Support Vector Machines

Geometry, Convex Optimization, and Kernels

Yoann Pull

Laboratoire d'Economie d'Orléans
Square Research Center

Version: October 23, 2025

Teaching Material.

About this course

This course is taught in the ESA Master's programme at the University of Orléans. It is intended for Master's-level students with prerequisites in analysis, linear algebra and some optimization. Certain sections involve more advanced mathematics; complementary resources are indicated throughout the text.

Contact — For typos, errors or suggestions, please write to yoann.pull.pro@gmail.com.

Figure code — The code used to generate the figures is available on GitHub: https://github.com/YoannPull/svm_courses.

Technical section.

Sections prefixed with this pictogram are more technical. At the end of these sections, a box entitled *"Further reading"* provides additional references.

References The main references used to write this course are:

- Bishop (2006)
- Hastie et al. (2009)
- Boyd and Vandenberghe (2004)
- Hurlin (2025)

Selected application studies Selection of SVM/SVR Applications in Risk Management and other fields

- **Risk Management.** (Baesens et al., 2003; Loterman et al., 2012; Tobback et al., 2014; Yao et al., 2015, 2017)
- **Text categorisation.** (Joachims, 1998)
- **Bioinformatics.** (Guyon et al., 2002)
- **Remote sensing.** (Melgani and Bruzzone, 2004)
- **Computer vision.** (Osuna et al., 1997)

Contents

1 A brief history of SVMs and linear classifiers

The first milestone goes back to the **Perceptron** of Rosenblatt (1958), an iterative algorithm that updates $(\boldsymbol{w}, b)$ through successive corrections. Soon after, Novikoff (1962) established its convergence under the assumption of linear separability, thereby providing the first theoretical framework for linear classifiers.

In the 1960s and 1970s, Vapnik (1998) (with Chervonenkis) shifted the core question: rather than explaining *how* to learn, they investigated *when* learning generalizes. With the VC dimension and the principle of *Structural Risk Minimization* (SRM), they provided a compass: controlling model complexity to ensure out-of-sample performance. This shift—from weight updates to generalization bounds—paved the way for a method that is both geometrically and statistically grounded.

In the early 1990s, the modern version of Support Vector Machines took shape. Boser et al. (1992) formulated margin maximization as a convex quadratic program and made nonlinear boundaries natural through the *kernel trick*—already sketched by Aizerman et al. (1964)—which replaces explicit feature projections with implicit inner products. Cortes and Vapnik (1995) extended the approach to non-separable cases with the *soft margin* (parameter C) and the *hinge loss*, balancing error tolerance with model complexity control.

On the theoretical side, the anchoring in Reproducing Kernel Hilbert Spaces (RKHS) and the *representer theorem* (Kimeldorf and Wahba, 1971; Schölkopf and Smola, 2002) explain why the optimal solution can be written as a linear combination of kernels centered on only a few observations: the *support vectors*. In short, the field moved from a local update rule (Perceptron) to a global optimization principle (SVM) that unites geometry, regularization, and statistical foundations of generalization.

2 Euclidean geometry essentials

Before introducing SVMs, we fix the notation and recall a few tools from vector geometry used throughout the course.

2.1 Euclidean space, inner product and norm

Definition 2.1 (Euclidean space)**.** A Euclidean space is a finite-dimensional real vector space E endowed with an inner product $\langle \cdot, \cdot \rangle : E \times E \to \mathbb{R}$ that is symmetric, bilinear, and positive definite. The associated norm is $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ and the distance is $\operatorname{dist}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$.

We work in $\mathbb{R}^d$ with the **Euclidean inner product**

$$\langle \boldsymbol{x}, \boldsymbol{z} \rangle = \sum_{k=1}^{d} x_k z_k,$$

and the associated **Euclidean norm**

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} = \Big(\sum_{k=1}^{d} x_k^2 \Big)^{1/2}.$$

Definition 2.2 (Orthogonal and collinear vectors)**.** Two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ are *orthogonal* if $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$. They are *collinear* if there exists $\lambda \in \mathbb{R}$ such that $\boldsymbol{u} = \lambda \boldsymbol{v}$.

Remark 2.1 (Pythagoras)*.* If $\boldsymbol{u} \perp \boldsymbol{v}$, then $\|\boldsymbol{u} + \boldsymbol{v}\|^2 = \|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2$. This is the generalization of Pythagoras' theorem.

2.2 Classical inequalities

Theorem 2.1 (Cauchy–Schwarz). *For all $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$,*

$$|\langle \boldsymbol{u}, \boldsymbol{v} \rangle| \ \le \ \|\boldsymbol{u}\| \, \|\boldsymbol{v}\|,$$

with equality iff $\boldsymbol{u}$ and $\boldsymbol{v}$ are collinear.

Theorem 2.2 (Triangle inequality). *For all $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$,*

$$\|\boldsymbol{u} + \boldsymbol{v}\| \ \le \ \|\boldsymbol{u}\| + \|\boldsymbol{v}\|.$$

Sketch of proof. *By expansion, $\|\boldsymbol{u} + \boldsymbol{v}\|^2 = \|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2 + 2\langle \boldsymbol{u}, \boldsymbol{v} \rangle \le (\|\boldsymbol{u}\| + \|\boldsymbol{v}\|)^2$ via Cauchy–Schwarz, then take square roots.*

Remark 2.2 (Length of a vector (Pythagoras)). In the canonical basis, $\|\boldsymbol{x}\| = \sqrt{x_1^2 + \cdots + x_d^2}$: this is the "length" of $\boldsymbol{x}$, derived from Pythagoras' theorem in dimension d.

2.3 Affine subspaces, hyperplanes, and normal vectors

Definition 2.3 (Affine subspace). An *affine subspace* is a set of the form

$$\mathcal{A} = \boldsymbol{x}_0 + \mathcal{V} \ = \ \{\boldsymbol{x}_0 + \boldsymbol{v} : \boldsymbol{v} \in \mathcal{V}\},$$

where $\boldsymbol{x}_0 \in \mathbb{R}^d$ and $\mathcal{V}$ is a linear subspace.

Definition 2.4 (Hyperplane). A *hyperplane* in $\mathbb{R}^d$ is an affine subspace of dimension $d-1$, equivalently a set

$$H \ = \ \{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0\},$$

where $\boldsymbol{w} \in \mathbb{R}^d \setminus \{0\}$ and $b \in \mathbb{R}$. The vector $\boldsymbol{w}$ is a **normal vector** to H (orthogonal to every direction in H).

Remark 2.3. A hyperplane splits the space into two. This property underlies the classification rule used later.

Definition 2.5 (Normal line to a hyperplane through a point). Let $H = \{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0\}$ and $\boldsymbol{x}_0 \in \mathbb{R}^d$. The *normal line* to H passing through $\boldsymbol{x}_0$ is

$$\mathcal{N}(\boldsymbol{x}_0) \ = \ \{\boldsymbol{x}(t) = \boldsymbol{x}_0 + t\,\boldsymbol{w} : t \in \mathbb{R}\}.$$

2.4 Projections: scalar and vector

Definition 2.6 (Scalar projection). The *scalar projection* of $\boldsymbol{v}$ onto $\boldsymbol{u} \neq 0$ is

$$\operatorname{comp}_{\boldsymbol{u}}(\boldsymbol{v}) \ = \ \frac{\langle \boldsymbol{v}, \boldsymbol{u} \rangle}{\|\boldsymbol{u}\|}.$$

Definition 2.7 (Vector projection onto a direction). The *vector projection* of $\boldsymbol{v}$ onto the line spanned by $\boldsymbol{u} \neq 0$ is

$$\operatorname{proj}_{\boldsymbol{u}}(\boldsymbol{v}) \ = \ \frac{\langle \boldsymbol{v}, \boldsymbol{u} \rangle}{\|\boldsymbol{u}\|^2}\,\boldsymbol{u}.$$

Proposition 2.1 (Orthogonal projection onto a hyperplane). *Let $H = \{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0\}$ with $\boldsymbol{w} \neq 0$ and any $\boldsymbol{x}_0 \in \mathbb{R}^d$. The orthogonal projection $\Pi_H(\boldsymbol{x}_0)$ of $\boldsymbol{x}_0$ onto H is*

$$\Pi_H(\boldsymbol{x}_0) \ = \ \boldsymbol{x}_0 \ - \ \frac{\langle \boldsymbol{w}, \boldsymbol{x}_0 \rangle + b}{\|\boldsymbol{w}\|^2}\,\boldsymbol{w}.$$

Justification. *Subtract from $\boldsymbol{x}_0$ its component along the normal $\boldsymbol{w}$.*

2.5 Point–hyperplane distances (signed and unsigned)

Definition 2.8 (Point–hyperplane distance)**.** The (unsigned) distance from a point $\boldsymbol{x}_0$ to the hyperplane $H = \{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0\}$ is

$$\operatorname{dist}(\boldsymbol{x}_0, H) \;=\; \frac{|\langle \boldsymbol{w}, \boldsymbol{x}_0 \rangle + b|}{\|\boldsymbol{w}\|}.$$

Definition 2.9 (Signed distance)**.** Fixing the orientation by the normal $\boldsymbol{w}$, the *signed distance* is

$$\operatorname{sdist}_{\boldsymbol{w}}(\boldsymbol{x}_0, H) \;=\; \frac{\langle \boldsymbol{w}, \boldsymbol{x}_0 \rangle + b}{\|\boldsymbol{w}\|}.$$

It is positive on the side where $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b > 0$ and negative on the other.

Remark 2.4 (Distance between two parallel hyperplanes)*.* If $H_1 = \{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b_1 = 0\}$ and $H_2 = \{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b_2 = 0\}$ with the *same* normal $\boldsymbol{w}$, then

$$\operatorname{dist}(H_1, H_2) \;=\; \frac{|b_1 - b_2|}{\|\boldsymbol{w}\|}.$$

Proposition 2.2 (Orthogonal decomposition with respect to a hyperplane)**.** *For any $\boldsymbol{x} \in \mathbb{R}^d$,*

$$\boldsymbol{x} \;=\; \underbrace{\left(\boldsymbol{x} - \frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{\|\boldsymbol{w}\|^2}\,\boldsymbol{w}\right)}_{\text{projection onto } H} \;+\; \underbrace{\frac{\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b}{\|\boldsymbol{w}\|^2}\,\boldsymbol{w}}_{\text{normal component}}\;.$$

The two components are orthogonal.

3 Rosenblatt's Perceptron

We first recall the linear decision rule and the partition of the space induced by a hyperplane, then introduce the perceptron, its loss function, the associated optimization problem, and finally the algorithm with its classical convergence result.

Let

$$\mathcal{T} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$$

be our training data composed of N instances in a space $\mathcal{X} \subseteq \mathbb{R}^d$, such that $\forall i \in \{1, \dots, N\}$, $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$. The goal is to separate $\mathcal{T}$ into two classes using a hyperplane

$$h(\boldsymbol{x}) \;=\; \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b,$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is a normal vector to the hyperplane and $b \in \mathbb{R}$ is a bias.

This hyperplane induces two half-spaces

$$H^+ \;=\; \{\boldsymbol{x} \in \mathcal{X} :\; h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \geq 0\}, \qquad H^- \;=\; \{\boldsymbol{x} \in \mathcal{X} :\; h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b < 0\},$$

to which we associate classes +1 and −1, respectively; see Figure 1.

