THADOMAL SHAHANI TSEC ENGINEERING COLLEGE
Om Shet
C1 07
2103163

Assignment 5

Q.1 Discuss the need of SVM? Explain why they are called as optimal binary classifier?

⇒

- Support Vector Machine (SVM) is a type of supervised learning that can be used for classification or regression.
- Even if the data points are unseen, SVM classify the data properly.
- The need of SVM :-

1) Effective in high dimensional spaces

⇒ SVMs perform well in case of where the number of features is large compared to the number of samples.
- This is particularly useful in areas like text classification, image recognition, and bioinformatics, where data often has many features.

2) Robust to overfitting

⇒ SVMs aims to find the optimal separating bounding between classes with a maximum margin, making them less prone to overfitting, especially in high-dimensional spaces.

3) Handles Outliers

⇒ SVMs are relatively robust to outliers due to use of a margin and support vectors, focusing only on the most important data points near the bounday.

4) Well-suited for binary classification

⇒ SVMs are particularly effective for binary classification problems, as they aim to maximise the margin between two classes, resulting in a clear and strong decision boundry.

- Why SVMs are called Optimal Binary classification

⇒

- SVMs are termed as optimal binary classifier because they focus on finding the most effective way to separate two classes by constructing an optimal hyperplane.
- This hyperplane is positioned in such a way that it maximizes the margin between the two classes - this margin being the distance between the hyperplane and the closest data points from each class, known as support vectors.
- By maximizing this margin, SVM ensures that the model is not only accurate but also generalizes well to new data, minimizing the risk of overfitting.
- Thus, SVM are known for their optimality in achieving both high accuracy and good generalization, especially in binary classification problem.

Q.2 Explain the following terminologies with the help of appropriate illustrations:

i) Optimal Decision Boundary

⇒

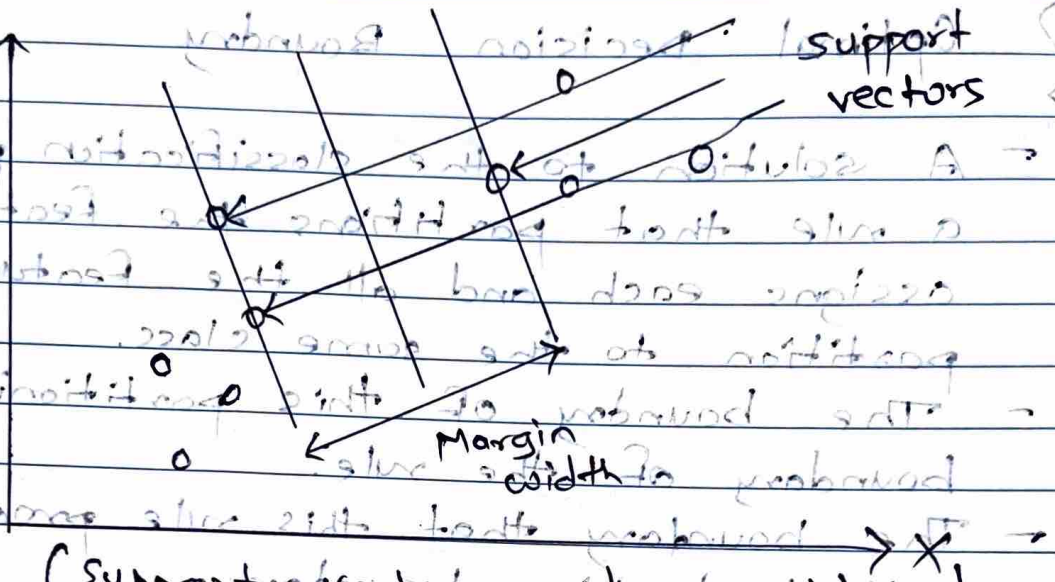
- A solution to the classification problem is a rule that partitions the features and assigns each and all the features and partition to the same class.
- The boundary of this partitioning is decision boundary of the rule.
- The boundary that this rule produces is the optimal decision boundary.
- A decision boundary is a region of a problem space in which the output label of a classifier is ambiguous.
- If the decision surface is a hyperplane, then classification is linear, and the classes are linearly separable.

ii) Support Vectors

⇒

- Support vectors are the datapoints that lie closest to the decision boundary or hyperplane
- These points are critical because they determine the position of the optimal hyperplane.
- The decision boundary is determined based only on these support vectors, while other points do not influence its position.

