Linear Boundaries for Binary Classification
A Comprehensive Study Guide

Machine Learning Course

Abstract

This document provides a comprehensive explanation of linear boundaries for binary classification, synthesizing material from both lecture slides and the corresponding transcript. It covers the geometric intuition behind linear classifiers, their mathematical formulation in both two and higher dimensions, and lays the groundwork for understanding the perceptron algorithm. The material is organized to build intuition progressively while maintaining mathematical rigor.

Contents

1 Introduction and Context

> **Slide Note:** We've covered regression and conditional probability estimation, both reformulated as optimization tasks. Now we'll apply optimization to classification.

> **Lecture Note:** We have spent a little bit of time talking about regression and conditional probability estimation. And in each case, we approach the learning problem by reformulating it as an optimization task. Now, let's circle back round to classification and bring the same optimization mindset to it.

Classification is a fundamental task in machine learning where we aim to assign discrete labels to data points. In this study guide, we focus specifically on binary classification using linear boundaries—one of the most elegant and foundational approaches in machine learning.

2 Geometric Intuition of Linear Classification

2.1 Linear Decision Boundaries in Two Dimensions

Slide Note: Decision boundary in 2D space with y-intercept = 4, slope = -4/3

Let's begin with a concrete example in two dimensions to build intuition. Consider a 2D feature space where each point has coordinates (x_1, x_2).
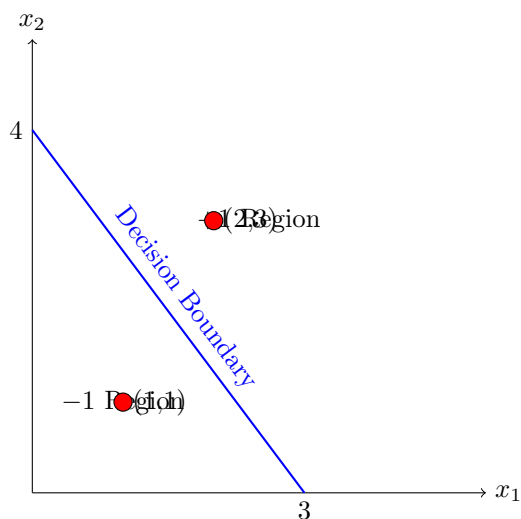

Figure 1: A linear decision boundary in 2D space with positive and negative regions

Lecture Note: Let's start with a quick review of the geometry of linear classification. So here, we have a decision boundary in two dimensions. What is the formula for this boundary? Let's go ahead and derive that.

The equation of this line can be derived as follows:

$$x_2 = -\frac{4}{3}x_1 + 4 \quad \text{(slope-intercept form)} \tag{1}$$

We can rearrange this to get the standard form:

$$\frac{4}{3}x_1 + x_2 - 4 = 0 \tag{2}$$

$$\Rightarrow 4x_1 + 3x_2 - 12 = 0 \quad \text{(multiplied by 3)} \tag{3}$$

Key Insight: Linear decision boundaries will always be expressed in the form of a linear function equal to zero. This standardization simplifies both the mathematical analysis and implementation.

2.2 Making Predictions with Linear Boundaries

Once we have the equation of the decision boundary, how do we use it to classify points? The key insight is that the linear function divides the space into two regions:

- Points where the function evaluates to positive values

- Points where the function evaluates to negative values

Slide Note: For point (2,3): $4(2) + 3(3) - 12 = 8 + 9 - 12 = 5 > 0 \Rightarrow$ Predict +1

Let's evaluate our linear function $f(x) = 4x_1 + 3x_2 - 12$ at the point $(2, 3)$:

$$f(2,3) = 4(2) + 3(3) - 12 \tag{4}$$
$$= 8 + 9 - 12 \tag{5}$$
$$= 5 \tag{6}$$

Since $f(2, 3) > 0$, we classify this point as $+1$.

