CS 412 — Introduction to Machine Learning (UIC)	February 25, 2025
Lecture 11	
Instructor: Aadirupa Saha	*Scribe(s): Harsh Kothari*

Overview

In the last lecture, we covered the following main topics:

1. Gradient Descent Convergence Analysis
2. Stochastic Gradient Descent + Convergence Guarantees
3. Batched SGD
4. Variants of Gradient Descent

This lecture focuses on:

1. Primer on ”Vector Algebra” & Margin Computation
2. Understanding Hyperplanes and Their Properties
3. Support Vector Machine Conditions (SVM)
4. Optimization Objective for SVM

1 Primer on ”Vector Algebra” & Margin Computation

1.1 Geometry & Vector Algebra Primer

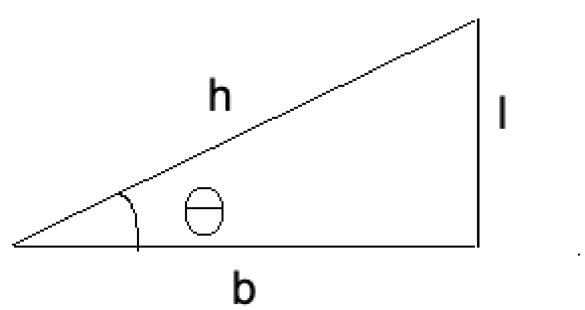


Figure 1: A traingle with thetha angle between b and h .

- Basic trigonometric relationships:

$$\cos\theta = \frac{b}{h}$$
$$\sin\theta = \frac{l}{h}$$
$$\tan\theta = \frac{l}{b}$$

1.2 Dot Product& Projection

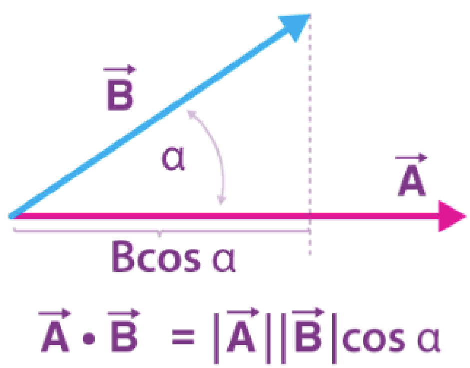


Figure 2: Two vector at alpha angle .

Problem Statement

Prove in 2D, assuming polar representations of vectors $\mathbf{v}$ and $\mathbf{w}$:

$$\mathbf{v} = (||\mathbf{v}||\cos\theta_1, ||\mathbf{v}||\sin\theta_1)$$

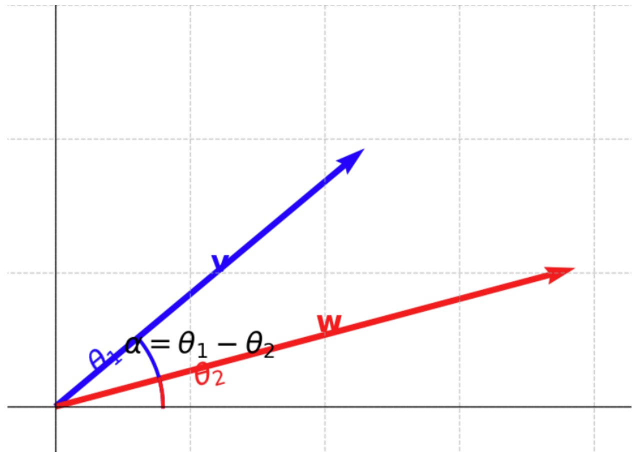
$$\mathbf{w} = (||\mathbf{w}||\cos\theta_2, ||\mathbf{w}||\sin\theta_2)$$

where the angle difference is defined as:

$$\alpha = \theta_1 - \theta_2$$

Hint: You need to apply the cosine angle difference identity:

$$\cos(\theta_1 - \theta_2) = \cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2$$



Solution

Step 1: Compute the Dot Product

The dot product of two vectors in 2D is given by:

$$\mathbf{v} \cdot \mathbf{w} = v_x w_x + v_y w_y$$

Substituting the given vector components:

$$\mathbf{v} \cdot \mathbf{w} = (||\mathbf{v}|| \cos\theta_1)(||\mathbf{w}|| \cos\theta_2) + (||\mathbf{v}|| \sin\theta_1)(||\mathbf{w}|| \sin\theta_2)$$

