NTNU

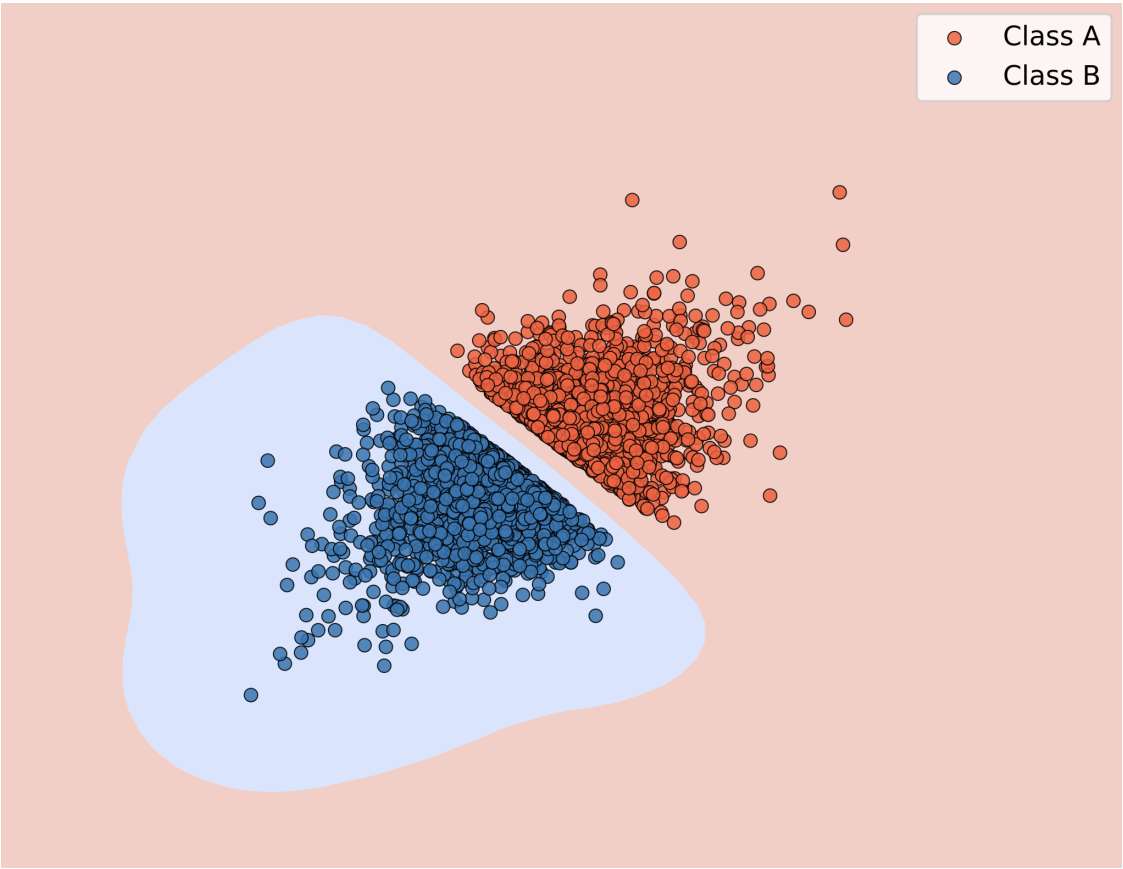
Kunnskap for en bedre verden

Department of Mathematical Sciences

TMA4180 - Optimization 1

Support Vector Classification

Theoretical Analysis and Numerical Methods for Convex Optimization Problems



Authors:
Emilie Hansen Hårklau, Hanna Rød, Thea Boge, Karine Grande

27.04.2025

Table of Contents

1 Introduction **1**

2 Mathematical Analysis of the Optimization Problem **1**

2.1 Existence of solution to the Primal Problem 1

2.2 Uniqueness of solution to the Primal Problem 2

2.3 The Lagrangian Dual 2

2.4 Existence of solution to the Dual Problem 3

2.5 Uniqueness of solution to the Dual Problem 3

2.6 Solution of the Primal Problem 4

3 Numerical methods **5**

3.1 Numerical Solution of the Primal Problem 5

3.1.1 Quadratic Penalty 5

3.2 Numerical Solution of the Dual Problem 6

3.2.1 The projection π_{Ω} 6

3.2.2 Step Length Selection and Line Search 7

3.2.3 Training the SVM 7

3.2.4 Convergence of the Method 7

3.2.5 The Linear Case 8

3.2.6 The Non–Linear Kernels 8

3.2.7 Numerical Analysis 10

4 Conclusion **11**

Bibliography **12**

Abstract

In this project, we investigate support vector machine (SVM) classification from an optimization perspective. We conduct a theoretical analysis of the existence and uniqueness of solutions for the soft-margin primal and dual formulations of the SVM optimization problem. Furthermore, we develop and implement numerical methods for solving both formulations: a quadratic penalty method for the primal problem and a projected gradient descent method for the dual. Through numerical experiments, we illustrate the behavior of linear and nonlinear classifiers and demonstrate how different kernel choices influence the resulting decision boundaries.

1 Introduction

In this project, we examine SVMs from an optimization viewpoint, with a particular focus on soft-margin classification. We analyze the existence and uniqueness of solutions to the primal and dual problem, derive the dual formulation, and implement efficient numerical methods to solve the associated optimization problems.

Throughout the report, we aim to provide clear explanations of the mathematical and numerical methods employed, compare the performance of linear and nonlinear classifiers, and discuss how the choice of kernel functions influences classification outcomes.

2 Mathematical Analysis of the Optimization Problem

In Soft Margin Classification, the Support Vector Machine (SVM) optimization problem can be formulated in two ways; the primal problem, which directly optimizes the separating hyperplane, and the dual problem, which allows kernelization and provides insight into the role of support vectors.

The primal problem is given by:

$$\min_{\substack{w\in\mathbb{R}^d,\\ b\in\mathbb{R},\\ \xi\in\mathbb{R}^M}} \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{M}\xi_i \qquad \text{s.t.} \quad \begin{cases} y_i(\langle w, x_i\rangle + b) \geq 1-\xi_i, \\ \xi_i \geq 0 \end{cases} \qquad \text{for } i = 1,\dots,M. \tag{1}$$

We will now discuss the existence and uniqueness of solutions for both the primal problem (1) and it's corresponding dual problem.

2.1. Existence of solution to the Primal Problem.

Theorem 1. *There exists a solution to the primal problem (1).*

Proof. The objective function consists of a quadratic term $\frac{1}{2}\|w\|_2^2$ and a linear term $C\sum_{i=1}^{M}\xi_i$, both of which are continuous. Consequently, their sum is continuous and, by [Gra25b, Lemma 1.29], also lower semi-continuous (LSC).

