Support Vector Machines

Introduction

Support vector machines (SVMs) are intended for the binary classification scenario in which there are only two classes

The SVMs are a generalization of a simple classifier called the *maximal margin classifier* which is relevant for classes that can be separated by a linear boundary

Hyperplanes

Hyperplanes

In a p-dimensional space, a *hyperplane* is a flat affine subspace of dimension $p-1$
So,
in a 2-dimensional space, the hyperplane is a line
in a 3-dimensional space, the hyperplane is a plane

In 2-dimensions, the hyperplane is defined by the equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0 \qquad (1)$$

Then, any point $X = (X_1, X_2)$ that satisfies (1) lays on the line (and hence the hyperplane)

In p-dimensions, equation (1) generalises to

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \qquad (2)$$

We say that (2) defines the hyperplane

Hyperplanes

Consistent with the 2-dimensional example, if the point $X = \left(X_1, \dots, X_p\right)$ in p-dimensional space satisfies (2) then it lies on the hyperplane

What happens X if doesn't satisfy (2)?

There are two possibilities:

Point X_a satisfies:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \qquad (4)$$

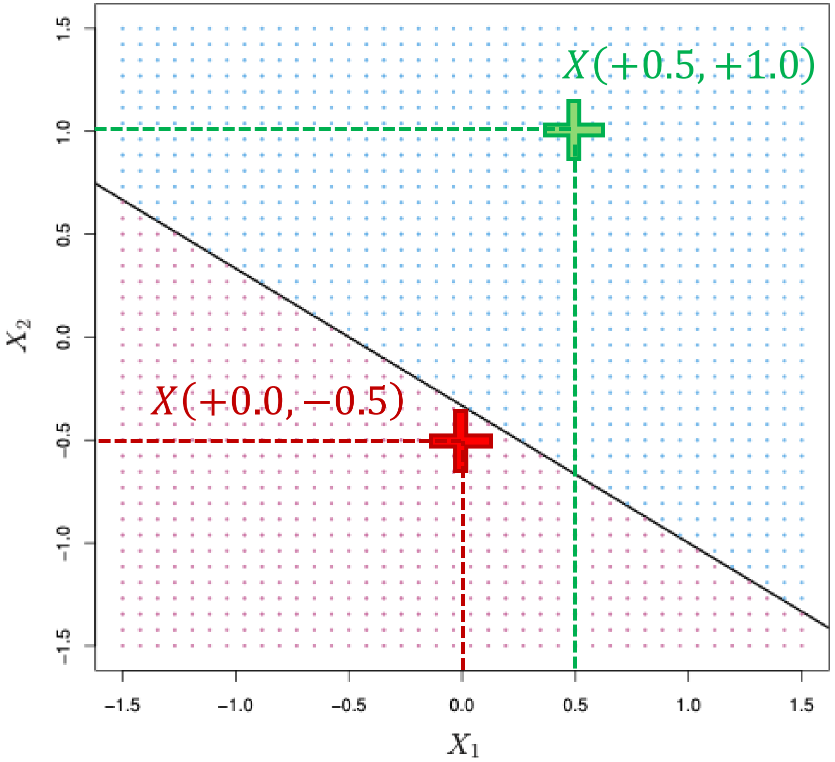
Point X_b satisfies

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \qquad (5)$$

In this case, X_a and X_b lie on opposite sites of the hyperplane

Hyperplanes

For example, the figure below shows the 2-dimensional hyperplane for $1 + 2X_1 + 3X_2 = 0$



$$X(+0.5, +1.0) = 1 + 2 \cdot 0.5 + 3 \cdot 1.0 = +6.0$$

$$X(+0.0, -0.5) = 1.0 + 0.0 \cdot 0.5 + 3.0 \cdot -0.5 = -1.5$$

Hyperplanes

Now, suppose we have $\mathbf{n}$ training observations in a p-dimensional space.

These observations can be represented in an $n \times p$ matrix X

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \quad \cdots, \quad x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}, \tag{5}$$

From the 2-dimensional case example, we see that these observations will fall into one of two classes according to which side of the hyperplane they are located.

Mathematically, we can state this as

$$y_0, y_1 \cdots y_n \in \{-1, +1\} \tag{6}$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{2}$$

Hyperplanes

So, if we can find a hyperplane satisfying the condition.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \qquad (2)$$

Then we have a classification system that can classify new test observations according to where they are located relative to the hyperplane.

So, for a test observation $\boldsymbol{x}^*$ we have

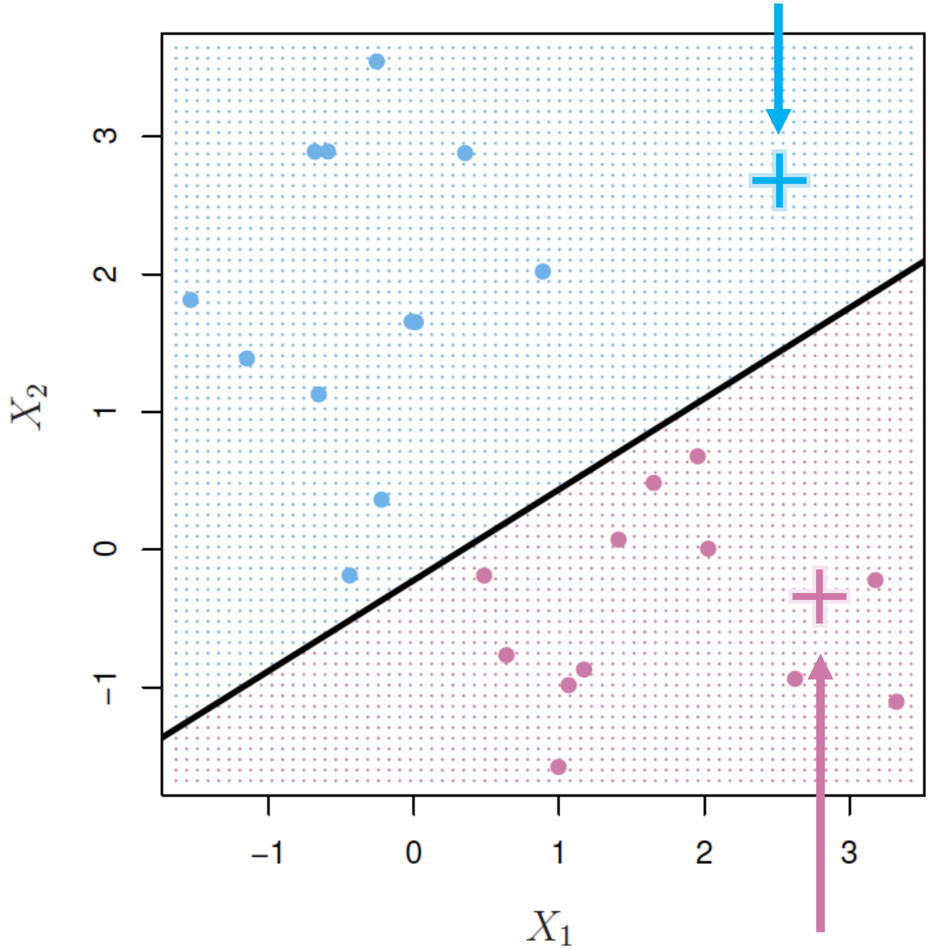
$$f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^* = 0 \qquad (6)$$

