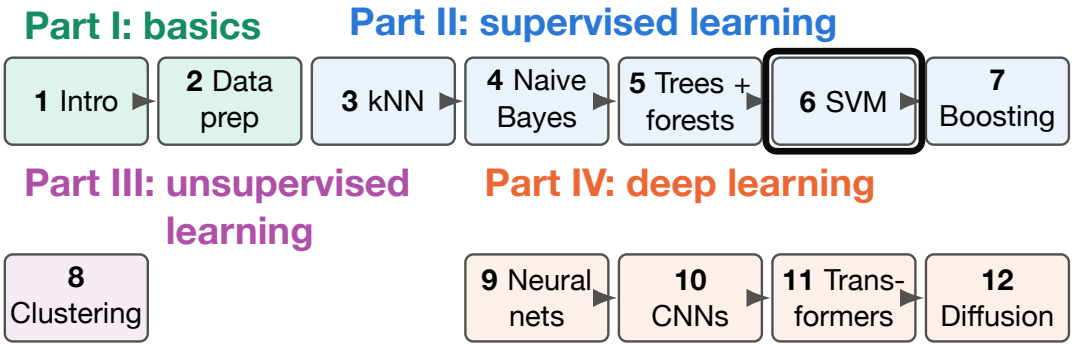
Support vector machines

229352 Lecture 6: support vector machines

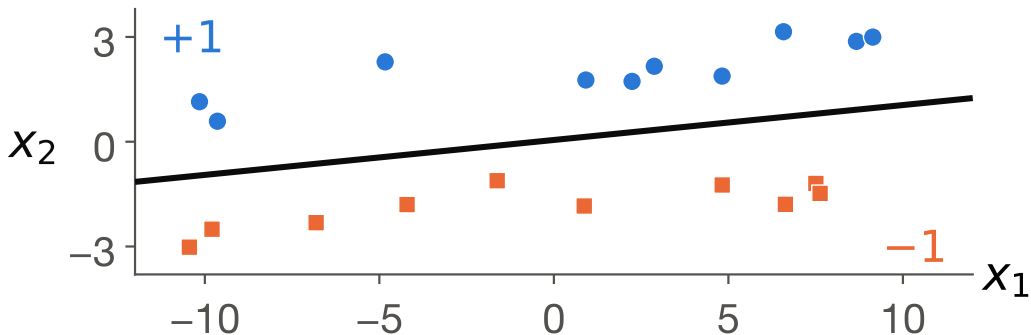
Course map


Part I: basics
Part II: supervised learning
1 Intro
2 Data prep
3 kNN
4 Naive Bayes
5 Trees + forests
6 SVM
7 Boosting
Part III: unsupervised learning
Part IV: deep learning
8 Clustering
9 Neural nets
10 CNNs
11 Trans-formers
12 Diffusion

Classify by a line

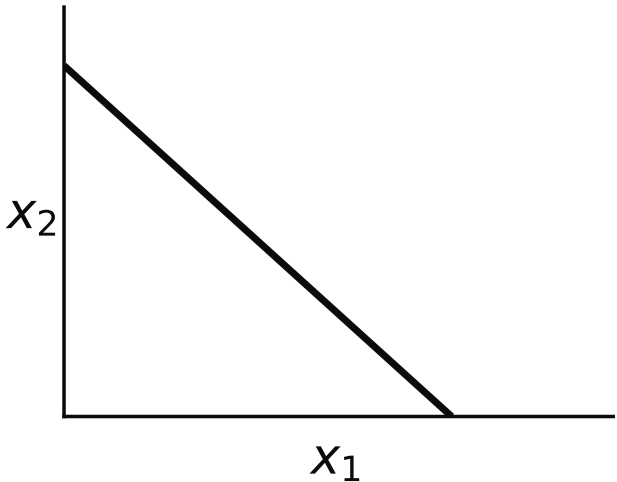
Naive Bayes (Lecture 4) compared two probabilities.

Today we separate the two classes by a line.



a new point gets the class of its side of the line

Line equation

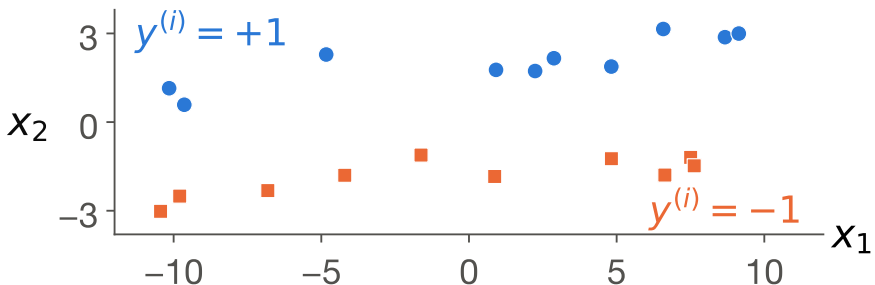


Notation

Points $x^{(i)} \in \mathbb{R}^d$ with labels $y^{(i)} \in \{-1, +1\}$, $i = 1, \ldots, n$

(labels -1 and $+1$ in this lecture, not 0 and 1)

Feature j of point i: $x_j^{(i)}$

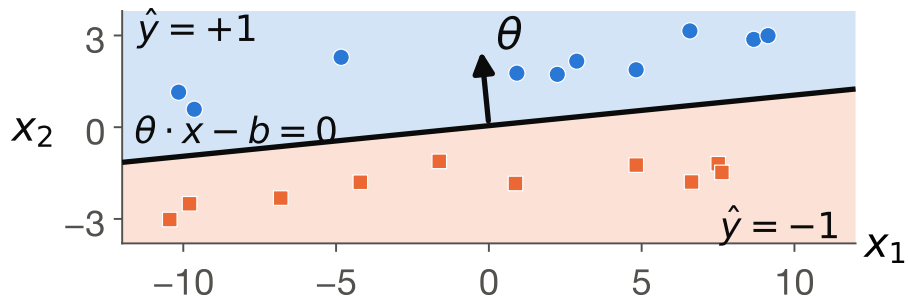


our simulated data: $n = 20$ points, $d = 2$ features

A linear classifier

$$\hat{y} = \begin{cases} +1 & \text{if } \theta \cdot x - b > 0 \\ -1 & \text{if } \theta \cdot x - b \leq 0 \end{cases} \qquad \theta \cdot x = \theta_1 x_1 + \cdots + \theta_d x_d$$

weights θ; **offset** b; **decision boundary**: $\theta \cdot x - b = 0$



θ is perpendicular to the line and points to the side $\hat{y} = +1$

Is point i classified correctly?

