Support Vector Machine (SVM)

Nguyen Minh Bao 23520123
Nguyen Minh Triet 23521652

Math for Computer Science

January 11, 2025

Table of Contents

- **Section 1: Introduction**

Table of Contents

- **Section 1: Introduction**
- **Section 2: Hard margin SVM**

Table of Contents

- **Section 1: Introduction**
- **Section 2: Hard margin SVM**
- **Section 3: Soft Margin and Kernel Function**

Table of Contents

- **Section 1: Introduction**
- **Section 2: Hard margin SVM**
- **Section 3: Soft Margin and Kernel Function**
- **Section 4: Advantages and Drawbacks**

- **Section 1: Introduction**

Support Vector Machine (SVM) - Definition

- A supervised learning algorithm primarily used for classification tasks.
- The objective is to find the optimal hyperplane that separates data points of different classes.
- Based on the characteristics of the dataset, there are different variations of SVM, such as: Linear SVM, Non-linear SVM,...

SVMs are versatile tools widely used in various fields, its key applications include:

1. Text and Document Classification

- **Spam Detection:** Classify emails as spam or non-spam.
- **Sentiment Analysis:** Analyze sentiment in texts (e.g., positive, neutral, or negative).
- **Topic Categorization:** Categorize documents into predefined topics.

2. Image Processing

- **Object Recognition:** Identify objects or patterns in images.
- **Face Detection:** Separate face and non-face regions in images.
- **Handwriting Recognition:** Classify handwritten characters or digits.

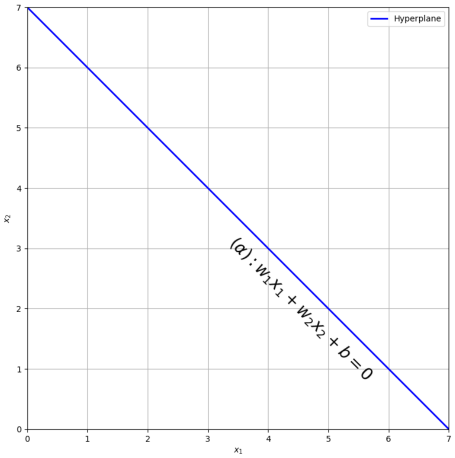
In a n-dimensional space, a hyperplane (α) is defined as a subspace with a dimension of n - 1, represented by the equation:

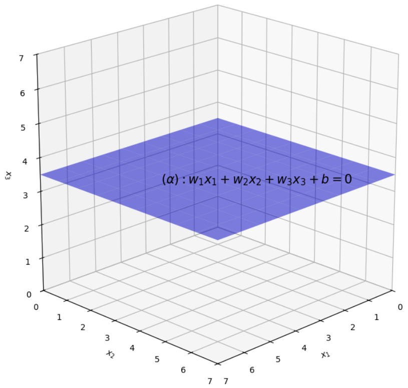
$$w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \mathbf{w}^\top \mathbf{x} + b = 0,$$

Where:

- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$: coordinates of a point on the hyperplane.
- $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T$: a normal vector of (α).
- b: a scalar constant, also called the **bias**.

Hyperplane Visualization in $\mathbb{R}^2$ and $\mathbb{R}^3$





Key Concept:

A hyperplane divides its space into two parts: the **Positive** and **Negative** sides.

Conditions:

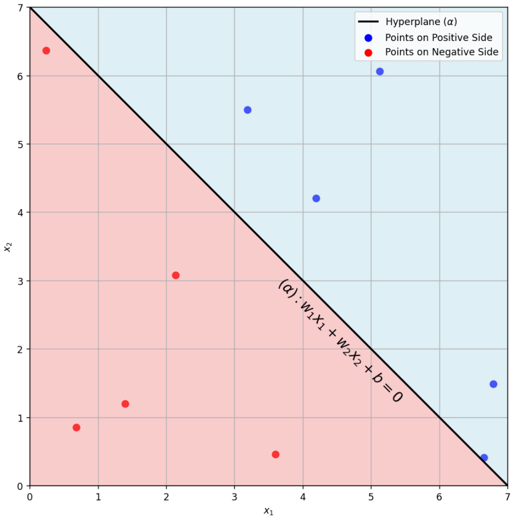
- A point $\mathbf{x}_0$ belongs to the **Positive side** if:

$$\mathbf{w}^\top \mathbf{x}_0 + b > 0$$

- A point $\mathbf{x}_0$ belongs to the **Negative side** if:

$$\mathbf{w}^\top \mathbf{x}_0 + b < 0$$

Positive and Negative Sides Visualization



Distance from a Point to a Hyperplane

In a n-dimensional space, the distance d from a point $\mathbf{x}_0 = [x_{01}, x_{02}, \ldots, x_{0n}]^T$ to the hyperplane α is defined by the equation:

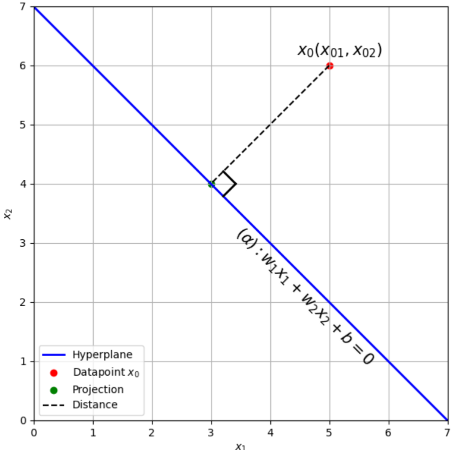
$$d = \frac{|w_1 x_{01} + w_2 x_{02} + \cdots + w_n x_{0n} + b|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} = \frac{|\mathbf{w}^T \mathbf{x}_0 + b|}{\|\mathbf{w}\|_2}$$

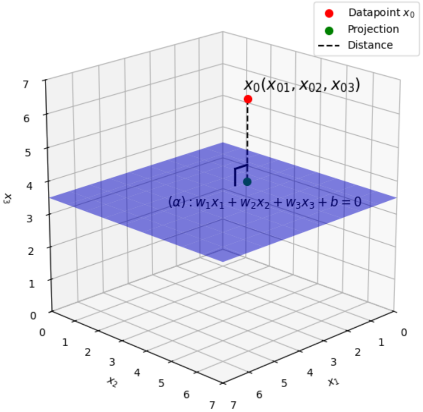
where:

$$\|\mathbf{w}\|_2 = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2} = \sqrt{\mathbf{w}^T \mathbf{w}}$$

is the ℓ_2-norm of $\mathbf{w}$.

