* Support Vector Machine (SVM)

- Supervised algorithm used for classification and regression.

  SVC: Support Vector Classifier

  SVR: Support Vector Regressor

- We plot each of the data points in n-dimensional space (n = no. of features) with value of each feature being the coordinate of each data point.
- Then we try to find hyperplane which seperates data points for classificaton or try to find hyperplane which has max. no. of data points for regression.
- **Advantages:**
  1. Effective in high dimensional space
  2. Effective if $n >$ no. of samples
  3. Versatile as different Kernels can be used for descision function.
- **Disadvantages:**
  1. If number of features too much high, i.e no. of dimensions high then overfitting occurs, to overcome this we need to choose our Kernel wisely.
  2. Don't provide probability estimate directly and need to use 5 cross validation technique.

A. Building Formula by intution

- Equation of simple line is, (Ref. linear regression)

$$y = mx + c$$

or

$$y = m_1x_1 + m_2x_2 + \dots + m_nx_n + c$$

or

$$h_{\theta_0}(x) = \theta_0 + \theta_1x_1 + \theta_2x_2 + \theta_3x_3 + \dots + \theta_nx_n$$

- Algebrically it is same as, (mul. by constants)

$$ax + by + c = 0$$

with,

$$y = \underbrace{\boxed{\frac{-a}{b}}}_{\text{coefficient}} x - \overbrace{\boxed{\frac{c}{b}}}^{\text{intercept}}$$

also for multiple features,

$$ax_1 + bx_2 + dx_3 + \dots + z_nx_n + c = 0$$

to make it bit generalized replace coeff by w_n

$$\boxed{w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + c = 0}$$

Converting this equation to matrix for ease, where,

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}_{n\times 1} \qquad X = [x_1 \ x_2 \ \dots \ x_n]_{1\times n}$$

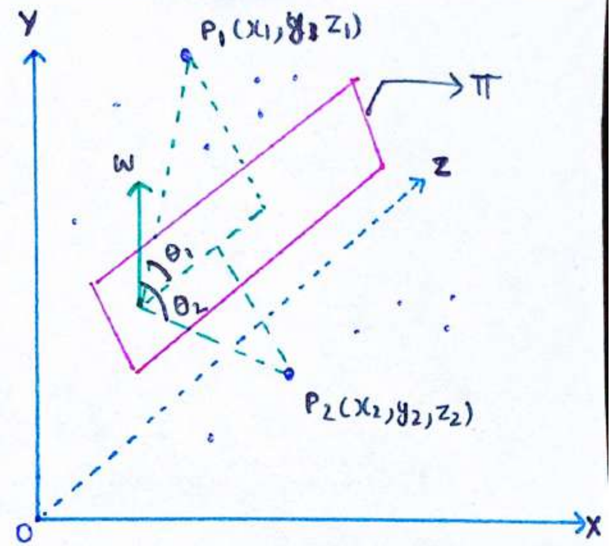
To put up in equation change the order of w by taking Transpose,

$$\boxed{W^TX + c = 0}$$

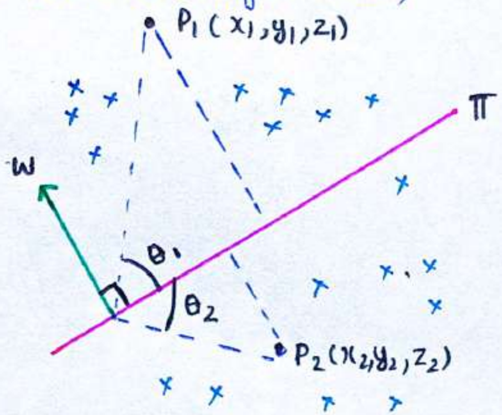
If intercept $c = 0$, $W^TX = 0$ is eq. of line passing via origin.

B. Distance of point from plane

- For simplification, Let we have 3 features, i.e 3 dimensions.
- Our features x_1, x_2, x_3 encoded as x, y, z for geometric intution
- Their are two data points $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$



For ease lets bring it to 2D,



- W: unit vector on π ($W \perp \pi$)
- π: a plane
- Points : x
- Distance of P_1 from π is

$$\frac{W^TP_1}{||W||}$$

- Distance of P_2 from π is

$$\frac{W^TP_2}{||W||}$$

- In general, Distance of a point from a plane (d) is

$$d = \frac{W^T P}{\|W\|}$$

or

$$\boxed{d = \|W\| \cdot \|P\| \cdot \cos\theta}$$

- **Observation:**
  1. Points above the plane make θ_1 angle with $\hat{w}$ and from formula and fact that θ_1 is always less than 90° as $\hat{w} \perp \pi$

     $\therefore$ Value of $\cos\theta$ for $\theta < 90$ is always +ve.

