Support Vector Machine

Jasmine. Hao[1]

[1]University of Hong Kong

ECON 3225: Big Data Economics

- What is the support vector machine (SVM)?
  - SVM is an approach for classification that was developed in the computer science community in the 1990s and that has grown in popularity since then.
  - SVMs have been shown to perform well in a variety of settings, and are often considered one of the best “out of the box” classifiers.
  - SVM is a generalization of a simple and intuitive classifier called the maximal margin classifier.

- Introduce the maximal margin classifier.
- Discuss the support vector classifier, an extension for broader cases.
- Present the support vector machine for nonlinear class boundaries.
- Extend support vector machines to multi-class scenarios.
- Explore connections between support vector machines and logistic regression.

Outline

Outline

What is a 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 p > 3 dimensions, it can be hard to visualize a hyperplane, but the notion of a (p – 1)-dimensional flat subspace still applies.
- The mathematical definition of a hyperplane is:
  - In two dimensions,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0, \tag{9.1}$$

  - with parameters $\beta_0, \beta_1, \beta_2$

p-dimensional hyperplane

- The equation in the last slide can be easily extended to the p-dimensional setting:
- $$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0, \tag{9.2}$$
- defines a p-dimensional hyperplane, again in the sense that if a point $X = (X_1, X_2, ..., X_p)\top$ in p-dimensional space (i.e. a vector of length p) satisfies (9.2), then X lies on the hyperplane.

[Illustration]

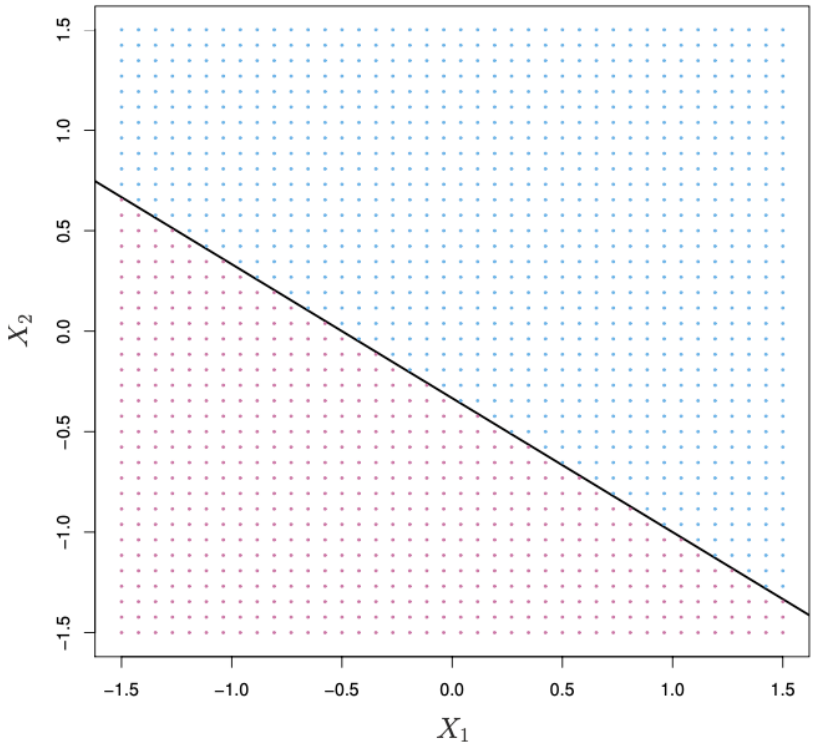

FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$*.*

p-dimensional hyperplane

- Consider when X does not satisfy (9.2), but instead:

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p > 0. \tag{9.3}$$

- This indicates X lies on one side of the hyperplane.
- Conversely, if

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p < 0, \tag{9.4}$$

- X is on the opposite side. Thus, the hyperplane divides p-dimensional space.
- The side of the hyperplane a point lies on can be determined by the sign of (9.2).
- A two-dimensional hyperplane is illustrated in Figure 9.1.

