Support Vector Machines

- The support vector machine (SVM), an approach for classification that was developed in the computer science community in the 1990s and that has grown in popularity since then.

Hyperplane:

In a p-dimensional space, a hyperplane is a flat affine subspace of dimension $p - 1$. For instance, in two dimensions, a hyperplane is a flat one-dimensional subspace—in other words, a line. In three dimensions, a hyperplane is a flat two-dimensional subspace—that is, a plane.

In two dimensions, a hyperplane is defined by the equation

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

for parameters β_0, β_1 and β_2. When we say that (1) defines the hyperplane, we mean that any $X = (X_1, X_2)'$ for which (1) holds is a point on that hyperplane.

In p- dimensional setting $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$ (2)

defines p- dimensional hyperplane, again in the sense that if a point $X = (X_1, X_2, \ldots, X_p)'$ in p- dimensional satisfies (2), then X lies on the hyperplane.

Now, suppose that X does not satisfy (2); rather,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0.$$

Then this tells us that X lies on the one side of the hyperplane. On the other hand if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0$, then X lies on the other side of the hyperplane.

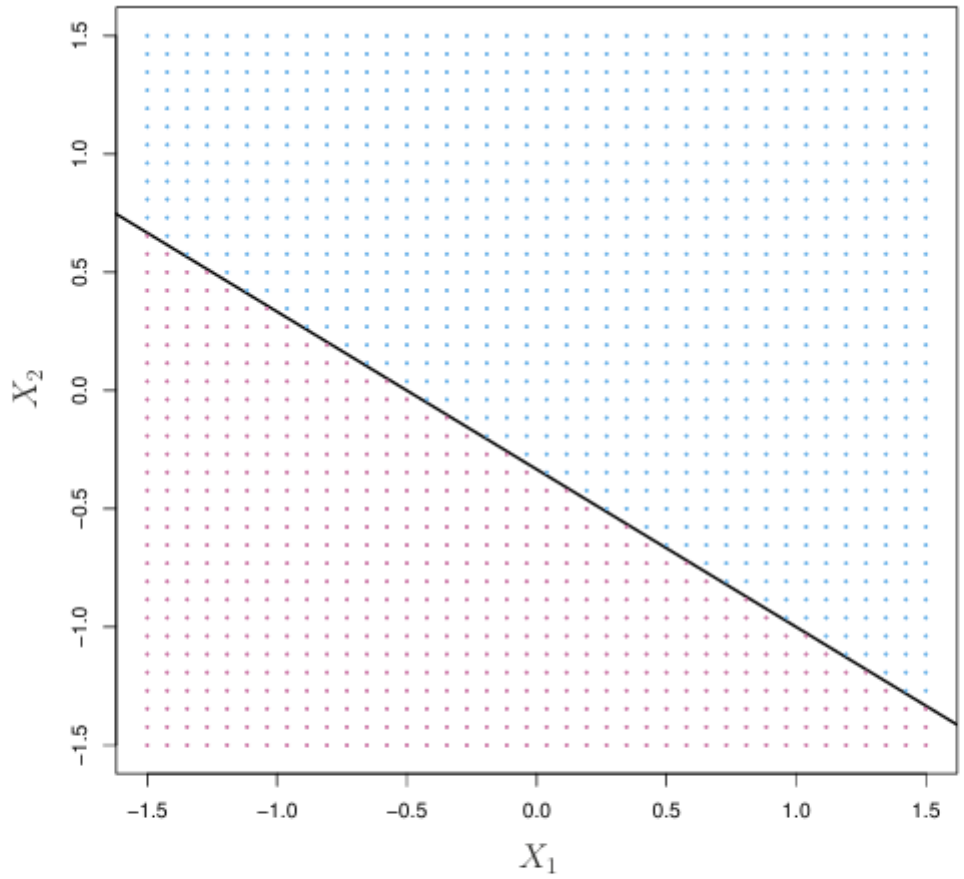


FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$*.*

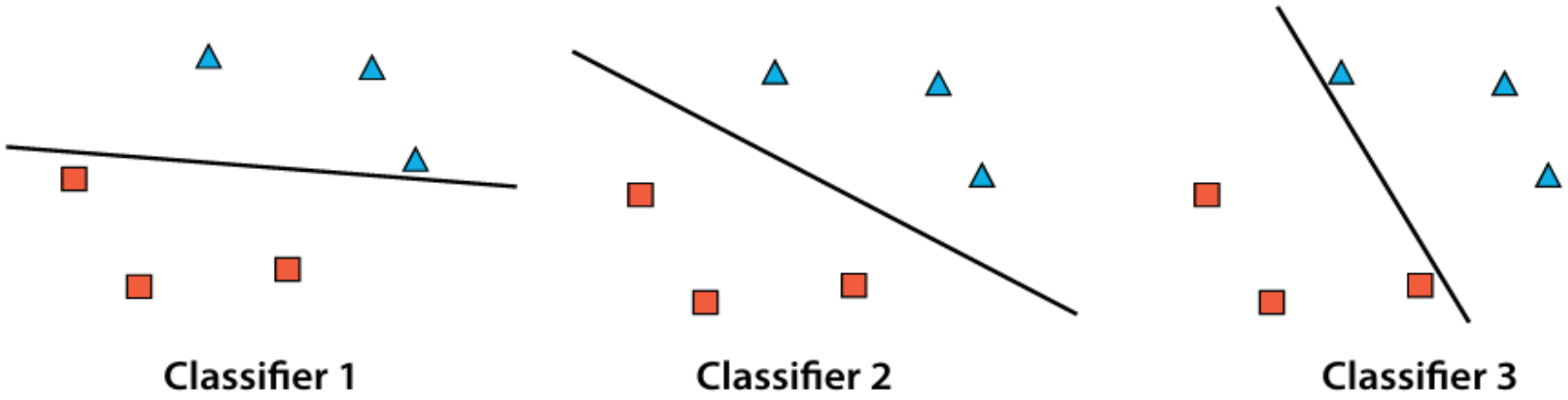


Figure 11.1 Three classifiers that classify our data set correctly. Which should we prefer, classifier 1, 2, or 3?

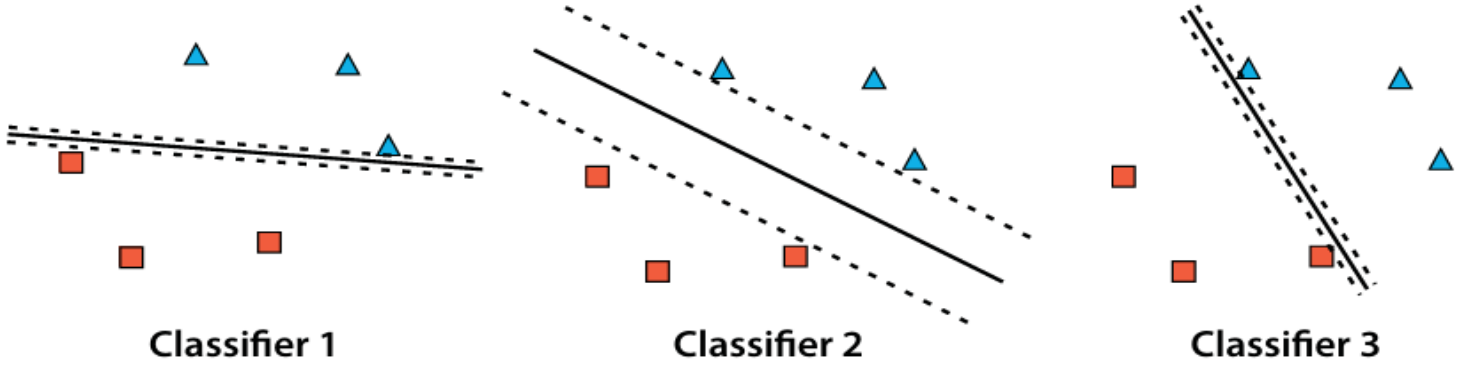


Figure 11.2 We draw our classifier as two parallel lines, as far apart from each other as possible. We can see that classifier 2 is the one where the parallel lines are the farthest away from each other. This means that the middle line in classifier 2 is the one best located between the points.

