



University
of Manitoba

Hyperplanes, SVM and least-squares classification

MATH 2740 – Mathematics of Data Science – Unit U06

Julien Arino

`julien.arino@umanitoba.ca`

Department of Mathematics @ University of Manitoba

Fall 2026

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis. We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

Outline

Supervised vs unsupervised linear methods

Geometry of hyperplanes and support vector machines

Least-squares classification

Supervised vs unsupervised linear methods

Geometry of hyperplanes and support vector machines

Least-squares classification

Unsupervised vs supervised linear methods

In Unit 05 (PCA), we studied **unsupervised** dimensionality reduction:

- ▶ Given unlabelled feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ (rows of the data matrix $X \in \mathcal{M}_{np}$)
- ▶ Find orthogonal directions that maximise total sample variance

Here, we study **supervised binary classification**:

- ▶ Given a labelled training dataset of n observations

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$$

where $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of p features and $y_i \in \{-1, +1\}$ (or $\{1, 2\}$) indicates the class to which observation $\mathbf{x}_i$ belongs

- ▶ Find a linear decision boundary or 1D projection that separates the two classes and predicts the label $\hat{y}(\mathbf{x})$ of any new point $\mathbf{x} \in \mathbb{R}^p$

Does maximum variance always separate classes?

Suppose two classes form elongated, parallel point clouds that are closely spaced along their perpendicular direction

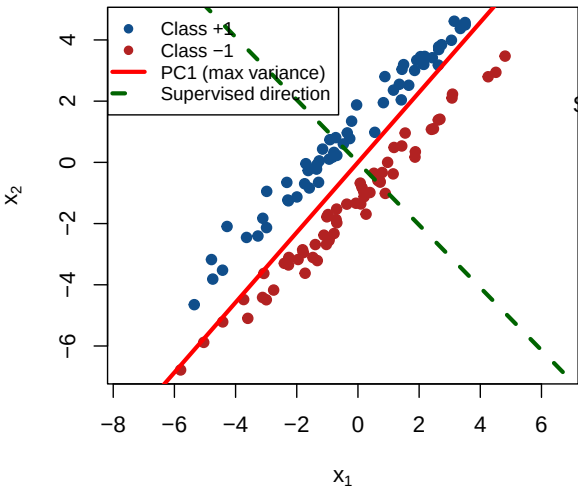
- ▶ Because PCA ignores the class labels y_i , it pools all n points together and finds the direction of largest overall spread
- ▶ When the spread *along* the clouds is larger than the distance *between* the clouds, the first principal component (PC1) runs parallel to the clouds
- ▶ Projecting onto PC1 mixes the two classes completely, whereas projecting onto the perpendicular direction separates them cleanly

When PCA fails to separate classes

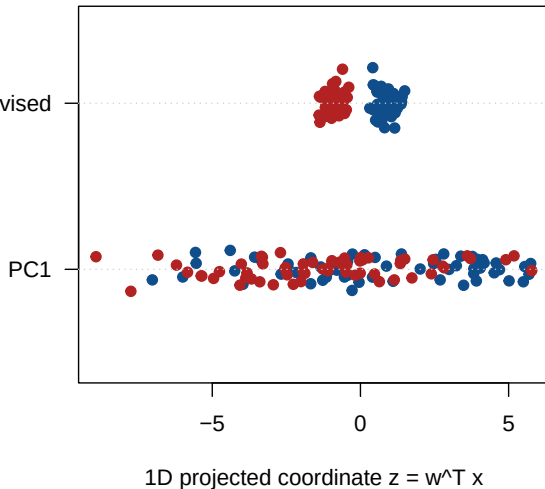
Simulate two elongated parallel classes in $\mathbb{R}^2$ and compare PC1 (red) with the supervised separating direction (green)

```
set.seed(2740)
n_c = 60
t1 = rnorm(n_c, 0, 2.5); t2 = rnorm(n_c, 0, 2.5)
X1 = cbind(t1 + rnorm(n_c, 0, 0.25), t1 + 1.2 + rnorm(n_c, 0, 0.25))
X2 = cbind(t2 + rnorm(n_c, 0, 0.25), t2 - 1.2 + rnorm(n_c, 0, 0.25))
X_all = rbind(X1, X2)
y_all = c(rep(1, n_c), rep(-1, n_c))
pc1 = eigen(cov(X_all))$vectors[, 1]
SW = (n_c - 1) * (cov(X1) + cov(X2))
w_sep = solve(SW, colMeans(X1) - colMeans(X2))
w_sep = w_sep / sqrt(sum(w_sep^2))
```

2D data with PC1 and separating directions



1D projections: PC1 vs supervised



Two linear classification frameworks in this unit

Both methods use a linear score $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ to classify points, but choose $\mathbf{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$ using different linear algebraic principles:

1. **Geometry of hyperplanes and support vector machines (SVM):**
characterise decision boundaries via normal vectors, orthogonal projections and maximum-margin separation
2. **Least-squares classification:** treat binary labels $y_i \in \{-1, +1\}$ as regression targets and solve the normal equations $(X^T X) \tilde{\mathbf{w}} = X^T \mathbf{y}$ from Unit 03

Supervised vs unsupervised linear methods

Geometry of hyperplanes and support vector machines

Least-squares classification

Geometry of hyperplanes and support vector machines

Hyperplanes and normal vectors

Orthogonal distance to a hyperplane

Linear separability and hard-margin SVM

Hyperplanes in $\mathbb{R}^p$

Definition 88 (Hyperplane in $\mathbb{R}^p$)

Let $\mathbf{w} \in \mathbb{R}^p \setminus \{\mathbf{0}\}$ be a non-zero vector and $b \in \mathbb{R}$ be a scalar. The **hyperplane** determined by $\mathbf{w}$ and b is the set of points

$$\mathcal{H} = \left\{ \mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = 0 \right\} = \left\{ \mathbf{x} \in \mathbb{R}^p : \mathbf{w} \bullet \mathbf{x} + b = 0 \right\}$$

The vector $\mathbf{w}$ is called a **normal vector** to $\mathcal{H}$ and b is called the **offset** (or **bias**)

- ▶ In $\mathbb{R}^2$, a hyperplane is a line $w_1 x_1 + w_2 x_2 + b = 0$
- ▶ In $\mathbb{R}^3$, a hyperplane is a plane $w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$
- ▶ Multiplying $\mathbf{w}$ and b by any non-zero constant $\alpha \neq 0$ leaves the set $\mathcal{H}$ unchanged

Linear subspace vs affine subspace

When is a hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = 0\}$ a vector subspace of $\mathbb{R}^p$?