Then, if

$$f(x^*) = \begin{cases} +1\,;\ \text{assign to class } +1 \\ -1\,;\ \text{assign to class } -1 \end{cases}$$

Hyperplanes

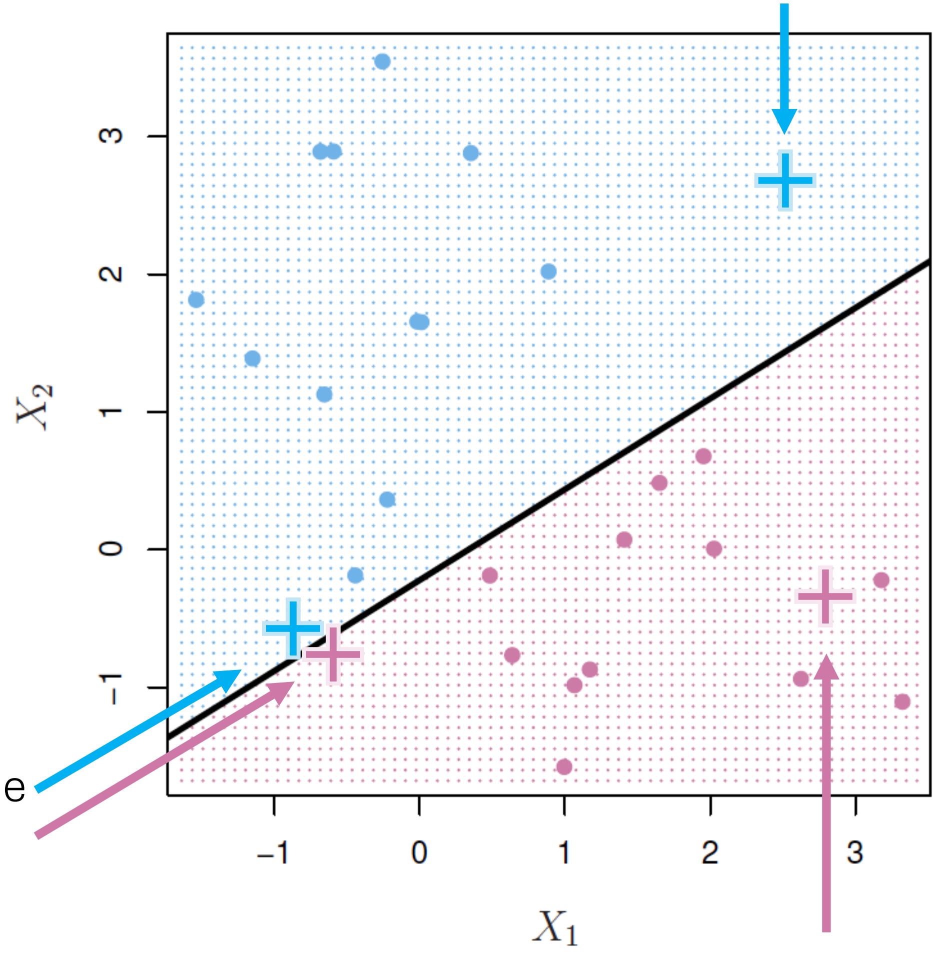
if x^* falls this side of the hyperplane,
assign to the blue class



if x^* falls this side of the hyperplane,
assign to the purple class

Hyperplanes

if x^* falls this side of the hyperplane, assign to the blue class



Both these observations are close to the hyperplane, so we should have less confidence in their assignment

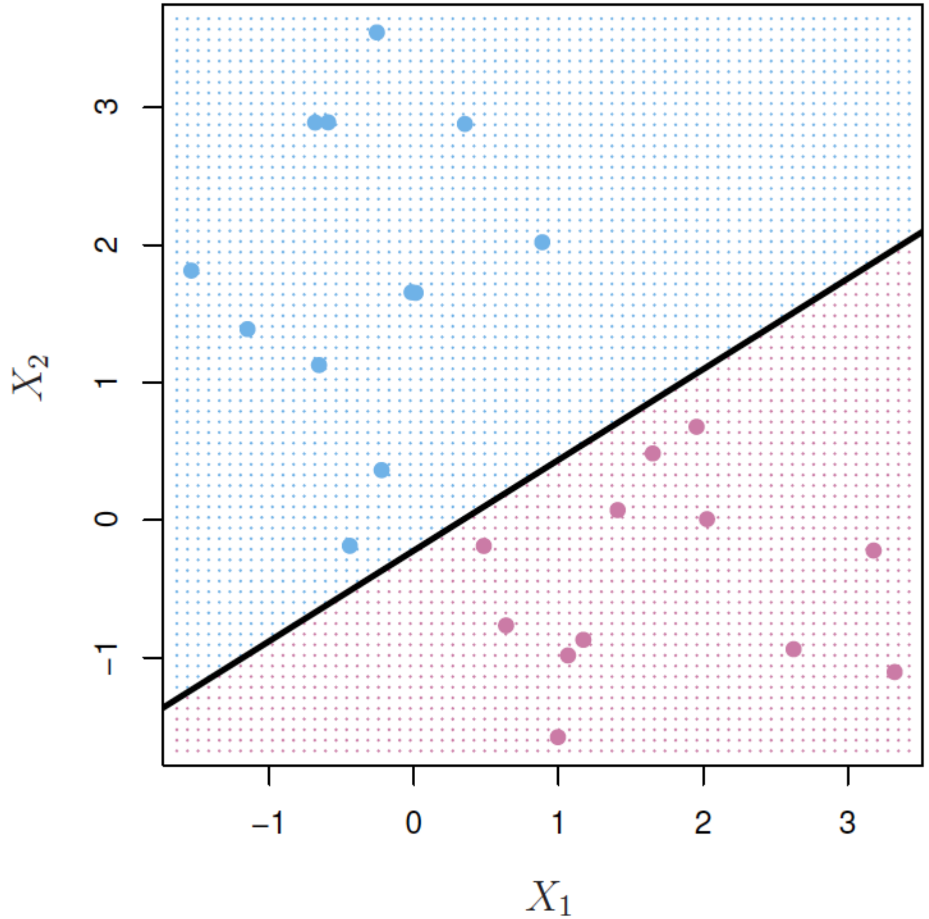
if x^* falls this side of the hyperplane, assign to the purple class