Factor out the magnitudes $||\mathbf{v}||||\mathbf{w}||$:

$$\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||||\mathbf{w}||(\cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2)$$

Step 2: Apply the Cosine Angle Difference Identity

From trigonometry, we know that:

$$\cos(\theta_1 - \theta_2) = \cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2$$

Using this identity in our equation:

$$\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||||\mathbf{w}|| \cos(\theta_1 - \theta_2)$$

Since we defined $\alpha = \theta_1 - \theta_2$, we rewrite it as:

$$\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||||\mathbf{w}|| \cos\alpha$$

Conclusion

This confirms the well-known dot product formula in terms of magnitudes and angles:

$$\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||||\mathbf{w}|| \cos\alpha$$

Thus, we have successfully proved the relation using the given polar representations of the vectors.

2 Understanding Hyperplanes and Their Properties

2.1 Definition of a Hyperplane

A hyperplane is a geometric concept that represents a subspace of one dimension less than its ambient space. In different dimensions:

- In **2D**, a hyperplane is a **straight line**.
- In **3D**, a hyperplane is a **flat plane**.
- In **d-dimensions**, a hyperplane is a **(d-1)-dimensional subspace** that divides the space into two halves.

2.2 Equation of a Hyperplane in 2D

A hyperplane (which is a line in 2D) can be represented as:

$$mx_1 + b = x_2 \tag{1}$$

Rearranging this equation:

$$mx_1 - x_2 + b = 0 \tag{2}$$

To express this in matrix form:

$$\begin{pmatrix} m & -1 & b \end{pmatrix} \begin{pmatrix} x_1 \, x_2 \, 1 \end{pmatrix} = 0 \tag{3}$$

This equation matches the general hyperplane equation:

$$\mathbf{w}^T \mathbf{x} + b = 0 \tag{4}$$

where:

- $\mathbf{w} = \begin{pmatrix} m & -1 \, b \end{pmatrix}$ is the **normal vector**.
- $\mathbf{x} = \begin{pmatrix} x_1 \, x_2 \, 1 \end{pmatrix}$ represents a **point on the hyperplane**.
- b is the **bias term** that shifts the hyperplane.

2.3 General Form of a Hyperplane in d-Dimensions

In higher dimensions, a hyperplane is defined as:

$$\mathbf{w}^T \mathbf{x} + b = 0 \tag{5}$$

which expands to:

$$\begin{pmatrix} w_1 & w_2 & \dots & w_d & b \end{pmatrix} \begin{pmatrix} x_1 \, x_2 \vdots x_d \end{pmatrix} = 0 \tag{6}$$

where:

- $\mathbf{w} = (w_1, w_2, ..., w_d)$ is the **normal vector**.
- $\mathbf{x} = (x_1, x_2, ..., x_d)$ represents a **point on the hyperplane**.
- b is the bias term.

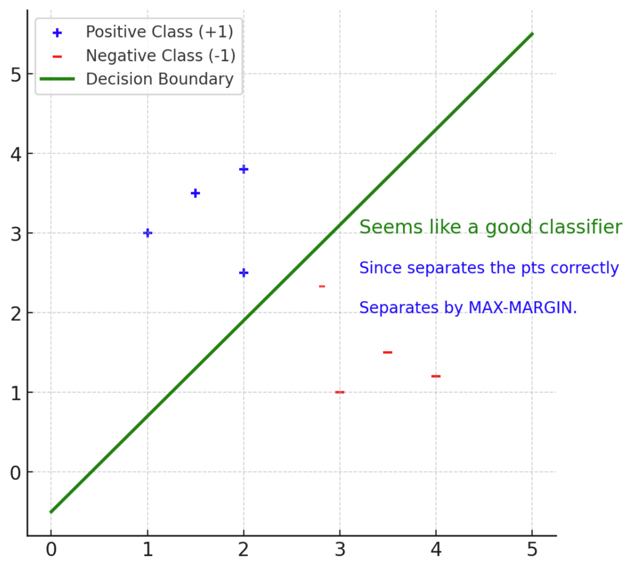


Figure 3: A hyperplane in 2D space which divide + and - classes .

