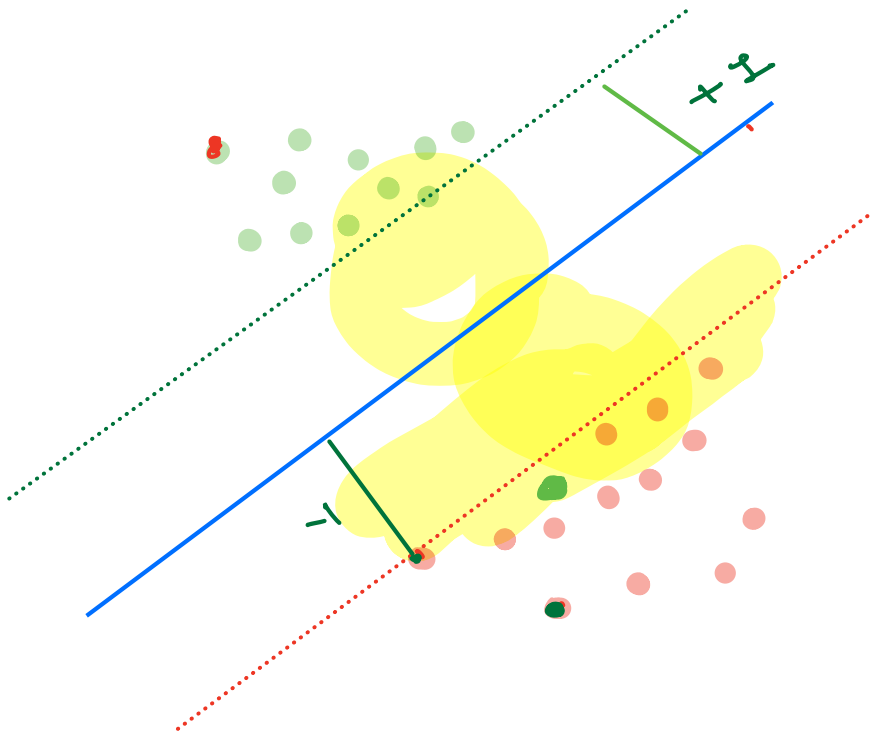
Recap

Hard Margin Classifier

$$\underset{\|w\|}{\operatorname{argmax}}\left(\frac{2}{\|w\|}\right) \quad \text{S.t} \quad \forall i : 1 \to N \quad y_i \times w^T x_i + b \geq 1$$



* Support Vectors are point which are responsible for Margin

* No misclassification allowed

Soft Margin Classifier

$$\min_{w,b} \frac{\|w\|}{2} + \frac{C}{N}\sum_{i=1}^{N} \xi_i$$

$$s.t \quad (w^T x + b) y_i \geq 1 - \xi_i \qquad \forall i : 1 \to N$$

* Optimal values of w and $b \Rightarrow w^*, b^*$

* Primal Form of SVM

* Allows misclassification for a generalized solution

* In optimization theory

Primal - Dual Equivalence

[For every optimization problem there exists a Dual form that gives same results]

Dual form of Soft SVM

Kernel

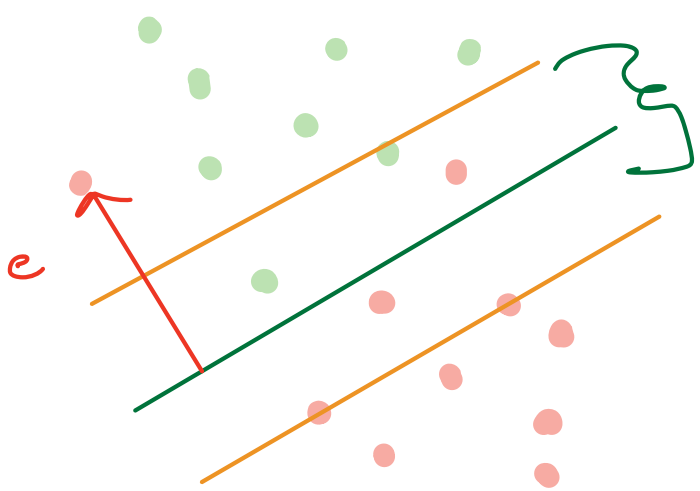
$$\max_{\alpha_i} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \times \alpha_j \times y_i \times y_j \times x_i^T \cdot x_j \right)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C$$