- ▶ A vector subspace must contain the zero vector $\mathbf{0}$, and $\mathbf{0} \in \mathcal{H} \iff \mathbf{w}^T \mathbf{0} + b = 0 \iff b = 0$
- ▶ When $b = 0$, the set $\mathcal{H}_0 = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} = 0\} = \{\mathbf{w}\}^\perp$ is the orthogonal complement of $\text{span}(\mathbf{w})$, which is a $(p - 1)$ -dimensional vector subspace of $\mathbb{R}^p$
- ▶ When $b \neq 0$, $\mathcal{H}$ does not pass through the origin; it is an **affine subspace** obtained by translating $\mathcal{H}_0$ by any fixed point $\mathbf{x}^* \in \mathcal{H}$:

$$\mathcal{H} = \mathbf{x}^* + \mathcal{H}_0 = \left\{ \mathbf{x}^* + \mathbf{u} : \mathbf{w}^T \mathbf{u} = 0 \right\}$$

Example in $\mathbb{R}^2$ – Hyperplanes with $b = 0$ and $b \neq 0$

In $\mathbb{R}^2$, hyperplanes are lines orthogonal to $\mathbf{w} \neq \mathbf{0}$

- ▶ **Line through the origin ($b = 0$):**

$$\mathcal{H}_0 = \{ \mathbf{x} \in \mathbb{R}^2 : \mathbf{w}^T \mathbf{x} = 0 \} = \{ \mathbf{w} \}^\perp$$

Since $\mathbf{0} \in \mathcal{H}_0$, it is a 1D **vector subspace**

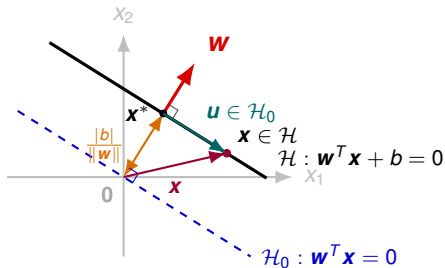
- ▶ **Line not through origin ($b \neq 0$):**

$$\mathcal{H} = \{ \mathbf{x} \in \mathbb{R}^2 : \mathbf{w}^T \mathbf{x} + b = 0 \}$$

$\mathbf{0} \notin \mathcal{H}$; **affine subspace** translated by $\mathbf{x}^* \in \mathcal{H}$:

$$\mathcal{H} = \mathbf{x}^* + \mathcal{H}_0 = \{ \mathbf{x}^* + \mathbf{u} : \mathbf{u} \in \mathcal{H}_0 \}$$

- ▶ Every $\mathbf{x} \in \mathcal{H}$ satisfies $\mathbf{w}^T \mathbf{x} = -b$



Why $\mathbf{w}$ is orthogonal to the hyperplane

Proposition 89 (Orthogonality of the normal vector)

Let $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = 0\}$. For any two points $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{H}$, the displacement vector $\mathbf{x}_1 - \mathbf{x}_2$ lying along $\mathcal{H}$ is orthogonal to $\mathbf{w}$:

$$\mathbf{w} \bullet (\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) = 0$$

Proof:

- ▶ Since $\mathbf{x}_1 \in \mathcal{H}$ and $\mathbf{x}_2 \in \mathcal{H}$, both points satisfy the hyperplane equation:

$$\mathbf{w}^T \mathbf{x}_1 + b = 0 \quad \text{and} \quad \mathbf{w}^T \mathbf{x}_2 + b = 0$$

- ▶ Subtracting the second equation from the first eliminates b :

$$(\mathbf{w}^T \mathbf{x}_1 + b) - (\mathbf{w}^T \mathbf{x}_2 + b) = \mathbf{w}^T (\mathbf{x}_1 - \mathbf{x}_2) = 0$$

- ▶ Hence $\mathbf{w} \perp (\mathbf{x}_1 - \mathbf{x}_2)$ for every pair of points $\mathbf{x}_1, \mathbf{x}_2$ on $\mathcal{H}$



Point-normal form of a (hyper)plane

A hyperplane is determined by a **point** $\mathbf{x}_0 \in \mathcal{H}$ and a **normal vector** $\mathbf{w} \neq \mathbf{0}$

- ▶ For any $\mathbf{x} \in \mathcal{H}$, the displacement $\mathbf{x} - \mathbf{x}_0$ lies along $\mathcal{H}$, so $\mathbf{w} \perp (\mathbf{x} - \mathbf{x}_0)$:

$$\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0 \quad (\text{point-normal form})$$

- ▶ Expanding gives $\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{x}_0 = 0$, which is

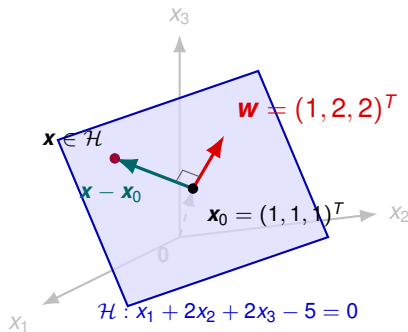
$$\mathbf{w}^T \mathbf{x} + b = 0 \quad \text{with } b = -\mathbf{w}^T \mathbf{x}_0$$

