DSC5001: Statistical Machine Learning I

Week 11: Kernel Machines, Regularization, and Support Vector Machines

Kangning Cui
City University of Hong Kong (Dongguan)

Semester A, 2026/27

Final Examination Format

The final examination format has been updated. The previously announced 40 multiple-choice questions have been reduced to **20 questions**.

Component	Questions	Marks
Calculation-based MCQs	10	20
Conceptual MCQs	10	20
Written questions	3	60
Total	**23**	**100**

- All questions are compulsory.
- Examination duration: **3 hours**.
- Up to **two double-sided A4 sheets of notes** are permitted.

Group Registration and Slides Submission

Group Registration

During today's class break, please send **one representative from each group** to me.

- Register the names of **all group members**.
- Make sure every student is assigned to the correct group.

Slides Submission

- Upload your group presentation slides to Canvas.
- **Deadline: Evening of 27 November 2026.**

Learning Objectives

- Classification losses and regularization
- Hard- and soft-margin SVMs
- Support vectors and the dual formulation
- Nonlinear classification with kernels

Loss and regularization

DSC5001 ▪ Semester A, 2026/27

Motivation for Loss Functions

Accuracy only records the predicted label. A score of $+0.01$ and a score of $+5$ both predict $+1$, although one is much closer to the boundary.

$$(\boldsymbol{x}_i, y_i) \longrightarrow f(\boldsymbol{x}_i) \longrightarrow \boxed{\ell\big(y_i, f(\boldsymbol{x}_i)\big)}.$$

A **loss function** assigns a cost to the score, making parameter optimization possible.

Loss per Observation

Training requires a loss for each observed label y_i and predicted score $f(\boldsymbol{x}_i)$:

$$\boxed{\ell_i = \ell(y_i, f(\boldsymbol{x}_i)) \geq 0.}$$

Squared loss measures numerical errors. Logistic and hinge losses assess classification scores. The loss chosen determines what the algorithm tries to improve.

Classification Margins

For labels $y \in \{-1, +1\}$, the **signed margin** $m = yf(\boldsymbol{x})$ is positive for a correct prediction.

$$\ell_{01}(m) = \mathbf{1}\{m \le 0\}, \qquad \ell_{\text{hinge}}(m) = \max(0, 1 - m).$$

m	-1	$1/2$	2
Zero–one loss	1	0	0
Hinge loss	2	$1/2$	0

Hinge loss also penalizes a correct prediction with an insufficient margin.

Empirical Risk

The population loss is unknown, so we minimize the average loss on the n available training cases:

$$\widehat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, f(\boldsymbol{x}_i)\big).$$

This is the **empirical risk**. A small training risk is useful, but it does not by itself guarantee reliable predictions on new data.

Training Fit and Generalization

A flexible classifier may lower training loss by following noise or exceptional observations.

$$\boxed{\text{training loss} + \lambda \times \text{complexity penalty.}}$$

The penalty makes a more complicated solution worthwhile only when its improvement in fit is large enough.

Regularization

Regularization adds a complexity penalty to the empirical risk:

$$\boxed{\widehat{f} = \arg\min_{f \in \mathcal{F}} \{\widehat{R}(f) + \lambda \Omega(f)\}.}$$

For linear scores $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$, a common choice is $\Omega(f) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$. We will encounter this same penalty in the SVM objective.

Regularized Model Comparison

Compare two models when $\lambda = 1$:

Model	Training loss	Penalty	Total
A	0.20	0.05	$\boxed{0.25}$
B	0.15	0.20	0.35

Model B fits the training data better, but Model A has the smaller *regularized* objective. The penalty changes which solution is preferred.

