Dr Osita Onyejekwe

Part II

Support Vector Machines

Using Lagrangian Optimization

04/17/2024

Original Optimization Problem:

$$\min_{w} \underbrace{\tfrac{1}{2}\|w\|^2}_{\text{loss function } \mathcal{L}(w)} \quad s.t \quad \underbrace{y^i(x^{iT}\vec{w} + b) - 1 \geq 0}_{f_i(w) \text{ estimate}} \quad \forall i \in \{1, \dots, n\}$$

constraints

- i.e We want the function $f_i(w) \geq 0$

We want to correctly classify each point i, in other words, we want to put it on the right side of the hyperplane.

- $\Rightarrow$ $f_i(\vec{w}, b) \geq 0$

$\vec{w}$: w-vector, b: scalar b $\Rightarrow$ $\tilde{w} = \begin{bmatrix} b \\ \vec{w} \end{bmatrix}$

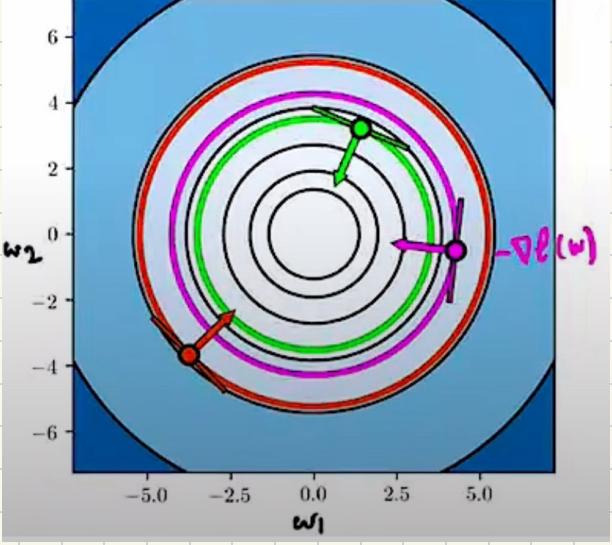
∴ We have many constraints. We have a constraint for each data point "i".

∴ The problem can be formulized in this way:

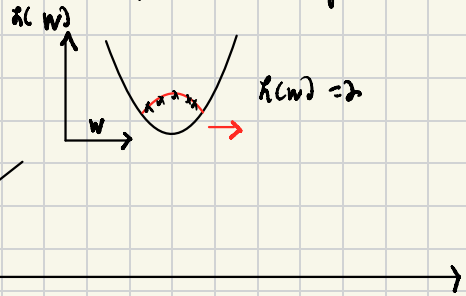
$$\min_{w} \underbrace{\mathcal{L}(w)}_{\text{loss function}} \text{ over the space of parameters } w, \quad w \in \mathbb{R}^2, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$s.t \quad f(w) \geq C = 0 \leftarrow \text{This condition is fulfilled}$$

General formulization of a constrained optimization Problem

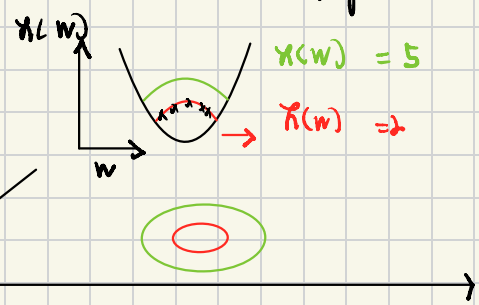


- Very Challenging Problem to solve mathematically.
- Gradient Decent : few properties about the gradient direction.
- When we plot the contours of our Loss function.

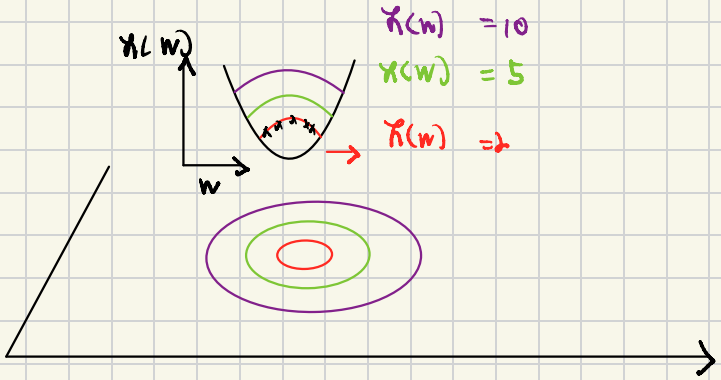


We defined the contour or the level contour of the loss function where the Loss function at each contour, & when we get the $\vec{w}$ we know that all those points in the loss function = constant, &

$L(w) = 2$, and then we plotted it. Imagine this is a 3-D space, and then we plotted it.

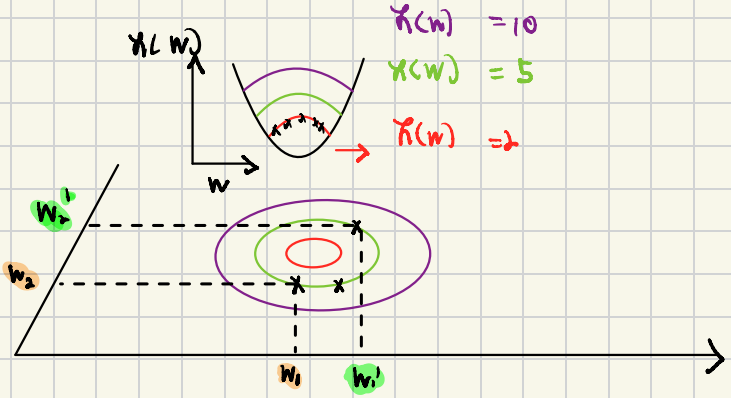


- We find the second contour, and this is where $L(w)$ takes a bigger value. and the L is actually constants along the contour.
-



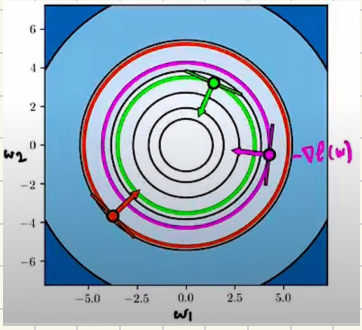
Then, we have another one, $R(w) = 10$

∴ If we select a random point:



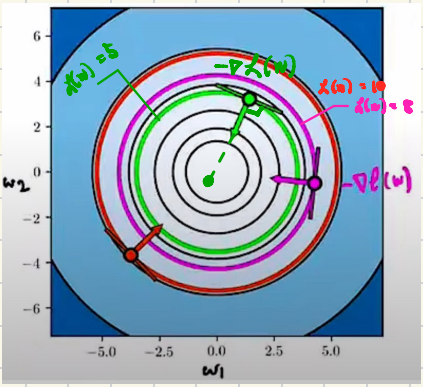
We know that all points here, that satisfy this, are all equal to 5 because they all lie on the green contour.

① In gradient Descent, the gradient at each point that lies on the contour, the gradient (negative of it) is perpendicular to the contours.