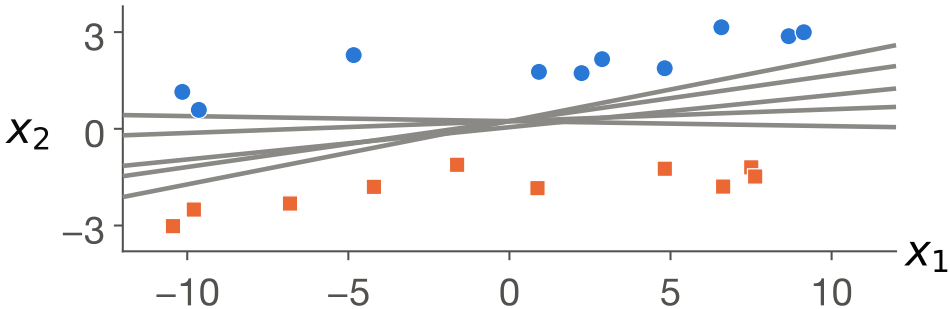
- $y^{(i)} = +1$ and $\theta \cdot x^{(i)} - b > 0$: correct
- $y^{(i)} = -1$ and $\theta \cdot x^{(i)} - b < 0$: correct
- In both cases $y^{(i)}$ and $\theta \cdot x^{(i)} - b$ have the same sign, so their product is positive.

$$y^{(i)}(\theta \cdot x^{(i)} - b) \begin{cases} > 0 & \text{correctly classified} \\ < 0 & \text{misclassified} \end{cases}$$

If the product equals 0, the point lies on the line, and the rule predicts -1 there.

Many lines separate the classes

A **separating line** satisfies $y^{(i)}(\theta \cdot x^{(i)} - b) > 0$ for every i: every point is on its correct side.



five of the many separating lines: which one is best?

The support vector machine

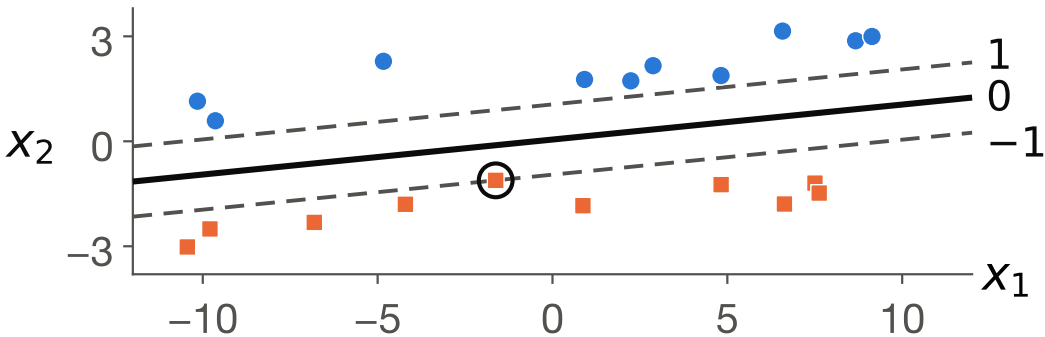
- The **support vector machine** (SVM) chooses the separating line with the widest **margin**, the empty band around the line.
- The SVM solves a convex problem (explained below), and standard solvers find the solution efficiently.
- We find the margin lines, then the largest margin width.

Step 1: construct the margin lines

- Take a separating line. Multiplying θ and b by the same positive number gives the same line $\theta \cdot x - b = 0$.
- Choose that number so that the points nearest the line have $y^{(i)}(\theta \cdot x^{(i)} - b) = 1$, and all other points a value greater than 1.
- The **margin lines**: $\theta \cdot x - b = +1$ and $\theta \cdot x - b = -1$. No data point lies between them; the band between them is the margin.
- **Support vectors**: the data points on the margin lines (for now; the dual form below extends this).

Margin lines of one separating line

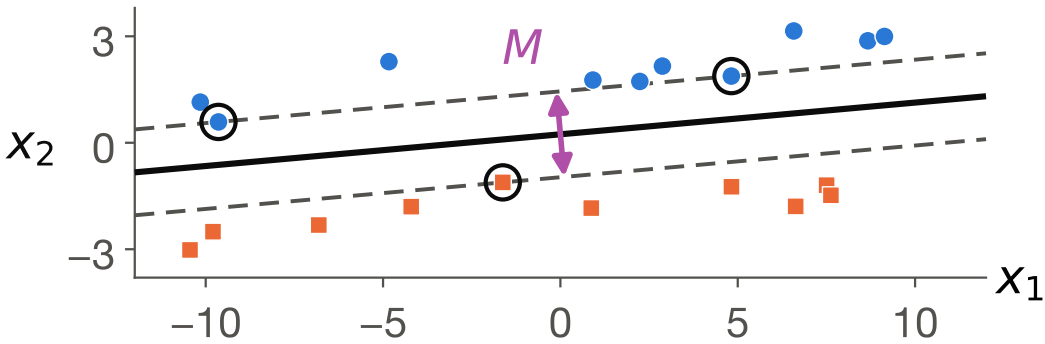
The line of the first slide, with the lines $\theta \cdot x - b = 1,\ 0,\ -1$:



circled: the only point on a margin line; margin width 2.00

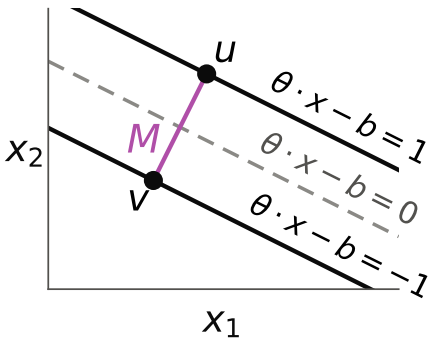
Step 2: maximize the margin width

Take the separating line whose margin lines are farthest apart.