We proceed by the *direct method in the calculus of variations* [Gra25b, p. 9].

Since the feasible set Ω is assumed non-empty, there exists a minimizing sequence $(w_k, b_k, \xi_k) \subset \Omega$ satisfying

$$\lim_{k\to\infty} f(w_k, b_k, \xi_k) = \inf_{(w,b,\xi)\in\Omega} f(w, b, \xi) =: f^*.$$

We first show that the sequence (w_k, b_k, ξ_k) is bounded.

The quadratic term $\frac{1}{2}\|w\|_2^2$ penalizes large values of w, implying that (w_k) must be bounded. Similarly, since $\xi_i \geq 0$ and the objective function contains the term $C\sum_{i=1}^{M}\xi_i$ with $C > 0$, the sequence (ξ_k) must also be bounded to prevent the objective from diverging to $+\infty$.

To establish boundedness of (b_k), observe that the feasibility constraint

$$y_i(\langle w_k, x_i\rangle + b_k) \geq 1 - \xi_{i,k}, \quad \text{for all } i = 1,\dots,M,$$

must hold for each k. Since (w_k) and (ξ_k) are bounded, and x_i are fixed, the terms $\langle w_k, x_i\rangle$ are bounded. Therefore, (b_k) must also be bounded to maintain feasibility.

Thus, the sequence (w_k, b_k, ξ_k) is bounded in $\mathbb{R}^{d+1+M}$.

Since Ω is closed and (w_k, b_k, ξ_k) is bounded, the Heine–Borel theorem [Gra25b, Theorem 1.36] implies that there exists a convergent subsequence $(w_{k'}, b_{k'}, \xi_{k'})$ converging to some limit point $(w^*, b^*, \xi^*) \in \mathbb{R}^{d+1+M}$.

Now, we show that the limit point, (w^*, b^*, ξ^*), is feasible. Since each element of the sequence satisfies the constraints

$$y_i(\langle w_k, x_i\rangle + b_k) \geq 1 - \xi_{i,k}, \quad \xi_{i,k} \geq 0,$$

taking the limit as $k' \to \infty$ and using continuity of inner products and inequalities preserved under limits yields

$$y_i(\langle w^*, x_i \rangle + b^*) \geq 1 - \xi_i^*, \quad \xi_i^* \geq 0.$$

Hence, $(w^*, b^*, \xi^*) \in \Omega$.

Finally, by LSC of the objective function,

$$\liminf_{k' \to \infty} f(w_{k'}, b_{k'}, \xi_{k'}) \geq f(w^*, b^*, \xi^*).$$

Since $(w_{k'}, b_{k'}, \xi_{k'})$ is a minimizing subsequence, it follows that

$$f(w^*, b^*, \xi^*) = f^*,$$

establishing that (w^*, b^*, ξ^*) is indeed an optimal solution.

Thus, the primal problem (1) admits at least one solution. □

2.2. Uniqueness of solution to the Primal Problem. We now examine whether the solution to the primal problem is unique. By [BC99, Theorem 1] the solution to a convex programming problem, for which the objective function is strictly convex, is unique. This theorem also states that positive definiteness of the Hessian implies strict convexity of the objective function.

The Hessian of the objective function of the primal problem (1) is:

$$H_f = \begin{bmatrix} I_d & 0_{d\times 1} & 0_{d\times M} \\ 0_{1\times d} & 0 & 0_{1\times M} \\ 0_{M\times d} & 0_{M\times 1} & 0_{M\times M} \end{bmatrix}.$$

Since H_f is positive semi-definite, the objective function is convex. However, we need strict convexity, which requires H_f to be positive definite.

In this case, the w-block (I_d) is positive definite, ensuring strict convexity in w, which implies that w^* is unique. And the b-block and ξ-block are zero matrices, meaning the function is linear in b and ξ, which does not enforce strict convexity in these variables. As a result, the solution for b^* and ξ^* may not be unique.

Nevertheless, it is important to note that convexity implies that every local minimizer is also a global minimizer. Thus, while convexity does not guarantee uniqueness, it ensures that any solution we obtain is globally optimal. In fact, there may be multiple global minimizers, as we have shown that at least one exists. Additional constraints may still enforce uniqueness.

2.3. The Lagrangian Dual. Furthermore, we will now derive the Lagrangian dual of the primal problem (1).

Theorem 2. *The Lagrangian dual of problem* (1) *is given by*

$$\min_{\alpha \in R^M} \frac{1}{2}\langle \alpha, YGY\alpha \rangle - \langle 1_M, \alpha \rangle \quad s.t \quad \begin{cases} \langle y, \alpha \rangle = 0, \\ 0 \leq \alpha_i \leq C \quad for \quad i = 1, ..., M, \end{cases} \tag{2}$$

with the matrices $Y, G \in \mathbb{R}^{M\times M}$*, defined as:*

$$Y = diag(y_1, y_2, ..., y_M) \quad and \quad G = (\langle x_i, x_j \rangle)_{i,j=1,...,M}.$$

Proof. We start by finding the Lagrangian of the soft margin classification (1). We apply two sets of Lagrange multipliers:

1. $\alpha_i \geq 0$ ensures that $y_i(x_i^T \omega + b) - 1 + \xi_i \geq 0$.

2. $\mu_i \geq 0$ ensures that $\xi_i \geq 0$.

Thus, the corresponding Lagrangian function is

$$\mathcal{L}(\omega, b, \alpha, \mu, \xi) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{M} \xi_i - \sum_{i=1}^{M} \alpha_i \left(y_i(x_i^\top \omega + b) - 1 + \xi_i\right) - \sum_{i=1}^{M} \mu_i \xi_i. \tag{3}$$

We obtain the dual objective function by taking partial derivatives and setting them to zero:

$$\frac{\partial \mathcal{L}}{\partial \omega} = \omega - \sum_{i=1}^{M} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad \omega = \sum_{i=1}^{M} \alpha_i y_i x_i, \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{M} \alpha_i y_i = 0, \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad \Rightarrow \quad \mu_i = C - \alpha_i.$$

Since $\alpha_i \geq 0$, equation (2.3) leads to the constraint $0 \leq \alpha_i \leq C$.