See Image

① We also want to move along the negative direction of the gradients towards our optimum. Because this is a loss function, therefore the values in the framing contours are decreasing as we go closer to the center.



Remember our goal is to minimize our loss therefore we are slowly jumping from one contour to another using gradient descent

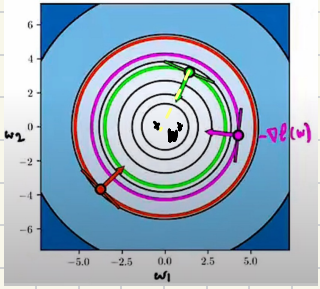
∴ What we want to do is to minimize this:

$$\min_{w} \; h(w), \quad w \in \mathbb{R}^2, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

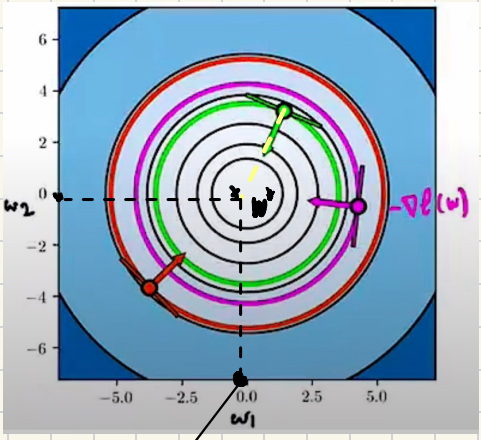
$$s.t. \; f(w) \geq c$$

constraint

∴ If we remove the constraint, and we have a general loss function that we want to minimize, we will just keep moving along these contours until we find an optimum for w



original contours of $L(w)$



Here, it is located at (0,0) ($w_1 = 0$, $w_2 = 0$) ∃ w is a $\neq 0$ vector $\Rightarrow w \in \mathbb{R}^2$, however, we have an additional condition

① If we solve this optimization problem it also boils down to solving this condition:

$$\min_{w} \; h(w), \quad w \in \mathbb{R}^2, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\text{s.t} \quad f(w) \geq c$$

$$\text{s.t} \quad f(w) = c \quad (c \text{ constant})$$

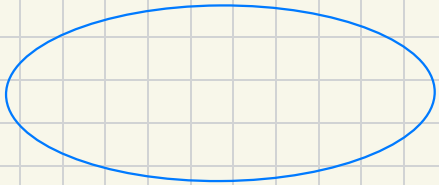
can be solved by setting the condition to a constant value c.

→ This is exactly a contour line of the function

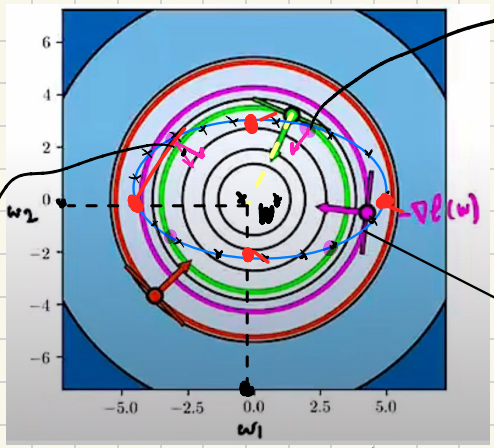
$\Rightarrow$ The function is constant.

- We are trying to find all w's, such that $f(w) = c$

∴ Let's say that this is the contour, s.t $f(w) = c$.



and let's say it lies exactly here.



→ Not drawn to scale but assume that $f(w)$ intersects those contours of h at those points. and we want both curves/contours to be tangent. We want the "w" to lie on both contours

→ $f(w) = c$

→ not interested in other intersection points because of the direction of the gradient they move. ↓ the most tangent curves.

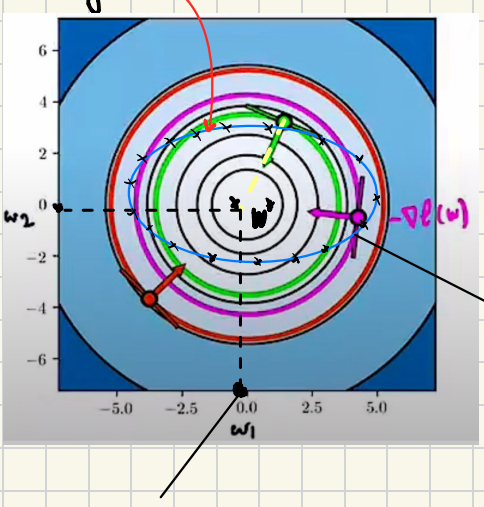
∴ On this contour, we have that $f(w) = c$.

∴ We are looking for the vector $\vec{w}$ (the set of parameters) s.t we are minimizing L^k and also all points should lie on the contour.

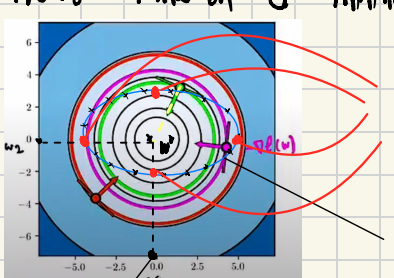
- We want to find all possible solutions to this constraint optimization problem such that the contours of the Loss function $h(w)$ are tangent to the contours of $f(w)$.

- Imagine that the contour f takes on so many values

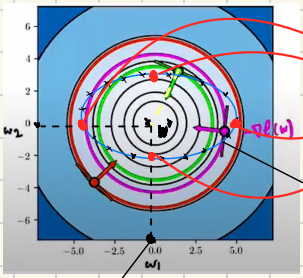
, but we are only interested in particular with the value c. because that is our condition, so we want to find our optimum w on this contour specifically.



(2) We also want to lie on a minimal contour of h.
∴ we know that our w should minimize the value of our Loss function ∴ it should lie on a lower contour of h whereby $h(w)$ takes on a minimum value.

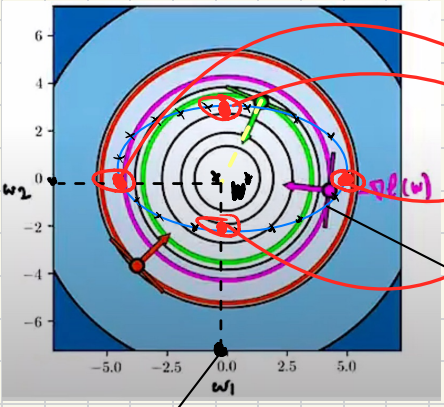


We want these key points



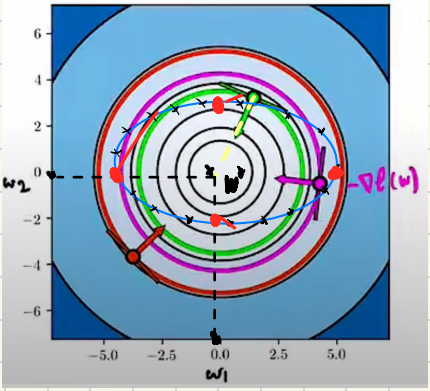
key points.