Classification Using a Separating Hyperplane

- Consider a $n \times p$ data matrix X with n training observations in p-dimensional space:

$$x_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1p} \end{bmatrix}, \quad x_n = \begin{bmatrix} x_{n1} \\ \vdots \\ x_{np} \end{bmatrix} \tag{9.5}$$

- Assume these observations fall into two classes, with a test observation $x^* = (x^{*1}, \ldots, x^{*p})^\top$.
- Our goal is to develop a classifier based on training data to correctly classify the test observation.
- Previous approaches include linear discriminant analysis, logistic regression (Chapter 4), and classification trees, bagging, and boosting (Chapter 8).
- We now introduce a new approach based on the concept of a separating hyperplane.

[Illustration]Separating Hyperplane

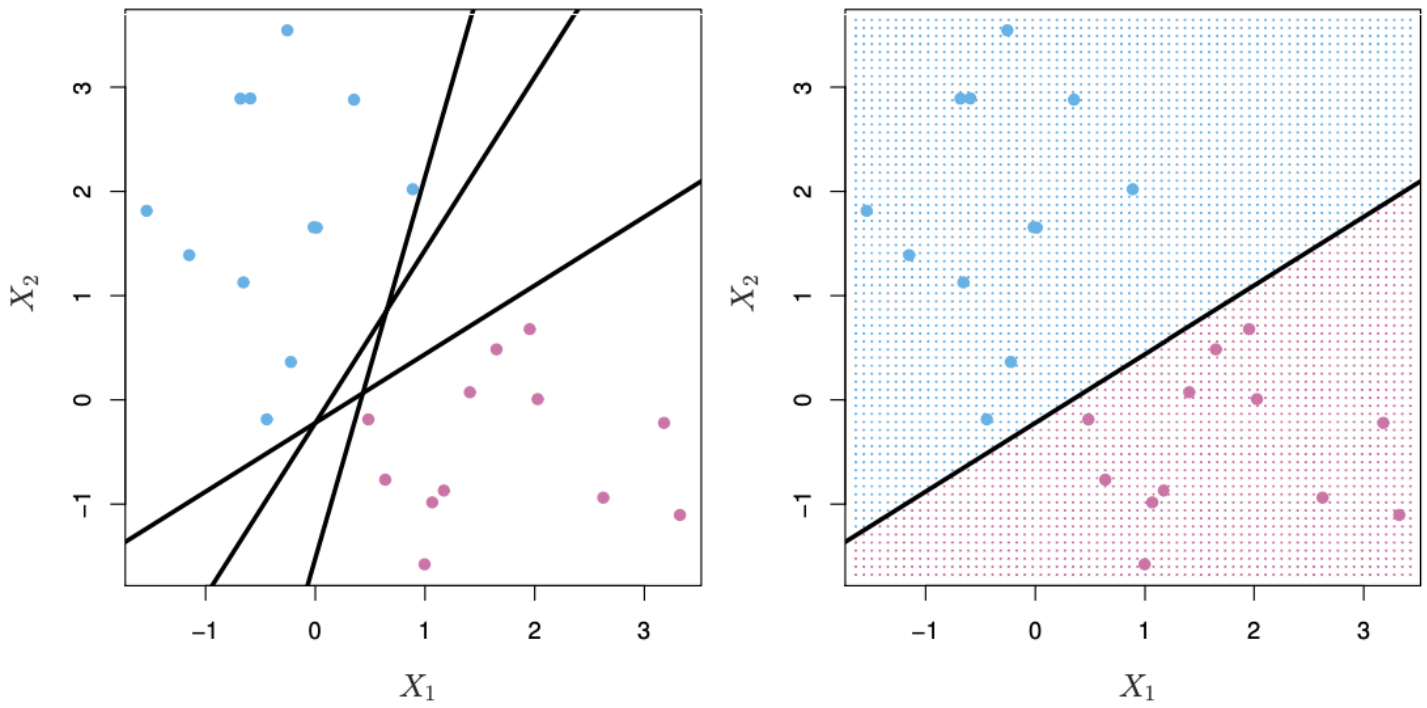

FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision 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.*