- ▶ **Example in $\mathbb{R}^3$:** Point $\mathbf{x}_0 = (1, 1, 1)^T$ and normal $\mathbf{w} = (1, 2, 2)^T$:

$$1(x_1 - 1) + 2(x_2 - 1) + 2(x_3 - 1) = 0$$

$$\iff x_1 + 2x_2 + 2x_3 - 5 = 0$$

$$\text{Here } b = -\mathbf{w}^T \mathbf{x}_0 = -(1 + 2 + 2) = -5$$



Half-spaces and linear classifiers

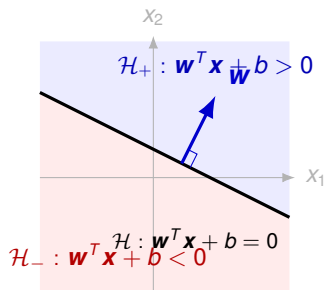
A hyperplane $\mathcal{H}$ divides $\mathbb{R}^p \setminus \mathcal{H}$ into two open **half-spaces**:

$$\mathcal{H}_+ = \left\{ \mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b > 0 \right\}$$

$$\mathcal{H}_- = \left\{ \mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b < 0 \right\}$$

- ▶ The normal vector $\mathbf{w}$ points into the positive half-space $\mathcal{H}_+$
- ▶ The corresponding **linear classifier** assigns class labels via the sign function:

$$\hat{y}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b) = \begin{cases} +1 & \text{if } \mathbf{x} \in \mathcal{H}_+ \\ -1 & \text{if } \mathbf{x} \in \mathcal{H}_- \end{cases}$$



Example – Hyperplane geometry and classification in $\mathbb{R}^2$

Using the line $\mathcal{H}$ in $\mathbb{R}^2$ given by $2x_1 - x_2 - 3 = 0$, classify $\mathbf{p} = (3, 1)^T$ and $\mathbf{q} = (0, 0)^T$

- ▶ **Normal vector and offset:** $\mathbf{w} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$ and $b = -3$, with norm
 $\|\mathbf{w}\| = \sqrt{2^2 + (-1)^2} = \sqrt{5}$
- ▶ **Two points on $\mathcal{H}$:** setting $x_1 = 0$ gives $\mathbf{x}_1 = \begin{pmatrix} 0 \\ -3 \end{pmatrix}$; setting $x_2 = 1$ gives
 $\mathbf{x}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$
- ▶ **Orthogonality check:** $\mathbf{x}_2 - \mathbf{x}_1 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ and $\mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) = 2(2) + (-1)(4) = 0$
- ▶ **Classifying test points with $\hat{y}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$:**
 - ▶ For $\mathbf{p} = (3, 1)^T$: $\mathbf{w}^T \mathbf{p} + b = 2(3) - 1(1) - 3 = +2 > 0 \implies \hat{y}(\mathbf{p}) = +1$ ($\mathbf{p} \in \mathcal{H}_+$)
 - ▶ For $\mathbf{q} = (0, 0)^T$: $\mathbf{w}^T \mathbf{q} + b = 2(0) - 1(0) - 3 = -3 < 0 \implies \hat{y}(\mathbf{q}) = -1$ ($\mathbf{q} \in \mathcal{H}_-$)

Geometry of hyperplanes and support vector machines

Hyperplanes and normal vectors

Orthogonal distance to a hyperplane

Linear separability and hard-margin SVM

Orthogonal distance from a point to a hyperplane

Theorem 90 (Distance from a point to a hyperplane)

Let $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = 0\}$ be a hyperplane with normal vector $\mathbf{w} \neq \mathbf{0}$, and let $\mathbf{x}_0 \in \mathbb{R}^p$ be any point. The Euclidean distance from $\mathbf{x}_0$ to $\mathcal{H}$ is

$$d(\mathbf{x}_0, \mathcal{H}) = \min_{\mathbf{x} \in \mathcal{H}} \|\mathbf{x}_0 - \mathbf{x}\| = \frac{|\mathbf{w}^T \mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$

- ▶ Setting $\mathbf{x}_0 = \mathbf{0}$ gives the orthogonal distance from the origin to $\mathcal{H}$:

$$d(\mathbf{0}, \mathcal{H}) = \frac{|b|}{\|\mathbf{w}\|}$$

- ▶ The quantity $\frac{\mathbf{w}^T \mathbf{x}_0 + b}{\|\mathbf{w}\|}$ is the **signed distance** from $\mathbf{x}_0$ to $\mathcal{H}$ (positive on $\mathcal{H}_+$, negative on $\mathcal{H}_-$)

Proof – Distance from a point to a hyperplane

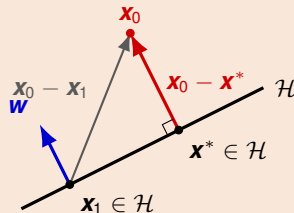
Let $\mathbf{x}^* \in \mathcal{H}$ be the orthogonal projection of $\mathbf{x}_0$ onto $\mathcal{H}$, and pick any $\mathbf{x}_1 \in \mathcal{H}$

- ▶ Since $\mathbf{x}_0 - \mathbf{x}^* \perp \mathcal{H}$, it equals the orthogonal projection of $\mathbf{x}_0 - \mathbf{x}_1$ onto $\mathbf{w}$:

$$\mathbf{x}_0 - \mathbf{x}^* = \text{proj}_{\mathbf{w}}(\mathbf{x}_0 - \mathbf{x}_1) = \frac{\mathbf{w}^T(\mathbf{x}_0 - \mathbf{x}_1)}{\|\mathbf{w}\|^2} \mathbf{w}$$

- ▶ Because $\mathbf{x}_1 \in \mathcal{H}$, we have $\mathbf{w}^T \mathbf{x}_1 = -b$, so $\mathbf{w}^T(\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{w}^T \mathbf{x}_0 + b$
- ▶ Taking the Euclidean norm gives

$$d(\mathbf{x}_0, \mathcal{H}) = \|\mathbf{x}_0 - \mathbf{x}^*\| = \frac{|\mathbf{w}^T \mathbf{x}_0 + b|}{\|\mathbf{w}\|}$$



Example – Distance to a line in $\mathbb{R}^2$

Example 1 ($\mathbb{R}^2$): Find the distance from $\mathbf{x}_0 = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$ to the line

$\mathcal{H} : 3x_1 - 4x_2 + 2 = 0$ and the closest point $\mathbf{x}^* \in \mathcal{H}$