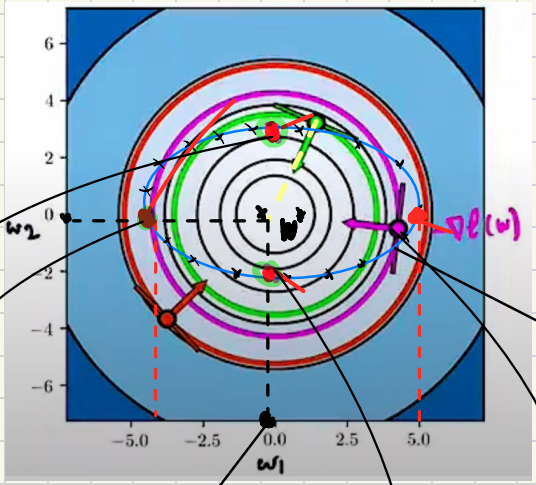
- We want to find out where the contours for $f(w)$ intersect the contours of h
- We want both the contours to be tangent.



Here we have 4 candidate points.

If we plot them, we know their coordinates, but how to find out which one is the best.





$f(w) = c$

For this one, I know that $f(w) = c$, however for h, I know that the point lies on the purple constraint $\Rightarrow h = 8$

$h(w) = 8$

$h(w) = 2$

$h(w) = 2$

$\Downarrow$

Best candidate point here.

$\Rightarrow$ Looking at those 4 points, we know that the bottom one is the one that supports $h(w) = 2$.

$\Rightarrow$ Those are our best candidates.

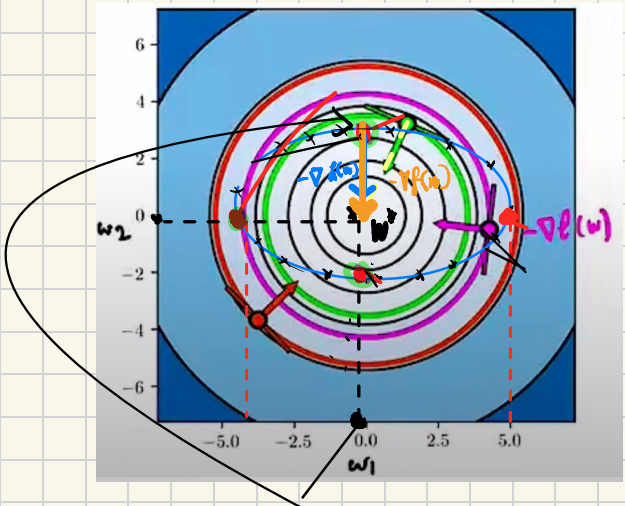
At optimal point w^* we should have the negative of the gradient of the loss function

loss function — constraint

$$-\nabla h(w) = \alpha \nabla f(w) - c$$

colinear

- Notice that the negative of the gradient of the loss function $-\nabla h(w)$ and if we calculate the negative of the gradient of the w-function $-f(w)$, we notice that the 2 vectors are collinear and they point towards the same direction because the curves are tangent. So both of these vectors/gradients should be perpendicular to the contour of the loss function and the constraint.



We also want the 2 gradients to be colinear.

- We know at optimal point, w^* !

$$-\nabla h(w) = -\alpha \nabla f(w) - c$$

(colinear) same direction

f-function

The constraint itself

c doesn't matter because it is a constant

We can remove the negative

$$\therefore \nabla h(w) = \alpha \nabla f(w)$$

∴ The Condition that the optimal point should satisfy is thus given by:

$$\nabla \mathcal{L}(w) = \alpha \nabla f(w) \qquad \alpha \geq 0$$

The most important thing to get out of this is that the gradients should be colinear and that $\alpha \geq 0$ because the 2 vectors should point to the same direction

Let's now define what we call the Lagrangian function $\mathcal{L}(w)$ as:

We want $f(w) - C = 0$ ≠ because our "C" (in this case C=0)

$\underbrace{y_i(x_i^T w + b) - 1 \geq 0}_{\text{constraints}}$ //

$\mathcal{L}(w) = \mathcal{L} - \alpha[f(w) - C]$, we could get rid of the "C" because it is equal to 0, however let's just leave it like that for now.

∴ Writing it more formally we have that:

$$\mathcal{L}(w) = \mathcal{L}(w) - \alpha[f(w) - C]$$

$\mathcal{L}(w)$: Lagrangian of w; $\mathcal{L}(w)$: Loss; Lagrangian function

∴ If we take the gradient of the Lagrangian with respect to w, what do we find?

$$\nabla \mathcal{L} = \nabla \mathcal{L}(w) - \alpha \nabla f(w) - \nabla C$$

constant so ϕ

∴

$$\nabla \mathcal{L} = \nabla \mathcal{L}(w) - \alpha \nabla f(w) \quad *$$

some books write: $\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$

$$\nabla \mathcal{L} = \nabla R(w) - \alpha \nabla f(w)$$

Remember that $\nabla R(w) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial w_1} \\ \frac{\partial \mathcal{L}}{\partial w_2} \end{pmatrix}$ We have 2 components for w here.

$\in \mathbb{R}^d$, because our vector $\vec{w}$ belongs to $\mathbb{R}^d$ ($\vec{w} \in \mathbb{R}^d$) also, our samples belong to $\mathbb{R}^d$ ($x \in \mathbb{R}^d$)

- We know that the gradient of the lagrangian $= 0$ [$\nabla \mathcal{L} = 0$].

∴ This is an important condition that will help us solve this problem.

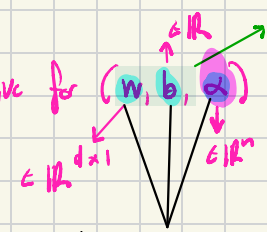
① Original Optimization Problem: (Primal Optimization Problem):

$$\min_w \tfrac{1}{2} \|w\|^2 \quad \text{s.t} \quad \underbrace{y^i(x^{iT} w + b) - 1 \geq 0}_{\text{constraints}} \qquad \forall i \in \{1, \dots, m\}$$

② Lagrangian: $\mathcal{L}$ = cost function to optimize $- \alpha \times$ [constraint]

α: Lagrangian multiplier per constraint, $\alpha \geq 0$ (particular)

③ Solve for (w, b, α) using:



We need to solve for these parameters, and to do that we need to solve the following set of linear Equations using linear Algebra tools.

$$\begin{cases} \nabla \mathcal{L} = 0 \\ + \\ \text{all constraints} \end{cases}$$

Cost gradient $= 0$ & compute the gradient of the lagrangian)
We also need to use all the constraints!!

We know our parameters w & b satisfy particular constraints

$$\nabla \mathcal{L} = \nabla R(w) - \alpha \nabla f(w) = 0$$

By adding the constraints we can find our optimal W^* & b^*

$$\therefore \quad \min_w \tfrac{1}{2}\|w\|^2 \quad \text{s.t} \quad \underbrace{y^i(x^{iT}w + b) - 1 \geq 0}_{\text{constraints}} \qquad \forall i \in \{1, \dots, m\}$$

$\mathcal{L}$ = cost function to optimize $- \alpha \times$ [constraint]

$\mathcal{L}$ agrangian multiplier, $\alpha \geq 0$, per constraint

$$\mathcal{L} = \underbrace{\tfrac{1}{2}\|w\|^2}_{\text{Loss function } L(w)} - \sum_{i=1}^{n} \alpha_i \left[y^i(x^{iT}w + b) - 1 \right]$$

summing over all our constraints.