the Maximal Margin Classifier

Maximal Margin Classifier

The maximal Margin Classifier provides a way to identify a “best” hyperplane

The figure on the last slide shows the hyperplane we chose to classify our observations



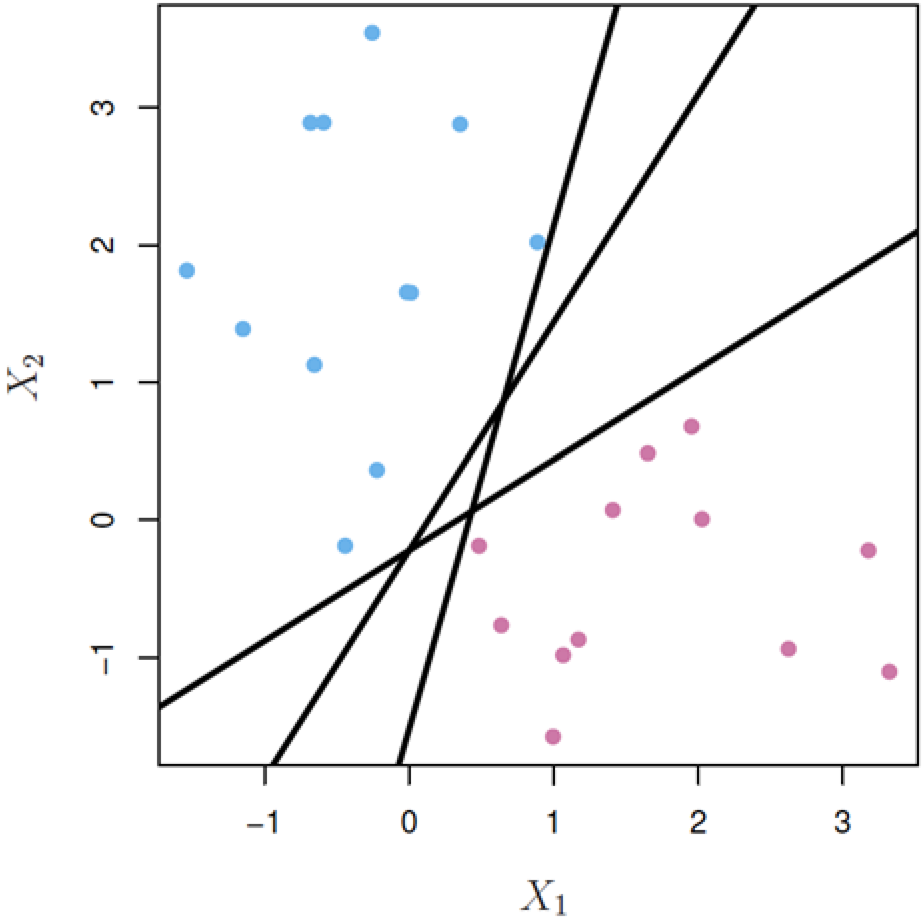
Maximal Margin Classifier

The maximal Margin Classifier provides a way to identify a “best” hyperplane

The figure on the last slide shows the hyperplane we chose to classify our observations

However, this is only one of an infinite number of possibilities?

How did we choose this one?



