Support Vector Machines

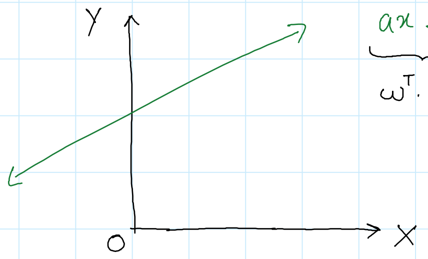
14 October 2021 18:46

1. In an n - dimensional space, the equation of a "hyperplane" is given by :

$$\Rightarrow \quad \omega^T X + b = 0$$

where $\vec{\omega}$ and $\vec{X}$ are n dimensional vectors and b is the bias term. $\in \mathbb{R}$

Eg: 2-D space (1-D line)



$$\underbrace{ax + by}_{\omega^T . X} + c = 0$$

where :

$$\omega = \begin{bmatrix} a \\ b \end{bmatrix} \;\&$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$(1\times 2) X (2 \times 1) \Rightarrow (1 \times 1) \in \mathbb{R}$$

So, we can always characterize or define a d - dimensional hyperplane with $\vec{\omega}$ & b.
(Think of ω as the slope of a 1-D line and b as the intercept.)

Thus, for a line -

$$ax + by + c = 0 \iff \omega_2 x_1 + \omega_1 x_0 + \omega_0 = 0$$

$$\Updownarrow$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}^T \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} + \omega_0 = 0$$

for a plane -

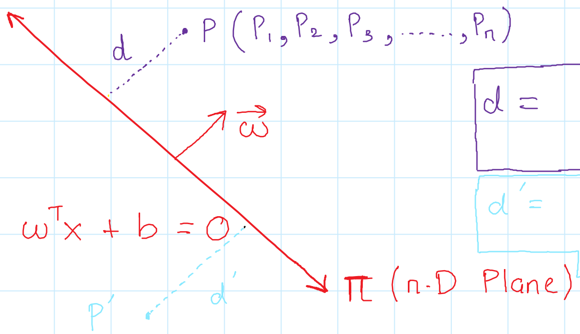
$$ax + by + cz + d = 0 \iff \omega_3 x_2 + \omega_2 x_1 + \omega_1 x_0 + \omega_0 = 0$$

$$\Updownarrow$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^T . \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \omega_0 = 0$$

2. To 'cut' or distribute an n - dimensional space, we need an (n-1) dimensional 'hyperplane'.
   - $\Rightarrow$ Divide a line with a point.
   - $\Rightarrow$ Cut a cubical space with a plane sheet.

3. Distance of a point from a plane (n-D)



$$\boxed{d = \frac{\vec{\omega}.\vec{P}}{\|\omega\|}} = \|\vec{P}\| \cos\theta$$

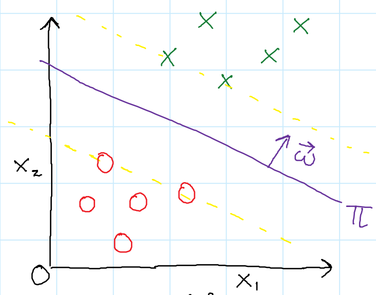
$$\boxed{d' = \frac{\vec{\omega}.\vec{P'}}{\|\omega\|}} = \|\vec{P}'\| \cos\theta'$$

This is pretty intuitive, we take the projection of $\vec{P}$ on the normal $\vec{\omega}$ and divide by its norm to get d.

ARMED WITH THIS KNOWLEDGE, WE CAN ATTACK SUPPORT VECTOR MACHINES !

AIM : To create a Maximum Margin Classifier for some linearly separable data. Or else use Kernel Function(s) to create higher dimensional inner products.

HYPOTHESIS : Once your model is ready, it ideally should classify all + samples & - samples correctly. So, it should lie such that all points on one side are + samples and all points on the other side are - samples. with the widest gutter width and "cushioning" on both the sides to maximize generalization.