Distance Visualization in $\mathbb{R}^2$ and $\mathbb{R}^3$

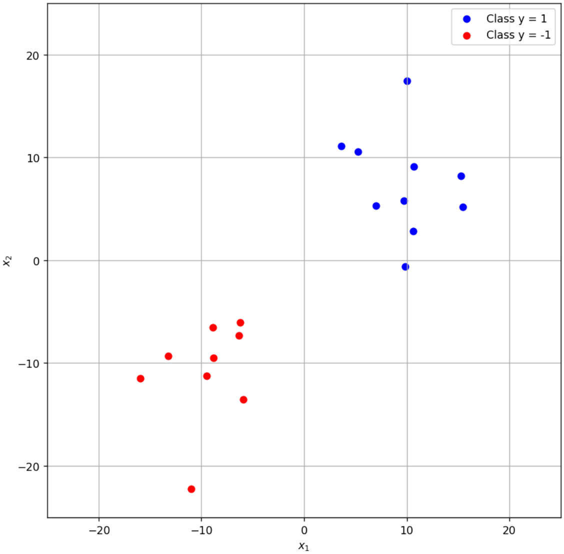




- **Section 2: Hard margin SVM**

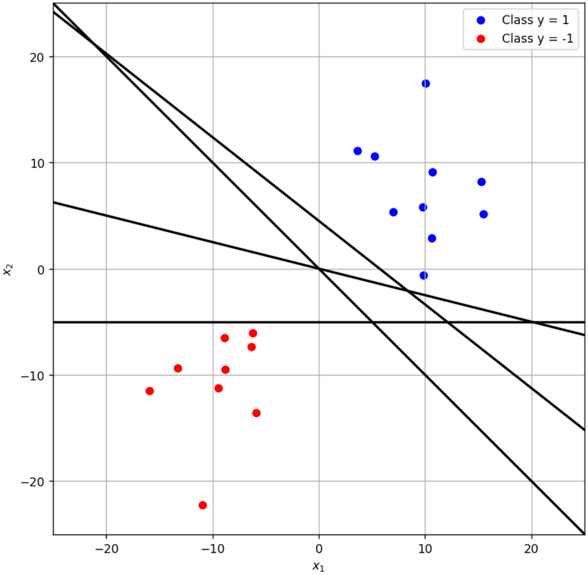
Given:

- A dataset $D = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$, where:
  - $\mathbf{x}^{(i)} \in \mathbb{R}^n$ (feature vectors),
  - $y^{(i)} \in \{-1, 1\}$ (class labels), for $i = 1, \ldots, m$.
- Assume the two classes of data points ($y^{(i)} = 1$ and $y^{(i)} = -1$) are **linearly separable**.



Find the **best hyperplane** to separate these two classes.

Problem Statement

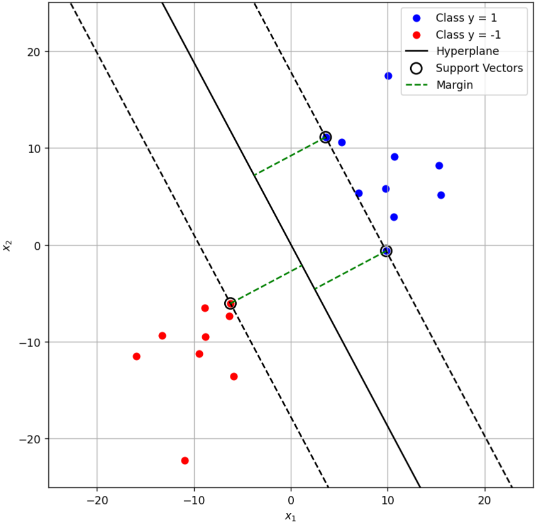


How can we find the **best hyperplane** to separate these two classes?

Definition:

- **Hard margin SVM** is designed specifically for classification tasks where the 2 classes of data is **linearly separable**.
- The goal of **Hard margin SVM** is to find the **optimal hyperplane** that **maximize the margin**, which is the distance between the hyperplane and the nearest data points (called **support vectors**) from both classes.

Margin visualization



Definition:

- In the n-dimensional space, the separating hyperplane (α) has the form:

$$w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \mathbf{w}^\top \mathbf{x} + b = 0,$$

Conditions:

Since the hyperplane separates the two classes of data points, for all data pairs $(\mathbf{x}^{(i)}, y^{(i)}) \in D$, the following conditions must hold:

$$\begin{cases} \mathbf{w}^\top \mathbf{x}^{(i)} + b \geq 0, & \text{if } y^{(i)} = 1, \\ \mathbf{w}^\top \mathbf{x}^{(i)} + b < 0, & \text{if } y^{(i)} = -1. \end{cases}$$

Alternatively, these conditions can be written as:

$$y^{(i)} \left(\mathbf{w}^\top \mathbf{x}^{(i)} + b \right) = \left| \mathbf{w}^\top \mathbf{x}^{(i)} + b \right| \geq 0. \quad \textbf{(C1)}$$

Distance to the Hyperplane

- The distance d from each data point $(\mathbf{x}^{(i)}, y^{(i)})$ to the hyperplane α is given by:

$$d = \frac{\left|w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots + w_n x_n^{(i)} + b\right|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} = \frac{\left|\mathbf{w}^\top \mathbf{x}^{(i)} + b\right|}{\|\mathbf{w}\|_2}.$$

- Using (**C1**), the distance can also be written as:

$$d = \frac{y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}{\|\mathbf{w}\|_2}.$$

- If $(\mathbf{x}^{(a)}, y^{(a)})$ is a **support vector**, it satisfies:

$$y^{(a)}(\mathbf{w}^\top \mathbf{x}^{(a)} + b) = c, \quad c \in \mathbb{R}^+.$$

- The set of coefficients $(\mathbf{w}, b)$ for a hyperplane α is not unique. Scaling them by any positive constant $k \in \mathbb{R}^+$ still represents the same hyperplane.
- By choosing $k = \frac{1}{c}$, we can assume:

$$y^{(a)}(\mathbf{w}^\top \mathbf{x}^{(a)} + b) = 1, \quad \textbf{(Eq.1)}$$

without affecting the relative geometry of the problem.

