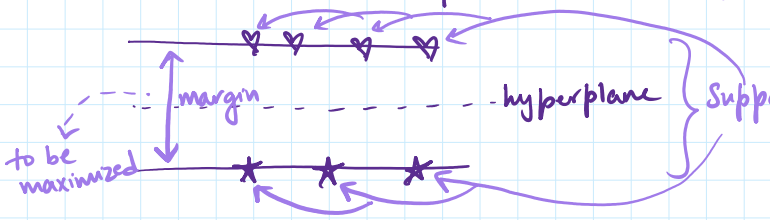
SVM

04 October 2024 04:32

Margin = dist. b/w decision boundary + data point of each class.

→ SVM tries to MAXIMIZE MARGIN —how?→ FINDING OPTIMAL HYPERPLANE.



Support vectors = data points that lie close to the decision boundary

↓

determine the hyperplane (support)

↓

This is used to calculate margin

So, ① Support vectors are the ones closest to boundary

② Based off this, a hyperplane is made

③ Only after you get the hyperplane, you look for the margin

④ Now maximize the margin (separate as much as you can)

Remember the perceptron?

- It was guaranteed to converge
- gave us a LINEAR hyperplane
- acc. to the weights, all hypotheses treated equally.

↓ Problem

The perceptron couldn't find an optimal hypothesis

Big skrissue.

This meant that each time you ran the perceptron, you'd get a new hyperplane!

Say you're a two dimensional input pt (x_1, x_2)

⇒ One dimensional hyperplane (straight line)

$x_2 = ax_1 + b$

$\Rightarrow ax_1 - x_2 + b = 0$

$\Rightarrow \begin{bmatrix} a \\ -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b = 0$

$\Rightarrow \bar{w} \cdot \bar{x} + b = 0$

ie, $\boxed{\bar{w}\bar{x} + b = 0}$ → Holds true for all points on the hyperplane.

But what does it mean for a random point (not) on hyperplane?

↓

DISTANCE OF $\bar{x}$ FROM HYPERPLANE

↓

So say we've $(\bar{x}, y)$ + hyperplane defined by $\bar{w} + b$.

$\Rightarrow \boxed{\beta = \bar{w}\bar{x} + b}$ where β is the dist. of $\bar{x}$ from hyperplane

Say you have a dataset $D = \left[(\bar{x}_i, \bar{y}_i) \mid \bar{x}_i \in R^n, y_i \in \{-1, 1\} \right]_{i=1}^{m}$

Let B = smallest β over all m samples

$\boxed{B = \min_{i=1..m} \beta_i}$ → Then find this for each hyperplane over m samples.

↓

Then choose the hyperplane that maximises B over k hyperplanes.

→ Cuz you want the most seperation of the classes.

Don't get lost:

β = distance of pt. from hyperplane.

B = min distance of all samples from given hyperplane

m = no. of samples

k = no. of hyperplanes.

↓

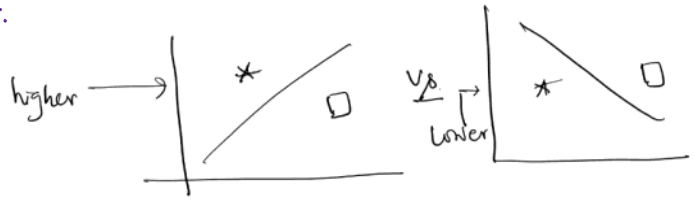
What are we doing?

```
for (hyperplane in hyperplanes):
    for (sample in samples):
```

for (hyperplane in hyperplanes):
    for (sample in samples):
        calculate β.
    return min β as B — ① get min distance B of all samples.
return max B — ② get max B of all hyperplanes.

seperation of the classes.

Now, look at this:



Both classify correctly. So wassup?

They're producing opposite signs!

Oki. Let's resolve.

Our training sample is of the form $(\bar{x}, y)$

What does f do?
- When we looked @ the margin, we were looking @ how well seperated points were.
- f looks at that AND whether those classifications are correct.