We want to maximize the margin width M. What is M?

The width of the margin



Let u be a point on the margin line $\theta \cdot x - b = 1$, v a point on the margin line $\theta \cdot x - b = -1$, with $u - v$ perpendicular to both lines.

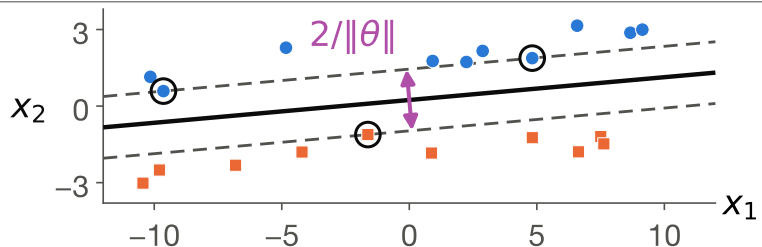
$M = \|u - v\|$, the length of $u - v$:

Hard-margin SVM

Hard margin: every point must lie on its own side of its margin line (no slack).

Find θ and b that minimize $\frac{1}{2}\|\theta\|^2$

subject to $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1$ for $i = 1, \ldots, n$



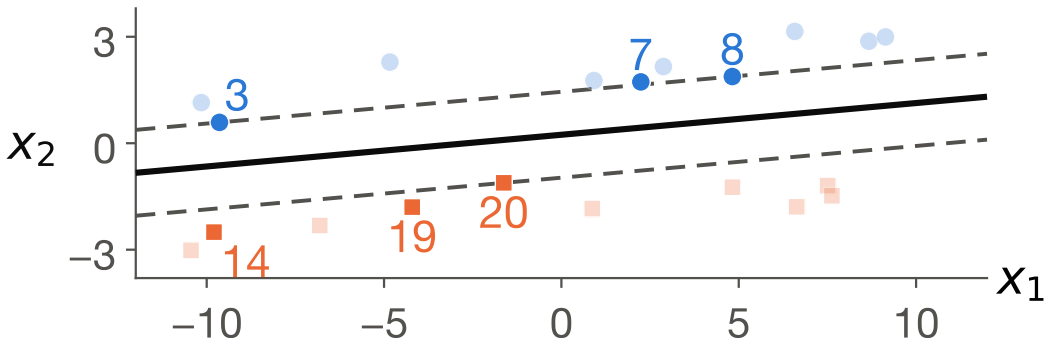
$\|\theta\| = \sqrt{\theta_1^2 + \cdots + \theta_d^2}$; the widest margin has the smallest $\|\theta\|$

A convex problem

- The objective $\frac{1}{2}\|\theta\|^2$ is **convex** (a bowl), and each constraint is linear in θ and b.
- When a separating line exists, this problem has exactly one best θ (the bowl has one lowest point), and standard solvers find that θ efficiently.
- The factor $\frac{1}{2}$ does not change the minimizer; the $\frac{1}{2}$ simplifies the formulas of the dual problem (a later slide).
- Our data: $\theta = (-0.07,\ 0.83)$, $b = 0.20$, 3 support vectors.

Six points of our data

We use six points: the three of each class nearest the SVM's line, $i \in \{3, 7, 8, 14, 19, 20\}$.

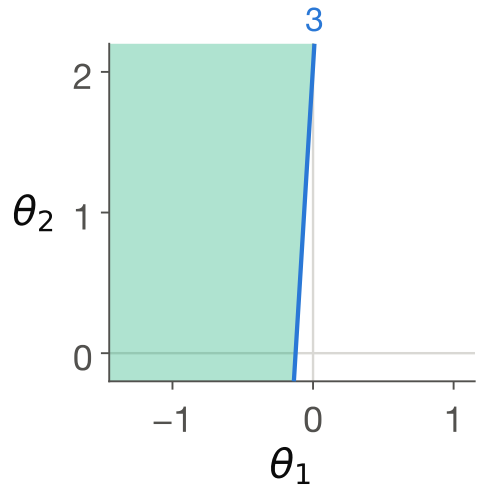


labels: the index i of each of the six points

The allowed region of θ for one point

We fix $b = 0.20$ (the SVM's value); only $\theta = (\theta_1, \theta_2)$ varies.

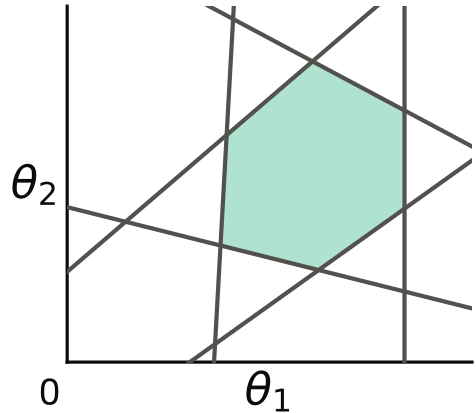
The pairs $\theta = (\theta_1, \theta_2)$ with $y^{(3)}(\theta \cdot x^{(3)} - b) \geq 1$ form a half-plane (aqua).



Six data points: a sketch

The pairs θ with $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1$ for all six points i form the intersection of six half-planes (aqua).

Sketch with six made-up constraints, not our data: each line is one constraint, and the region is a hexagon.



