FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत

NPTEL

MHRD

Introduction to Machine Learning

- Prof. Balaraman Ravindran | IIT Madras

Problem Solving Session (Week-4)

Shreya Bansal

PMRF PhD Scholar
IIT Ropar

Week-4 Contents

1. Perceptron
2. Support Vector Machine

Introduction to Linear Classifiers

- Definition: A linear classifier is a model that makes predictions based on a linear function of input features.
- Two main approaches:

    Discriminant Function Approach – Defines a decision boundary based on comparisons of discriminant functions.

    Direct Modeling of Separating Hyperplane – Explicitly models the decision boundary as a hyperplane.

- Example: Binary classification of emails as spam or non-spam using word frequencies.

Separating Hyperplane (Precap)

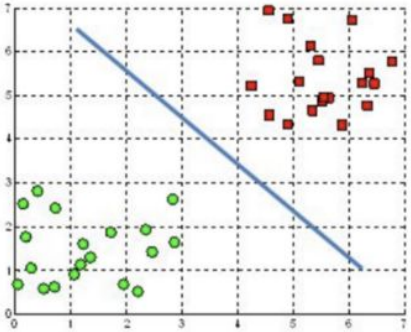
- **A hyperplane is a decision boundary that separates classes.**
- **Defined as: $f(x)= \beta^T X = 0$**
- **If $f(x)>0$**

  classify as Class 1;

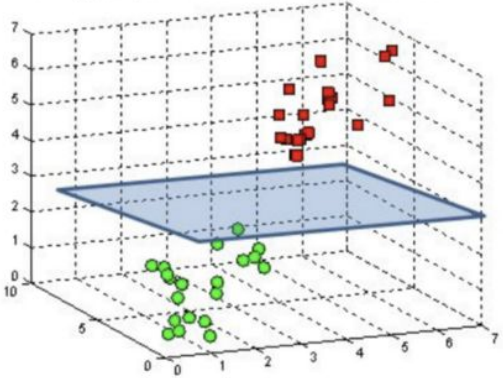
  elif $f(x)<0$

  classify as Class 2.

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane

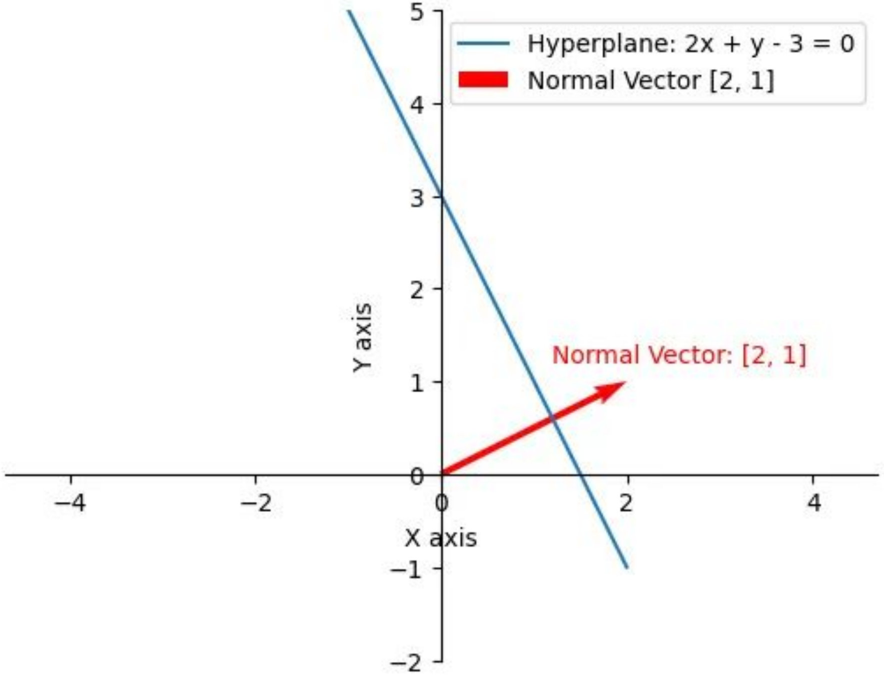


Properties of the Hyperplane

- If x_1, x_2 belong to the hyperplane L, then:

$$\beta^T(x_1-x_2)=0$$

- This means β is perpendicular to L .
- $\beta^T x_0 = -\beta_0$

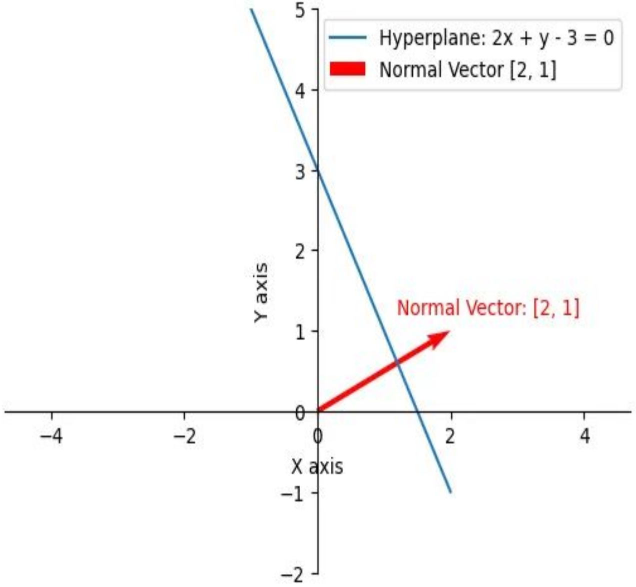


Distance from a Point to a Hyperplane

- The distance of a point from the hyperplane L is:

$$\beta^{*T}(x–x_0) = f(x)/||\beta|| = f(x)/f'(x)$$

- If(x) is proportional to the signed distance from x to the hyperplane.
- Normalizing β to unit norm makes f(x) equal the exact signed distance.



Example

- Example: Consider a hyperplane defined by 2x + 3y –3=0 and a point P(4,2)
- The distance of from the hyperplane is computed as follows:

Compute : f(P)= 2*4 + 3*2 –3 =11

Compute : $||\beta||=\sqrt{2^2 + 3^2} = \sqrt{13} = 3.60$

- Compute distance: 11/3.60=3.05
- Thus, the perpendicular distance of the point from the hyperplane is approximately 3.05.

Perceptron Decision Rule

- **Decision boundary is given by:**

  $x_i^T\beta + \beta_0 > 0$ —> Class +1 , otherwise class -1

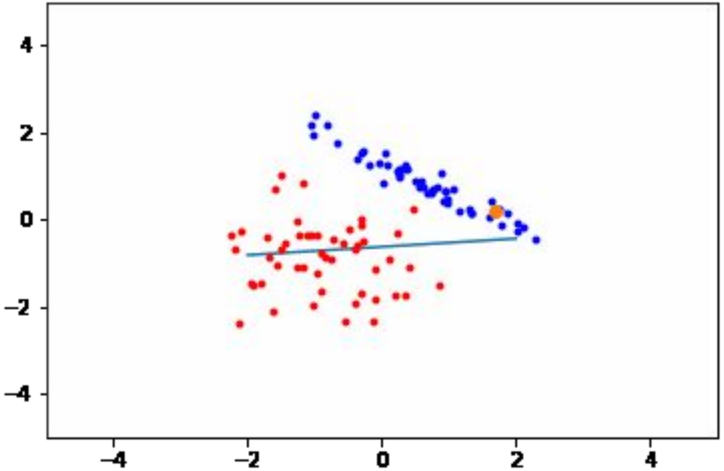
- **If a point is misclassified:**

  $f(x) < 0$ when the true class is +1

  $f(x) > 0$ when the true class is -1

Perceptron Objective Function

- Minimise distance of misclassified points to L
- Defined as: $D = -\sum_{xi \in M} y_i (x_i^T \beta + \beta_0)$
- Where M is the set of misclassified points.
- Goal: Minimize D, ideally making M empty.
- Works well when data is linearly separable.