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

- α_i's are your parameter of Model
  i.e. instead of finding w^* and b^* here we are going to find α_i^*
- We are maximizing the dual form
- The input samples always will be in pairs

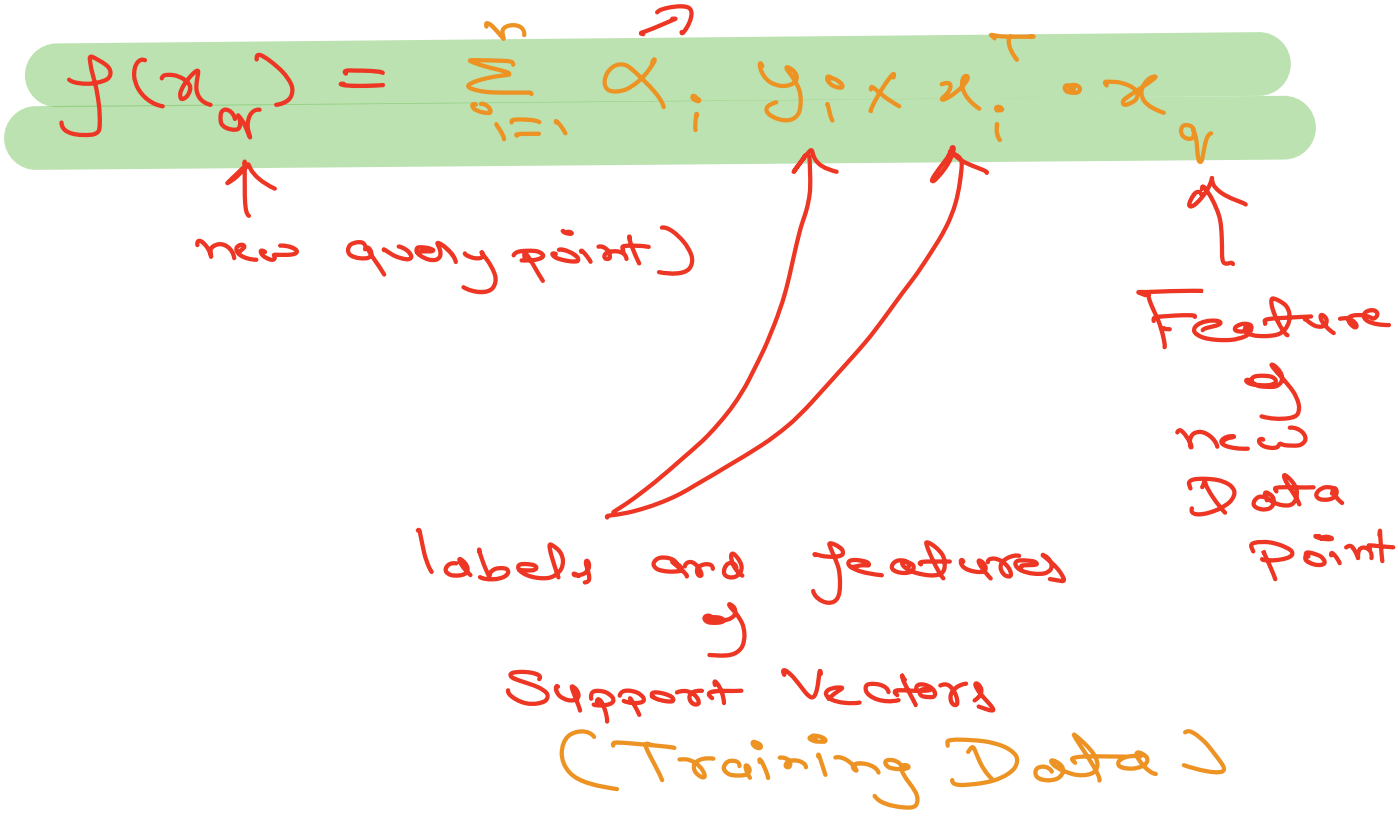
	x_i	y_i
x_1		1
x_2		1
x_3		1
x_n		



Support Vectors :

- Points which are within Margin
- Points which are on the Margin line
- Points which are Mis-classified

Prediction in dual Form



Let's say we have 10k points

- $\alpha_i \rightarrow$ for Non Support Vectors will be 0.
- $\alpha_i > 0$ for Support Vectors