① Compute gradient of $\mathcal{L}$ wrt w:

$$\frac{\delta \mathcal{L}}{\delta w} = -\sum_{i=1}^{n} \in \mathbb{R}^?$$

partial derivative of Regularization with respect to w.

Hints:

$$\frac{\delta x^T w}{\delta w} = x$$

$$w^T w = \|w\|^2$$

$$\therefore \quad \frac{\delta w^T w}{\delta w} = 2w$$

② Set gradient $\frac{\delta \mathcal{L}}{\delta w} = 0$

$$\frac{\delta \mathcal{L}}{\delta w} = 0 \Rightarrow w = \; \leftarrow \text{Optimal parameter.}$$

Our Goal is to solve: $\min_{\vec{w}, b} \max_{\vec{\alpha} \geq 0} L(\vec{w}, \vec{\alpha})$

Side Note : Dual SVM Derivation : The Linearly Separable Case :

(Primal) $\min_{\vec{w},b} \max_{\vec{\alpha}\geq 0} \frac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i[(\vec{w}\cdot\vec{x}_i + b)y_i - 1]$ (1)

Swap min and max

(Dual) $\max_{\vec{\alpha}\geq 0} \min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 - \sum_j \alpha_j[(\vec{w}\cdot\vec{x}_j + b)y_j - 1]$ (2)

Note that switching the $\vec{w}$ & $\vec{x}$ order would/could result in putting the transpose on the $x \Rightarrow (\vec{x}_j^T \vec{w}_j + b)y_i$

Slater's Condition from convex optimization guarantees that these two optimization problems are equivalent!

$\therefore \quad \frac{\partial \mathcal{L}}{\partial w} = \frac{1}{2}\,dw - \sum_{i=1}^{\hat{}} \alpha_i y_i x^i \in \mathbb{R}^d = 0$ *

set to 0

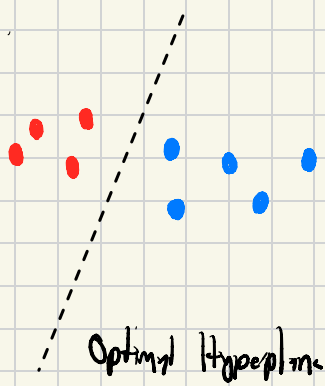
$\Rightarrow \quad w^* = \sum_{i=1}^{\hat{}} \alpha_i y_i x^i$ (label, training sample)

↓

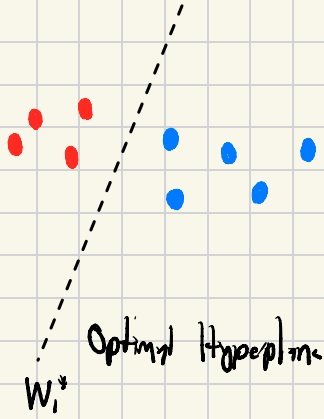
optimal w which is important because w defines the margin and hence the hyperplane ∴ it carries all the information we need to carry our Linear Classification.

∴ w depends on our training samples x^i & their labels y_i.

Eg :

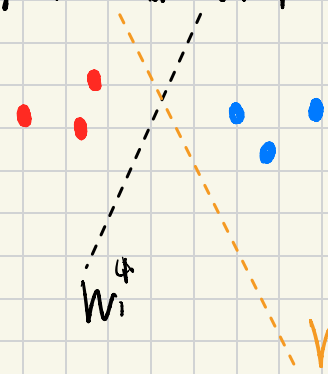


Optimal Hyperplane that Seperates them (w_1^*)

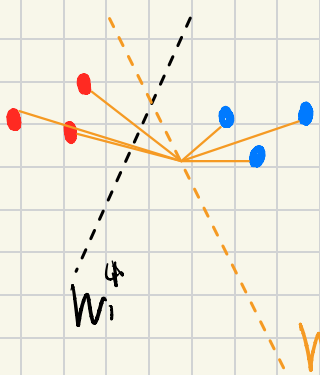


Optimal hyperplane that separates them (w_1^*)

∴ If we remove some points, you can see that the optimal w is completely different.



This is very important because we know that w_2^* depends on the training samples.



* this is visible through this equation:

$$w^* = \sum_{i=1} \alpha_i y_i x^i$$

training samples

So now, let us try to compute the "b"

$$\mathcal{L} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left[y^i (x^{iT} w + b) - 1 \right]$$

③ Compute gradient of $\mathcal{L}$ w.r.t to b:

$$\frac{\delta \mathcal{L}}{\delta b} = 0 - \sum_{i=1}^{n} \alpha_i y_i = 0$$

Note that $\frac{d\alpha x}{dx} = \alpha$

④ Set gradient $\frac{\delta \mathcal{L}}{\delta b} = 0$

$$\frac{\delta \mathcal{L}}{\delta b} = 0 \Rightarrow b \Rightarrow \boxed{\sum_{i=1}^{n} \alpha_i y_i = 0}$$

α labels

sum over all the training points

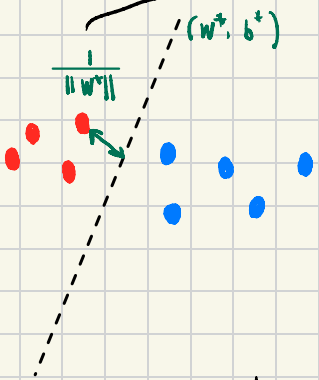
This is a new constraint that we have found that

We can use to find the optimal w^* & b

Remember that we need to add any new constraints that pop up along the way o.t.

$$\begin{cases} \nabla \mathcal{L} = 0 \\ + \\ \text{all constraints} \end{cases} \longrightarrow \begin{matrix} w^* \\ b^* \end{matrix}$$

What do you notice? → position of the hyperplane } Depend on the training points $y_i \neq 0$

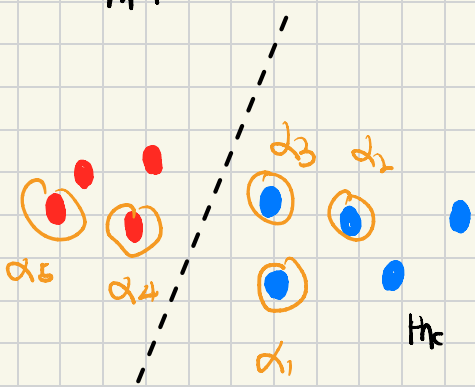


margin

$$w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \rightarrow \text{Explicit form}$$

$$b^* \Rightarrow \sum \alpha_i y_i = 0$$

- W can be seen as a linear combination of all our training samples and the α is the weights. (Weighted sum of all our training samples.
- Not all of them will have the same weight/alpha.