Issues with Non-Linearly Separable Data

- If data is not linearly separable, the perceptron algorithm struggles.
- It continues updating without convergence.
- Leads to the introduction of Gradient Descent.

Gradient Descent for Perceptron

- **Compute the gradient:**

$$\partial D/\partial \beta = -\sum_{xi \in M} y_i x_i \qquad \partial D/\partial \beta_0 = -\sum_{xi \in M} y_i$$

- **Use Stochastic Gradient Descent (SGD):**
- **Take small steps in the direction of the gradient.**
- **Adjust the hyperplane iteratively.**
- **Avoid large jumps since changes dynamically.**

Sample Question: Perceptron

x1	x2	y
2	1	+1
1	–1	–1
–1	–2	–1
2	–2	+1

Sample Question: Perceptron

Step-1: Initialization

- Initial weight vector: $\beta=(0,0)$
- Initial bias: $\beta_0=0$
- Learning rate: $\eta=1$

Sample Question: Perceptron

Step-2: iteration-1 : $\beta=(0,0),\beta_0=0, \eta=1$

x1	x2	y	$(x_i^T\beta+\beta_0)$	$\beta^{new} = \beta^{old}+\eta(y_i x_i)$	$\beta_0^{new} = \beta_0^{old}+\eta(y_i)$
2	1	+1	0 (incorrect)	(0,0)+(1)(1)(2,1)=(2,1)	0+(1)(1)=(1)
1	-1	-1	2 (incorrect)	(2,1)+(1)(-1)(1,-1)=(1,2)	1+(1)(-1)=(0)
-1	-2	-1	-5(correct)	No update	No update
2	-2	+1	-2(incorrect)	(1,2)+(1)(1)(2,-2)=(3,0)	0+(1)(1)=(1)

Sample Question: Perceptron

Step-3: iteration-2: $\beta=(3,0), \beta_0=1, \eta=1$

x1	x2	y	$(x_i^T\beta+\beta_0)$	$\beta^{new} = \beta^{old}+\eta(y_i x_i)$	$\beta_0^{new} = \beta_0^{old}+\eta(y_i)$
2	1	+1	7 (correct)	No update	No update
1	-1	-1	4 (incorrect)	(3,0)+(1)(-1)(1,-1)=(2,1)	1+(1)(-1)=(0)
-1	-2	-1	-3(correct)	No update	No update
2	-2	+1	4(correct)	No update	No update

Sample Question: Perceptron

Step-4: iteration-3: $\beta=(2,1),\beta_0=0, \eta=1$

x1	x2	y	$(x_i^T\beta+\beta_0)$	$\beta^{new} = \beta^{old}+\eta(y_i x_i)$	$\beta_0^{new} = \beta_0^{old}+\eta(y_i)$
2	1	+1	5 (correct)	No update	No update
1	-1	-1	1 (incorrect)	(2,1)+(1)(-1)(1,-1)=(1,2)	0+(1)(-1)=(-1)
-1	-2	-1	-6(correct)	No update	No update
2	-2	+1	-3(incorrect)	(1,2)+(1)(1)(2,-2)=(3,0)	-1+(1)(1)=(0)

Sample Question: Perceptron

Step-5: iteration-4: $\beta=(3,0),\beta_0=0, \eta=1$

x1	x2	y	$(x_i^T\beta+\beta_0)$	$\beta^{new} = \beta^{old}+\eta(y_ix_i)$	$\beta_0^{new} = \beta_0^{old}+\eta(y_i)$
2	1	+1	6(correct)	No update	No update
1	-1	-1	3 (incorrect)	(3,0)+(1)(-1)(1,-1)=(2,1)	0+(1)(-1)=(-1)
-1	-2	-1	-5(correct)	No update	No update
2	-2	+1	1(correct)	No update	No update

Sample Question: Perceptron

Step-6: iteration-5: $\beta=(2,1), \beta_0=-1, \eta=1$

x1	x2	y	$(x_i^T\beta+\beta_0)$	$\beta^{new} = \beta^{old}+\eta(y_i x_i)$	$\beta_0^{new} = \beta_0^{old}+\eta(y_i)$
2	1	+1	4(correct)	No update	No update
1	-1	-1	0 (correct)	No update	No update
-1	-2	-1	-5(correct)	No update	No update
2	-2	+1	1(correct)	No update	No update

Optimal Separating Hyperplane

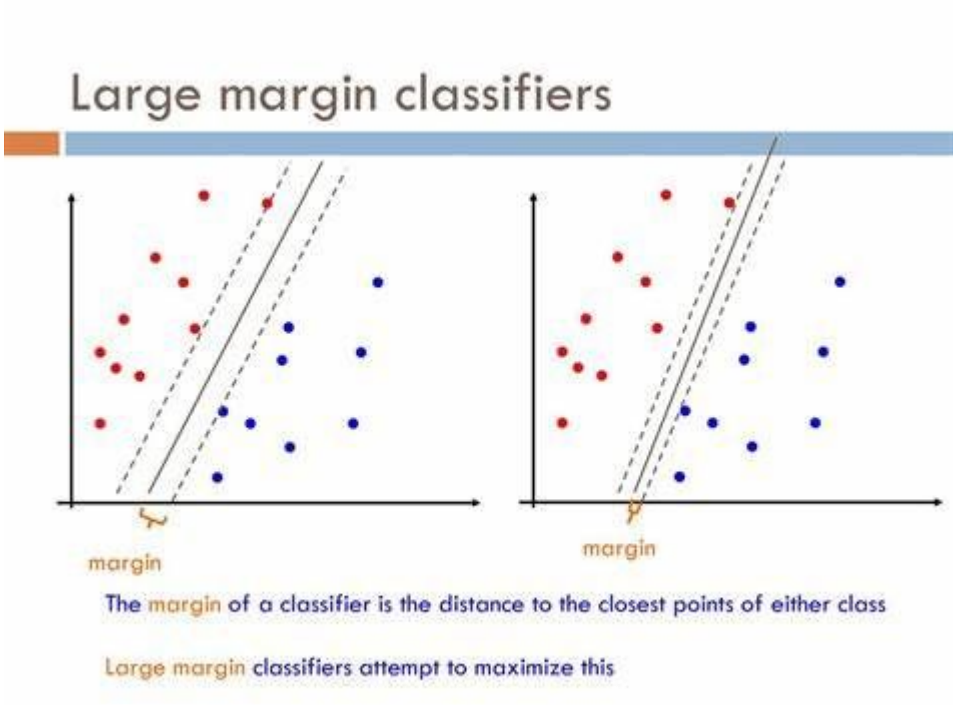
- Consider linearly separable data.
- Define optimality:

  Maximize the distance of the closest point to the hyperplane.