A sample "crosses" the hyperplane if

$\vec{\omega}.\vec{x} \geqslant c$ (covered properly in my notes)

$\Rightarrow \vec{\omega}.\vec{x} - c \geqslant 0$

$\Rightarrow \vec{\omega}.\vec{x} + b \geqslant 0$

$\Rightarrow \quad \vec{\omega}.\vec{x}_+ + b \geqslant 0$ (for all + samples)

$$\vec{\omega}\cdot\vec{x}_- + b \leq 0 \qquad (\text{for all} - \text{samples})$$

$$\Rightarrow \quad \vec{\omega}\cdot\vec{x}_+ + b \geq 0 \qquad \text{for} \quad y = +1$$
$$\vec{\omega}\cdot\vec{x}_- + b \leq 0 \qquad \text{for} \quad y = -1$$

$$\Rightarrow \quad y_i(\vec{\omega}\cdot\vec{x}_i + b) \geq 0 \qquad \forall \; i \in [1, 2, \ldots, m]$$
(samples)

Now let's come back to the training phase. We want to have -

$$\vec{\omega}\cdot\vec{x}_i + b \geq +1 \qquad \text{for} + \text{samples}$$
$$\vec{\omega}\cdot\vec{x}_i + b \leq -1 \qquad \text{for} - \text{samples}$$

(covered properly in my notes)

A new explaination for the above eq^{ns}:

Distance of a point from a hyperplane

$$= \frac{\vec{\omega}\cdot\vec{x}}{\|\vec{\omega}\|}$$

So for + samples,

$$\frac{\vec{\omega}\cdot\vec{x}_+}{\|\vec{\omega}\|} \geq \nu \qquad (\text{some margin threshold})$$

for − samples,

$$\frac{\vec{\omega}\cdot\vec{x}_-}{\|\vec{\omega}\|} \leq -\nu \qquad (\text{some margin threshold})$$

$\Rightarrow$ for − samples,
For + samples,

$$\vec{\omega}\cdot\vec{x}_- \leq \{-\nu\|\omega\| = -1\}$$
$$\vec{\omega}\cdot\vec{x}_+ \geq \{\nu\|\omega\| = 1\}$$

Because our normal vector $\vec{\omega}$ is independent of scaling, changing its magnitude won't do anything because it's only meant for directing our hyperplane. (Discussed properly in my notes)

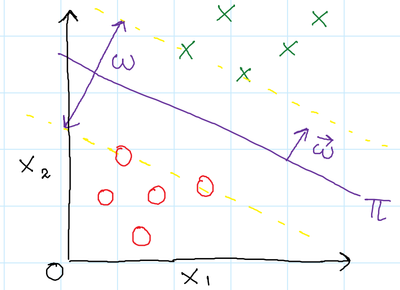
OPTIMIZATION FOR HARD MARGIN :

So till now we discussed our expectations from the algorithm in the form of constraints and to sum them up,

$$\omega\cdot x_+ + b \geq +1 \quad \Big\} \; (1)$$

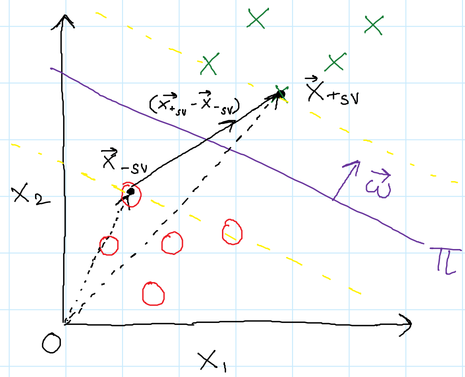
$$\left.\begin{aligned} \omega . X_{+} + b &\geqslant +1 \\ \omega . X_{-} + b &\leqslant -1 \end{aligned}\right\} \quad (1)$$

Our motive was to get the Maximal Margin Classifier which has the widest gutter / street.



Maximize ω

Note that ω is completely different from $\vec{\omega}$.



Width of street = Projection of $(\vec{X}_{+sv} - \vec{X}_{-sv})$ on a unit vector in the direction of $\vec{\omega}$

$$= (\vec{X}_{+sv} - \vec{X}_{-sv}) \cdot \frac{\vec{\omega}}{\|\vec{\omega}\|} = \frac{\vec{\omega}.\vec{X}_{+sv} - \vec{\omega}.\vec{X}_{-sv}}{\|\omega\|}$$

$$= \frac{1-b-(-1-b)}{\|\omega\|} = \boxed{\frac{2}{\|\omega\|}} \quad (\text{From } (1))$$

So, we want to maximize $\frac{2}{\|\omega\|}$ or minimize $\|\omega\|$

MINIMIZE $\left\{\frac{1}{2}\|\omega\|^2\right\}$ (for mathematical ease)
(Quadratic problem with just a global minima)

subject to $y_i\{\omega . x_i + b\} = 1$ (We don't have inequality here because our model only depends on SVs)

We will use Lagrangian here to solve our optimization problem.