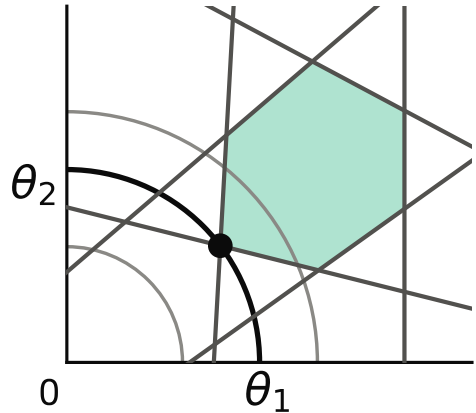
Minimization with constraints

Find the pair θ in the aqua region with the smallest $\|\theta\|^2$.

Each circle: one value of $\|\theta\|^2$.

The smallest circle that touches the region (black) gives the solution (dot).

Our six points, $b = 0.20$: the solution $\theta = (-0.07,\ 0.83)$ is the SVM's θ (the six include every support vector).



Primal and dual form

Primal: minimize $\frac{1}{2}\|\theta\|^2$ over θ, b subject to $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1$ for all i

gives the same θ as

Dual: find the **dual variables** $\alpha_1, \ldots, \alpha_n$ (one per data point) that maximize

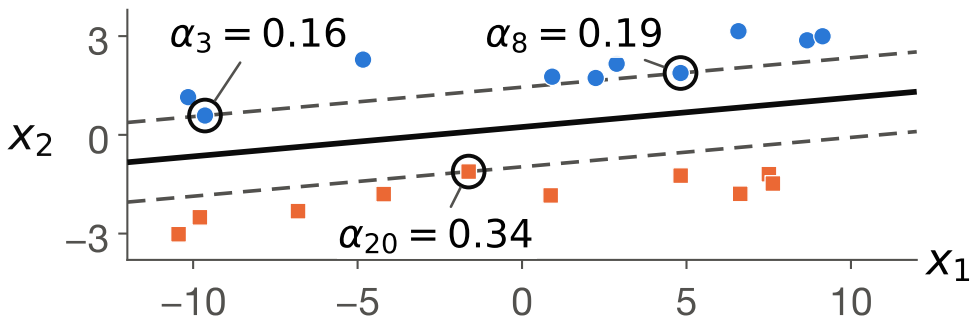
$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_\ell \alpha_i \alpha_\ell \, y^{(i)} y^{(\ell)} \, x^{(i)} \cdot x^{(\ell)}$ $(i, \ell = 1, \ldots, n)$

subject to $\alpha_i \geq 0$ for all i, and $\sum_i \alpha_i y^{(i)} = 0$

then $\theta = \sum_i \alpha_i y^{(i)} x^{(i)}$, $b = \theta \cdot x^{(\ell)} - y^{(\ell)}$ for any ℓ with $\alpha_\ell > 0$

Support vectors and α_i

$\alpha_i > 0$ only for the points on the margin lines (support vectors).
So $\theta = \sum_i \alpha_i y^{(i)} x^{(i)}$ depends on the support vectors only.



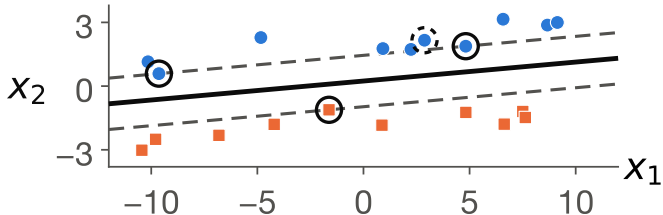
our data: 3 support vectors; $\alpha_i = 0$ for the other 17 points

Summary: hard-margin SVM

- Classifier: $\hat{y} = +1$ if $\theta \cdot x - b > 0$, else -1; the margin lines are $\theta \cdot x - b = \pm 1$.
- Hard-margin SVM: minimize $\frac{1}{2}\|\theta\|^2$ subject to $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1$ for all i; this gives the separating line with the widest margin.
- A convex problem: one best θ, found efficiently.
- Dual: $\theta = \sum_i \alpha_i y^{(i)} x^{(i)}$, with $\alpha_i > 0$ only for the support vectors, the points on the margin lines.

Checkpoint 1

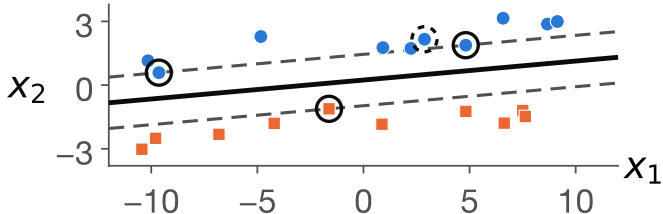
We delete the dashed-circle point and train again.



What happens to the line?

Checkpoint 1

We delete the dashed-circle point and train again.



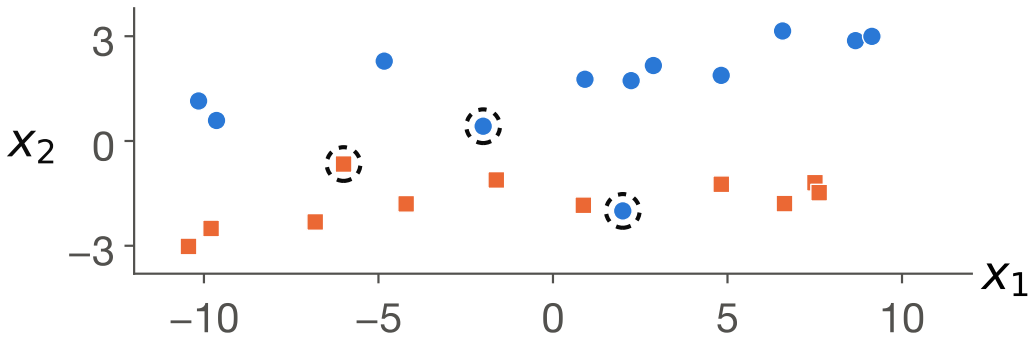
What happens to the line?

Nothing: θ and b stay the same

($\alpha_i = 0$ for the deleted point i).

Is a separating line always possible?