Definition 3.1 (Sign function and decision rule)**.** Let $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$ be the equation of a hyperplane in $\mathcal{X}$. Define $\operatorname{sign} : \mathbb{R} \to \{-1, 1\}$ by

$$\operatorname{sign}(t) = \begin{cases} 1, & \text{if } t \geq 0, \\ -1, & \text{if } t < 0. \end{cases}$$

For a new instance $\boldsymbol{x}_{\text{new}}$, the predicted class is

$$\widehat{y} \;=\; \operatorname{sign}(h(\boldsymbol{x}_{\text{new}})) \;=\; \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x}_{\text{new}} \rangle + b).$$

Remark. By convention, points $\boldsymbol{x}$ such that $h(\boldsymbol{x}) = 0$ (on the hyperplane) are assigned to class +1. If one wishes to distinguish this case, use $\operatorname{sign}_0 : \mathbb{R} \to \{-1, 0, 1\}$ with $\operatorname{sign}_0(0) = 0$.

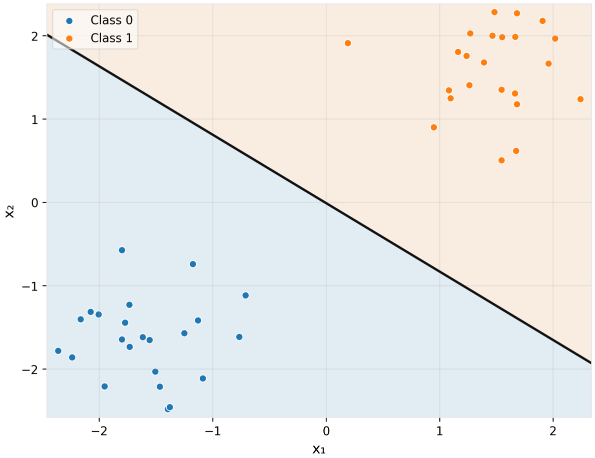


Figure 1 – Illustration of a separating hyperplane and the decision rule.

3.1 Losses and perceptron formulation

Guiding idea. The perceptron seeks a hyperplane that makes *all* signed margins $m_i = y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)$ positive. The perceptron loss looks only at misclassified (or on-the-boundary) points and pushes $(\boldsymbol{w}, b)$ in the direction that fixes the current error.

Definition 3.2 (Linear score and signed margin)**.** Given a hyperplane $h(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x}\rangle + b$, the *linear score* is $h(\boldsymbol{x})$. For $(\boldsymbol{x}_i, y_i)$ with $y_i \in \{-1, 1\}$, the *signed margin* is

$$m_i(\boldsymbol{w}, b) \;=\; y_i\, h(\boldsymbol{x}_i) \;=\; y_i\,(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b). \tag{3.1}$$

Then $m_i > 0$ means that $\boldsymbol{x}_i$ is correctly classified, $m_i < 0$ that it is misclassified, and $m_i = 0$ that it lies on the hyperplane.

Remark 3.1 (Why the product $y_i\, h(\boldsymbol{x}_i)$?)*.* Without the factor y_i, the sign of $h(\boldsymbol{x}_i)$ is interpreted differently across classes. Multiplying by $y_i \in \{-1, 1\}$ *unifies* the correct-classification condition:

$$(\boldsymbol{x}_i, y_i) \text{ correctly classified} \quad \Longleftrightarrow \quad y_i\, h(\boldsymbol{x}_i) > 0.$$

This single expression simplifies modeling and updates.

Definition 3.3 (Perceptron loss)**.** The *perceptron loss* penalizes only the misclassified (or zero-margin) observations:

$$\ell(z) = \max(0, -z), \qquad L(\boldsymbol{w}, b) = \sum_{i=1}^{N} \ell\big(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)\big) = \sum_{i=1}^{N} \max\big(0, -y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)\big). \tag{3.2}$$

Proposition 3.1 (Convex optimization problem)**.** *The perceptron solves*

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\, b\in\mathbb{R}} L(\boldsymbol{w}, b) = \min_{\boldsymbol{w}, b} \sum_{i=1}^{N} \max\big(0, -y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)\big). \tag{3.3}$$

The function L is convex in $(\boldsymbol{w}, b)$ (a maximum of affine functions), but non-differentiable when $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) = 0$.

Proposition 3.2 (Subgradients)**.** *Writing $h_i = \langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b$ and $\mathcal{M}(\boldsymbol{w}, b) = \{i : \; y_i h_i < 0\}$ for the error set, a subgradient of L at $(\boldsymbol{w}, b)$ is*

$$\frac{\partial L(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = -\sum_{i\in\mathcal{M}(\boldsymbol{w},b)} y_i\, \boldsymbol{x}_i, \qquad \frac{\partial L(\boldsymbol{w}, b)}{\partial b} = -\sum_{i\in\mathcal{M}(\boldsymbol{w},b)} y_i. \tag{3.4}$$

At the non-differentiable point $y_i h_i = 0$, any vector between $\mathbf{0}$ and $-y_i\boldsymbol{x}_i$ is a valid subgradient for the ith term.

3.2 Algorithm and convergence

The perceptron algorithm is an *online* subgradient descent: we loop over examples and update only upon error (or zero margin). The augmented notation folds the bias into the weight vector.

Remark 3.2 (Augmented variables)*.* Let

$$\tilde{\boldsymbol{x}}_i = \begin{pmatrix} \boldsymbol{x}_i \\ 1 \end{pmatrix} \in \mathbb{R}^{d+1}, \qquad \tilde{\boldsymbol{w}} = \begin{pmatrix} \boldsymbol{w} \\ b \end{pmatrix} \in \mathbb{R}^{d+1},$$

so that $h(\boldsymbol{x}_i) = \langle \tilde{\boldsymbol{w}}, \tilde{\boldsymbol{x}}_i \rangle$ and updates can be written compactly.

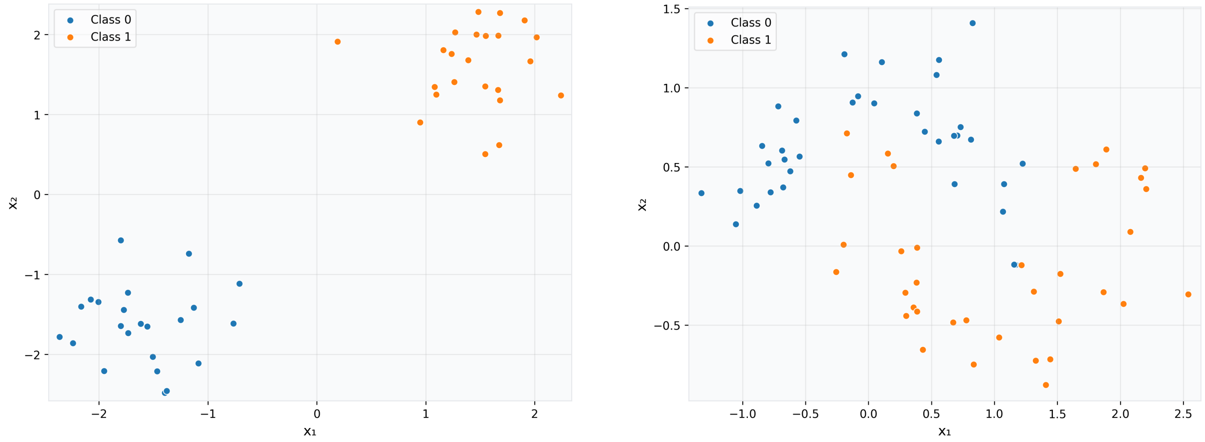
Algorithm 1 Perceptron algorithm (online subgradient descent)

```
Require: Data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, step size $\eta > 0$, iterations $T$
Initialize $\tilde{\boldsymbol{w}}^{(0)} = \boldsymbol{0} \in \mathbb{R}^{d+1}$        ▷ or a small random value
for $t = 0, 1, \dots, T-1$ do
    for $i = 1$ to $N$ do
        Compute $s_i = y_i \langle \tilde{\boldsymbol{w}}^{(t)}, \tilde{\boldsymbol{x}}_i \rangle$
        if $s_i \le 0$ then        ▷ error or zero margin
            $\tilde{\boldsymbol{w}}^{(t)} \leftarrow \tilde{\boldsymbol{w}}^{(t)} + \eta\, y_i\, \tilde{\boldsymbol{x}}_i$
        end if
    end for
end for
return $\tilde{\boldsymbol{w}}^{(T)}$ (hence $\boldsymbol{w}$ and $b$)
```

Definition 3.4 (Linearly separable data)**.** A labeled set $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ is said to be *linearly separable* if there exist $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ (and, equivalently, a margin $M > 0$ after rescaling $(\boldsymbol{w}, b)$) such that

$$y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \right) \ \ge\ M \ >\ 0, \qquad \forall i = 1, \dots, N. \tag{3.5}$$

In other words, the two classes are strictly separated by the hyperplane $\{\boldsymbol{x} : \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0\}$.



(a) Linearly separable data

(b) Non–linearly separable data

Figure 2 – Visual comparison between separable and non separable cases.

Theorem 3.1 (Novikoff (1962), mistake bound)**.** *Assume* $\|\boldsymbol{x}_i\| \le R$ *for all* i *and that the data are linearly separable with geometric margin* $M > 0$*. Then the perceptron algorithm makes at most* $(R/M)^2$ *updates (mistakes) and therefore stops in a finite number of steps.*

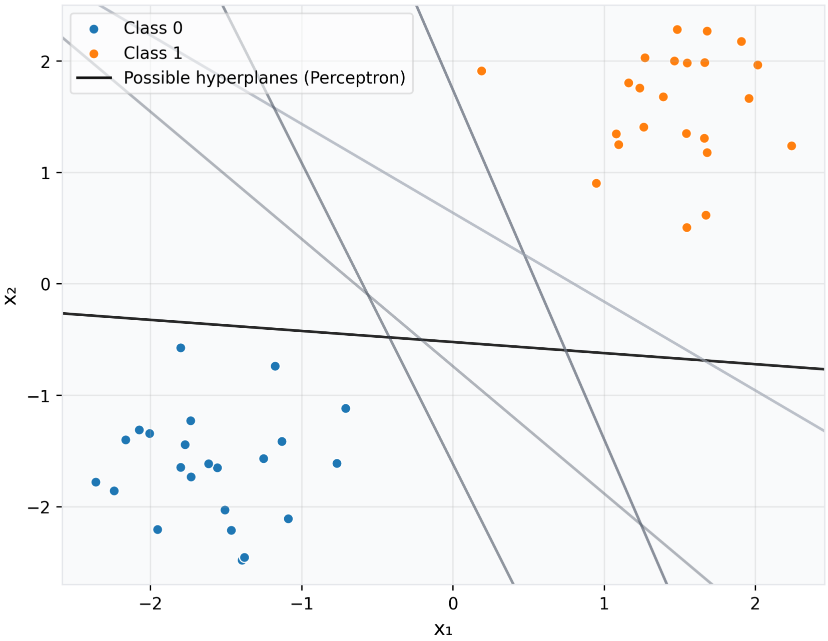


Figure 3 – Examples of separating hyperplanes obtained by the perceptron depending on initialization and presentation order: the solution is not unique under separability.