Now plugging this back into the Lagrangian:

$$\mathcal{L}(\omega, b, \alpha, \beta, \xi) = \frac{1}{2} \left(\sum_{i=1}^{M} \alpha_i y_i x_i \right)^\top \left(\sum_{j=1}^{M} \alpha_j y_j x_j \right) - \left(\sum_{i=1}^{M} \alpha_i y_i x_i \right)^\top \left(\sum_{j=1}^{M} \alpha_j y_j x_j \right) + \sum_{i=1}^{M} \alpha_i,$$

This results in a modified Wolfe dual-optimization formula

$$\max_{\alpha} \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j y_i y_j x_i^\top x_j,$$

subject to $\quad \sum_{i=1}^{M} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall i.$

With the matrices $Y, G \in \mathbb{R}^{M \times M}$, the dual can then be written as

$$\min_{\alpha \in \mathbb{R}^M} \frac{1}{2} \langle \alpha, YGY\alpha \rangle - \langle 1_M, \alpha \rangle \quad \text{s.t} \quad \begin{cases} \langle y, \alpha \rangle = 0, \\ 0 \leq \alpha_i \leq C \quad \text{for} \quad i = 1, ..., M, \end{cases} \tag{6}$$

which is what we wanted to show. □

2.4. Existence of solution to the Dual Problem.

Theorem 3. *There exists a solution to the dual problem (2).*

Proof. The dual objective function is convex, and we only have linear constraints. By Slater's constraint qualification, strong duality holds if and only if the primal problem is feasible [Gra25a, Theorem 10.5].

That is, there exists a point (w^*, b^*, ξ^*) with $w^* \in \mathbb{R}^d$, $b^* \in \mathbb{R}$, and $\xi^* \in \mathbb{R}^M$ satisfying the primal constraints

$$y_i(\langle w^*, x_i \rangle + b^*) \geq 1 - \xi_i^*, \quad \xi_i^* \geq 0.$$

Since Theorem 1 establishes primal feasibility and boundedness, there exists an optimal dual solution α^*. □

2.5. Uniqueness of solution to the Dual Problem. The Hessian of the dual objective function is given by:

$$H_g = \nabla^2 g(\alpha) = YGY. \tag{7}$$

Since G is a Gram matrix, it is positive semi-definite. Consequently, $H_g = YGY$ is also positive semi-definite. If the Gram matrix is of full rank, it is positive definite, which implies that the Hessian is also positive definite. Thus, the objective function of the dual problem is strictly convex if and only if the Gram matrix is of full rank.

In the linear case, where $G(x_i, x_j) = x_i^T x_j$, the Gram matrix is full rank if and only if the feature vectors $\{x_1, \ldots, x_M\}$ are linearly independent in $\mathbb{R}^d$. If $M > d$, G cannot be full rank.

For kernel methods, $G := K(x_i, x_j)$ is full rank if the kernel function maps data into a sufficiently high-dimensional space where the transformed feature vectors remain linearly independent. The Gaussian and Laplacian kernels, for instance, ensure full rank unless duplicate points exist.

When solving the dual problem, one obtains a solution for α^*. Even if α^* is unique, this does not necessarily imply that the corresponding primal variables ξ^* and b^* are unique.

2.6. Solution of the Primal Problem. Moreover, we will show how one obtains the solution of the primal problem. We start by introducing a proposition which will be used in further analysis.

Proposition 1. *Derived from the Lagrangian* (3)*, the Karush-Kuhn-Tucker (KKT) conditions in* (x^*, α^*, μ^*) *are:*

$$\begin{array}{ll} y_i(\langle w, x_i^* \rangle + b) \geq 1 - \xi_i, & \alpha_i^* + \mu_i^* = C, \\ \xi_i \geq 0, & \alpha_i^* \geq 0, \\ w^* = \sum_{i=1}^M \alpha_i^* y_i x_i, & \mu_i^* \geq 0, \\ \sum_{i=1}^M \alpha_i^* y_i = 0, & \alpha_i^*(y_i(\langle w^*, x_i^* \rangle + b) - 1 + \xi_i) = 0, \forall i \\ & \mu_i^* \xi_i = 0 \quad \forall i. \end{array}$$

Proof. The result follows by applying the KKT conditions to the Lagrangian (3) of the primal problem (1).

The KKT-conditions are derived in four steps; stationary, complementary slackness, primal and dual feasibility.

For the stationary conditions, taking partial derivatives and setting them to zero gives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w} &= w - \sum_{i=1}^M \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w^* = \sum_{i=1}^M \alpha_i^* y_i x_i, \\ \frac{\partial \mathcal{L}}{\partial b} &= -\sum_{i=1}^M \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^M \alpha_i^* y_i = 0, \\ \frac{\partial \mathcal{L}}{\partial \xi_i} &= C - \alpha_i - \mu_i = 0 \quad \Rightarrow \quad \alpha_i^* + \mu_i^* = C. \end{aligned}$$

The primal feasibility demands we fulfill the original constraints from problem (1):

$$y_i(\langle w^*, x_i \rangle + b^*) \geq 1 - \xi_i^*, \quad \xi_i^* \geq 0.$$

For the dual feasibility we require the Lagrangian multipliers α_i^* and μ_i^* to be greater than or equal to 0.

Lastly, for the complementary slackness we demand:

$$\alpha_i^*(y_i(\langle w^*, x_i \rangle + b^*) - 1 + \xi_i^*) = 0 \quad \text{and} \quad \mu_i^* \xi_i^* = 0.$$

These nine conditions jointly define the KKT system. □

These conditions fully characterize the optimality of the solution to the primal-dual pair. The stationarity condition yields a closed-form expression for w^* as a linear combination of training examples. The complementarity conditions identify the support vectors, indicating that only vectors satisfying $0 < \alpha_i^* < C$ lie exactly on the margin and can thus be used to determine b^*. Finally, the dual feasibility and the sum constraint $\sum_{i=1}^M \alpha_i^* y_i = 0$ play a central role in ensuring the optimal balance between maximizing the margin and handling constraint violations.

This results in the following theorem.

Theorem 4. *One obtains the solution of the primal problem* (1) *with the formulas*

$$w^* = \sum_{i \in I_S} \alpha_i^* y_i x_i, \tag{8}$$

and

$$b^* = y_i - \langle w^*, x_i \rangle \qquad \textit{for any } i \in \tilde{I}_S = \{1 \leq i \leq M : 0 < \alpha_i^* < C\}. \tag{9}$$

Proof. From Proposition 1 we can read the optimal weight w^* as

$$w^* = \sum_{i=1}^M \alpha_i^* y_i x_i.$$

Moreover, for the support vectors the constraint $\alpha_i^*(y_i(\langle w,^* x_i^* \rangle + b) = 1 - \xi_i$ is active. We then get two possible cases:

$$\begin{aligned} y_i(\langle w^*, x_i \rangle + b) &= 1, \qquad && \forall i \qquad && 0 < \alpha_i^* < C, \\ y_i(\langle w^*, x_i \rangle + b) &< 1, && \forall i && a_i^* = C. \end{aligned}$$

Solve for b^*, and we get

$$b^* = y_i - \langle w^*, x_i \rangle \qquad \text{for any } i \in \tilde{I}_S = \{1 \leq i \leq M : 0 < \alpha_i^* < C\}.$$

□

Note that the value of b^* is not uniquely determined by the KKT conditions alone unless further assumptions (e.g. strict convexity) are imposed, as discussed in Section 2.2. However, any value computed from (9) using support vectors in the set $\tilde{I}_S$ will yield a valid solution.