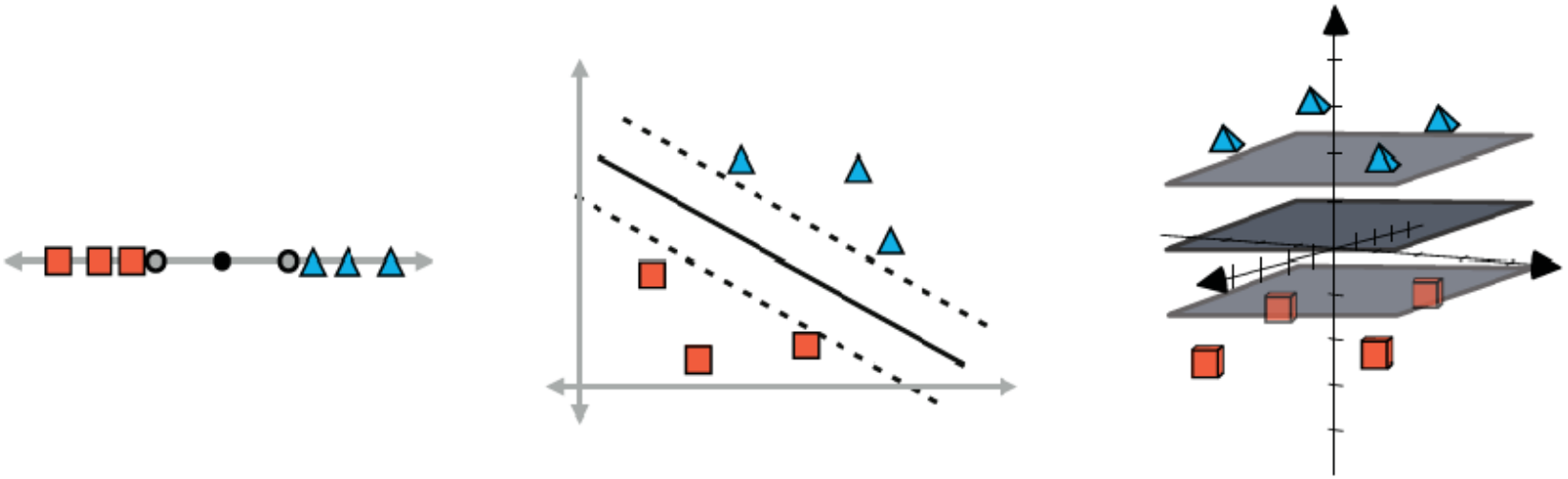


Figure 11.3 Linear boundaries for datasets in one, two, and three dimensions. In one dimension, the boundary is formed by two points, in two dimensions by two lines, and in three dimensions by two planes. In each of the cases, we try to separate these two as much as possible. The middle boundary (point, line, or plane) is illustrated for clarity.

Classification Using a Separating Hyperplane:

Now suppose that we have a $n \times p$ data matrix X that consists of n training observations in p-dimensional space,

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}$$

and these observations fall into two classes- that is $y_1, y_2, \ldots, y_n \in \{-1, 1\}$ where -1 represents one class and 1 the other class. We also have a test observation, a p-vector of observed features $x^* = \left(x_1^*, x_2^*, \ldots x_p^*\right)$. Our goal is to develop a classifier based on the training data that will correctly classify the test observation using its feature measurements.

A separating hyperplane has the property that

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

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

Equivalently, a separating hyperplane has the property that

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

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.

We classify the test observation x^* based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^*$.

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.

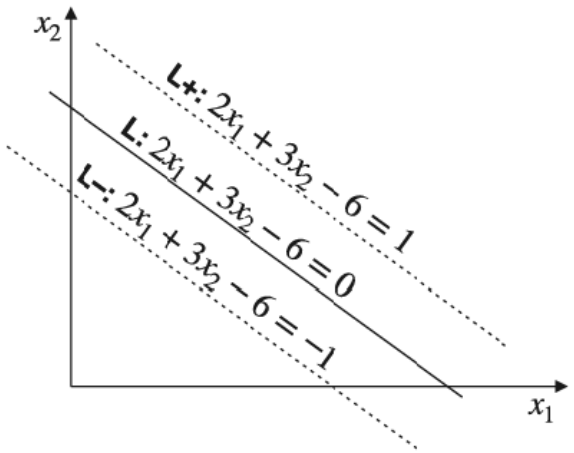


Figure 11.4 Our main line L is the one in the middle. We build the two parallel equidistant lines L+ and L– by slightly changing the equation of L.

