- Our goal is to develop a classifier based on the training data that will correctly classify the test observation using its feature measurements.
- Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels.
- Then a separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \quad \text{if} \quad y_i = 1,$$

and

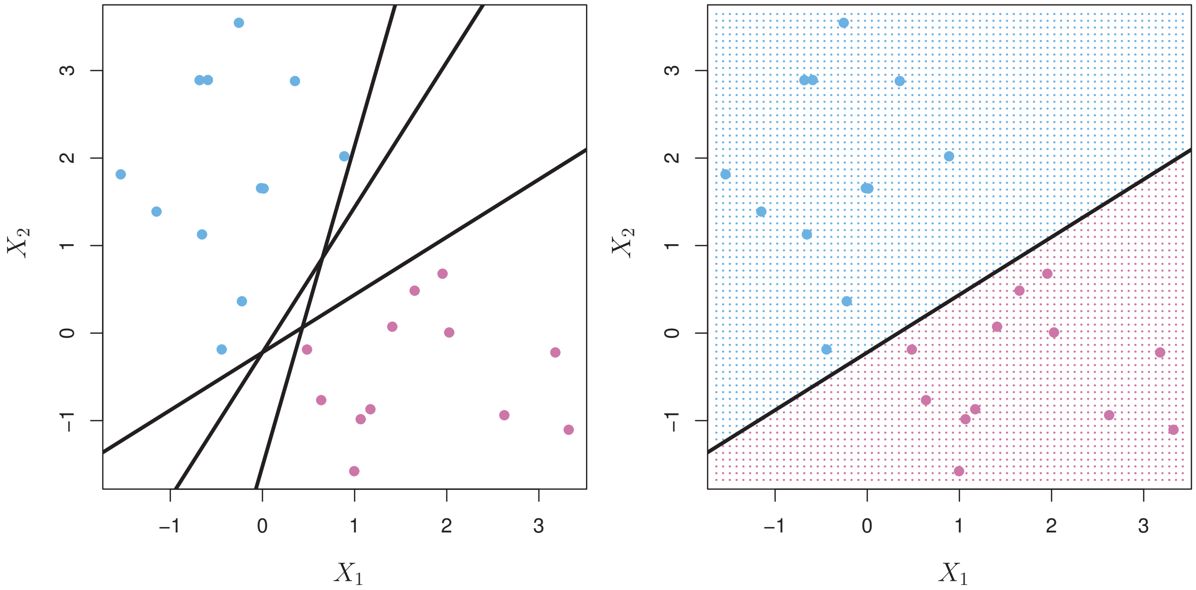
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \quad \text{if} \quad y_i = -1,$$

Equivalently, a separating hyperplane has the property that

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0$$

for all $i = 1, 2, \ldots, n$.

- If a separating hyperplane exists, we can use it to construct a very natural classifier: a test observation is assigned a class depending on which side of the hyperplane it is located.
- That is, we classify the test observation x^* based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$. If $f(x^*)$ is positive, then we assign the test observation to class 1, and if $f(x^*)$ is negative, then we assign it to class -1.
- We can also make use of the magnitude of $f(x^*)$.
  - If $f(x^*)$ is far from zero, then this means that x^* lies far from the hyperplane, and so we can be confident about our class assignment for x^*.
  - On the other hand, if $f(x^*)$ is close to zero, then x^* is located near the hyperplane, and so we are less certain about the class assignment for x^*.