Margin Size:

- Using (**Eq.1**) the margin size is calculated as:

$$\text{margin} = \frac{y^{(a)}(\mathbf{w}^\top \mathbf{x}^{(a)} + b)}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2}.$$

- Since $(\mathbf{x}^{(a)}, y^{(a)})$ is a **support vector**, we can conclude that for every $i = 1, \ldots, m$, the following holds:

$$\frac{y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}{\|\mathbf{w}\|_2} \geq \frac{1}{\|\mathbf{w}\|_2}.$$

- The term $\|\mathbf{w}\|_2$ represents a positive scalar, allowing us to rewrite the inequality as:

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 \qquad \textbf{(C2)}$$

Objective:

- Our goal is to **maximize the margin size**, which is equivalent to solving for the pair of optimal values $(\mathbf{w}^*, b^*)$ of the following optimization problem:

$$(\mathbf{w}^*, b^*) = \arg\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2},$$

subject to:

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, m.$$

Reformulated Problem:

- The above problem is equivalent to minimizing the squared norm of $\mathbf{w}$:

$$(\mathbf{w}^*, b^*) = \arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2,$$

subject to: **(P1)**

$$1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \leq 0, \quad \forall i = 1, \ldots, m.$$

Statement:
(**P1**) satisfies **Slater's condition** if there exists a pair $(\mathbf{w}, b)$ that is **strictly feasible**.

Strict Feasibility Condition:
In (**P1**), a pair $(\mathbf{w}, b)$ is **strictly feasible** if:

$$1 - y^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) < 0, \quad \forall i = 1, 2, \ldots, m.$$

Therefore, to prove that (**P1**) satisfies **Slater's condition**, we need to demonstrate that there exists $(\mathbf{w}, b)$ such that:

$$1 - y^{(i)}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b\right) < 0, \quad \forall i = 1, 2, \ldots, m.$$

Proving that (P1) Satisfies Slater's Condition

D is **linearly separable**, there exists a pair $(\mathbf{w}_0, b_0)$ such that:

$$1 - y^{(i)}\left(\mathbf{w}_0^\top \mathbf{x}^{(i)} + b_0\right) \leq 0, \quad \forall i = 1, \ldots, m.$$

$$\Leftrightarrow 1 - y^{(i)}\left(2\mathbf{w}_0^\top \mathbf{x}^{(i)} + 2b_0\right) \leq -1 < 0, \quad \forall i = 1, \ldots, m.$$

$(\mathbf{w}_1, b_1) = 2(\mathbf{w}_0, b_0)$, then:

$$1 - y^{(i)}\left(\mathbf{w}_1^\top \mathbf{x}^{(i)} + b_1\right) < 0, \quad \forall i = 1, \ldots, m.$$

Conclusion:
In (**P1**), there always exists a **strictly feasible** pair $(\mathbf{w}_1, b_1)$. Therefore, (**P1**) satisfies **Slater's condition**.

Lagrangian:

- The Lagrangian for the optimization problem (**P1**) is defined as:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m} \lambda_i \left(1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\right),$$

  where:

  - $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]^\top$ is the Lagrange multipliers vector.
  - $\lambda_i \geq 0, \quad \forall i = 1, \ldots, m.$

KKT Conditions:

$$1 - y^{(i)}\left((\mathbf{w}^*)^\top \mathbf{x}^{(i)} + b^*\right) \leq 0, \quad \forall i = 1, \ldots, m. \quad \textbf{(C2.1)}$$

$$\lambda_i^* \geq 0, \quad \forall i = 1, \ldots, m. \quad \textbf{(C2.2)}$$

$$\lambda_i^*\left(1 - y^{(i)}\left((\mathbf{w}^*)^\top \mathbf{x}^{(i)} + b^*\right)\right) = 0, \quad \forall i = 1, \ldots, m. \quad \textbf{(C2.3)}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^*} = \mathbf{w}^* - \sum_{i=1}^{m} \lambda_i^* y^{(i)} \mathbf{x}^{(i)} = 0. \quad \textbf{(C2.4)}$$

$$\frac{\partial \mathcal{L}}{\partial b^*} = \sum_{i=1}^{m} \lambda_i^* y^{(i)} = 0. \quad \textbf{(C2.5)}$$

Note: $\boldsymbol{\lambda}^*$ represents the **optimal solution** for the **dual problem** of (**P1**).

Motivation:

- Directly solving for $\mathbf{w}^*, b^*, \boldsymbol{\lambda}^*$ using the KKT conditions can be computationally intensive.
- Instead, solving for $\boldsymbol{\lambda}$ in the Lagrange dual problem of $(\mathbf{P1})$ is more efficient and commonly done.

Lagrange Dual Function:

- The dual function $g(\boldsymbol{\lambda})$ is defined as:

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}),$$

where the Lagrangian $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda})$ is given by:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m} \lambda_i \left(1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\right).$$

Why Solving for the Dual Problem is More Efficient

- For any value of $\boldsymbol{\lambda} \geq 0$, the dual function $g(\boldsymbol{\lambda})$ provides a **lower bound** for the **optimal value** of the **primal problem**.
- To obtain the best lower bound, we maximize the dual function subject to constraints on $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}),$$

  subject to:

$$\lambda_i \geq 0, \quad \forall i = 1, \ldots, m.$$

- This is the **dual problem** of the primal problem, and it is always a **convex optimization problem**.
- When **strong duality** holds, $g(\boldsymbol{\lambda}^*)$ equals the **optimal value of the primal problem**.