Regularization Strength

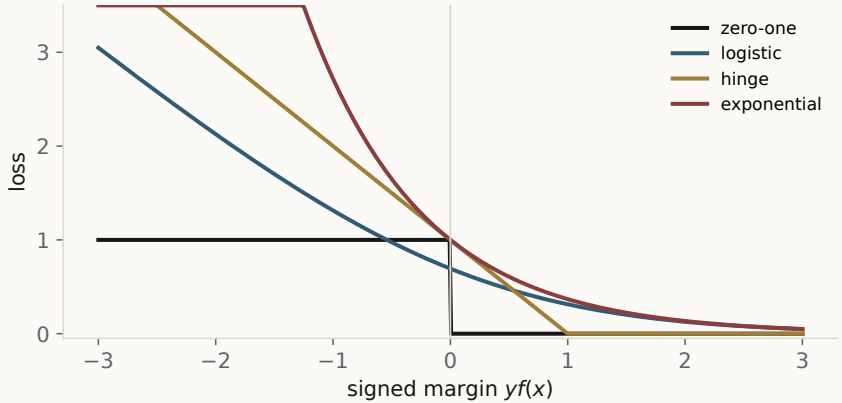
The selected model depends on the regularization strength λ:

$$J_A(\lambda) = 0.20 + 0.05\lambda, \qquad J_B(\lambda) = 0.15 + 0.20\lambda.$$

λ	J_A	J_B	Preferred
0	0.20	0.15	B
$1/2$	0.225	0.25	A
1	0.25	0.35	A

Greater λ places more weight on complexity.

Classification Loss Curves



At a positive margin, the prediction is correct. Hinge loss becomes zero only at margin 1; logistic loss decreases smoothly.

Loss Functions and Margins

Write $m = yf(\boldsymbol{x})$ to compare losses on the same scale:

$$\ell_{01}(m) = \mathbf{1}\{m \le 0\},$$
$$\ell_{\log}(m) = \log(1 + e^{-m}),$$
$$\ell_{\text{hinge}}(m) = \max(0, 1 - m).$$

SVMs use **hinge loss** because it rewards a correct classification with a margin of at least one.

Separation Quality Beyond Accuracy

Two classifiers correctly label the same three training observations, but they assign different signed scores $m_i = y_i f(\boldsymbol{x}_i)$.

	m_1	m_2	m_3	Total hinge loss
Classifier A	0.2	0.4	0.8	1.6
Classifier B	0.8	1.1	1.3	0.2

Both achieve 100% training accuracy. Their margins tell a different story.

Raw scores change when $\boldsymbol{w}, b$ are rescaled. To compare actual separation, we need *geometric distance* and a constraint on the weight norm.

Maximum-margin classification

DSC5001 ▪ Semester A, 2026/27

Motivation for Maximum Margin

Many lines can separate perfectly classified training points. A line close to a point can change its prediction after a small perturbation.

choose the widest separating gap.

This is the motivation for the **maximum-margin classifier**. We first need to measure distance from a point to a line.

Linear Decision Boundaries

For labels $y \in \{-1, +1\}$, a linear classifier uses

$$\boxed{f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b, \qquad \widehat{y} = \operatorname{sign}(f(\boldsymbol{x})).}$$

The boundary satisfies $f(\boldsymbol{x}) = 0$. The vector $\boldsymbol{w}$ is perpendicular to this boundary; b determines its position.

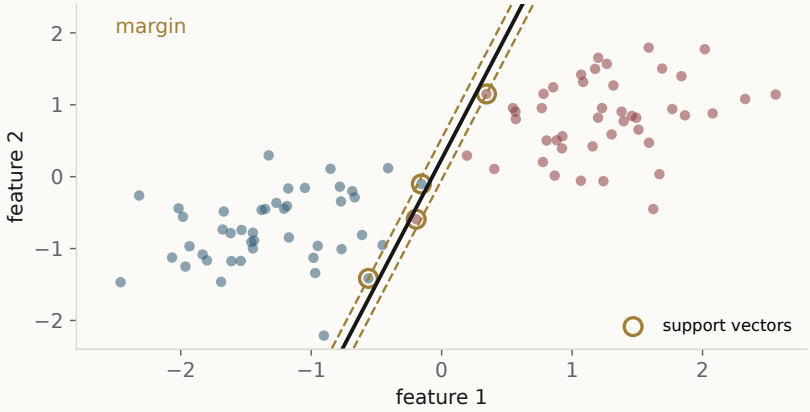
Scores and Boundary Geometry

Take $f(\boldsymbol{x}) = 2x_1 - x_2 - 2$. Its boundary is

$$\boxed{x_2 = 2x_1 - 2}, \qquad \boldsymbol{w} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

Changing b moves the line; changing the direction of $\boldsymbol{w}$ rotates it. Multiplying both $\boldsymbol{w}$ and b by the same positive factor changes the *score*, but not the boundary.