- Support vectors are essential in SVM because they help maximize the margin between two classes.



(Support vectors and Margin)

ii) Margins

⇒

- Margin refers to the distance between the decision boundary and the nearest data points of each class.
- SVM aims to maximize margin, which leads to a robust decision boundary.
- A larger margin implies a more confident and generalizable classifier, reducing the risk of overfitting.
- The margin is measured from the support vectors to the decision boundary, and maximizing it ensures the model's optimal performance.

Q.3 What do you understand by linear classifiers and non-linear classifiers? Can you use SVM as non-linear classifier? If yes, explain how SVM can be used as non-linear classifier.

⇒

- **Linear classifier**

⇒ A linear classifier is a classification algorithm that makes its predictions based on linear combination of input features.

- The decision boundary it creates to separate different classes is a straight line, a plane, or a hyperplane.

- **Non-linear classifier**

⇒ A non-linear classifier is one that can draw a more complex decision boundary to separate different classes, which isn't just a straight line or hyperplane.

- These classifiers are used when the data is not linearly separable.

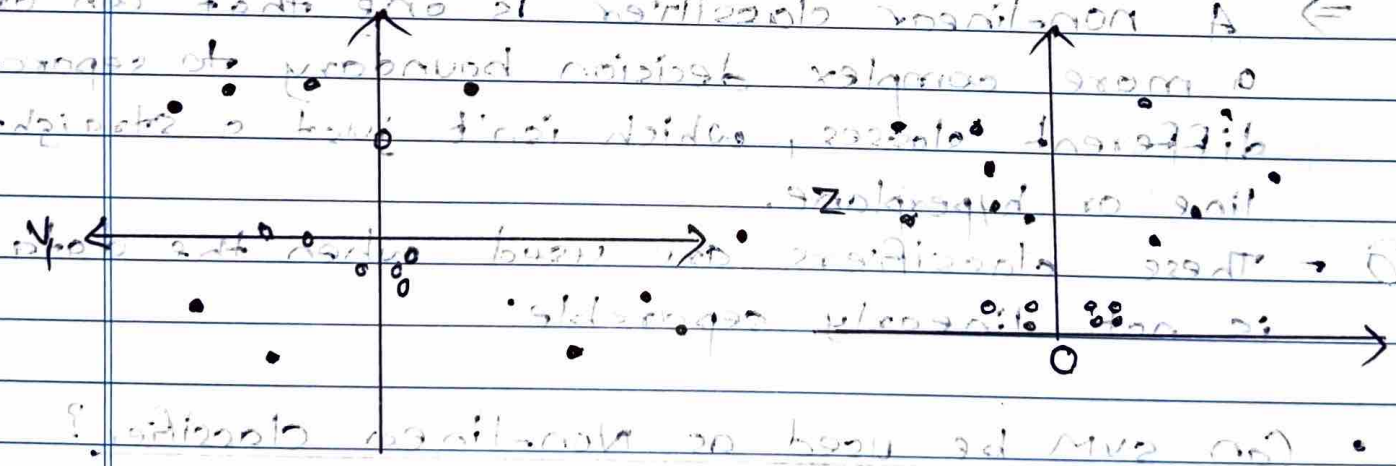
- **Can SVM be used as Non-linear classifier?**