Separating Hyperplane

- Suppose that it is possible to construct a hyperplane that separates the training observations perfectly according to their class labels. Examples of three such separating hyperplanes are shown in the left-hand panel of Figure 9.2.
- We can label the observations from the blue class as yi = 1 and those from the purple class as $y_i = -1$. Then a separating hyperplane has the property that
- $$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 \text{ if } y_i = 1, \tag{9.6}$$
- equivalently
- $$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 \text{ if } y_i = -1, \tag{9.7}$$

Using a Hyperplane to Define a Classifier

- If a separating hyperplane exists, we can construct a natural classifier: a test observation is assigned a class based on which side of the hyperplane it lies.
- Refer to the right-hand panel of Figure 9.2 for an example.
- We classify the test observation x^* based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$:
  - Class 1 if $f(x^*)$ is positive.
  - Class -1 if $f(x^*)$ is negative.
- The magnitude of $f(x^*)$ indicates confidence in the class assignment:
  - Far from zero: confident about the class assignment.
  - Close to zero: less certain about the class assignment.
- A classifier based on a separating hyperplane leads to a linear decision boundary (see Figure 9.2).

The Maximal Margin Hyperplane

- If data can be perfectly separated by a hyperplane, there are infinitely many such hyperplanes.
- These hyperplanes can be shifted or rotated without touching any observations (see left-hand panel of Figure 9.2).
- To choose a specific hyperplane for classification, we use the maximal margin hyperplane, which is farthest from the training observations.

The Maximal Margin Classifier

- The maximal margin hyperplane is chosen such that the minimal perpendicular distance (margin) from the training observations to the hyperplane is maximized.
- A test observation is classified based on which side of the maximal margin hyperplane it lies on, forming the maximal margin classifier.
- A large margin on training data is hoped to translate to a large margin on test data for accurate classification.
- However, the maximal margin classifier may overfit when p (the number of features) is large.

[Illustration] The Maximal Margin Classifier

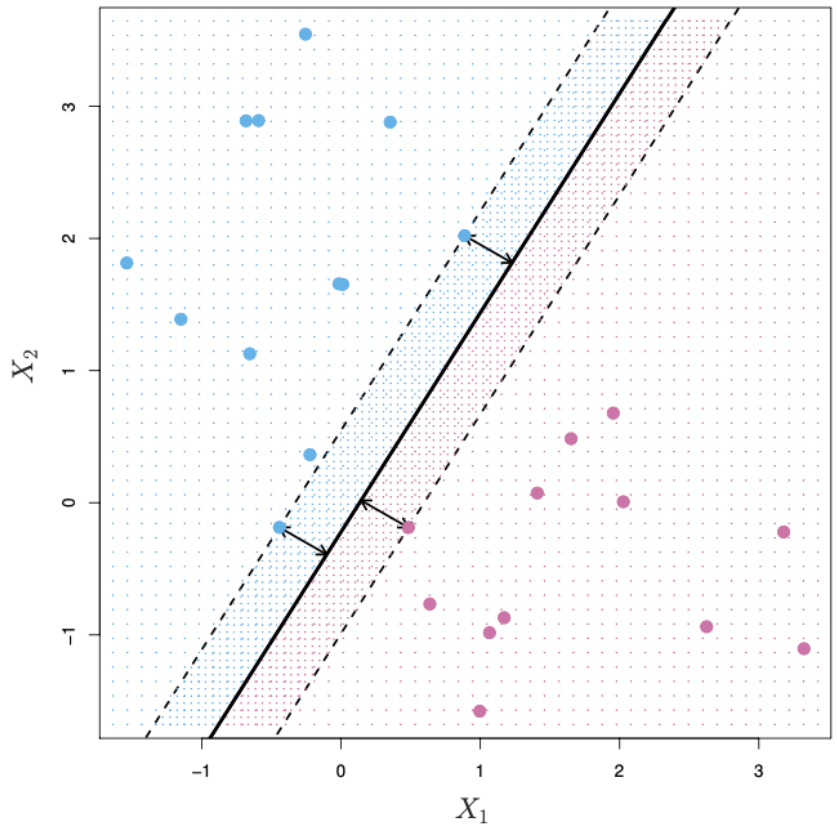

FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support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