Support Vectors



The ringed points are closest to the boundary. They limit the widest possible gap and become the **support vectors**.

Active Margin Constraints

A distant point can often move without changing the widest separating gap. A point touching a margin line can force the boundary to move.

$$\boxed{\text{support vectors} \longrightarrow \text{active margin constraints.}}$$

This is why the maximum-margin solution depends most strongly on a small subset of observations.

Scores versus Distances

The score $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ cannot itself be a distance: multiplying it by 10 preserves the boundary but multiplies all scores by 10.

$$\boxed{\text{unit normal} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}.}$$

Projecting onto the unit normal removes the arbitrary scale.

Hyperplane Distance

Let $\boldsymbol{x}_0$ lie on the boundary: $\boldsymbol{w}^\top \boldsymbol{x}_0 + b = 0$.

$$\text{signed distance} = \frac{\boldsymbol{w}^\top (\boldsymbol{x} - \boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2}$$

$$= \boxed{\frac{\boldsymbol{w}^\top \boldsymbol{x} + b}{\|\boldsymbol{w}\|_2}}.$$

The denominator converts a model score into a geometric distance.

Signed Geometric Margin

Multiplying signed distance by the true label makes a correct-side distance positive:

$$\boxed{\gamma_i = \frac{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|_2}.}$$

The **geometric margin** is small near the boundary and negative for a misclassified observation. We will maximize the smallest training margin.

Distance Calculation

For $\boldsymbol{w} = (3, 4)^\top$, $b = -5$, and $\boldsymbol{x} = (2, 1)^\top$:

$$f(\boldsymbol{x}) = 3(2) + 4(1) - 5 = 5, \qquad \|\boldsymbol{w}\|_2 = 5.$$

$$\boxed{\text{signed distance} = 5/5 = 1.}$$

Doubling $\boldsymbol{w}$ and b doubles both the score and norm, leaving the distance unchanged.

Canonical Margin Planes

For separable data, rescale f so the nearest observations have signed score 1:

$$\boxed{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1.}$$

The two margin planes are $f(\boldsymbol{x}) = +1$ and $f(\boldsymbol{x}) = -1$, separated by

$$\boxed{\text{margin width} = \frac{2}{\|\boldsymbol{w}\|_2}.}$$

Canonical Score Scaling

Both f and $10f$ make the same predictions, and their geometric margins agree:

$$\frac{10y_i f(\boldsymbol{x}_i)}{\|10\boldsymbol{w}\|_2} = \frac{y_i f(\boldsymbol{x}_i)}{\|\boldsymbol{w}\|_2}.$$

Fixing the nearest signed score at 1 removes this scaling freedom. We can then maximize the margin by minimizing $\frac{1}{2}\|\boldsymbol{w}\|_2^2$.

Maximum-Margin Optimization

We want the widest margin while classifying every training point correctly:

$$\begin{aligned} \text{maximize} \quad & \frac{2}{\|\boldsymbol{w}\|_2} \\ \text{subject to} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1. \end{aligned}$$

The equivalent objective $\min \frac{1}{2}\|\boldsymbol{w}\|_2^2$ is easier to optimize.

Hard-Margin Feasibility

The **hard-margin** constraints demand perfect linear separation:

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \quad \text{for every } i.$$

If the same feature vector occurs with opposite labels, the constraints cannot both hold. Overlapping classes create the same problem, motivating a relaxed formulation.