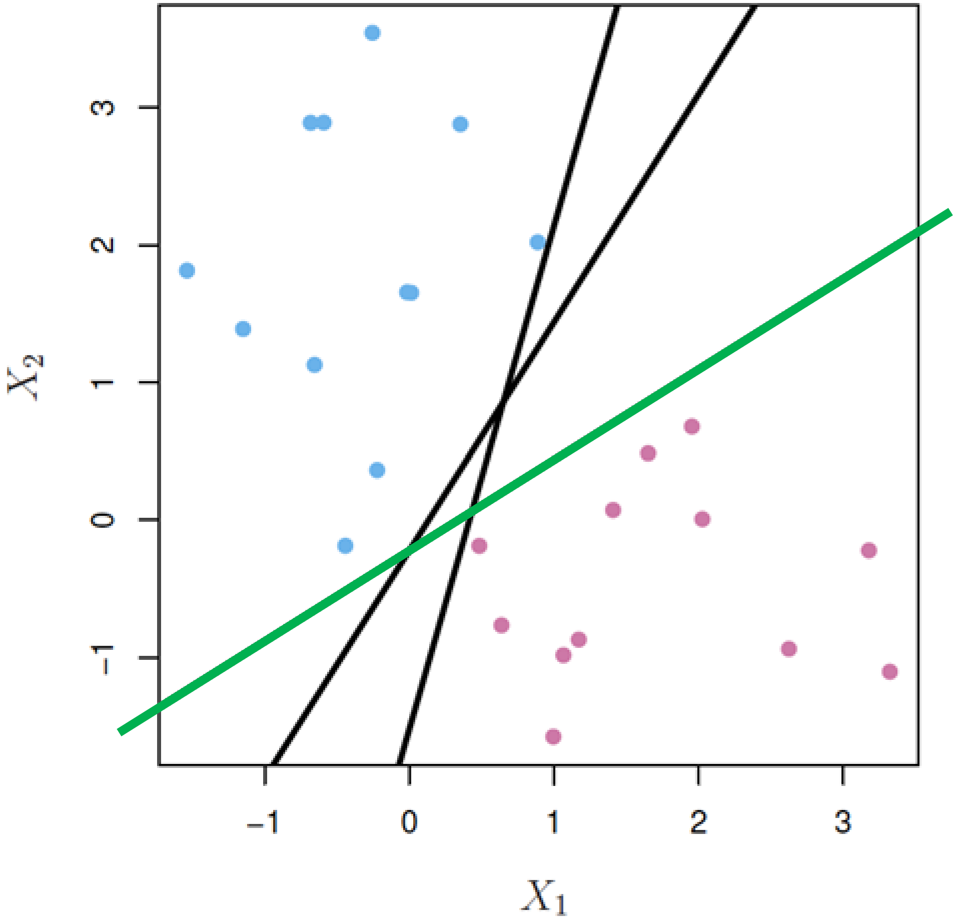
Maximal Margin Classifier

The maximal Margin Classifier provides a way to identify a “best” hyperplane

The figure on the last slide shows the hyperplane we chose to classify our observations

However, this is only one of an infinite number of possibilities?

How did we choose this one?



Maximal Margin Classifier

The maximal Margin Classifier selects the separating hyperplane that is furthest from the training observations. i.e., for each candidate hyperplane, we can calculate the perpendicular distance from each point to the hyperplane.

Then, for a set of training observations $x_1, \ldots, x_n \in \mathbb{R}^{\mathrm{p}}$ and class labels $y_1, \ldots, y_n \in \{-1,1\}$ we seek the hyperplane that optimises

$$\underset{\beta_0,\beta_1,\ldots\beta_p,M}{maximize} \; M \tag{7}$$

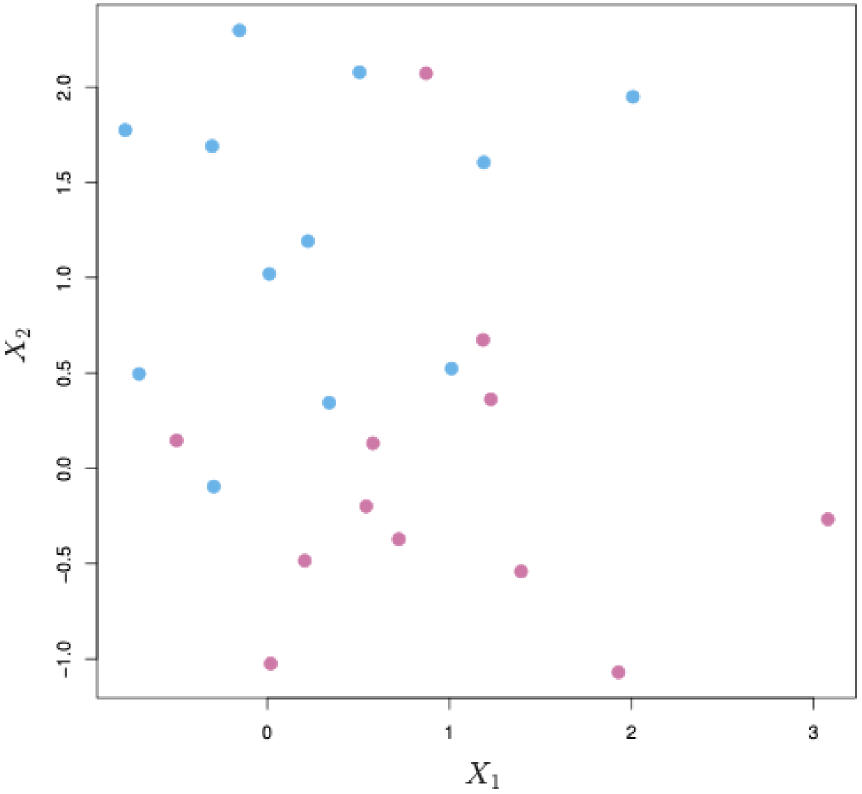
$$Subject\ to \sum_{j=1}^{p} \beta_j^2 = 1 \tag{8}$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M \quad \forall \; i = 1, \ldots, n \tag{9}$$

