Support Vector Machine

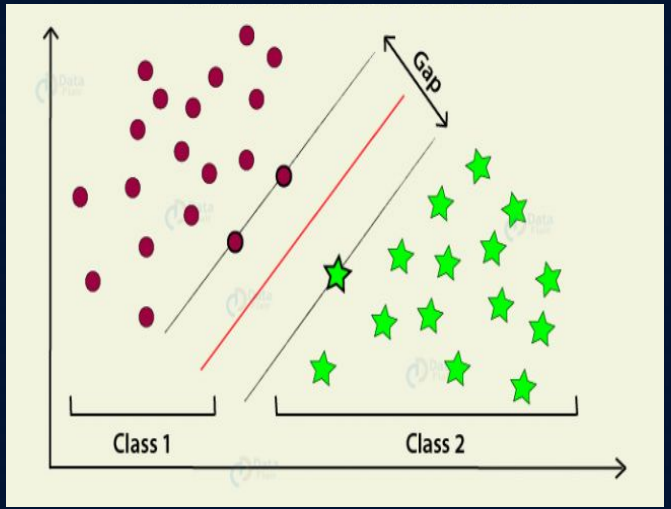
Intuitive Introduction

Group - 7, 8

WHAT IS SVM?

- Support Vector Machine (SVM) is a supervised machine learning algorithm primarily used for classification problems, although it can also handle regression.

- The core idea of SVM is to find the "maximum marginal hyperplane" that best separates the dataset into classes.



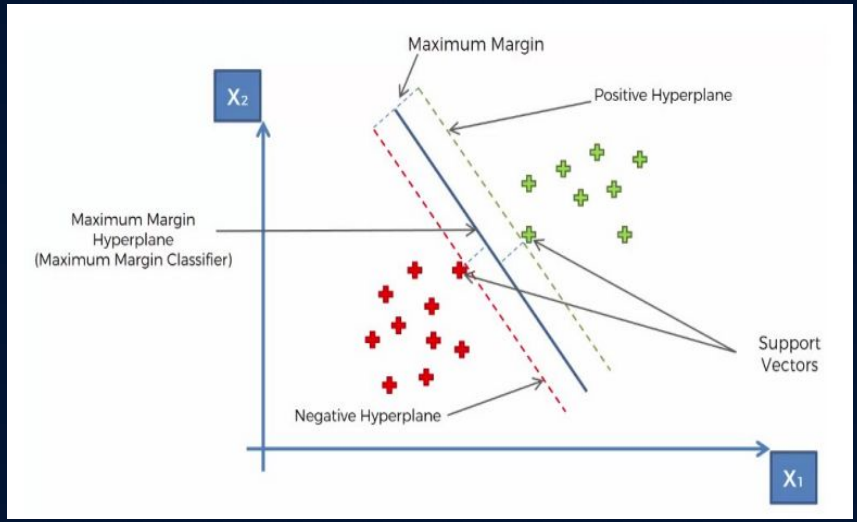
SVM: Core Concepts

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Support Vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins.



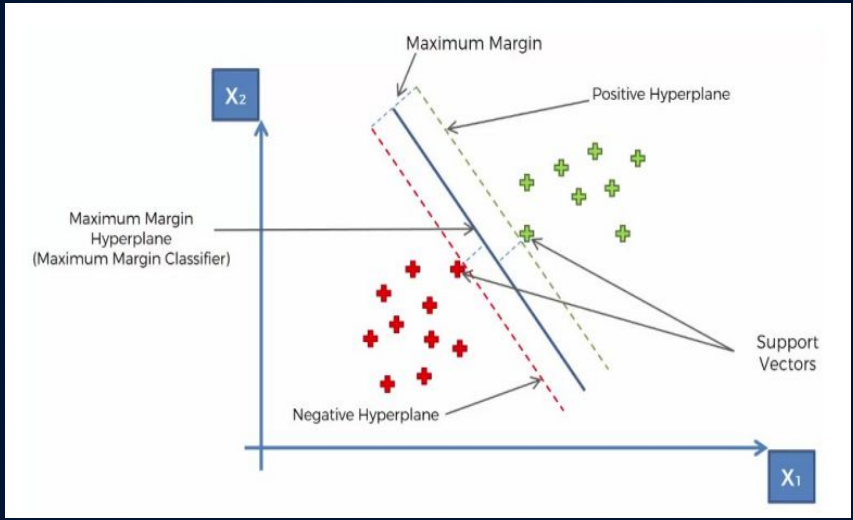
SVM: Core Concepts

Margin

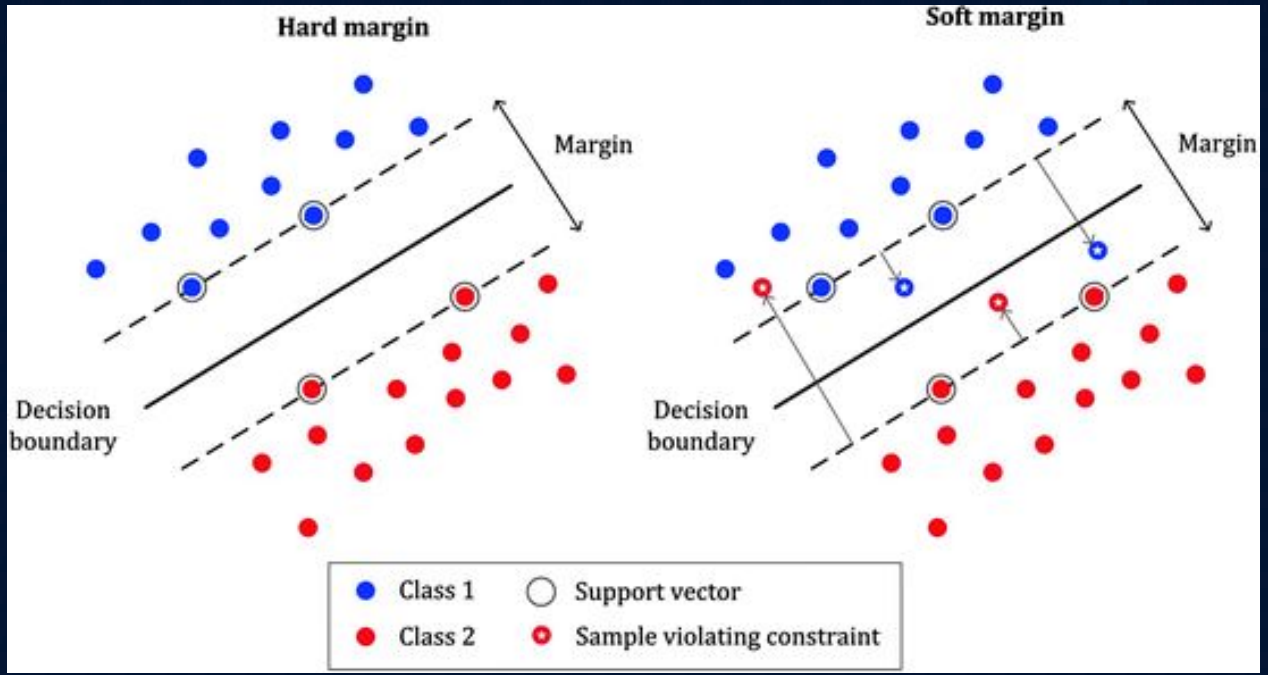
A margin is a gap between the two lines on the closest class points.

This is calculated as the perpendicular distance from the line to support vectors or closest points.

If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.