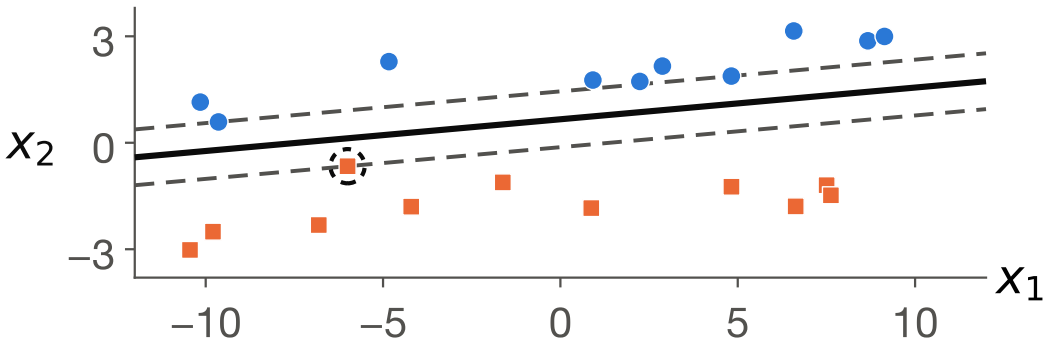
Our 20 points and three more:



dashed circles: three new points; no separating line exists

Is a separating line always a good idea?

Our 20 points and one more (dashed circle): still separable.



one added point reduces the margin width from 2.41 to 1.57

Slack variables

- Allow point i inside the margin or on the wrong side with a **slack** (variable) $r_i \geq 0$:

$$y^{(i)}(\theta \cdot x^{(i)} - b) \;\geq\; 1 - r_i$$

- The smallest slack that satisfies this:
  $r_i = \max\{0,\; 1 - y^{(i)}(\theta \cdot x^{(i)} - b)\}$. With this r_i:
  - $r_i = 0$: on or beyond its margin line, correct side
  - $0 < r_i < 1$: inside the margin, correct side
  - $r_i \geq 1$: on $\theta \cdot x - b = 0$ or on the wrong side

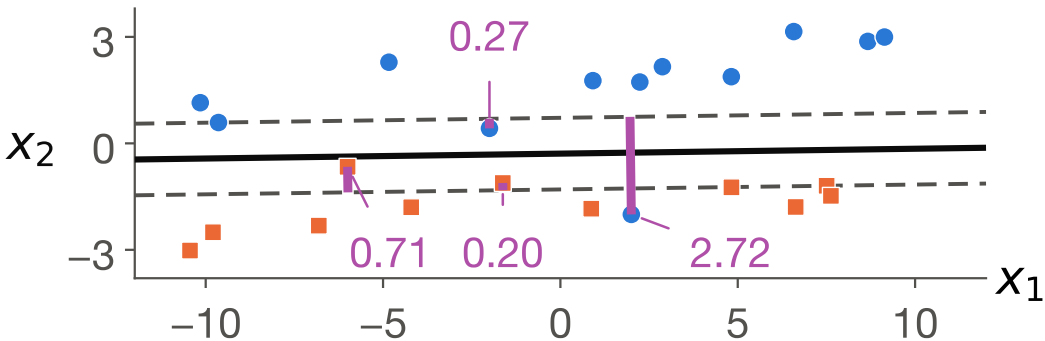
Soft-margin SVM

Find θ, b and $r_1, \ldots, r_n$ that minimize $\frac{1}{2}\|\theta\|^2 + C\sum_i r_i$
subject to $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1 - r_i$ and $r_i \geq 0$ for all i

- $C > 0$: the cost of one unit of slack, set before training.
- A wide margin has a small $\|\theta\|$ but puts more points inside the margin, and each such point needs slack: C balances margin width against slack.
- Still a convex problem.

Slack in our data: $C = 1$

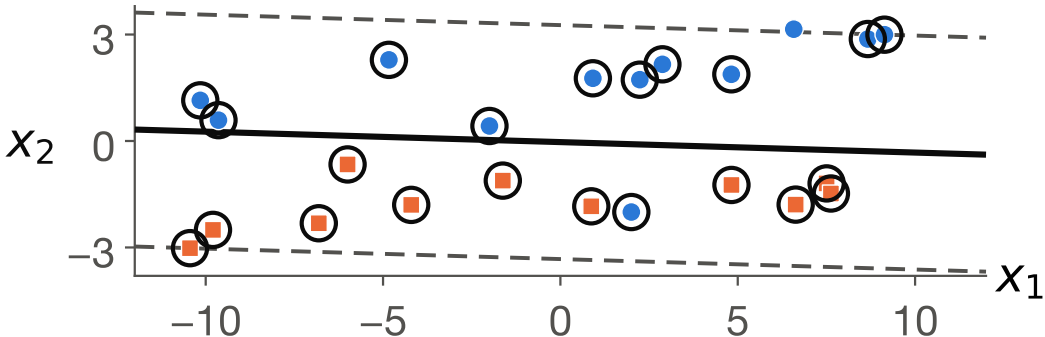
The soft-margin SVM on the 23 points, with $C = 1$:



purple: each $r_i > 0$, with a segment to its margin line

Soft margin: $C = 0.01$

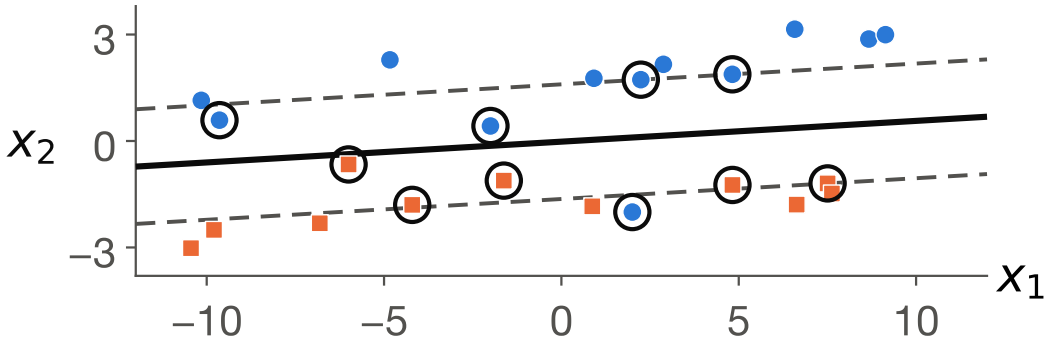
Circled: support vectors, the points with $\alpha_i > 0$ (defined on the dual slide below).