2.4 Orthogonality of the Normal Vector

Mathematical Explanation
The normal vector $\mathbf{w}$ is perpendicular to the hyperplane. Consider two points $\mathbf{x}_1$ and $\mathbf{x}_2$ that lie on the hyperplane:

$$\mathbf{w}^T\mathbf{x}_1 + b = 0 \tag{7}$$

$$\mathbf{w}^T\mathbf{x}_2 + b = 0 \tag{8}$$

Subtracting these equations:

$$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0 \tag{9}$$

Since $\mathbf{x}_1 - \mathbf{x}_2$ is a vector **along the hyperplane**, this equation states that $\mathbf{w}$ is perpendicular to all such vectors.
Geometric Intuition

- A hyperplane divides space into two regions.
- The normal vector $\mathbf{w}$ **points in the direction perpendicular to the hyperplane**.
- Any movement along the hyperplane does not change the dot product with $\mathbf{w}$, reinforcing its orthogonality.
- This is similar to how a ceiling fan's rod is perpendicular to the floor—any movement along the floor does not affect its height.

2.5 Key Takeaways

- **In 2D, a hyperplane is a straight line; in 3D, it is a flat plane; in d-dimensions, it is a (d-1)-dimensional subspace.**

- **The general equation of a hyperplane is $\mathbf{w}^T\mathbf{x} + b = 0$.**
- **The normal vector $\mathbf{w}$ is always perpendicular to the hyperplane.**
- **Hyperplanes play a key role in classification, optimization, and geometry.**

3 Support Vector Machines (SVMs)

3.1 Introduction to SVMs

Support Vector Machines (SVMs) are a type of supervised learning model used for classification and regression tasks. They are particularly powerful in binary classification problems.

3.2 Problem Setup

Assume we are given a set of **data points**:

$$D = (x_i, y_i)_{i=1}^{N} \tag{10}$$

where:

- $x_i \in \mathbb{R}^d$ (each data point is a d-dimensional vector).
- $y_i \in -1, 1$ (labels are either **+1 (positive class)** or **-1 (negative class)**).
- N represents the total number of data points.

3.3 Objective of SVM

The goal of SVM is to **find a classifier that separates the positive and negative labels as much as possible**. This is done by constructing a **decision boundary (hyperplane)** that maximizes the margin between the two classes.

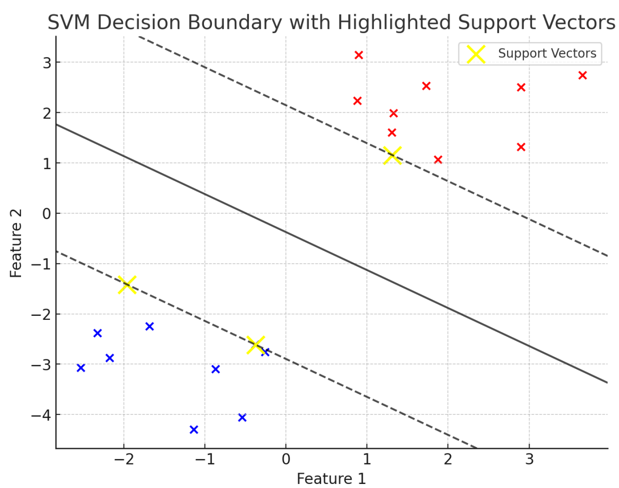


Figure 4: Hyperplane with Support Vectors .

3.4 Understanding Linearly and Non-Linearly Separable Data

3.4.1 Linearly Separable Data

A dataset D is considered **linearly separable** if there exists a **hyperplane** that perfectly separates the data points into two distinct classes:

- **Positive class (+1)** on one side.
- **Negative class (-1)** on the other side.

In such cases, a **linear classifier (such as an SVM with a linear kernel)** can correctly classify the data.

3.4.2 Examples of Linearly and Non-Linearly Separable Data

Example 1: Linearly Separable Data

- A **straight line (or hyperplane in higher dimensions)** can perfectly separate the two classes.
- The **red line** in the first diagram represents such a **decision boundary**.
- **SVM with a linear kernel** is effective here.

Example 2: Non-Linearly Separable Data (Encircled Cluster)

- A **single straight line cannot separate the two classes**.
- The data forms a **circular pattern**, requiring a **non-linear decision boundary**.
- A **kernel trick (e.g., RBF kernel in SVM)** can help **map the data to a higher-dimensional space** where separation is possible.