3 Numerical methods

In this section, we will discuss and develop numerical methods for solving both the primal and dual problems. Note that additional models and convergence plots can be found in the attached Jupyter Notebook.

3.1. Numerical Solution of the Primal Problem.

3.1.1. Quadratic Penalty. The method we used to solve the primal soft-margin SVM problem is the Quadratic Penalty (QP) method. This method was chosen because it transforms the original constrained and non-smooth optimization problem into an unconstrained, smooth one, which is more suitable for gradient-based algorithms. In our case, the penalty parameter is fixed to $C = 1$, representing a seemingly fitting price for misclassifications.

Specifically, the QP method replaces the hinge loss $l_h(t) = \max(0, 1 - t)$ with the squared hinge loss $l_q(t) = (\max(0, 1 - t))^2$, resulting in a continuously differentiable objective function:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{M}(\max(0, 1 - y_i(\langle x_i, w \rangle + b)))^2.$$

This smooth formulation allows the use of efficient gradient-based optimization methods without the need to handle inequality constraints explicitly, making the implementation simpler and the convergence behavior more predictable.

The method proceeds by computing the gradient of the objective with respect to w and b, where only points violating the margin, i.e. those with $y_i(\langle x_i, w \rangle + b) < 1$ contribute to the loss and gradient. Convergence is monitored by the norm of the gradient, and the algorithm terminates once it falls below a prescribed tolerance.

This smooth and convex formulation enables stable gradient descent steps and avoids the need for dual variables or KKT conditions, making the method both efficient and easy to implement. It tends to converge quickly on moderate-sized datasets due to the availability of analytic gradients and allows direct control over the margin-loss trade-off via the regularization parameter C. Numerical testing has shown that the method is sensitive to the choice of learning rate and regularization strength, and may not scale optimally to very large datasets. However, with proper tuning, it remains a robust and effective approach for linear SVM training.

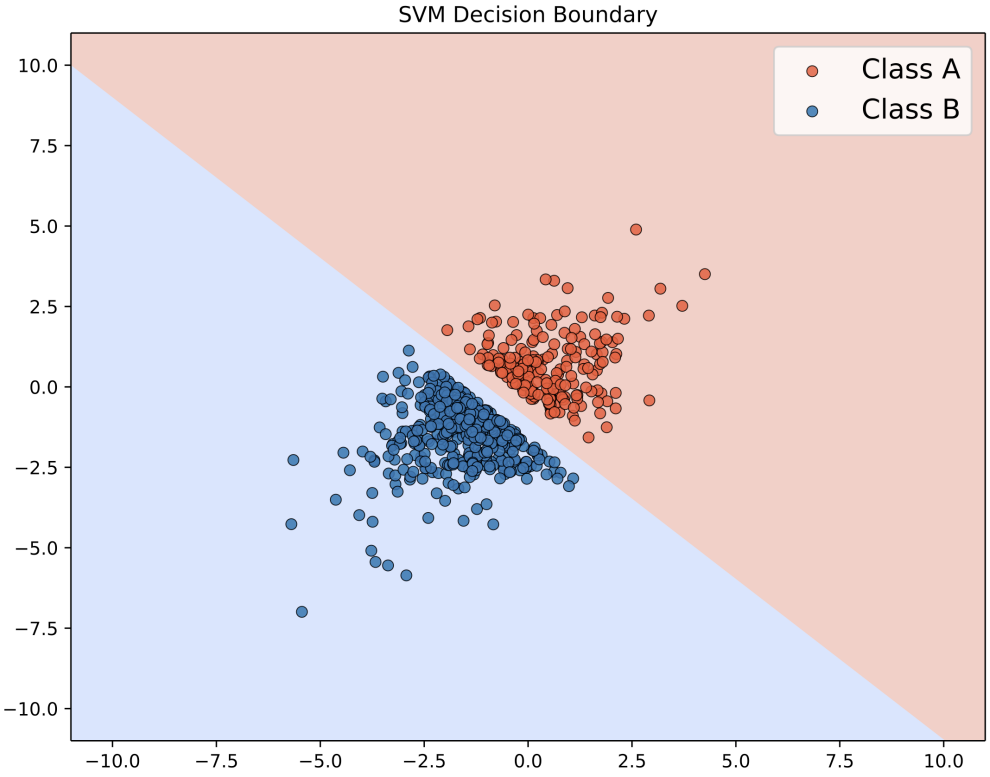


Figure 1: Training the SVM with the quadratic penalty algorithm. M = 600, $\gamma = 0.5$.

As expected, the method has converged to an accurate solution, showing that the QP method is a good option for our problem.

The primal problem has M slack variables and $2M$ constraints. The direct solution of the primal problem scales poorly with increasing M, as the size of the gradient grows proportional to M, making the problem numerically unstable. Each iteration has a computational cost of $\mathcal{O}(Md)$, where M is the number of data points and d is the number of features (i.e. dimensions of each x_i). This makes the method unsuitable for large datasets. Therefore, while the quadratic penalty approach works well for moderate M, it becomes impractical for large-scale problems.

3.2. Numerical Solution of the Dual Problem. We will now define a numerical method for the solution of the dual problem (2) using either the standard Gram matrix $G := (\langle x_i, x_j \rangle)_{i,j=1,\dots,M}$, or various fixed kernels K from an RKHS, where $G := K(x_i, x_j)$. The method we are using has been developed in [DF06], and is based on a projected gradient descent method. That is, one computes the iterates as

$$\alpha^{(k+1)} = \pi_\Omega(\alpha^{(k)} - \tau_k \nabla f(\alpha^{(k)})), \tag{11}$$

where

$$f(\alpha) := \frac{1}{2}\langle \alpha, YGY\alpha \rangle - \langle 1_M, \alpha \rangle,$$

is the function to be minimized, and

$$\Omega = \{\alpha \in \mathbb{R}^M : \langle y, \alpha \rangle = 0 \text{ and } 0 \le \alpha \le C\},$$

is the feasible set, π_Ω denotes the projection onto Ω, and $\tau_k > 0$ is a suitable step length. To ensure convergence of the method, we will also include an additional line search, so the update is computed as

$$\alpha^{(k+1)} = \pi_\Omega(\alpha^{(k)} + \theta d^{(k)}), \quad \text{for some } 0 < \theta_k \le 1 \text{ and } \quad d^{(k)} = \alpha^{(k+1)} - \alpha^{(k)}$$