SVM: Core Concepts



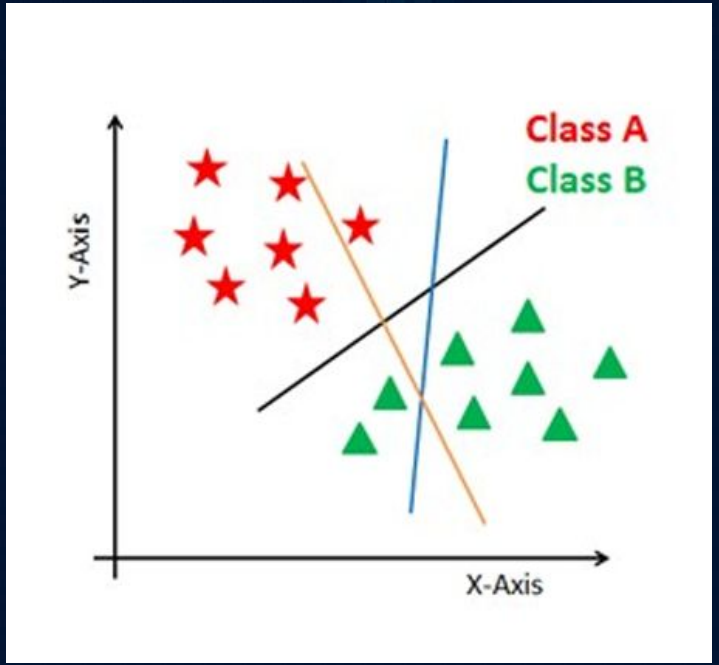
How SVM Works?

1. **Data Plotting**

Plot each data point as a point in N-dimensional space, where N equals the number of features in your dataset.

1. **Hyperplane Selection**

Find all possible hyperplanes that can separate the different classes in the feature space.



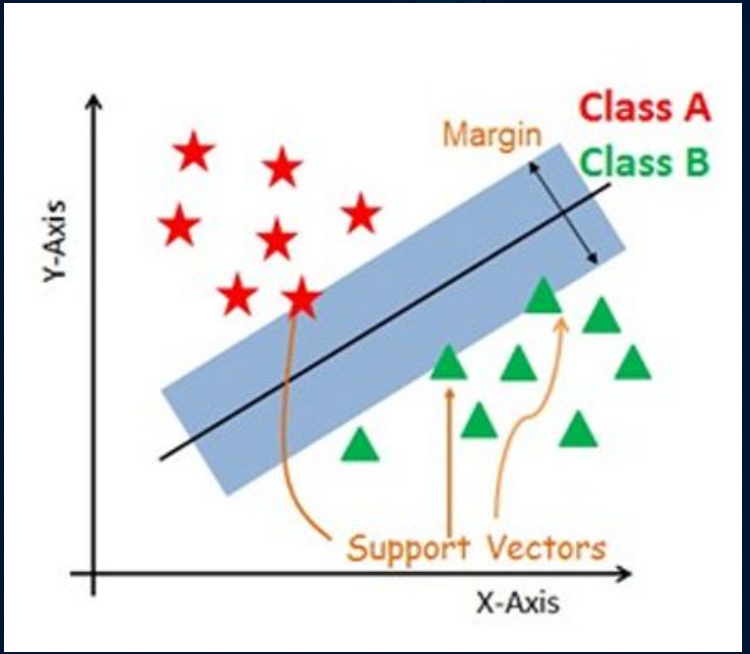
How SVM Works?

3. Margin Maximization

Select the hyperplane that maximizes the margin.

3. Classification

Use the optimal hyperplane to classify new, unseen data points based on which side of the boundary they fall.



The Mathematics of a Simple SVM

Equation of a hyperplane:

$$w^T x + b = 0$$

w: A weight vector perpendicular to a hyperplane.
x: A data point vector.
b: The bias term, which determines the offset of the hyperplane from the origin.

This equation defines the decision boundary.

The sign of the value (wx+b) tells us on which side of the hyperplane a data point x lies.

For class 1, we want wx+b > 0 and for class 2, we want wx+b < 0.

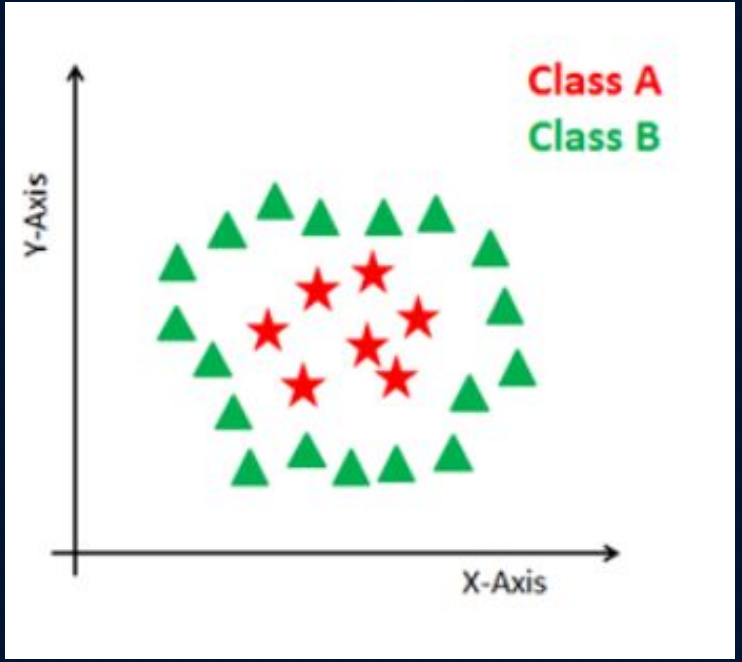
$$\hat{y} = \begin{cases} 1 & : w^T x + b \geq 0 \\ 0 & : w^T x + b < 0 \end{cases}$$

A New Challenge: Non-Linear Data

So far, we've discussed cases where a straight line (or flat hyperplane) can separate the classes.

But what happens when the data is not linearly separable?

For example, what if one class is clustered inside another? A single line simply can't work.



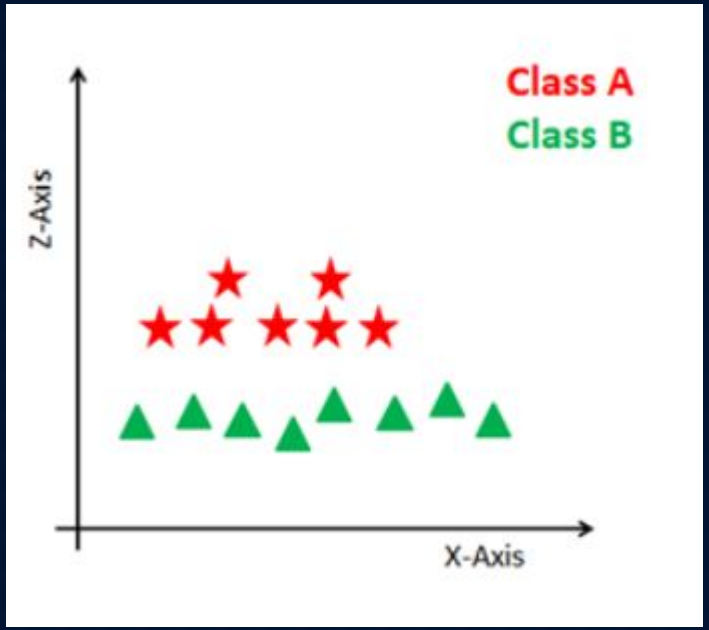
Transforming to a Higher Dimension

The core idea to handle non-linear data is to map the data from its original low-dimensional space into a new, higher dimensional feature space.