An Example Where Strong Duality Holds

Example: In $\mathbb{R}$, consider the following optimization problem:

$$x^* = \arg\max_x \left(0.5x^2 - 5x + 7\sin(x) + 10\right),$$

subject to:

$$(x-2)^2 - 4 \leq 0.$$

Lagrangian: The Lagrangian for this problem is:

$$\mathcal{L}(x, \lambda) = 0.5x^2 - 5x + 7\sin(x) + 10 + \lambda\left((x-2)^2 - 4\right),$$

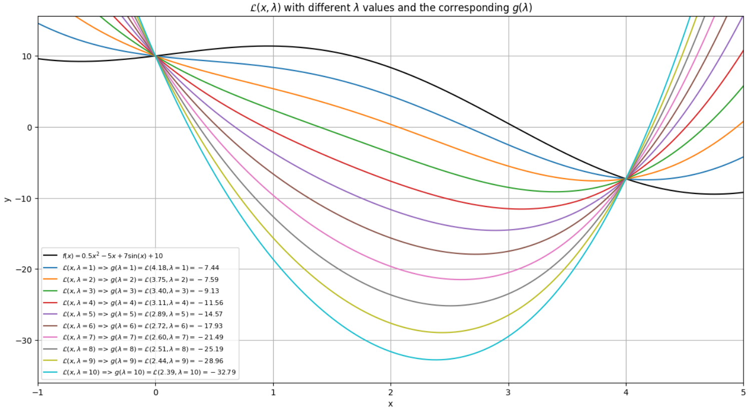
where $\lambda \geq 0$ is the Lagrange multiplier.
Dual Problem: The dual problem is defined as:

$$\lambda^* = \arg\max_\lambda(\inf_x \mathcal{L}(x, \lambda))$$

subject to:

$$\lambda \geq 0.$$

$\mathcal{L}(x, \lambda)$ with Different λ and the Corresponding $g(\lambda)$



Key Steps:

- To find $\inf_{\mathbf{w},b} \mathcal{L}$, set the partial derivatives of $\mathcal{L}$ with respect to $\mathbf{w}$ and b to zero.

Partial Derivatives:

- With respect to $\mathbf{w}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \lambda_i y^{(i)} \mathbf{x}^{(i)} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \lambda_i y^{(i)} \mathbf{x}^{(i)}. \quad \textbf{(Eq.2)}$$

- With respect to b:

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{m} \lambda_i y^{(i)} = 0. \quad \textbf{(Eq.3)}$$

Substituting (Eq.2) and (Eq.3) into $g(\boldsymbol{\lambda})$:

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}. \quad \textbf{(Eq.4)}$$

Dual Problem Formulation:

- By combining (**Eq.3**), (**Eq.4**), and the constraints on $\boldsymbol{\lambda}$, we obtain the Lagrange dual problem of (**P1**):

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}),$$

subject to: **(P2)**

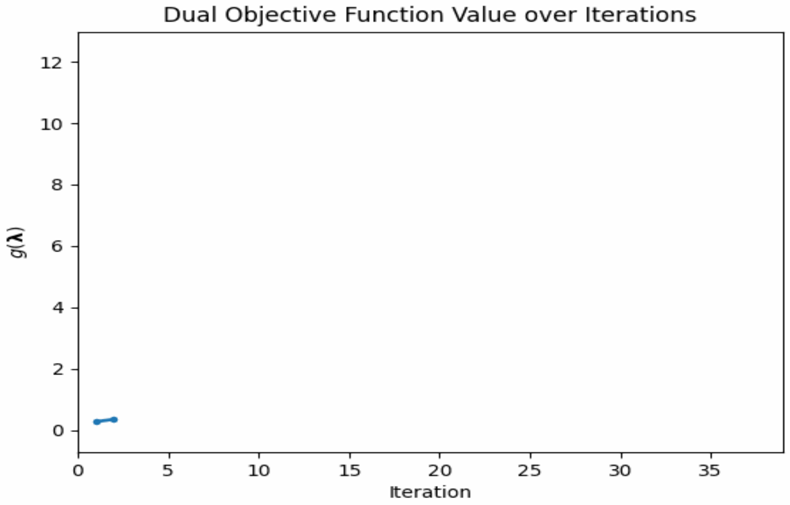
$$\lambda_i \geq 0, \quad \forall i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} \lambda_i y^{(i)} = 0.$$

Solving the Dual Problem:

- **(P2)** is a **quadratic programming problem**.
- To solve it, we can use:
  - **Sequential Minimal Optimization** (SMO),
  - Libraries such as **CVXOPT**, **sklearn**

$g(\lambda)$ over iterations



Calculating $\mathbf{w}^*$ and b^* from KKT Conditions

Observation:

- From **(C2.3)**:

$$\lambda_i^* \left(1 - y^{(i)} \left((\mathbf{w}^*)^\top \mathbf{x}^{(i)} + b^*\right)\right) = 0, \quad \forall i = 1, \ldots, m.$$

- $\lambda_i^* > 0$ only if:

$$y^{(i)} \left((\mathbf{w}^*)^\top \mathbf{x}^{(i)} + b^*\right) = 1,$$

  meaning $\mathbf{x}^{(i)}$ is a **support vector**.
- Define the set of **support vectors** as:

$$S = \{i \mid \lambda_i^* \neq 0\}.$$

- Calculate $\mathbf{w}^*$ using **(C2.4)**:

$$\mathbf{w}^* = \sum_{i=1}^{m} \lambda_i^* y^{(i)} \mathbf{x}^{(i)} = \sum_{i \in S} \lambda_i^* y^{(i)} \mathbf{x}^{(i)}.$$

Step 3: Calculate b^*:

- Since $\mathbf{x}^{(i)}$ is a **support vector** for every $i \in S$, we have:

$$y^{(i)}\left((\mathbf{w}^*)^\top \mathbf{x}^{(i)} + b^*\right) = 1.$$

- For each $i \in S$, we can calculate:

$$b^* = \frac{1}{y^{(i)}} - (\mathbf{w}^*)^\top \mathbf{x}^{(i)}.$$

- Alternatively, for numerical stability, we can calculation b^* by taking the mean of all possible b^* values:

$$b^* = \frac{1}{|S|}\sum_{i \in S}\left(y^{(i)} - (\mathbf{w}^*)^\top \mathbf{x}^{(i)}\right).$$

Separating Hyperplane:

- The separating hyperplane α is defined as:

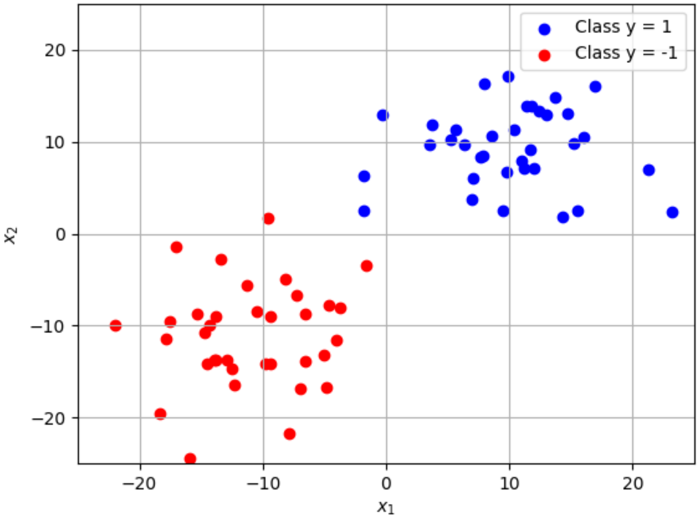
$$\alpha : (\mathbf{w}^*)^\top \mathbf{x} + b^* = \sum_{i \in S} \lambda_i^* y^{(i)} \mathbf{x}^{(i)} + \frac{1}{|S|} \sum_{i \in S} \left(y^{(i)} - (\mathbf{w}^*)^\top \mathbf{x}^{(i)} \right) = 0.$$

Prediction for a New Data Point $\mathbf{x}^{(n)}$:

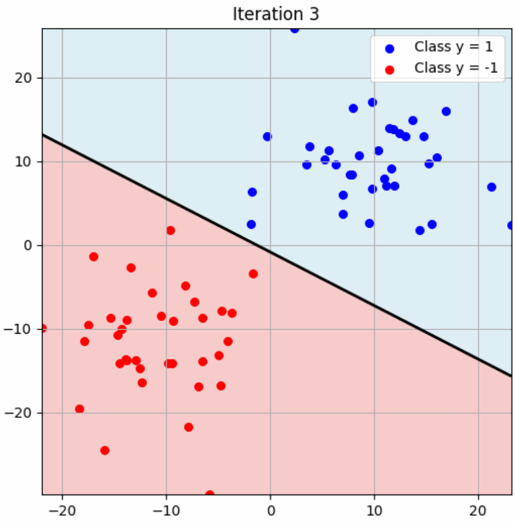
- The label $y^{(n)}$ for a new data point $\mathbf{x}^{(n)}$ is determined as follows:

$$y^{(n)} = \begin{cases} 1, & \text{if } (\mathbf{w}^*)^\top \mathbf{x}^{(n)} + b^* \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

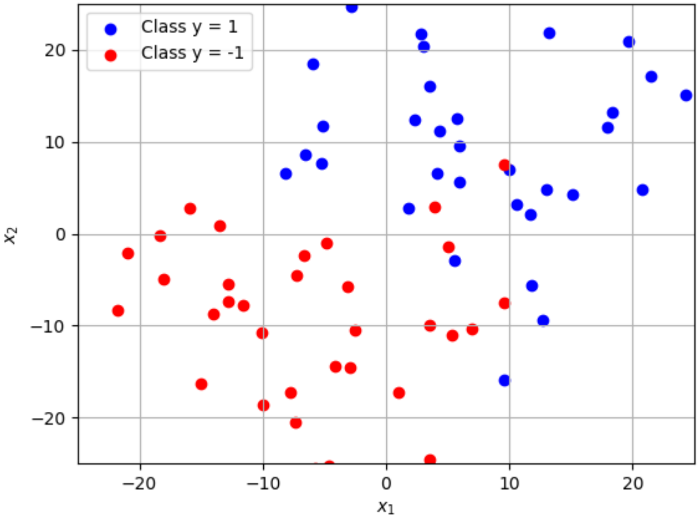
Dataset visualization



Hyperplane over iterations



Dataset visualization

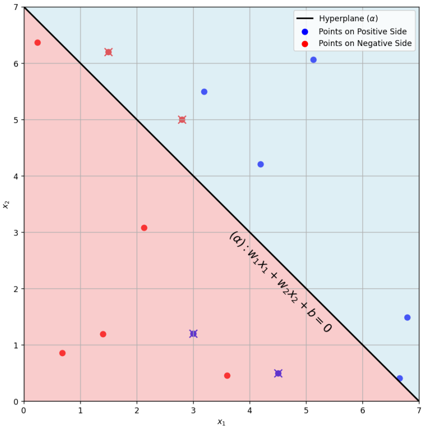


- **Section 3: Soft Margin and Kernel Function**

- **Soft Margin**
- **Kernel Function**
  - Linear Kernel
  - Polynomial Kernel
  - Radial Basis Function (RBF) Kernel
  - Sigmoid Kernel

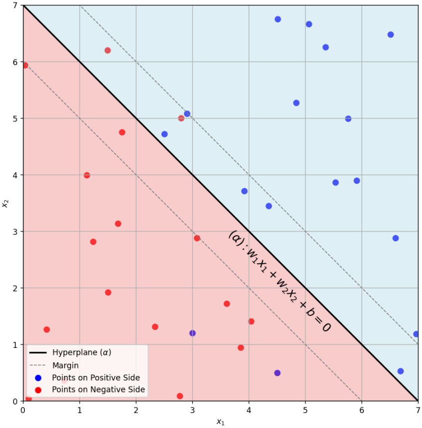
Hard Margin

- **Assumption**: The data is perfectly linearly separable, meaning there exists a hyperplane that can separate the two classes without any misclassification.
- **Goal**: Maximize the margin between classes with no points inside the margin.
- **Conditions**:
  - No points allowed within the margin.
  - No misclassifications.