10 k Datapoint → 1 – 2 %

100 – 200 Support Vectors

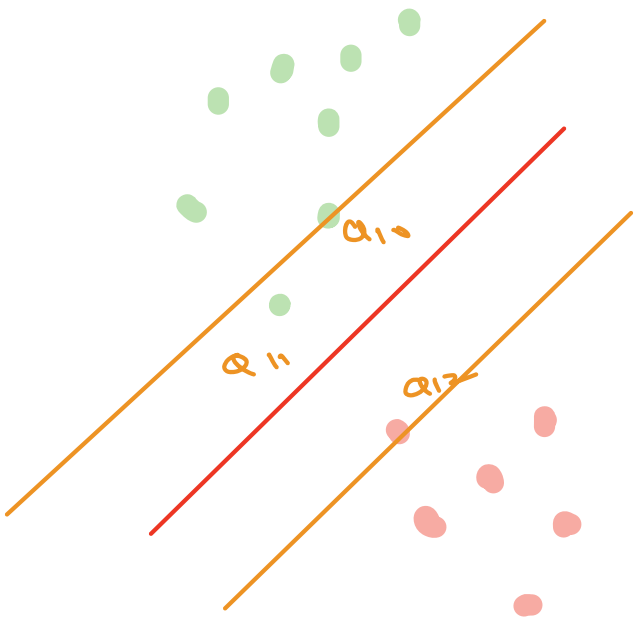
Let's assume we have 100 S.V.

as support Vector

$\alpha_i > 0 \rightarrow$ 100 non-zero

$\alpha_i = 0 \rightarrow$ 9900 zero

Ex:



α_{10} α_{11} α_{12}

$[x_{10}, y_{10}]$ $[x_{11}, y_{11}]$ $[x_{12}, y_{12}]$

ϕ

x_i	y_i
x_1	y_1
x_2	y_2
x_{10}	
x_{11}	
x_n	y_n

$$\hat{y}_q \rightarrow \sum_{i=1}^{n} \alpha_i y_i \vec{x_i} \cdot x_q$$

$$\hat{y} = \alpha_{10} y_{10} \propto x_{10}^{T} \cdot x_q + \alpha_{11} y_{11} \propto x_{11}^{T} \cdot x_q + \alpha_{12} y_{12} \propto x_{12}^{T} \cdot x_q$$

Summary

- α_i for Support Vector
- x_i, y_i for Support Vectors

$K(x_i^T, x_j)$

$$\max_{\alpha_i} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \times \alpha_j \times y_i \times y_j \times x_i^T \cdot x_j \right)$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C$$

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

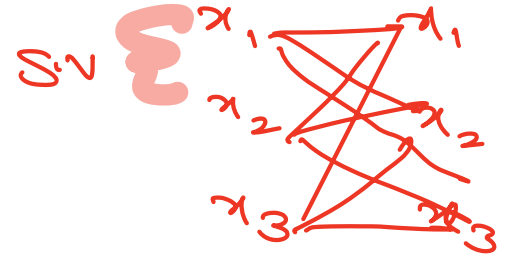
The input samples always will be in pairs