⇒ Yes, SVM can be used as non-linear classifier

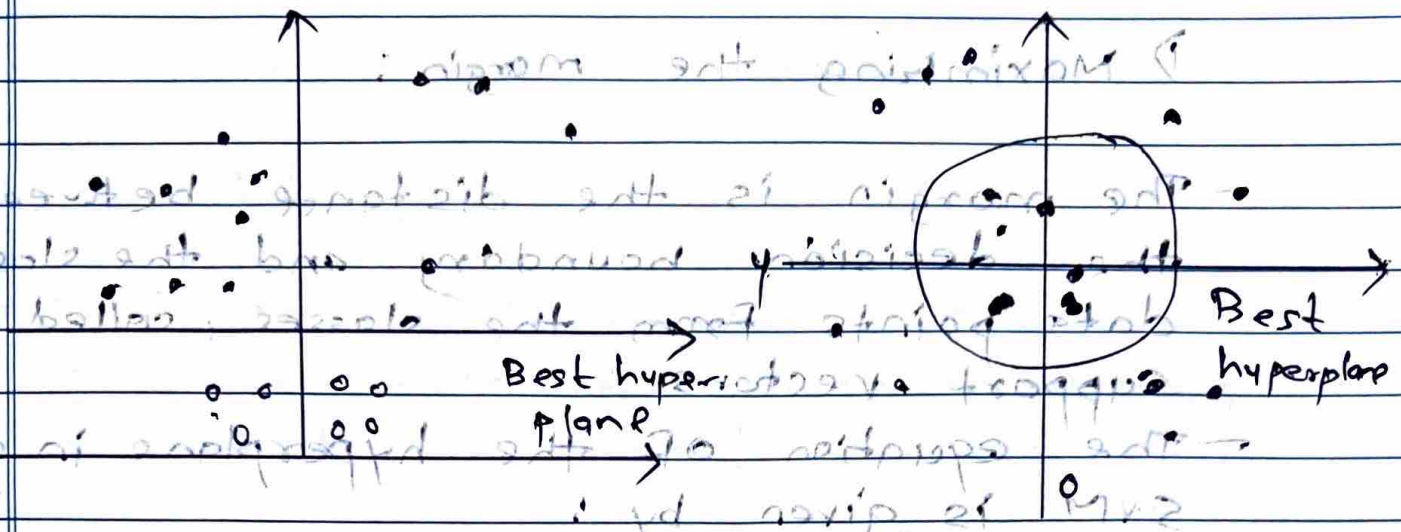
- Although the basic SVM algorithm is linear classifier, it can be extended to handle non-linear classification tasks through a method called the kernel trick.

Non-linear SVM

- If a data is linear, we can separate it by using a straight line, but if data is non-linear, then we cannot draw a single straight line.
- So to separate these data points, we need one more dimension.
- For linear data, we have used two dimensions x and y so for non-linear data, we add a third dimension z.
- It can be calculated as $z = x^2 + y^2$
- After adding the third dimension, the sample space becomes:



- Now, SVM will divide the datasets into classes in the following way:



- If we put $z=1$, then it will become as above,
  $x^2 + y^2 = 1$.
- We get a circumference of radius 1 as hyperplane surface in case of non-linear data.

Q.4 Express SVM as a constrained optimization problem. Discuss how predictions can be done by using SVM.

⇒
- SVMs aims to find the optimal separating hyperplane that maximizes the margin between two classes in dataset.
- To achieve this, SVMs are formulated as a constrained optimization problem.
- constrained optimization is a set of method to identify efficiently and systematically the best solution to a problem.
- The problem is characterized by a number of potential solution in the presence of identified constrained.

1) Maximizing the margin:

- The margin is the distance between the decision boundary and the closest data points from the classes, called support vectors.
- The equation of the hyperplane in an SVM is given by:

$$w \cdot x + b = 0$$

w = weight vector
b = bias (offset)
x = input vector.