3.3 Limitations

The perceptron algorithm nevertheless exhibits several **limitations** (see, e.g., Ripley (1996)):

- **Non-uniqueness under separability.** When the data are separable, there are generally *infinitely many* separating hyperplanes (see Figure 3). The perceptron may converge to different solutions depending on the *initialization* and the *order* of examples.
- **No convergence on non-separable data.** When the data are not linearly separable, the algorithm *does not converge* and may enter *cycles* (oscillating updates), especially in the presence of label noise or outliers.
- **Potentially long convergence time.** Theorem 3.1 shows that the number of perceptron updates is bounded by $(R/M)^2$. Intuitively, the smaller the margin M (points "skimming" the hyperplane), the more corrections may be needed before stabilization. Thus, *the smaller the margin* (data "barely" separable), *the slower the convergence can be.*

We will later see how the initial problem in (3.3) can be modified to address these limitations.

4 Support Vector Classifier (Hard-Margin)

In the literature, several names refer to this model: Support Vector Machines with a hard margin (hard-margin SVM), Optimal Separating Hyperplanes (Hastie et al. (2009)), and Support Vector Classifier (SVC) with a hard margin. Hereafter, we refer to this model as the hard-margin SVC. We present the *large-margin* intuition and the logic that naturally leads from the "maximize the margin" problem to the convex formulation "minimize $\frac{1}{2}\|\boldsymbol{w}\|^2$" under separation constraints. We then give the primal problem, its Lagrange dual, the KKT conditions, and the primal–dual equivalence.

4.1 Large-margin principle and hard-margin SVC formulation

The perceptron puts each point on the "correct side" of a hyperplane without caring about its distance to the boundary. Yet, as soon as the data are separable, there are infinitely many valid hyperplanes; the solution then depends on initialization and example order, and can be

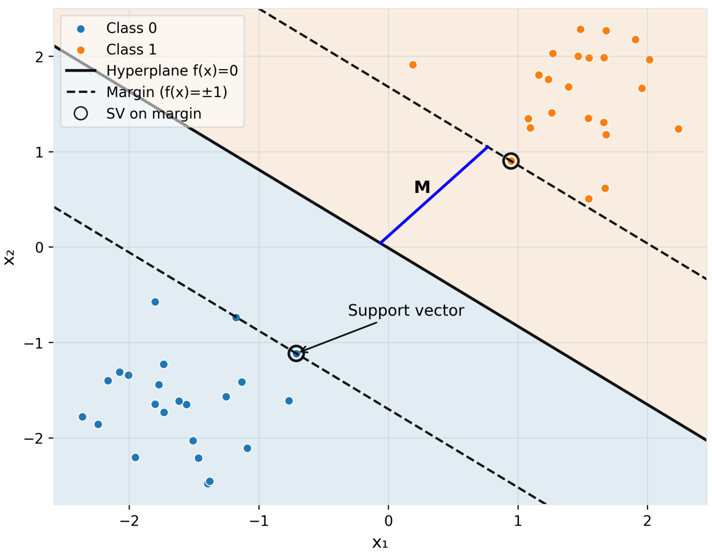


Figure 4 – Geometric margin in a hard-margin SVC: the band delimited by $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = \pm 1$ has a margin $M = 1/\|\boldsymbol{w}\|$ and a band width $2/\|\boldsymbol{w}\|$, and the points in contact are the support vectors.

unstable to small perturbations. The large-margin idea is to prefer, among all correct separators, the one that leaves the largest "safety cushion" around the decision boundary.

For a pair $(\boldsymbol{w}, b)$ and an example $(\boldsymbol{x}_i, y_i)$, define

$$\widehat{M}_i(\boldsymbol{w}, b) := y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right)$$

as the example's *functional margin*: it is positive if the example is correctly classified and larger the more comfortably it lies on the correct side. Aggregating via the minimum,

$$\widehat{M}(\boldsymbol{w}, b) := \min_i \widehat{M}_i(\boldsymbol{w}, b),$$

but this quantity depends on scale, since $(\lambda \boldsymbol{w}, \lambda b)$ describes the same hyperplane while multiplying $\widehat{M}$ by $\lambda > 0$.

We remove the scale dependence by dividing by the norm of the normal vector: the *geometric margin*

$$M(\boldsymbol{w}, b) := \min_i \frac{\widehat{M}_i(\boldsymbol{w}, b)}{\|\boldsymbol{w}\|} = \min_i \frac{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)}{\|\boldsymbol{w}\|}$$

coincides with the minimal signed distance of the points to the hyperplane and depends only on the geometric position of the boundary. This is the quantity we want to maximize for a robust separator; Figure 4 illustrates this.

Since $(\boldsymbol{w}, b)$ and $(\lambda \boldsymbol{w}, \lambda b)$ represent the same hyperplane, we fix the scale by the *canonical normalization*

$$\min_i y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) = 1.$$

Support vectors satisfy $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) = 1$, the margin is $M(\boldsymbol{w}, b) = 1/\|\boldsymbol{w}\|$ and the width of the margin band (between $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = \pm 1$) is $2/\|\boldsymbol{w}\|$.

With the canonical normalization above,

$$\max_{\boldsymbol{w}, b} \; M(\boldsymbol{w}, b) \quad \Longleftrightarrow \quad \max_{\boldsymbol{w}, b} \; \frac{1}{\|\boldsymbol{w}\|} \quad \text{s.t.} \quad y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 \;\; \forall i.$$

Maximizing $1/\|\boldsymbol{w}\|$ is equivalent to *minimizing* $\|\boldsymbol{w}\|$, and for analytic convenience we use the convex quadratic objective $\frac{1}{2}\|\boldsymbol{w}\|^2$.

4.2 Primal formulation, dual derivation (hard-margin)

Under the canonical normalization $\min_i y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) = 1$, maximizing the geometric margin amounts to minimizing $\|\boldsymbol{w}\|$. We obtain a convex quadratic program with affine constraints.

Definition 4.1 (Hard-margin SVM — primal formulation)**.**

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\big) \ \geq \ 1, \qquad i = 1, \dots, N. \end{aligned} \tag{4.1}$$

Comment on the primal. The objective is strictly convex and the constraints are linear. Under strict separability, there exists at least one primal feasible solution. The direction of $\boldsymbol{w}^\star$ is then unique in general position, and the bias $b^\star$ follows from the support vectors.

From primal to dual: construction and dual function. For each constraint $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1$, introduce a Lagrange multiplier $\lambda_i \geq 0$. The Lagrangian associated with (4.1) is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \lambda_i \Big(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \Big), \quad \text{with } \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)^\top,\ \lambda_i \geq 0.$$

Define the dual function $g(\boldsymbol{\lambda})$ as the infimum of $\mathcal{L}$ with respect to the primal variables:

$$g(\boldsymbol{\lambda}) = \inf_{\boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}).$$

Minimize first over $\boldsymbol{w}$. The term in $\boldsymbol{w}$ is

$$\frac{1}{2}\|\boldsymbol{w}\|^2 \ - \ \Big\langle \boldsymbol{w},\ \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \Big\rangle,$$

a strictly convex quadratic form in $\boldsymbol{w}$. Its unique minimizer is given by the first-order condition

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \quad \boldsymbol{w}^\star = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i.$$

Replacing $\boldsymbol{w}$ with $\boldsymbol{w}^\star$, the contribution in $\boldsymbol{w}$ becomes

$$\frac{1}{2}\|\boldsymbol{w}^\star\|^2 - \Big\langle \boldsymbol{w}^\star,\ \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \Big\rangle = \frac{1}{2}\|\boldsymbol{w}^\star\|^2 - \|\boldsymbol{w}^\star\|^2 = -\frac{1}{2}\|\boldsymbol{w}^\star\|^2 = -\frac{1}{2}\Big\| \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \Big\|^2.$$

Next minimize over b. The Lagrangian is affine in b via the term $-b\sum_{i=1}^{N} \lambda_i y_i$. If $\sum_i \lambda_i y_i \neq 0$, the infimum in b is $-\infty$. To make $g(\boldsymbol{\lambda})$ finite, we therefore impose the dual equality constraint

$$\sum_{i=1}^{N} \lambda_i y_i = 0.$$

In that case the dependence on b vanishes, yielding

$$g(\boldsymbol{\lambda}) = -\frac{1}{2}\Big\| \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i \Big\|^2 + \sum_{i=1}^{N} \lambda_i.$$

Expanding the squared norm gives the classical bilinear form

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \lambda_i \ - \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_i \lambda_j \, y_i y_j \, \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle.$$

Proposition 4.1 (Hard-margin SVM — dual formulation)**.**

$$\begin{aligned}
\max_{\boldsymbol{\lambda}\in\mathbb{R}^N} \quad & \sum_{i=1}^N \lambda_i - \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \lambda_i\lambda_j\, y_i y_j\, \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\
s.t. \quad & \lambda_i \geq 0, \quad i = 1, \dots, N, \\
& \sum_{i=1}^N \lambda_i y_i = 0.
\end{aligned} \tag{4.2}$$

Comment on the dual. The dual is a concave quadratic program over a polyhedron (cone $\lambda_i \geq 0$ and hyperplane $\sum_i \lambda_i y_i = 0$). At the optimum, the coefficients $\lambda_i^\star$ are zero for points that do not influence the boundary and strictly positive for points that "support" the hyperplane—hence the term *support vectors.*

Proposition 4.2 (Slater's condition for the hard-margin SVC)**.** *Assume the data are linearly separable, i.e., there exists* $(\boldsymbol{w}_0, b_0)$ *such that*

$$y_i\big(\langle \boldsymbol{w}_0, \boldsymbol{x}_i\rangle + b_0\big) > 0, \qquad i = 1, \dots, N.$$

Then there exists $(\boldsymbol{w}, b)$ *such that*

$$y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) > 1, \qquad i = 1, \dots, N.$$

In particular, for the primal problem (4.1)*, Slater's condition holds. Since the problem is convex with affine constraints, strong duality holds: the optimal primal value matches the optimal dual value, and the KKT conditions characterize optimality.*

Sketch of proof. By separability, define $m := \min_i y_i\big(\langle \boldsymbol{w}_0, \boldsymbol{x}_i\rangle + b_0\big) > 0$. For any $c > 1/m$, set $\boldsymbol{w} = c\,\boldsymbol{w}_0$ and $b = c\,b_0$. Then, for all i,

$$y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) = c\,y_i\big(\langle \boldsymbol{w}_0, \boldsymbol{x}_i\rangle + b_0\big) \geq c\,m \ > \ 1.$$

We have thus exhibited a strictly feasible primal point. By Slater's condition (for affine inequalities in a convex problem), strong duality follows and the KKT conditions are necessary and sufficient. □

Definition 4.2 (Karush–Kuhn–Tucker (KKT) conditions)**.** In a convex problem satisfying Slater's condition, a triplet $(\boldsymbol{w}^\star, b^\star, \boldsymbol{\lambda}^\star)$ is optimal iff the following four families of conditions hold:

1. *Primal feasibility*: for all i, $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) \geq 1$.
2. *Dual feasibility*: for all i, $\lambda_i^\star \geq 0$, and $\sum_{i=1}^N \lambda_i^\star y_i = 0$.
3. *Stationarity*: $\boldsymbol{w}^\star = \sum_{i=1}^N \lambda_i^\star y_i \boldsymbol{x}_i$. The condition $\sum_i \lambda_i^\star y_i = 0$ comes from minimization with respect to b.
4. *Complementary slackness*: for all i,

$$\lambda_i^\star\Big(y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) - 1\Big) = 0.$$

Complementary slackness reads constraint-by-constraint. For a given index i, the product of the multiplier $\lambda_i^\star$ and the "at-threshold" functional margin $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) - 1$ is zero. Two exclusive cases occur.