Example 3: Non-Linearly Separable Data (Wavy Pattern)

- The decision boundary is **highly complex and nonlinear**.
- A simple **hyperplane is insufficient** to separate the classes.
- A more advanced technique such as **polynomial or RBF kernel SVM, neural networks, or deep learning models** may be needed for classification.

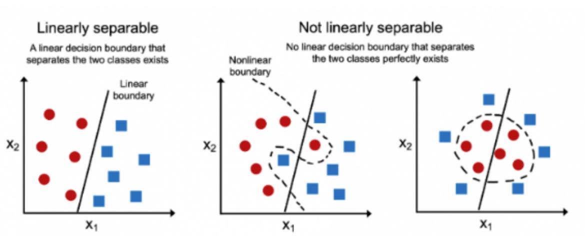


Figure 5: Linearly Separable and Non Linear Separable Hyperplane .

3.5 Finding a Max-Margin Classifier Using SVM Objectives

3.5.1 How to Find a Max-Margin Classifier?

- The goal is to **find a decision boundary (hyperplane) that maximizes the margin** between two classes.
- This problem is solved using **Support Vector Machines (SVMs)**.

3.5.2 Case 1: Linearly Separable Dataset

- Assume the dataset D **is linearly separable**.
- We consider a **3D case (**$d = 3$**)** for visualization.
- The data points from two different classes are **separated by a hyperplane**.

3.5.3 Understanding the Hyperplane

A **hyperplane** is defined as:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{11}$$

where:

- $\mathbf{w}$ is the **normal vector** to the hyperplane.
- $\mathbf{x}$ is a **data point**.
- b is the **bias term**.

The **hyperplane linearly separates** the dataset into two classes.

3.5.4 Max-Margin Concept in SVM

- **SVM finds the hyperplane that maximizes the margin** (distance between the nearest positive and negative points).
- The **margin** is the distance d and d' in the visualization.

3.5.5 Classification Conditions

The classification rule based on the hyperplane equation:

- **For positive class** ($y_n = +1$):
$$\mathbf{w}^T\mathbf{x}_n + b > 0 \tag{12}$$
- **For negative class** ($y_n = -1$):
$$\mathbf{w}^T\mathbf{x}_n + b < 0 \tag{13}$$
- This ensures that all **positive points lie above the hyperplane** and **negative points lie below the hyperplane**.

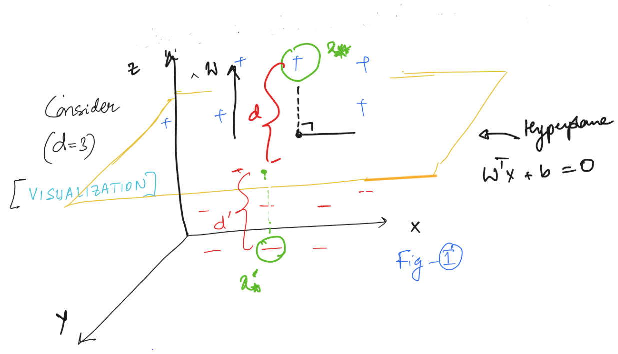


Figure 6: Hyperplane in 3D space .

3.6 Key Takeaways

1. **SVM finds the optimal hyperplane that maximizes the margin.**
2. The **hyperplane equation** is given by:
$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{14}$$
3. **Points on either side of the hyperplane satisfy the conditions:**
   - $\mathbf{w}^T\mathbf{x}_n + b > 0$ for $y_n = +1$.
   - $\mathbf{w}^T\mathbf{x}_n + b < 0$ for $y_n = -1$.
4. **Linearly separable data** can be classified using a **linear kernel SVM**.
5. **Non-linearly separable data** requires **kernel tricks** to transform data into a higher-dimensional space.
6. **SVM with a maximum margin** ensures **better generalization to unseen data**.

4 Optimization Objective for SVM

4.1 Finding the Distance of a Point from the Hyperplane

The distance of a point x_* from the hyperplane $\mathbf{w}^T\mathbf{x} = 0$ is given by:

$$d = \frac{|(\mathbf{x}_* - \mathbf{x})^T\mathbf{w}|}{|\mathbf{w}|} \tag{15}$$

This formula is derived using the **projection** of the vector $(\mathbf{x}_* - \mathbf{x})$ onto $\mathbf{w}$.