- We know that all these alphas are positive, but we know that they might also take on different values.

∴ W depends on the samples but the samples do not have the same weights

⇒ They contribute differently ∴ samples contribute differently to finding w^* ∴ after finding the optimal parameters:

$$\therefore L(w^*, b^*) = \frac{1}{2} \underbrace{\|w\|^2}_{w^T w} - \sum_{i=1}^{n} \alpha_i [y^i (x^{iT} w + b) - 1]$$

→ plug w^*

$L(w, b)$ = We now need to compute the

Lagrangian : Goal is to work with our PRIMAL

$$\min_{w} \mathcal{L}(w) \quad \text{s.t} \quad f(w) \geq c$$

$\Rightarrow$ Need to max the Dual Problem.

$$\max_{\alpha} \mathcal{L}(w^*, \alpha)$$ We want the value that the Lagrangian takes at the optimal w to be max by finding the best set of α's.

(Primal) $$\min_{\vec{w}, b} \max_{\vec{\alpha} \geq 0} \tfrac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i [(\vec{w} \cdot \vec{x}_i + b) y_i - 1] \quad (1)$$

Swap min and max

(Dual) $$\max_{\vec{\alpha} \geq 0} \min_{\vec{w}, b} \tfrac{1}{2}\|\vec{w}\|^2 - \sum_j \alpha_j [(\vec{w} \cdot \vec{x}_j + b) y_j - 1] \quad (2)$$

α → Finding the best set of Lagrangian Multipliers. So The

first step will be to compute the Lagrangian at the optimum & then we try to find the maximum of the Lagrangian with respect to the α

So $$\mathcal{L}(w^*, b^*) = \tfrac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{iT}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^j\right) -$$

Squared norm therefore we need to change the index:

$$\left(\sum_{i=1}^{n} \alpha_i y^i x^{iT}\right) \sum_{j=1}^{n} \alpha_j y_i x^j \Big) - \sum_{i=1}^{n} \alpha_i y_i b$$

$$+ \sum_{i=1}^{n} \alpha_i$$

So We know that the dim $= 0$ & "b" is a constant so we can put it outside the summation.

since it also does not depend on the index.

Recall that: $\sum \alpha_i y_i = 0$ so that entire term goes away.

$$\mathcal{L}(w^*, b) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{iT}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^j\right) -$$

squared norm therefore we need to change the index:

$$\left(\sum_{i=1}^{n} \alpha_i y^i x^{iT}\right)\sum_{j=1}^{n} \alpha_j y_i x^j\Big) - \sum_{i=1}^{n} \alpha_i y_i b$$

$$+ \sum_{i=1}^{n} \alpha_i$$

and also, those two terms are equal,

$$\Rightarrow \mathcal{L}(w^*, b^*) = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{iT}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^j\right) -$$

$$\left(\sum_{i=1}^{n} \alpha_i y_i x^{iT}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^j\right) - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i$$

$$\therefore \mathcal{L}(w^*, b^*) = -\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{iT}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^j\right) + \sum_{i=1}^{n} \alpha_i$$

[Ex: $\frac{1}{2}x - x = -\frac{1}{2}x$]

$$\therefore \mathcal{L}(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j x^{iT} x^j$$

Labels · Dot product between 2 training samples

↳ Explicit formula for Lagrangian evaluated at optimal parameters

We now want

$$\max_{\alpha_i} \; \mathcal{L}(w^*, b^*, \alpha_i) \quad \text{s.t.} \quad w^* = \sum \alpha_i y_i x^i$$

(to find)

We have not explicitly computed it,

Lagrangian dependent

α_i : We want to maximize it and find the best set of parameters α_i, that will enable us to compute the w because it depends on these α_i's.

$$\therefore w^* = \sum \alpha_i y_i x^i$$

[Need to maximize the Lagrangian and then it will enable us to find α_i]

$\therefore$ Looking at the Lagrangian at the optimal point:

$$\mathcal{L}(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x^{i^T} x^j$$

What does it depend on?

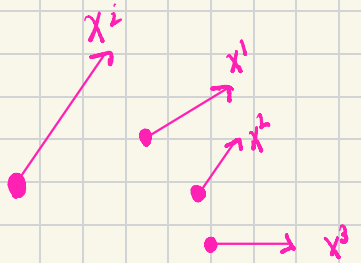
[=> It depends on the training point and dot product between two training samples.

Dot product => Projecting one sample onto another enables us to find the optimal parameters.

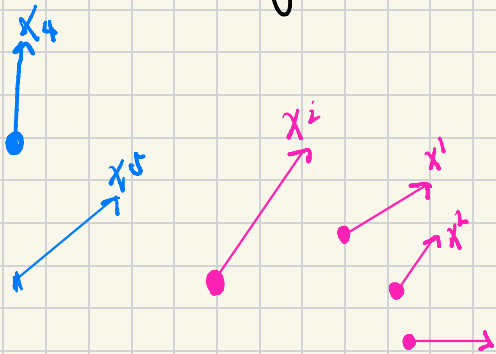
:o Modeling the pairwise relationship between 2 samples is very important

Eg:

This is our sample x^i, and the other samples



And we have samples belonging to the other class.



We have a double summation, so this means that

$$\mathcal{L}(w^*, b^*) = -\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{i\,T}\right)\left(\sum_{i=1}^{n} \alpha_i y_i x^i\right) + \sum_{i=1}^{n} \alpha_i$$

$$\mathcal{L}(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i} \sum_{i} \alpha_i \alpha_j y_i y_j x^{i\,T} x^j$$

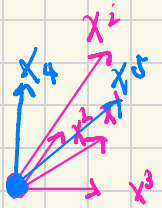
We are merging the double summations.

$$\mathcal{L}(w^*, b^*) = -\tfrac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x^{i\top}\right)\left(\sum_{j=1}^{n} \alpha_j y_j x^{j}\right) + \sum_{i=1}^{n} \alpha_i$$

merge them together.

$$\mathcal{L}(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i} \sum_{j} \alpha_i \alpha_j y_i y_j x^{i\top} x^{j}$$

∴ Considering the pairwise projection between all possible pairs of training samples.



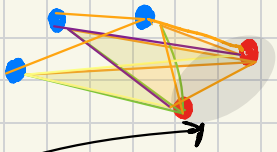
∴ x^i, projecting all other samples onto sample x^i and summing all those projections by the weighted sum of all those projections for each point

$$\mathcal{L}(w^*, b^*) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \underbrace{\sum_{i} \sum_{j}}_{\text{sum}} \underbrace{\alpha_i \alpha_j y_i y_j x^{i\top} x^{j}}_{\text{weight}}$$

The dot product tells us how two vectors are related. If they are close/ far away/ similar magnitude/ opp/ same direction... etc. ∴ the dot product between 2 vectors $x^i \cdot x^j$ models the relationship/ captures the relationship between the features of 2 training points.