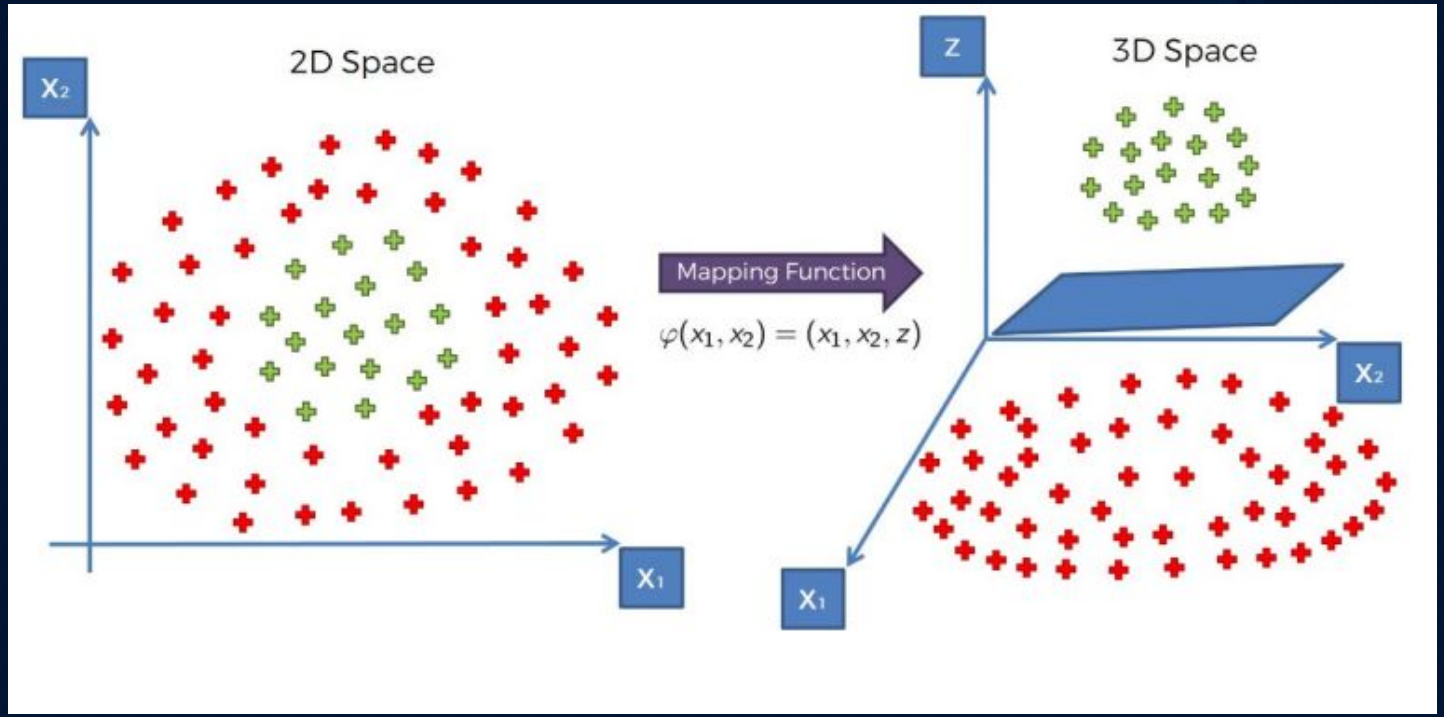
In this higher-dimensional space, the data points might become linearly separable.

For example, the data points are now plotted on the X-axis and Z-axis (where z is the squared sum of both x and y: $z = x^2 + y^2$).

Now, we can easily segregate these points using linear separation.



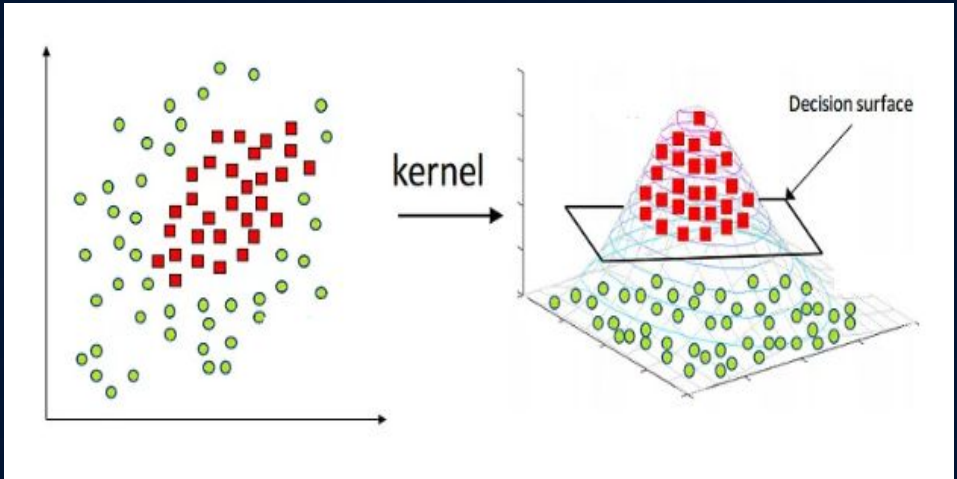
Transforming to a Higher Dimension



SVM Kernels

The SVM algorithm is implemented in practice using a kernel.

A kernel takes a low-dimensional input space and transforms it into a higher dimensional space.



In other words, you can say that it converts a non-separable problem to separable problem by adding more dimensions to it.

Mathematics of Simple SVM

Equation of a hyperplane:

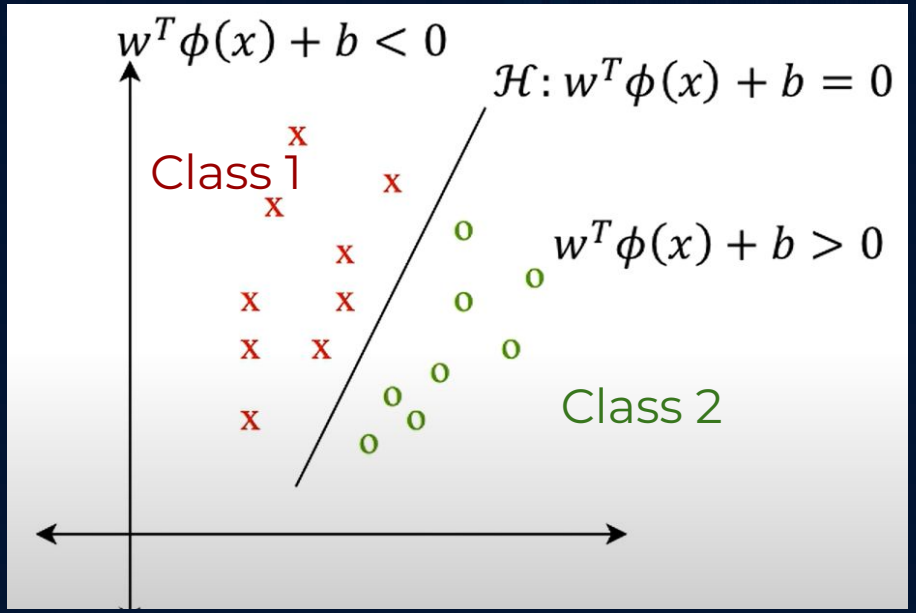
$$\mathcal{H}: w^T\phi(x) + b = 0$$

Kernel Function:

Maps data into a higher-dimensional feature space where linear separation becomes possible

$$x \in \mathbb{R}^D$$

$$\phi: \mathbb{R}^D \to \mathbb{R}^M \quad \phi(x) \in \mathbb{R}^M$$



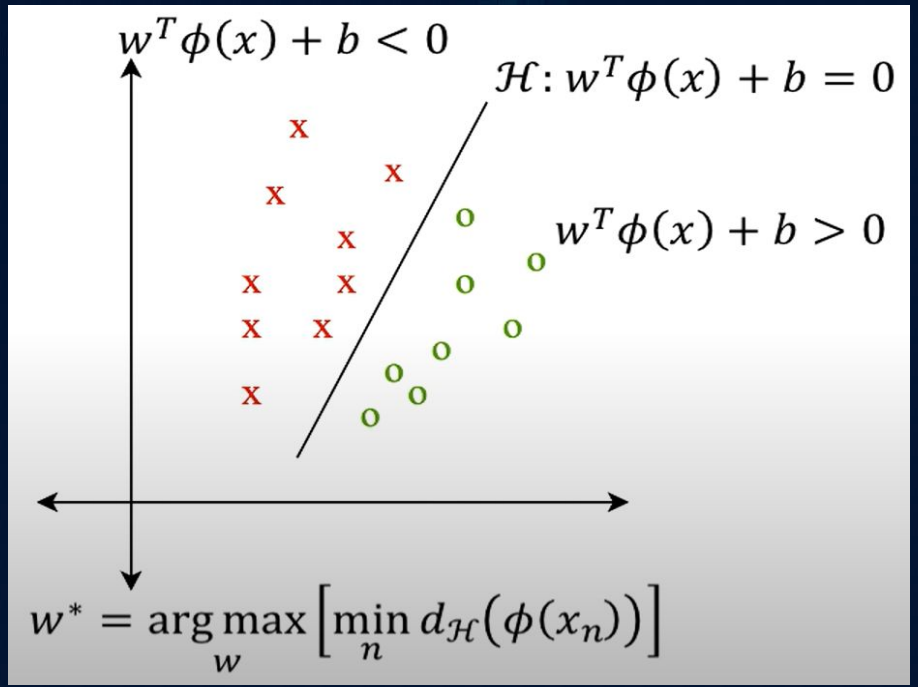
Mathematics of a Simple SVM

The Optimal Hyperplane

- We want to find the hyperplane that has the largest minimum distance to the training example
- This becomes an optimization problem:

where,

$$d_{\mathcal{H}}(\phi(x_0)) = \frac{|w^T\phi(x_0) + b|}{\|w\|_2}$$



Mathematics of a Simple SVM

Perfect separation

(Hard Margin)

$$w^* = \arg\max_w \frac{1}{\|w\|_2} \left[\min_n y_n[w^T\phi(x_n) + b]\right]$$

Let $\min_n y_n[w^T\phi(x_n) + b] = 1$

$$w^* = \arg\max_w \frac{1}{\|w\|_2}$$

$$s.t \quad \min_n y_n[w^T\phi(x_n) + b] = 1$$

Maximization

Primal Form of SVM

$$\min_w \frac{1}{2}||w||_2$$

$$s.t \quad y_n[w^T\phi(x_n) + b] \geq 1 \quad \forall n$$

Minimization

Mathematics of a Simple SVM

Non- Perfect separation

(Soft Margin)

C = 0 -> less complex boundary

C = infinity -> more complex boundary (no misclassifications allowed)

~~$y_n[w^T\phi(x_n)+b] > 0 \ \forall n$~~

$$y_n[w^T\phi(x_n)+b] \leq 0 \ \ \exists n$$

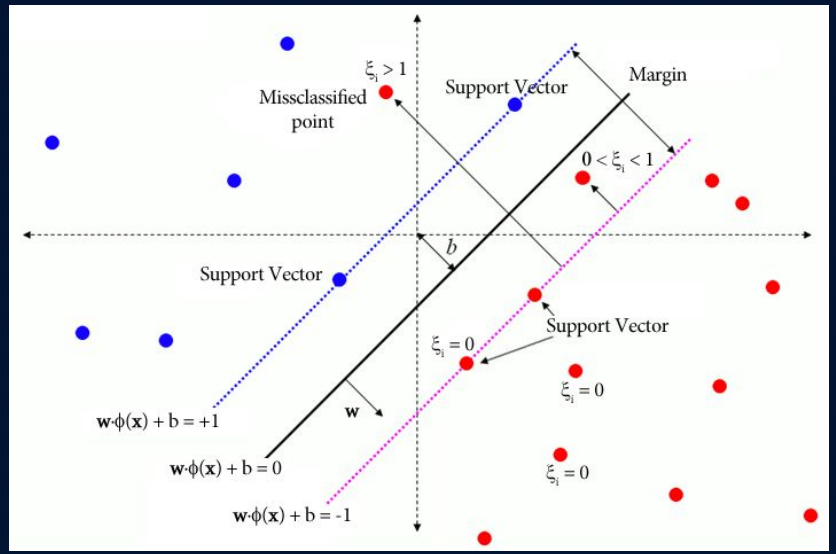
New Primal Form of SVM

$$\min_{w,b,\{\xi_n\}} \frac{1}{2}||w||_2 + C\sum_n \xi_n$$

$$s.t \ \ y_n[w^T\phi(x_n)+b] \geq 1-\xi_n \ \ \forall n$$

$$\xi_n \geq 0 \ \forall n$$

Mathematics of a Simple SVM



~~$y_n[w^T\phi(x_n) + b] > 0 \ \forall n$~~

$$y_n[w^T\phi(x_n) + b] \leq 0 \ \exists n$$

New Primal Form of SVM

$$\min_{w,b,\{\xi_n\}} \frac{1}{2}||w||_2 + C\sum_n \xi_n$$

$$s.t \ y_n[w^T\phi(x_n) + b] \geq 1 - \xi_n \ \forall n$$

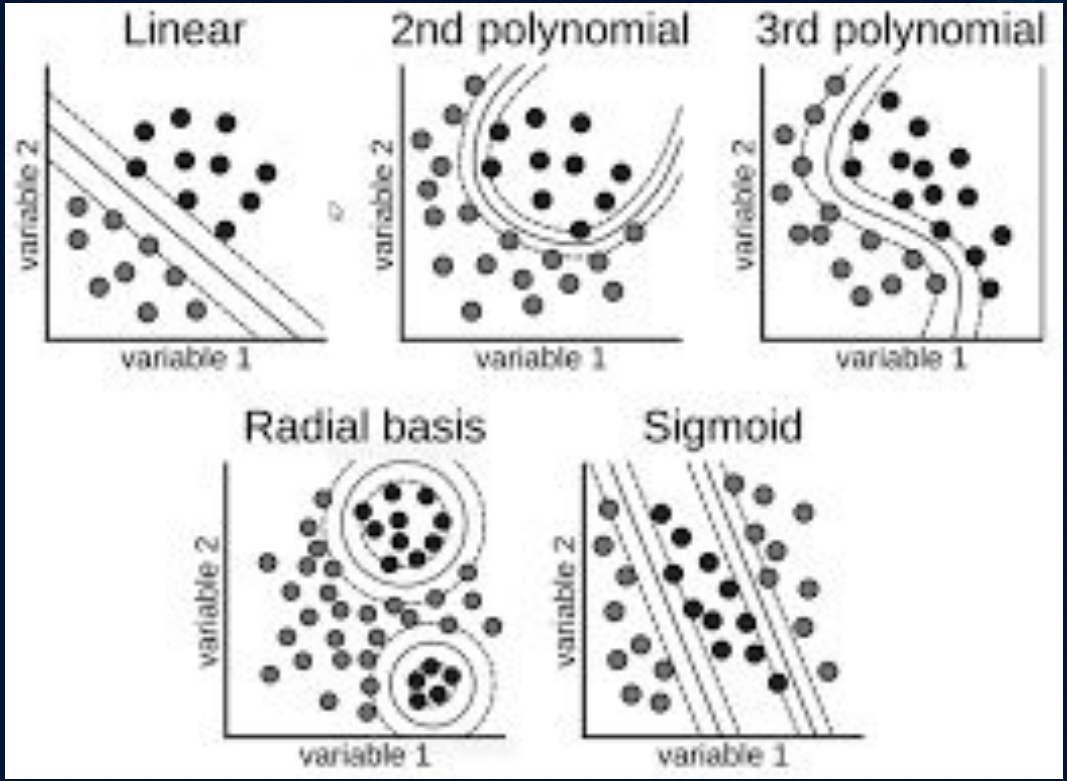
$$\xi_n \geq 0 \ \forall n$$

Mathematics of a Simple SVM

$$\underbrace{\min \|w\|^2}_{\text{regularization term}} + \underbrace{C \sum_{i=1}^{l} \max\left(0, 1 - y_i f(x_i)\right)}_{\text{loss function term}}$$

Hinge Loss

Common Kernel Functions



Support Vector Regression (SVR)

Definition of SVR

Support Vector Regression (SVR) is a supervised learning technique based on Support Vector Machines (SVM). It is used to estimate the relationship between a **dependent variable** (output) and one or more **independent variables** (inputs).