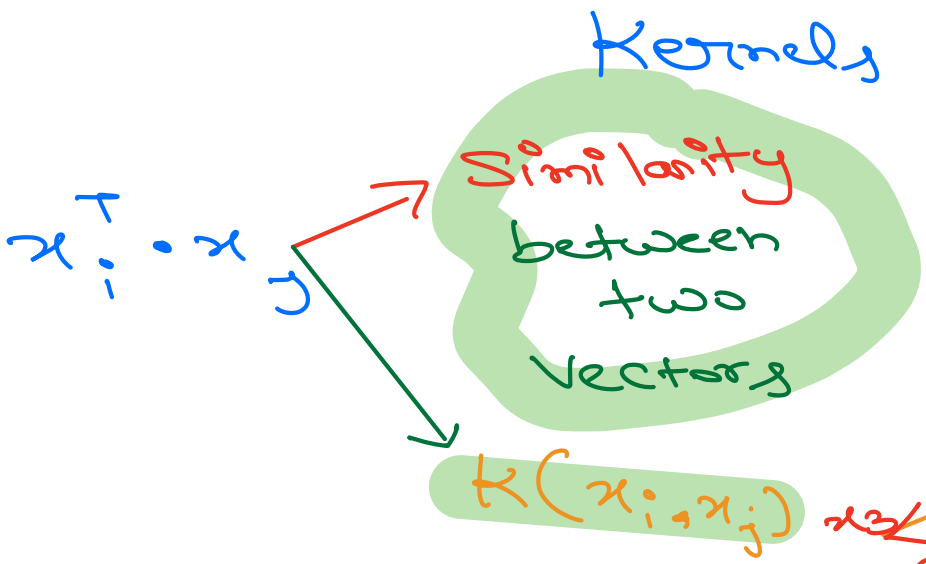
x_i and x_j are all the possible pairs

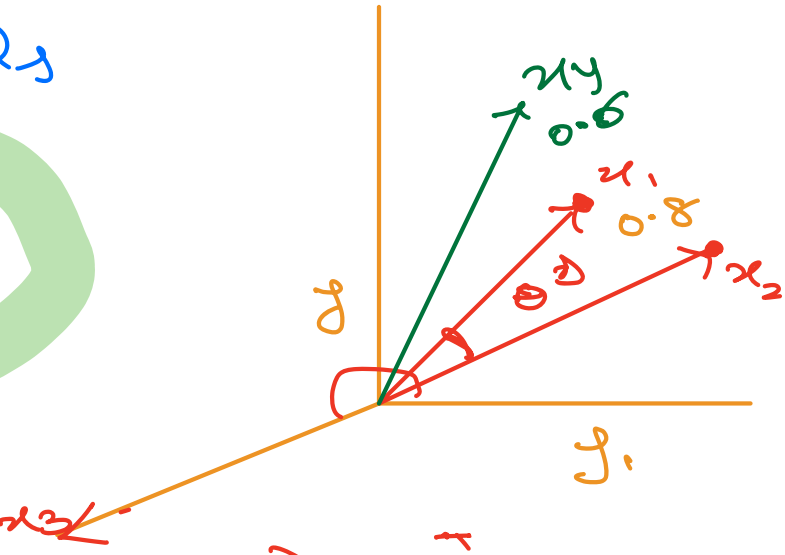
$x_1 - x_2 \quad \alpha > 0$

$x_3 - x_2 \Rightarrow \alpha = 0$

$x_3 - x_1 \Rightarrow \alpha = 0$





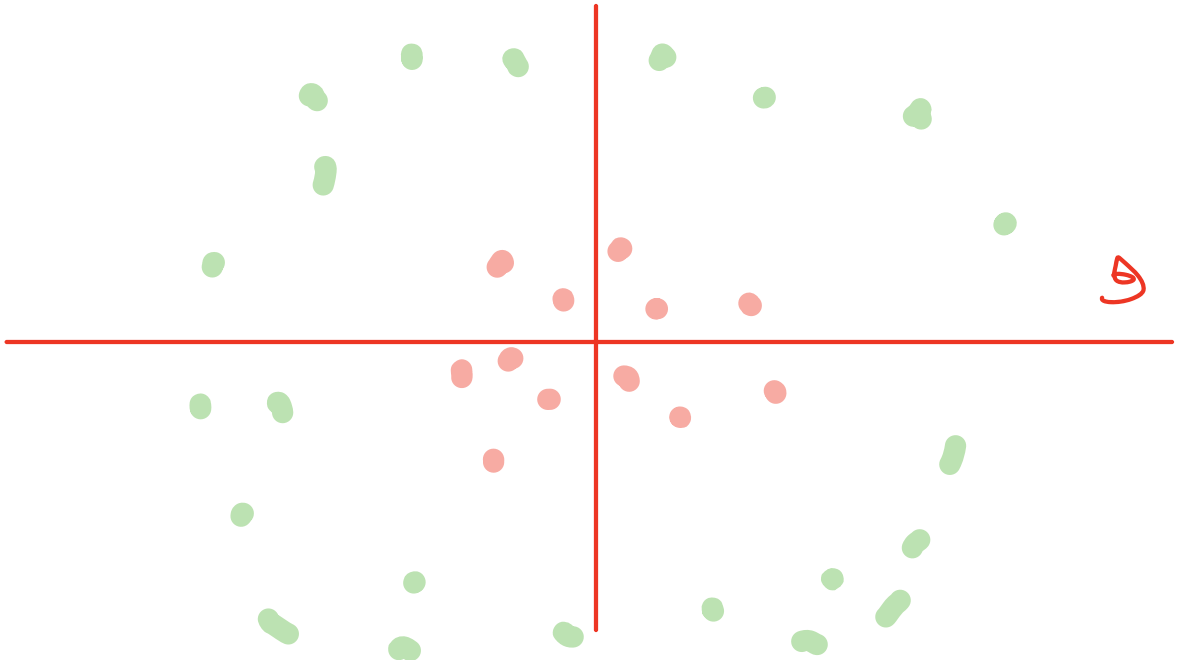


$\cos\theta \Rightarrow \frac{x_1^T \cdot x_2}{\|x_1\| \|x_2\|}$

$\theta \Rightarrow \cos^{-1} \frac{x_1^T \cdot x_2}{\|x_1\| \|x_2\|}$

$\theta_{x_1, x_2} \Rightarrow 0.8$

$\theta_{x_2, x_3} \Rightarrow -1$



$\Rightarrow y_1^2$ and F_2

→ Linear Model (Logistic) or SVM

(manual feature Transformation

Polynomial kernel

$K(x_1, x_2) \Rightarrow (x_1^T x_2 + c)^m$ ← degree

Constant ⇒ 1

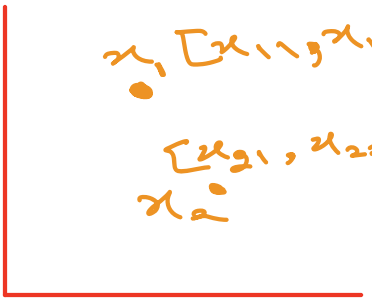
Quadratic kernel ⇒ $(x_1^T \cdot x_2 + 1)^2$

Cubic kernel ⇒ $(x_1^T \cdot x_2 + 1)^3$

Example: Quadratic kernel

$x_1 \Rightarrow [x_{11}, x_{12}]$

$x_2 \Rightarrow [x_{21}, x_{22}]$



$$K(x_1, x_2) = (x_1^T \cdot x_2 + 1)^2$$