- The decision boundary separates two classes such that:

$$w \cdot x_i + b \geq 1, \text{ for } y_i = +1$$
$$w \cdot x_i + b \leq -1, \text{ for } y_i = -1$$

2) Constrained Optimization problem:

- Thus, SVM optimization problem can be written as:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

- Subject to the constraints :

$$y_i (w \cdot x_i + b) \geq 1 \quad \forall i$$

- This is a convex optimization problem because both the objective function and constraints are convex.

3) Soft Margin SVM

⇒

- If the data is not perfectly separable, we introduce slack variables ξ_i to allow for some misclassifications.
- The optimization problem becomes :

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

- Subject to :

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i \quad \forall i$$

$$\xi_i \geq 0 \quad \forall i$$

Q.5 What are kernel functions? List some kernal functions. What is their use in SVM?

⇒

- Kernel functions is a method used to take data as input and transform it into the required form of processing data.
- "Kernel" is used due to set of mathematical functions used in SVM providing the window to manipulate the data.
- So, Kernel function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces.
- Basically, it returns the inner product between two points in a standard feature dimension.
- Some common kernel functions :-

1) Linear Kernel

⇒ This kernel is used when the data is linearly separable in the original space.

- It's formula is:

$$K(x_i, x_j) = x_i \times x_j$$

2) Polynomial Kernel

⇒ The polynomial kernel allows SVM to create non-linear decision boundaries by using polynomial functions of the input data.

→ $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$

3) Gaussian kernel

⇒ The guassian kernel maps the data into an infinite-dimensional space, allowing for highly complex decision boundaries.

→ $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

4) Sigmoid kernel

⇒ This kernel function resembles the activation function of a neural network.

→ $K(x_i, x_j) = \tanh(\alpha(x_i \cdot x_j) + r)$

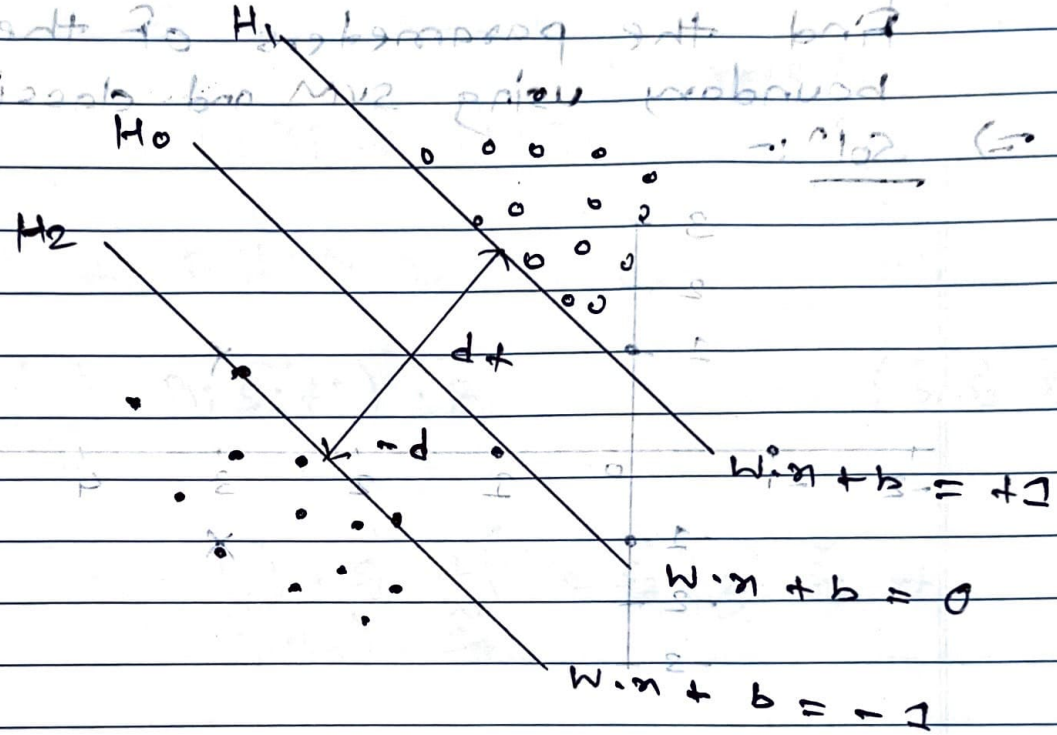
→ When the relationship between features is similar to the neural network structure or in case where sigmoid-like non-linearities are expected.