— If $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) > 1$, then $\lambda_i^\star = 0$. Point i lies strictly outside the margin band; it does not contribute to $\boldsymbol{w}^\star$ in the decomposition $\boldsymbol{w}^\star = \sum_j \lambda_j^\star y_j \boldsymbol{x}_j$.

— If $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) = 1$, the constraint is active; typically $\lambda_i^\star > 0$. Point i is on the margin: a *support vector*. Only these active points determine $\boldsymbol{w}^\star$.

Reconstructing the classifier from the dual. Once optimal $\boldsymbol{\lambda}^\star$ are computed, reconstruct the normal $\boldsymbol{w}^\star = \sum_i \lambda_i^\star y_i \boldsymbol{x}_i$. To determine the bias, use any point k such that $\lambda_k^\star > 0$[1] (hence $y_k(\langle \boldsymbol{w}^\star, \boldsymbol{x}_k\rangle + b^\star) = 1$) and set

$$b^\star = y_k - \langle \boldsymbol{w}^\star, \boldsymbol{x}_k\rangle.$$

In practice, average this value over several support vectors to reduce numerical noise.

Primal–dual equality and optimal value. Under Slater's condition, the duality gap is zero. The optimal primal value equals the optimal dual value. In particular,

$$\frac{1}{2}\|\boldsymbol{w}^\star\|^2 = \sum_{i=1}^N \lambda_i^\star \; - \; \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \lambda_i^\star\lambda_j^\star \, y_i y_j \, \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle.$$

Further reading. For the basics of convex duality, Slater's condition, and KKT, see for instance Boyd and Vandenberghe (2004).

5 Support Vector Classifier (Soft-Margin)

The hard-margin model assumes perfect separability: all observations can be held at distance at least $1/\|\boldsymbol{w}\|$ from the hyperplane. In many real-world situations, the data are only *almost separable"*: there is noise, a few outliers, and sometimes slightly overlapping classes. It is then desirable to allow *controlled violations* of the margin rather than forcing an unrealistic separation.

5.1 From $\max M$ to soft-margin: intuition and convexity

Starting point. In the hard-margin case, we pose the problem directly as a maximization of the geometric margin M, fixing the scale with $\|\boldsymbol{w}\| = 1$ (unit normal):

$$\max_{\boldsymbol{w},b,M} \; M \qquad \text{s.t.} \qquad y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) \;\ge\; M, \quad i = 1, \dots, N, \quad \|\boldsymbol{w}\| = 1.$$

This means: with a unit normal, the quantity $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b)$ is exactly the signed distance from $\boldsymbol{x}_i$ to the hyperplane, and M is the smallest of these distances.

Almost separable" case. When classes overlap or noise is present, we keep the idea "maximize M" but allow controlled violations via slack variables $\xi_i \ge 0$ (one per point), see Figure 5, while penalizing their sum. Two natural ways to relax the margin constraint exist. *Additive slack.* Allow an *absolute deficit* ξ_i relative to the target M:

$$y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) \;\ge\; M - \xi_i, \qquad \xi_i \ge 0, \qquad \max \; M - \alpha\sum_i \xi_i, \qquad \|\boldsymbol{w}\| = 1.$$

Intuition: ξ_i is expressed "in meters" (same unit as M), so each point may encroach by some *distance* on the global margin. Non-convexity. To see the issue, eliminate M by working at free scale. Let

$$\tilde{\boldsymbol{w}} = \frac{\boldsymbol{w}}{M}, \qquad \tilde{b} = \frac{b}{M}, \qquad \tilde{\xi}_i = \frac{\xi_i}{M} \quad (M > 0).$$

The constraint becomes

$$y_i\big(\langle \tilde{\boldsymbol{w}}, \boldsymbol{x}_i\rangle + \tilde{b}\big) \;\ge\; 1 - \tilde{\xi}_i,$$

which is *affine* in $(\tilde{\boldsymbol{w}}, \tilde{b}, \tilde{\xi})$. However, the objective rewrites as

$$M - \alpha\sum_i \xi_i \;=\; \frac{1}{\|\tilde{\boldsymbol{w}}\|} \;-\; \alpha\sum_i \frac{\tilde{\xi}_i}{\|\tilde{\boldsymbol{w}}\|} \;=\; \frac{1 - \alpha\sum_i \tilde{\xi}_i}{\|\tilde{\boldsymbol{w}}\|},$$

[1] The attentive reader will note that k is therefore a support vector

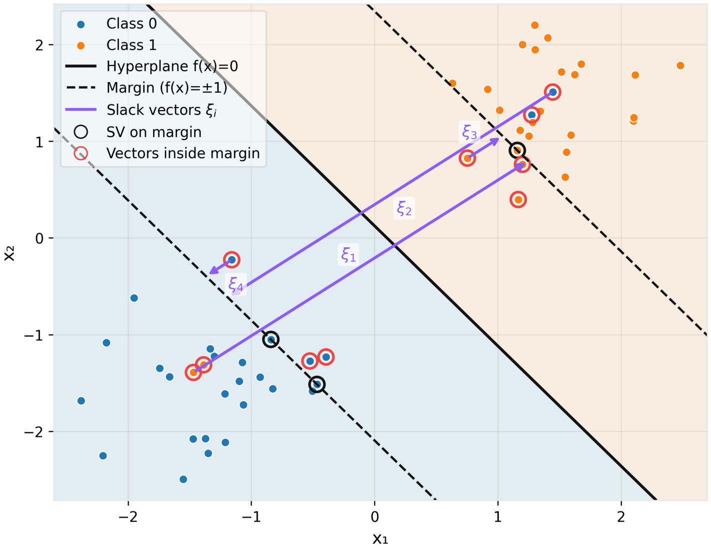


Figure 5 – Soft-margin with slack variables ξ_i: $0 < \xi_i < 1$ for a point inside the band but correctly classified, $\xi_i \geq 1$ for a misclassified point.

which couples $\tilde{\boldsymbol{w}}$ and $\tilde{\xi}$ nonlinearly (a ratio linear/$\|\tilde{\boldsymbol{w}}\|$). This is neither concave (for maximization) nor easily convexified: the problem is *non-convex*. This is the difficulty noted in Hastie et al. (2009) for the "additive" choice.

Multiplicative slack. Require each point to meet a *fraction* $(1 - \xi_i)$ of the global margin:

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \;\geq\; M\,(1 - \xi_i), \qquad \xi_i \geq 0, \qquad \max\; M - \alpha \sum_i \xi_i, \qquad \|\boldsymbol{w}\| = 1.$$

Intuition: ξ_i is a *percentage* of "tolerance"; e.g., $\xi_i = 0.1$ allows point i to lie at 90% of the target margin. Towards a convex form. Divide again by M and set

$$\tilde{\boldsymbol{w}} = \frac{\boldsymbol{w}}{M}, \qquad \tilde{b} = \frac{b}{M}.$$

We obtain the same affine constraint as above:

$$y_i(\langle \tilde{\boldsymbol{w}}, \boldsymbol{x}_i \rangle + \tilde{b}) \;\geq\; 1 - \xi_i,$$

but now ξ_i is *not* rescaled. Moreover, $\|\boldsymbol{w}\| = 1$ implies $M = 1/\|\tilde{\boldsymbol{w}}\|$, so maximizing $M - \alpha \sum_i \xi_i$ amounts to maximizing

$$\frac{1}{\|\tilde{\boldsymbol{w}}\|} \;-\; \alpha \sum_i \xi_i.$$

Rather than maximizing $1/\|\tilde{\boldsymbol{w}}\|$ (non-convex), we adopt the convex quadratic objective $\frac{1}{2}\|\tilde{\boldsymbol{w}}\|^2$, which is *monotone* with respect to $1/\|\tilde{\boldsymbol{w}}\|$ (these criteria rank solutions the same way after choosing a parameter). This leads to the standard SVC:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi} \geq 0} \; \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \xi_i \qquad \text{s.t.} \qquad y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \;\geq\; 1 - \xi_i, \quad i = 1, \ldots, N. \tag{5.1}$$

Here everything is convex: a strictly convex quadratic objective in $\boldsymbol{w}$, affine constraints, domain $\xi_i \geq 0$. The parameter $C > 0$ plays the same role as α after rescaling and controls the trade-off between margin width and the amount of violations.

Geometric reading. With the constraint $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i$, we recover the same margin band as in the hard-margin case: $H_\pm$: $\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = \pm 1$. A point has $\xi_i = 0$ if it is outside the band (or on ± 1); it has $0 < \xi_i < 1$ if it lies inside the band but on the correct side; it has $\xi_i \geq 1$ if it is misclassified. Increasing C reduces violations but shrinks the margin; decreasing C widens the margin but accepts more errors.

5.2 Primal formulation, dual derivation (soft-margin)

At the canonical scale (target margin fixed at 1), allowing controlled margin violations introduces slack variables $\xi_i \geq 0$ and penalizes their sum. We obtain a convex quadratic program with affine constraints.

Definition 5.1 (Soft-margin SVM — primal formulation)**.**

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d,\ b\in\mathbb{R},\ \boldsymbol{\xi}\in\mathbb{R}^N_+} \quad & \frac{1}{2}\,\|\boldsymbol{w}\|^2 \;+\; C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b\big) \;\geq\; 1-\xi_i, \qquad i=1,\dots,N, \\ & \xi_i \;\geq\; 0, \qquad i=1,\dots,N. \end{aligned} \tag{5.2}$$

Remark 5.1 (Comment on the primal). The objective is strictly convex in $\boldsymbol{w}$; the constraints are linear. The parameter $C > 0$ controls the trade-off between margin width (via $\|\boldsymbol{w}\|$) and total violation (via $\sum_i \xi_i$). Geometrically, $\xi_i = 0$ means a point outside the band (or on the margin), $0 < \xi_i < 1$ a point inside the band but on the correct side, and $\xi_i \geq 1$ a misclassified point.

From primal to dual: construction and dual function. For the constraints $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1 + \xi_i \geq 0$, introduce multipliers $\lambda_i \geq 0$; for $\xi_i \geq 0$, multipliers $\nu_i \geq 0$. The Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\lambda_i\Big(y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1 + \xi_i\Big) - \sum_{i=1}^{N}\nu_i\,\xi_i.$$

The dual function $g(\boldsymbol{\lambda}) = \inf_{\boldsymbol{w},b,\boldsymbol{\xi}} \mathcal{L}$ follows by minimizing successively:

- In $\boldsymbol{w}$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \implies \boldsymbol{w}^\star = \sum_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i, \qquad \inf_{\boldsymbol{w}}\Big(\tfrac{1}{2}\|\boldsymbol{w}\|^2 - \langle \boldsymbol{w}, \sum_i \lambda_i y_i \boldsymbol{x}_i\rangle\Big) = -\tfrac{1}{2}\Big\|\sum_i \lambda_i y_i \boldsymbol{x}_i\Big\|^2.$$

- In b:

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{N}\lambda_i y_i = 0 \implies \sum_{i=1}^{N}\lambda_i y_i = 0,$$

otherwise the infimum in b equals $-\infty$.