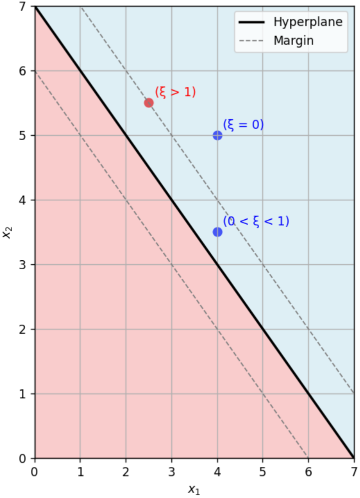
Soft Margin

- **Problem**: In real-world scenarios, data is often noisy and not perfectly linearly separable. So we can not find w and b. Therefore, a model that allows for some misclassification is needed to handle these cases.



- **Solution**: To address the problem of non-separable data, we use slack variables ξ_i for each data point.
- **Role of** ξ:
  - ξ_i measures the degree of misclassification for each data point.
  - $\xi_i = 0$: The point is correctly classified and outside the margin.
  - $0 < \xi_i < 1$: The point is lying between hyperplane and margin.
  - $\xi_i > 1$: The point is misclassified.
- **Constraints**:

$$\begin{cases} \mathbf{w}^\top \mathbf{x}^{(i)} + b \geq 1 - \xi_i, & \text{if } y^{(i)} = 1, \\ \mathbf{w}^\top \mathbf{x}^{(i)} + b \leq -1 + \xi_i, & \text{if} y^{(i)} = -1, \\ \xi_i \geq 0, & \forall i = 1, 2, \ldots, m. \end{cases}$$



Soft Margin Optimization Problem

- **Optimization Objective:**
  - Find:

$$(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*) = \arg\min_{\mathbf{w}, b, \boldsymbol{\xi}} \left(\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\xi_i\right) \quad \textbf{(P3)}$$

  Subject to:
    - $1 - \xi_i - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \leq 0, \quad \forall i = 1, 2, \ldots, m$
    - $-\xi_i \leq 0, \quad \forall i = 1, 2, \ldots, m$
- **Role of C:**
  - C balances margin size and misclassification penalty.
  - **Small** C:The model allows some data points to be wrongly classified to maximize the distance between the two layers (larger margins).
  - **Large** C:The model will try to accurately classify all data points and accept a narrower margin.
- **Slater's Condition:**
  - For all $i = 1, 2, \ldots, m$ and $(\mathbf{w}, b)$, **there always exist** positive numbers ξ_i, $i = 1, 2, \ldots, m$, large enough such that:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) + \xi_i > 1, \quad \forall i = 1, 2, \ldots, m.$$

- **Dual Function:**

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \textbf{(Eq3.1)}$$

- **The Lagrange function is:**

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\lambda_n\big(1-\xi_i - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)\big) - \sum_{i=1}^{m}\mu_i\xi_i \quad \textbf{(Eq3.2)}$$

where:

- $\lambda_i \geq 0$ and $\mu_i \geq 0$ are Lagrange multipliers.

- For each pair (λ, μ), we find $(\mathbf{w}, b, \boldsymbol{\xi})$ that satisfies the derivative conditions:

$$\nabla_w \mathcal{L} = 0 \Rightarrow w = \sum_{i=1}^{m} \lambda_i y^{(i)} x^{(i)} \qquad \textbf{(Eq3.3)}$$

$$\nabla_b \mathcal{L} = 0 \Rightarrow \sum_{i=1}^{m} \lambda_i y^{(i)} = 0 \qquad \textbf{(Eq3.4)}$$

$$\nabla_\xi \mathcal{L} = 0 \Rightarrow \lambda_i = C - \mu_i \qquad \textbf{(Eq3.5)}$$

- **Insights from (Eq 3.5):**
  - We only need to consider pairs (λ, μ) such that $\lambda_i = C - \mu_i$.
  - This implies $0 \leq \lambda_i, \mu_i \leq C, \quad \forall i = 1, 2, \ldots, m$.
- **Lagrange Dual Function:**
  After substituting w, ξ, and λ into the Lagrange function, we get:

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}. \quad \textbf{(Eq3.6)}$$

- **Subject to the following constraints:**

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda})$$

subject to:

- $0 < \lambda^{(i)} \leq C, \quad \forall i = 1, \ldots, N$
- $\sum_{i=1}^{m} \lambda^{(i)} y^{(i)} = 0$

- **KKT Conditions for Soft Margin:**

$$\xi_i \geq 0, \quad \lambda_i{}^* \geq 0, \quad \mu_i \geq 0 \quad \mu_i \xi_i = 0 \quad \textbf{(C3.1, C3.2, C3.3, C3.4)}$$

$$y^{(i)}\left((\mathbf{w}^*)^T \cdot \mathbf{x}^{(i)} + b^*\right) \geq 1 - \xi_i, \quad \forall i = 1, \ldots, N \qquad \textbf{(C3.5)}$$

$$\lambda_i(y^{(i)}((\mathbf{w}^* \cdot \mathbf{x}^{(i)}) + b^*) - 1 + \xi_i) = 0 \qquad \textbf{(C3.6)}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^*} = \mathbf{w}^* - \sum_{i=1}^{m} \lambda_i^* y^{(i)} \mathbf{x}^{(i)} = 0. \quad \textbf{(C3.7)}$$

$$\frac{\partial \mathcal{L}}{\partial b^*} = \sum_{i=1}^{m} \lambda_i^* y^{(i)} = 0. \quad \textbf{(C3.8)}$$

$$\lambda_i = C - \mu_i \quad \textbf{(C3.9)}$$

- **If** $\lambda_i^* > 0$**: (C3.7)**: Contributes to finding the solution $\mathbf{w}$ in the soft margin SVM problem.

$$\mathbf{w}^* = \sum_{n \in \mathcal{S}} \lambda_n^* y^{(n)} \mathbf{x}^{(n)}$$

$\mathcal{S} = \{n : 0 < \lambda_n\}$, the support vectors between or on the boundaries.