Margin Width and Weight Norm

Under canonical scaling,

$$\boxed{\text{margin width} = \frac{2}{\|\boldsymbol{w}\|_2}.}$$

$\|\boldsymbol{w}\|_2$	1	2
Margin width	2	1

Minimizing the weight norm enlarges the margin. Squaring it preserves the minimizer and simplifies differentiation.

Classification Constraints

Each labeled point must lie on or beyond its class's margin plane:

$$\boxed{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \qquad i = 1, \ldots, n.}$$

Equality indicates a point on a margin plane. Strict inequality places it farther away. These constraints prevent training errors in the hard-margin model.

Hard-Margin SVM

Combining the margin width with perfect separation gives

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{subject to} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, n. \end{aligned}$$

The **objective** widens the gap; the **constraints** place every training point correctly.

One-Dimensional Hard Margin

Use two observations $(-2, -1)$ and $(2, +1)$ with $f(x) = wx + b$:

$$2w - b \geq 1, \qquad 2w + b \geq 1.$$

The smallest feasible weight is $w = 1/2$ with $b = 0$, so the boundary is $x = 0$.

$$\boxed{\text{margin width} = \frac{2}{|w|} = 4.}$$

Both points touch their respective margin planes.

Hard-Margin Sensitivity

The two-point SVM from the previous example has a wide separating gap. Now add two correctly labeled observations close to the boundary.

Training observations	Canonical score $f(x)$	Margin width
$(-2,-1),(2,+1)$	$x/2$	4
Also $(0.4,-1),(0.6,+1)$	$10x-5$	0.2

Both datasets are linearly separable, yet the closest two observations shrink the widest gap by a factor of 20.

A soft margin can permit controlled violations without forcing the classifier to fit every borderline observation exactly.

Soft margin and duality

DSC5001 ▪ Semester A, 2026/27

Noisy and Overlapping Classes

Measurement noise and overlapping classes can make perfect separation impossible. We still want a wide margin while allowing limited violations:

$$\boxed{y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \qquad \xi_i \geq 0.}$$

The new quantity ξ_i measures how far the point violates the required margin.

Slack Variables

A **slack variable** relaxes the signed-score requirement:

$$\boxed{y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i, \qquad \xi_i \geq 0.}$$

- $\xi_i = 0$: on or beyond the margin plane
- $0 < \xi_i < 1$: correctly classified, inside the margin
- $\xi_i > 1$: misclassified

Slack Calculation

The smallest feasible slack follows directly from the signed score:

$$\boxed{\xi_i = \max\{0, 1 - y_i f(\boldsymbol{x}_i)\}.}$$

Signed score	-1	0	$1/2$	1	2
Slack	2	1	$1/2$	0	0

A correctly classified point may still have positive slack.

Penalty for Margin Violations

Without a cost for slack, allowing violations would make the margin constraints ineffective. Add a penalty for each unit of violation:

$$\boxed{J = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_i \xi_i.}$$

The first term rewards a wide margin; $C > 0$ controls how expensive violations are.

Soft-Margin SVM

The **soft-margin SVM** balances width and violations:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \qquad (i = 1, \ldots, n). \end{aligned}$$

A larger C puts more pressure on fitting the training labels.

Slack and Hinge Loss

For fixed $\boldsymbol{w}, b$, the minimum required slack is

$$\boxed{\xi_i = \max\{0, 1 - y_i f(\boldsymbol{x}_i)\}.}$$

This is the **hinge loss**. At signed score 0.4, the point is correctly classified but has loss 0.6, because it remains inside the unit margin.

Loss and Regularization Form

Replacing each slack by its smallest feasible value gives

$$\min_{\boldsymbol{w},b} \left\{ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_i \max\{0, 1 - y_i f(\boldsymbol{x}_i)\} \right\}.$$

The SVM is now explicitly **regularization plus classification loss**: norm penalty for margin width, hinge loss for violations.

Correct Labels, Small Margins

Two correctly classified points can have very different margins:

Signed score	0.1	1.5
Zero–one loss	0	0
Hinge loss	0.9	0

Hinge loss discourages a boundary that barely separates a point, even when its label is correct.