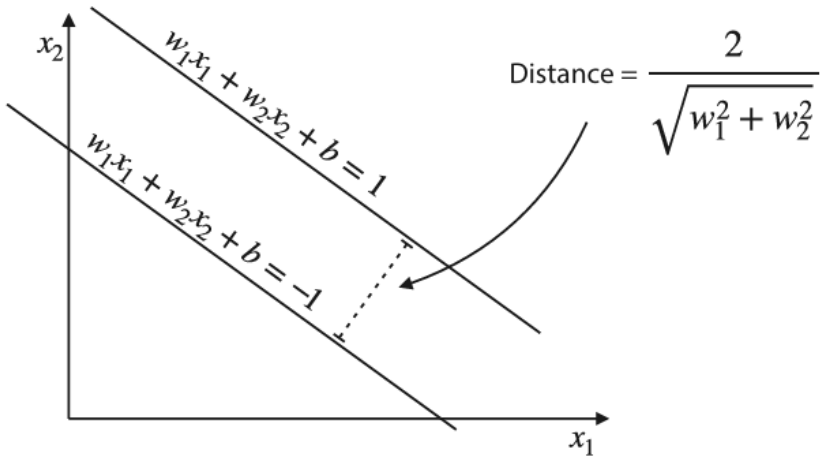
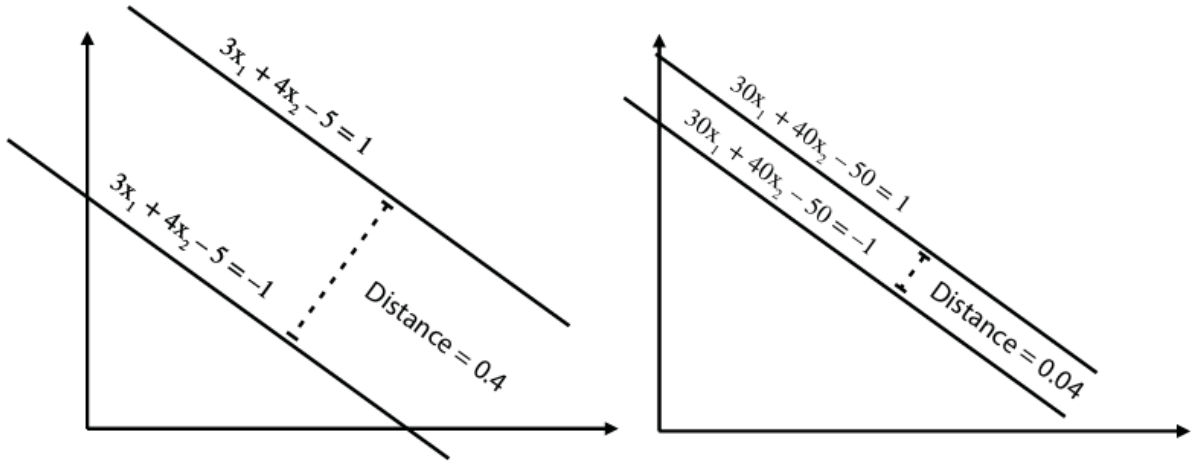
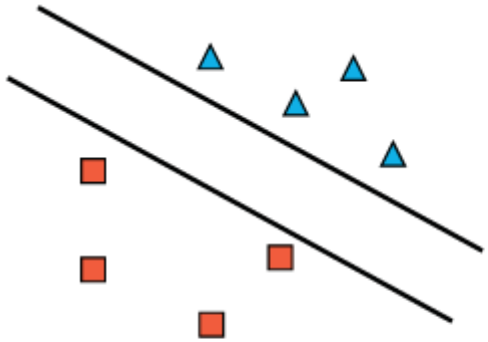


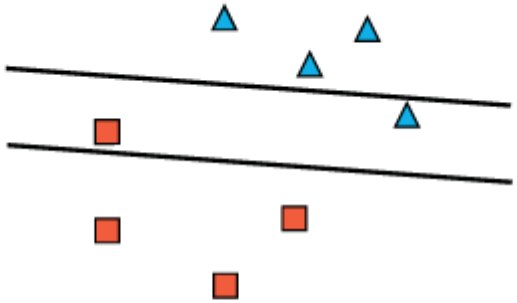
Figure 11.6 The distance between the two parallel lines can be calculated based on the equations of the lines.