3.2.1. The projection π_Ω. The projection $\pi_\Omega(\beta)$ can be formulated as the solution to the quadratic programme

$$\min_{\alpha \in \mathbb{R}^M} \frac{1}{2}\|\alpha - \beta\|_2^2 \quad \text{s.t.} \begin{cases} \langle y, \alpha \rangle = 0, \\ 0 \le \alpha_i \le C \quad \text{for } i = 1, \dots, M. \end{cases} \tag{12}$$

This is equal to utilizing the partial Lagrangian and finding the solution of the Lagrangian penalty problem

$$\min_{\alpha\in\mathbb{R}^M} \mathcal{L}(\alpha;\lambda) := \min_{\alpha\in\mathbb{R}^M} \frac{1}{2}\|\alpha-\beta\|_2^2 - \lambda\langle y,\alpha\rangle \qquad \text{s.t.} \quad 0\le\alpha\le C, \tag{13}$$

which has for each $\lambda\in\mathbb{R}$ a unique solution on the explicit form

$$\alpha(\lambda)_i = \min\{\max\{\beta_i+\lambda y_i, 0\}, C\} = \text{median}\{0, C, \beta_i+\lambda y_i\}.$$

In order to find the correct value of λ such that $r(\lambda) = \langle y, \alpha(\lambda)\rangle = 0$, we implemented a two-part algorithm consisting of a bracketing phase, and a secant phase. Since $r(\lambda)$ is piecewise-linear, continuous, and monotonically non-decreasing in λ, the problem $r(\lambda) = 0$ is well-posed whenever the constraints are consistent. If the problem (2) is feasible, then the bracketing phase is guaranteed to terminate with a bracket $[\lambda_l, \lambda_u]$ such that $r(\lambda_l) < 0$ and $r(\lambda_u) > 0$, which contains the solution of the equation $r(\lambda) = 0$. This is done by expanding outward from an initial guess using a fixed step size and geometric scaling until a sign change is detected. Once a bracket is located, the root of $r(\lambda) = 0$ is found by repeated use of the secant method. At each iteration we evaluate $r(\lambda)$. If $r(\lambda) > 0$ and λ lies in the left half of the interval, then a secant step on the next iteration is based on λ_l and λ. If λ lies in the right half, then a secant step is based on either λ_u and λ, or a step to the point $\frac{3}{4}\lambda_l + \frac{1}{4}\lambda$ is taken, whichever yields a smaller step. We terminate the secant phase if preset tolerances on either $r(\lambda)$ or $\Delta\lambda$ are met.

3.2.2. Step Length Selection and Line Search. We define the step length τ_k in (11) as the *Barzilai-Borwein step length* with

$$s^{(k)} := \alpha^{(k+1)} - \alpha^{(k)} \quad \text{and} \quad z^{(k)} := \nabla f(\alpha^{(k+1)}) - \nabla f(\alpha^{(k)}),$$

which yields the step length

$$\tau_{k+1}^{(2)} := \frac{\langle s^{(k)}, s^{(k)}\rangle + \langle s^{(k-1)}, s^{(k-1)}\rangle}{\langle s^{(k)}, z^{(k)}\rangle + \langle s^{(k-1)}, z^{(k-1)}\rangle}. \tag{14}$$

In case $\langle s^{(k)}, z^{(k)}\rangle \le 0$ and $\langle s^{(k-1)}, z^{(k-1)}\rangle \le 0$, this does not yield a well-defined step length, so we define the new step length as $\tau_{k+1} := \max\{\min\{\tau_{k+1}^{(2)}, \tau_{\max}\}, \tau_{\min}\}$, where $\tau_{\min}$ and $\tau_{\max}$ are pre-defined bounds. However, this does not guarantee convergence of the algorithm, and we need to introduce an additional line search.

When performing an exact line search, θ_k is defined as the solution of the problem $\min_{0\le\theta\le1} f(\alpha^{(k)} + \theta d^{(k)})$, which analytical solution can be easily computed. To determine whether a line search should be carried out, a dynamic reference value f_{ref} is maintained. Initially, $f_{\text{ref}} = +\infty$, and the best observed value is $f_{\text{best}} = f(\alpha^{(0)})$. A candidate value f_c is also tracked. A line search is triggered if the new function value $f(\alpha^{(k)} + d^{(k)}) > f_{\text{ref}}$. To balance efficiency and robustness, we allow the function values to increase sometimes, as long as it does not happen too often. The number L denotes how many iterations in a row we are allowed to perform without decreasing f_{best} before f_{ref} is relaxed to f_c.

3.2.3. Training the SVM. To train the SVM, we solve the problem (2), using the projected gradient descent method described above. The Lagrange multipliers α are initialized to zero and iteratively updated using the Barzilai–Borwein step length with exact line search. The algorithm terminates when the change in α falls below a fixed tolerance or a maximum number of iterations is reached. Upon convergence, support vectors x_i are identified as those with $\alpha_i \in (0, C)$. Predictions are made using the decision function

$$f(x) = \sum_{i\in I_s} \alpha_i y_i K(x_i, x) + b,$$

where K is the kernel function (or $K(x_i, x) = G(x_i, x) = \langle x_i, x\rangle$ in the linear case), and b is the bias term computed from the support vectors. The predicted label is given by $\text{sign}(f(x))$, and the decision boundary is implicitly defined by the set $\{x\in\mathbb{R}^d : f(x) = 0\}$.

3.2.4. Convergence of the Method. By Proposition 2.4 [Gra25a] it follows that if x^* is a solution of our primal problem, then

$$x^* = \pi_\Omega(x^* - \tau\nabla f(x^*)) \quad \text{for any} \quad \tau > 0.$$

This implies that every local solution of the primal problem is a fixed point of the mapping $K : \mathbb{R}^d \to \mathbb{R}^d$,

$$K(x) = \pi_\Omega(x - \tau\nabla f(x)).$$

Now, solving the problem by the projected gradient descent method, the algorithm converges if f is a strongly convex C^1-function, and the step lenght $\tau > 0$ is chosen sufficiently small. Moreover, by adding a non-monotone line search, we obtain a globaly convergent algorithm as $\|x_k - \pi_\Omega(x_k - \tau\nabla f(x_k))\| \to 0$.

We will assume that our dual objective function is a strongly convex C^1-function, as we have not been assigned to explicitly prove this. Further, our Barzilai-Borwein step size ensures τ is choosen sufficiently small. Lastly the line search ensures the global convergence of the algorithm. Thus, the method converges.