Hinge-Loss Examples

Hinge loss covers three cases with one expression:

$$\boxed{\ell_{\text{hinge}}(m) = \max(0, 1 - m)}.$$

Signed margin m	$-1/2$	0.4	1.2
Hinge loss	1.5	0.6	0

It penalizes errors, small correct margins, and gives no penalty beyond the unit margin.

The Role of Penalty C

Compare two candidate solutions under $J = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_i \xi_i$:

	$\frac{1}{2}\|\boldsymbol{w}\|_2^2$	$\sum_i \xi_i$
A	1	3
B	3	1

At $C = 1/2$, $J_A = 2.5 < J_B = 3.5$. At $C = 2$, $J_B = 5 < J_A = 7$. A larger C makes reducing violations more important.

Motivation for the Dual Form

The primal SVM is written in terms of $\boldsymbol{w}$, but each training point also supplies a margin constraint.

$$\boxed{\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i.}$$

The **dual form** will express the classifier through weighted training examples. This will identify support vectors and later permit kernels.

Constraint Multipliers

Assign a nonnegative multiplier α_i to each constraint $y_i f(\boldsymbol{x}_i) \geq 1$.

$$\boxed{\alpha_i \geq 0, \qquad i = 1, \ldots, n.}$$

These multipliers measure how strongly the margin constraints affect the fitted solution. We obtain them by optimizing a **Lagrangian**.

Lagrangian Weight Stationarity

For hard-margin SVM, form the Lagrangian:

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_i \alpha_i \{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1\}.$$

Stationarity with respect to the weights gives

$$\mathbf{0} = \nabla_{\boldsymbol{w}} \mathcal{L} = \boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{x}_i \quad \Longrightarrow \quad \boxed{\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i.}$$

Weight Representation

The stationarity equation builds the separating direction from labeled points:

$$\boxed{\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i.}$$

For $(-2, -1)$ and $(2, +1)$ in one dimension,

$$w = 2\alpha_1 + 2\alpha_2.$$

A point with $\alpha_i = 0$ contributes nothing directly to $\boldsymbol{w}$.

Intercept Stationarity

The intercept b is not penalized. Stationarity in b gives

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_i \alpha_i y_i = 0,$$

so the dual weights must satisfy

$$\boxed{\sum_i \alpha_i y_i = 0.}$$

For two points with opposite labels, this requires $\alpha_1 = \alpha_2$.

Hard-Margin Dual

Substituting the two stationarity conditions removes $\boldsymbol{w}$ and b:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0. \end{aligned}$$

Training observations now enter only through **inner products** $\boldsymbol{x}_i^\top \boldsymbol{x}_j$.

Soft-Margin Dual Bounds

For soft-margin SVM, the multipliers also satisfy upper bounds:

$$\boxed{0 \le \alpha_i \le C, \qquad \sum_i \alpha_i y_i = 0.}$$

The cap comes from the slack penalty. The dual still depends on the same pairwise inner products, which makes nonlinear kernels possible.

Support Vector Conditions

The **complementary-slackness** condition for hard-margin SVM is

$$\boxed{\alpha_i \left[y_i f(\boldsymbol{x}_i) - 1 \right] = 0.}$$

A point strictly beyond its margin plane has $\alpha_i = 0$. Points with nonzero multipliers touch the margin and are **support vectors**.

Dual Multiplier Example

Return to $(-2, -1)$ and $(2, +1)$ with $f(x) = \frac{1}{2}x$.

$$\boxed{w = \tfrac{1}{2}, \quad b = 0, \quad \alpha_1 = \alpha_2 = \tfrac{1}{8}.}$$

Then $\alpha_1 y_1 + \alpha_2 y_2 = 0$ and

$$\alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 = \tfrac{1}{8}(2) + \tfrac{1}{8}(2) = \tfrac{1}{2} = w.$$

Both points touch the margin planes.

Margin Penalty and Validation

The soft-margin penalty controls the cost of violations:

$$\boxed{J = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_i \xi_i.}$$

Large C penalizes slack strongly and may yield a narrower margin. Small C tolerates more violations to allow a wider margin. Validation determines which balance predicts better on new cases.