22 support vectors; margin width 6.59

Soft margin: $C = 0.1$

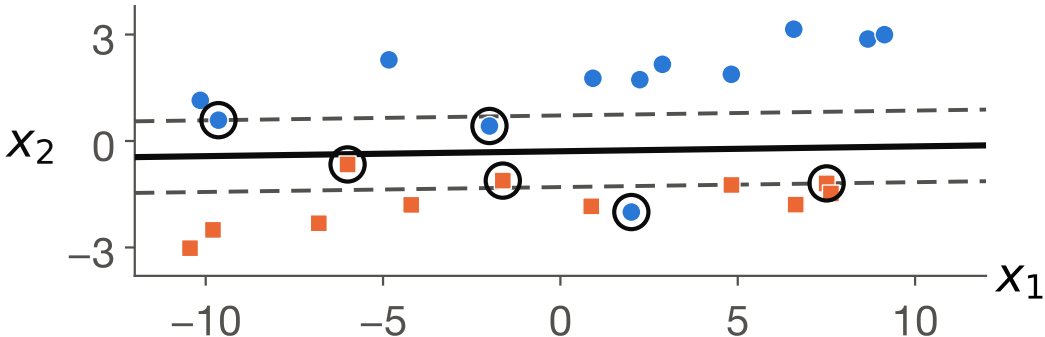
Circled: support vectors, the points with $\alpha_i > 0$ (defined on the dual slide below).



10 support vectors; margin width 3.23

Soft margin: $C = 1$

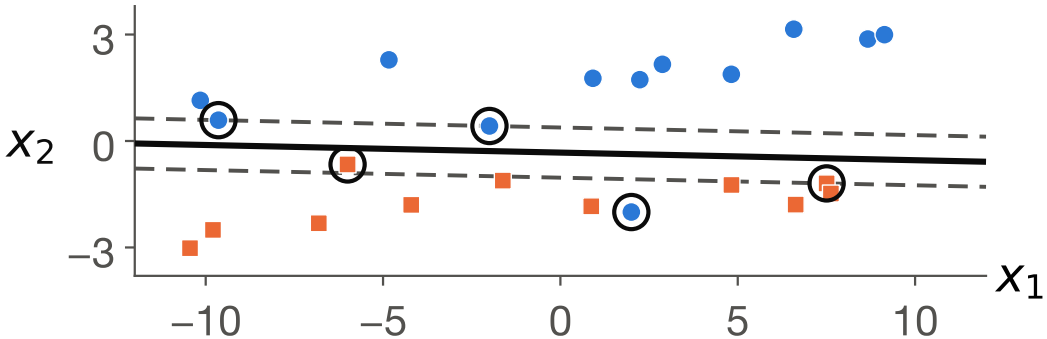
Circled: support vectors, the points with $\alpha_i > 0$ (defined on the dual slide below).



6 support vectors; margin width 2.02

Soft margin: $C = 100$

Circled: support vectors, the points with $\alpha_i > 0$ (defined on the dual slide below).



5 support vectors; margin width 1.41

Choosing C

- C is chosen before training and then stays fixed.
- Choose C on a validation set or by cross-validation (Lecture 3): fit for each C in a list, keep the C with the smallest validation error.
- $C = 0$ does not work: $\theta = 0$ then minimizes $\frac{1}{2}\|\theta\|^2$, which gives no line.

Dual form of the soft-margin SVM

Dual: find $\alpha_1, \ldots, \alpha_n$ that maximize
$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_\ell \alpha_i \alpha_\ell \, y^{(i)} y^{(\ell)} \, x^{(i)} \cdot x^{(\ell)}$
subject to $0 \leq \alpha_i \leq C$ for all i, and $\sum_i \alpha_i y^{(i)} = 0$

- This is the hard-margin dual with the constraint $\alpha_i \leq C$ added. Then $\theta = \sum_i \alpha_i y^{(i)} x^{(i)}$, and the support vectors are the points with $\alpha_i > 0$; each has $y^{(i)}(\theta \cdot x^{(i)} - b) \leq 1$:
  - $0 < \alpha_i < C$: on a margin line
  - $\alpha_i = C$: on a margin line, inside the margin, or misclassified
- $b = \theta \cdot x^{(\ell)} - y^{(\ell)}$ for any ℓ with $0 < \alpha_\ell < C$.

Checkpoint 2

On our 23 points, the SVM has margin width 2.02 at $C = 1$ and 1.41 at $C = 100$. We train with $C = 10$. Is the margin wider than 2.02, between 1.41 and 2.02, or narrower than 1.41?

Checkpoint 2

On our 23 points, the SVM has margin width 2.02 at $C = 1$ and 1.41 at $C = 100$. We train with $C = 10$. Is the margin wider than 2.02, between 1.41 and 2.02, or narrower than 1.41?

Between: 1.54.

A larger C never widens the margin.

Summary: soft-margin SVM

- A slack $r_i \geq 0$ lets point i lie inside the margin ($0 < r_i < 1$), or on the line or the wrong side ($r_i \geq 1$).
- Soft-margin SVM: minimize $\frac{1}{2}\|\theta\|^2 + C\sum_i r_i$ subject to $y^{(i)}(\theta \cdot x^{(i)} - b) \geq 1 - r_i,\ r_i \geq 0$.
- A larger C makes slack cost more: the best θ trades margin width for less slack, so a larger C never gives a wider margin. Our data: margin width 6.59 at $C = 0.01$ and 1.41 at $C = 100$.
- $C > 0$ is set before training and chosen by validation.