- ▶ Here $\mathbf{w} = (3, -4)^T$ and $b = 2$, so $\|\mathbf{w}\| = \sqrt{3^2 + (-4)^2} = \sqrt{25} = 5$
- ▶ Signed numerator: $\mathbf{w}^T \mathbf{x}_0 + b = 3(4) - 4(3) + 2 = 2 > 0$ ($\mathbf{x}_0$ lies in $\mathcal{H}_+$)
- ▶ Distance: $d(\mathbf{x}_0, \mathcal{H}) = \frac{|2|}{5} = \frac{2}{5} = 0.4$, and closest point:

$$\mathbf{x}^* = \mathbf{x}_0 - \frac{\mathbf{w}^T \mathbf{x}_0 + b}{\|\mathbf{w}\|^2} \mathbf{w} = \begin{pmatrix} 4 \\ 3 \end{pmatrix} - \frac{2}{25} \begin{pmatrix} 3 \\ -4 \end{pmatrix} = \begin{pmatrix} 94/25 \\ 83/25 \end{pmatrix}$$

Example – Distance to a plane in $\mathbb{R}^3$

Example 2 ($\mathbb{R}^3$): Find the distance from $\mathbf{x}_0 = (1, -2, 3)^T$ to the plane $\mathcal{H} : x_1 + 2x_2 + 2x_3 - 5 = 0$

► Here $\mathbf{w} = (1, 2, 2)^T$, $b = -5$ and $\|\mathbf{w}\| = \sqrt{1^2 + 2^2 + 2^2} = 3$, so

$$d(\mathbf{x}_0, \mathcal{H}) = \frac{|1(1) + 2(-2) + 2(3) - 5|}{3} = \frac{|-2|}{3} = \frac{2}{3}$$

Geometry of hyperplanes and support vector machines

Hyperplanes and normal vectors

Orthogonal distance to a hyperplane

Linear separability and hard-margin SVM

Linearly separable point sets

Definition 91 (Linear separability)

Two sets of points $X_+ = \{\mathbf{x}_i : y_i = +1\}$ and $X_- = \{\mathbf{x}_i : y_i = -1\}$ in $\mathbb{R}^p$ are **linearly separable** if there exist $\mathbf{w} \in \mathbb{R}^p \setminus \{\mathbf{0}\}$ and $b \in \mathbb{R}$ such that

$$\mathbf{w}^T \mathbf{x}_i + b > 0 \quad \text{for all } \mathbf{x}_i \in X_+ \quad \text{and} \quad \mathbf{w}^T \mathbf{x}_i + b < 0 \quad \text{for all } \mathbf{x}_i \in X_-$$

- ▶ Since $y_i \in \{-1, +1\}$, both inequalities can be written as a single condition:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) > 0 \quad \text{for all } i = 1, \dots, n$$

- ▶ Whenever a finite dataset is linearly separable, there are infinitely many separating hyperplanes – how should we pick the most robust one?

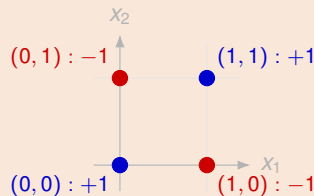
When are point sets not linearly separable? The XOR example

Not every two-class dataset in $\mathbb{R}^2$ is linearly separable

Example (XOR configuration):

- ▶ Let $X_+ = \{(0, 0)^T, (1, 1)^T\}$ ($y = +1$) and $X_- = \{(1, 0)^T, (0, 1)^T\}$ ($y = -1$)
- ▶ Suppose $\mathbf{w}^T \mathbf{x} + b > 0$ on X_+ and $\mathbf{w}^T \mathbf{x} + b < 0$ on X_- :

on X_+ : $b > 0$ and $w_1 + w_2 + b > 0$
on X_- : $w_1 + b < 0$ and $w_2 + b < 0$
- ▶ Adding the first pair gives $w_1 + w_2 + 2b > 0$,
whereas adding the second pair gives $w_1 + w_2 + 2b < 0$ – contradiction!



Scale invariance of separating hyperplanes

Let $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = 0\}$ be a hyperplane. For any constant $k > 0$, consider the rescaled parameters $\tilde{\mathbf{w}} = k\mathbf{w}$ and $\tilde{b} = kb$:

- ▶ **Same geometric hyperplane:** for all $\mathbf{x} \in \mathbb{R}^p$,

$$\tilde{\mathbf{w}}^T \mathbf{x} + \tilde{b} = (k\mathbf{w})^T \mathbf{x} + kb = k(\mathbf{w}^T \mathbf{x} + b) = 0 \iff \mathbf{w}^T \mathbf{x} + b = 0$$

Thus, $\{\mathbf{x} : \tilde{\mathbf{w}}^T \mathbf{x} + \tilde{b} = 0\}$ and $\{\mathbf{x} : \mathbf{w}^T \mathbf{x} + b = 0\}$ describe the exact same hyperplane $\mathcal{H}$

- ▶ **Same classification decisions:** since $k > 0$, signs are preserved:

$$\text{sgn}(\tilde{\mathbf{w}}^T \mathbf{x} + \tilde{b}) = \text{sgn}(k(\mathbf{w}^T \mathbf{x} + b)) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$$

Every point $\mathbf{x}$ is assigned to the exact same class (+1 or -1)

- ▶ **Same geometric distance:** by Theorem 90, $\frac{|\tilde{\mathbf{w}}^T \mathbf{x} + \tilde{b}|}{\|\tilde{\mathbf{w}}\|} = \frac{k|\mathbf{w}^T \mathbf{x} + b|}{k\|\mathbf{w}\|} = \frac{|\mathbf{w}^T \mathbf{x} + b|}{\|\mathbf{w}\|}$

Takeaway: Multiplying or dividing both $\mathbf{w}$ and b by any positive scalar $k > 0$ does not change the decision boundary or the classifications

Suppose $(\mathbf{w}, b)$ separates a finite dataset, so

$$\min_{1 \leq i \leq n} y_i(\mathbf{w}^T \mathbf{x}_i + b) = m > 0$$

By scale invariance, dividing both $\mathbf{w}$ and b by m (setting $\tilde{\mathbf{w}} = \frac{\mathbf{w}}{m}$ and $\tilde{b} = \frac{b}{m}$) does not change the decision boundary $\mathcal{H}$, and scales the minimum to 1:

$$y_i(\tilde{\mathbf{w}}^T \mathbf{x}_i + \tilde{b}) = \frac{1}{m} y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq \frac{m}{m} = 1$$