Nonlinear Classification

Some class patterns cannot be separated by a straight line. For example, one class may occupy the center of a circle while the other forms an outer ring.

$$\boxed{\text{nonlinear pattern} \longrightarrow \text{new features.}}$$

A **feature map** can make the classes linearly separable in a transformed space.

Feature Interactions and Separability

Consider four points at the corners of a square. The opposite corners have the same class label.

$\boldsymbol{x}$	$(-1,-1)$	$(-1,1)$	$(1,-1)$	$(1,1)$
Class y	$+1$	-1	-1	$+1$
Feature x_1x_2	1	-1	-1	1

No straight line separates these labels in the original plane. But one new interaction feature does:

$$\boxed{\phi(\boldsymbol{x}) = x_1x_2, \qquad f(\boldsymbol{x}) = \phi(\boldsymbol{x}).}$$

Transformed coordinates can simplify classification. For many new features, we will seek a more efficient representation.

Kernel machines

DSC5001 ▪ Semester A, 2026/27

Explicit Feature Map

For a scalar x, construct $\phi(x) = (x, x^2)^\top$. A linear classifier in these features becomes

$$f(x) = w_1 x + w_2 x^2 + b.$$

Choose $w_1 = 0$, $w_2 = 1$, $b = -1$:

$$\boxed{f(x) = x^2 - 1, \quad \text{boundaries } x = \pm 1.}$$

The original input now has a nonlinear decision rule.

Feature-Space Classification

In general, transform $\boldsymbol{x}$ into new coordinates $\boldsymbol{\phi}(\boldsymbol{x})$:

$$\boxed{f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b.}$$

The classifier is linear in the *transformed* features but can have a curved boundary in the original space.

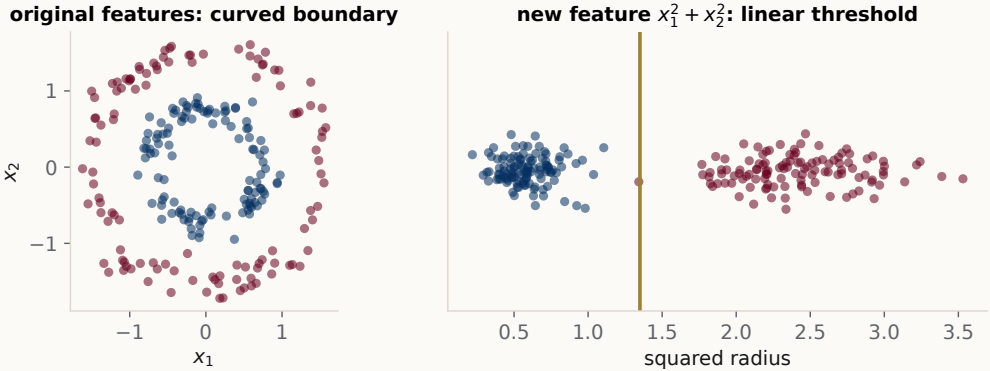
Circular Decision Boundary

Suppose one class lies inside a circle of radius 2 and the other lies outside it.

$$\phi(x_1, x_2) = x_1^2 + x_2^2, \qquad \boxed{f(\boldsymbol{x}) = \phi(\boldsymbol{x}) - 4.}$$

A threshold at $\phi = 4$ is a straight boundary in the new feature, but a circle in the original plane.

Squared-Radius Feature Map



The squared-radius feature turns the curved inside/outside boundary into one linear threshold.

Motivation for Kernels

Polynomial feature maps can generate many terms, making explicit construction expensive in high dimensions.

$$\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) \quad \longrightarrow \quad \boxed{K(\boldsymbol{x}_i, \boldsymbol{x}_j)}.$$

Because the SVM dual uses only inner products, a **kernel** can compute them without forming every transformed coordinate.