- **If** $0 < \lambda_i^* < C$ and **(C3.9)**,**(C3.4)**,**(C3.6)**: Indicates that these points lie exactly on the margin boundary.

$$y^{(n)}((\mathbf{w}^*)^T \mathbf{x}^{(n)} + b) = 1 \quad \forall n \in \mathcal{M}$$

$$b^* = \frac{1}{N_{\mathcal{M}}} \sum_{m \in \mathcal{M}} \left(y^{(m)} - (\mathbf{w}^*)^T \mathbf{x}^{(m)} \right)$$

$\mathcal{M} = \{n : 0 < \lambda_n < C\}$, the set of points on the margin boundary.

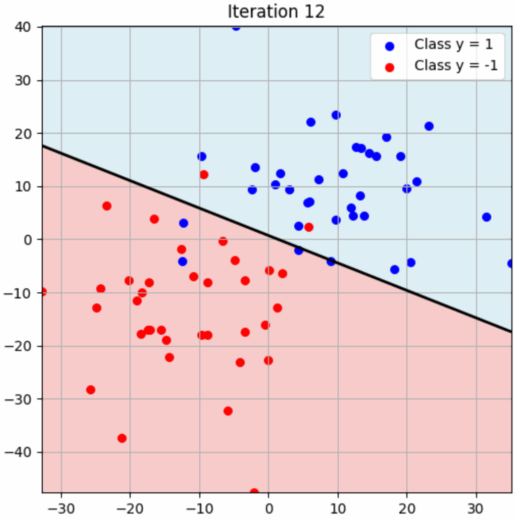
Final Decision Function:

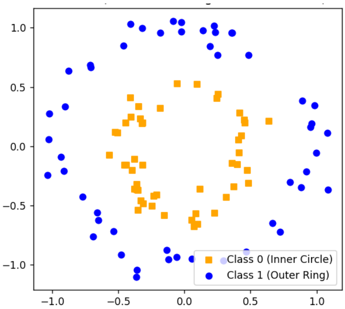
$$(\mathbf{w}^*)^T \mathbf{x} + b^* = \sum_{m \in \mathcal{S}} \lambda_m^* y^{(m)} \mathbf{x}^{(m)T} \mathbf{x} + \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(y^{(n)} - \sum_{m \in \mathcal{S}} \lambda_m^* y^{(m)} \mathbf{x}^{(m)^T} \mathbf{x}^{(n)} \right) \quad \textbf{(Eq3.8)}$$

C	Accuracy	Margin Width
0.0001	0.750000	4.232477
0.0010	0.953810	1.779354
0.0100	0.978095	1.093401
0.1000	0.983810	0.731362
1.0000	0.986190	0.599481
10.0000	0.986190	0.588683
100.0000	0.986190	0.587925

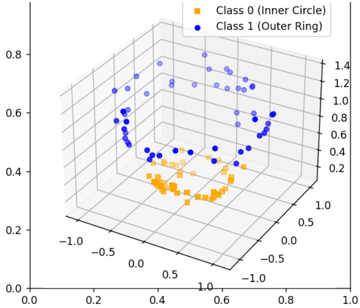
Table: Impact of C on Accuracy and Margin Width for Soft Margin SVM

Hyperplane over iterations





$$\phi : \mathbf{x} \rightarrow \phi(\mathbf{x})$$



Basic concept: It is always possible to transform the initial feature space into a higher-dimensional feature space in which the training set exhibits separability.

- **Based on the problem linear SVM:**

$$\lambda^* = \arg\max_{\lambda}\left(\sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y^{(i)}y^{(j)}\mathbf{x}^{(i)\top}\mathbf{x}^{(j)}.\right) \qquad \textbf{(Eq3.9)}$$

subject to:

$$\sum_{i=1}^{m}\lambda_i y^{(i)} = 0, \quad 0 \le \lambda_i \le C, \quad \forall i$$

- **After finding λ for problem (Eq3.9)**: the label of a new data point will be determined by

$$\text{class}(x) = \text{sgn}\left(\sum_{m\in S}\lambda_m y^{(n)}\mathbf{x}^{(m)\,T}\mathbf{x} + \frac{1}{N_{\mathcal{M}}}\sum_{n\in\mathcal{M}}\left(y^{(n)} - \sum_{m\in S}\lambda_m y^{(m)}\mathbf{x}^{(m)\,T}\mathbf{x}^{(n)}\right)\right) \qquad \textbf{(Eq3.10)}$$

Kernel Trick

- **Assume** that we can find a function $\Phi(\cdot)$ such that the data points $\Phi(x)$ are (approximately) linearly separable in the new space.

$$\lambda^* = \arg\max_{\lambda} \left(\sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} \Phi(\mathbf{x}^{(i)})^T \Phi(\mathbf{x}^{(j)}) \right) \quad \textbf{(Eq3.9.1)}$$

- **By defining the kernel function** $\mathbf{k}(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z})$, we can rewrite problem **(Eq3.9)** as follows:

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} \left(\sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \right) \quad \textbf{(Eq3.9.2)}$$

- **Subject to**:

$$\sum_{i=1}^{m} \lambda_i y^{(i)} = 0, \quad 0 \leq \lambda_i \leq C, \quad \forall i = 1, 2, \ldots, m.$$