More nuance:
- point is correctly classified & far from decision boundary ↳ large, +ve
- misclassified → -ve
- point is on or close to db. (correctly classf) → small, +ve

① Multiply β by y to get a new number f:

$f = y \times \beta$

$f = y(\bar{w}\bar{x} + b)$

} Who is f? THE FUNCTIONAL MARGIN OF THE SAMPLE.

② Now we compare:

y is +ve, $(\bar{w}\bar{x}+b)$ is +ve | f is +ve
y is -ve, $(\bar{w}\bar{x}+b)$ is -ve | f is +ve
→ Both cases imply correct classification ↓ f is +ve for corr. classifn.

y is -ve, $(\bar{w}\bar{x}+b)$ is +ve | f is -ve
y is +ve, $(\bar{w}\bar{x}+b)$ is -ve | f is -ve
→ Both cases imply wrong classification ↓ f is -ve for wrong classifn.

③ SO, for a given dataset D that has m samples:

$F = \min_{i=1\cdots m} f_i$

$F = \min y_i(\bar{w}\bar{x}_i + b)$

} Who is F? THE FUNCTIONAL MARGIN OF THE DATASET

④ But why take minimum?

First understand the functional margin.
- It's like a fence seperating two groups of animals.
- It tells you how far each animal is from the fence & if they're on the correct side of it.
- The farther an animal is from the fence (+ on the correct side), the more confident you are that the fence is placed correctly.

Got that so far?

Now, for a classifier to be effective, it should:
a) classify all training samples correctly.
b) classify correctly WITH CONFIDENCE

Now, for a classifier to be effective, it should:

a) classify all training samples correctly.
b) classify correctly WITH CONFIDENCE

What does $\min_{i=1..m} f_i$ imply?

f_i is essentially the 'confidence' of the SVM in its predictions.

a) if $y_i = 1$:
large +ve value of $(\bar{w}\bar{x}+b)$ gives large, +ve f.
⇒ Strong confidence that point is +ve.

b) if $y_i = -1$
large -ve value of $(\bar{w}\bar{x}+b)$ gives large +ve f
⇒ Strong confidence that point is -ve.

Why $\min f_i$ then? Don't we want most confidence?

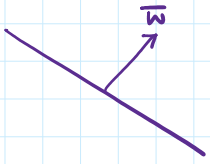
By finding $\min f_i$, we're finding the sample CLOSEST to the hyperplane.
I mean, it's cuz they're so close that the confidence is so low.

Why is the sample that's closest to hyperplane matter?

a) It represents the hardest case for classification
b) By focusing on this minimum margin, we ensure that challenging examples are classified correctly.

Now one more issue omg.

SCALE INVARIANCE SKISSUE.



The hyperplane is orthogonal to $\bar{w}$.

The hypothesis is $\bar{w}\bar{x}+b$.

If $|\bar{w}|$ or $|b|$ are of higher magnitude, the hypothesis doesn't change. It just gets scaled ↑ or ↓.

$\bar{w}$ still represents the same hyperplane.

To fix this scaling up & down business, we'll take the unit vector of $\bar{w}$.

From before, $f = y \cdot (\bar{w}\bar{x}+b)$

↓ get unit vector now

$$\gamma = y\left(\frac{\bar{w}}{|\bar{w}|}\bar{x} + \frac{b}{|\bar{w}|}\right)$$

Geometric margin of sample ←

Magnitude wise, $GM = \frac{FM}{|\bar{w}|}$ (geometric margin; functional margin)

So per usual, for the dataset, it becomes: → Dw/ m samples.

$$M = \min_{i=1...m} \gamma_i$$] Smallest margin over dataset.

← Geometric margin of dataset

ie

$$M = \min_{i=1..m} y_i\left(\frac{\bar{w}}{|\bar{w}|}\cdot\bar{x} + \frac{b}{|\bar{w}|}\right)$$

So, net-net:

a) functional margin of a data point wrt the hyperplane is a measure of confidence in the classification of that point.