  Equal distance from the hyperplane for closest points on both sides

Concept of Margin

- **Margin: distance of the closest point to the hyperplane.**
- **Goal: Maximize the margin.**
- **Optimal classifiers are called Max-Margin Classifiers.**



Large margin classifiers

The margin of a classifier is the distance to the closest points of either class

Large margin classifiers attempt to maximize this

Mathematical Formulation

- Distance of a point from the hyperplane: $x_i^T\beta+\beta_0$.
- Constraint: Every point should be at least M away from the hyperplane.
- Optimization problem:

$$\max M, \text{ subject to } y_i(x_i^T\beta+\beta_0)\geq M$$

Normalization Constraint

- Introduce constraint $||\beta||$=1
- Avoid arbitrarily large β.
- Normalize by $||\beta||$ to remove constraint.

Reformulating the Optimization Problem

- Normalize β so that $||\beta||=1/M$.
- Equivalent optimization problem:

$$\text{Min } (1/2)\ ||\beta||^2, \quad \text{subject to } y_i(x_i^T\beta+\beta_0)\geq 1$$

- Ensures all data points are correctly classified and maximizes margin.

Geometric Interpretation

- Margin is $1/||\beta||$.
- Optimization finds:

  Minimum $||\beta||$.

  Correctly classified points at least $1/||\beta||$ away.
- Ensures maximum margin hyperplane.

Introduction to SVM

- SVM is a supervised learning algorithm used for classification and regression.
- It finds the optimal hyperplane that maximizes the margin between different classes.
- Works well for high-dimensional spaces and is effective when the number of dimensions is greater than the number of samples.

The Optimization Problem

- Given a labeled dataset $_{i=1}{}^{n}\{(xi,yi)\}$, where $x_i \in R^d$ and $y_i \in$ {-1,1}, the SVM optimization problem is formulated as:

$$\text{Min}_{w,b}\ (½)||w||^2 \text{ subject to } y_i(w^T x_i + b) \geq 1, \forall\, i.$$

- This formulation ensures that all data points are correctly classified with maximum margin.

Understanding Lagrange Multipliers

Minimize f(x,y)subject to g(x,y)=0

The constraint g(x,y)=0 means that the valid solutions lie on a curve. At the optimal point (highest or lowest value of f(x,y) on the curve), the level curves of f(x,y) must be parallel to the constraint curve g(x,y).

Mathematically, this means:

$\nabla f(x,y)=\lambda\nabla g(x,y)$

$\nabla f(x,y)$ is the gradient (direction of steepest increase) of f

$\nabla g(x,y)$ is the gradient of g (which is perpendicular to the constraint).

λ is a Lagrange multiplier, which scales the gradient of g to match the gradient of f.

The Lagrangian Function

To solve the constrained optimization problem, we define a new function called the Lagrangian function:

$$L(x,y,\lambda)=f(x,y)-\lambda g(x,y)$$

We then solve:

$\partial L/\partial x = 0$, $\partial L/\partial y = 0$, $\partial L/\partial \lambda = 0$

This gives us three equations to solve for x, y, and λ.

Example: Finding the Maximum on a Circle

Problem: Find the maximum of $f(x,y)=x+y$ subject to the constraint $x^2+y^2=1$ (a circle).

Step 1: Define the Lagrangian $L(x,y,\lambda)=(x+y)-\lambda(x^2+y^2-1)$

Step 2: Take Partial Derivatives

$\partial L/\partial x=1-\lambda(2x)=0\Rightarrow\lambda=1/2x$ $\partial L/\partial y=1-\lambda(2y)=0\Rightarrow\lambda=1/2y$
$\partial L/\partial\lambda=x^2+y^2-1=0$(constraint equation)

Step 3: Solve for x and$y$

Since $1/2x=1/2y$, it follows that $x=y$

Plugging into $x^2+y^2=1$, we get: $2x^2=1\Rightarrow x=\pm1/\sqrt{2}$, $y=\pm1/\sqrt{2}$

So the maximum occurs at $(1/\sqrt{2})$,$(1/\sqrt{2})$ and the minimum occurs at $(-1/\sqrt{2})$,$(-1/\sqrt{2})$

Lagrangian and Dual Formulation

- To solve the constrained optimization problem, we introduce Lagrange multipliers $\alpha_i \geq 0$ for each constraint:

$$L(w,b,\alpha)=(½)||w||^2-\sum_{i=1}^{n}\alpha_i[y_i(w^Tx_i+b)-1]$$

- Taking derivatives and setting them to zero:

$$\partial L/\partial w=0 \Rightarrow w=\sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\partial L/\partial \mathrm{b}=0 \Rightarrow \mathrm{b}=\sum_{i=1}^{n}\alpha_i y_i$$

- Substituting into the Lagrangian leads to the dual problem:

$$\mathrm{Max}_{\alpha}\ \sum_{i=1}^{n}\alpha_i-(½)\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_{iT}x_j \qquad \text{subject to:} \sum_{i=1}^{n}\alpha_i y_i=0,\ \ \alpha_i\geq 0\ \forall i.$$

Support Vectors and KKT condition

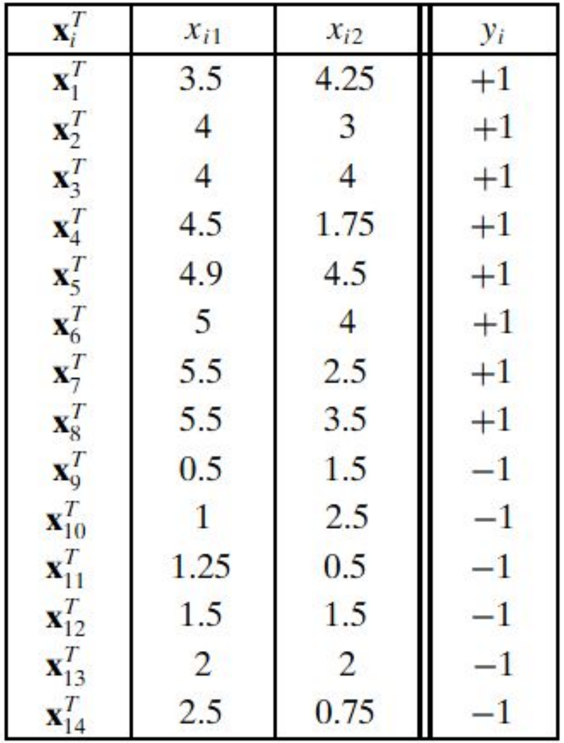
- $\alpha_i y_i(w^T x_i+b) -1=0$
- two cases:

  (1) $\alpha_i=0$, or(2) $y_i(w^T x_i+b) -1 = 0$,

  which implies $y_i(w^T x_i+b) =1$

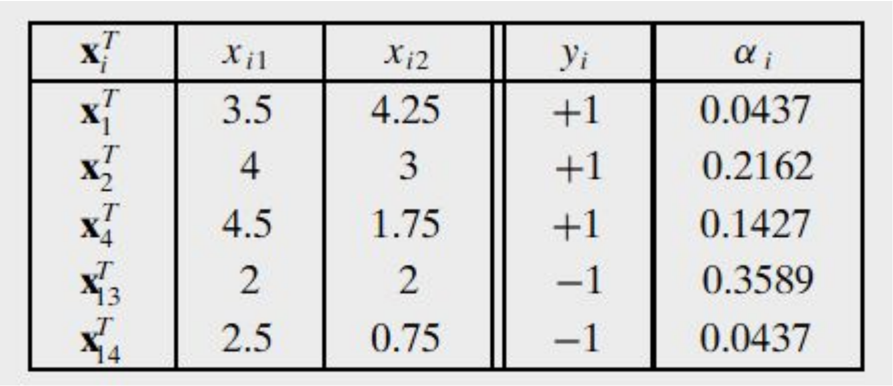
- Points for which $\alpha_i>0$ are called support vectors.
- These define the decision boundary and margin.