Support vectors and canonical margin hyperplanes

Dropping the tildes gives the **canonical form**:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for all } i = 1, \dots, n$$

The slab free of training points is bounded by two parallel hyperplanes:

- ▶ $\mathcal{H}_{+1} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = +1\}$ (positive margin boundary)
- ▶ $\mathcal{H}_{-1} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^T \mathbf{x} + b = -1\}$ (negative margin boundary)

Points on $\mathcal{H}_{+1}$ or $\mathcal{H}_{-1}$ (satisfying $y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$) are **support vectors**

Geometric derivation of the SVM margin width

Using Theorem 90, how wide is the margin between $\mathcal{H}_{+1}$ and $\mathcal{H}_{-1}$?

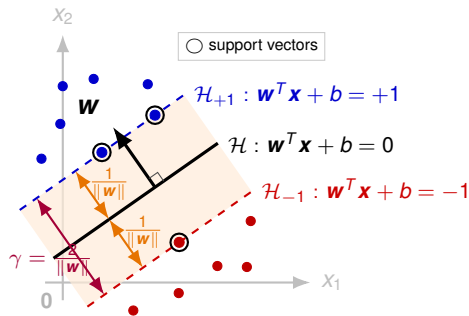
- ▶ For any point $\mathbf{x}_+ \in \mathcal{H}_{+1}$, we have $\mathbf{w}^T \mathbf{x}_+ + b = +1$, so its distance to $\mathcal{H}$ is

$$d(\mathbf{x}_+, \mathcal{H}) = \frac{|\mathbf{w}^T \mathbf{x}_+ + b|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- ▶ For any point $\mathbf{x}_- \in \mathcal{H}_{-1}$, we have $\mathbf{w}^T \mathbf{x}_- + b = -1$, so its distance to $\mathcal{H}$ is also $\frac{1}{\|\mathbf{w}\|}$

- ▶ Therefore, the total **margin width** between $\mathcal{H}_{+1}$ and $\mathcal{H}_{-1}$ is

$$\gamma = \frac{2}{\|\mathbf{w}\|}$$



Hard-margin support vector machine (SVM)

To make the separating slab as wide as possible, we want to maximise $\gamma = \frac{2}{\|\mathbf{w}\|}$, which is equivalent to minimising $\|\mathbf{w}\|$ (or $\frac{1}{2}\|\mathbf{w}\|^2$):

$$\min_{\mathbf{w} \in \mathbb{R}^p, b \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{for } i = 1, \dots, n$$

- ▶ This is the **hard-margin support vector machine (SVM)** optimisation problem
- ▶ The optimal hyperplane is completely determined by the support vectors lying on the margin boundaries $\mathbf{w}^T \mathbf{x}_i + b = \pm 1$
- ▶ For small problems in $\mathbb{R}^2$, we can find $(\mathbf{w}, b)$ directly by **geometry and linear equations**, without any Lagrange multiplier machinery

Example 1 – Hard-margin SVM for 2 points in $\mathbb{R}^2$

Problem: Find the canonical maximum-margin hyperplane $(\mathbf{w}, b)$, decision boundary and margin width γ for

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (y_1 = -1) \quad \text{and} \quad \mathbf{x}_2 = \begin{pmatrix} 3 \\ 5 \end{pmatrix} \quad (y_2 = +1)$$

Step 1 (Geometry):

- ▶ With one point in each class, both $\mathbf{x}_1$ and $\mathbf{x}_2$ are support vectors, and the decision boundary $\mathcal{H}$ is the perpendicular bisector of the segment $\mathbf{x}_1 \mathbf{x}_2$
- ▶ The normal vector $\mathbf{w}$ must point from $\mathbf{x}_1$ ($y_1 = -1$) towards $\mathbf{x}_2$ ($y_2 = +1$), so $\mathbf{w}$ is a positive multiple of $\mathbf{x}_2 - \mathbf{x}_1$:

$$\mathbf{x}_2 - \mathbf{x}_1 = \begin{pmatrix} 3 - 1 \\ 5 - 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \quad \implies \quad \mathbf{w} = \alpha \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad (\alpha > 0)$$

Example 1 – Scaling ($\mathbf{w}$, b) and computing the margin

Step 2 (Canonical equations): Since $\mathbf{x}_1 \in \mathcal{H}_{-1}$ and $\mathbf{x}_2 \in \mathcal{H}_{+1}$, we require

$$\mathbf{w}^T \mathbf{x}_1 + b = -1 \quad \text{and} \quad \mathbf{w}^T \mathbf{x}_2 + b = +1$$

- ▶ Subtracting the first equation from the second gives $\mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) = 2$:

$$\alpha \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \iff 10\alpha = 2 \iff \alpha = \frac{1}{5} \implies \mathbf{w} = \begin{pmatrix} 1/5 \\ 2/5 \end{pmatrix}$$

- ▶ Substituting $\mathbf{w}$ into $\mathbf{w}^T \mathbf{x}_1 + b = -1$ gives the offset b :

$$\frac{1}{5}(1) + \frac{2}{5}(1) + b = -1 \implies b = -1 - \frac{3}{5} = -\frac{8}{5}$$

Step 3 (Decision boundary and margin width):

- ▶ Decision boundary: $\frac{1}{5}x_1 + \frac{2}{5}x_2 - \frac{8}{5} = 0 \iff x_1 + 2x_2 - 8 = 0$ (passes through the midpoint $\mathbf{m} = (2, 3)^T$)
- ▶ Margin width: $\|\mathbf{w}\| = \sqrt{(1/5)^2 + (2/5)^2} = \frac{1}{\sqrt{5}}$, so $\gamma = \frac{2}{\|\mathbf{w}\|} = 2\sqrt{5} = \|\mathbf{x}_2 - \mathbf{x}_1\|$