$$\mathcal{L}(\omega, b) = \frac{1}{2}\,\omega\cdot\omega - \sum_{i=1}^{m} \alpha_i\left[y_i(\omega . x_i + b) - 1\right]$$

where α_i : i^{th} Lagrangian Multiplier $(\alpha_i \geq 0)$

{More details about Lagrangian are in my NOTES ¿}

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{m} \alpha_i [y_i x_i] = 0$$

$$\frac{\partial L}{\partial b} = 0 - \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$\Rightarrow \boxed{\omega_m = \sum_{i=1}^{m} \alpha_i y_i x_i} \quad - (2)$$

$$\Rightarrow \boxed{\sum_{i=1}^{m} \alpha_i y_i = 0} \quad - (3)$$

Using (2) & (3) in our Primal Lagrangian Problem,

$$L(\omega, b) = \frac{1}{2}\omega \cdot \omega - \sum_{i=1}^{m} \alpha_i y_i \omega . x_i - b \overbrace{\sum_{i=1}^{m} \alpha_i y_i}^{0} + \sum_{i=1}^{m} \alpha_i$$

$$L(\omega, b) = \frac{1}{2}\omega . \omega - \sum_{i=1}^{m} \alpha_i y_i \omega . x_i + \sum_{i=1}^{m} \alpha_i$$

$$L(\omega, b) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{m} \alpha_i$$

$$\boxed{L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)}$$

where $\alpha_i \geq 0$

and $\sum_{i=1}^{m} \alpha_i y_i = 0$

Now, why did we do this ?

Well, our primal lagrangian had 3 variables to tune in order to find the minima of the objective (and the lagrangian itself). We converted that to maximizing our New Lagrangian over α_i's by substituting ω and b as a function of α. This was done by using the property that derivatives at min = 0.

$$\left(\frac{\partial L}{\partial \omega}, \frac{\partial L}{\partial b}\right)$$

Refer to 'Introduction to Machine Learning: Support Vector Machines' for understanding about Lagrangian.

Eg: minimize $2 - x^2 - 2y^2$
x, y

subject to : ① $x + y - 1 = 0$ (Equality constraint)

② (You can also have inequality constraints)

$$\mathcal{L}(x,y,\alpha) = (2 - x^2 - 2y^2) - \alpha(x + y - 1)$$

and now we have an unconstrained problem with respect to x, y and α (Lagrangian multip.)

$\min \mathcal{L}(x, y, \alpha)$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial x} = 0, \quad \frac{\partial \mathcal{L}}{\partial y} = 0, \quad \frac{\partial \mathcal{L}}{\partial \alpha} = 0$$

$-2x - \alpha = 0$ — ①

$-4y - \alpha = 0$ — ②

$x + y - 1 = 0$ — ③

$\Rightarrow x + y = 1, \quad \alpha = -2x = -4y \Rightarrow \boxed{x = 2y}$

$$\boxed{y = 1/3, \quad x = 2/3, \quad \alpha = -4/3}$$

Similarly, you can have m constraints.

Eg 2:

$$\text{extr.}(f(x,y)) = 8x^2 - 2y$$

$$g(x,y) = x^2 + y^2 - 1 = 0$$

$$\text{extr.}(\mathcal{L}(x,y,\alpha)) = 8x^2 - 2y - \alpha(x^2 + y^2 - 1)$$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial x} = 16x - 2\alpha x = 0$$

$$\frac{\partial \mathcal{L}}{\partial y} = -2 - 2y\alpha - 0$$

$$\frac{\partial \mathcal{L}}{\partial \alpha} = -(x^2 + y^2 - 1) = 0$$

$x^2 + y^2 = 1$ — ①

$x(16 - 2\alpha) = 0$ — ②

$1 + \alpha y = 0$ — ③

i) $x = 0$ -

$y^2 = 1 \Rightarrow y = \pm 1$

$\alpha = \mp 1$

$x = 0, y = 1, \alpha = -1$

$x = 0, y = -1, \alpha = +1$

ii) $\alpha = 8$:-

$y = -1/8 \Rightarrow x = \sqrt{1 - \frac{1}{64}} = \pm\frac{\sqrt{63}}{8}$

$x = \pm\frac{\sqrt{63}}{8}, y = \frac{-1}{8}, \alpha = 8$

Now check which of the 4 solns. give us our maxima/minima.

Once you solve the Dual Problem, you will get your support vectors (those having non-zero α_i). After getting your support vectors, you can derive your ω^*.

$$\omega^* = \sum_{i=1}^{m} \alpha_i y_i x_i$$

[This raised a question in my mind, if we used a kernel function to get the α's, then we won't be able to use this formula because we won't have the x_i's in the higher dimensions.]