Nonlinear boundaries

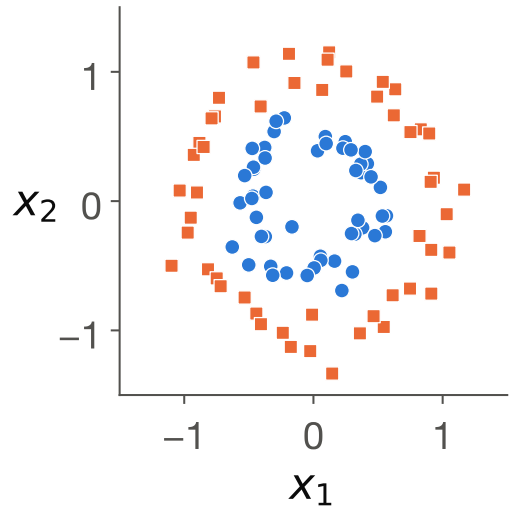
$x = (x_1, x_2), \; y \in \{-1, +1\}$

inner circle $+1$, outer ring -1

No line separates the two classes.

How do we get a curved decision boundary?

simulated: scikit-learn's `make_circles`, 100 points



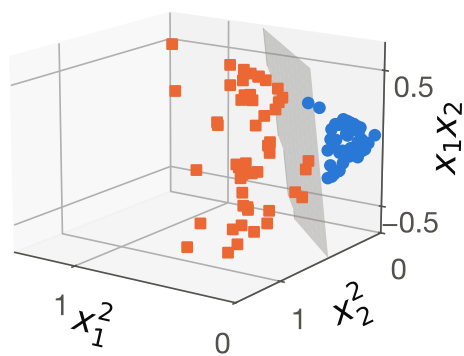
Adding new features

New features $x_1^2,\ x_2^2,\ x_1x_2$: the **feature map**

$\Phi(x_1, x_2) = (x_1, x_2, x_1^2, x_2^2, x_1x_2)$

Right: the points in the three new features.

A plane (gray) separates the classes. The plane comes from a linear SVM on the last three coordinates of $\Phi(x)$ only.



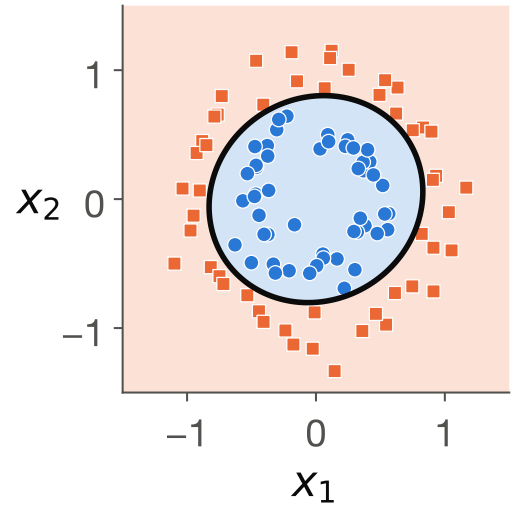
The boundary in (x_1, x_2)

The plane of the previous slide:

$$9.9\,x_1^2 + 10.5\,x_2^2 - 1.5\,x_1 x_2 = 6.7$$

In (x_1, x_2), this equation gives an ellipse (black).

A plane in the new features is a curve in (x_1, x_2).



Adding new features in d dimensions

We start with d features: $x = (x_1, \ldots, x_d)$.

We add the squares and the products of two different features:

$$x_1^2,\ x_2^2,\ \ldots,\ x_d^2 \qquad\qquad x_1x_2,\ x_1x_3,\ \ldots,\ x_{d-1}x_d$$

The feature map:

$$\Phi(x) = (x_1, \ldots, x_d,\ x_1^2, \ldots, x_d^2,\ x_1x_2, \ldots, x_{d-1}x_d)$$

The dimension of $\Phi(x)$

$$\Phi(x) = (x_1, \ldots, x_d,\ x_1^2, \ldots, x_d^2,\ x_1x_2, \ldots, x_{d-1}x_d)$$

What is the dimension of $\Phi(x)$?

The dual needs only dot products

Recall the dual form of the soft-margin SVM:

find $\alpha_1, \ldots, \alpha_n$ that maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_\ell \alpha_i \alpha_\ell \, y^{(i)} y^{(\ell)} \left(x^{(i)} \cdot x^{(\ell)} \right)$$

subject to $0 \leq \alpha_i \leq C$ for all i, and $\sum_i \alpha_i y^{(i)} = 0$

The data enter only through the dot products $x^{(i)} \cdot x^{(\ell)}$.

Kernel trick

Train on the feature-map vectors $\Phi(x^{(i)})$ instead:

find $\alpha_1, \ldots, \alpha_n$ that maximize
$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_\ell \alpha_i \alpha_\ell \, y^{(i)} y^{(\ell)} \left(\Phi(x^{(i)}) \cdot \Phi(x^{(\ell)}) \right)$
subject to $0 \leq \alpha_i \leq C$ for all i, and $\sum_i \alpha_i y^{(i)} = 0$

For some maps Φ, we can compute $\Phi(x^{(i)}) \cdot \Phi(x^{(\ell)})$ without writing out Φ (the **kernel trick**).

A **kernel** is a function $k(x, z) = \Phi(x) \cdot \Phi(z)$ of two inputs x and z, for some map Φ; a kernel measures how similar x and z are.

Computing the dot product

In 2D, with $x = (x_1, x_2)$, we use a second map, also called Φ from here on:

$\Phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$, and $\Phi(z)$ the same way for $z = (z_1, z_2)$.

What is $\Phi(x) \cdot \Phi(z)$?