$f_i = y_i(\bar{w}\bar{x}_i + b)$ ← confidence

($\bar{w}$: weight vector; b: bias term)

→ FM tells us how far $\bar{x}_i$ is from the decision boundary in terms of the raw output of the linear function.

b) the geometric margin is the "normalized" version of FM for scale invariance (by dividing by $|\bar{w}|$)

This gives us the actual distance from the data point to decision boundary.

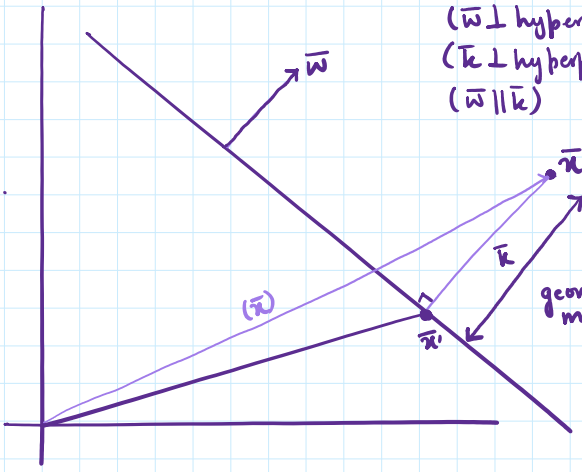
$\gamma_i = y_i\left(\frac{\bar{w}}{|\bar{w}|}\bar{x}_i + \frac{b}{|\bar{w}|}\right)$ ← distance

→ gives the $\perp^{ar}$ distance from the point to the hyperplane.

↳ More direct measure of distance of pt. from hyperplane.

Now, let's look at it geometrically:

for a point $\bar{x}$, γ (geometric margin) of x is the $\perp^{ar}$ distance from hyperplane, $\bar{k}$.



($\bar{w} \perp$ hyperplane)
($\bar{k} \perp$ hyperplane, from $\bar{x}$)
($\bar{w} \parallel \bar{k}$)

(this d is just a general expression ↳ will multiply by y_i at the end to get the gm for a particular sample.)

$|\bar{k}| = d$

Since $\bar{k} \parallel \bar{w}$, $\hat{k} = \frac{\bar{w}}{|\bar{w}|}$

So, $\bar{k}$, the vector distance of $\bar{x}$ from hyperplane, becomes:

$$\bar{k} = (\text{magnitude})(\text{direction}) = d \cdot \frac{\bar{w}}{|\bar{w}|}$$

(d: geometric margin; $\frac{\bar{w}}{|\bar{w}|}$ → direction of $\bar{w}$.)

Using the Δ^{le} law of vector addn:

$$\bar{x} = \bar{x}' + \bar{k}$$

$$\Rightarrow \bar{x}' = \bar{x} - \bar{k}$$
$$= \bar{x} - d \cdot \frac{\bar{w}}{|\bar{w}|}$$

What $\bar{x}'$ is on the hyperplane ($\bar{w}\bar{x}_i + b = 0$)
plug in $\bar{x}'$ for $\bar{x}_i$

$$\bar{w}\left(\bar{x} - d\frac{\bar{w}}{|\bar{w}|}\right) + b = 0$$

$$\bar{w}\bar{x} - d\frac{\bar{w}\cdot\bar{w}}{|\bar{w}|} + b = 0.$$

$$\bar{w}\bar{x} - d\cdot\frac{|\bar{w}|^2}{|\bar{w}|} + b = 0$$

$$\bar{w}\bar{x} - d|\bar{w}| + b = 0$$

$$d|\bar{w}| = \bar{w}\bar{x} + b$$

$$d = \left[\frac{\bar{w}}{|\bar{w}|}\bar{x} + \frac{b}{|\bar{w}|}\right]$$

→ Now we multiply by y so that we get a hyperplane that correctly classifies data.

$$\gamma_i = y\left[\frac{\bar{w}}{|\bar{w}|}\bar{x} + \frac{b}{|\bar{w}|}\right]$$

And then when we find the gm for another plane, we just change the values of $\bar{w}, b$.

Oof.

Now how do we find the $\bar{w}$ that gives us the highest gm?