Once we get ω, we can plug in that into:

$$\omega^* \cdot x_{sv} + b^* = y_{sv} \quad \text{— (4)}$$

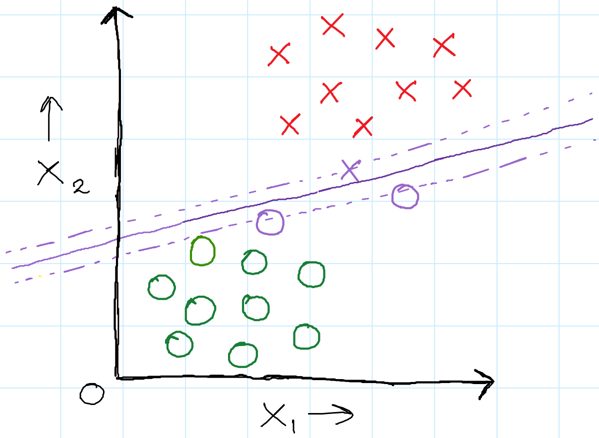
for some SV lying exactly on our margin.

$$\Rightarrow \boxed{b^* = y_{sv} - \omega \cdot x_{sv}}$$

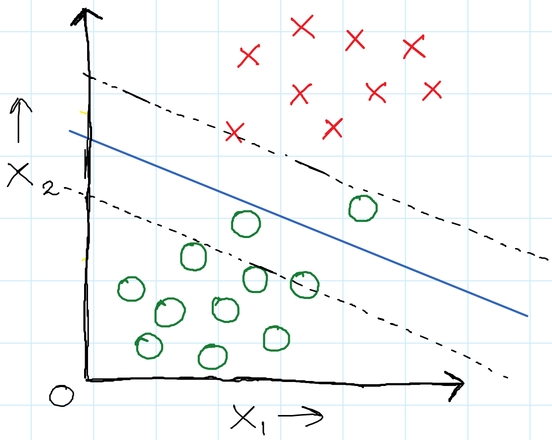
SOFT MARGIN CLASSIFICATION

For soft margin classifier, we need to allow some missclassifications so that we have a tradeff between the margin width and the 'correctness' of our classifier. This indicates that we need to have some 'slack' or 'dheel' (ढील) in classifying all the data points correctly so that our margin width does not suffer because of noise in our dataset.

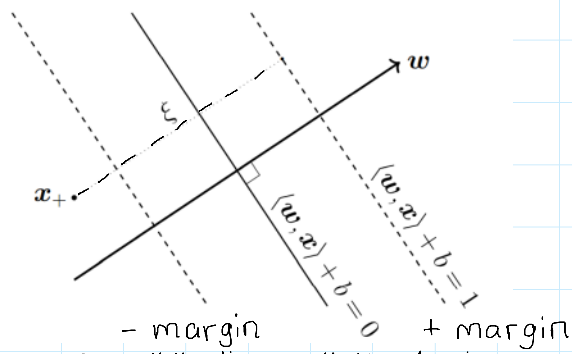
Eg: It is unlikely that any real world data will be linearly separable, even if it is, it might look somewhat like :



A soft margin SVM will/can allow some slackness in the process which will give us a wider margin at the cost of a few misclassifications.



So we aim at minimizing the total amounts of slacks while maximizing the margin width. Let's introduce a slack variable ε_i for every training example $i = 1, \ldots, m$.



Source: Mathematics for Machine Learning

Where ε quantifies the 'error' or deviation of a + sample x_+ from the + margin : $\omega \cdot x + b = +1$. For a SV, $\varepsilon_{sv} > 1$ means that it has been misclassified as it lies on the other side of the hyperplane. If $\varepsilon_{sv} < 1$, then it lies inside that margin yet correctly classified.

OPTIMIZATION:

$$\min_{\omega, b} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m} \varepsilon_i$$

subject to ① $y_i(\omega \cdot x_i + b) \geq (1 - \underbrace{\varepsilon_i}_{})$ $\forall\ i = 1, 2, \dots, m$

The amount of slack for the i^{th} sample.

② $\varepsilon_i \geq 0$ $\forall\ i = 1, 2, \dots, m$

If C is large, then even a very little slack will increase the cost. If C is kept small, then relatively more slack can be allowed.

Thus, reducing C helps in increasing the no. of SVs inside and on the margin which results in decreasing the variance and reducing over-fitting because our model will generalize better.