$$= (1 + [x_{11}, x_{12}] \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix})^2$$

$$= (1 + x_{11} x_{21} + x_{12} x_{22})^2$$

$$(a+b+c)^2 = a^2 + b^2 + c^2 + 2ab + 2bc + 2ca$$

$$= 1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2 x_{11} x_{21} \times 1$$

$$+ 2 x_{11} x_{21} \times x_{12} x_{22} + 2 \times 1 \times x_{12} x_{22}$$

$$x_1' = [1, x_{11}^2, x_{12}^2, \sqrt{2} x_{11}, \sqrt{2} x_{12}, \sqrt{2} x_{11} x_{12}]$$

$$x_2' = [1, x_{21}^2, x_{22}^2, \sqrt{2} x_{21}, \sqrt{2} x_{22}, \sqrt{2} x_{21} x_{22}]$$

$$K(x_1, x_2) = (x_1'^T \cdot x_2')$$

$(1 + x_1^T \cdot x_2)^2$ → 6 dim Vector

* Kernel trick performs implicit feature transformation

$$2d \longrightarrow K_d \longrightarrow 6d$$

$$\downarrow$$

$$d=2$$

Solving the kernel function is equivalent to finding hyperplane in higher dimension

---

- You need to Tune d

  or

  You can use RBF

RBF

(Radial Basis Function)

$$K_{RBF}(x_1, x_2) = e^{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}}$$

