SVM Algorithm

with Lagrange multipliers and KKT conditions

Matteo Conti

July 27, 2021

University of Bologna

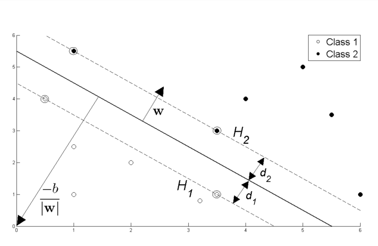
Table of contents

Problem Formulation

Our Problem

initial constrainted problem:

$$\min \quad \frac{1}{2}\|w\|^2$$
$$\text{s.t.} \quad \forall_i \; y_i(x_i \cdot w + b) - 1 \geq 0$$



the points (circled) $H1$ and $H2$ that lie closest to the separating hyperplane, i.e. the **Support Vectors**, can be described by

$$x_i \cdot w + b = +1 \quad \text{for} \quad H1$$
$$x_i \cdot w + b = -1 \quad \text{for} \quad H2$$

Lagrangian Multipliers Strategy

we formulate our problem as:

$$L_P \ \equiv \ \frac{1}{2}\|w\|^2 - \alpha[\forall_i \ y_i(x_i \cdot w + b) - 1]$$

where α_i are the Lagrange **multipliers** such that:

$$\forall_i \ \alpha_i \ \geq 0$$

we want to **min** $L_P(w, b)$ and then **max** $L_P(\alpha)$.

Final Problem Formulation

Dual form of the primary Lagrangian problem:

$$\max_{\alpha} \; L_D \;\equiv\; \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \equiv\; \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \alpha^T H \alpha$$

where:

$$H \;\equiv\; y_i y_j x_i \cdot x_j$$

subject to:

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \quad \text{and} \quad \forall_i \; \alpha_i \geq 0$$

Quadratic-Programming Optimization

Quadratic-Programming Solver

We have used the *cvxopt* Python library in order to obtain the solutions for our Quadratic-Programming problem.

This library solve a problem in the form

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^T P x + q^T x \\ s.t. \quad & Gx \leq h \\ & Ax = b \end{aligned}$$

So we've ended up in adapting our max_x L_D problem to the above formulation in order to find α_i solutions (look at the code for more details).

Soft-Margin Assumption

Soft-Margin Assumption

Dual form of the primary Lagrangian problem:

$$\max_{\alpha} \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H \alpha$$

subject to:

$$\sum_{i=1}^{L} \alpha_i y_i = 0 \quad \text{and} \quad \forall_i \ \mathbf{C} \geq \boldsymbol{\alpha_i} \geq \mathbf{0}$$

where **C** controls the trade-off between the slack variable penalty and the size of the margin (a low **C** makes the hyperplane decision surface smooth, while a high **C** aims at classifying all points correctly).

Hard-Margin vs Soft-Margin

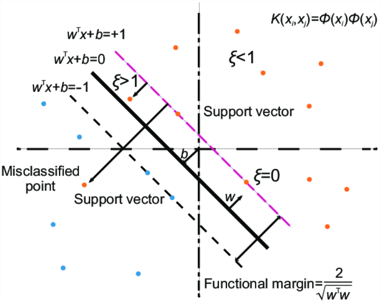


Figure 1: Hard-Margin

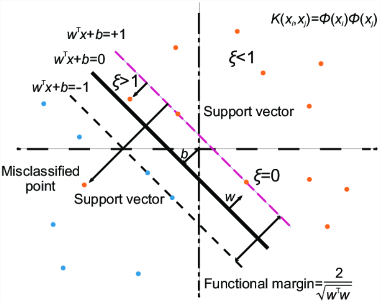


Figure 2: Soft-Margin

Non-Linear Kernel

Non-Linear Kernel

Dual form of the primary Lagrangian problem:

$$\max_{\alpha} \; L_D \equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T \mathbf{H}\alpha$$

where:

$$H \equiv y_i y_j \mathbf{k}(\mathbf{x_i}, \mathbf{x_j})$$

linear kernel is defined as:

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

polynomial kernel is defined as:

$$k(x_i, x_j) = (\phi(x_i) \cdot \phi(x_j) + 1)^{deg}$$

Non-Linear Kernel

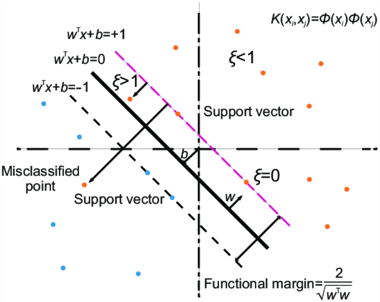


Figure 3: Linear Kernel

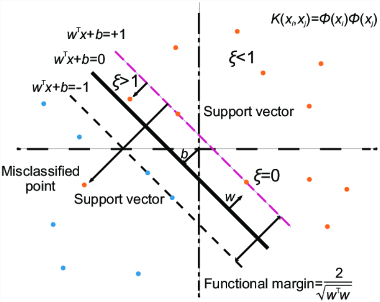


Figure 4: Polynomial Kernel

https://github.com/contimatteo/SVM-from-scratch