Example 1 revisited – Solving via Lagrange multipliers

Recall Example 1: $\mathbf{x}_1 = (1, 1)^T$ ($y_1 = -1$) and $\mathbf{x}_2 = (3, 5)^T$ ($y_2 = +1$)

With multipliers $\alpha_1, \alpha_2 \geq 0$, form the Lagrangian for the hard-margin SVM:

$$L(\mathbf{w}, b, \alpha_1, \alpha_2) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^2 \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1]$$

Setting partial derivatives to zero gives the stationarity conditions:

- ▶ $\nabla_{\mathbf{w}} L = \mathbf{0} \implies \mathbf{w} = \sum_{i=1}^2 \alpha_i y_i \mathbf{x}_i = -\alpha_1 (1, 1)^T + \alpha_2 (3, 5)^T$
- ▶ $\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^2 \alpha_i y_i = -\alpha_1 + \alpha_2 = 0 \implies \alpha_1 = \alpha_2 = \alpha$

Therefore $\mathbf{w} = \alpha(\mathbf{x}_2 - \mathbf{x}_1) = \alpha(2, 4)^T = (2\alpha, 4\alpha)^T$

Example 1 revisited – Determining α , $\mathbf{w}$ and b

Since both points are support vectors ($\alpha > 0$), their margin constraints are active ($= 1$):

► At $\mathbf{x}_1$: $y_1(\mathbf{w}^T \mathbf{x}_1 + b) = -[2\alpha(1) + 4\alpha(1) + b] = 1 \implies 6\alpha + b = -1$

► At $\mathbf{x}_2$: $y_2(\mathbf{w}^T \mathbf{x}_2 + b) = 2\alpha(3) + 4\alpha(5) + b = 1 \implies 26\alpha + b = 1$

Subtracting the first equation from the second:

$$20\alpha = 2 \implies \alpha = \frac{1}{10}$$

Substituting $\alpha = 1/10$ gives the exact same parameters as the geometric approach:

$$\mathbf{w} = \begin{pmatrix} 2(1/10) \\ 4(1/10) \end{pmatrix} = \begin{pmatrix} 1/5 \\ 2/5 \end{pmatrix}, \quad b = -1 - 6 \left(\frac{1}{10} \right) = -\frac{8}{5}$$

Key insight: In general, $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ is a linear combination of only the support vectors (points with $\alpha_i > 0$)

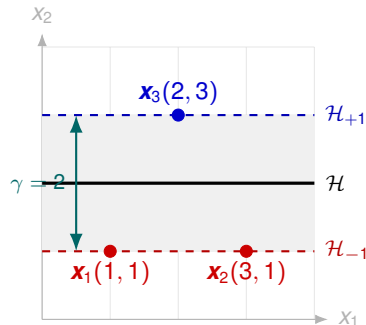
Example 2 – Hard-margin SVM for 3 points in $\mathbb{R}^2$

Problem: Find $(\mathbf{w}, b)$ and margin γ for:

- ▶ Class -1 : $\mathbf{x}_1 = (1, 1)^T$ and $\mathbf{x}_2 = (3, 1)^T$
- ▶ Class $+1$: $\mathbf{x}_3 = (2, 3)^T$

Step 1 (Direction of the boundary):

- ▶ Both $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to class -1 and are equidistant from $\mathbf{x}_3$, so all three points are support vectors
- ▶ Since $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{H}_{-1}$, the vector $\mathbf{x}_2 - \mathbf{x}_1 = (2, 0)^T$ lies along $\mathcal{H}_{-1}$
- ▶ By orthogonality,
 $\mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) = 2w_1 = 0 \implies w_1 = 0$, so
 $\mathbf{w} = (0, w_2)^T$



Example 2 – Solving the linear equations

Step 2 (Canonical equations): Substituting $\mathbf{w} = (0, w_2)^T$ into $\mathbf{w}^T \mathbf{x}_i + b = y_i$ for the support vectors gives

$$\text{at } \mathbf{x}_1 = (1, 1)^T \text{ and } \mathbf{x}_2 = (3, 1)^T : \quad w_2(1) + b = -1$$

$$\text{at } \mathbf{x}_3 = (2, 3)^T : \quad w_2(3) + b = +1$$

- ▶ Subtracting the first equation from the second gives $2w_2 = 2 \implies w_2 = 1$
- ▶ Substituting $w_2 = 1$ into $w_2 + b = -1$ gives $b = -2$, so

$$\mathbf{w} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{and} \quad b = -2$$

- ▶ Decision boundary: $x_2 - 2 = 0$ (the horizontal line $x_2 = 2$)
- ▶ Margin width: $\|\mathbf{w}\| = 1 \implies \gamma = \frac{2}{\|\mathbf{w}\|} = 2$

Solving hard-margin SVMs on larger sets with R

For larger datasets ($n \gg 2$) or in higher dimensions ($p > 2$), quadratic programming solvers find the optimal multipliers $\alpha_i \geq 0$ numerically

In R, the package `e1071` (interface to LIBSVM) provides `svm()`:

```
fit = svm(y ~ ., data = dat, kernel = "linear", cost = 1000, scale =  
        FALSE)
```

- ▶ `kernel = "linear"`: fits the linear decision boundary $\mathbf{w}^T \mathbf{x} + b = 0$
- ▶ `cost = 1000`: a large cost penalty enforces a hard margin (no margin violations)
- ▶ `scale = FALSE`: operates directly in raw feature coordinates

The fitted object contains:

- ▶ `fit$index`: indices of the support vectors ($\alpha_i > 0$)
- ▶ `fit$coefs`: the signed multipliers $\alpha_i y_i$ for each support vector
- ▶ $\mathbf{w} = \sum_{i \in \text{SV}} \alpha_i y_i \mathbf{x}_i$: computed via `drop(t(fit$coefs) %*% fit$SV)`
- ▶ Bias b : given by `-fit$rho`