3.2.5. The Linear Case. In the linear case, the standard Gram matrix defined by

$$G := (\langle x_i, x_j \rangle)_{i,i=1,\ldots,M},$$

is used when we assume that the data is linearly separable, or at least approximately so, with the soft margin formulation. By working directly in the original space, we save computational time, as we do not map the data into a higher-dimensional feature space.

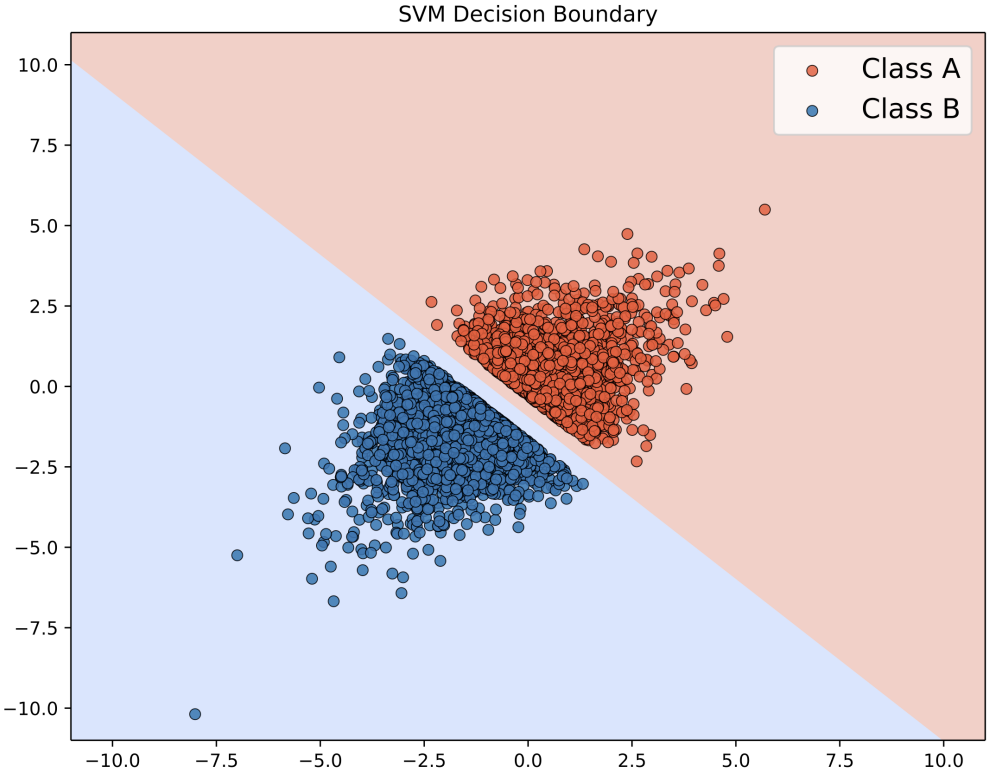


Figure 2: Training the SVM with the linear kernel. M = 4000, $\gamma = 0.5$.

Training of the SVM results in a hyperplane clearly separating the classes A and B.

3.2.6. The Non–Linear Kernels. In many situations, the assumption that we can more or less neatly separate the two classes A and B by a hyperplane is not realistic. Because of that, it is necessary to allow the functions $f : \mathbb{R}^d \to \mathbb{R}$ that define our classifier to be more general. The idea is that we replace the affine function $f_{b,w}(x) = \langle w, x \rangle + b$ which we used for linear support vector classification, by a function of the form $f(x) = w(x) + b$, where w is an element of a fixed RKHS. Now we compute the Gram matrix by

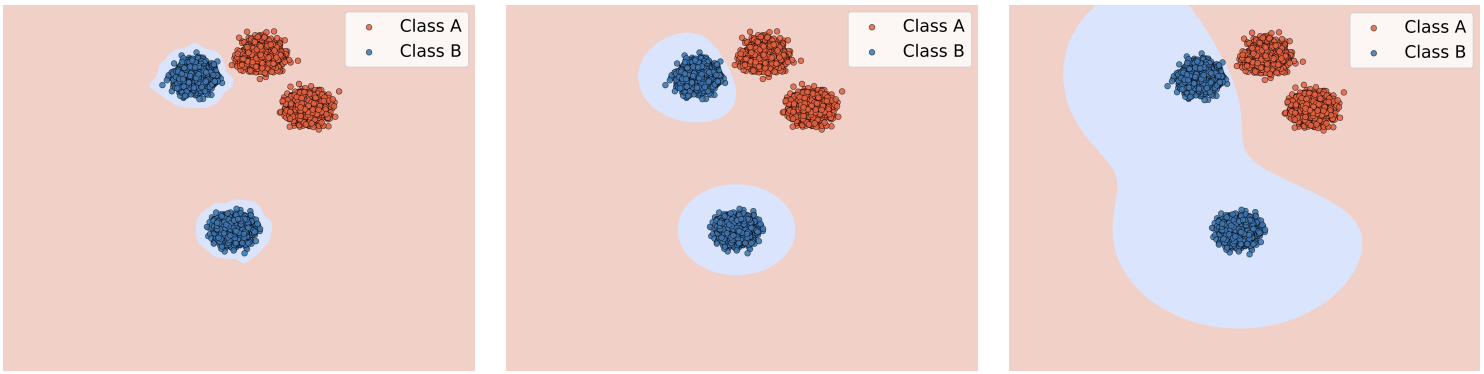
$$G_{ij} = K(x_i, x_j).$$

With this new G matrix, we again aim to solve the problem (2), using several different kernels.

The Gaussian kernel with bandwidth $\sigma > 0$ defined by

$$K(x, y) = e^{-\frac{\|x-y\|_2^2}{2\sigma^2}},$$

is used for learning smooth and flexible decision boundaries. The behavior of the algorithm depends heavily on σ: small values (e.g. $\sigma = 0.2$) lead to tightly fitted, underfitting-prone boundaries with slow convergence, while large values (e.g. $\sigma = 2.5$) result in overfitting and poor separation. Moderate values (e.g. $\sigma = 1.3$) yield a good trade-off between flexibility and generalization. Gaussian is generally a good default for smooth, medium-scale patterns.