Kernel Inner Products

A kernel represents an inner product in a feature space:

$$\boxed{K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{z}).}$$

Its Gram matrix $G_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ satisfies

$$\boldsymbol{a}^\top \boldsymbol{G} \boldsymbol{a} = \left\| \sum_i a_i \boldsymbol{\phi}(\boldsymbol{x}_i) \right\|_2^2 \geq 0.$$

This property makes the kernel consistent with an inner product.

Polynomial Kernel Example

For scalar inputs, let $\phi(x) = (1, \sqrt{2}x, x^2)^\top$.

$$\begin{aligned}\phi(x)^\top \phi(z) &= 1 + 2xz + x^2 z^2 \\ &= \boxed{(1 + xz)^2} = K(x, z).\end{aligned}$$

At $x = 2$ and $z = 3$, the kernel gives $K(2, 3) = \boxed{49}$ without constructing $\phi(2)$ and $\phi(3)$.

Kernel Trick

The linear SVM dual contains pairwise products $\boldsymbol{x}_i^\top \boldsymbol{x}_j$. In a mapped feature space they become

$$\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) = \boxed{K(\boldsymbol{x}_i, \boldsymbol{x}_j)}.$$

This substitution is the **kernel trick**: the optimization uses nonlinear similarity while retaining the dual SVM structure.

Kernel Score Derivation

In feature space, the fitted direction is

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{\phi}(\boldsymbol{x}_i).$$

For a new input,

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + b \\ &= \sum_i \alpha_i y_i \underbrace{\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x})}_{K(\boldsymbol{x}_i, \boldsymbol{x})} + b. \end{aligned}$$

No explicit transformed coefficient vector is needed.

Kernel Gram Matrix

The **Gram matrix** stores kernel values for all pairs of training points. For $x_1 = 0, x_2 = 1$ and $K(x, z) = (1 + xz)^2$:

$$\boldsymbol{G} = \begin{pmatrix} K(0,0) & K(0,1) \\ K(1,0) & K(1,1) \end{pmatrix} = \boxed{\begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}}.$$

The dual optimizer uses this matrix instead of the original feature dot products.

Kernel SVM Dual

Replace every feature-space dot product in the dual by a kernel:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_i \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \quad \sum_i \alpha_i y_i = 0. \end{aligned}$$

Only kernel similarities and labels are required.

Kernel SVM Prediction

After fitting the dual weights, compute the class score for a new input:

$$\boxed{f(\boldsymbol{x}) = \sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \quad \widehat{y} = \operatorname{sign} f(\boldsymbol{x}).}$$

Terms with $\alpha_i = 0$ vanish. Prediction therefore depends only on the **support vectors**.

Kernel Prediction Example

Take two support vectors with $\alpha_1 = \alpha_2 = 1/2$, labels $(+1, -1)$, and $b = 0$:

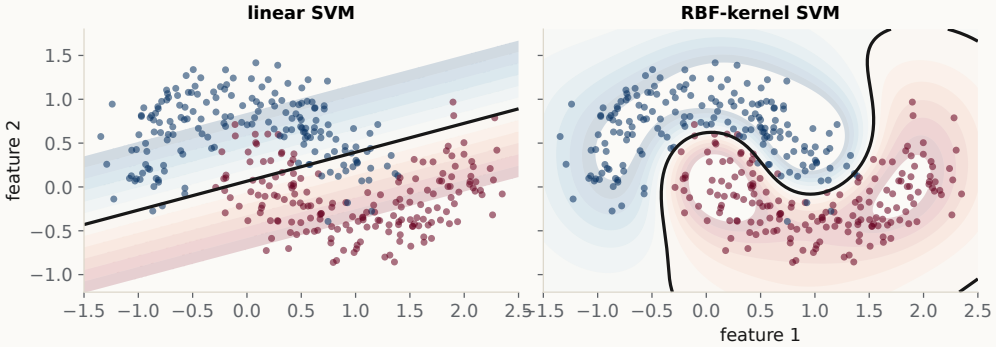
$$f(x_*) = \tfrac{1}{2}K(x_1, x_*) - \tfrac{1}{2}K(x_2, x_*).$$

If the kernel values are 3 and 1, then

$$\boxed{f(x_*) = \tfrac{1}{2}(3) - \tfrac{1}{2}(1) = 1, \qquad \widehat{y} = +1.}$$

No mapped feature vector is needed at prediction time.