Fitting a linear SVM in R – 50-point dataset

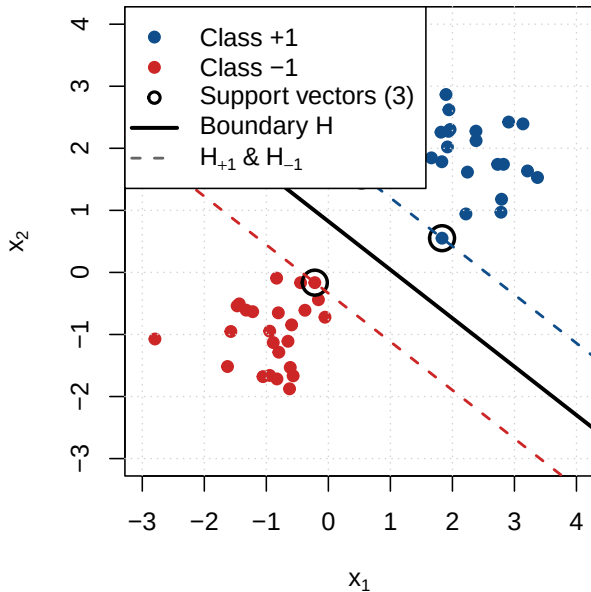
```
library(e1071)
set.seed(42); n = 25
X1 = matrix(rnorm(2 * n, mean = 2, sd = 0.6), n, 2)
X2 = matrix(rnorm(2 * n, mean = -1, sd = 0.6), n, 2)
dat = data.frame(x1 = c(X1[, 1], X2[, 1]), x2 = c(X1[, 2], X2[, 2]),
                 y = factor(c(rep(1, n), rep(-1, n))))
fit = svm(y ~ ., data = dat, kernel = "linear",
          cost = 1000, scale = FALSE)
w = drop(t(fit$coefs) %*% fit$SV)
b = -fit$rho
cat("Support vectors:", length(fit$index), "out of", nrow(dat), "\n")

## Support vectors: 3 out of 50

cat("w:", round(w, 4), " b:", round(b, 4),
    " gamma:", round(2 / sqrt(sum(w^2)), 4), "\n")

## w: 0.6734 0.8626  b: -0.7102  gamma: 1.8275
```

Hard-margin SVM on 50 points in R (e1071)



Supervised vs unsupervised linear methods

Geometry of hyperplanes and support vector machines

Least-squares classification

Framing classification as linear least squares

Can we fit a linear classifier $\hat{y}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$ using the method of least squares from Unit 03?

We ask the affine score $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$ to match the numeric class label $y_i \in \{-1, +1\}$ as closely as possible:

$$\mathbf{w}^T \mathbf{x}_i + b \approx y_i \quad \text{for } i = 1, \dots, n$$

To write this as a single matrix product, absorb the intercept b by defining the **augmented feature vector** and **augmented weight vector**:

$$\tilde{\mathbf{x}}_i = \begin{pmatrix} \mathbf{x}_i \\ 1 \end{pmatrix} \in \mathbb{R}^{p+1}, \quad \tilde{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix} \in \mathbb{R}^{p+1}$$

so that, by symmetry of the dot product ($\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}}$):

$$\mathbf{w}^T \mathbf{x}_i + b = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}}$$

We write $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}}$ so that the row vector $\tilde{\mathbf{x}}_i^T$ naturally matches the i -th row of the data matrix X (next slide).

Matrix formulation and normal equations

Stack the augmented rows $\tilde{\mathbf{x}}_i^T$ into the (augmented) **data matrix** $X \in \mathcal{M}_{n,p+1}$ (in regression and classification, often called the **design matrix** or **feature matrix**) and the labels into $\mathbf{y} \in \{-1, +1\}^n$:

$$X = \begin{pmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_n^T & 1 \end{pmatrix} \in \mathcal{M}_{n,p+1}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

Minimising the sum of squared errors $\|X\tilde{\mathbf{w}} - \mathbf{y}\|^2 = \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$ yields the **normal equations**:

$$X^T X \tilde{\mathbf{w}} = X^T \mathbf{y} \quad \implies \quad \tilde{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

- ▶ What we are explaining here **only works when we have more observations than features**: we require $n \geq p + 1$ and $\text{rank}(X) = p + 1$ for $X^T X \in \mathcal{M}_{p+1}$ to be invertible
- ▶ If $n < p + 1$ or if features are collinear, $X^T X$ is singular and $(X^T X)^{-1}$ does not exist

Example – 4-point least-squares classifier in $\mathbb{R}^2$

Problem: Find the least-squares linear classifier $(\mathbf{w}, b)$ for the $n = 4$ points:

- ▶ Class -1 ($y_1 = y_2 = -1$): $\mathbf{x}_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ and $\mathbf{x}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$
- ▶ Class $+1$ ($y_3 = y_4 = +1$): $\mathbf{x}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\mathbf{x}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Step 1 (Set up X and $\mathbf{y}$):

$$X = \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \in \mathcal{M}_{43}, \quad \mathbf{y} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \in \mathbb{R}^4$$

Example – Solving the normal equations by hand

Step 2 (Compute $X^T X$ and $X^T \mathbf{y}$):

$$X^T X = \begin{pmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \quad X^T \mathbf{y} = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}$$

Step 3 (Solve for $\tilde{\mathbf{w}}$): Since $X^T X = \text{diag}(2, 2, 4)$ is diagonal, its inverse is immediate:

$$\tilde{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/4 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \implies \mathbf{w} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad b = 0$$

The decision boundary is $\mathbf{w}^T \mathbf{x} + b = x_1 + x_2 = 0$ (the line $x_2 = -x_1$), which separates the two classes cleanly

Example – Fitted values, residuals and classifying new points

With $\tilde{\mathbf{w}} = (1, 1, 0)^T$ ($\mathbf{w} = (1, 1)^T$ and $b = 0$), let us check the fitted scores $\hat{\mathbf{y}} = X\tilde{\mathbf{w}}$ and residual vector $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = X\tilde{\mathbf{w}} = \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = \mathbf{y} \quad \Rightarrow \quad \mathbf{e} = \mathbf{y} - X\tilde{\mathbf{w}} = \mathbf{0}$$

- ▶ Because these 4 points happen to lie on the lines $x_1 + x_2 = -1$ and $x_1 + x_2 = +1$, the least-squares residual is zero!
- ▶ For a new observation $\mathbf{x}_{\text{new}} = \begin{pmatrix} 2 \\ -0.5 \end{pmatrix}$, we compute $\mathbf{w}^T \mathbf{x}_{\text{new}} + b = 2 - 0.5 = 1.5 > 0$, so $\hat{y}(\mathbf{x}_{\text{new}}) = \text{sgn}(1.5) = +1$

Sensitivity of least-squares classification to outliers

Why is least squares not always suitable for classification?