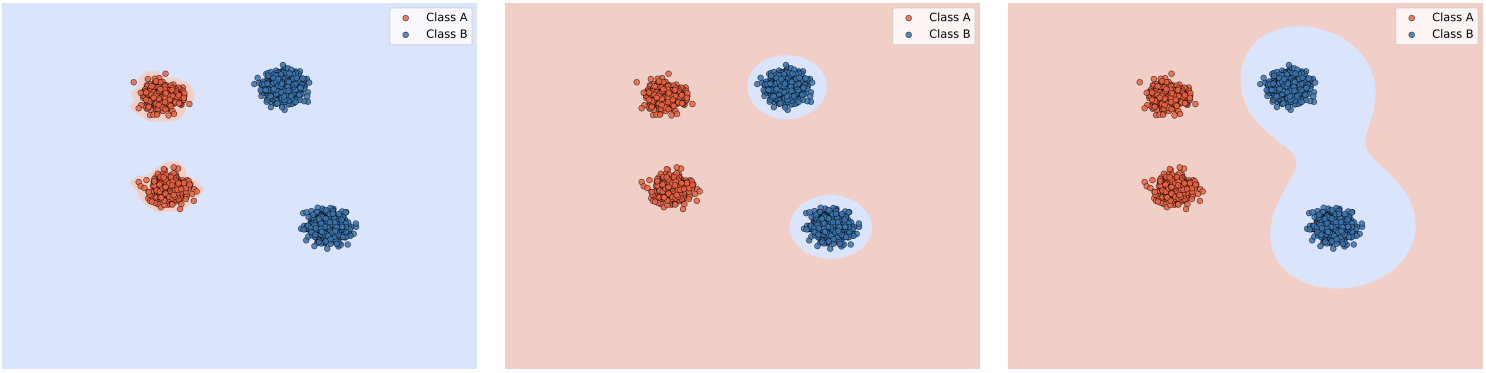
(a) Underfitting, $\sigma = 0.2$. (b) Good trade-off, $\sigma = 1.3$. (c) Overfitting, $\sigma = 2.5$.

Figure 3: Comparison of decision boundaries produced by different values for the hyperparameters for the Gaussian kernel. Each model was trained on the same dataset with $M = 9000$. The differences highlight how the choice of σ affects the classification boundary.

The Laplacian kernel with bandwidth $\sigma > 0$ defined by

$$K(x, y) = e^{-\frac{\|x-y\|_2}{\sigma}},$$

has a sharper decay than the Gaussian and allows for more localized and abrupt decision boundaries. It is slightly more robust to changes in σ and did not become unstable at higher values. Optimal performance was observed around $\sigma = 1$. However, it converged slower than the Gaussian kernel across most settings. The Laplacian kernel is particularly suitable for datasets with abrupt class transitions or lower underlying smoothness.



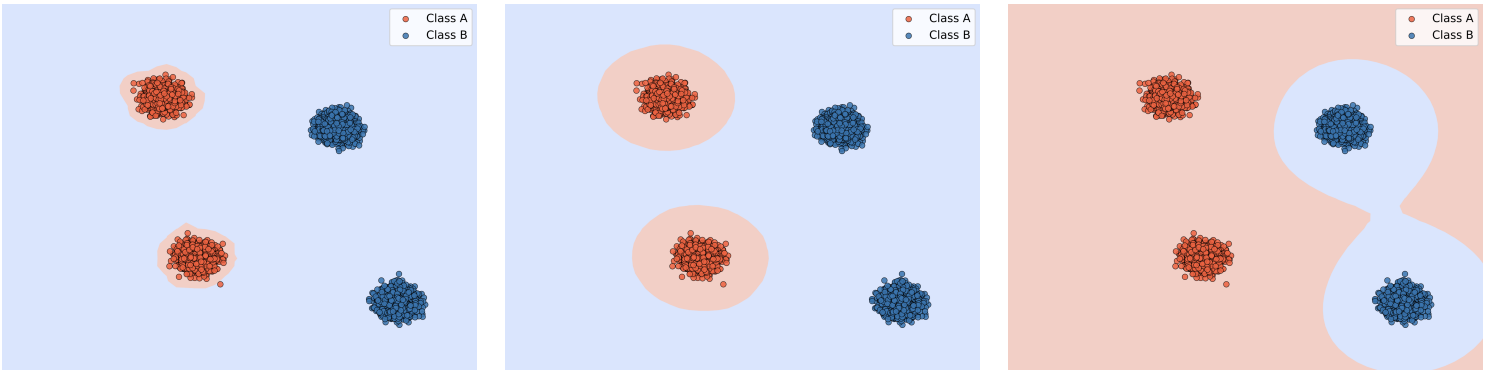
(a) Underfitting, $\sigma = 0.1$. (b) Good trade-off, $\sigma = 1$. (c) Overfitting, $\sigma = 2$.

Figure 4: Comparison of decision boundaries produced by different values for the hyperparameters for the Laplacian kernel. Each model was trained on the same dataset with $M = 3000$. The differences highlight how the choice of σ affects the classification boundary.

The Inverse Multiquadric kernel with bandwidth $\sigma > 0$ and parameter $s > 0$ defined by

$$K(x, y) = \frac{1}{(\sigma^2 + \|x - y\|_2^2)^s},$$

yields broader and smoother decision boundaries due to its heavy-tailed nature. It was less sensitive to parameter changes than the other kernels. We notably observed that changes in s did not affect the results significantly. Fast convergence and compact decision regions were achieved for $\sigma = 0.5$ and $s = 1$. Larger σ values smoothed out the boundary excessively and slowed convergence. This kernel is beneficial when the class separation is globally smooth or when one wants to avoid the extreme locality of the Gaussian kernel, provided that care is taken to avoid underfitting and sensitivity to outliers.



(a) Underfitting, $\sigma = 0.1$, $s = 1$.

(b) Good trade-off, $\sigma = 0.5$, $s = 1$.

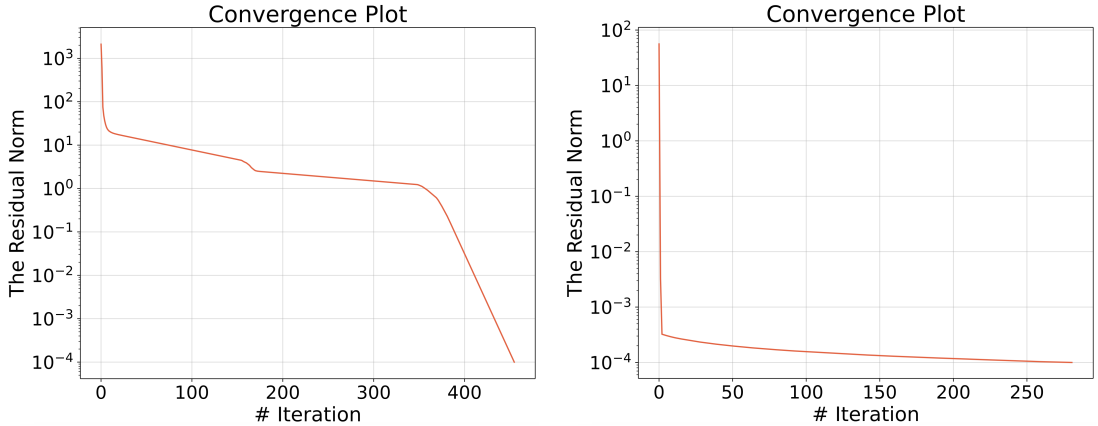
(c) Overfitting, $\sigma = 2$, $s = 1$.