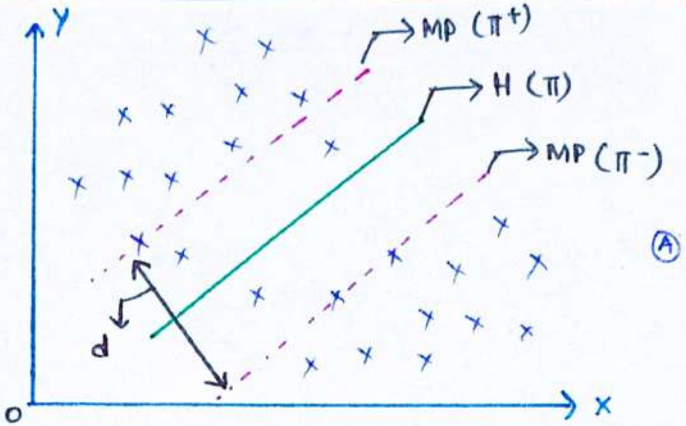
     $\Rightarrow$ **Points above plane always have +ve 'd'.**
  2. Points below plane make θ_2 angle with $\hat{w}$ and from formula and fact θ_2 always more than 90°.

     $\therefore$ value of $\cos\theta$ for $\theta > 90$ is always -ve.

     $\Rightarrow$ **Points below plane always have -ve 'd'**

C Support Vector Classifier

- **Support Vectors (SV):** Data points that are closer to hyperplane and influence the position and orientation of hyperplane.
- **Marginal Plane (MP):** Planes closer to the SV are marginal plane and help in choosing hyperplane.
- **Margin (d):** Distance between two marginal plane.
- **Hyperplane (H):** Best plane which clearly seperates data points with highest margin.



- Our aim is to find best plane which can clearly sepperate data points.
- This plane has maximum margin and called hyperplane (π).
- Our goal is to maximize the margin (d), classifier using such methodology is called **maximal margin classifier** and that maximum margin plane is called **margin maximizing plane**.
- Maximizing margin (M) given,

$$C + x_1 + x_2 + \dots + x_n \quad \text{(i)}$$

$$y_i\,(C + W_1 x_{1i} + W_2 x_{2i} + \dots + W_n x_{ni}) \geq M, \quad i = 1 \dots n \quad \text{(ii)}$$

- This equation means define margin M by tunning coefficents of all variables such that margin is maximized (ii) and product of predicted (observed) value with equation of respective input features should be greater than margin (ii)

- If hyperplane able to clearly sepperate data points like fig A then it is called **Hard margin SVC**.

In such case,

$$\pi^+ = W^T x_1 + C = +1$$

$$\pi^- = W^T x_2 + C = -1$$

If we add these two we get,

$$d = \frac{W^T (x_1 - x_2)}{\|W\|}$$

$$\therefore \boxed{d = \frac{2}{\|W\|}}$$

as $(\pi^+ + \pi^-) = W^T(x_1 - x_2) = 2$

- So, $d = 2/\|W\|$ is margin in case of hard margin classifier.
- We need to maximize 'd' by changin coefficents of features in matrix X which are present in matrix W^T.
- This margin is basically observed values distances under constraint,

$$y_i \begin{cases} 1, & W^T x + C \geq 1 \\ -1, & W^T x + C \leq 1 \end{cases}$$

for all points which is basically error.

- Combining constraints we get,

$$\boxed{y_i(W^T x + c) \geq 1}$$

- As $d = 2/\|w\|$ is our margin with respect to a support vector. which is also error if we inverse it,

  $\therefore \|w\|/2$

So, we have to maximize our margin $2/\|w\|$ or minimize our loss or error $\|w\|/2$.

- In most of cases data is not linearly seperable by hyperplane and this condition is resolved by Soft margin SVC.

- For classifying under this case we introduce slack variable $(\xi)(x_i)$ in our equation yielding,

$$y_i(w^T x + c) \ge 1 - \xi_i$$

if $\xi_i = 0$, point is correctly classified else
if $\xi_i > 0$, point incorrectly classified

- Incorrect classification means ξ variable is in incorrect dimension.
- ξ_i is basically an error associated with ξ variable

$$\therefore \text{Average error} = \frac{1}{n}\sum_{i=1}^{n} \xi_i$$

- Our objective is to minimize cost function, which is:

$$\boxed{J = \underset{(w,c)}{\text{minimize}} \frac{\|w\|}{2} + c_i \sum_{i=1}^{n} \xi_i}$$

where

c_i : how many points we can ignore for miss classification

ξ_i : summation of incorrect data points from marginal plane

$c_i \sum_{i=1}^{n} \xi_i$ is Hinge Loss function.

$\Rightarrow$ c_i and ξ_i are hyperparameters.

D. Support Vector Regressor

- As a Regressor it triest to fit a best plane which has maximum number of points.
- In this case our marginal planes equation are updated by introducing marginal error (ϵ)

$\therefore$

$$\pi^+ = w^T x + c + \epsilon$$

$$\pi = w^T x + c$$

$$\pi^- = w^T x + c - \epsilon$$

- Thus our cost function updates, insted of $\xi(x_i)$ we use ε_i (ita of i).

$$\therefore \boxed{J = \underset{(w,c)}{\text{minimize}} \frac{\|w\|}{2} + c_i \sum_{i=1}^{n} \varepsilon_i}$$

- And constraints are;

$$\boxed{|y_i - w^T x| \le \epsilon + \varepsilon_i}$$

where

ϵ : margin of error (to decide original plane)

ε_i : error above the margin