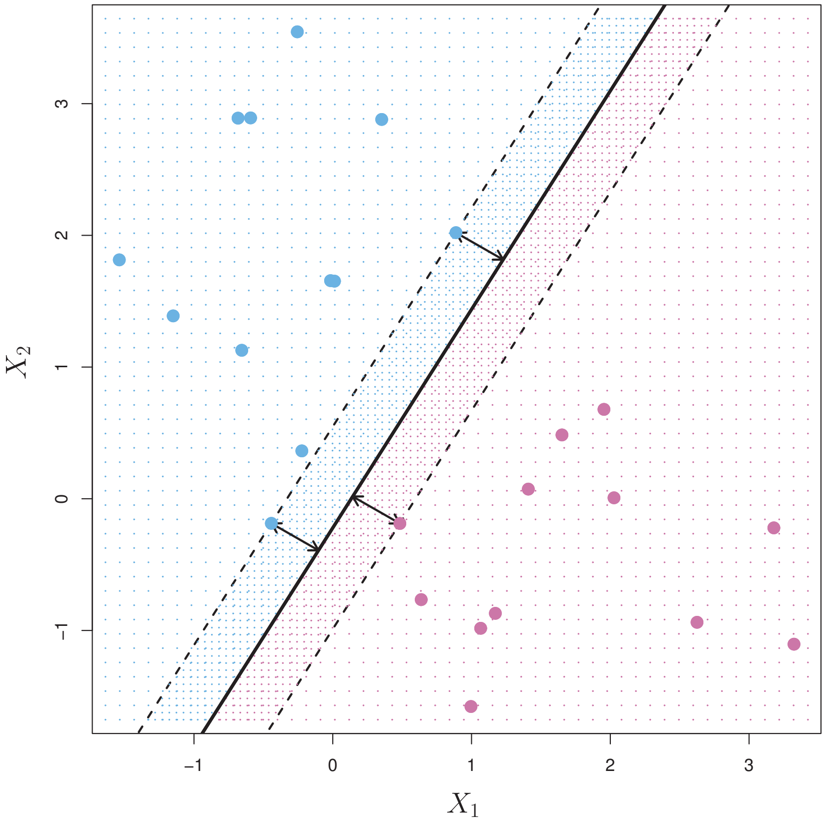
There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- A natural choice (among the infinite number of such hyperplanes) is the maximal margin hyperplane (also known as the optimal separating hyperplane), which is the separating hyperplane that is farthest from the training observations.
- That is, we can compute the (perpendicular) distance from each training observation to a given separating hyperplane; the smallest such distance

is the minimal distance from the observations to the hyperplane, and is known as the *margin*.

- The maximal margin hyperplane is the separating hyperplane for which the margin is largest.
- We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as the *maximal margin classifier*.



There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the hyperplane is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.

Construction of the Maximal Margin Classifier

The maximal margin hyperplane is the solution to the optimization problem

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

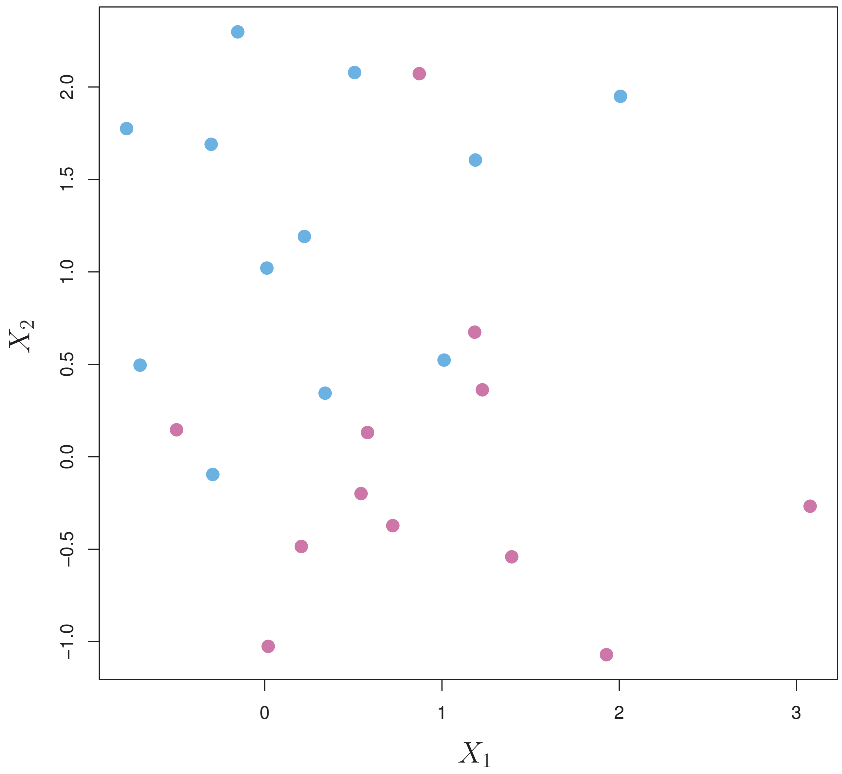
$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \qquad (1)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geqslant M, \; \forall \; i = 1, \ldots, n. \qquad (2)$$

The constraints (1) and (2) ensure that each observation is on the correct side of the hyperplane and at least a distance M from the hyperplane. Hence, M represents the margin of our hyperplane, and the optimization problem chooses $\beta_0, \ldots, \beta_p$ to maximize M.