Support Vector Classifiers

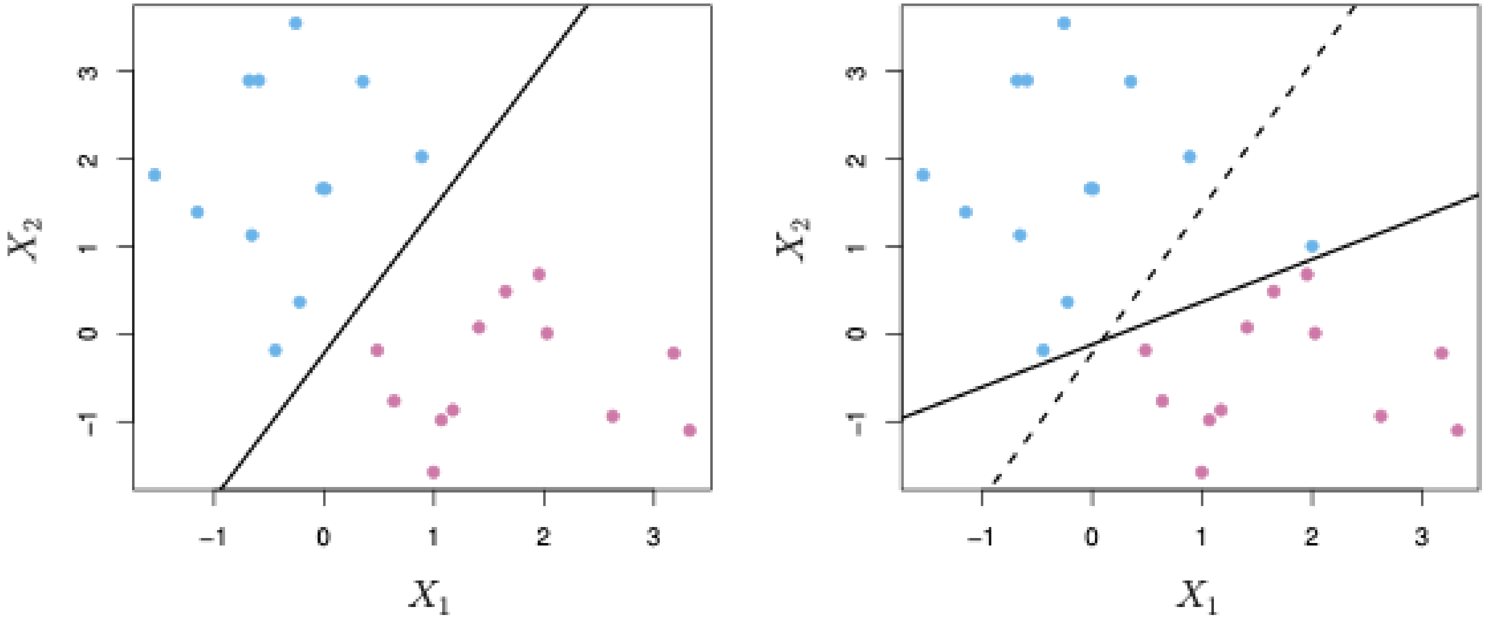
Support Vector Classifier

The limitation of the Maximal Margin Classifier is that it can only find a separable hyperplane. For example, its use is not feasible for the following training data.



Support Vector Classifier

Another issue is that the identified separating hyperplane may be sensitive to the training data.



The training data differs by a single point, but the identified hyperplane is notably different between the two datasets. Also, many of the training data are very close to the hyperplane

Support Vector Classifier

The Support Vector Classifier extends the concept of the Maximal Margin Classifier to seek a hyperplane that almost separates the two classes.

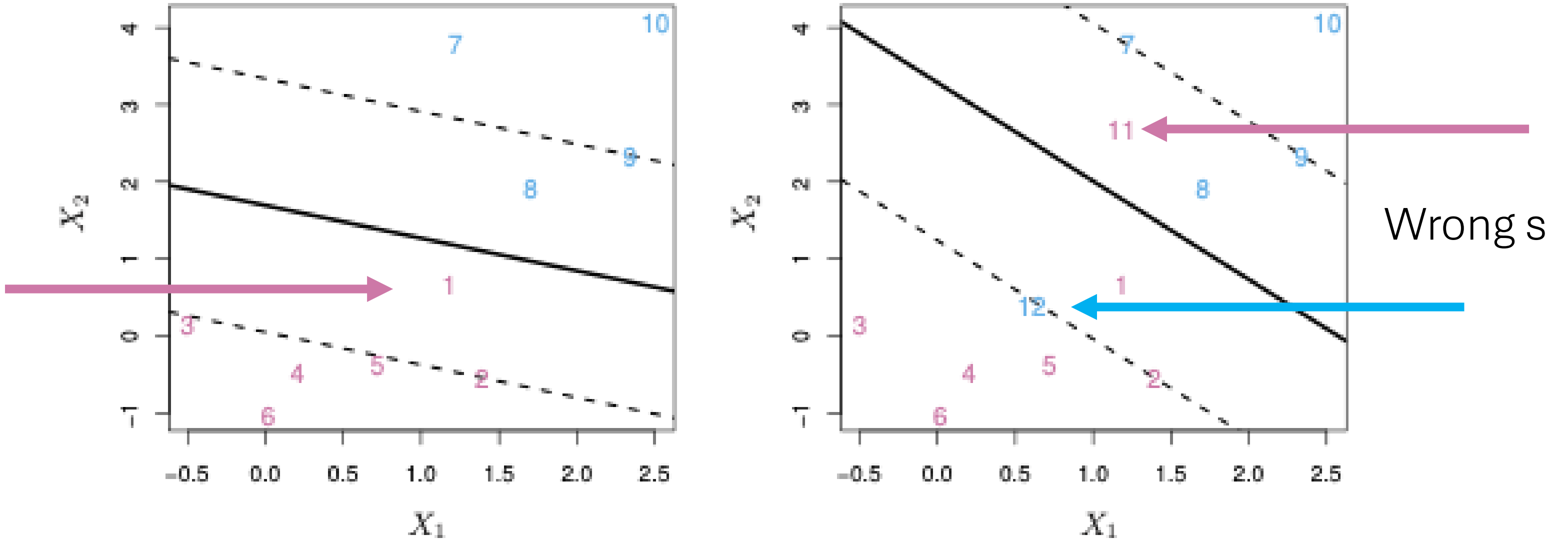
We seek a balance between performance (i.e., correct classification of training data) and reduced sensitivity (to individual training data)

i.e., we misclassify a few training observations to do a better job classifying the remaining observations.

Rather than seeking the largest possible margin so that every observation is both on the correct side of the hyperplane and the margin, we allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

We say the margin is soft since it can be violated by some of the training observations.

Support Vector Classifier



Wrong side of the margin

Wrong side of hyperplane

Support Vector Classifier

For the same set of training observations $x_1, \ldots, x_n \in \mathbb{R}^{\mathrm{p}}$ and class labels $y_1, \ldots, y_n \in \{-1,1\}$ the optimisation problem becomes

$$\underset{\beta_0,\beta_1,\ldots\beta_p,M}{maximize} \; M \qquad (10)$$

$$Subject\ to \sum_{j=1}^{p} \beta_j^2 = 1 \qquad (11)$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M(1 - \epsilon_i) \quad \forall\ i = 1, \ldots, n \qquad (12)$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \qquad (13)$$

C is a non-negative tuning parameter, M is the margin width and $\epsilon_1, \ldots, \epsilon_n$ are slack variables (that allow observations to be on the wrong side of the margin or hyperplane)

