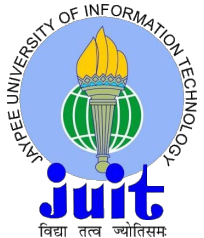


Machine Learning

(Course Code: 18B1WCI634)

LR and SVM

Mr. Sandeep Kumar Patel

Assistant Professor

Jaypee University of Information Technology

Waknaghat, Solan, HP-173234

What is Logistic Regression?

- A supervised learning algorithm
- Used for binary classification
- Outputs probability between 0 and 1

Sigmoid Function

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

- Converts linear output to probability
- Range: $(0, 1)$

Decision Boundary

Sigmoid Function with Decision Rule

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1 \mid x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Example

- Input: Hours studied
- Output: Pass/Fail
- Predicts probability of passing

Advantages

- Simple and easy to implement
- Fast computation
- Interpretable results

Limitations

- Assumes linearity
- Not suitable for complex data

Support Vector Machine (SVM)

Definition

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression by finding an optimal separating hyperplane.

- Maximizes margin between classes
- Uses only critical points (support vectors)
- Effective in high-dimensional spaces

Hyperplane

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

- w = weight vector (normal to hyperplane)
- b = bias term

Classification Rule:

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

Functional Margin

$$\gamma_i = y_i(w^T x_i + b)$$

- Measures confidence of classification
- $\gamma_i > 0 \rightarrow$ correctly classified
- Depends on scaling of w

Why Functional Margin is Not Enough?

- If we scale w and b, value of γ_i changes
- So it is not a reliable measure of distance

We need a scale-invariant measure $\rightarrow$ Geometric Margin

Geometric Margin

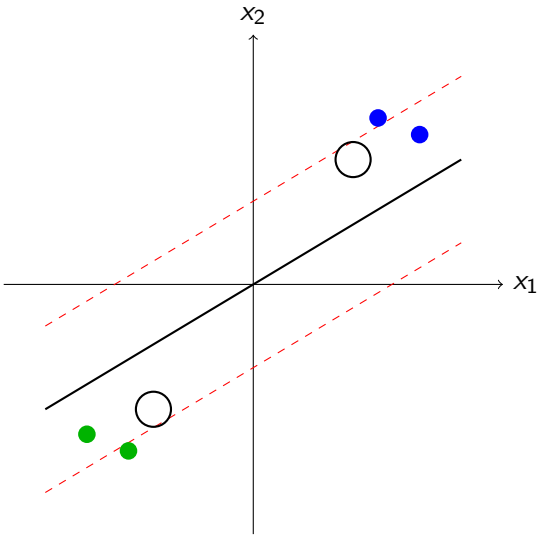
$$\hat{\gamma}_i = \frac{y_i(w^T x_i + b)}{\|w\|}$$

- Actual perpendicular distance from hyperplane
- Scale invariant
- Used in SVM optimization

Margin Intuition

- Margin = distance between two decision boundaries
- Boundaries: $w^T x + b = +1$ and $w^T x + b = -1$

SVM Visualization (Hyperplane + Margin)



Derivation of Margin

Step 1: Parallel Hyperplanes

$$w^T x + b = +1 \quad \text{and} \quad w^T x + b = -1$$

Step 2: Distance Formula

$$\text{Distance} = \frac{|c_1 - c_2|}{\|w\|}$$

Step 3: Apply to SVM

$$\text{Margin} = \frac{|1 - (-1)|}{\|w\|} = \frac{2}{\|w\|}$$

Final Result:

$$\boxed{\text{Margin} = \frac{2}{\|w\|}}$$

SVM Objective

- Maximize margin:

$$\max \frac{2}{\|w\|} \quad \Rightarrow \quad \min \frac{1}{2}\|w\|^2$$

- Leads to optimal separating hyperplane

Functional Margin Problem

Problem:

Given:

$$w = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \quad b = -1$$

Data points:

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; y_1 = +1$$

$$x_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \; y_2 = -1$$

Compute the functional margin for each point.