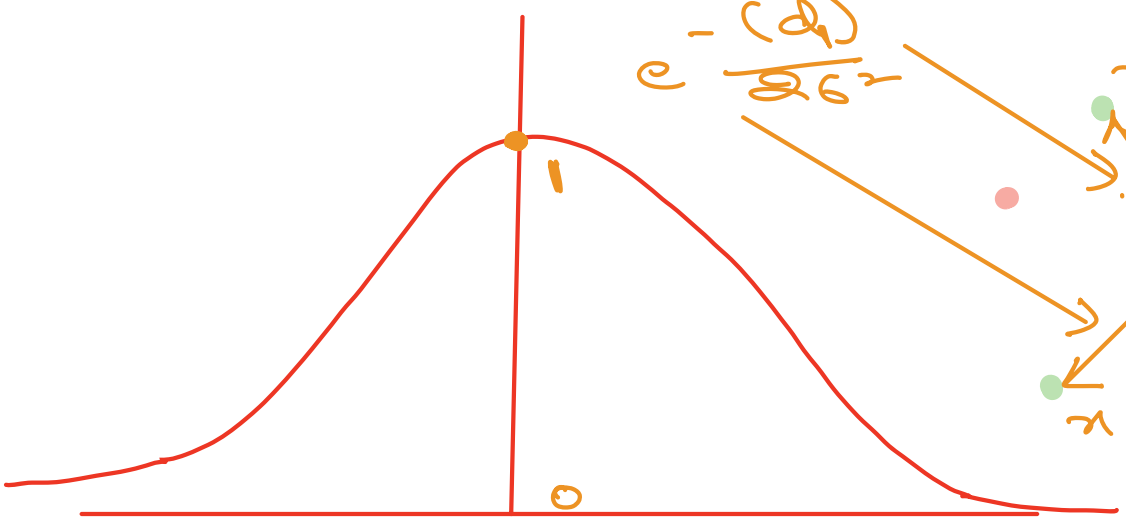
Euclidean Dist

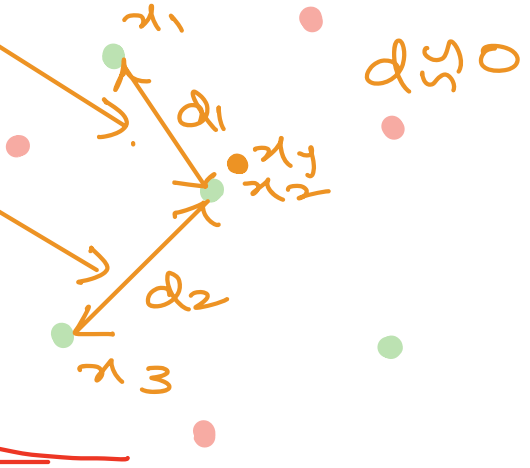
σ ← Hyperparameter

$d_1 < d_2$

$rbf_{x_1, x_2} > rbf_{x_2, x_3}$

$$e^{-\frac{(d_1)^2}{2\sigma^2}}$$





as σ increases the spread of distribution also increases

$rbf(x_2, x_3) \Rightarrow d \approx 0$

$K \Rightarrow e^{-0} \approx 1$

(Point are highly Similar

σ → low

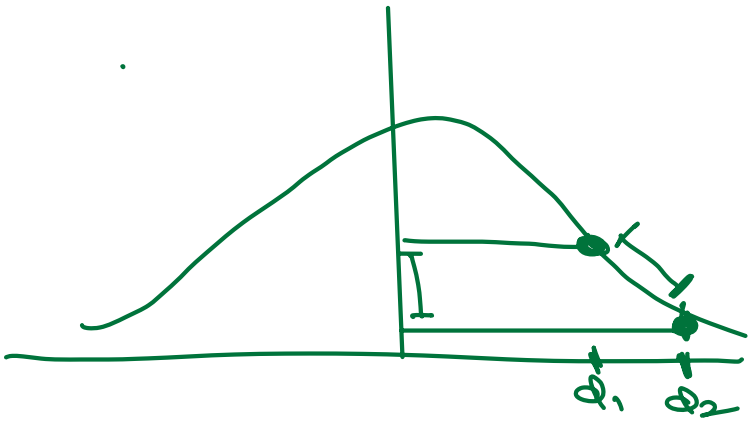
σ, rbf value will not change much

$x_1 \rightarrow x$

$x \longrightarrow x_2$

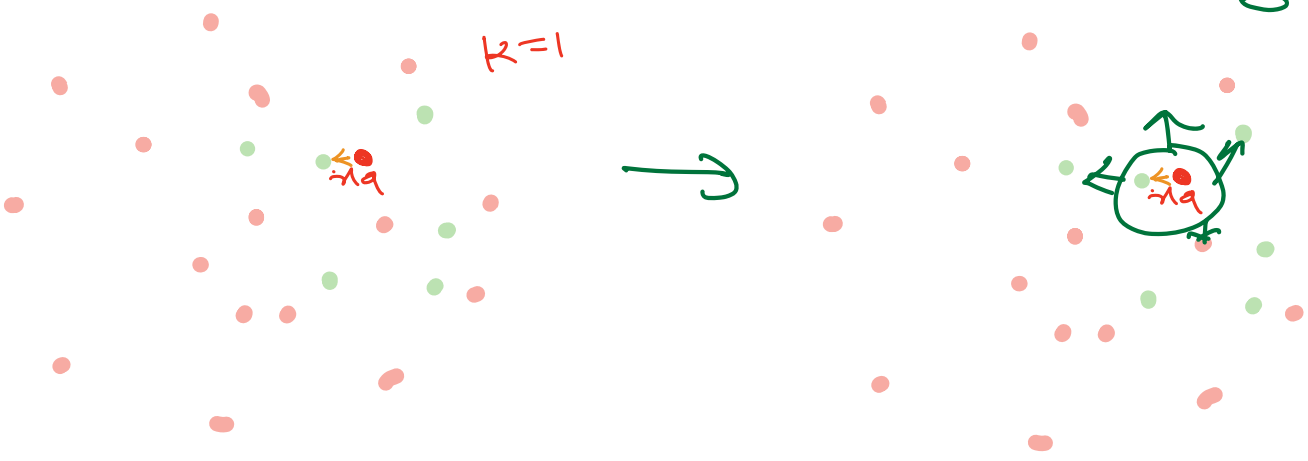
σ → High

Similarity for closer points will appear high



k = ?

σ



Prediction