- Unlike ordinary regression, SVR tries to fit a function within a margin of tolerance (ε), making it less sensitive to outliers

The Concept of Support Vector Regression

- **Support Vector Regression (SVR)** is an extension of **Support Vector Machines (SVM)** for regression problems.
- Instead of finding a separating hyperplane (like in classification), SVR tries to fit the data within a **tube** (margin) of size **±ε** around the regression line.
- Data points inside this ε-tube are considered “well predicted” and do not contribute to the loss
- Only the points **outside** the ε-tube (support vectors) influence the model.

Key Concepts:

1. Hyperplane (f(x))

The regression function is a straight line (for linear SVR) or curve (for non-linear SVR).

$$f(x)= W^T X+ b$$

2. ε-Insensitive Margin (Tube)

- A tolerance range (±ε) is drawn around the regression function.
- Errors smaller than ε are **ignored**.
- This makes SVR less sensitive to small fluctuations/noise.

3. Support Vectors

- Data points lying **on or outside** the ε-margin.
- They determine the final position of the regression hyperplane.

4. Slack Variables & Penalty (C)

- If points fall outside the ε-tube, slack variables measure the error.
- Parameter **C** controls the trade-off between margin width and tolerance for errors

How SVR Works

Imagine a dataset of points roughly aligned in a line:

Step 1: Fit a regression line (hyperplane).

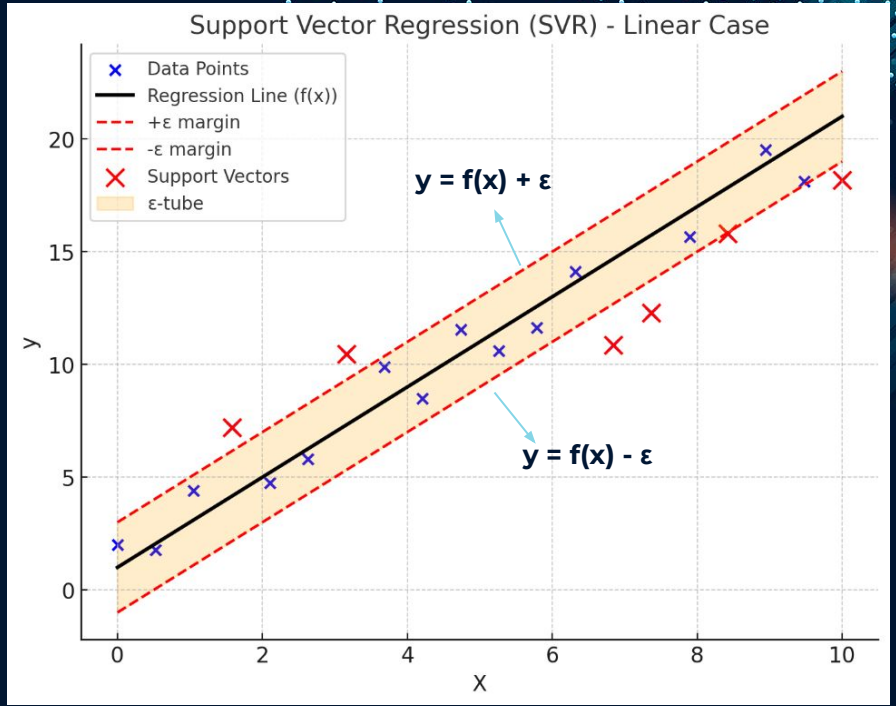
Step 2: Draw two parallel boundary lines at distance **+ε and -ε** from the regression line.

- These form the **ε-tube (margin)**.

Step 3: Identify support vectors — the points outside or on the margin.

Step 4: Optimize W and b such that the margin is maximized while keeping most data points inside the tube.

$$f(x) = W^T X + b$$



The main SVR algorithms

01

SVR Primal with Hard Margin

(Linear Problems)

02

SVR Primal with Soft Margin

(Linear Problems)

03

SVR Dual with Soft Margin

(Linear Problems)

04

SVR Dual with Kernel Trick

(Non-Linear Problems)

SVR Primal with Hard Margin

(Linear Problems)

In this formulation, there are as many constraints as data points and number of decision variables is equal the number of features plus one. Also in this formulation **no data points can lay on the margin area.**

Concepts:

1. Key Definitions

- xi = The independent variables, where $xi \in X$
- yi = The dependent variable
- wi = It is a vector normal to the hyperplane, where $wi \in W$
- ε = Maximum allowed deviation value
- b = The bias parameter that indicates how shifted the hyperplane is from the origin

2. Hyperplane Formulation

- Regression Function

$$f(x) = W^T X + b$$

- Constraints:

$$yi - (W^T X + b) \leq \varepsilon$$
$$(W^T X + b) - yi \leq \varepsilon$$

- Ensures all points are inside the ε-tube.

3. Optimization Problem

- Objective: Maximize the margin $d = 2\varepsilon / \| w \|$
- Equivalent to minimizing $\frac{1}{2} \| w \|^2$
- Final optimization problem

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

Subject to:

$$y_i - (W^T X + b) \leq \varepsilon \, , \, (W^T X + b) - y_i \leq \varepsilon$$

4. Geometric Interpretation

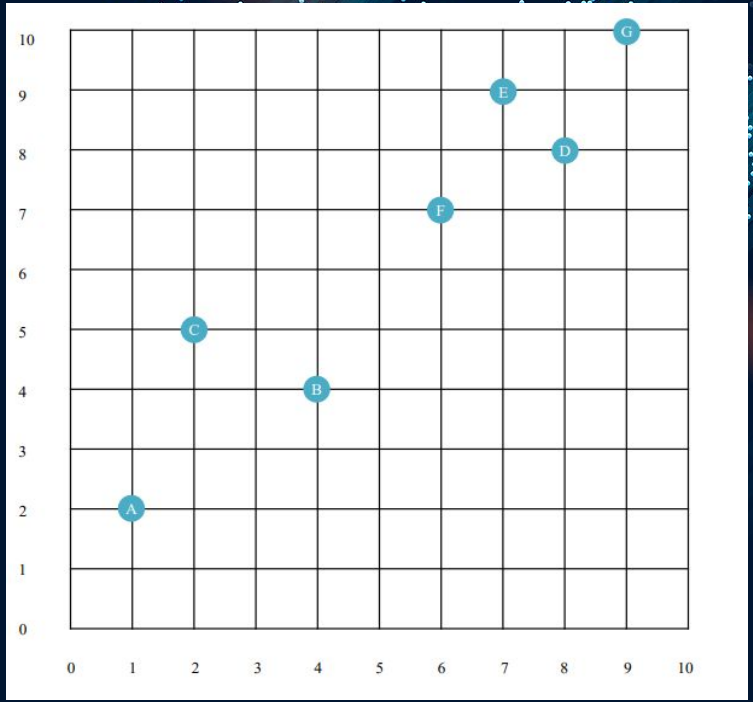
- ε-tube defines a tolerance band around the regression line.
- Support vectors: Points lying exactly on the ±ε boundaries.
- No point is allowed outside the tube in **Hard Margin**.
- Prediction:

$$Y_i = (W^T X + b)$$

Example:

> Dataset

ID	X	Y
A	1	2
B	4	4
C	2	5
D	8	8
E	7	9
F	6	7
G	9	10



- Tolerance parameter:

$$\varepsilon = 1.5$$

- Goal: Find w and b so all points lie inside the ε-tube.