Slide Note: For point (1,1): $4(1) + 3(1) - 12 = 4 + 3 - 12 = -5 < 0 \Rightarrow$ Predict -1

Similarly, for the point $(1, 1)$:

$$f(1,1) = 4(1) + 3(1) - 12 \tag{7}$$
$$= 4 + 3 - 12 \tag{8}$$
$$= -5 \tag{9}$$

Since $f(1, 1) < 0$, we classify this point as -1.

Classification Rule

For a linear function $f(x) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b$:

- If $f(x) > 0$, classify as $+1$

- If $f(x) < 0$, classify as -1

- If $f(x) = 0$, the point lies exactly on the decision boundary

---

Lecture Note: At any given point, the way we make a prediction is to just plug the point into this linear function. If the result is positive, we predict plus one. If the result is negative, we predict minus one. It's very simple.

3 Generalization to D-Dimensional Space

3.1 Mathematical Formulation

While the 2D case is easy to visualize, the real power of linear classification comes from its ability to generalize to higher dimensions.

Slide Note: Data points $\mathbf{x} \in \mathbb{R}^d$, Labels $y \in \{-1, +1\}$

In the general case:

- Data points are vectors in d-dimensional space: $\mathbf{x} \in \mathbb{R}^d$

- Labels remain binary: $y \in \{-1, +1\}$

A linear classifier in d dimensions is defined by:

- A weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$

- A bias term (or offset) $b \in \mathbb{R}$

The decision boundary is the hyperplane defined by:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{10}$$

where $\mathbf{w} \cdot \mathbf{x} = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$ is the dot product.

Lecture Note: A linear classifier is now given by a vector, W, a D dimensional vector, as well as an offset b. The form of the boundary is w . x plus b equals zero.

Key Insight: The dot product $\mathbf{w} \cdot \mathbf{x}$ can be interpreted geometrically as the projection of $\mathbf{x}$ onto the direction of $\mathbf{w}$, scaled by the magnitude of $\mathbf{w}$. The bias term b shifts the decision boundary away from the origin.

3.2 Prediction Rule in Higher Dimensions

The prediction rule generalizes naturally from the 2D case:

$$\hat{y} = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \tag{11}$$

This can be written more compactly using the sign function:

$$\hat{y} = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{12}$$

Lecture Note: And in when we wanna make a prediction on a new point x, we simply compute the linear function, and then look at the sign of the outcome. If it's positive, we predict plus one. If it's negative, we predict minus one.

3.3 Correctness Condition

How do we know if our prediction is correct? Let y be the true label of a point $\mathbf{x}$.

Slide Note: Prediction is correct if and only if: True label $y = +1$ and $\mathbf{w} \cdot \mathbf{x} + b > 0$, OR True label $y = -1$ and $\mathbf{w} \cdot \mathbf{x} + b < 0$

Our prediction is correct if and only if one of these conditions holds:

- $y = +1$ and $\mathbf{w} \cdot \mathbf{x} + b > 0$

- $y = -1$ and $\mathbf{w} \cdot \mathbf{x} + b < 0$

Slide Note: Compact form: $y(\mathbf{w} \cdot \mathbf{x} + b) > 0$

These conditions can be combined into a single, elegant expression:

$$y(\mathbf{w} \cdot \mathbf{x} + b) > 0 \tag{13}$$

Lecture Note: A more compact way to write this is simply to say that the true label, times w . x plus b should be greater than zero. That is either both of these things are positive, or both of them are negative. So instead of having two conditions like we have over here, we just have a single compact condition, and we are gonna make heavy use of this particular formulation.

Key Insight: This compact formulation is not just mathematically elegant—it will be the foundation for the perceptron algorithm and many other classification methods. It effectively measures whether the prediction has the same sign as the true label.