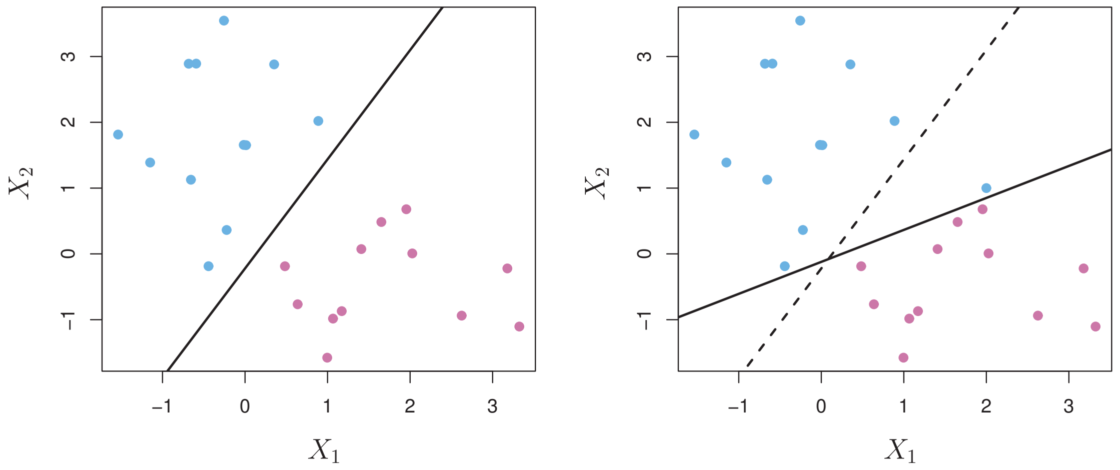
Notes:

- In many cases no separating hyperplane exists, and so there is no maximal margin classifier. In this case, the optimization problem has no solution with $M > 0$.



There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.

- Even if a separating hyperplane does exist, then there are instances in which a classifier is extremely sensitive to individual observations.

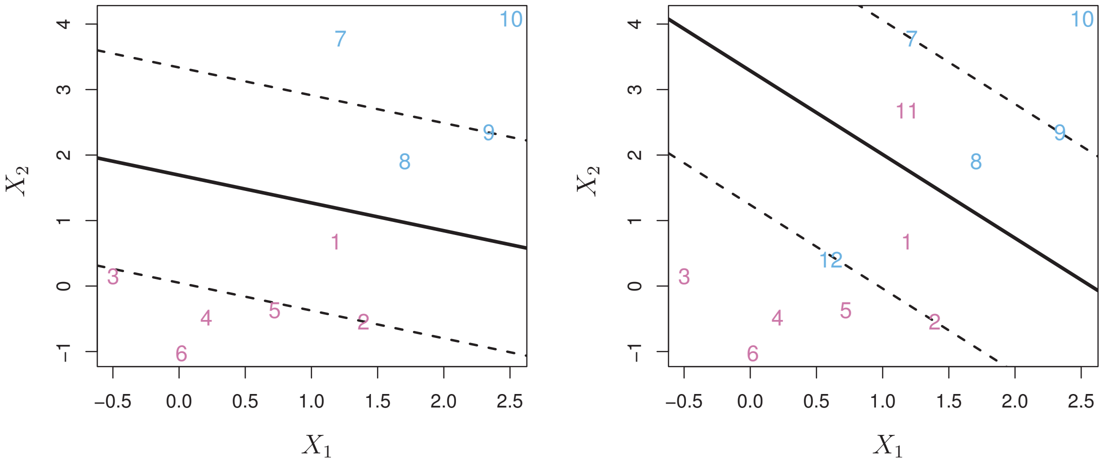


Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane. Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.

- In this case, we might be willing to consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of
  - Greater robustness to individual observations, and
  - Better classification of most of the training observations.

That is, it could be worthwile to misclassify a few training observations in order to do a better job in classifying the remaining observations.

- The support vector classifier, sometimes called a soft margin classifier, does exactly this.



Left: A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Purple observations: Observations 3,4,5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observations: Observations 7 and 10 are on the correct side of the margin, observation 9 is on the margin, and observation 8 is on the wrong side of the margin. No observations are on the wrong side of the hyperplane. Right: Same as left panel with two additional points, 11 and 12. These two observations are on the wrong side of the hyperplane and the wrong side of the margin.

Support Vector Classifiers

- The support vector classifier classifies a test observation depending on which side of a hyperplane it lies.
- The hyperplane is chosen to correctly separate most of the training observations into the two classes, but may misclassify a few observations.