- In ξ_i:

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = C - \lambda_i - \nu_i = 0 \implies \nu_i = C - \lambda_i \;\; (\geq 0) \implies \boxed{0 \leq \lambda_i \leq C}.$$

The dependence on $\boldsymbol{\xi}$ then disappears. We obtain

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{N}\lambda_i \;-\; \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_j\, y_i y_j\,\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle,$$

valid under $\sum_i \lambda_i y_i = 0$ and $0 \leq \lambda_i \leq C$.

Proposition 5.1 (Soft-margin SVM — dual formulation)**.**

$$\begin{aligned} \max_{\boldsymbol{\lambda}\in\mathbb{R}^N} \quad & \sum_{i=1}^{N}\lambda_i \;-\; \frac{1}{2}\,\boldsymbol{\lambda}^\top (YKY)\boldsymbol{\lambda} \\ s.t. \quad & 0 \leq \lambda_i \leq C, \quad i=1,\dots,N, \\ & \boldsymbol{y}^\top\boldsymbol{\lambda} = 0. \end{aligned} \tag{5.3}$$

Here $K \in \mathbb{R}^{N\times N}$ is the Gram matrix, $K_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$, and $Y = \operatorname{diag}(y_1, \dots, y_N)$. The dual thus depends on the data only through inner products $\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$—the key structure for the kernel trick.

Remark 5.2 (Reading the dual)*.* This is a concave quadratic program over the polyhedron $\{\, \boldsymbol{\lambda} : \ 0 \le \lambda_i \le C, \ \boldsymbol{y}^\top \boldsymbol{\lambda} = 0 \,\}$. At the optimum, only a few components $\lambda_i^\star$ are nonzero: active indices correspond to support vectors.

Proposition 5.2 (Slater's condition for the soft-margin SVC)**.** *For any dataset, there exists a strictly feasible primal point. For example,* $\boldsymbol{w} = \boldsymbol{0}$*,* $b = 0$*,* $\xi_i = 2$ *satisfy* $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + b) - 1 + \xi_i = 1 > 0$ *and* $\xi_i > 0$ *for all* i*. Slater's condition is therefore satisfied; strong duality holds and the KKT conditions characterize optimality.*

Definition 5.2 (Karush–Kuhn–Tucker (KKT) conditions — soft margin)**.** A quintuplet $(\boldsymbol{w}^\star, b^\star, \boldsymbol{\xi}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\nu}^\star)$ is optimal iff:

1. *Primal feasibility*: $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) \ge 1 - \xi_i^\star$ and $\xi_i^\star \ge 0$ for all i;
2. *Dual feasibility*: $\lambda_i^\star \ge 0$, $\nu_i^\star \ge 0$ for all i, and $\sum_i \lambda_i^\star y_i = 0$;
3. *Stationarity*: $\boldsymbol{w}^\star = \sum_i \lambda_i^\star y_i \boldsymbol{x}_i$ and $C - \lambda_i^\star - \nu_i^\star = 0$ for all i;
4. *Complementary slackness*: for all i,

$$\lambda_i^\star\Big(y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) - 1 + \xi_i^\star\Big) = 0, \qquad \nu_i^\star\, \xi_i^\star = 0.$$

Interpretation. If $0 < \lambda_i^\star < C$, then $\nu_i^\star > 0$ so $\xi_i^\star = 0$, and the constraint is active: $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) = 1$ (a "free" support vector, on the margin). If $\lambda_i^\star = C$, then $\nu_i^\star = 0$ and $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) = 1 - \xi_i^\star \le 1$ (a "bound" support vector, inside the band or misclassified if $\xi_i^\star \ge 1$). If $\lambda_i^\star = 0$, then $\nu_i^\star = C$ and $\xi_i^\star = 0$, hence $y_i(\langle \boldsymbol{w}^\star, \boldsymbol{x}_i\rangle + b^\star) \ge 1$ (outside the band, non-support).

Reconstructing the classifier from the dual. We have $\boldsymbol{w}^\star = \sum_{i=1}^N \lambda_i^\star y_i \boldsymbol{x}_i$. For the bias, choose k such that $0 < \lambda_k^\star < C$ (a "free" support) and set

$$b^\star = y_k - \langle \boldsymbol{w}^\star, \boldsymbol{x}_k\rangle,$$

then average over several "free" supports for numerical stability. The decision function is

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^N \lambda_i^\star y_i \,\langle \boldsymbol{x}_i, \boldsymbol{x}\rangle + b^\star\right).$$

Primal–dual equality and optimal value. Under Slater's condition, the duality gap is zero; in particular,

$$\frac{1}{2}\|\boldsymbol{w}^\star\|^2 + C\sum_{i=1}^N \xi_i^\star = \sum_{i=1}^N \lambda_i^\star \ - \ \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \lambda_i^\star \lambda_j^\star\, y_i y_j \,\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle.$$

Further reading. For convex duality (Slater's condition, KKT), see e.g. Boyd and Vandenberghe (2004). A didactic presentation of the soft-margin SVM (primal/dual, *hinge* loss) is given by Hastie et al. (2009). For the origin and a more theoretical treatment, see Cortes and Vapnik (1995) and Vapnik (1998).

6 Support Vector Machine (SVM): Kernel Trick

Extending the soft-margin SVC to nonlinear decision boundaries consists in replacing the Euclidean inner product $\langle x, x'\rangle$ by a *kernel* $K(x, x')$ that implicitly induces an embedding ϕ into a space (often high- or even infinite-dimensional) where the separation becomes linear again. This idea is supported by Cover (1965): *"Projecting data nonlinearly into a higher-dimensional space generally increases the probability of linear separability."* In the feature space, the classifier's complexity is controlled by the norm of the normal vector (or, in an RKHS, by $\|f\|_{\mathcal{H}_K}$), while the parameter C preserves regularization of the margin/violation trade-off. The foundational works of Boser et al. (1992); Cortes and Vapnik (1995) formalized this framework, which we detail below.

6.1 Intuition and examples of kernels

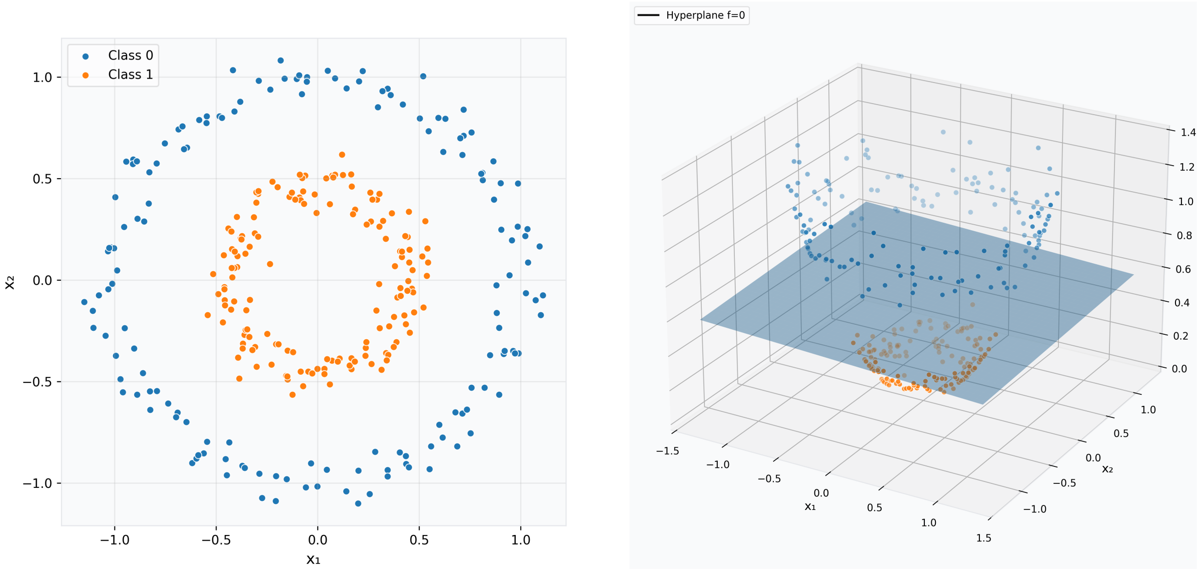
Starting point: to make a linear separation possible when boundaries are nonlinear in $\mathbb{R}^d$, consider an *embedding* $\phi : \mathbb{R}^d \to \mathcal{H}$ into a Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ where the data become (almost) separable. The soft-margin SVC "in $\mathcal{H}$" reads

$$\min_{\boldsymbol{w}\in\mathcal{H},\ b\in\mathbb{R},\ \boldsymbol{\xi}\geq 0} \ \frac{1}{2}\|\boldsymbol{w}\|_{\mathcal{H}}^2 + C\sum_{i=1}^{N}\xi_i \quad \text{s.t.} \quad y_i\big(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} + b\big) \geq 1 - \xi_i.$$

Cover (1965) formalizes the intuition: a nonlinear embedding into a higher-dimensional space generally increases the probability of linear separability (Figure 6). *However*, estimating $\boldsymbol{w} \in \mathcal{H}$ directly quickly becomes impractical: for a Gaussian kernel, $\mathcal{H}$ is *infinite*-dimensional; even for a *polynomial* kernel of degree p, the embedding dimension (all monomials up to degree p) grows combinatorially with d. The kernel trick circumvents this difficulty: one never needs to construct ϕ nor store $\boldsymbol{w}$ in $\mathcal{H}$, because the SVC *dual* depends on the data only through inner products of the form

$$\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}} \ =: \ K(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

where K is a *kernel* function. By replacing every $\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$ with $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the optimization is solved as if we were working linearly in $\mathcal{H}$, *without* ever computing ϕ.



(a) Data in $\mathbb{R}^2$ not linearly separable

(b) Data in $\mathbb{R}^3$ linearly separable

Figure 6 – Embedding $\mathbb{R}^2$ into $\mathbb{R}^3$

Informal definition: a *kernel* $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a function for which there exist a Hilbert space $\mathcal{H}$ and a feature map ϕ satisfying

$$K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}')\rangle_{\mathcal{H}}.$$

The learned decision then takes the form

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{N} \lambda_i^\star y_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) \ + \ b^\star\right),$$

where only the *support vectors* (indices i with $\lambda_i^\star > 0$) appear: neither $\boldsymbol{w}$ nor $\phi(\boldsymbol{x})$ are explicitly manipulated.

Some common kernels:

— **Linear**: $K(\boldsymbol{x},\boldsymbol{x}') = \langle \boldsymbol{x},\boldsymbol{x}'\rangle$ (baseline case, identity ϕ).

— **Polynomial (degree p)**: $K(\boldsymbol{x},\boldsymbol{x}') = (\langle \boldsymbol{x},\boldsymbol{x}'\rangle + c)^p$, $p \in \mathbb{N}$, $c \geq 0$ (finite embedding: monomials up to degree p).

— **Gaussian / RBF**: $K(\boldsymbol{x},\boldsymbol{x}') = \exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}'\|^2)$, $\gamma > 0$ (an *infinite*-dimensional embedding).

— **Sigmoid (neural-network type)**: $K(\boldsymbol{x},\boldsymbol{x}') = \tanh(\kappa\,\langle \boldsymbol{x},\boldsymbol{x}'\rangle + \theta)$ (p.s.d. only for certain parameter ranges).