∴ $x^{i\top} \cdot x^{j} = x^i_1 x^j_1 + x^i_2 x^j_2$ $\in \mathbb{R}^2$

feature feature feature feature

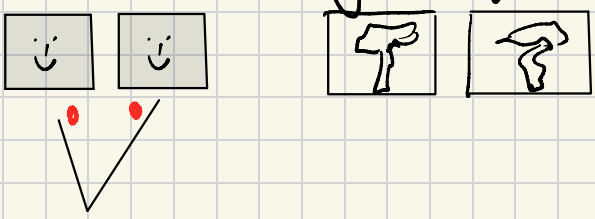


} comparing the pairwise relationship between all the samples

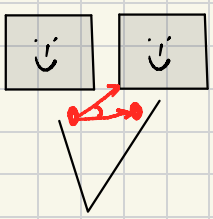
If they are too close, their dot product will tell us how close they are.

⇒ a pair that might be very similar,

for example classifying images of trees and faces



These faces will be similar, so I will expect the dot product between them (2 vectors) to be quite large ⇒ They are similar. ⇒



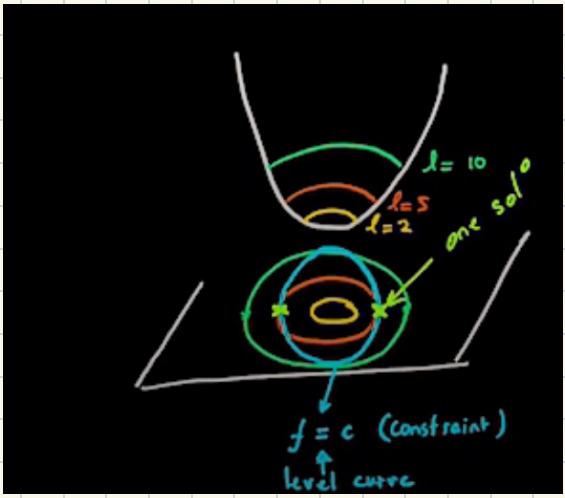
These 2 points shall also be similar.

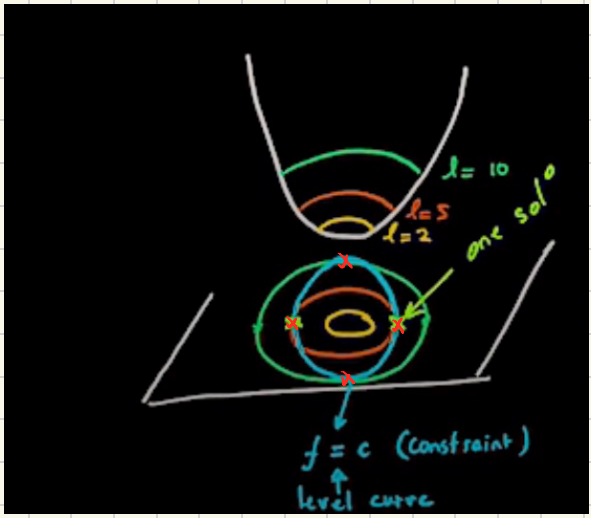
⇒ What happens to $\mathcal{L}$ at the optimum (w^*, b^*) ?

Remember our goal is to find the α.

$$\Rightarrow \quad \mathcal{L}(w, b) = d(w, b) - \sum_{i=1}^{n} \alpha_i f_i(w, b)$$

$$\mathcal{L}(w^*, b^*) = ?$$





We have different solutions and not all of them are right, so to find the optimal solution, we need to also get the α's. That will help us find the optimal one because the primal problem has different solutions so by optimizing over the α, we will refine our first solution.

$$\mathcal{L}(w^*, b^\alpha) = ?$$

We will find the right condition to refine these solutions.

$$\max_{\alpha_i} \mathcal{L}(w^*, b^*) = \underbrace{\mathcal{L}(w^*, b^\alpha)}_{\min_{w,b}} - \sum_{i=1}^{n} \alpha_i \underbrace{f_i(w^*, b^\alpha)}_{\text{strictly positive}}$$

$\alpha_i \geq 0$

solve the primal problem

positive

$\therefore$ We know that (w^*, b^α) here minimize our loss function

$\therefore$

10

$\max_{\alpha_i} \mathcal{L}(w^*, b^*) = \mathcal{L}(w^*, b^*) - \sum_{i=1}^{n} \alpha_i f_i(w^*, b^*)$

$\mathcal{L}(w^*, b^*)$: solve the primal problem; min b

$\alpha_i \geq 0$ → over all α_i

α_i: positive; $f_i(w^*, b^*)$: strictly positive

↓↓

knowing that this is all positive,

What is the maximum that the Lagrangian at the optimum that we computed earlier can take?

$\max_{\alpha_i} \mathcal{L}(w^*, b^*) = \underbrace{\mathcal{L}(w^*, b^*)}_{\text{minimum value} \geq 0} - \sum_{i=1}^{n} \underbrace{\alpha_i}_{\geq 0} \underbrace{f_i(w^*, b^*)}_{\geq 0}$

$\mathcal{L}(w^*, b^*) = \frac{1}{2}\|w\|^2$

$y_i(x_i^T w + b) - 1 \geq 0$

≥ 0

(A) term

- Subtracting something positive from the L quantity will always decrease it.
- To maximize the Lagrangian over the α, we ideally want to set this term to 0.

$\sum_{i=1}^{n} \underbrace{\alpha_i}_{\geq 0} \underbrace{f_i(w^*, b^*)}_{\geq 0}$ (A) term ≥ 0

$y_i(x_i^T w + b) - 1 \geq 0$

Set this term to zero.

- The maximum value that it will take is going to be

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

$L(w^*, b^*)$

minimum value

≥ 0

$\Rightarrow$ Since we are multiplying two positive quantities α_i & f_i, if we set them to zero, one of them should be equal to zero or both equal to zeros.

We can't have them both $=0$. Not possible in our case.

Since $\begin{cases} \alpha_i \geq 0 \\ f_i \geq 0 \end{cases} \Rightarrow A = 0 \begin{cases} \alpha_i = ? \ 0 \\ \text{or} \\ f_i(w^*, b^*) = ? \ 0 \end{cases}$

$\Downarrow$

constraint at the optimal point.