Example : SVM (Source: Zaki)



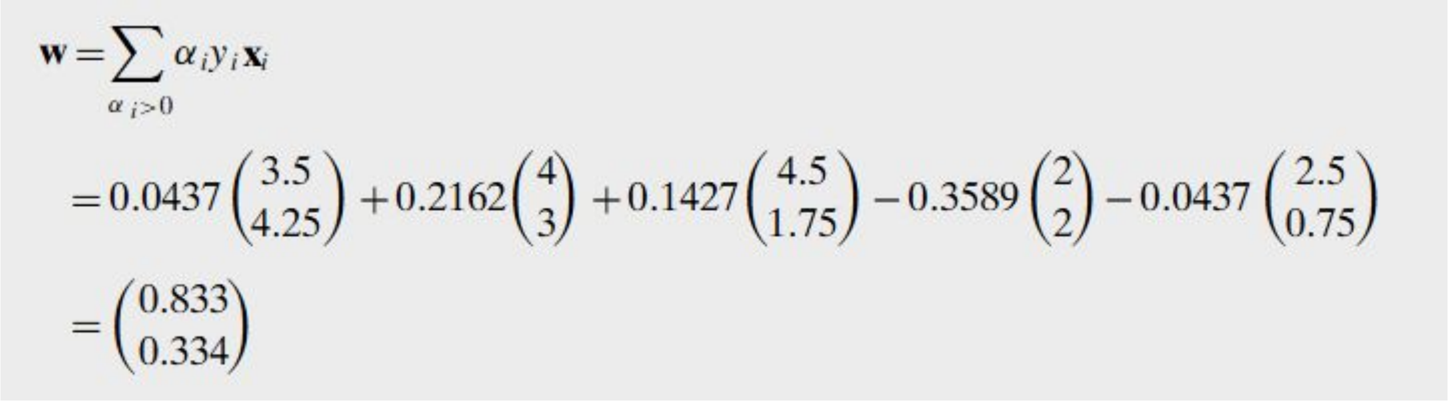
$\mathbf{x}_i^T$	x_{i1}	x_{i2}	y_i
$\mathbf{x}_1^T$	3.5	4.25	+1
$\mathbf{x}_2^T$	4	3	+1
$\mathbf{x}_3^T$	4	4	+1
$\mathbf{x}_4^T$	4.5	1.75	+1
$\mathbf{x}_5^T$	4.9	4.5	+1
$\mathbf{x}_6^T$	5	4	+1
$\mathbf{x}_7^T$	5.5	2.5	+1
$\mathbf{x}_8^T$	5.5	3.5	+1
$\mathbf{x}_9^T$	0.5	1.5	−1
$\mathbf{x}_{10}^T$	1	2.5	−1
$\mathbf{x}_{11}^T$	1.25	0.5	−1
$\mathbf{x}_{12}^T$	1.5	1.5	−1
$\mathbf{x}_{13}^T$	2	2	−1
$\mathbf{x}_{14}^T$	2.5	0.75	−1

Step-1: Solve Dual problem with quadratic programming to get α_i



$\mathbf{x}_i^T$	x_{i1}	x_{i2}	y_i	α_i
$\mathbf{x}_1^T$	3.5	4.25	+1	0.0437
$\mathbf{x}_2^T$	4	3	+1	0.2162
$\mathbf{x}_4^T$	4.5	1.75	+1	0.1427
$\mathbf{x}_{13}^T$	2	2	−1	0.3589
$\mathbf{x}_{14}^T$	2.5	0.75	−1	0.0437

Example : SVM (Source: Zaki)



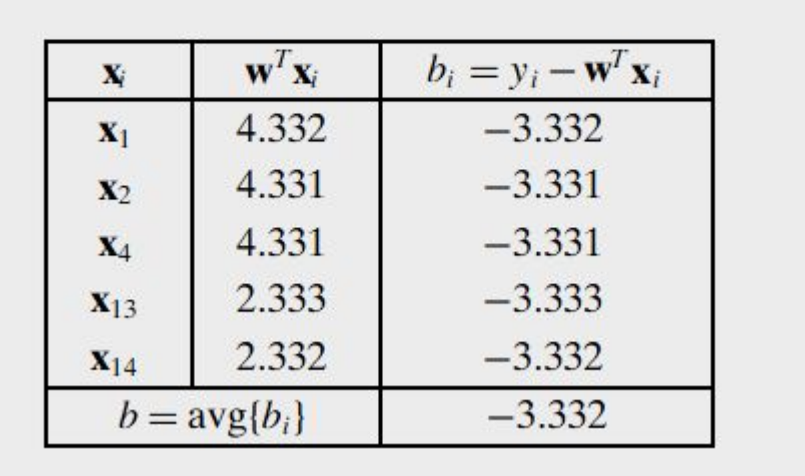
$$\mathbf{w} = \sum_{\alpha_i > 0} \alpha_i y_i \mathbf{x}_i$$

$$= 0.0437\begin{pmatrix}3.5\\4.25\end{pmatrix} + 0.2162\begin{pmatrix}4\\3\end{pmatrix} + 0.1427\begin{pmatrix}4.5\\1.75\end{pmatrix} - 0.3589\begin{pmatrix}2\\2\end{pmatrix} - 0.0437\begin{pmatrix}2.5\\0.75\end{pmatrix}$$

$$= \begin{pmatrix}0.833\\0.334\end{pmatrix}$$

Step-2: Calculate $w={}_{i=1}\Sigma^{n}\alpha_i y_i x_i$

Pitfalls of Linear Regression for Classification



$\mathbf{x}_i$	$\mathbf{w}^T\mathbf{x}_i$	$b_i = y_i - \mathbf{w}^T\mathbf{x}_i$
$\mathbf{x}_1$	4.332	−3.332
$\mathbf{x}_2$	4.331	−3.331
$\mathbf{x}_4$	4.331	−3.331
$\mathbf{x}_{13}$	2.333	−3.333
$\mathbf{x}_{14}$	2.332	−3.332
$b = \text{avg}\{b_i\}$		−3.332

Step-3: Calculate **b**

Problem with Perceptrons:

- Perceptrons do not converge if the data is linearly inseparable.
- The standard perceptron algorithm assumes the existence of a linear separator, which may not always be the case.