Let's look at optimizations:

Unconstrained — if we've to minimize $f(x)$, here, we'll search ALL possible values of x.

Constrained — let's work with the example, $\min f(x)$, such that (st) $x = 2$.

We write it as $\min f(x)$ st $x - 2 = 0$

The feasible set will be the values of x that can optimize $f(x)$.

for $\min f(x)$ st $x = 2$, the feasible set is the set of all points (x, y) such that $(x, y) \equiv (2, y)$

Now, we can have many constraints too,
look at this: $\min f(x, y, z)$ st $x = 2$, $y = -8$, $z = -3$
now rewrite it as:

$\min_x f(x, y, z)$
st $x - 2 = 0$, $y + 8 = 0$, $z + 3 = 0$

↓ or just represent it as a vector!

$\min f(\bar{x})$
st $x_1 - 2 = 0$
$x_2 + 8 = 0$
$x_3 + 3 = 0$

Now,

SVM Optimization Problem.

① What we have is a linearly separable dataset D.

$$D = \left[(x_i, y_i) \mid x_i \in R^n,\ y_i \in \{+1, -1\}\right]$$

① What we have is a linearly separable dataset D.

$$D = \left[(x_i, y_i) \mid x_i \in R^n,\ y_i \in \{+1, -1\} \right]$$

and also vector $\bar{w}$ + bias b.

② wkt the gm of the dataset $M = \min_{i=1\ldots m} \gamma_i$ → dataset has m samples.

and what's γ_i? $\gamma_i = y_i \left(\frac{\bar{w}}{|\bar{w}|} \cdot \bar{x}_i + \frac{b}{|\bar{w}|} \right)$ } gm of a sample i among the m samples in the dataset.

③ we also know that the optimal hyperplane is the hyperplane defined by $\bar{w}$ + b such that **M is the LARGEST**

→ where M is the gm over the dataset D.

Now, to find $\bar{w}$ + b, we solve the optimization problem

↳ Constraint: margin of each sample γ_i should be greater than or equal to M.

↓

Maximize M such that $\gamma_i \geq M$, $i = 1 \ldots m$

From before, the gm is the fm divided by $|\bar{w}|$. → $M = \frac{F}{|\bar{w}|}$

Let's rewrite them.

The optimizn problem becomes:

Maximize M such that $\frac{f_i}{|\bar{w}|} \geq \frac{F}{|\bar{w}|}$, $i = 1 \ldots m$ (fi → gm for individual data points.)

We can simplify further.

Maximize M such that $f_i \geq F$, $i = 1 \ldots m$.

Hurrah. What does this tell us, though?

In the hyperplane, if we scale up $\bar{w}, b$:

→ The hyperplane direction DOES NOT CHANGE.

→ Only FUNCTIONAL MARGIN gets bigger.

If we scale down $\bar{w}, b$ such that $F = 1$ — OPTIMAL GM REMAINS SAME.

↓

Let's check it out:

→ maximize M st $f_i \geq F$.

→ maximize $\frac{F}{|\bar{w}|}$ st $f_i \geq F$

! now, F is scaled to 1. !

→ maximize $\frac{1}{|\bar{w}|}$ st $f_i \geq F$

Maximizing $\frac{1}{|\bar{w}|}$ = Minimizing $|\bar{w}|$.

↓

minimize $|\bar{w}|$ st $f_i \geq 1$.

Minimizing $|\bar{w}|$ = Minimizing $\frac{1}{2}|\bar{w}|^2$

minimize $|\bar{w}|$ s.t $f_i \geq 1$.

Minimizing $|\bar{w}|$ = Minimizing $\frac{1}{2}|\bar{w}|^2$

minimize $\frac{1}{2}|\bar{w}|^2$ s.t $f_i \geq 1$

$f_i = y_i(\bar{w}\bar{x}_i + b)$

→ minimize $\frac{1}{2}|\bar{w}|^2$ s.t $y_i(\bar{w}\bar{x}_i + b) - 1 \geq 0$

Let's look @ CONSTRAINED OPTIMIZATION now:

↓

Lagrange Multiplier.