Q. 6 Explain the concept of kernel trick? Discuss with example, how kernel tricks can speed up the computations.

⇒ - A kernel trick is a simple method where a Non-linear data is projected onto a higher dimensional space so as to make it easier to classify the data. Then it is linearly divided by a plane.

- This is mathematically achieved by Lagragian formula using Lagrangian multipliers.
- Our basic idea of kernel trick is to find the plane which can separate, classify or split the data with maximum margin is also called street width.
- The distance from the point (x, y) to a line $Ax + By + C = 0$ is $\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$
- In order to maximize the margin, the distance between H_0 and H_1 is then $\frac{|w \cdot x + b|}{\|w\|} = \frac{1}{\|w\|}$, so the total distance between H_1 and H_2 is $\frac{2}{\|w\|}$
- In order to maximize the margin, we need to minimise $\|w\|$.
- Thus we are trying to optimize the margin or street width by maximising the distance by maximising distance

between the support vectors.



7. How Kernel Trick Speeds up Computation :-

1) Avoid Explicit transformation

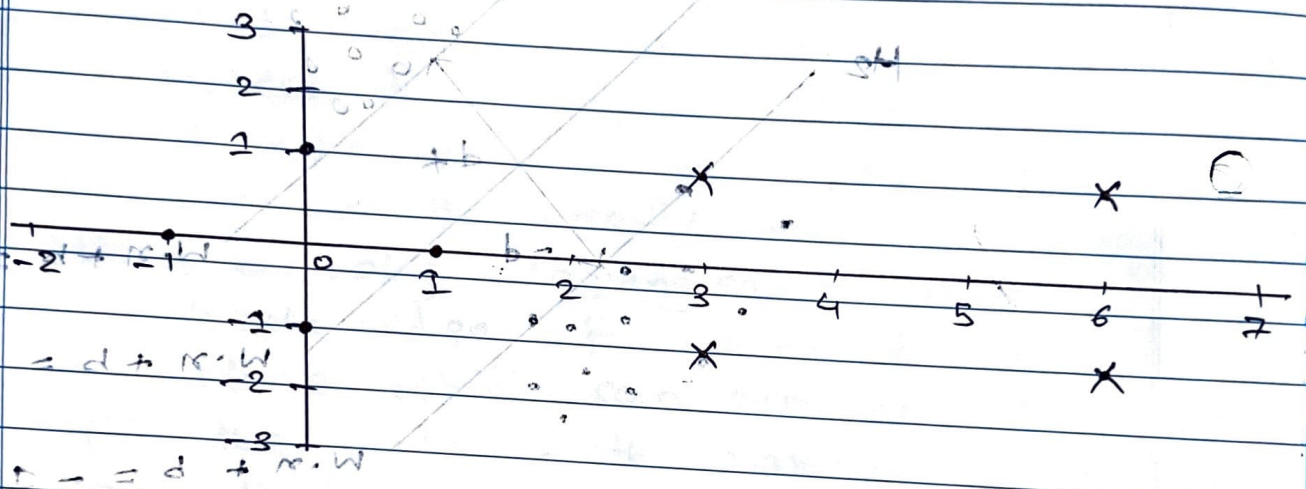
⇒ In cases where the transformation $\phi(x)$ maps the data into the high-dimensional or even infinite-dimensional space, explicitly calculating the transformed data is impractical.

2) Efficient Computation

⇒ The kernel trick allows SVM to compute the dot product between data points in the transformed space without the need to explicitly calculate the coordinates of the data in that space.