We know that those conditions should always be satisfied (always positive)

$\Rightarrow$ for α_i strictly positive

$\Rightarrow$ for $\alpha_i > 0$ & $f_i(w^*, b^*) = 0$

$\Rightarrow$ $A = 0 \rightarrow \alpha_i = 0$ or $f_i(w^*, b^*) = 0$, $\alpha_i > 0$

$A = 0$ → $\alpha_i = 0$ ⇒ $f_i > 0$ (2)

or

$f_i(w^*, b^*) = 0$, $\alpha_i > 0$

⇓

for $\alpha_i > 0$ ⇒ $f_i(w^*, b^*) = 0$ (1)

⇒ $\underline{w}^* = \sum_{i=1}^{n} \alpha_i y_i x^i$ ⇒ Recall

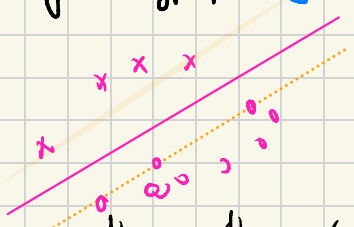
$y^i(x^{iT} w + b) - 1 \geq 0$

$\forall i \in \{1, \ldots, m\}$

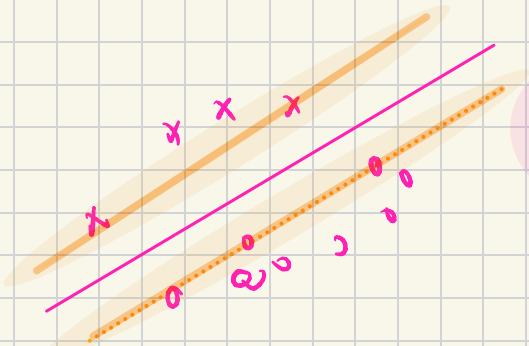
constraints

- $= 0$ ⇒ The points should lie on the hyperplane

We know that points lie on the marginal hyperplane they satisfy that inequality by definition



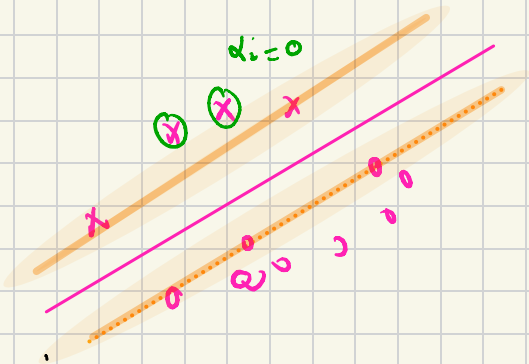
- If they are greater than zero then they are correctly classified By greater we mean strictly greater than Zero.
- This condition states that those training points x^i should lie on the marginal plane.
- for points that lie on the marginal hyperplane the α_i's should be strictly positive



$\alpha_i > 0$ if $f_i(w^*, b^*) = 0$

① condition 1

and for the others, when they are correctly classified, We know that the $\alpha_i = 0$



i.e all other points that do not lie on the marginal plane, they have an $\alpha_i = 0$.

Again, looking at our optimal w^* which depends on the α_i

$$\underline{w}^* = \sum_{i=1}^{n} \alpha_i y_i \underline{x}^i$$

we know that if α_i is strictly positive ($\alpha_i > 0$), then the point x^i lies on the marginal hyperplane and we know that if $\alpha_i = 0$, then x^i is not marginal

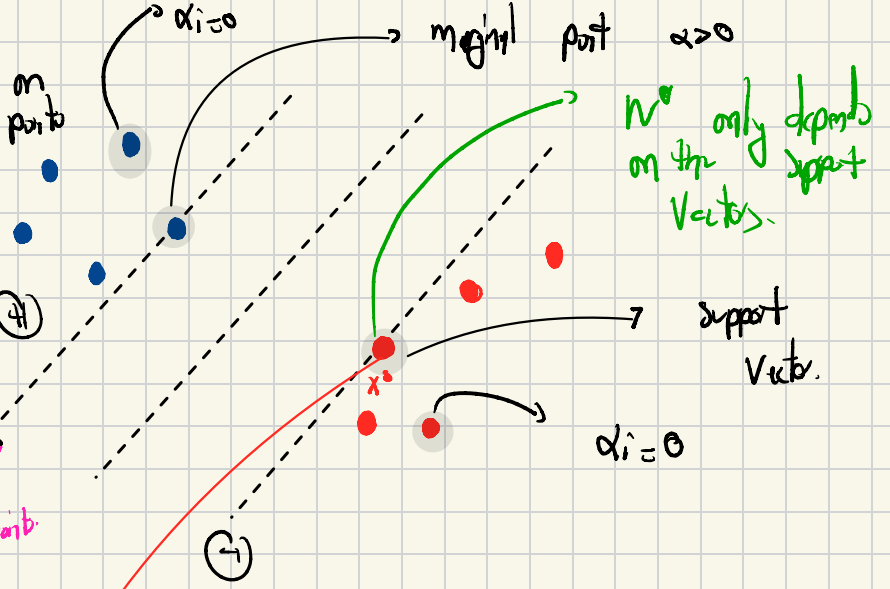
w^* depends only on the strictly positive points because

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

$\Downarrow$

$$w^* = \sum_{s} \alpha_s y_s x^s$$

s: margin points. support points.



∴ Now if we want to find the b^*, look at the equality that the point satisfies (support points)

Estimate b^* given $w^* = \sum_{i=1}^{n} \alpha_i y_i x^i$ using margin points.

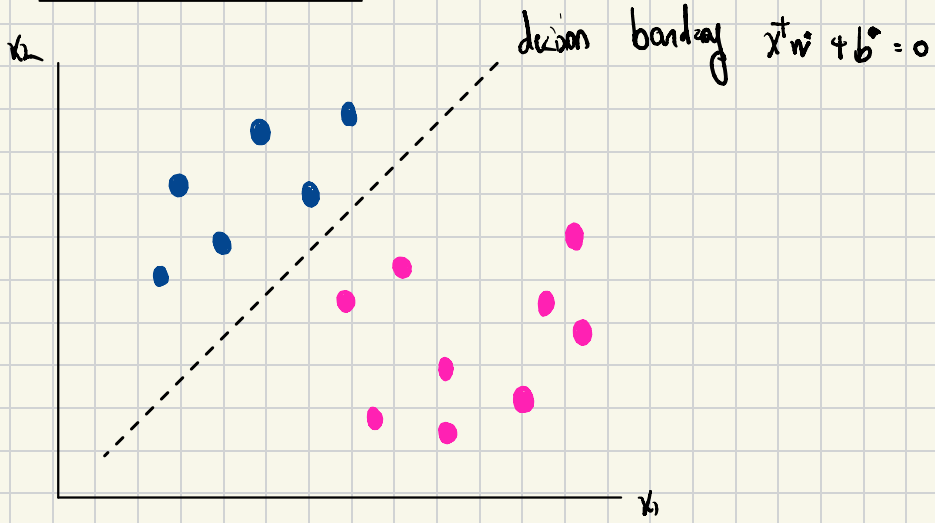
$$\Rightarrow x^{T} w^* + b^* = y \Rightarrow \boxed{b^* = y - x^T \cdot w^*}$$

