Slacks





(forcing margin)

not linearly separable → SVC fail

not a good thing

introduce some kind of regularization

Sensitivity

allow points to violate a few rules

introduce slacks

TOTAL SLACK

$\xi \ \sum_{i=1}^{n} \xi_i$

$\min \|w_2\|^2 + C \cdot \xi$

if C is small give more slack

if C big don't give too much slack.

strength of regularization

X: X_{pos}, X_{neg}

$X_1 \in X_{pos}: X_1 w^T + b \geq 1 - \xi_i$

introduce ξ_i for each X_i

$\xi_i \geq 0$

+ve number

so you don't force it on one side

you allow it to be on wrong side

ξ_i for each $X_i \in X_{neg}$

$X_i \in X_{neg}: Y_i w^T + b \leq -1 + \xi_i$

if ξ_i is greater than 1 point is allowed to even be on other side, so its fine

thats how much slack we give.

Hinge Loss

non differentiable because of other variations.

called hinge loss ξ_i

$$\min \|w\|_2^2 + C\left(\sum_{i=1}^{n} \max\left(0, 1 - \text{sign}(y_i - 0.5)(x w^T + b)\right)\right)$$

$\cancel{\min \|w\|_2^2 + C\left(\sum_{i=1}^{n} \max\right.}$

equivalent

$$\min \|w\|_2^2 + \underbrace{C \sum_i \xi_i \, (\xi_i \geq 0)]}_{\text{Loss function}} \text{ loss}.$$

can be replaced by

when we have Slacks, we are not just having regularization we are also doing gradient descent.

when we have linear kernel.

Faster linear SVC

Hinge Loss

non differentiable because of other variations.

called hinge loss

$$\min \|\omega\|_2^2 + C\left(\sum_{i=1}^{n} \max\left(0, 1 - \text{sign}(y_i - 0.5)\overbrace{\left(x(\omega^T + b)\right)}^{\xi_i}\right)\right)$$

equivalent

$$\cancel{\min \|\omega\|_2^2 + C\left(\sum_{i=1}^{n} \max\right.}$$

$$\min \|\omega\|_2^2 + \underbrace{C \sum_i \xi_i \, (\xi_i \geq 0)}_{\text{Loss function}}] \text{ loss.}$$

Loss function → can be replaced by

when we have Slacks, we are not just having regularization we are also doing gradient descent.
→ when we have linear kernel.

faster linear SVC