$$f(x_q) \Rightarrow \sum_{i=1}^{N_s} \alpha_i y_i \; x_i^T \cdot x_q$$

$$\downarrow$$

$$f(x_q) \Rightarrow \sum_{i=1}^{N_s} \alpha_i y_i \; k(x_i^T, x_q)$$

$$\downarrow$$

$\alpha_i > 0$ only for Support Vectors

$O(\text{No of Support Vectors} \times d)$

$O(n \times d)$

SVM will be much faster for predictions than KNN

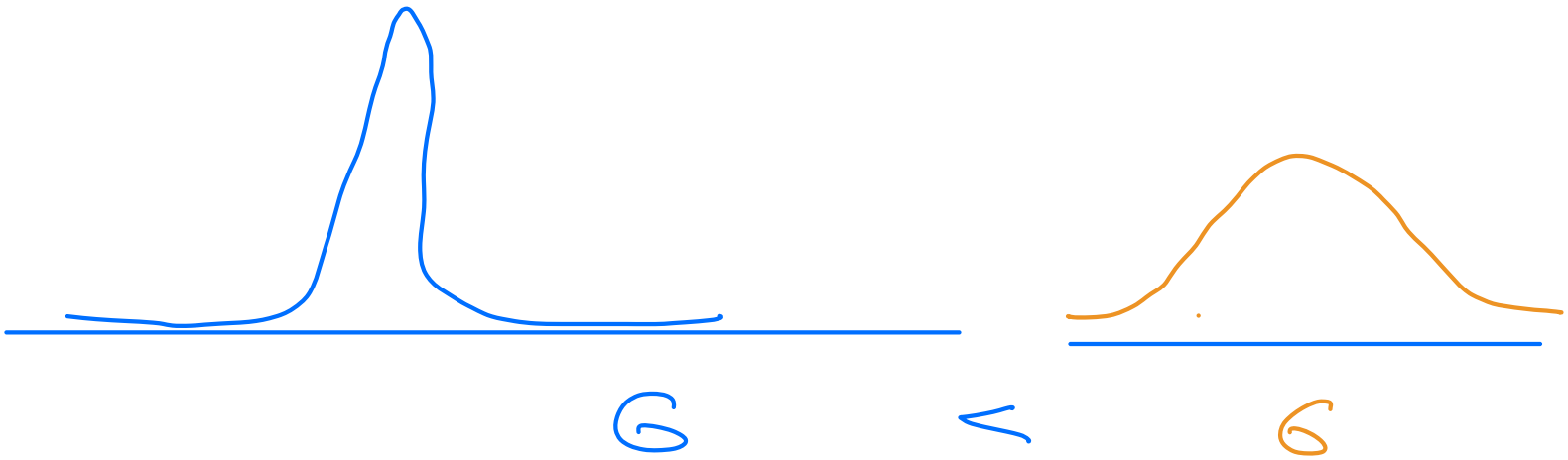
Effect of Outlier

1) If outlier are Non Support Vectors then No problem (Non Kernel SVM)
2) Kernel SVM since we are calculating distance/similarity, they will have higher impact of outliers

Training Time of SVM

$O(n \times n)$ → Very High

→ High Training Time Complexity



G < G