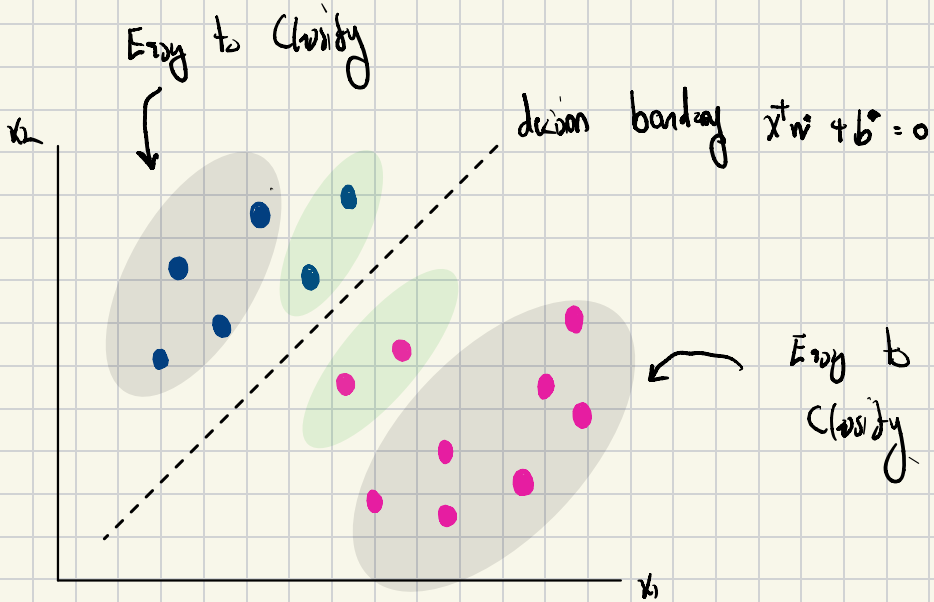
A support vector example

Dot product between the support vector sample and the optimal w^*.

$$\therefore w^* = \sum_{s} \alpha_s y_s x^s$$

This is intuitive.

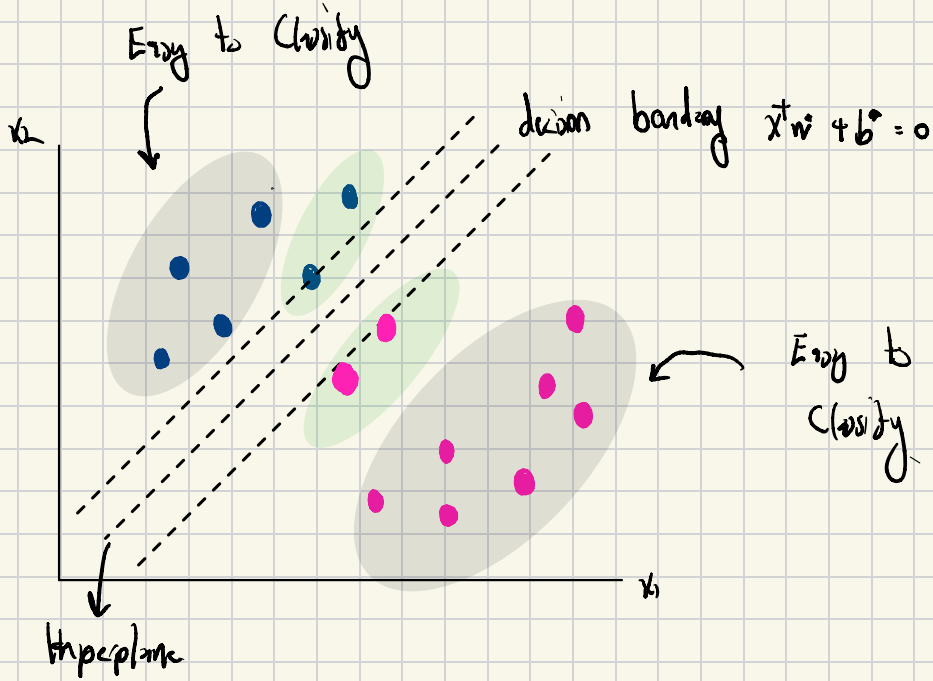




The most difficult ones are the pts close to the boundary.

"If you identify the marginal pts/ pts close to the margin, then you have solved the problem because you now know

that those vectors/ points define the margin.



We are then looking for the extreme points/ marginal points that lie on the hyperplane

i.e. ① What are the samples that are most difficult to classify?
- support samples that lie on the marginal hyperplane

② What directly affects the process of finding the optimum location of the decision boundary?

[w^* depends on?]

- w^* only depends on the support vectors. If we can find them then our problem is solved

This is not always the case!

Let's say that this is our training matrix: X^{tr}

X^{tr} = [class 1 ⋮ class 2]

training samples

and we want to map them from the parameters W^*, b^* which maps them into the label space.

X^{tr} = [class 1 ⋮ class 2] $\xrightarrow{W^*, b^*}$ [label]

, but it not obvious ... you can't easily find the support points because when you look at them, there are a plethora of things to keep in mind

Relationship
- ① Samples, pairwise relationships
- ② We don't know which vectors lie on the margin ⇝ We are looking for those S.V's.

⇝ Only a very small subset of training samples (S.V's) can fully specify the decision function.

$$W^* = \sum_{s} \alpha_s y_s x^s \quad , \quad s = S.V's, \quad \alpha_s \neq 0$$

Testing stage:

* What is the decision Rule for a new testing Sample?

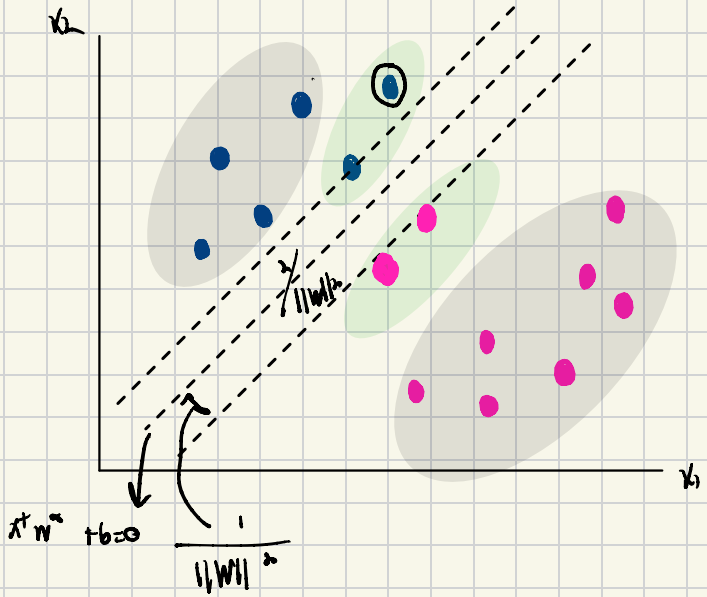
$$\text{sign}\left(x^{test\,T} w^* + b^* \right)$$

$$\text{sign}\left(x^{test\,T} \sum_s \alpha_s y_s x^s + b^* \right)$$

Compute sign

tells us if point lies above or below the margin

$$\frac{\gamma}{\|w\|_2}$$



Key Take Aways:

① The dot product allows for the quantification of the similarity between two samples.

② If 2 samples are very similar in features (dot product ~ 1), then they would have similar labels (i.e. lie on the same side with respect to the classifier hyperplane).

③ Quantification of the similarity between samples remains a vibrant research area in machine learning.

④ Kernels can be used to define a similarity between two samples.

⑤ Once we find optimal w^* & b^*, you can determine if the support vector (sample) lies on -1 or $+1$ marginal hyperplane by checking the sign $(x^T w^* + b^*)$. If it's positive then it lies on the $+1$ hyperplane, otherwise it lies on the -1 hyperplane.