4   Geometric Interpretation

To deepen our understanding, let's explore the geometric interpretation of linear classification:

- The weight vector $\mathbf{w}$ is perpendicular (normal) to the decision boundary hyperplane

- The magnitude of $\mathbf{w}$ doesn't affect the orientation of the boundary, only its scale

- The bias term b determines how far the hyperplane is from the origin

- The signed distance from a point $\mathbf{x}$ to the decision boundary is proportional to $\frac{\mathbf{w}\cdot\mathbf{x}+b}{||\mathbf{w}||}$
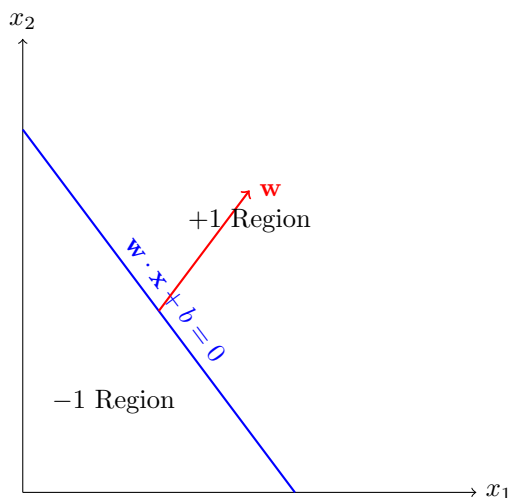

Figure 2: Geometric interpretation of the weight vector $\mathbf{w}$ as normal to the decision boundary

5   Connection to the Perceptron Algorithm

Slide Note: We'll use this formulation to develop the perceptron algorithm—one of the most influential algorithms for linear classification.

The formulation we've developed provides the foundation for the perceptron algorithm, which is one of the earliest and most influential algorithms for linear classification.

Lecture Note: What we'll be doing next is to use this formulation to write down one of the most beautiful and influential algorithms for linear classification, the perceptron.

The perceptron algorithm:

- Iteratively updates the weight vector $\mathbf{w}$ and bias b based on misclassified points

- Uses the correctness condition $y(\mathbf{w} \cdot \mathbf{x} + b) > 0$ to identify misclassifications

- Converges to a solution if the data is linearly separable (i.e., if a perfect linear boundary exists)

Key Insight: The perceptron algorithm is not just historically significant—it forms the conceptual foundation for neural networks and deep learning. A modern neural network can be viewed as multiple layers of perceptron-like units with non-linear activation functions.

6   Practice Problems

To solidify your understanding, try these practice problems:

Problem 1

Consider the linear boundary $2x_1 - 3x_2 + 6 = 0$ in 2D space.

(a) Sketch this boundary.

(b) Determine which side corresponds to the positive region.

(c) Classify the points $(1, 2)$, $(3, 1)$, and $(0, 0)$.

Problem 2

For a linear classifier in 3D with $\mathbf{w} = (1, 2, -1)$ and $b = -3$:

(a) Write the equation of the decision boundary.

(b) Classify the points $(2, 1, 0)$, $(1, 0, 1)$, and $(0, 0, 0)$.

(c) Is the point $(1, 1, 1)$ correctly classified if its true label is -1?

Problem 3

Prove that scaling the weight vector $\mathbf{w}$ and bias b by the same positive constant does not change the predictions of a linear classifier.

7   Summary and Key Takeaways

- Linear classification uses a hyperplane to separate data into two classes

- In 2D, the decision boundary is a line; in higher dimensions, it's a hyperplane

- A linear classifier is defined by a weight vector $\mathbf{w}$ and bias b

- The decision rule is based on the sign of $\mathbf{w} \cdot \mathbf{x} + b$

- A prediction is correct when $y(\mathbf{w} \cdot \mathbf{x} + b) > 0$

- This formulation is the foundation for the perceptron algorithm

Key Insight: Linear classification, despite its simplicity, remains a powerful tool in machine learning. It provides excellent interpretability, computational efficiency, and serves as a building block for more

complex models. Understanding linear boundaries thoroughly is essential for mastering more advanced classification techniques.

8 Further Reading

- The Perceptron Algorithm and its convergence properties

- Support Vector Machines: Optimal linear boundaries with maximum margin

- Logistic Regression: Probabilistic interpretation of linear boundaries

- Neural Networks: From perceptrons to deep learning