New Primal Problem :

New Primal Problem :

$$\mathcal{L}(\omega, b, \varepsilon, \alpha, \mu) = \frac{1}{2}\omega.\omega + C\sum_{i=1}^{m}\varepsilon_i - \sum_{i=1}^{m}\alpha_i\left[y_i(\omega.x_i+b)-(1-\varepsilon_i)\right] - \sum_{i=1}^{m}\mu_i\varepsilon_i$$

$$\alpha_i \geq 0 \qquad \forall\, i = 1, 2, \ldots, m$$
$$\mu_i \geq 0$$

$$\min_{\omega, b, \varepsilon} \; \max_{\alpha, \mu} \; \mathcal{L}(\omega, b, \varepsilon, \alpha, \mu)$$

NOTE : $y_i(\omega.x_i + b) \geq 1-\varepsilon_i$

if $\varepsilon_i = 0$, then this i^{th} sample is correctly classified.

if $y_i(\omega.x_i + b) = 1 - 0$

then it's a correctly classified S.V.

else if $\varepsilon_i > 0$, then this sample might be misclassified.

if $y_i(\omega.x_i + b) = 1-\varepsilon_i$ and $\varepsilon_i > 1$

then this '<u>is</u>' a misclassified data point.

else if $y_i(\omega.x_i + b) \geq 1-\varepsilon_i$ is smaller than 1, then it's misclassified.

Partially differentiating the Primal Lagrangian wrt the Primal Variables ω, b and ε gives us :

$$\frac{\partial\mathcal{L}}{\partial\omega} = \omega - \sum_{i=1}^{m}\alpha_i y_i x_i$$

$$\frac{\partial\mathcal{L}}{\partial b} = -\sum_{i=1}^{m}\alpha_i y_i$$

$$\frac{\partial\mathcal{L}}{\partial\varepsilon_i} = C - \alpha_i - \mu_i$$

Equating them to zero gives :

$$\boxed{\omega = \sum_{i=1}^{m}\alpha_i y_i x_i} \quad - \text{(1)}$$

$$\boxed{\omega = \sum_{i=1}^{m} \alpha_i y_i x_i} \quad - \text{①}$$

$$\boxed{\sum_{i=1}^{m} \alpha_i y_i = 0} \quad - \text{②}$$

$$\boxed{C - \alpha_i - \mu_i = 0} \quad - \text{③}$$

Substituting ①, ② and ③ after simplifying our Primal Problem :

$$\mathcal{L}_P : \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{m} \alpha_i y_i \left(\left\{ \sum_{j=1}^{m} \alpha_j y_j x_j \right\} \cdot x_i \right) - b \underbrace{\sum_{i=1}^{m} \alpha_i y_i}_{0}$$

$$+ \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i \varepsilon_i - \sum_{i=1}^{m} \mu_i \varepsilon_i + C \sum_{i=1}^{m} \varepsilon_i$$

$$\mathcal{L}_D : -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{m} \underbrace{(C - \alpha_i - \mu_i)}_{0} \varepsilon_i$$

$$\boxed{\mathcal{L}_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)}$$

Clearly, $\alpha_i \geq 0$ and $\mu_i \geq 0$ $\forall$ $i = 1, 2, \ldots, m$ as they are Langrangian Multipliers.

Given that $\alpha_i = C - \mu_i$ (from ③)

$$\Rightarrow \boxed{0 \leq \alpha_i \leq C}$$

All data points for which $0 < \alpha_i < C$ holds true, are our Support Vectors that lie exactly on our margin. This is explained by Karush - Kuhn Tucker conditions.

Now, solving the Dual Problem shown above, we will get our Lagrangian Multipliers back. This enables us to get back our Primal Variables ω and b which parametrize our separating hyperplane.

m ( )

parametrize our separating hyperplane.

$$\boxed{w^* = \sum_{i=1}^{m} \alpha_i y_i x_i} \qquad \text{(From 1)}$$

Once we get w, we can plug in that into:

$$w^* \cdot x_{sv} + b^* = y_{sv} \qquad \text{— 4}$$

for some SV lying exactly on our margin.

$$\Rightarrow \boxed{b^* = y_{sv} - w \cdot x_{sv}}$$

If everything was hunky-dory, we were done at this point.

BUT

It is possible that none of the SVs lie exactly on the margin i.e. all SVs violate the margin. Then we can't use 4 as our assumption fails. By some more math, we can say that in such a case we can find b^* as follows:

$$\boxed{b^* = \text{Median}\left(|y_s - w^* \cdot x_s|\right) \quad \forall \; s \in SVs.}$$