Solution:

- **Optimization problem solved**

$$\min \tfrac{1}{2} \| w \|^2$$

$$\text{s.t.} \quad y_i - (W^T X + b) \leq 1.5$$

$$(W^T X + b) - y_i \leq 1.5$$

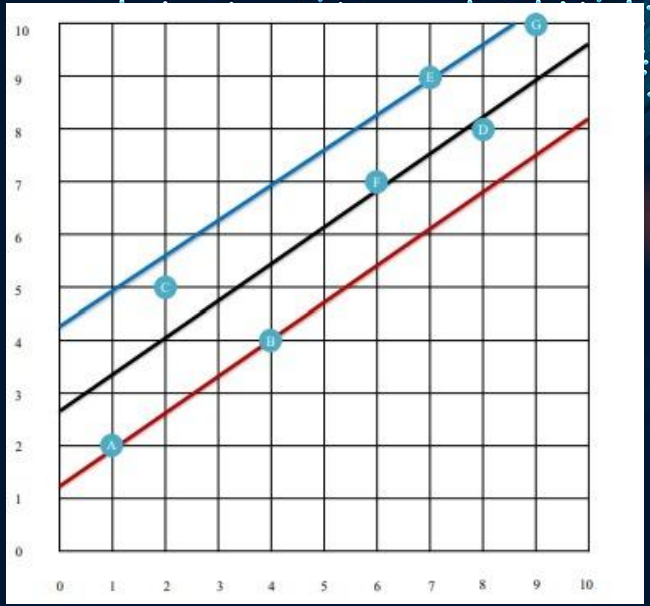
- **Result**

$w \approx 0.67$

$b \approx 2.83$

- **Prediction line:**

$$Y_i = 0.67 x + 2.83$$



- **Points A,E,B act as support vectors (on ε-boundary)**

SVR Primal with Soft Margin

(Linear Problems)

In this formulation, there are as many constraints as data points and number of decision variables is equal twice the number of features plus one. Also in this formulation some data points **can** lay on the margin area

Concepts:

1. Key Definition

- xi = The independent variables, where $xi \in X$
- yi = The dependent variable
- wi = It is a vector normal to the hyperplane, where $wi \in W$
- ε = Maximum allowed deviation value
- b = The bias parameter that indicates how shifted the hyperplane is from the origin

2. Soft Margins

In real life, some points will not fit inside the ε-tube. To allow this, we introduce slack variables:

- $\xi^{+} \geq 0$: amount by which prediction exceeds the upper ε-bound.
- $\xi^{-} \geq 0$: amount by which prediction falls below lower ε-bound.

3. Constraints with Slack Variable

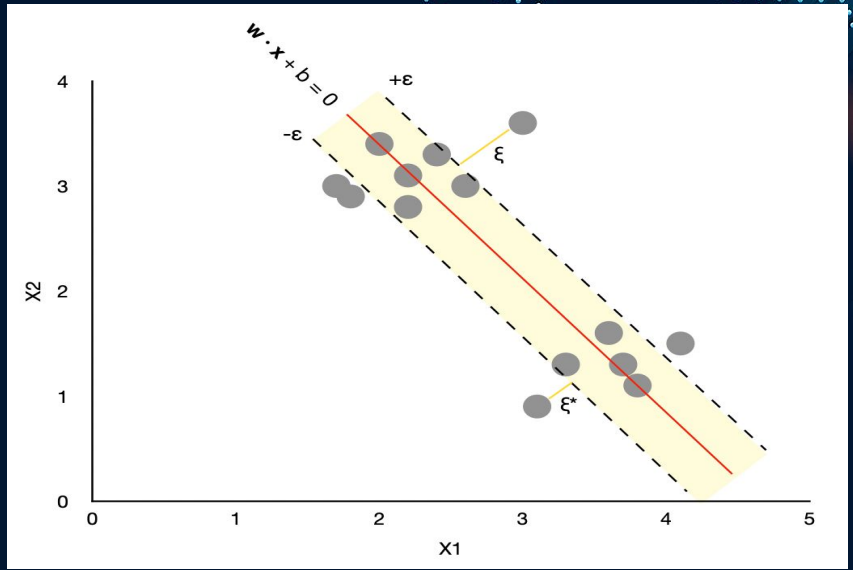
$$y_i - (w^T x_i + b) \leq \varepsilon + \xi^+$$

$$(w^T x_i + b) - y_i \leq \varepsilon + \xi^-$$

$$\xi^+ \geq 0, \qquad \xi^- \geq 0$$

4. Objective Function

$$\text{Min } (\tfrac{1}{2} ||w||^2 + C \sum_i (\xi^+ + \xi^-))$$



Example:

> Dataset

ID	X	Y
A	1	2
B	6	8
C	5	5
D	3	4
E	5	9
F	4	6
G	7	10

Epsilon (ε) = 1.5 , C = 1.0

Objective Function:

$$\text{Min } (\tfrac{1}{2} ||w||^2 + 1 \times \sum_i (\xi^+ i + \xi^- i))$$

s.t. $y_i - (w^T x_i + b) \le 1.5 + \xi^+$

$(w^T x_i + b) - y_i \le 1.5 + \xi^-$

$\xi^+ \ge 0, \quad \xi^- \ge 0$

Per-point primal slack split into nonnegative parts:

- For each data point ‘i’ we computed:
- $\xi^+_i = \max(0, y_i - f(x_i) - \varepsilon)$
- $\xi^-_i = \max(0, f(x_i) - y_i - \varepsilon)$

Continued:

Result: Slack variables (ξ^+, ξ^-):

$\xi^+ = [0, 0, 0, 0, 0.67, 0, 0]$

$\xi^- = [0, 0, 0.33, 0, 0, 0, 0, 0]$

Primal variables (from fit):

$w = 0.83$

$b = 2.67$

Norm term:

$\frac{1}{2} \; // w // ^2 = 0.34445$

Sum of slack terms:

$\sum_i (\xi^+_i + \xi^-_i) = 0.33 + 0.67 = 1$

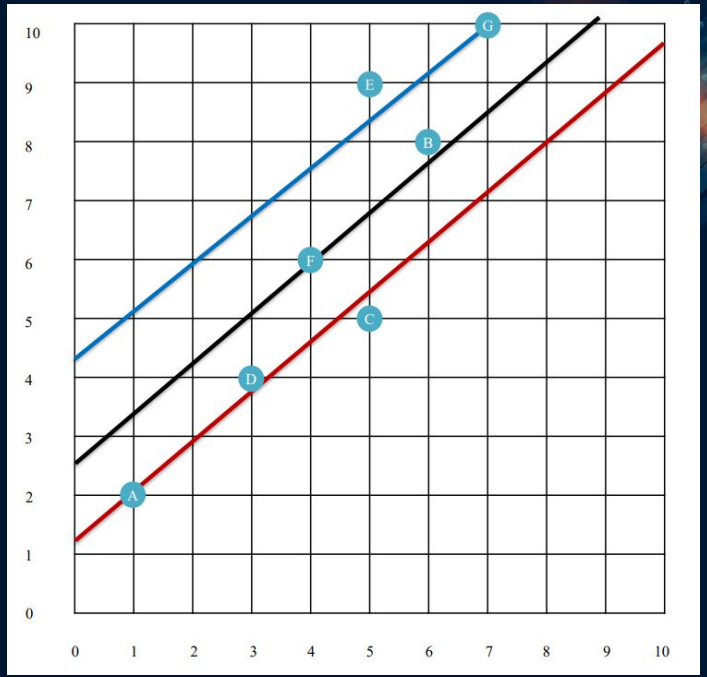
Primal objective value:

$\frac{1}{2} ||w||^2 + C \sum (\xi i + \xi i^*) \approx 0.3 + 1.0 \times 1 = 0.34445$

Resultant Line:

$Y - 0.83X - 2.67 = 0$

A and G are support vectors.



SVR Dual with Soft Margin

(Linear Problems)

In this formulation, there is basically one constraint and there are as many decision variables as data points. Also in this formulation some data points can lay on the margin area, and the inner product of the datapoints is used instead of the datapoint values.

1. Introduction

Dual formulation of ε-SVR with soft margin. That's the step where we introduce Lagrange multipliers and end up with a kernel-friendly optimization problem.