Explicit example — quadratic polynomial kernel with $c = 1$ in dimension 2. Let $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{x}' = (x_1', x_2') \in \mathbb{R}^2$, and

$$K(\boldsymbol{x},\boldsymbol{x}') = (\langle \boldsymbol{x},\boldsymbol{x}'\rangle + 1)^2 = \left(x_1x_1' + x_2x_2' + 1\right)^2.$$

Expanding,

$$(\langle \boldsymbol{x},\boldsymbol{x}'\rangle+1)^2 = (x_1x_1'+x_2x_2')^2+2(x_1x_1'+x_2x_2')+1 = x_1^2x_1'^2+2x_1x_2x_1'x_2'+x_2^2x_2'^2+2x_1x_1'+2x_2x_2'+1.$$

This can be written as a *Euclidean inner product* in dimension 6:

$$K(\boldsymbol{x},\boldsymbol{x}') = \langle \phi(\boldsymbol{x}),\ \phi(\boldsymbol{x}')\rangle_{\mathbb{R}^6},$$

where a corresponding feature map (one realization of the feature space associated with K) is

$$\phi : \mathbb{R}^2 \to \mathbb{R}^6, \qquad \phi(\boldsymbol{x}) = (x_1^2,\ \sqrt{2}\,x_1x_2,\ x_2^2,\ \sqrt{2}\,x_1,\ \sqrt{2}\,x_2,\ 1).$$

With this normalization, one checks directly that

$$\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}')\rangle = x_1^2x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 + 2x_1x_1' + 2x_2x_2' + 1 = \left(\langle \boldsymbol{x},\boldsymbol{x}'\rangle + 1\right)^2 = K(\boldsymbol{x},\boldsymbol{x}').$$

Remark 6.1. The feature map ϕ is not unique (any orthogonal transformation of ϕ also works); what is determined by K is the equivalence class of the feature space (isomorphic to the associated RKHS).

Without the kernel trick, Using a degree-2 polynomial kernel would require estimating a normal vector $\boldsymbol{w}$ in $\mathbb{R}^6$ (and, for higher degrees and/or larger d, in a combinatorially explosive dimension); with the kernel, it suffices to evaluate $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ and to optimize the dual in dimension N via the kernel Gram matrix $K = [K(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{i,j}$.

The preceding reasoning also clarifies the algorithmic choice. In the kernelized primal, the normal vector $\boldsymbol{w}$ lives in $\mathcal{H}$, whose dimension can be huge (even *infinite* for an RBF): estimating $\boldsymbol{w}$ directly is quickly impractical. Using the Lagrangian, the KKT conditions instead yield the representation

$$\boldsymbol{w}^\star \;=\; \sum_{i=1}^N \lambda_i^\star\, y_i\, \phi(\boldsymbol{x}_i), \qquad 0 \leq \lambda_i^\star \leq C, \quad \sum_{i=1}^N \lambda_i^\star y_i = 0,$$

and lead to the dual

$$\max_{\boldsymbol{\lambda}\in[0,C]^N} \; \sum_{i=1}^N \lambda_i \;-\; \frac{1}{2}\sum_{i,j=1}^N \lambda_i\lambda_j y_i y_j \; \underbrace{\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}}}_{K(\boldsymbol{x}_i,\boldsymbol{x}_j)} \quad \text{s.t. } \sum_{i=1}^N \lambda_i y_i = 0.$$

In other words, the entire numerical challenge concentrates in the *kernel Gram matrix* $K = [K(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{i,j}$: we manipulate only inner products in $\mathcal{H}$, never building ϕ nor storing $\boldsymbol{w}$. This is precisely why one prefers the dual: the complexity depends primarily on N (sample size), not on $\dim(\mathcal{H})$, which may blow up in the primal.

6.2 Formalization of the kernel trick

Definition 6.1 (Hilbert space)**.** A Hilbert space is a real vector space $\mathcal{H}$ endowed with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, complete for the induced norm $\|u\|_{\mathcal{H}} = \sqrt{\langle u,u\rangle_{\mathcal{H}}}$. Completeness means that every Cauchy sequence (for the norm) converges in $\mathcal{H}$.

Definition 6.2 (p.s.d. kernel and RKHS, (Aronszajn, 1950))**.** Let $\mathcal{X}$ be a nonempty set. A symmetric function $K : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ is *positive semidefinite* (p.s.d.) if, for every $n\in\mathbb{N}$ and every choice $(x_1,\dots,x_n)\subset\mathcal{X}$,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\,K(x_i,x_j)\;\ge\;0\quad\text{for all }(\alpha_1,\dots,\alpha_n)\in\mathbb{R}^n,$$

i.e., the Gram matrix $K_n = [K(x_i,x_j)]_{i,j}$ is p.s.d. By the Moore–Aronszajn theorem, every p.s.d. kernel K is associated with a *reproducing kernel Hilbert space* (RKHS) $(\mathcal{H}_K,\langle\cdot,\cdot\rangle_{\mathcal{H}_K})$ and a canonical map

$$\Phi:\;\mathcal{X}\to\mathcal{H}_K,\qquad \Phi(x) := K_x := K(x,\cdot),$$

such that, for all $x,z\in\mathcal{X}$,

$$K(x,z)\;=\;\langle\Phi(x),\Phi(z)\rangle_{\mathcal{H}_K}\qquad\text{and}\qquad\forall f\in\mathcal{H}_K,\;\;f(x)=\langle f,\;K_x\rangle_{\mathcal{H}_K}\;\;\text{(reproducing property)}.$$

Remark 6.2 (Two equivalent views of a kernel)*.* Saying that K is a p.s.d. kernel is equivalent to (i) the existence of a *feature map* Φ into a Hilbert space $(\mathcal{H}_K,\langle\cdot,\cdot\rangle)$ such that $K(x,z)=\langle\Phi(x),\Phi(z)\rangle$, *or* (ii) every Gram matrix formed with K is p.s.d. The first view is geometric (feature map); the second is algorithmic (well-posed QP in the dual).

Remark 6.3 (Examples and closure properties)*.* Common p.s.d. examples: $K_{\mathrm{lin}}(x,z)=\langle x,z\rangle$, $K_{\mathrm{poly}}(x,z)=(\langle x,z\rangle+c)^p$ ($c\ge 0,\ p\in\mathbb{N}$), $K_{\mathrm{RBF}}(x,z)=\exp(-\gamma\|x-z\|^2)$ ($\gamma>0$). Closure: if K_1,K_2 are p.s.d., then aK_1+bK_2 ($a,b\ge 0$), $K_1\cdot K_2$, and $x\mapsto\phi(x)^\top A\,\phi'(x)$ with $A\succeq 0$ are also p.s.d. (useful for building domain-adapted kernels).

Theorem 6.1 (Representer theorem, Kimeldorf and Wahba (1971); Schölkopf and Smola (2002))**.** *Let $(\mathcal{H}_K,\|\cdot\|_{\mathcal{H}_K})$ be an RKHS, $\Omega:\mathbb{R}_+\to\mathbb{R}$ a strictly increasing regularizer, and $L:\boldsymbol{Y}\times\mathbb{R}\to\mathbb{R}_+$ a loss. For*

$$\min_{f\in\mathcal{H}_K}\;\Omega(\|f\|_{\mathcal{H}_K})\;+\;\sum_{i=1}^{N}L(y_i,\;f(x_i)),$$

every minimizer admits a finite representation

$$f^\star(\cdot)=\sum_{i=1}^{N}\alpha_i\,K(x_i,\cdot).$$

Proof sketch. Decompose $f = g+h$ with $g\in\operatorname{span}\{K_{x_i}\}_{i=1}^N$ and $h\perp\operatorname{span}\{K_{x_i}\}$. By the reproducing property, $f(x_i)=g(x_i)$ for all i; thus the data-fitting term depends only on g. Since Ω is increasing and $\|f\|^2=\|g\|^2+\|h\|^2$, keeping $h\neq 0$ never helps: at the optimum $h=0$, yielding the finite form. □

Remark 6.4 (Direct connection to the kernelized SVM)*.* With $\Omega(t)=\frac{1}{2}t^2$ and the hinge loss $L(y,u)=\max(0,1-yu)$, one obtains the SVC in RKHS:

$$\min_{f\in\mathcal{H}_K,\;b\in\mathbb{R}}\;\frac{1}{2}\|f\|^2_{\mathcal{H}_K}+C\sum_{i=1}^{N}\max\left(0,\;1-y_i(f(x_i)+b)\right).$$

By Theorem 6.1, $f(\cdot)=\sum_i\alpha_iK(x_i,\cdot)$. Introducing slack variables and writing the dual as in the linear case, *all* inner products $\langle x_i,x_j\rangle$ are replaced by $K(x_i,x_j)$:

$$\max_{\boldsymbol{\lambda}}\;\sum_{i=1}^{N}\lambda_i\;-\;\frac{1}{2}\sum_{i,j=1}^{N}\lambda_i\lambda_j\,y_iy_j\,K(x_i,x_j)\quad\text{s.t.}\quad 0\le\lambda_i\le C,\;\sum_i\lambda_iy_i=0,$$

and the decision function

$$f(x) = \sum_{i=1}^{N} \lambda_i^\star y_i \, K(x_i, x) + b^\star.$$

The kernel thus allows us to operate as if we were linear in a (potentially infinite-)dimensional $\mathcal{H}_K$, without ever making Φ explicit or estimating $\boldsymbol{w}$ in that space.

Remark 6.5 (Role of the RKHS norm and complexity control)**.** In a kernelized SVM, the quantity $\|f\|_{\mathcal{H}_K}$ plays the role of the *normal vector norm* in feature space: minimizing $\frac{1}{2}\|f\|^2_{\mathcal{H}_K}$ amounts to *maximizing the margin* in $\mathcal{H}_K$. The parameter C balances this margin against hinge-type violations. This realizes the intuition behind Cover's theorem (improved separability after embedding) while avoiding overfitting via explicit RKHS regularization.

Remark 6.6 (Why kernelization resolves the infinite-dimensional issue)**.** Attempting to optimize directly over $\boldsymbol{w} \in \mathcal{H}$ (e.g., with an RBF kernel) would yield an *infinite*-dimensional decision variable. The dual never exposes $\boldsymbol{w}$: it manipulates only the kernel Gram matrix $K = [K(x_i, x_j)]$ of size $N \times N$, and the dual coefficients λ_i. Feasibility and concavity of the dual rely precisely on the p.s.d.-ness of K.

Further reading. For a clear introduction to the kernel trick and feature maps $\varphi(\cdot)$, see Hastie et al. (2009) and Bishop (2006). For a rigorous presentation via reproducing kernel Hilbert spaces (RKHS), the positive-definiteness axiom, and the *representer theorem*, see Schölkopf and Smola (2002).

7 Practical Aspects and Extensions of SVMs

7.1 Probabilistic outputs: from margin to probability

The SVM returns a *margin score*

$$h(\boldsymbol{x}) \;=\; \sum_{i=1}^{N} \lambda_i^\star \, y_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) \;+\; b^\star \qquad \text{(linear when } K(\boldsymbol{x}_i, \boldsymbol{x}) = \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle\text{)},$$

which quantifies the signed distance to the decision boundary. This score is well ordered (the larger $h(\boldsymbol{x})$, the higher the confidence for class $+1$), but it is *not* a probability. To obtain an estimate of $\Pr(Y = 1 \mid \boldsymbol{x})$, one applies a *calibration* to the score.