Solution: Maximizing the Margin

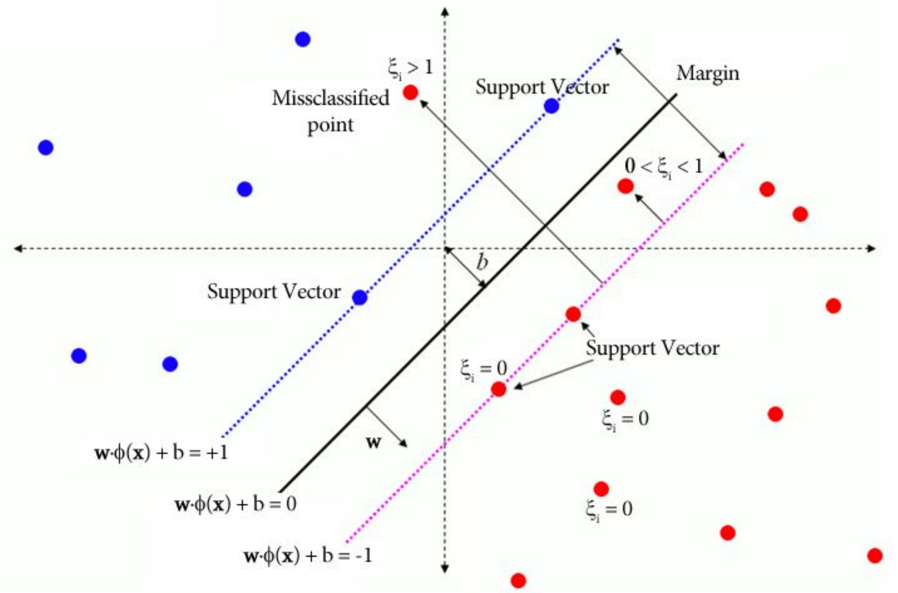
- Instead of just trying to classify data correctly, we maximize the margin (the distance between the decision boundary and the nearest data points).
- However, real-world data is often noisy and not perfectly separable, so a strict margin might not work well.

Handling Margin Violations

- Some points lie inside the margin → we want to minimize their deviation.
- Some points are incorrectly classified → we need a trade-off.
- Instead of minimizing the maximum violation, we minimize the sum of violations.

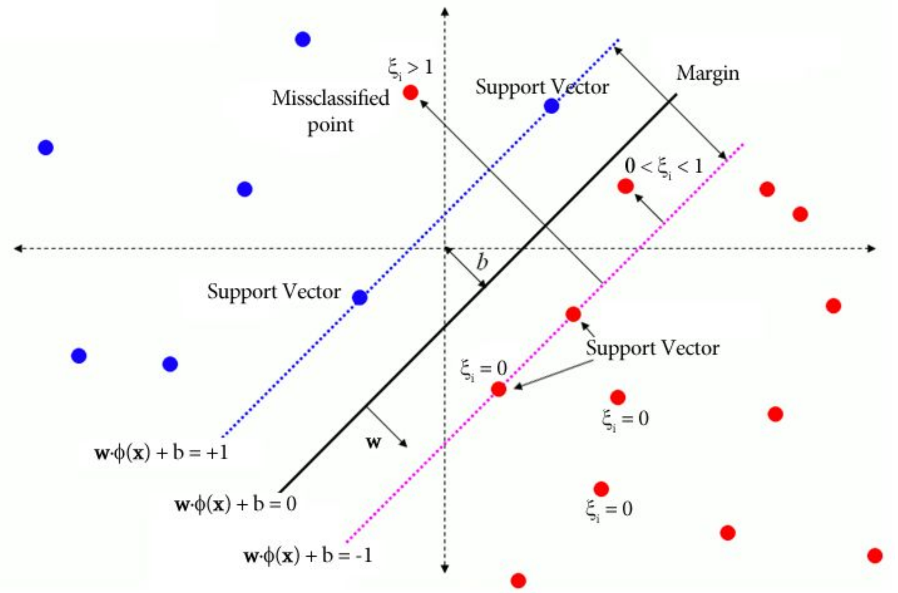
Slack Variables (ξi) for Soft Margin SVM

- Introduce slack variables *ξi* to allow some misclassification.
- These variables measure how much each data point violates the margin constraint.
- The objective function includes a term that minimizes the sum of these slack variables.


$\xi_i > 1$
Missclassified point
Support Vector
Margin
$0 < \xi_i < 1$
b
Support Vector
Support Vector
$\xi_i = 0$
$\xi_i = 0$
$\xi_i = 0$
w
$\mathbf{w} \cdot \phi(\mathbf{x}) + b = +1$
$\mathbf{w} \cdot \phi(\mathbf{x}) + b = 0$
$\mathbf{w} \cdot \phi(\mathbf{x}) + b = -1$

Introducing Slack Variables (ξi)

- Relaxing Constraints
- We introduce slack variables (ξi) to allow flexibility:
- Ideally, all $\xi_i = 0$ (perfect classification).
- But in real-world data, some ξ_i will be nonzero to allow margin violations.
- Constraints:
- $\xi_i \geq 0$ (can’t have negative slack).
- Keep total slack under a certain limit.



Optimization Formulation

- **Modified Constraint**
- **Instead of:**$y_i f(x_i) \geq M$
- **We allow slack:**$y_i f(x_i) \geq M(1 - \xi_i)$
- **This keeps the optimization problem convex.**
- ξ_i **represents how much a point violates the margin.**

Soft Margin SVM

- **For non-linear separable data, introduce slack variables $\xi_i \geq 0$**

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

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

- **The parameter *C* controls the trade-off between margin size and classification error.**

Regularization Trade-off (C Parameter)

- **Controlling the Trade-off**
- **C determines how much we penalize margin violations:**
- **Large C → Smaller margin, fewer training errors.**
- **Small C → Larger margin, allows some errors.**
- **If C → ∞, we recover the hard-margin SVM (strict separation).**

Impact of Noisy Data

- Choosing a Good Margin
- If data has noise, strict separation may lead to overfitting.
- Allowing some errors (small C) leads to a more robust classifier.
- Example:
- Hard-margin → Fits outliers too closely.
- Soft-margin → Finds a better overall separation.

Introduction to Inner Products and Kernels

- Inner products are key in evaluating dual formulations and classifiers.
- Efficient computation of inner products allows optimization in transformed spaces.
- Basis transformations enhance linear classifiers by expanding the feature space.

Basis Expansion and Feature Space Transformation

- Basis expansion replaces x with $h(x)$, a function mapping to a higher-dimensional space.
- Example: Quadratic basis expansion $h(x)$ includes squared terms.
- Dual formulations rely only on inner products in the transformed space.

Kernel Functions as Similarity Measures

- Kernels compute inner products between transformed vectors without explicit feature expansion.
- Kernel trick: Avoids direct computation in high-dimensional space, reducing complexity.

Kernel Trick

- In non-linearly separable cases, we use kernel functions to map input space into a higher-dimensional feature space.
- Mathematically:
- $K(x_i,\mathrm{x}_j)=\phi(x_i)\cdot\phi(\mathrm{x}_j)$
-
- where
- $\phi(x_i)$ is the feature mapping function.