Q.7 Give '+'ly labelled data points as {(3,1), (3,-1), (6,1), (6,-1)} and '-'ly labelled data points as {(1,0), (0,1), (0,-1), (-1,0)} Find the parameters of the decision boundary using SVM and classify point (1,3).

⇒ Sol^n :-



- The circles are negatively labelled data points and cross are positively labelled data points.
- There are three support vectors -

$S_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $S_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$, $S_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$

- After adding bias input to S_1, S_2, S_3 we get,

$\tilde{S}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$, $\tilde{S}_2 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}$, $\tilde{S}_3 = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$

- Now,

$$D(z) = w^T z + b$$

$$= (\Sigma \lambda_i \tilde{s}_i t_i) \cdot z + b$$

$$= \Sigma \quad \text{(ignore b)}$$

$$\therefore (D(z)) = (\Sigma \lambda_i \tilde{s}_i t_i) \cdot z \quad \text{— (1)}$$

- We need to find $\lambda_1, \lambda_2, \lambda_3$ using eq (1).

$$y_i = \left(\sum_{i=1}^{3} \lambda_i \tilde{s}_i t_i\right) \cdot z \quad , \quad z \in (\tilde{s}_1, \tilde{s}_2, \tilde{s}_3)$$

$$\therefore y_1 = (\lambda_1 \tilde{s}_1 t_1) \cdot \tilde{s}_1 + (\lambda_2 \tilde{s}_2 t_2) \cdot \tilde{s}_2 + (\lambda_3 \tilde{s}_3 t_3) \cdot \tilde{s}_3$$

- Now, $y_1 = 1$

$$\lambda_1 (\tilde{s}_1 \cdot \tilde{s}_1) t_1 + \lambda_2 (\tilde{s}_2 \cdot \tilde{s}_1) t_2 + \lambda_3 (\tilde{s}_3 \cdot \tilde{s}_1) t_3 = -1$$
$$\lambda_1 (\tilde{s}_1 \cdot \tilde{s}_2) t_1 + \lambda_2 (\tilde{s}_2 \cdot \tilde{s}_2) t_2 + \lambda_3 (\tilde{s}_3 \cdot \tilde{s}_2) t_3 = 1$$
$$\lambda_1 (\tilde{s}_1 \cdot \tilde{s}_3) t_1 + \lambda_2 (\tilde{s}_2 \cdot \tilde{s}_3) t_2 + \lambda_3 (\tilde{s}_3 \cdot \tilde{s}_3) t_3 = 1$$

- Putting the values in above equations,

$$\lambda_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} t_1 + \lambda_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} t_2 + \lambda_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} t_3$$

$$= -1$$

$$2\lambda_1 + 4\lambda_2 + 4\lambda_3 = -1 \quad \text{— (2)}$$

- Similarly solving for rest two equations, we get -

$$4\lambda_1 + 11\lambda_2 + 9\lambda_3 = 1 \quad -(3)$$
$$4\lambda_1 + 9\lambda_2 + 11\lambda_3 = 1 \quad -(4)$$

Now, solving eq (2), (3), (4) we get,

$$\lambda_1 = -3.5$$
$$\lambda_2 = 0.75$$
$$\lambda_3 = 0.75$$

$$\therefore \tilde{w} = \sum_{i=1}^{3} \lambda_i \cdot s_i \cdot t_i$$

$$\therefore \tilde{w} = -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix} \begin{matrix} \rightarrow w \\ \rightarrow b \end{matrix}$$

- Since the support vectors were augmented with bias, ~~entries~~, entries in $\tilde{w}$ ~~comp~~ correspond to hyperplane with offset b,

where in, $y = w^T x + b$,

$W = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $b = -2$.

- The point (1, 3) is less than 2, since the decision boundary is $x_1 = 2$, this point lies on the negative side of the decision boundary.

11/10/22