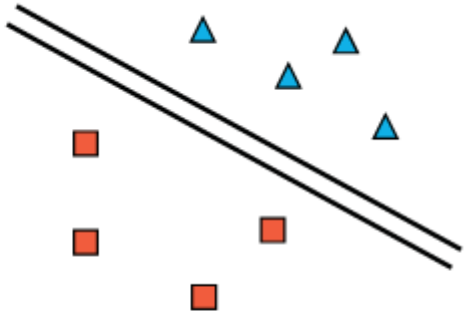
Good Classifier **Bad Classifier**



Good classifier



Bad classifier
(makes errors)



Bad classifier
(lines are too close together)

Maximal Margin Classifier:

- The maximal margin hyperplane (also known as the optimal separating hyperplane), which is the separating hyperplane that is farthest from the training observations. Applicable when classes are separable by a linear boundary.
- 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—that is, it is the hyperplane that has the farthest minimum distance to the training observations.
- 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.

- If $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients of the maximal margin hyperplane, then the maximal margin classifier classifies the test observation x^* based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^*$.

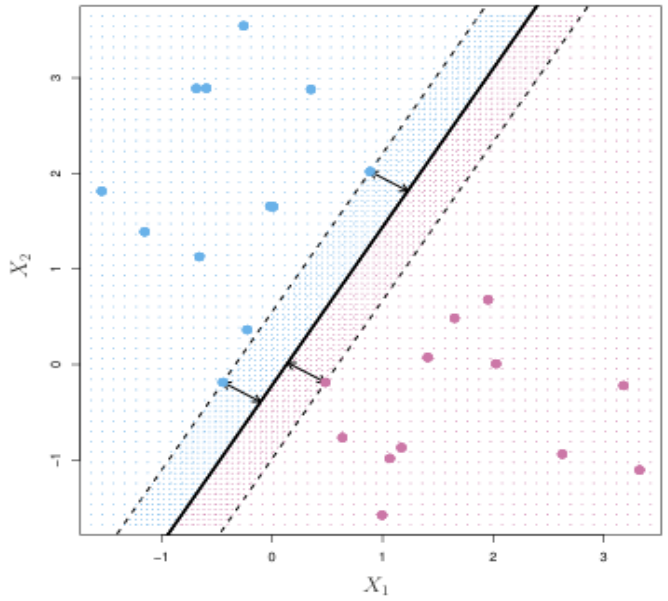


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support 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.*

From the above graph we see that

- Three training observations are equidistant from the maximal margin hyperplane and lie along the dashed lines indicating the width of the margin.
- These three observations are known as support vectors and they "support" the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyper plane would move as well.
- Interestingly, the maximal margin hyperplane depends directly on the support vectors, but not on the other observations.
- A movement to any of the other observations would not affect the separating hyperplane, provided that the observation's movement does not cause it to cross the boundary set by the margin.

Construction of the Maximal Margin Classifier:

Now we construct the maximal margin hyperplane based on a set of n training observations $x_1, x_2, \dots, x_p \in R^p$ and associated class labels $y_1, y_2, \dots, y_n \in \{-1,1\}$. The maximal margin hyperplane is the solution to the optimization problem

$$\underset{\beta_0,\beta_1,\dots,\beta_p,M}{maximize} M$$

$$\text{Subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M \quad \forall\, i = 1,2, \dots, n$$

- The constraint $y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M \; \forall\, i = 1,2, \dots, n$ guarantees that each observation will be on the correct side of the hyper plane, provided that M is positive.
- M represents the margin of our hyperplane, and the optimization problem chooses $\beta_0, \beta_1, \dots, \beta_p$ to maximize M.

Non-separable Case:

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.
- In many cases no separating hyperplane exists, and so there is no maximal margin classifier. In this the optimization problem has no solution with $M > 0$.
- We can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called soft margin.
- The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier.