Common kernels

- Linear: $K(x_i,x_j)=x_i^T x_j$
- Polynomial: $K(x_i,x_j)=(<x_i x_j >+c)^d$
- Radial Basis Function (RBF): $K(x_i,x_j)=\mathrm{exp}(-\gamma||x_i-x_j||^2)$
- Sigmoid Kernel: $K(x_i,x_j)=\mathrm{tanh}(\alpha x_{\mathrm{i}} \cdot \mathrm{x}_{\mathrm{j}}+c)$

Example - Polynomial kernel

- **$x=(x_1\ x_2)$ - 2D space**
- **$K(x,x')=(1+ <x,x'>)^2 = (1+x_1x_1'+ x_2x_2')^2 = (1 + x^2{}_1x^2{}_1' +x^2{}_2x^2{}_2'+ 2x_1x_1' + 2x_2x_2' +2x_1x_1'x_2x_2')$**

$h_1(x)=1$ $h_2(x)=\sqrt{2}\ x_1$ $h_3(x)=\sqrt{2}\ x_2$ $h_4(x)=x^2{}_1$ $h_5(x)=x^2{}_1$ $h_6(x)=\sqrt{2}\ x_1x_2$

- **$\phi(x)=[h_1(x),h_2(x),h_3(x),h_4(x),h_5(x),h_6(x)]$ 6 D space**

- For an **n-dimensional input** vector x=(x1,x2,...,xn), a degree ddd polynomial basis expansion The number of features in the expanded space is: New feature count=$C_d^{(n+d)}$

Advantages of the Kernel Trick

- Enables SVM to handle complex decision boundaries
- Avoids explicit transformation into high-dimensional space (saves computation)
- Works well with small-to-medium-sized datasets

Choosing the Right Kernel

- Linear Kernel → When data is already linearly separable
- Polynomial Kernel → When data has curved boundaries but is not too complex
- RBF Kernel → Best for most cases; handles complex non-linearity
- Sigmoid Kernel → Less commonly used, mimics neural networks

Introduction to SVM′s Primal Objective Function

Originally, SVM optimization was written in terms of α, but it can be reformulated using λ

The function $f(x)=x_i^T\beta+\beta_0$ determines classification.

The hinge loss is introduced via the plus function:

$$\max(0,1-yf(x))$$

This means that when $yf(x)$ is positive and above 1, the loss is 0. Otherwise, it contributes linearly.

Connection to Ridge Regression:

SVM's objective function looks like ridge regression:

Loss + $\lambda||\beta||^2$

The norm regularization term in ridge regression ensures a small-weight solution (prevents overfitting).

L1 Regularized SVM

- What if L1 norm ($||\beta||_1$) replaces L2 norm ($||\beta||_2^2$)?
- L1 induces sparsity (fewer support vectors).
- Harder optimization problem but valid.
- Fewer support vectors improve generalization.

Squared Loss vs. Hinge Loss

- Squared loss penalizes all points, even correctly classified ones.
- Hinge loss focuses only on misclassified or margin-violating points.
- SVMs perform better with hinge loss in classification problems.

0-1 Loss and Theoretical Considerations

- 0-1 Loss: Ideal function (0 if correct, 1 if incorrect).
- Why not optimize it? → Computationally infeasible.
- Instead, we use proxy loss functions like hinge loss or squared loss.
- Theoretical ML research focuses on finding equivalent solutions.

Logistic Regression vs. SVM Loss Functions

- Logistic regression optimizes log loss:

  $\log(1+e-yf(x))$

- Never goes to zero, unlike hinge loss.
- Logistic loss is derived from Maximum Likelihood Estimation (MLE).
- Used in probabilistic classification, while SVMs focus on margin maximization.

Assignment-4 (Cs-101- 2024) (Week-4)

Source

Question-1

01:00

In the context of the perceptron learning algorithm, what does the expression $f(x)/||f'(x)||$ represent?

a) The gradient of the hyperplane
b) The signed distance to the hyperplane
c) The normal vector to the hyperplane
d) The misclassification error

Question-1- Correct answer

In the context of the perceptron learning algorithm, what does the expression f(x)/||f′(x)|| represent?

a) The gradient of the hyperplane

b) The signed distance to the hyperplane

c) The normal vector to the hyperplane

d) The misclassification error

Correct options: (b)

Question-2

01:00

Why do we normalize by // β // (the magnitude of the weight vector) in the SVM objective function?

a) To ensure the margin is independent of the scale of β
b) To minimize the computational complexity of the algorithm
c) To prevent overfitting
d) To ensure the bias term is always positive

Question-2- Correct answer

Why do we normalize by // β // (the magnitude of the weight vector) in the SVM objective function?

a) To ensure the margin is independent of the scale of β

b) To minimize the computational complexity of the algorithm

c) To prevent overfitting

d) To ensure the bias term is always positive

Correct options: (a)

Question-3

01:00

Which of the following is NOT one of the KKT conditions for optimization problems with inequality constraints?

a) Stationarity: $\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{p} \nu_j \nabla h_j(x^*) = 0$

b) Primal feasibility: $g_i(x^*) \leq 0$ for all i , and $h_j(x^*) = 0$ for all j

c) Dual feasibility: $\lambda_i \geq 0$ for all i

d) Convexity: The objective function $f(x)$ must be convex

Question-3 - Correct answer

Which of the following is NOT one of the KKT conditions for optimization problems with inequality constraints?

a) Stationarity: $\nabla f(x*)+i=1\sum m\ \lambda i\nabla gi(x*)+j=1\sum p\nu j\nabla hj(x*)=0$

b) Primal feasibility: $gi(x*)\leq 0$ for all i , and $hj(x*)=0$ for all j

c) Dual feasibility: $\lambda i\geq 0$ for all i

d) Convexity: The objective function $f(x)$ must be convex

Correct options: (d)

Question-4

03:00

Consider the 1 dimensional dataset: (x is input and y is output)
State true or false: The dataset becomes linearly separable after using basis expansion with the following basis function $\phi(x)=[1\ x^3]^T$

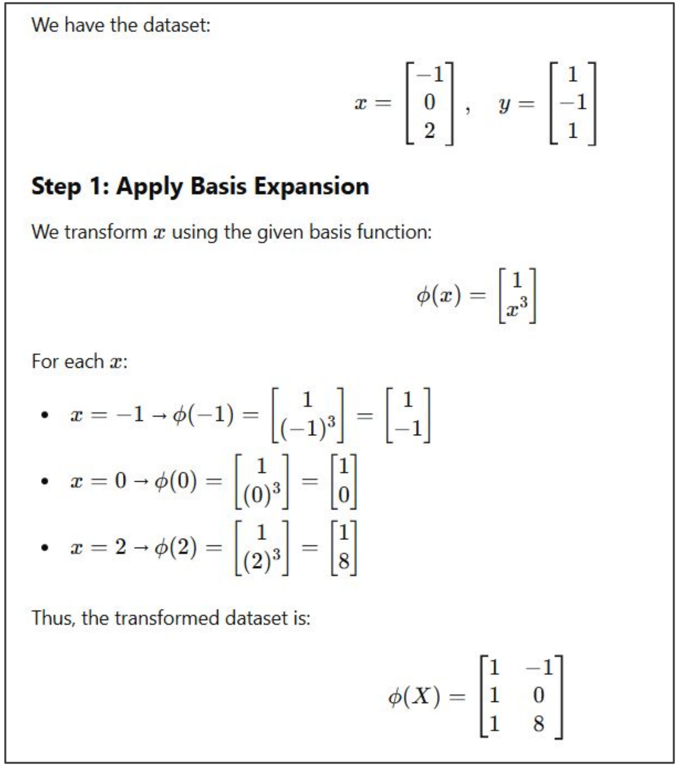
a) True

b) False

x	y
-1	1
0	-1
2	1

Question-4- Explanation

03:00



We have the dataset:

$$x = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Step 1: Apply Basis Expansion

We transform x using the given basis function:

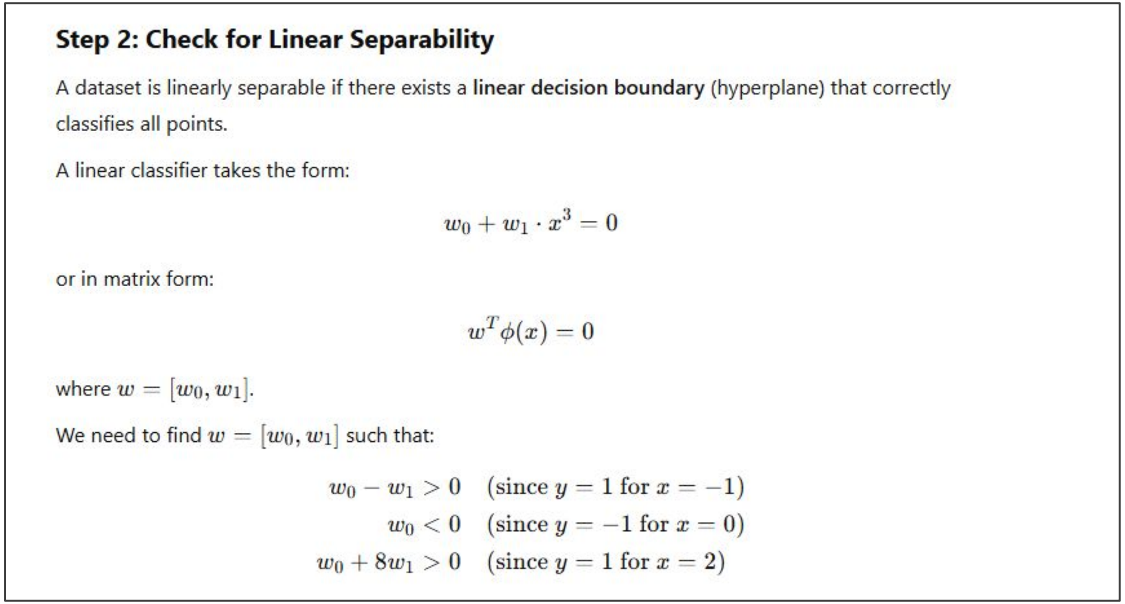
$$\phi(x) = \begin{bmatrix} 1 \\ x^3 \end{bmatrix}$$

For each x:

- $x = -1 \rightarrow \phi(-1) = \begin{bmatrix} 1 \\ (-1)^3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
- $x = 0 \rightarrow \phi(0) = \begin{bmatrix} 1 \\ (0)^3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- $x = 2 \rightarrow \phi(2) = \begin{bmatrix} 1 \\ (2)^3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \end{bmatrix}$

Thus, the transformed dataset is:

$$\phi(X) = \begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 8 \end{bmatrix}$$



Step 2: Check for Linear Separability

A dataset is linearly separable if there exists a **linear decision boundary** (hyperplane) that correctly classifies all points.

A linear classifier takes the form:

$$w_0 + w_1 \cdot x^3 = 0$$

or in matrix form:

$$w^T \phi(x) = 0$$

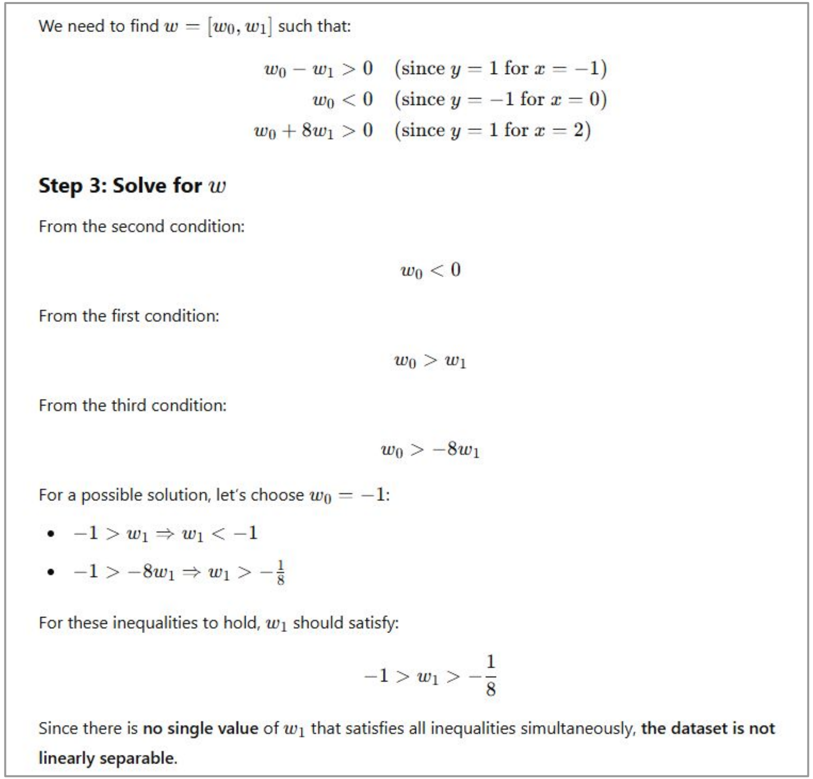
where $w = [w_0, w_1]$.

We need to find $w = [w_0, w_1]$ such that:

$$\begin{aligned} w_0 - w_1 > 0 &\quad (\text{since } y = 1 \text{ for } x = -1) \\ w_0 < 0 &\quad (\text{since } y = -1 \text{ for } x = 0) \\ w_0 + 8w_1 > 0 &\quad (\text{since } y = 1 \text{ for } x = 2) \end{aligned}$$

Question-4- Explanation

03:00



We need to find $w = [w_0, w_1]$ such that:

$$\begin{aligned} w_0 - w_1 > 0 &\quad (\text{since } y = 1 \text{ for } x = -1) \\ w_0 < 0 &\quad (\text{since } y = -1 \text{ for } x = 0) \\ w_0 + 8w_1 > 0 &\quad (\text{since } y = 1 \text{ for } x = 2) \end{aligned}$$

Step 3: Solve for w

From the second condition:

$$w_0 < 0$$

From the first condition:

$$w_0 > w_1$$

From the third condition:

$$w_0 > -8w_1$$

For a possible solution, let's choose $w_0 = -1$:

- $-1 > w_1 \Rightarrow w_1 < -1$
- $-1 > -8w_1 \Rightarrow w_1 > -\frac{1}{8}$

For these inequalities to hold, w_1 should satisfy:

$$-1 > w_1 > -\frac{1}{8}$$

Since there is **no single value** of w_1 that satisfies all inequalities simultaneously, **the dataset is not linearly separable**.

Question-4 - Correct answer

Consider the 1 dimensional dataset: (x is input and y is output)

State true or false: The dataset becomes linearly separable after using basis expansion with the following basis function $\phi(x)=[1\ x^3]^T$

a) True

b) False

Correct options: (b)

Question-5

01:00

Consider a polynomial kernel of degree d operating on p -dimensional input vectors. What is the dimension of the feature space induced by this kernel?

a) p x d

b) (p+1) x d

c) C_d^{p+d}

d) p^d

Question-5 - Correct answer

Consider a polynomial kernel of degree d operating on p -dimensional input vectors. What is the dimension of the feature space induced by this kernel?

a) p x d

b) (p+1) x d

c) C_d^{p+d}

d) p^d

Correct options: (c)

Question-6

01:00

State True or False: For any given linearly separable data, for any initialization, both SVM and Perceptron will converge to the same solution

a) True

b) False

Question-6 - Correct answer

State True or False: For any given linearly separable data, for any initialization, both SVM and Perceptron will converge to the same solution

a) True

b) False

Correct options: (b)

Question-7,8 data

01:00

Kindly download the modified version of Iris dataset from this link.