RBF Decision Boundaries



The curved two-moon pattern cannot be separated well by a line. The RBF kernel can follow the gap, with flexibility controlled by its width.

RBF Length Scale

The **RBF kernel** decreases with squared distance:

$$\boxed{K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|_2^2}{2\ell^2}\right).}$$

For distance 1:

$$K_{\ell=1} = e^{-1/2}, \qquad K_{\ell=2} = e^{-1/8}.$$

Larger ℓ makes distant points more similar; smaller ℓ gives more local influence.

Kernel Locality and Overfitting

A small length scale ℓ makes kernel similarity decay quickly with distance:

$$K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|_2^2}{2\ell^2}\right).$$

The resulting boundary can follow individual training points, including noise. A broader kernel is smoother but may miss genuine local structure.

SVM Complexity Controls

An RBF SVM has two important complexity controls:

- C: cost of margin violations; larger C pressures the model to fit training labels.
- ℓ: kernel length scale; smaller ℓ allows more local variation.

Tune C and ℓ together by validation.

Joint Parameter Selection

The penalty and kernel width affect flexibility differently:

	Broad kernel	**Narrow kernel**
Small C	Smooth; more violations	Local influence; weak fitting pressure
Large C	Strong training fit pressure	Flexible; higher overfit risk

Compare combinations using the *same* validation folds; the best setting cannot be chosen from training fit alone.

Feature Scale and RBF Similarity

The RBF kernel uses Euclidean distance. Take $\boldsymbol{x} = (1, 10)^\top$, $\boldsymbol{z} = (2, 20)^\top$, and $\ell = 1$:

$$\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 = 101, \qquad K(\boldsymbol{x}, \boldsymbol{z}) = e^{-101/2}.$$

Express the second feature in units ten times as large. Now $\boldsymbol{x}' = (1, 1)^\top$ and $\boldsymbol{z}' = (2, 2)^\top$:

$$\|\boldsymbol{x}' - \boldsymbol{z}'\|_2^2 = 2, \qquad K(\boldsymbol{x}', \boldsymbol{z}') = e^{-1}.$$

The same two cases receive very different kernel similarities. **Before tuning C and ℓ,** set feature scaling using training data and apply it consistently to later observations.

SVMs Today

DSC5001 ▪ Semester A, 2026/27

Validation and Final Testing

Flexible kernels can fit training data very closely. Their settings must therefore be chosen without using final test labels.

1. Standardize features within each training fold.
2. Compare (C, ℓ) settings by cross-validation.
3. Refit the chosen setting on the training data.
4. Evaluate once on untouched test data.

Regularization Across Models

The same **loss plus regularization** principle appears in logistic regression, linear prediction heads, and SVMs:

$$\boxed{\min_{\boldsymbol{w}} \left\{ \text{training loss}(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_2^2 \right\}.}$$

SVMs add a geometric meaning: weight norm controls margin width, while hinge loss penalizes margin violations.

SVM Construction

The construction follows one logical chain:

$$\boxed{\text{loss} \longrightarrow \text{regularization} \longrightarrow \text{maximum margin}}$$

$$\boxed{\text{soft margin} \longrightarrow \text{dual} \longrightarrow \text{kernel SVM.}}$$

Every step addresses a limitation of the previous formulation: fit, stability, imperfect separation, or nonlinear boundaries.

Summary

- **Loss and regularization:** balance data fit and complexity.
- **Maximum margin:** seek a wide separating gap.
- **Soft margin:** penalize violations through hinge loss.
- **Dual:** identify support vectors.
- **Kernels:** learn nonlinear boundaries through similarities.

Choose complexity using validation, then evaluate on untouched data.

Week 11: Kernel Machines, Regularization, and SVM

Questions and discussion