Ex 1: $f(x,y) = 3x + 4y$ → Objective

s.t $g(x,y): x^2 + y^2 = 1$ → Constraint.

$$\nabla f = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\nabla g = \begin{bmatrix} \partial g/\partial x \\ \partial g/\partial y \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

A contour plot is a CONSTANT VALUE SURFACE.

↳ Since all points in the contour plot have the same value.

↳ rate of change in tangent to contour

↳ BUT, gradient ⊥ contour.

Since we're doing $f(x,y)$ and its constraint $g(x,y)$, $\nabla g \parallel \nabla f$

$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

$$\nabla f(x,y) - \lambda \nabla g(x,y) = 0.$$

Now, let's assume $L(x,y,\lambda) = f(x,y) - \lambda g(x,y)$

→ $\nabla L(x,y,\lambda) = \nabla f(x,y) - \lambda \nabla g(x,y)$

primal objective f^n — $f(x,y)$ ← We then solve for $\nabla L = 0$ since $\nabla f(x,y) - \lambda \nabla g(x,y) = 0$.

Lagrangian dual — $L(x,y,\lambda)$

Let's get back to our example now:

$f(x,y) = 3x + 4$

$g(x,y): x^2 + y^2 = 1$

$$\nabla L = \nabla f - \lambda \nabla g = 0$$

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix} - \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix} = 0$$

$$\nabla L = \nabla f - \lambda \nabla g = 0$$

$$\begin{vmatrix} 3 \\ 4 \end{vmatrix} - \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix} = 0$$

$$\begin{vmatrix} 3 \\ 4 \end{vmatrix} = \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$3 = \lambda 2x \longrightarrow x = 3/2\lambda$
$4 = \lambda 2y \longrightarrow y = 4/2\lambda$

Now, what did the constraint say? $x^2+y^2=1$
plug in the new values of x, y

$$\left(\frac{3}{2\lambda}\right)^2 + \left(\frac{4}{2\lambda}\right)^2 = 1$$

$9 + 16 = 4\lambda^2$
$25 = 4\lambda^2$
$\lambda = \pm 5/2$ → $(x,y) = (3/5, 4/5)$ or $(x,y) = (-3/5, -4/5)$

Next: Try w/ 2 constraints:

$f(x,y) = x^2+y^2$
$g_1(x,y): x+y=0$
$g_2(x,y): y+1=0$

Here, $L(x,y,\lambda) = f(x,y) - \lambda_1 g_1(x,y) - \lambda_2 g_2(x,y)$

$$\Rightarrow \nabla L(x,y,\lambda) = \nabla f(x,y) - \lambda_1 \nabla g_1(x,y) - \lambda_2 \nabla g_2(x,y)$$

(cuz we set $\nabla L = 0$)
$$0 = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix} - \lambda_1 \begin{bmatrix} \partial g_1/\partial x \\ \partial g_1/\partial y \end{bmatrix} - \lambda_2 \begin{bmatrix} \partial g_2/\partial x \\ \partial g_2/\partial y \end{bmatrix}$$

$$0 = \begin{bmatrix} 2x \\ 2y \end{bmatrix} - \lambda_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2x \\ 2y \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda_2 \end{bmatrix}$$

$$\Rightarrow \quad x = \lambda_1/2 \qquad y = \lambda_1 + \lambda_2/2$$

Two constraints: $x+y=0$ $\qquad$ $y+1=0$

substitute.

$\frac{\lambda_1}{2} + \frac{\lambda_1+\lambda_2}{2} = 0$ $\qquad$ $\frac{\lambda_1+\lambda_2}{2} = -1$

$2\lambda_1 + \lambda_2 = 0$ $\qquad$ $\lambda_1+\lambda_2 = -2$

So, we get a two variable system of two eqns.