Solution

$$\gamma_i = y_i(w^T x_i + b)$$

For x_1:

$$w^T x_1 = (2)(1) + (-1)(2) = 2 - 2 = 0$$

$$\gamma_1 = 1(0 - 1) = -1$$

For x_2:

$$w^T x_2 = (2)(2) + (-1)(1) = 4 - 1 = 3$$

$$\gamma_2 = (-1)(3 - 1) = -2$$

Final Answer:

- $\gamma_1 = -1$
- $\gamma_2 = -2$

Geometric Margin Problem

Recall:

$$\hat{\gamma}_i = \frac{y_i(w^T x_i + b)}{\|w\|}$$

Given:

$$w = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \quad b = -1$$

From previous result:

$$\gamma_1 = -1, \quad \gamma_2 = -2$$

Compute geometric margins.

Hard Margin Optimization

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

$$\text{subject to } y_i(w^T x_i + b) \geq 1$$

- Perfect separation
- No misclassification allowed

Soft Margin SVM

- Allows some errors using slack variables ξ_i

$$\min \frac{1}{2}||w||^2 + C\sum \xi_i$$

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

- C controls trade-off

Kernel Trick

Idea

Map data into higher-dimensional space to make it linearly separable.

$$x \rightarrow \phi(x)$$

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Kernel Functions Overview

Idea: Kernel functions measure similarity between two data points and enable non-linear classification.

Common Kernels:

$$\text{Linear: } K(x_i, x_j) = x_i^T x_j$$

$$\text{Polynomial: } K(x_i, x_j) = (x_i^T x_j + c)^d$$

$$\text{RBF: } K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

$$\text{Sigmoid: } K(x_i, x_j) = \tanh(x_i^T x_j + c)$$

Linear and Polynomial Kernels

Linear Kernel

$$K(x_i, x_j) = x_i^T x_j$$

- Simple dot product
- Works for linearly separable data

Polynomial Kernel

$$K(x_i, x_j) = (x_i^T x_j + c)^d$$

- Captures feature interactions
- d controls complexity

RBF and Sigmoid Kernels

RBF (Gaussian) Kernel

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

- Based on distance
- Most commonly used kernel

Sigmoid Kernel

$$K(x_i, x_j) = \tanh({x_i}^T x_j + c)$$

- Similar to neural activation
- Rarely used in practice

Numerical Example (Linear & Polynomial)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Linear Kernel

Numerical Example (Linear & Polynomial)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Linear Kernel

$$K = (1)(3) + (2)(4) = 11$$

Polynomial Kernel ($d = 2, c = 1$)

Numerical Example (Linear & Polynomial)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Linear Kernel

$$K = (1)(3) + (2)(4) = 11$$

Polynomial Kernel ($d = 2, c = 1$)

$$K = (11 + 1)^2 = 144$$

Numerical Example (RBF & Sigmoid)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

RBF Kernel ($\gamma = 0.5$)

Numerical Example (RBF & Sigmoid)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

RBF Kernel ($\gamma = 0.5$)

$$\|x_i - x_j\|^2 = 8$$

$$K = \exp(-4) \approx 0.0183$$

Sigmoid Kernel ($c = 1$)

Numerical Example (RBF & Sigmoid)

Given:

$$x_i = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad x_j = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

RBF Kernel ($\gamma = 0.5$)

$$\|x_i - x_j\|^2 = 8$$

$$K = \exp(-4) \approx 0.0183$$

Sigmoid Kernel ($c = 1$)

$$K = \tanh(12) \approx 1$$

Kernel SVM Decision Function

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

- Uses support vectors only
- Works in implicit feature space

Kernel Function vs Kernel SVM

Kernel Function

- A mathematical function that measures similarity between two data points
- Output: a scalar similarity value
- Examples: Linear, Polynomial, RBF, Sigmoid
- Does not perform classification

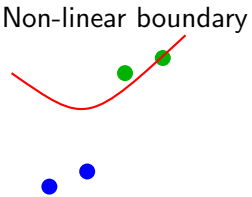
Kernel SVM

- A machine learning algorithm for classification
- Uses kernel functions to handle non-linear data
- Learns optimal decision boundary using training data

Key Difference:

- Kernel function = similarity measure
- Kernel SVM = classification model using kernel functions

Non-linear Decision Boundary



SVM

Advantages

- Works well in high dimensions
- Effective with small datasets
- Robust to overfitting

Limitations

- Computationally expensive for large data
- Choosing kernel is difficult

Conclusion

- SVM finds optimal separating hyperplane
- Kernel trick enables non-linear classification

$$\textbf{SVM} = \textbf{Maximum Margin} + \textbf{Kernel Trick}$$