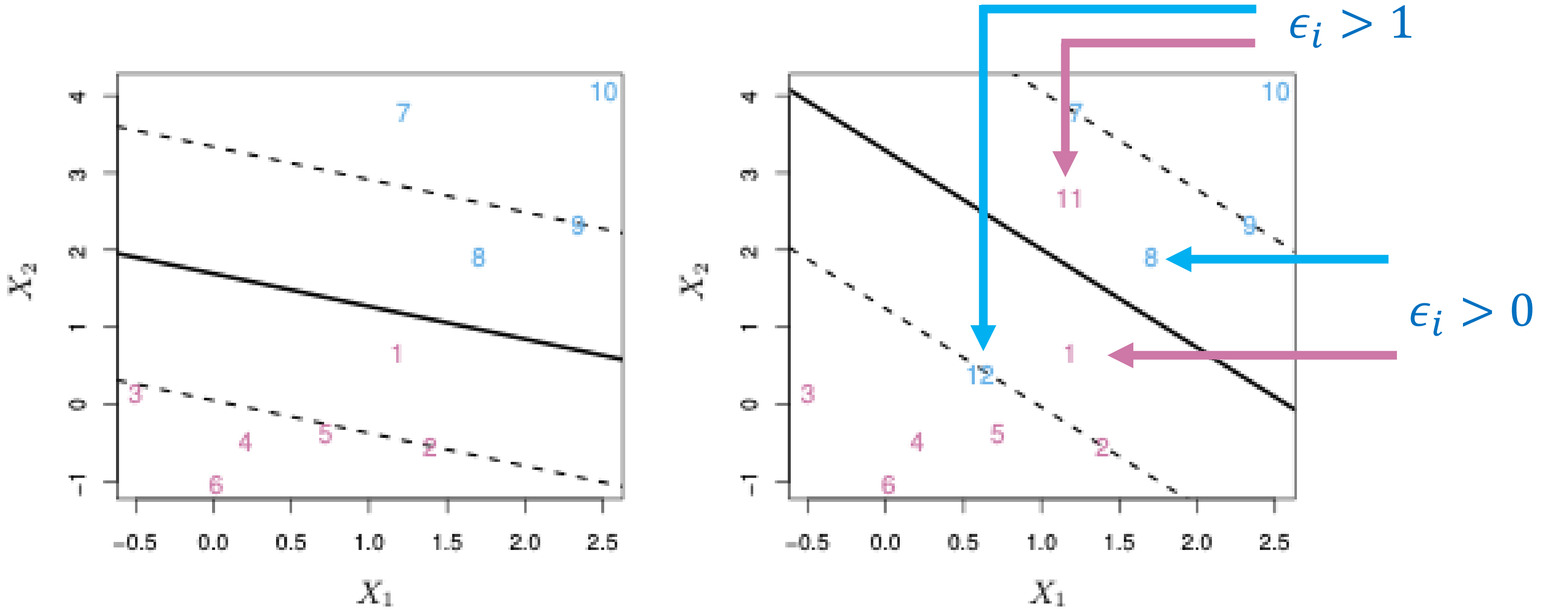
Support Vector Classifier

The slack variable tells us where the i^{th} observation is located relative to both the hyperplane and the margin.

if $\epsilon_i = 0$ the i^{th} observation is on the correct side of the margin

if $\epsilon_i > 0$ the i^{th} observation is on the wrong side of the margin

if $\epsilon_i > 1$ the i^{th} observation is on the wrong side of the hyperplane



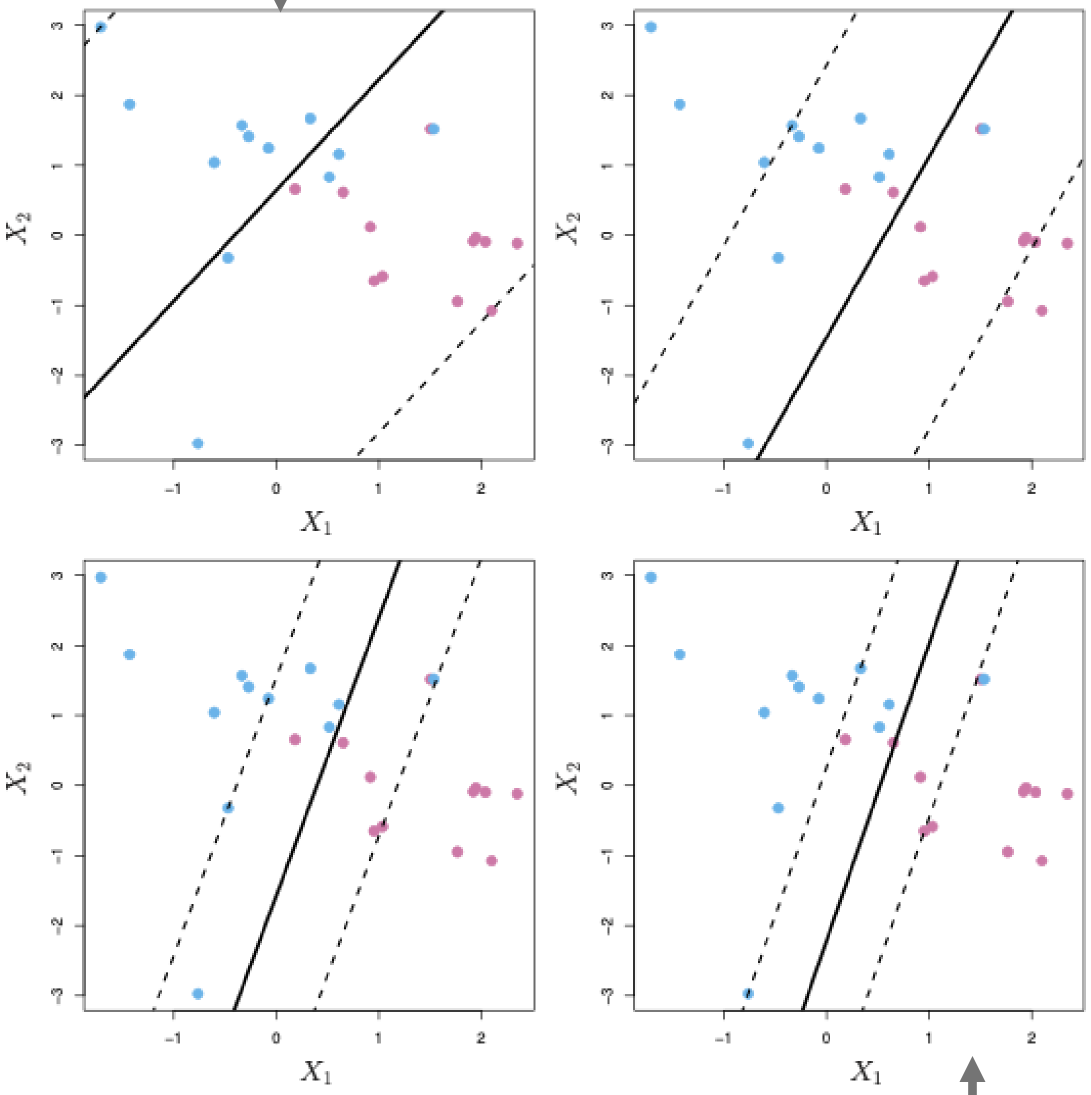
Support Vector Classifier

The tuning parameter C is defined as

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$

It tells us how much slack will be tolerated when identifying the optimal hyperplane, i.e., there can be no more than C observations on the wrong side of the hyperplane