- **Rewrite the hyperplane equation:**

$$(\mathbf{w}^*)^T\mathbf{x} + b^* = \sum_{m\in S}\lambda_m y^{(m)}k(\mathbf{x}^{(m)},\mathbf{x}) + \frac{1}{N_{\mathcal{M}}}\sum_{n\in\mathcal{M}}\left(y^{(n)} - \sum_{m\in S}\lambda_m y^{(m)}k(\mathbf{x}^{(m)},\mathbf{x}^{(n)})\right) \qquad \textbf{(Eq3.10)}$$

- **Example:** Consider a transformation of a point in a two-dimensional space $\mathbf{x} = [x_1, x_2]^T$ into a five-dimensional space as:

$$\Phi(\mathbf{x}) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2\right]^T$$

- **Compute two transformed points $\Phi(\mathbf{x})$ and $\Phi(\mathbf{z})$:**

$$\Phi(\mathbf{x})^T\Phi(\mathbf{z}) = 1 + 2x_1z_1 + 2x_2z_2 + x_1^2x_2^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2$$

- **Finally, we have:**

$$\Phi(\mathbf{x})^T\Phi(\mathbf{z}) = (1 + x_1z_1 + x_2z_2)^2 = \left(1 + \mathbf{x}^T\mathbf{z}\right)^2 = k(\mathbf{x},\mathbf{z})$$

- **Symmetry**: Kernel functions must be symmetric $k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$.
- **Mercer's Condition**:

$$\sum_{n=1}^{N} \sum_{m=1}^{N} k(\mathbf{x}^{(m)}, \mathbf{x}^{(n)}) c^{(m)} c^{(n)} \geq 0, \quad \forall c^{(i)} \in \mathbb{R}$$

  This condition ensures the kernel matrix $\mathbf{K}$ is positive semi-definite, allowing efficient optimization in dual problems.
- **Practical Consideration**: Some functions not satisfying Mercer's condition may still yield acceptable results and are used as kernels.

Polynomial:

$$K(\mathbf{x}, \mathbf{z}) = ((\mathbf{x} \cdot \mathbf{z}) + \theta)^d, \quad \theta \in \mathbb{R}, \, d \in \mathbb{N}$$

Gaussian radial basis function (RBF):

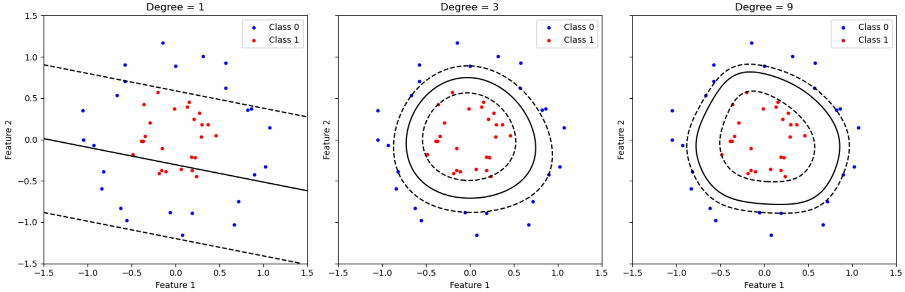
$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{z}\|^2\right), \quad \gamma \in \mathbb{R}$$

Sigmoid:

$$K(\mathbf{x}, \mathbf{z}) = \tanh(\gamma(\mathbf{x} \cdot \mathbf{z}) + r), \quad \gamma, r \in \mathbb{R}$$

Polynomial Kernel Visualization

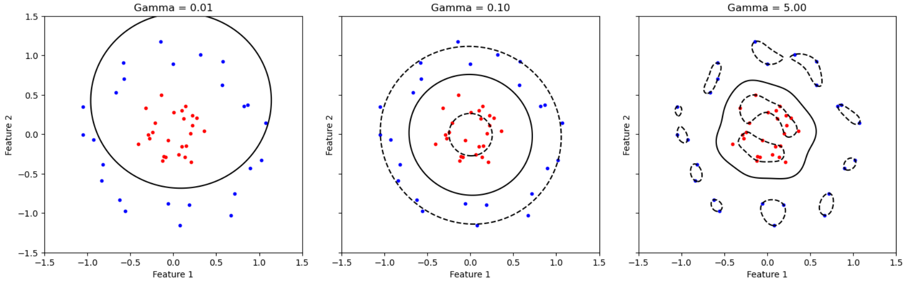
$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + \theta)^d, \quad \theta \in \mathbb{R}, \ d \in \mathbb{N}$$



Decision boundaries for polynomial kernels with degrees 1, 3, and 9.

RBF Kernel with Different Gamma Values

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{z}\|^2\right), \quad \gamma \in \mathbb{R}$$



Decision boundaries for RBF kernels with Gamma Values: 0.01, 0.1, and 5

- **Section 4: Advantages and Drawbacks**

Pros

- Work well with a clear margin of separation between classes
- Productive in high-dimensional spaces
- Effective when dimensions outnumber specimens

Pros

- Work well with a clear margin of separation between classes
- Productive in high-dimensional spaces
- Effective when dimensions outnumber specimens

Drawbacks

- Not suitable for large datasets
- Sensitive to the choice of kernel and parameters
- Memory-intensive due to storing the kernel matrix
- Not suitable for datasets with missing values

Linear SVM		Non-Linear SVM
Hard Margin	**Soft Margin**	**Kernel**
Perfectly linearly separable without any noise	Not completely linearly separable or contains noise	Not linearly separable, but can be linearly separable when mapped into a new space

References

- **Mathematics for Machine Learning** - Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong
- **Machine Learning Co Ban** - Vu Huu Tiep
- **Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines** - John C. Platt
- **Classification of Sentimental Messages** - `https://github.com/hmohebbi/SentimentAnalysis`

Demo

- Demo code: https://colab.research.google.com/drive/1DeOTjqwcZW9SZYj-gTgkdvJGIElLtBLf?authuser=0scrollTo=51xFCZIK590N

Thank you for your attention!