2. Dual Formation

We introduce Lagrange multipliers:

- $\alpha_i, \alpha_i^* \geq 0$ for the first two constraints.
- $\eta_i, \eta_i^* \geq$ for slack non-negativity

After derivation, we get the dual optimization:

$$\max_{\alpha_i, \alpha_i^*} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*)$$

Subject to:

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0$$

$$0 \leq \alpha_i, \alpha_i^* \leq C$$

Regression Function in Dual Form

Once solved, the regression function becomes:

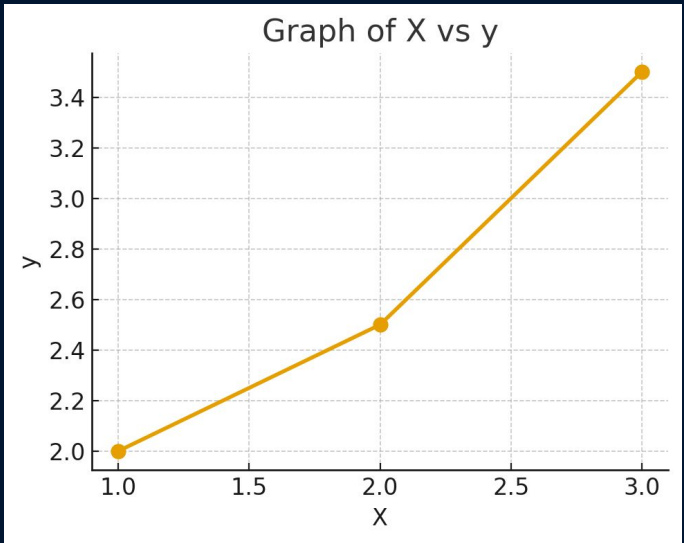
$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b$$

And with **kernels** $K(x_i, x_j)$:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

Example:

> Dataset



X=[1, 2, 3]
y=[2.0, 2.5, 3.5]

Primal variables (from fit):
Epsilon (ε) = 0.1 C = 1.0

Dual solution (variables)

- α=[0.1250000181, 0.0, 0.8749999819]
- α^*=[0.0, 1.0, 0.0]

So the differences $u_i=\alpha_i-\alpha_i^*u_i$ =

u=[0.1250000181, −1.0, 0.8749999819]

(You can see $\sum_i u_i=0$ - the equality constraints holds)

Continued:

Recovered primal parameters

- $w=\sum_i(\alpha_i-\alpha_i^*)x_i = 0.75 (\approx 0.749999964)$
- b (computed from KKT conditions using free support multipliers) ≈ 1.15

So the regression function is:

$f'(x)=0.75x+1.15$

Predictions at training x:

- f(1) ≈ 1.9000000362
- f(2) ≈ 2.65
- f(3) ≈ 3.4000000

Primal slacks (ξ): computed as $\max(0, |y-f(x)|-\epsilon)$

- [0.0, 0.05, ≈ 3.6e-08] → effectively only the middle point has a tiny slack of 0.05.

Continued:

Objective values:

- Primal objective:

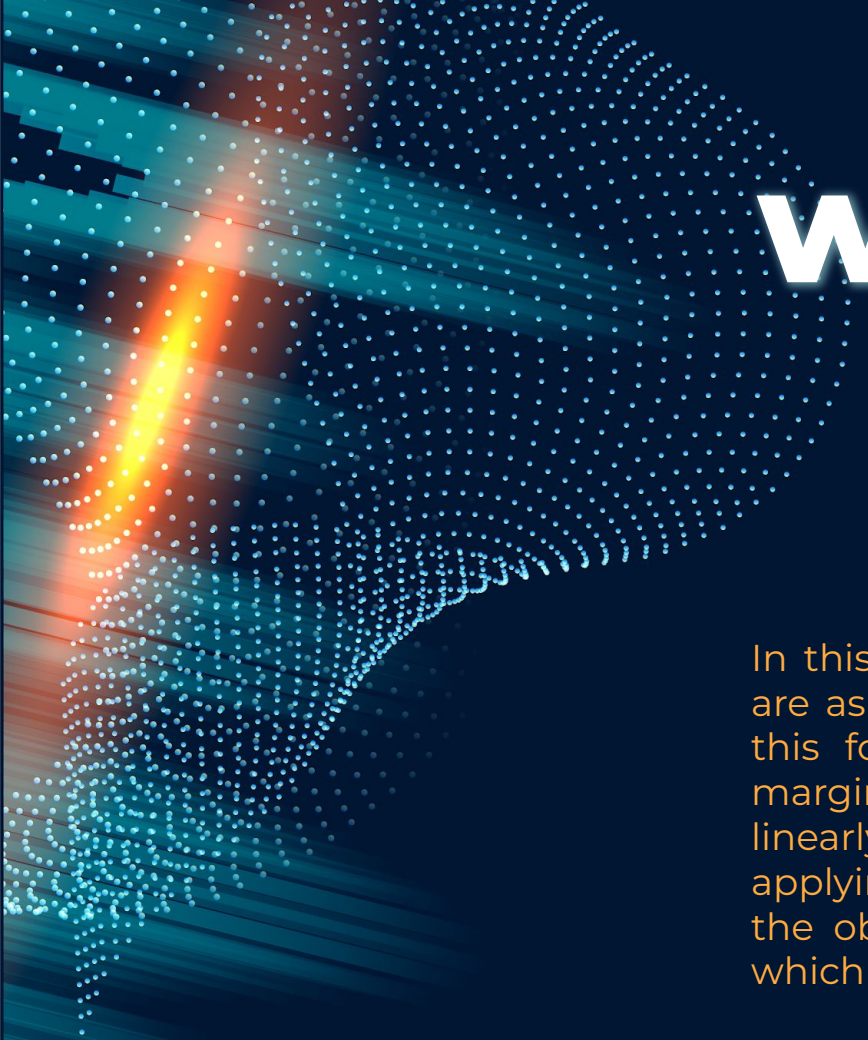
  $\frac{1}{2} \|w\|^2 + C\sum_i \xi_i \approx 0.3312500090$

Interpretation

- Some dual variables hit the bounds ($\alpha^*_2 = 1.0$ exactly), meaning that sample 2 is playing the role of a boundary/violator with α^* at C. The first and third points have nonzero α's and together shape the final w.

- The recovered line $0.75x+1.150.75x + 1.150.75x+1.15$ is the same thing you would get if you solved the primal directly — dual/primal match via KKT.

- The small slack at point 2 indicates it's slightly outside the ε-tube (by 0.05).



SVR Dual with Kernel Trick

(Non-Linear Problems)

In this formulation, there is one constraint and there are as many decision variables as data points. Also in this formulation some data points can lay on the margin area, and the Kernel Trick, which transforms linearly inseparable data to linearly separable ones (by applying a kernel function on each datapoint to map the observations into a higher-dimensional space in which they become separable), is used.

Why Kernel Trick?

- It extends SVR to handle non-linear relationships that cannot be fitted with a straight line or plane.
- Linear SVR works well when the relation between input and output is nearly linear, but in real-world datasets the patterns are usually non-linear.
- The kernel trick maps data into a higher-dimensional space where these patterns become linearly separable.
- This allows us to achieve non-linear regression without explicitly computing high-dimensional features.

Before writing the optimization problem, we define the notation used:

- x_i = input vector (independent variable)
- y_i = output value (dependent variable)
- $K(x_i, x_j)$ = kernel function applied to two data points
- a_i, a_i = Lagrange multipliers for upper and lower constraints
- ε = tolerance (epsilon-insensitive tube width)
- C = penalty parameter (controls margin violations)
- b = bias term (offset of the regression function)
- n = number of training samples

Concept

- The kernel trick allows SVR to use similarity functions instead of explicit dot products.
- This makes it possible to solve non-linear regression by replacing the inner product with a kernel function.
- The prediction function in kernel SVR becomes:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

Optimization with Kernel

- In the dual formulation, the optimization problem is rewritten in terms of the kernel function.
- We maximize the following objective under certain constraints:

Normal formula:

penalty for points outside the ϵ-tube.

$$\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) - \varepsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} y_i(\alpha_i - \alpha_i^*)$$

alignment with true outputs yi

Constraints:

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0, \quad 0 \le \alpha_i, \alpha_i^* \le C$$

Common Kernels

- Different kernel functions can be used depending on the data.
- Each kernel defines a different way of measuring similarity between data points.