Definition 7.1 (Logistic calibration)**.** Use the sigmoid $\sigma(t) = \dfrac{1}{1 + \mathrm{e}^{-t}}$ to transform $h(\boldsymbol{x})$ into a probability:

$$\Pr(Y = 1 \mid \boldsymbol{x}) \;=\; \sigma\big(\theta_1 + \theta_2 \, h(\boldsymbol{x})\big).$$

The "plain sigmoid" case fixes $(\theta_1, \theta_2) = (0, 1)$, i.e. $\Pr(Y = 1 \mid \boldsymbol{x}) = \sigma\big(h(\boldsymbol{x})\big)$. In practice, it is preferable to *learn* (θ_1, θ_2) by maximum likelihood on a validation set (or via cross-validation), from the pairs $\big(h(\boldsymbol{x}_i), t_i\big)$ with $t_i = \frac{1+y_i}{2} \in \{0, 1\}$:

$$\min_{\theta_1, \theta_2} \; -\sum_{i=1}^{N} \Big[t_i \log \sigma\big(\theta_1 + \theta_2 h(\boldsymbol{x}_i)\big) + (1 - t_i) \log\big(1 - \sigma\big(\theta_1 + \theta_2 h(\boldsymbol{x}_i)\big)\big) \Big].$$

Remark 7.1 (Intuition and practice)**.** Very positive margins ($h(\boldsymbol{x}) \gg 0$) yield probabilities close to 1, very negative margins ($h(\boldsymbol{x}) \ll 0$) close to 0, and $h(\boldsymbol{x}) \approx 0$ a probability near $1/2$. Estimation of (θ_1, θ_2) must be performed *out of sample* to avoid optimism (hold-out or k-fold CV). When the link between h and probability is more complex, a nonparametric alternative is *isotonic calibration* (more flexible but more prone to overfitting). These techniques apply unchanged to kernelized SVMs (only the expression of h changes).

7.2 Multiclass SVMs

Many applications involve $K \geq 3$ classes. While the SVM is defined for binary labels $y \in \{-1, +1\}$, multiclass classification is handled effectively via reduction strategies that keep the geometry and regularization of the binary SVMs intact (kernelization applies unchanged). We present the two standard schemes and their practical implications.

Definition 7.2 (One-vs-Rest (OvR)). Let $\mathcal{Y} = \{1, \dots, K\}$. For each class k, create binary labels

$$y_i^{(k)} = \begin{cases} +1 & \text{if } y_i = k, \\ -1 & \text{otherwise,} \end{cases} \qquad i = 1, \dots, N,$$

and train a (linear or kernel) soft-margin SVM

$$f_k(x) = \sum_{i=1}^{N} \lambda_i^{(k)} \, y_i^{(k)} \, K(x_i, x) + b^{(k)}.$$

Prediction is by winner-takes-all:

$$\hat{y}(x) = \arg \max_{k \in \{1, \dots, K\}} f_k(x).$$

Class imbalance is inherent (positives are typically $N_k \ll N$): use class weighting and stratified folds within the standard pipeline {StandardScaler → SVM}.

Definition 7.3 (One-vs-One (OvO)). For each ordered pair (a, b) with $1 \leq a < b \leq K$, train a binary SVM f_{ab} using only samples with $y \in \{a, b\}$ (labels $+1$ for a, -1 for b). At prediction time:

- **Majority vote:** each f_{ab} casts one vote for a if $f_{ab}(x) > 0$ and for b otherwise; choose the class with most votes.
- **Margin aggregation (tie-breaker):** for each class c, define

$$S_c(x) = \sum_{\substack{b=1 \\ b \neq c}}^{K} s_{cb}(x), \qquad s_{cb}(x) = \begin{cases} f_{cb}(x) & \text{if } c < b, \\ -f_{bc}(x) & \text{if } c > b, \end{cases}$$

  then predict $\hat{y}(x) = \arg\max_c S_c(x)$.

Both OvR and OvO accept any kernel from Section 6; the decision functions remain linear combinations of kernels centered at support vectors.

7.3 Least-Squares SVM

Least-Squares SVM (LS-SVM) (Suykens and Vandewalle (1999); Suykens et al. (2002)) is a variant of the soft-margin SVM that replaces the *hinge* loss and inequality constraints with a quadratic penalty on margin residuals and equality constraints. The problem remains convex, and training reduces to solving a linear system, with the same kernelization scheme as for standard SVMs.

Definition 7.4 (LS-SVM — primal formulation (classification)). Let $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ be a feature space, $\phi : \mathbb{R}^d \to \mathcal{H}$ a feature map, labels $y_i \in \{-1, 1\}$, and $\gamma > 0$. The LS-SVM primal problem is

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathcal{H},\, b \in \mathbb{R},\, \boldsymbol{\xi} \in \mathbb{R}^N} \quad & \frac{1}{2} \|\boldsymbol{w}\|_{\mathcal{H}}^2 + \frac{\gamma}{2} \sum_{i=1}^{N} \xi_i^2 \\ \text{s.t.} \quad & y_i \big(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i) \rangle_{\mathcal{H}} + b \big) = 1 - \xi_i, \qquad i = 1, \dots, N. \end{aligned} \tag{7.1}$$

Remark 7.2 (Comments). (i) The slack variables ξ_i are not sign-constrained; the penalty is symmetric around the target margin 1. (ii) The objective is strictly convex in $\boldsymbol{w}$ and the constraints are affine; the problem is a well-posed convex program. (iii) The parameter γ controls the trade-off between regularization and fitting the margin residuals.

Proposition 7.1 (Stationarity conditions and linear system)**.** *Introduce Lagrange multipliers $\lambda_i \in \mathbb{R}$ for the equalities and the Lagrangian*

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{w}\|_{\mathcal{H}}^2 + \frac{\gamma}{2}\sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} \lambda_i \Big(y_i(\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} + b) - 1 + \xi_i \Big).$$

Stationarity yields

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \iff \boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \, \phi(\boldsymbol{x}_i), \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \iff \sum_{i=1}^{N} \lambda_i y_i = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \iff \gamma\, \xi_i - \lambda_i = 0 \;\; \textit{for all } i \;\; \textit{hence} \;\; \xi_i = \lambda_i/\gamma.$$

Substituting into the constraints of (7.1) *and writing $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}}$ and $\Omega_{ij} = y_i y_j \, K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we obtain the block system*

$$\begin{bmatrix} 0 & \boldsymbol{y}^\top \\ \boldsymbol{y} & \Omega + \gamma^{-1} I_N \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1}_N \end{bmatrix}, \qquad \boldsymbol{y} = (y_1, \dots, y_N)^\top, \quad \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)^\top. \tag{7.2}$$

Definition 7.5 (Decision function)**.** Let $(b, \boldsymbol{\lambda})$ solve (7.2). The decision function is

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda_i \, y_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \qquad \widehat{y} = \operatorname{sign}\left(f(\boldsymbol{x})\right).$$

Proposition 7.2 (Dual formulation)**.** *Eliminating $(\boldsymbol{w}, b, \boldsymbol{\xi})$ in the Lagrangian leads to the dual problem*

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^N} \;\; \mathbf{1}_N^\top \boldsymbol{\lambda} \; - \; \frac{1}{2} \boldsymbol{\lambda}^\top (\Omega + \gamma^{-1} I_N) \boldsymbol{\lambda} \quad \textit{s.t.} \quad \boldsymbol{y}^\top \boldsymbol{\lambda} = 0.$$

Remark 7.3 (Geometric view, strengths and limitations). Writing $f(\boldsymbol{x}_i) = \sum_{j=1}^{N} \lambda_j y_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) + b$, the primal constraints give $\xi_i = 1 - y_i f(\boldsymbol{x}_i)$. We have $\xi_i \le 0$ for points beyond the margin on the correct side, $0 < \xi_i < 1$ for points inside the margin band but correctly classified, and $\xi_i \ge 1$ for misclassified points. *Strengths:* training reduces to a linear system of size $(N{+}1) \times (N{+}1)$; kernelization is identical to the standard SVM. *Limitations:* the solution is typically *dense* (less sparsity), and the quadratic penalty increases sensitivity to *outliers*.

7.4 Support Vector Regression

Support Vector Regression (SVR) (Smola and Schölkopf (2004)) seeks to estimate an affine function in feature space, $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x})\rangle_{\mathcal{H}} + b$, that tolerates deviations up to a threshold $\epsilon > 0$ without penalty (ϵ-*insensitive loss*). Geometrically, this amounts to fitting a *tube* of width 2ϵ around the predictor (see Figure 7): points inside the tube incur no penalty, whereas points outside give rise to slack variables.

Definition 7.6 (SVR — primal formulation)**.** Let $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ be a feature Hilbert space, $\phi : \mathbb{R}^d \to \mathcal{H}$ an embedding, $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ observations, and $C > 0$. The SVR primal problem is

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathcal{H},\, b \in \mathbb{R},\, \boldsymbol{\xi} \in \mathbb{R}_+^N,\, \boldsymbol{\xi}^\star \in \mathbb{R}_+^N} \quad & \frac{1}{2}\|\boldsymbol{w}\|_{\mathcal{H}}^2 \; + \; C \sum_{i=1}^{N} (\xi_i + \xi_i^\star) \\ \text{s.t.} \quad & y_i - (\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} + b) \;\le\; \epsilon + \xi_i, \qquad i = 1, \dots, N, \\ & (\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle_{\mathcal{H}} + b) - y_i \;\le\; \epsilon + \xi_i^\star, \qquad i = 1, \dots, N, \\ & \xi_i \ge 0, \quad \xi_i^\star \ge 0, \qquad i = 1, \dots, N. \end{aligned} \tag{7.3}$$

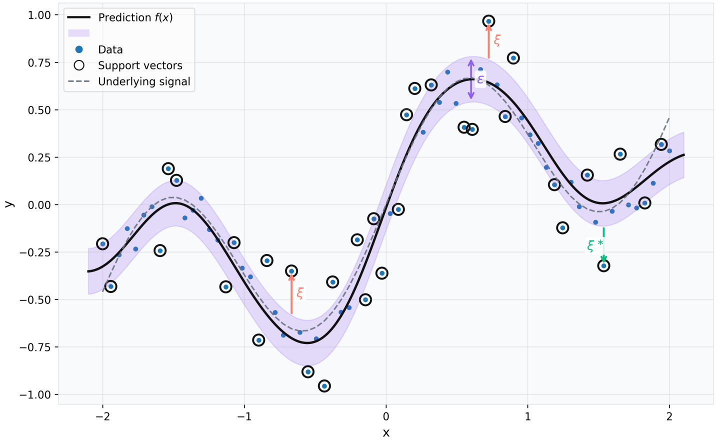


Figure 7 – **SVR and the ϵ-insensitive tube.** Estimated curve $f(\boldsymbol{x})$ (solid line), ϵ-tube (purple bands). Points outside the tube generate slack variables $(\xi_i, \xi_i^\star)$ and become *support vectors* of the regression.