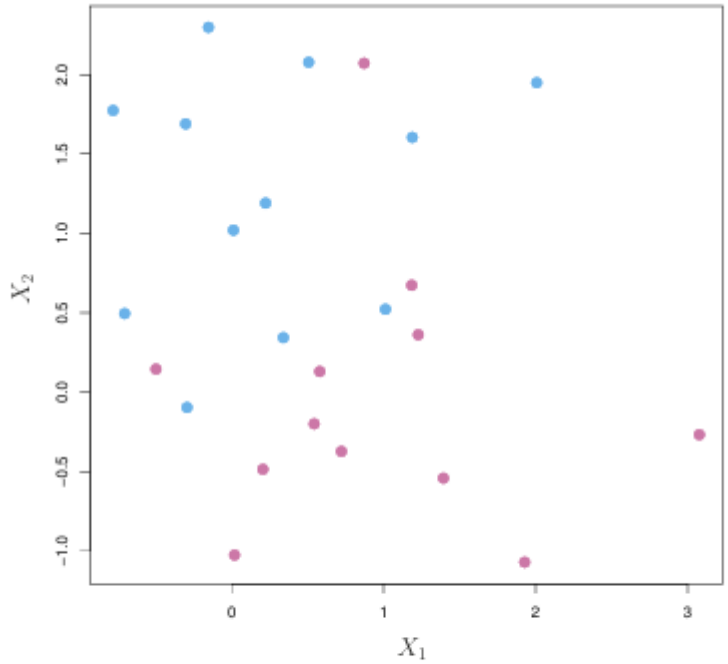


FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Support Vector Classifiers:

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 worthwhile to misclassify a few training observations in order to do a better job in classifying the remaining observations.

We allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

The support vector classifier, sometimes called a soft margin classifier, does exactly this. The margin is soft because it can be violated by some of the training observations.

Most of the observations are on the correct side of the margin. However, a small subset of the observations are on the wrong side of the margin.

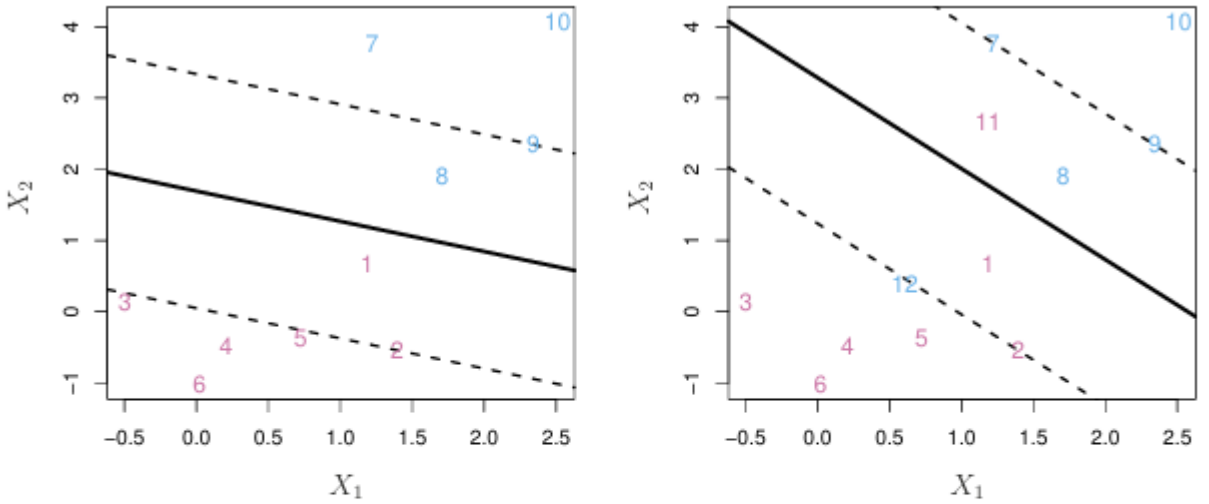


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation 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.*

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

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n,M}{maximize} M$$

$$\text{Subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) \geq M(1 - \varepsilon_i),$$

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

where C is a nonnegative tuning parameter.

M is the width of the margin; we seek to make this quantity as large as possible.

$\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane.

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

If $\varepsilon_i = 0$ then the i^{th} observation is on the correct side of the margin.

If $\varepsilon_i > 0$ then the i^{th} observation is on the wrong side of the margin and we say that the i^{th} observation has violated the margin.

If $\varepsilon_i > 1$ then it is on the wrong side of the hyperplane.