Available at: (https://goo.gl/vchhsd)

The dataset contains 150 points, and

each input point has 4 features and belongs to one among three classes.

Use the first 100 points as the training data and the remaining 50 as test data. In the following questions, to report accuracy, use the test dataset. You can round off the accuracy value to the nearest 2-decimal point number. (Note: Do not change the order of data points.)

Question-7

03:00

Train a Linear perceptron classifier on the modified iris dataset. We recommend using sklearn. Use only the first two features for your model and report the best classification accuracy for l1 and l2 penalty terms.

a) 0.91, 0.64
b) 0.88, 0.71
c) 0.71, 0.65
d) 0.78, 0.64

Question-7 - Correct answer

Train a Linear perceptron classifier on the modified iris dataset. We recommend using sklearn. Use only the first two features for your model and report the best classification accuracy for l1 and l2 penalty terms.

a) 0.91, 0.64
b) 0.88, 0.71
c) 0.71, 0.65
d) 0.78, 0.64

Correct options: (d)

Question-8

01:00

Train a SVM classifier on the modified iris dataset. We recommend using sklearn. Use only the first three features. We encourage you to explore the impact of varying different hyperparameters of the model. Specifically try different kernels and the associated hyperparameters. As part of the assignment train models with the following set of hyperparameters RBF-kernel, gamma=0.5 , one-vs-rest classifier, no-feature-normalization.

Try C=0.01,1,10 . For the above set of hyperparameters, report the best classification accuracy.

a) 0.98 b) 0.88 c)0.99 d) 0.92

Question-8- Correct answer

Train a SVM classifier on the modified iris dataset. We recommend using sklearn. Use only the first three features. We encourage you to explore the impact of varying different hyperparameters of the model. Specifically try different kernels and the associated hyperparameters. As part of the assignment train models with the following set of hyperparameters RBF-kernel, gamma=0.5 , one-vs-rest classifier, no-feature-normalization.

Try C=0.01,1,10 . For the above set of hyperparameters, report the best classification accuracy.

a) 0.98 b) 0.88 c)0.99 d) 0.92

Correct options: (a)

Assignment-4 (Cs-46- 2025) (Week-4)

Source

Question-1

01:00

The Perceptron Learning Algorithm can always converge to a solution if the dataset is linearly separable.

a) True
b) False
c) Depends on learning rate
d) Depends on initial weights

Question-1- Correct answer

The Perceptron Learning Algorithm can always converge to a solution if the dataset is linearly separable.

a) True

b) False

c) Depends on learning rate

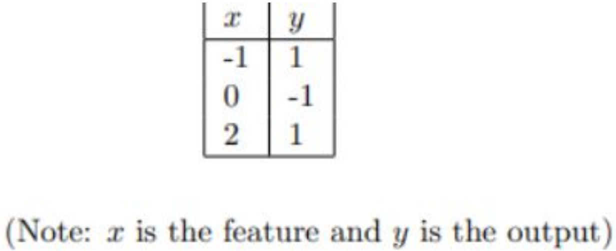
d) Depends on initial weights

Correct options: (a)

Question-2

01:00

Consider the 1 dimensional dataset:



x	y
-1	1
0	-1
2	1

(Note: x is the feature and y is the output)

State true or false: The dataset becomes linearly separable after using basis expansion with the following basis function $\phi(x)=[1\ x^2]^T$

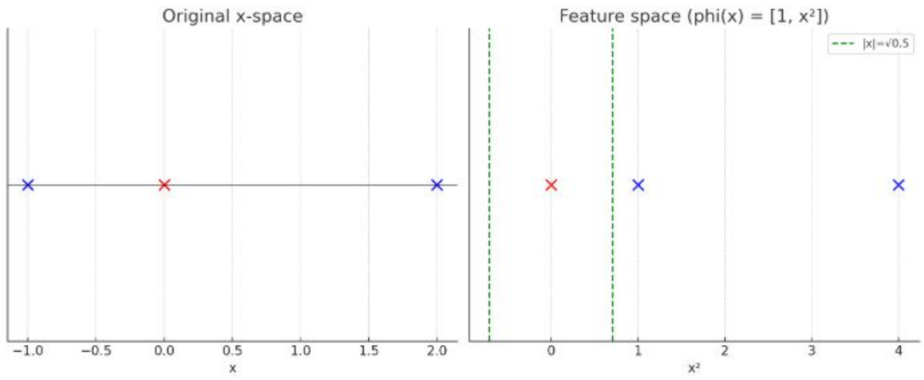
a) True

b) False

Question-2- explain

State true or false: The dataset becomes linearly separable after using basis expansion with the following basis function $\phi(x)=[1\ x^2]^T$

x1	x2	y
1	1	1
1	0	-1
1	4	1



Question-2- Correct answer

State true or false: The dataset becomes linearly separable after using basis expansion with the following basis function $\phi(x)=[1\ x^2]^T$

a) True

b) False

Correct options: (a)

Question-3

01:00

For a binary classification problem with the hinge loss function max(0,1 −y(w · x)) , which of the following statements is correct?

a) The loss is zero only when the prediction is exactly equal to the true label
b) The loss is zero when the prediction is correct and the margin is at least 1
c) The loss is always positive
d) The loss increases linearly with the distance from the decision boundary regardless of classification

Question-3 - Correct answer

For a binary classification problem with the hinge loss function $\max(0,1 - y(w \cdot x))$, which of the following statements is correct?

a) The loss is zero only when the prediction is exactly equal to the true label

b) The loss is zero when the prediction is correct and the margin is at least 1

c) The loss is always positive

d) The loss increases linearly with the distance from the decision boundary regardless of classification

Correct options: (b)

Question-4

03:00

Consider the following dataset:

Which of these is not a support vector when using a Support Vector Classifier with a polynomial kernel with degree = 3, C = 1, and gamma = 0.1?

(We recommend using sklearn to solve this question.)

x	y
1	1
2	1
4	-1
5	-1
6	-1
7	-1
9	1
10	1

a) 2
b) 1
c) 9
d) 10

Question-4 - Correct answer

Consider the following dataset:

Which of these is not a support vector when using a Support Vector Classifier with a polynomial kernel with degree = 3, C = 1, and gamma = 0.1?

(We recommend using sklearn to solve this question.)

a) 2

b) 10

c) 9

d) 1

Correct options: (b)

Question-5

01:00

For a dataset with n points in d dimensions, what is the maximum number of support vectors possible in a hard-margin SVM?

a) 2
b) d
c) n/2
d) n

Question-5 - Correct answer

For a dataset with n points in d dimensions, what is the maximum number of support vectors possible in a hard-margin SVM?

a) 2
b) d
c) n
d) n/2

Correct options: (c)

Question-6

01:00

In the context of soft-margin SVM, what happens to the number of support vectors as the parameter C increases?

a) Generally increases

b) Generally decreases

c) Remains constant

d) Changes unpredictably

Question-6 - Correct answer

In the context of soft-margin SVM, what happens to the number of support vectors as the parameter C increases?

a) Generally increases

b) Generally decreases

c) Remains constant

d) Changes unpredictably

Correct options: (b)

THANK YOU

Suggestions and Feedback

Next Session:

Sunday: 24-Aug-2025
3:00 - 5:00 PM