4.2 Objective: Maximizing the Margin

The goal is to find $\mathbf{w}$ that maximizes the margin d. This translates to the following optimization problem:

$$\max_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \quad \frac{|\mathbf{w}^T(\mathbf{x}_* - \mathbf{x})|}{|\mathbf{w}|} \tag{16}$$

subject to:

$$|\mathbf{w}^T x_* + b| = 1 \tag{17}$$

The constraint ensures that the distance of **support vectors** from the hyperplane is 1.

4.3 Reformulating the Optimization Problem

Since $\mathbf{w}^T x + b = 0$ defines the hyperplane, we can simplify:

$$|\mathbf{w}^T (\mathbf{x}* - \mathbf{x})| = |\mathbf{w}^T x * + b| \tag{18}$$

and from our scaling assumption:

$$|\mathbf{w}^T x_* + b| = 1 \tag{19}$$

Thus, the final optimization problem simplifies to:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2 \tag{20}$$

subject to:

$$y_i(\mathbf{w}^T x_i + b) \geq 1, \quad \forall i. \tag{21}$$

4.4 Optimal Choice of w in SVM

The final SVM optimization problem is given by:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2 \tag{22}$$

subject to:

$$y_n(\mathbf{w}^T x_n + b) \geq 1, \quad \forall n = 1, \ldots, N \tag{23}$$

At the **optimal solution** w, at least **one constraint must be active** for some n, meaning:

$$y_n(\mathbf{w}^T x_n + b) = 1. \tag{24}$$

4.5 Justification: Why Must at Least One Constraint Be Active?

If all constraints were strictly greater than 1, i.e.,

$$y_n(\mathbf{w}^T x_n + b) > 1, \quad \forall n \tag{25}$$

then we could rescale $\mathbf{w}$ and b by a small factor (say, dividing them by some constant $\alpha > 1$) while still satisfying all constraints. This would **decrease** $|\mathbf{w}|^2$, contradicting the fact that we found the **optimal solution**. Therefore, at least one data point must **lie exactly on the margin**, meaning:

$$y_n(\mathbf{w}^T x_n + b) = 1. \tag{26}$$

These points that satisfy the equality constraint are called **support vectors** because they determine the **optimal margin**.

4.6 Key Takeaways

- The SVM optimization problem is formulated as a quadratic minimization problem.
- The constraint ensures that all points are classified correctly while maximizing the margin.
- Support vectors lie exactly on the margin and play a critical role in defining the decision boundary.
- The final SVM objective ensures a balance between margin maximization and correct classification.
- This results in a convex optimization problem, which can be solved using Lagrange multipliers.

5 Hard Margin SVM and Its Solution

The **Hard Margin SVM** assumes that the given dataset is perfectly separable by a hyperplane, meaning there exists a decision boundary where all positive and negative samples can be classified without misclassification. The optimization problem is formulated as:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \quad \frac{1}{2}|\mathbf{w}|^2 \tag{27}$$

subject to:

$$y_n(\mathbf{w}^T x_n + b) \geq 1, \quad \forall n = 1, 2, \ldots, N \tag{28}$$

where is minimized to achieve a **maximum margin hyperplane**, and the constraint ensures all training points are correctly classified under the assumption of perfect separability. To solve this constrained optimization problem, we use **Lagrange multipliers** and the **Karush-Kuhn-Tucker (KKT) conditions**

Next Lecture

The next lecture will cover the following topics:
(i) KKT condition and strong duality to solve hard-margin SVMs
(ii) Support vector points for hard-margin SVM
(iii) Non-Linear Separable data -Kernel methods.

References:

- Avanti, A. (Stanford University). *Lecture Notes on Machine Learning: Support Vector Machines*. Stanford Machine Learning Course.
  A concise and structured introduction to Support Vector Machines (SVM) with practical insights and applications.
- Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.
  This book provides the theoretical foundation for Support Vector Machines (SVM) and statistical learning.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
  A practical introduction to SVMs, covering both linear and non-linear classification methods.

- Burges, C. J. C. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2), 121–167.
  A detailed tutorial explaining the mathematics and implementation of SVMs.

- Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
  A foundational text on convex optimization, including Lagrange multipliers and KKT conditions, which are used in solving SVM optimization problems.