C bounds the sum of the $\varepsilon_i's$, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate.

We can think of C as a budget for the amount that the margin can be violated by the n observations. If $C = 0$ then there is no budget for violations to the margin, and it must be the case that $\varepsilon_1 = \varepsilon_2 = \cdots = \varepsilon_n = 0$, in which is the maximal margin hyperplane optimization problem. (Of course, a maximal margin hyperplane exists only if the two classes are separable.)

For $C > 0$ no more than C observa tions can be on the wrong side of the hyperplane, because if an observation is on the wrong side of the hyperplane then $\varepsilon_i > 1$, and requires that $\sum_{i=1}^{n} \varepsilon_i \leq C$.

As the budget C increases, we become more tolerant of violations to the margin, and so the margin will widen.

Conversely, as C decreases, we become less tolerant of violations to the margin and so the margin narrows.

In practice, C is treated as a tuning parameter that is generally chosen via cross-validation. As with the tuning parameters that we have seen that, C controls the bias-variance trade-off of the statistical learning technique. When C is small, we seek narrow margins that are rarely violated; this amounts to a classifier that is highly fit to the data, which may have low bias but high variance. On the other hand, when C is larger, the margin is wider and we allow more violations to it; this

amounts to f itting the data less hard and obtaining a classifier that is potentially more biased but may have lower variance.

The optimization problem has a very interesting property: 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 the tuning parameter C is large, then the margin is wide, many observations violate the margin, and so there are many support vectors. In this case, many observations are involved in determining the hyperplane.

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.

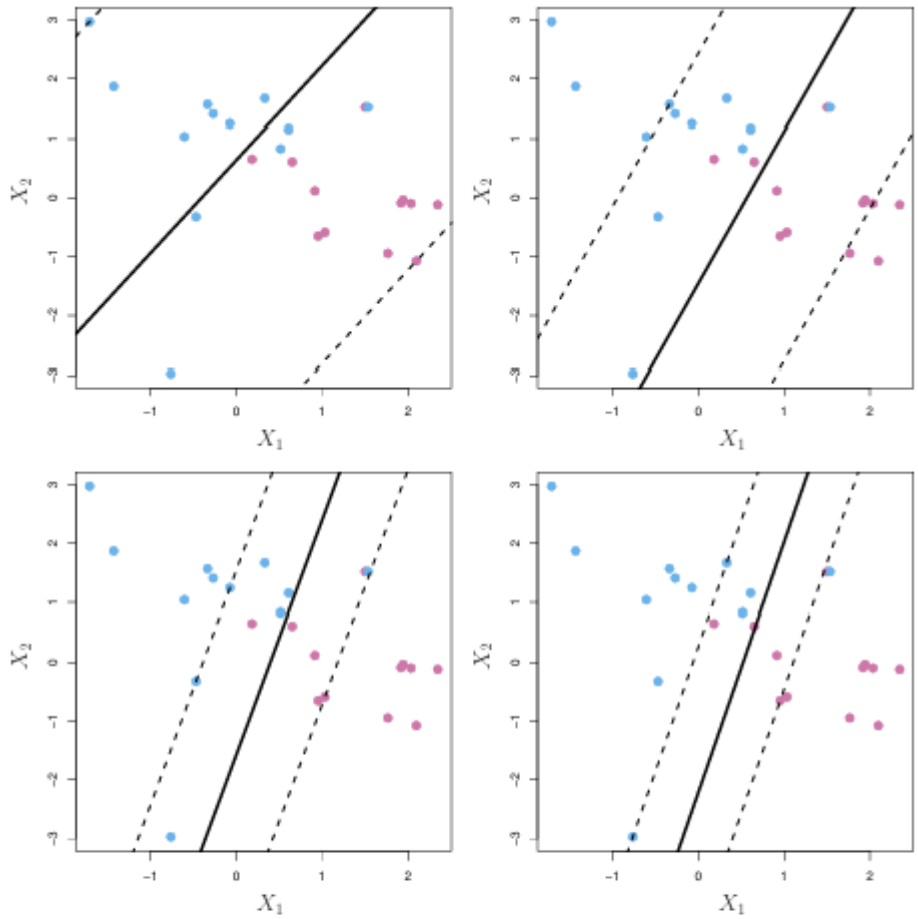


FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance 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.*