# Optimization Techniques for Machine Learning

AMLZC326 · #14 Machine Learning II

Anshid Aboobacker

# Table of Contents

# The Central Question

Among all separating lines,

**why choose one over another?**

# Learning Objectives

By the end of this lecture you should be able to:

- Motivate the maximum-margin classifier geometrically
- Derive the hard margin SVM as a constrained quadratic program
- Extend to soft margin SVM with slack variables and the regularisation parameter $C$
- Interpret hinge loss as the convex surrogate for 0–1 misclassification loss

# Linear Classification

- We want to separate data into two classes: $y \in \{+1, -1\}$

$$f(x) = \text{sign}(w^T x + b)$$

- Decision boundary:

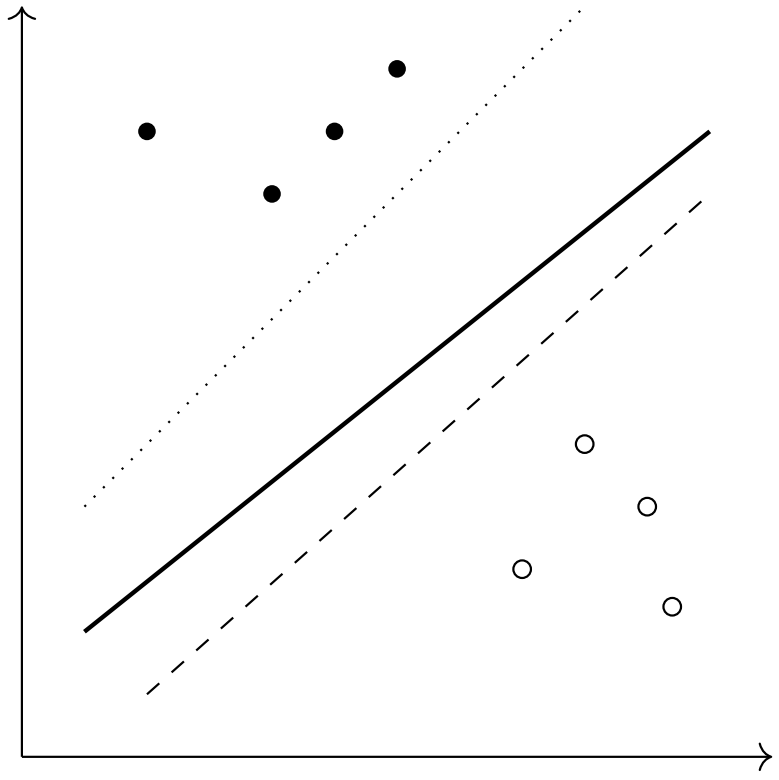
$$w^T x + b = 0$$

# Geometric Interpretation

- $w^T x + b > 0 \Rightarrow +1$
- $w^T x + b < 0 \Rightarrow -1$

**A hyperplane separates the data**

# Linearly Separable Data



Many separating lines exist

# Key Observation

All these classifiers perfectly separate the data

**Which one should we choose?**

# Scale Invariance

- The classifier depends on:

$$\text{sign}(w^T x + b)$$

- If we scale:

$$(w, b) \rightarrow (\lambda w, \lambda b)$$

- Then:

$$\text{sign}(w^T x + b) = \text{sign}(\lambda w^T x + \lambda b)$$

**Same classifier, different parameters**

# Normalization Idea

- Since scaling does not change classification,
- we can fix a scale:

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

- This removes ambiguity

# What Just Happened?

- Classification depends only on sign
- But geometry depends on magnitude

**We are now ready to define margin**

# Table of Contents

# What is a Hyperplane?

- Decision boundary:

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

- In $\mathbb{R}^2$: a line
- In $\mathbb{R}^3$: a plane
- In $\mathbb{R}^d$: a hyperplane

**This is the set of points where classification changes**

# Why is $w$ perpendicular?

Let $x_a$, $x_b$ lie on the hyperplane:

$$w^T x_a + b = 0, \quad w^T x_b + b = 0$$

Subtract:

$$w^T (x_a - x_b) = 0$$

- $(x_a - x_b)$ lies along the hyperplane
- Dot product = 0 $\Rightarrow$ orthogonal

**$w$ is normal to the hyperplane**

# Geometric Meaning of $w$

- Direction of $w$:
  - Points toward positive class
- Magnitude of $w$:
  - Controls how fast values change

**$w$ controls geometry, not just classification**

# Distance from a Point to Hyperplane

$$D = \frac{|w^T x + b|}{\|w\|}$$

- Numerator: raw score
- Denominator: normalization by scale

**Distance depends inversely on $\|w\|$**

# Signed Distance

$$\text{Signed distance} = \frac{w^T x + b}{\|w\|}$$

- Positive $\Rightarrow$ one side
- Negative $\Rightarrow$ other side

**This connects geometry to classification**

# Key Insight

- Classification uses:

$$\text{sign}(w^T x + b)$$

- Distance uses:

$$\frac{w^T x + b}{\|w\|}$$

**Only $\|w\|$ controls margin**

# Transition to Margin

We now know:

- $w$ defines direction
- $\|w\|$ defines scale of distance

**So what is the best hyperplane?**

# Table of Contents

# From Classification to Geometry

We don't just want to classify correctly

**We want the boundary to be as far as possible from all points**

# Normalization Recall

We fixed scale using:

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

- Closest points satisfy:

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

**These define the margin boundaries**

# Supporting Hyperplanes

$$w^T x + b = +1$$

$$w^T x + b = -1$$

- Parallel to decision boundary
- Touch closest points

**These are the supporting hyperplanes**

# What is the Margin?

Distance between the two supporting hyperplanes

**We want to maximize this distance**

# Deriving the Margin

Take one point on each hyperplane:

$$w^T x^+ + b = 1$$

$$w^T x^- + b = -1$$

Subtract:

$$w^T (x^+ - x^-) = 2$$

Project onto direction of $w$:

$$\text{distance} = \frac{2}{\|w\|}$$

# Margin Formula

$$M = \frac{2}{\|w\|}$$

Maximize margin $\iff$ minimize $\|w\|$

# Optimization Insight

Instead of:

$$\max \frac{2}{\|w\|}$$

We solve:

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

- Same solution
- Easier to optimize

# Who Defines the Margin?

Only the closest points matter

**These points are called support vectors**

# Key Insight

- Far away points do not affect the boundary
- Only boundary-touching points matter

**SVM depends only on a few critical points**

# Transition

We now know:

- What to optimize
- What constraints to enforce

**Let's turn this into an optimization problem**

# Table of Contents

# What Have We Built So Far?

- Linear classifier:

$$f(x) = \text{sign}(w^T x + b)$$

- Margin:

$$M = \frac{2}{\|w\|}$$

- Goal:

Maximize margin

# Rewriting the Goal

$$\max \frac{2}{\|w\|}$$

Equivalent to:

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

**Cleaner, convex objective**

# What About Correct Classification?

We must ensure all points are classified correctly:

$$y_i(w^T x_i + b) > 0$$

After normalization:

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

**Encodes both correctness + margin**

# Hard Margin SVM Formulation

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

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

**This is the hard margin SVM**

# Understanding the Constraints

- If $y_i = +1$:

$$w^T x_i + b \geq 1$$

- If $y_i = -1$:

$$w^T x_i + b \leq -1$$

**Points lie outside margin boundaries**

# Geometric Meaning

- Decision boundary:

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

- Margin boundaries:

$$w^T x + b = \pm 1$$

**All points lie outside the margin**

# Important Observation

Only points satisfying

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

matter

**These are the support vectors**

# Nature of the Problem

- Objective: quadratic
- Constraints: linear

**Convex optimization problem**

**Unique global solution**

# Limitation of Hard Margin

What if data is not perfectly separable?

**Hard margin fails completely**

# TRANSITION

We need flexibility

**Allow some violations of constraints**

# Table of Contents

# Why Hard Margin Fails

Real-world data is noisy

**Perfect separation may not exist**

# The Problem

- Even one outlier can break hard margin SVM

**We need to allow mistakes**

# Introducing Slack Variables

We relax constraints:

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

$$\xi_i \geq 0$$

$\xi_i$ **measures violation**

# Meaning of $\xi_i$

- $\xi_i = 0 \rightarrow$ correctly classified, outside margin
- $0 < \xi_i < 1 \rightarrow$ inside margin
- $\xi_i > 1 \rightarrow$ misclassified

**Slack = how bad the violation is**

# New Objective

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

- First term $\rightarrow$ maximize margin
- Second term $\rightarrow$ penalize violations

# Role of $C$

- Large $C$:
  - Penalize mistakes heavily
  - Smaller margin
- Small $C$:
  - Allow more violations
  - Larger margin

**$C$ controls the trade-off**

# Geometric Picture

- Some points allowed inside margin
- Some points even misclassified

**We trade perfection for robustness**

# Soft Margin SVM Formulation

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

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

$$\xi_i \geq 0$$

# Key Insight

We are balancing:

- Margin size
- Classification errors

**This is a trade-off problem**

# Transition

Can we write this without slack variables?

**Yes — using a loss function**

# Table of Contents

# From Constraints to Loss

We had:

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

Can we eliminate $\xi_i$?

# Key Observation

Constraint implies:

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

**Minimum $\xi_i$ is:**

$$\xi_i = \max(0,\ 1 - y_i(w^T x_i + b))$$

# Hinge Loss Definition

$$L_i = \max(0,\ 1 - y_i(w^T x_i + b))$$

- $0 \rightarrow$ correct and outside margin
- $> 0 \rightarrow$ violation

# Interpretation

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

$$L_i = 0$$

- If $y_i(w^T x_i + b) < 1$:

$$L_i > 0$$

**Loss depends on margin, not just correctness**

# Final Objective

$$J(w, b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \max(0,\ 1 - y_i(w^T x_i + b))$$

- First term $\rightarrow$ regularization
- Second term $\rightarrow$ hinge loss

# Big Picture

SVM = Regularization + Loss

Minimize Complexity + Error

# Why Hinge Loss?

- Penalizes margin violations
- Encourages large margin
- Convex $\rightarrow$ easy to optimize

**Designed for maximum margin classification**

# Key Insight

Classification is not enough

**Confidence (margin) matters**

# Connection to Modern ML

- Linear regression $\rightarrow$ squared loss
- Logistic regression $\rightarrow$ log loss
- SVM $\rightarrow$ hinge loss

**Same framework, different loss**

# Final Takeaway

SVM is not a special trick

**It is a principled optimization problem**

Thank you :)