The kernel $(1 + x \cdot z)^2$

For $x = (x_1, \ldots, x_d)$, with one $\sqrt{2}x_j x_m$ for each pair $j < m$:

$$\Phi(x) = (1,\ \sqrt{2}x_1, \ldots, \sqrt{2}x_d,\ x_1^2, \ldots, x_d^2,\ \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_{d-1}x_d)$$

$$\begin{aligned}\Phi(x) \cdot \Phi(z) &= 1 + 2\sum_j x_j z_j + \sum_j x_j^2 z_j^2 + 2\sum_{j<m} x_j x_m z_j z_m \\ &= 1 + 2\, x \cdot z + (x \cdot z)^2 = (1 + x \cdot z)^2\end{aligned}$$

MNIST: the dimension of $\Phi(x)$



a 28×28 image, as in MNIST

$x = (x_1, \ldots, x_{784})$: the 784 pixels

Kernel $(1 + x \cdot z)^2$: $\Phi(x)$ has dimension 308,505.

So $\theta = \sum_i \alpha_i y^{(i)} \Phi(x^{(i)})$ has dimension 308,505 as well.

The dual needs only $(1 + x^{(i)} \cdot x^{(\ell)})^2$: a dot product of 784 numbers per pair.

Do we need $\Phi(x)$ after all?

To classify a new point x, we need only the sign of $\theta \cdot \Phi(x) - b$:

$$\begin{aligned}\theta \cdot \Phi(x) - b &= \sum_i \alpha_i y^{(i)} \, \Phi(x^{(i)}) \cdot \Phi(x) - b \\ &= \sum_i \alpha_i y^{(i)} \, (1 + x^{(i)} \cdot x)^2 - b\end{aligned}$$

No Φ is needed: only dot products of the original vectors.

The sum runs over the support vectors only ($\alpha_i > 0$).

Summary: the kernel trick

- A feature map Φ adds features, such as squares and products; a line or plane in the new features is a curve in the old ones.
- The dual and the classifier use the data only through dot products $\Phi(x^{(i)}) \cdot \Phi(x^{(\ell)})$.
- Kernel trick: a kernel computes these dot products from $x^{(i)}$ and $x^{(\ell)}$ directly, for example $(1 + x^{(i)} \cdot x^{(\ell)})^2$.

Kernel SVM

1. **Feature map**: $x \mapsto \Phi(x)$.
2. **Learning**: solve the dual problem with $\Phi(x^{(i)}) \cdot \Phi(x^{(\ell)})$ in place of $x^{(i)} \cdot x^{(\ell)}$, and $0 \leq \alpha_i \leq C$.
3. **Classification**: for a new point x,

$$\hat{y} = \begin{cases} +1 & \text{if } \sum_i \alpha_i y^{(i)} \, \Phi(x^{(i)}) \cdot \Phi(x) - b > 0 \\ -1 & \text{otherwise} \end{cases}$$

Kernel SVM with any kernel

Any kernel k (a function $k(x, z) = \Phi(x) \cdot \Phi(z)$) can be used; Φ is never written out.

1. **Learning**: solve the dual problem with $k(x^{(i)}, x^{(\ell)})$ in place of $x^{(i)} \cdot x^{(\ell)}$, and $0 \leq \alpha_i \leq C$.

2. **Classification**: for a new point x,

$$\hat{y} = \begin{cases} +1 & \text{if } \sum_i \alpha_i y^{(i)} \, k(x^{(i)}, x) - b > 0 \\ -1 & \text{otherwise} \end{cases}$$

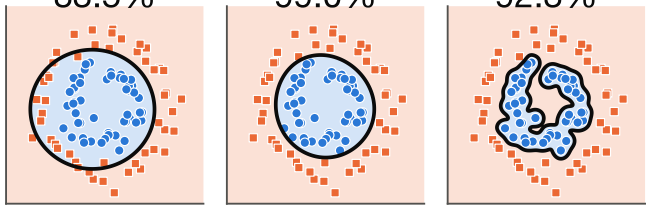
We compute b as in the soft-margin dual, using k.

Examples of kernels

- Polynomial of degree p: $k(x, z) = (1 + x \cdot z)^p$
- **Radial basis function** (RBF, Gaussian) kernel, scale $\sigma > 0$: $k(x, z) = \exp\big(-\|x - z\|^2/(2\sigma^2)\big)$
  In scikit-learn the parameter is $\gamma = 1/(2\sigma^2)$, so $k(x, z) = \exp(-\gamma\|x - z\|^2)$.
- Sigmoid, parameters η, ν: $k(x, z) = \tanh(\eta\, x \cdot z + \nu)$

The RBF kernel on the circles: $C = 1$

	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 100$
training accuracy	86.0%	100.0%	100.0%
test accuracy	83.5%	99.0%	92.3%



$\gamma = 100$ fits the training points exactly (100.0%), but its test accuracy (92.3%) is lower than at $\gamma = 1$ (99.0%).

The three γ are illustrative; in practice γ is chosen by validation.

Summary: support vector machines

- The SVM solves a convex problem: hard margin, or soft margin with the cost C.
- The dual gives the same θ and uses the data only through dot products; the support vectors are the points with $\alpha_i > 0$, and only those points enter θ.
- Nonlinear boundaries: map the data to more features $\Phi(x)$.
- Kernel SVM: a kernel $k(x, z)$ replaces $\Phi(x) \cdot \Phi(z)$; choose C and the kernel's parameters (γ for RBF) by validation.

SVM in scikit-learn

```
from sklearn.datasets import make_circles
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

X, y = make_circles(n_samples=300, noise=0.1, factor=0.5,
                    random_state=0)  # 300 points; slides: 100
y = 2 * y - 1  # labels -1, +1
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, random_state=0)  # 25% of the points for testing
linear = SVC(kernel="linear", C=1.0).fit(X_tr, y_tr)
rbf = SVC(kernel="rbf", gamma=1.0, C=1.0).fit(X_tr, y_tr)
# test accuracy; a line cannot separate the circles
print(linear.score(X_te, y_te), rbf.score(X_te, y_te))
# -> 0.3867 1.0000
print(rbf.support_vectors_.shape)  # -> (60, 2)
```

After fitting:

`linear.coef_` is θ, and `linear.intercept_` is $-b$.