$2\lambda_1 + \lambda_2 = 0$ $\quad\quad$ $\lambda_1 + \lambda_2 = -2$

So we got a two variable sitn w/ two eqns.

$$\begin{aligned} \lambda_1 + \lambda_2 &= -2 \\ 2\lambda_1 + \lambda_2 &= 0 \end{aligned}$$ (−)

$$-\lambda_1 = -2$$
$$\lambda_1 = 2$$
$$\lambda_2 = -4$$

next.

Wolfe Dual + Slater's Theorem

So from before, we've to minimize $f(\bar{w}) = \frac{1}{2}|\bar{w}|^2$ subject to m constraints of the form $g_i(\bar{w}, b)$

We'll call $y_i(\bar{w}\bar{x}_i + b)$ as $g_i(\bar{w}, b)$

$$g_i(\bar{w}, b) = y_i(\bar{w}\bar{x}_i + b) - 1 \quad i = 1, 2 \ldots m$$

$$L(\bar{w}, b, \alpha) = f(\bar{w}) - \sum \alpha_i g_i(\bar{w}, b)$$

$$= \frac{1}{2}|\bar{w}|^2 - \sum \alpha_i [y_i(\bar{w}\bar{x}_i + b) - 1]$$

Here, we have one lagrangian multiple α_i for every constraint g_i.

Let's solve using the dual.

The Duality Principle.

The Lagrangian problem has m inequality constraints (cuz m samples)

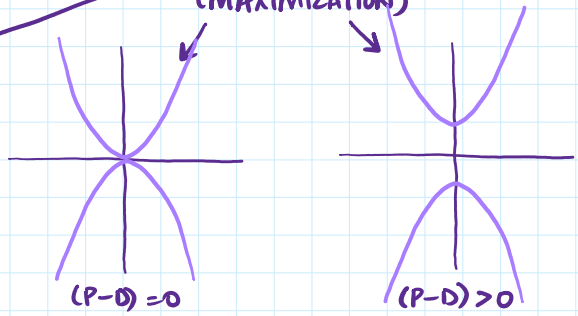
"An optimization can be solved in two ways."

- Primal Problem (MINIMIZATION)
- Dual problem (MAXIMIZATION)

Maximum of dual problem will be less than or equal to the minimal of the primal problem.

$\max(\text{dual}) \leq \min(\text{primal})$

dual provides lower bound to primal.



(P−D) = 0
STRONG DUALITY

(P−D) > 0
WEAK DUALITY.

In convex optimization, an AFFINE CONSTRAINT is a constraint that can be expressed as

$$\bar{a}^T \bar{x} \leq b$$

$\bar{a}$: constant vector; $\bar{x}$: decision variable; b: constant scalar

An affine function is linear and constant.

Slater's theorem says that strong duality holds for slater's condition, which in turn holds for affine constraints.

Done?

Let's jump back real quick to the lagrangian function

$$L(\bar{w}, b, \alpha) = \frac{1}{2}|\bar{w}|\cdot|\bar{w}| - \sum \alpha_i [y_i(\bar{w}\bar{x}_i + b) - 1]$$

LEGRANGIAN PRIMAL PROBLEM

minimize wrt $\bar{w}, b$ — maximize wrt α — At the same time!

$$\min_{w,b} \max_{\alpha} L(\bar{w}, b, \alpha) \quad \text{st } \alpha_i \geq 0 \ \forall \ i = 1, 2 \ldots m$$

① To solve the minimization, we take PD wrt $\bar{w}, b$.

$$\nabla_w L = \bar{w} - \sum \alpha_i y_i \bar{x}_i = 0 \quad \text{---- (i)}$$

$$\nabla_b L = -\sum \alpha_i y_i = 0$$

$$\Rightarrow \bar{w} = \sum \alpha_i y_i \bar{x}_i$$

Now plug this value into L

$$w(\alpha, b) = \frac{1}{2}\left(\sum \alpha_i y_i \bar{x}_i\right)\left(\sum \alpha_i y_i \bar{x}_i\right) - \sum \alpha_i \left[y_i \left(\sum \alpha_j y_j x_j\right) \cdot x_j + b\right] + \sum \alpha_i$$

↓ all the math later.

$$w = \sum \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j$$

$\alpha_i \alpha_j$: their alphas. $y_i y_j$: labels. $\bar{x}_i \cdot \bar{x}_j$: pair of samples.