- The maximal margin classifier classifies a test observation x^* based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$.
- Figure 9.3 shows the maximal margin hyperplane with a larger margin, representing the mid-line of the widest slab between the two classes.
- Three training observations, equidistant from the hyperplane, are support vectors. They support the hyperplane, meaning its position depends on them.
- The maximal margin hyperplane depends directly on the support vectors, but not on other observations, unless their movement crosses the margin boundary.

Construction of the Maximal Margin Classifier

- We now consider the task of constructing the maximal margin hyperplane based on a set of n training observations $x_1, ..., x_n \in R_p$ and associated class labels $y_1, ..., y_n \in -1, 1$. Briefly, the maximal margin hyperplane is the solution to the optimization problem
- $$maximize_{\beta_0,\beta_1,...,\beta_p,M} M \tag{9.9}$$
- $$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \tag{9.10}$$
- $$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M, \forall i = 1, \ldots n \tag{9.11}$$

- The constraint in (9.11) ensures that each observation is on the correct side of the hyperplane, with a positive margin M.
- The constraint in (9.10) is not directly about the hyperplane, but it gives meaning to (9.11) by defining the perpendicular distance from each observation to the hyperplane.
- Together, constraints (9.10) and (9.11) guarantee that each observation is on the correct side of the hyperplane and at least a distance M from it.
- M represents the margin, and the optimization problem seeks to maximize it by choosing appropriate values for $\beta_0, \beta_1, \ldots, \beta_p$.

The non-separable Case

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.
- However, as we have hinted, in many cases no separating hyperplane exists, and so there is no maximal margin classifier. In this case, the optimization problem (9.9)–(9.11) has no solution with $M > 0$.
- An example is shown in Figure 9.4. In this case, we cannot exactly separate the two classes.
- However, as we will see in the next section, 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.