Linear Kernel: $$K(x_i, x_j) = x_i^\top x_j$$

Polynomial Kernel: $$K(x_i, x_j) = (x_i^\top x_j + c)^d$$

RBF (Gaussian) Kernel: $$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

Sigmoid Kernel: $$K(x_i, x_j) = \tanh(\kappa x_i^\top x_j + \theta)$$

Example:

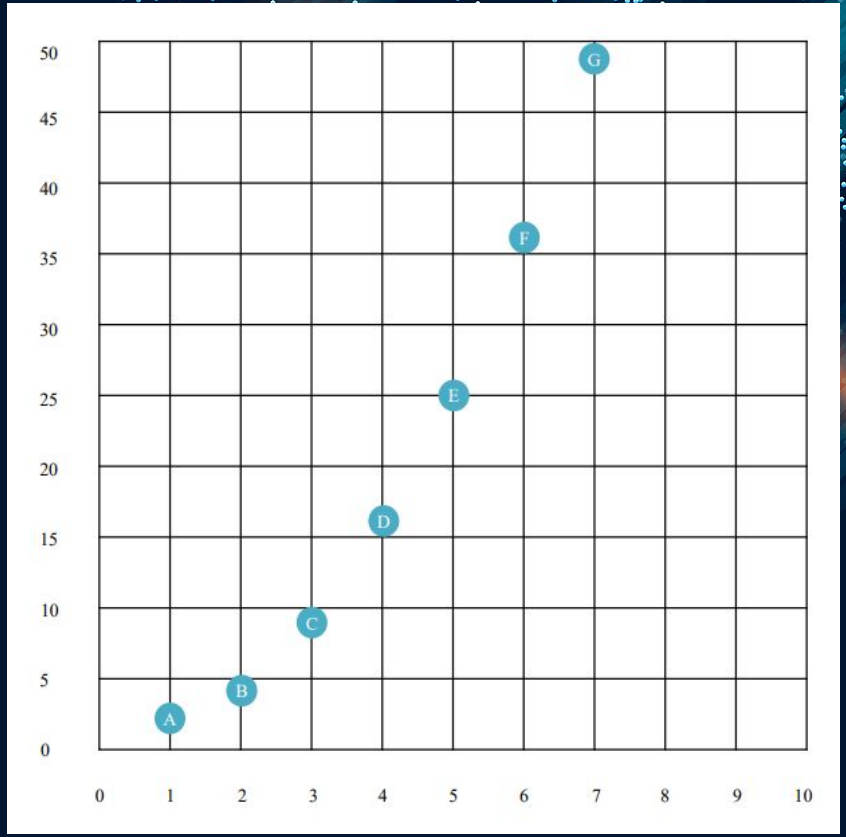
> Dataset

ID	X	Y
A	1	1
B	2	4
C	3	9
D	4	16
E	5	25
F	6	36
G	7	49

$\varepsilon = 3.00$

$C = 1.00$

$$K(x_i, x_j) = \left[(x_i)^t x_j\right]^2$$



Solution:

$$max\ -\frac{1}{2}\times\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i-\beta_i)\times(a_j-\beta_j)\times\boldsymbol{K}(\boldsymbol{x_i},\boldsymbol{x_j})-\varepsilon\times\sum_{i=1}^{n}(a_i+\beta_i)+\sum_{i=1}^{n}y_i\times(a_i-\beta_i)$$

$$s.t$$

$$0\le a_i,\beta_i\le C\ \forall i, i=1\dots n$$

$$\sum_{i=1}^{n}(a_i-\beta_i)=0$$

$$a_1=0; a_2=0; a_3=0; a_4=0; a_5=0; a_6=0; a_7=0.077$$

$$\beta_1=0; \beta_2=0; \beta_3=1; \beta_4=0; \beta_5=0; \beta_6=0.077; \beta_7=0$$

Since a7 and b6 are the only non-zero so 7 and 6 act as support vector points

$$z_i = \sum_{i=1}^{n} (a_i - \beta_i) \times K(x_i, x)$$

Zi is weighted kernel expansion

$z1 = 0.875$ $z2 = 3.500$ $z3 = 7.875$ $z4 = 14.000$
$z5 = 21.875$ $z6 = 31.500$ $z7 = 42.875$

$$\bar{b} = \frac{\sum_{m=1}^{M} y_m - z_m}{M}; M = \text{Number of Support Vectors}$$

After Solving this we get

$$\bar{b} = 3.125$$

$y_i = z_i + \overline{b} - \varepsilon$;

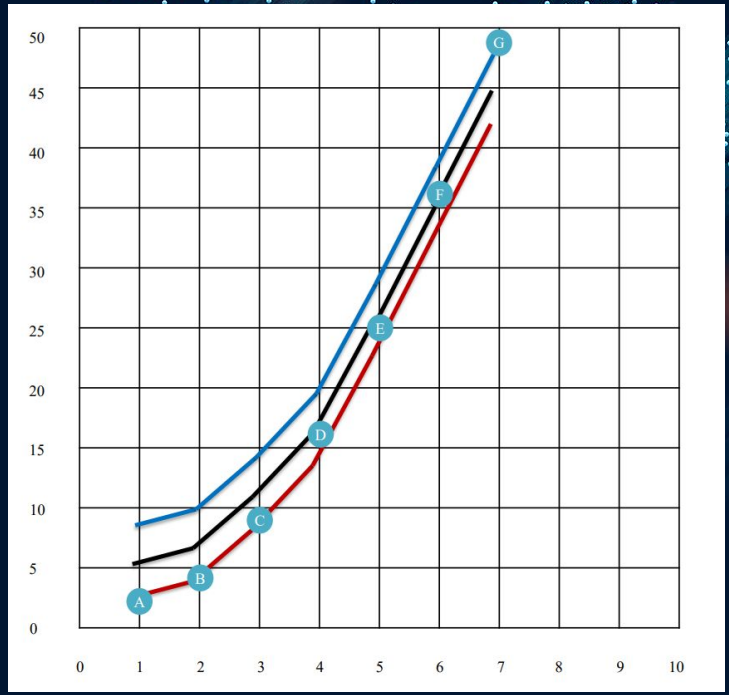
Y
1.00
3.63
8.00
14.13
22.00
31.62
43.00

$y_i = z_i + \overline{b}$;

Y
4.00
6.63
11.00
17.13
25.00
34.62
46.00

$y_i = z_i + \overline{b} + \varepsilon$

Y
7.00
9.63
14.00
20.13
28.00
37.62
49.00



Conclusion

- **Support Vector Regression (SVR)** is an adaptation of **Support Vector Machines (SVM)** for regression tasks.
- Instead of separating classes, SVR tries to fit a function within an **ε-insensitive tube**, ignoring small errors.
- Only **support vectors** (points outside the tube) influence the regression line/curve.
- SVR provides a balance between:
  - **Model flatness** (simple function with small weights).
  - **Prediction accuracy** (tolerance controlled by ε and penalty C).
- Works well with **both linear and non-linear data** (via kernel trick).

Key Differences: SVM vs SVR

Aspect	SVM (Classification)	SVR (Regression)
Goal	Find a hyperplane that separates classes with maximum margin.	Find a function (line/curve) that fits data within ε tolerance.
Output	Discrete class labels (e.g., Cat vs Dog).	Continuous values (e.g., House price).
Margin	Maximum-margin hyperplane between classes.	ε-insensitive tube around regression function.
Error Handling	Misclassified points incur penalty via slack variables.	Points outside ε-tube incur penalty via slack variables.
Support Vectors	Data points closest to separating hyperplane.	Data points outside or on the ε-tube.

GROUP 7

2201CS14
2201CS02
2201CS84
2201CS76
2201CS13
2201CS17
2201CS77
2201CS03
2201CS60
2201CS05
2201AI28
2201AI29

GROUP 8

2201AI52
2201AI41
2201AI42
2201AI09
2201AI39
2201AI53
2201AI33
2201AI40
2201AI50
2201AI01
2201CS93
2201CS89

Thank You