Optimization depends on the dot product of the pair of samples.

If $\partial L / \partial b = 0$, set $\sum \alpha_i y_i$ to zero!

$$w(\alpha) = \sum \alpha_i - \frac{1}{2} \sum_{i=1} \sum_{j=1} \alpha_i \alpha_j y_i y_j \cdot \bar{x}_i \bar{x}_j$$

Wolfe Dual Lagrangian function!

→ Use to solve primal problem.

Optimization becomes a WOLFE DUAL problem!

$$\underset{\alpha}{\text{maximize}} \sum_{j=1} \alpha_i - \frac{1}{2} \sum_{i=1} \sum_{j=1} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j, \text{ st } \alpha_i \geq 0 \text{ for } i = 1 \ldots m$$

and also $\sum \alpha_i y_i = 0$

and also $\sum \alpha_i y_i = 0$

Now, the main advantage of wolfe-dual is that the objective function ($\bar{w}$) only depends on lagrangian multipliers!

↓

ie $\bar{w} \leftrightarrow$ lagrangian multipliers ONLY.

Important stuff to infer:

① Most of the data points really do not matter.

② All datapoints $\bar{x}_i$ are vectors. — few of them are (support vectors) ← only support vectors have non-zero α_i

③ All $\alpha_i = 0$ belong to those datapoints that are FAR from decision boundary.

More about the lagrange multiplier:

① This method is used for problems w/ equality constraints BUT.

② We're using it with inequality constraints

→ So wassup?

↓

This method still works for inequality provided (some conditions) are satisfied.

1st order for optimal optimization

+ some regularity conditions

Slater's

svm + convex

→ KKT Condition (Karush, Kuhn, Tucker condition)

primal and dual at same time { Sufficient for zero duality

① STATIONARITY CONDITION

$$\nabla_w \alpha = w - \sum \alpha_i y_i \bar{x}_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum \alpha_i y_i$$

- if no constraint, ∇objective^{fn} is 0.
- w/ constraint, use lagrangian

② PRIMAL FEASIBILITY CONDITION

$$y_i(\bar{w}\bar{x}_i + b) - 1 \geq 0 \quad \forall \; i = 1 \dots m$$

- essentially, constraint of primal problem
- no skrissue.

③ DUAL FEASIBILITY CONDITION

$$\alpha_i \geq 0 \quad \forall \; i = 1 \dots m$$

④ COMPLEMENTARY SLEEKNESS

$$\alpha_i [y_i (\bar{w} \cdot \bar{x}_i + b) - 1] = 0 \quad \forall \; i = 1 \dots m$$

- either $\alpha_i = 0$ or $y_i(\bar{w}\bar{x}_i + b) - 1 = 0$
- $\alpha_i \neq 0$
- SO Constraint is active when $y_i(\bar{w}\bar{x}_i + b) - 1 = 0$.

Constraint is active when $y_i(\bar{w}\bar{x}_i + b) - 1 = 0$.

NOW

After we solve wolfe dual & get vector α of all $\alpha_i^{(i=1..m)}$, we find $\bar{w}$ & b.

Step1 : Calculate $\bar{w}$

$$\bar{w} = \sum \alpha_i y_i \bar{x}_i$$

Step2 : calculate b

Here we use constraint of primal problem to find b.

$y_i(\bar{w}\bar{x}_i + b) - 1 \geq 0$

cuz closest pt. to hyperplane has functional margin = 1.

For support vectors:

$y_i(\bar{w}\bar{x}_i + b) = 1$

$\downarrow$

$(y_i)\, y_i(\bar{w}\bar{x}_i + b) = y_i$

$(y_i)^2 = 1$

$\bar{w}\bar{x}_i + b = y_i \rightarrow b = y_i - \bar{w}\bar{x}_i + b$.

Ta daaaa

not done.

How do we find the class of an example x_j?