- ▶ The squared error $(\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$ penalises predictions that are **too far on the correct side** of the decision boundary
- ▶ Suppose a point with $y_i = +1$ lies deep inside $\mathcal{H}_+$ so that $\mathbf{w}^T \mathbf{x}_i + b = 10 > 0$: it is *correctly classified*, yet least squares assigns a large squared penalty $(10 - 1)^2 = 81$
- ▶ To shrink that large residual, least squares tilts and shifts the hyperplane towards the extreme point, which can push the boundary across points near the margin and cause misclassifications

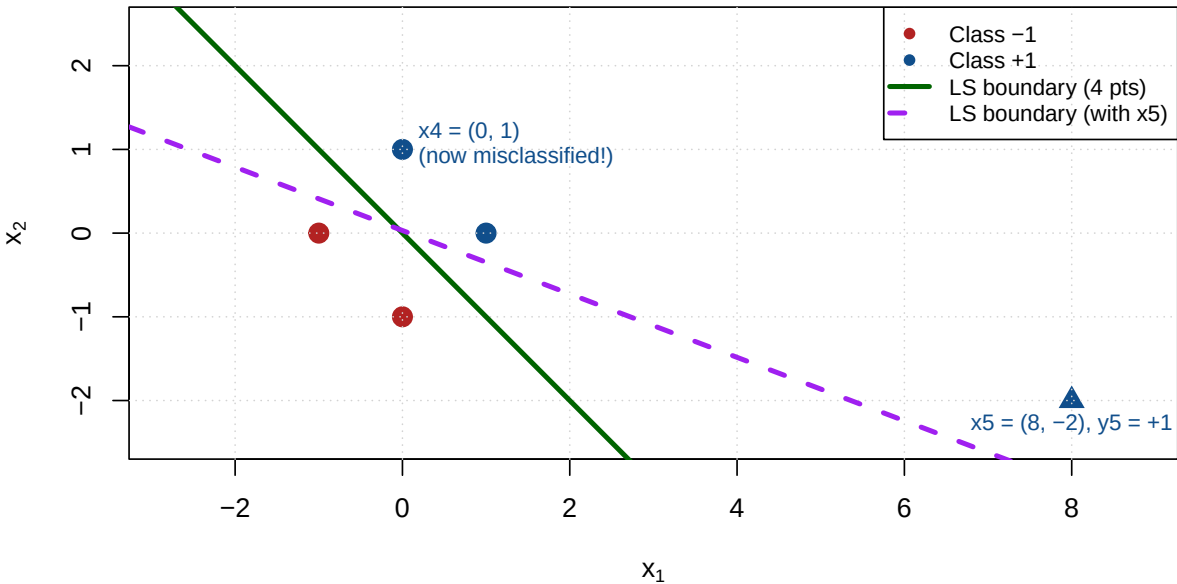
R: least-squares classifier and outlier sensitivity

Compare the 4-point least-squares boundary before and after adding a single “easy” Class +1 point at $\mathbf{x}_5 = (8, -2)^T$ (which lies well inside $x_1 + x_2 > 0$)

```
X4 = matrix(c(-1, 0, 1, 0, -1, 1, 1, 0, 1, 0, 1, 1),
            nrow = 4, byrow = TRUE)
y4 = c(-1, -1, 1, 1)
w_tilde_4 = solve(t(X4) %*% X4, t(X4) %*% y4)
# Add one outlier far on the correct (+1) side: x5 = (8, -2)^T
X5 = rbind(X4, c(8, -2, 1)); y5 = c(y4, 1)
w_tilde_5 = solve(t(X5) %*% X5, t(X5) %*% y5)
cbind(w_4pts = round(w_tilde_4, 3), w_5pts = round(w_tilde_5, 3))

##      [,1]    [,2]
## [1,]      1  0.433
## [2,]      1  1.142
## [3,]      0 -0.035
```

Sensitivity of least-squares classification to a far-side point



Ridge regularisation when $X^T X$ is singular

What if we have more features than observations ($n < p + 1$) or collinear features?

- ▶ In that case, $\text{rank}(X) < p + 1$, so $X^T X$ is singular and cannot be inverted
- ▶ To fix this, we add a **ridge penalty** with regularisation parameter $\lambda > 0$:

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^{p+1}} \left(\|X\tilde{\mathbf{w}} - \mathbf{y}\|^2 + \lambda \|\tilde{\mathbf{w}}\|^2 \right)$$

- ▶ Setting the gradient to zero yields the regularised normal equations:

$$(X^T X + \lambda \mathbb{I}) \tilde{\mathbf{w}} = X^T \mathbf{y} \quad \implies \quad \tilde{\mathbf{w}}_{\text{ridge}} = (X^T X + \lambda \mathbb{I})^{-1} X^T \mathbf{y}$$

- ▶ Adding $\lambda \mathbb{I}$ shifts all eigenvalues of $X^T X$ by $+\lambda > 0$: all its eigenvalues are strictly positive (we call such a matrix **positive definite**; we will discuss this in more detail later), which guarantees that it is always invertible

Summary – Comparing linear projection and classification methods

Method	Supervised?	Criterion optimised	Linear algebraic solution
PCA (Unit 05)	No	Max variance $\mathbf{w}^T \Sigma \mathbf{w}$	Top eigenvector of Σ
Hard-margin SVM	Yes	Max margin $\frac{2}{\ \mathbf{w}\ }$	Support vectors $\mathbf{w}^T \mathbf{x}_i + b = \pm 1$
Least squares	Yes	Min error $\ X\tilde{\mathbf{w}} - \mathbf{y}\ ^2$	$\tilde{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$