High C , many observations on the wrong side of the margin



Low C , few observations on the wrong side of the margin. Margin narrower

Support Vector Classifier

It turns out that the observations that are on the margin or «violate» it (i.e., are on the wrong side) are the only ones that affect the choice of the hyperplane

So, the high C/large margin case (top left) has many support vectors.

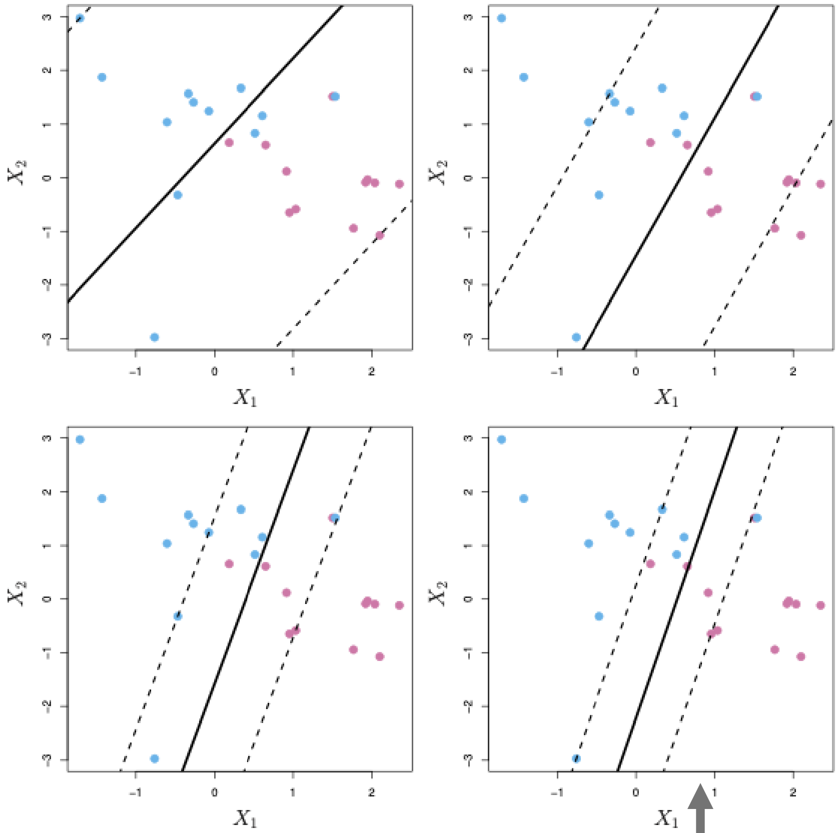
Conversely, a low C/large margin (bottom right) has few support vectors.

Seek a balance between bias and variance

bias : error introduced by model

variance : uncertainty due to dataset

High C , many observations on the wrong side of the margin



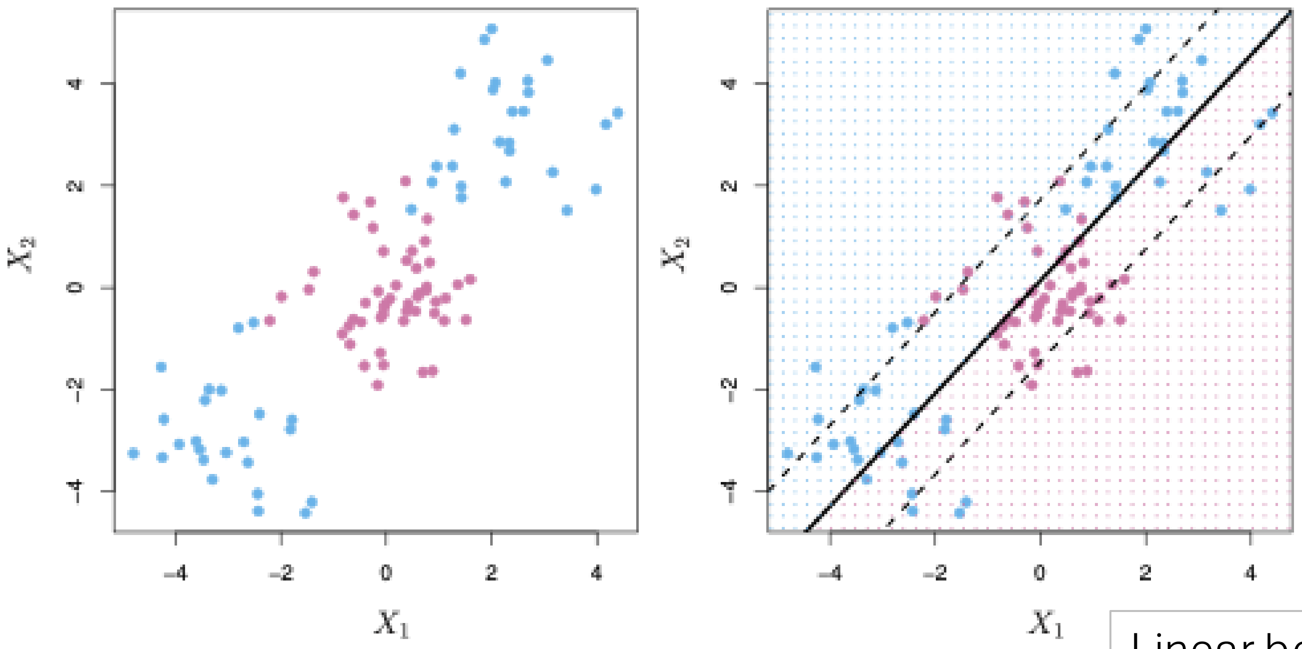
Low C , few observations on the wrong side of the margin. Margin narrower

Support Vector Machines

Support Vector Machine

The limitation of the Support Vector Classifier is that it can only handle linear boundaries.