- It is the solution to the optimization problem

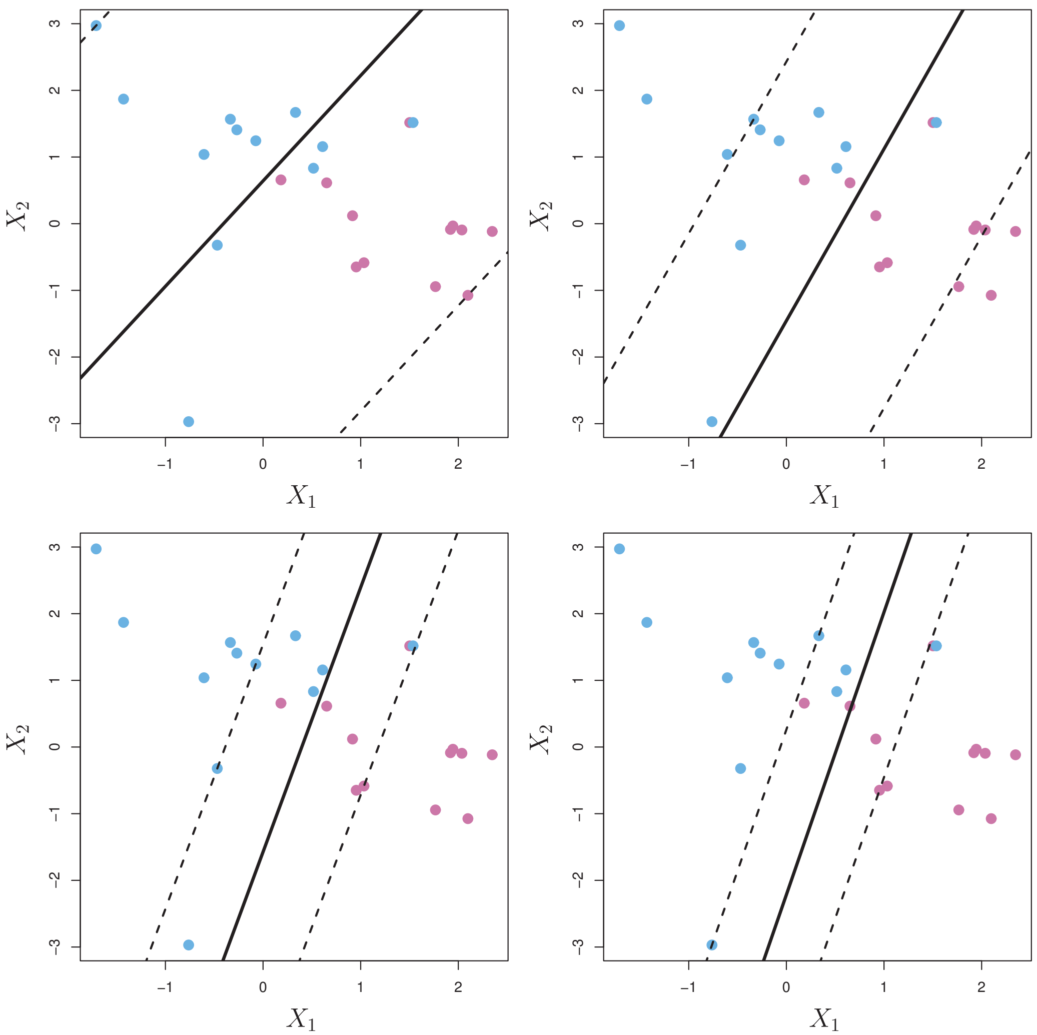
$$
\begin{aligned}
&\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M \\
&\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \\
&y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geqslant M(1 - \epsilon_i), \\
&\epsilon_i \geqslant 0, \quad \sum_{i=1}^{n} \epsilon_i \leqslant C,
\end{aligned}
$$

  where

  - M is the width of the margin;
  - $\epsilon_1, \ldots, \epsilon_n$ are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane;
  - And C is a nonnegative tuning parameter which determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate.

C is generally chosen via cross-validation.

- Once we have solved the problem, we classify a test observation x^* as before, by simply determining on which side of the hyperplane it lies. That is, we classify the test observation based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \ldots + \beta_p x_p^*$.



A support vector classifier was fit using four different values of the tuning parameter C. The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.

- It turns out that only observations that either lie on the margin or that violate the margin will affect the hyperplane, and hence the classifier obtained.

- In other words, an observation that lies strictly on the correct side of the margin does not affect the support vector classifier! Changing the position of that observation would not change the classifier at all, provided that its position remains on the correct side of the margin.
- Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as *support vectors*. These observations do affect the support vector classifier.

when tuning parameter C is Large, there are many support vectors and hence the resulting classifier will have Low variance but potentially high bias.
In contrast, if C is small, then there will be fewer support vectors and hence the resulting classifier will have low bias but high variance.
Therefore, C controls the bias-variance trade-off of the support vector classifier.

Performing Support Vector Classifier in R

The e1071 library contains implementations for a number of statistical learning methods. In particular, the svm() function can be used to fit a support vector classifier when the argument kernel = ”linear” is used.

We now use the svm() function to fit the support vector classifier for a given value of the cost parameter (the cost of a violation to the margin). Here we