Figure 5: Comparison of decision boundaries produced by different values for the hyperparameters for the Inverse Multiquadratic kernel. Each model was trained on the same dataset with $M = 7500$. The differences highlight how the choice of σ affects the classification boundary. To see how s affects the classification boundary, check out the Jupyter Notebook.

3.2.7. Numerical Analysis. The convergence rate of our dual SVM method, based on projected gradient descent, is theoretically sublinear for general convex problems, and linear if the dual problem is strongly convex (e.g., with a positive definite Gram matrix). However, the incorporation of Barzilai-Borwein step sizes and exact line search accelerates convergence in practice, often exhibiting superlinear behavior in the early stages, consistent with findings in [DF06].

The complexity of our dual SVM solver is $O(M^2)$ per iteration, primarily due to the Gram matrix-vector multiplications. The number of iterations depends on the problem's conditioning and convergence tolerance. In practice, the Barzilai-Borwein step size adaptation and exact line search significantly reduces iteration count. Compared to standard QP solvers, this approach demonstrates quicker computations for medium- to large-sized datasets.

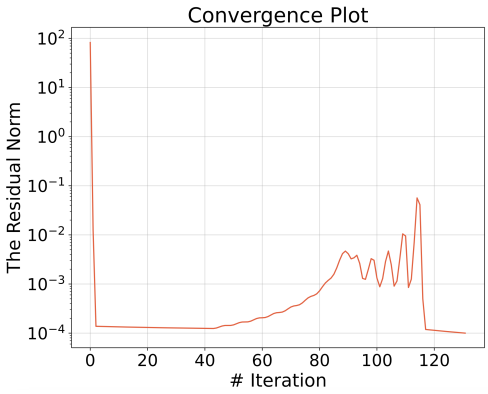
The method has numerical stability, maintaining feasibility through projections. The convergence is rapid in early iterations, but may slow down near the optimum. The line search prevents stalling in the wrong trajectory, eventually leading to convergence. We see that the line search is especially important for the non-linear kernels, as the trajectories fluctuate quite a lot, before eventually reaching the minima. This can be seen in Figures 6 and 7.



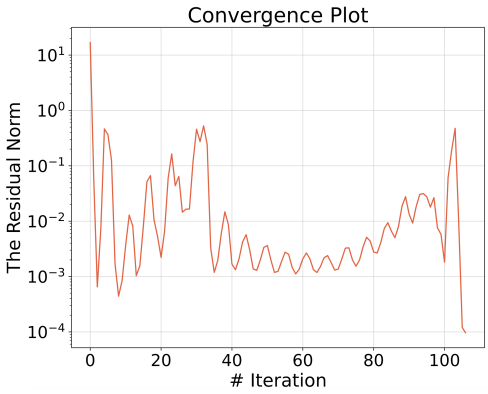
(a) Convergence plot for the Quadratic Penalty method.

(b) Convergence plot for the Projected Gradient Descent in the linear case.

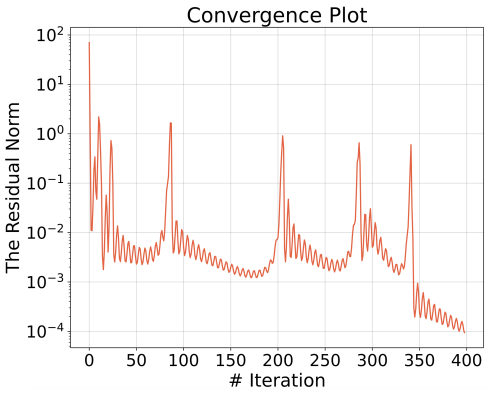
Figure 6: Convergence plots for solving the soft margin problem in the linear case.



(a) Convergence plot for the Projected Gradient Descent, with the Gaussian kernel.



(b) Convergence plot for the Projected Gradient Descent, with the Laplacian kernel.



(c) Convergence plot for the Projected Gradient Descent, with the Inverse Multi-quadratic kernel.

Figure 7: Convergence plots for solving the soft margin problem, using the non-linear kernels.

4 Conclusion

In this project, we investigated SVM through convex optimization. We showed that the primal problem admits at least one solution, and that the solution is globally optimal. Furthermore, we derived the Lagrangian dual formulation, proved strong duality, and examined under which conditions the dual solution is unique. Finally, we demonstrated how the optimal primal variables can be explicitly obtained from the dual solution through the corresponding KKT conditions.

On the numerical side, we implemented a quadratic penalty method for the primal problem, which proved efficient for moderate-sized datasets, but exhibited limitations in scalability. For the dual problem, we implemented a projected gradient descent method combined with a Barzilai–Borwein step size and dynamic line search, which demonstrated robust convergence and superior scalability compared to the quadratic penalty method.

The numerical experiments confirmed the theoretical properties of support vector classification. In particular, the linear classifiers successfully separated the classes, while the experiments with nonlinear kernels illustrated how kernel choices affect the complexity and flexibility of the decision boundary. These results highlight the effectiveness of convex optimization approaches for both linear and nonlinear classification problems.

Future work could focus on improving scalability for large datasets, optimizing kernel parameter selection, and extending the methods to more generalized or multiclass classification settings.

Bibliography

[BC99] Christopher J.C. Burges and David J. Crisp. *Uniqueness of the SVM Solution*. Conference paper. 1999. URL: https://papers.nips.cc/paper_files/paper/1999/file/c4492cbe90fbdbf88a5aec486aa81ed5-Paper.pdf.

[DF06] Yu-Hong Dai and Roger Fletcher. 'New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds'. In: *Mathematical Programming* 106.3 (2006), pp. 403–421.

[Gra25a] Markus Grasmair. *Optimisation with Convex Constraints*. Lecture notes. 2025. URL: https://www.math.ntnu.no/emner/TMA4180/2025v/notes/ConvexOptimisation.pdf.

[Gra25b] Markus Grasmair. *Theory of Unconstrained Optimisation*. Lecture notes. 2025. URL: https://www.math.ntnu.no/emner/TMA4180/2025v/notes/UnconstrainedOptimisation_Theory.pdf.

[NTN] Department of Marine Technology NTNU. *IMT Software Wiki - LaTeX*. URL: https://www.ntnu.no/wiki/display/imtsoftware/LaTeX (visited on 15th Sept. 2020).

Trondheim, Norway