The Support Vector Machine allows the possibility of non-linear boundaries



Linear boundary not appropriate here

Support Vector Machine

We have the same set of training observations $x_1, \dots, x_n \in \mathbb{R}^{\mathrm{p}}$ and class labels $y_1, \dots, y_n \in \{-1,1\}$ but now, instead of finding a linear hyperplane satisfying

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \qquad (2)$$

We introduce higher order terms.

For example, for a second order polynomial, instead of fitting a classifier to

$$X_1, X_2 \cdots, X_p,$$

We fit to $2\mathrm{p}$ features

$$X_1, X_1^2, X_2, X_2^2, \dots, X_p, X_p^2$$

Support Vector Machine

And the optimisation problem becomes

$$\underset{\beta_0, \beta_1, \beta_1^2, \beta_2, \beta_2^2, \ldots \beta_p, \beta_p^2, \epsilon_1, \ldots, \epsilon_n, M}{\text{maximise}} M \qquad (14)$$

$$Subject\ to\ y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j1}\, x_{ij} + \sum_{j=1}^{p} \beta_{j2}\, x_{ij}^2 \right) \geq M(1 - \epsilon_i) \qquad (15)$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \qquad \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1 \qquad (16)$$

In the original feature space, the decision boundary is non-linear, but in the enlarged feature it is linear

Support Vector Machine

This is the same approach used by SVMs. However, rather than performing the translation into the higher order feature space, it uses a kernel function to measure the similarity between two observations in the new space.

It also turns out that this comparison can be reduced to taking the dot product of the transformed points

$$\langle \mathbf{u} \cdot \mathbf{v} \rangle = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (1)(3) + (2)(4) = 3 + 8 = 11$$

And here, for two observations $x_i, x_{i'}$ this can be written as

$$\langle x_i \cdot x_{i'} \rangle = \sum_{j=1}^{p} x_{ij}\, x_{i'j} \qquad (17)$$

Support Vector Machine

And it transpires that the classifier can be represented in terms of these inner products as:

$$f(x) = \beta_0 + \sum_{j=1}^{\mathrm{n}} \alpha_{\mathrm{i}} \langle \mathrm{x}, x_{i'} \rangle \qquad (18)$$

To estimate the α_{i} we only need to calculate the $\binom{\mathrm{n}}{2} = ((\mathrm{n}-1)/2$ inner products between all pairs of training observations

Moreover, these α_{i} are non-zero only for the support vectors

Support Vector Machine

Finally, we can replace the inner product expression $\alpha_{\mathrm{i}}\langle \mathrm{x}, x_{i'}\rangle$ with a more general kernel function $\mathrm{K}(\mathrm{x}, x_{i'})$ and

$$F(x) = \beta_0 + \sum_{i \epsilon S} \alpha_i \, K(x, x_i) \tag{19}$$

For example, if

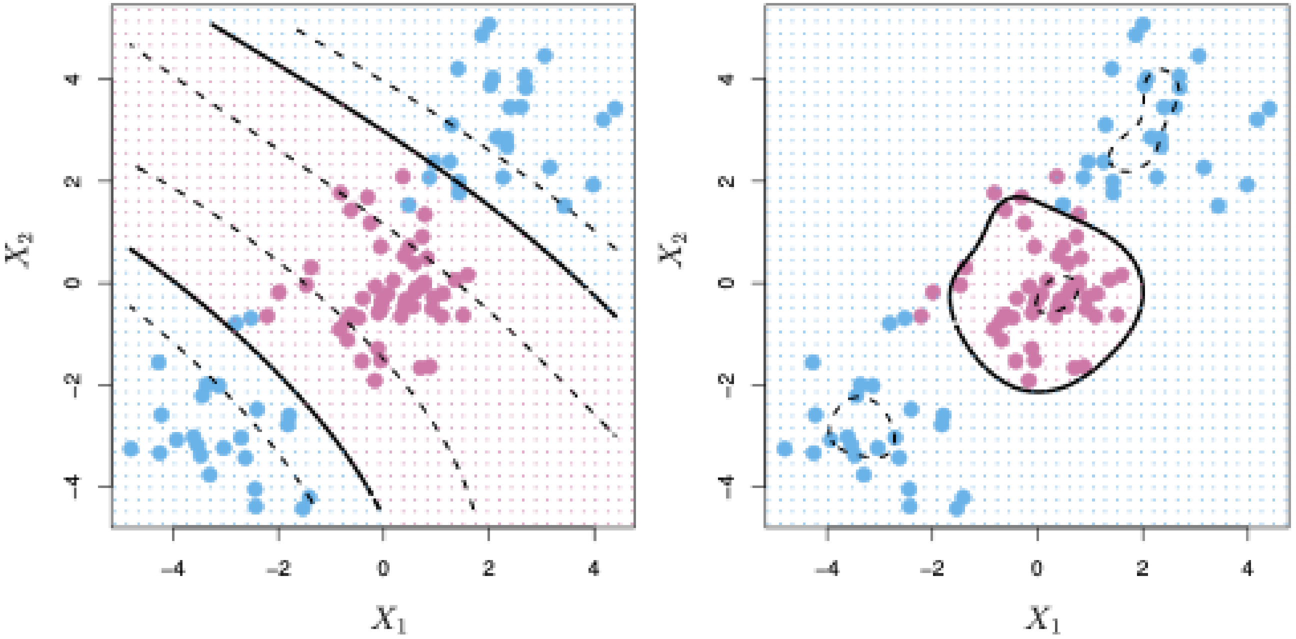
$$\mathrm{K}(\mathrm{x}, x_{i'}) = \sum_{j=1}^{p} x_{ij} \, x_{i'j} \tag{20}$$

We have the support vector classifier

But there are other common choices for the kernel function

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij}\, x_{i'j}\right)^d$$

polynomial kernel of degree d



Radial kernel

$$K(x_i, x_{i'}) = exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right)$$