Proposition 7.3 (Stationarity conditions and box constraints)**.** *Let* $\lambda_i, \lambda_i^\star \geq 0$ *be the Lagrange multipliers associated with the two families of inequalities in* (7.3)*, and let* $\mu_i, \mu_i^\star \geq 0$ *be those associated with* $\xi_i \geq 0$*,* $\xi_i^\star \geq 0$*. Writing* $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}}$*, the Lagrangian stationarity conditions yield*

$$\boldsymbol{w} = \sum_{i=1}^{N} (\lambda_i^\star - \lambda_i)\, \phi(\boldsymbol{x}_i), \qquad \sum_{i=1}^{N} (\lambda_i^\star - \lambda_i) = 0.$$

We also have $C - \lambda_i - \mu_i = 0$ *and* $C - \lambda_i^\star - \mu_i^\star = 0$ *for all* i*, with complementarity* $\mu_i \xi_i = 0$ *and* $\mu_i^\star \xi_i^\star = 0$*. It follows that the dual variables satisfy the* box *bounds*

$$0 \;\leq\; \lambda_i \;\leq\; C, \qquad 0 \;\leq\; \lambda_i^\star \;\leq\; C, \qquad i = 1, \ldots, N.$$

Proposition 7.4 (Kernelized dual)**.** *Eliminating* $(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^\star, \boldsymbol{\mu}, \boldsymbol{\mu}^\star)$ *from the Lagrangian associated with* (7.3) *leads to the dual problem*

$$\begin{aligned} \max_{\boldsymbol{\lambda}, \boldsymbol{\lambda}^\star} \quad & -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\lambda_i - \lambda_i^\star)(\lambda_j - \lambda_j^\star)\, K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & \quad - \epsilon \sum_{i=1}^{N} (\lambda_i + \lambda_i^\star) \;+\; \sum_{i=1}^{N} y_i (\lambda_i - \lambda_i^\star) \\ \text{s.t.} \quad & \sum_{i=1}^{N} (\lambda_i - \lambda_i^\star) = 0, \qquad 0 \leq \lambda_i, \lambda_i^\star \leq C, \quad i = 1, \ldots, N. \end{aligned} \tag{7.4}$$

Definition 7.7 (Predictor and bias computation)**.** For a solution $(\boldsymbol{\lambda}, \boldsymbol{\lambda}^\star)$ of (7.4), the prediction function is

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} (\lambda_i - \lambda_i^\star)\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

The term b follows from the KKT conditions by selecting an index i such that $0 < \lambda_i < C$ or $0 < \lambda_i^\star < C$. For instance,

$$b \;=\; y_i - \epsilon - \sum_{j=1}^{N} (\lambda_j - \lambda_j^\star)\, K(\boldsymbol{x}_j, \boldsymbol{x}_i) \quad \text{if } 0 < \lambda_i < C,$$

and

$$b \;=\; y_i + \epsilon - \sum_{j=1}^{N} (\lambda_j - \lambda_j^\star)\, K(\boldsymbol{x}_j, \boldsymbol{x}_i) \quad \text{if } 0 < \lambda_i^\star < C.$$

In practice, one averages b over several admissible indices for improved numerical stability.

Remark 7.4 (Geometric interpretation and role of hyperparameters). Points with $0 < \lambda_i < C$ or $0 < \lambda_i^\star < C$ lie *on* the tube and are support vectors; those with $\lambda_i = C$ or $\lambda_i^\star = C$ are *outside* the tube. The parameter ϵ controls the tube thickness: a larger value induces a smoother model and fewer support vectors, but higher bias. The parameter C tunes the penalty for overshooting: a large value favors tighter fit (higher variance), whereas a small value allows more tolerance (lower variance). Kernelization enters solely through $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, exactly as in classification.

7.5 Modelling: practical principles and hyperparameter tuning

This section gathers the *modelling* choices that drive SVM performance: (i) feature handling, (ii) the effect of hyperparameters on the geometry of the decision boundary, and (iii) a lean tuning/evaluation protocol. The figures illustrate the effect of a very large C (Figure 8) and a very large γ for an RBF kernel (Figure 9).

Feature handling — SVMs are *scale-sensitive*. We work within a pipeline that includes standardisation $z = (x - \mu)/\sigma$, fitted *exclusively* on the training folds inside cross-validation. Categorical variables are one-hot encoded. The presence of outliers justifies mild *robust scaling* (or winsorisation) to stabilise the margin when C is large. With class imbalance, use class weighting (`class_weight="balanced"` or manual weights) and imbalance-aware metrics (AUC-PR, F1, balanced accuracy).

Hyperparameters —

— (i) *Linear SVM:* $C > 0$ sets the *margin–violation* trade-off. A *large* C tightens the margin and reduces training errors (variance ↑); a *small* C widens the margin (bias ↑). The *squared hinge* loss strengthens the penalty on large violations, Figure 8

— (ii) *RBF:* two *coupled* parameters shape the boundary, C and $\gamma > 0$ (effective kernel width). A *large* γ yields highly *local, wiggly* boundaries (variance ↑); a *small* γ gives *smooth* boundaries (bias ↑). Search must be *joint* in (C, γ), Figure 9.

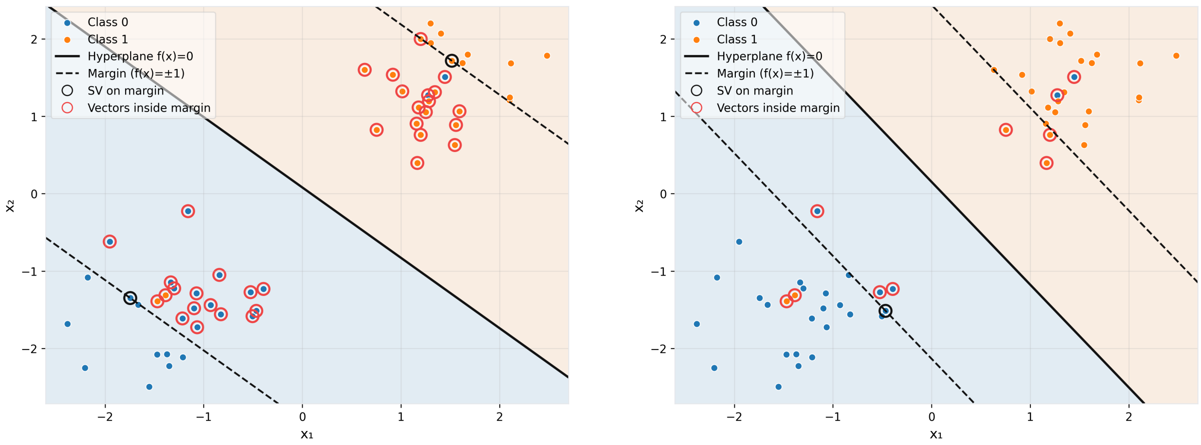
— (iii) *Polynomial (optional):* the degree p sets global complexity (in practice $p \in \{2, 3\}$), while $c_0 \geq 0$ modulates the influence of lower-order terms.

Remark 7.5. *Bias–variance view.* Increasing C or γ reduces bias but raises variance; under RBF these effects compound. Decreasing them smooths the boundary at the cost of higher bias.

Tuning and evaluation — Use a pipeline {StandardScaler → SVM} with stratified K-fold CV (K=5 by default). For RBF, explore *logarithmic* grids $C \in [10^{-3}, 10^{3}]$, $\gamma \in [10^{-4}, 10^{1}]$; a useful starting heuristic is

$$\gamma \;\approx\; \frac{1}{\text{median}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)},$$

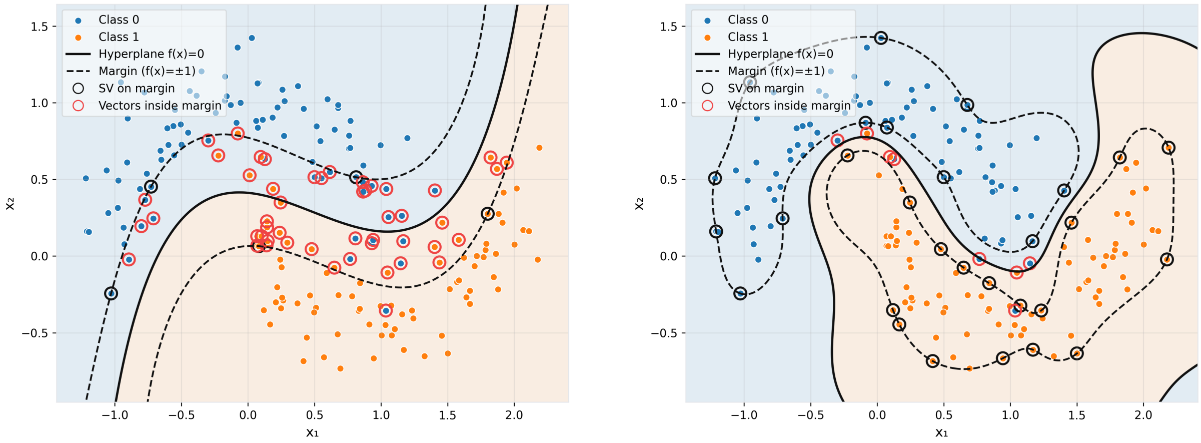
computed on a subsample. If calibrated probabilities are required (see Section 7.1), apply calibration (Platt or isotonic) *after* model selection, on a dedicated validation split.



(a) Linear SVM with $C = 0.010$

(b) Linear SVM with $C = 10000$

Figure 8 – **Effect of C (soft margin)** on the margin and the set of support vectors.



(a) RBF SVM with $\gamma = 0.2$ (C fixed at 8)

(b) RBF SVM with $\gamma = 5.0$ (C fixed at 8)

Figure 9 – **Effect of γ (RBF)** on the decision boundary and the apparent margin.

References

Mark A. Aizerman, Eduard M. Braverman, and Lev I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition. In *Automation and Remote Control*, volume 25, pages 821–837, 1964.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Bart Baesens, Tony Van Gestel, Stijn Viaene, Maria Stepanova, Johan A. K. Suykens, and Jan Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6):627–635, 2003. doi: 10.1057/palgrave.jors.2601545.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT)*, pages 144–152, 1992.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3): 326–334, 1965.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002. doi: 10.1023/A:1012487302797.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, New York, 2 edition, 2009.

C. Hurlin. Support vector machine, September 2025. URL https://doi.org/10.5281/zenodo.17115854.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML 1998)*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998. doi: 10.1007/BFb0026683.

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

Gert Loterman, Iain Brown, David Martens, Christophe Mues, and Bart Baesens. Benchmarking regression algorithms for loss given default modeling. *International Journal of Forecasting*, 28 (1):161–170, 2012. doi: 10.1016/j.ijforecast.2011.01.006.

Farid Melgani and Lorenzo Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8): 1778–1790, 2004. doi: 10.1109/TGRS.2004.831865.

A. B. J. Novikoff. On convergence proofs on perceptrons. *Proceedings of the Symposium on the Mathematical Theory of Automata*, 12:615–622, 1962.

Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: An application to face detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–136. IEEE, 1997. doi: 10.1109/CVPR.1997.598362.

Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 1996.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

Johan A. K. Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

Ellen Tobback, David Martens, Tony Van Gestel, and Bart Baesens. Forecasting loss given default models: impact of account characteristics and the macroeconomic state. *Journal of the Operational Research Society*, 65(3):376–392, 2014. doi: 10.1057/jors.2013.158.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

Xiao Yao, Jonathan Crook, and Galina Andreeva. Support vector regression for loss given default modelling. *European Journal of Operational Research*, 240(2):528–538, 2015. doi: 10.1016/j.ejor.2014.06.043.

Xiao Yao, Jonathan Crook, and Galina Andreeva. Enhancing two-stage modelling methodology for loss given default with support vector machines. *European Journal of Operational Research*, 263(2):679–689, 2017. doi: 10.1016/j.ejor.2017.05.017.