[Illustration] Non-separable Case

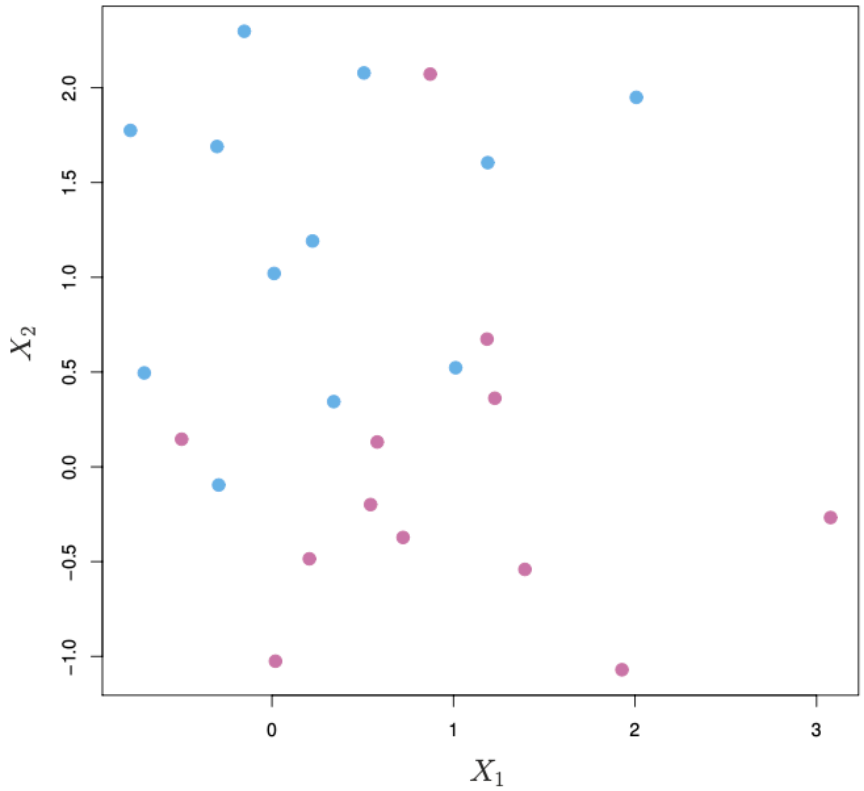

FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Support Vector Classifiers

- Even if a separating hyperplane exists, a classifier based on it might not always be desirable due to sensitivity to individual observations.
- A classifier based on a separating hyperplane will perfectly classify all training observations, which can lead to overfitting.
- Figure 9.5 demonstrates how the addition of a single observation can dramatically change the maximal margin hyperplane, resulting in a small margin.
- A small margin indicates low confidence in classification, and extreme sensitivity to a single observation suggests overfitting.

[Illustration]Non-existence of solution to Maximal Margin Hyperplane

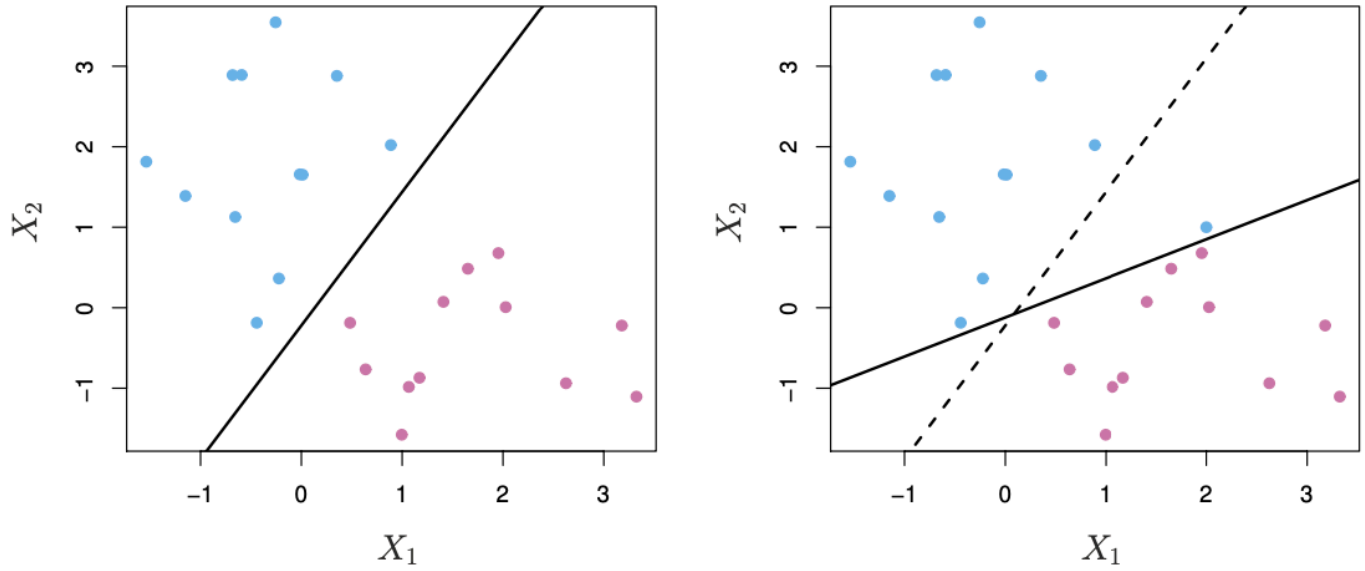

FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.*

Soft Margin Classifier

- We may consider a classifier based on a hyperplane that does not perfectly separate the two classes for greater robustness and better classification of most training observations.
- It can be worthwhile to misclassify a few training observations to improve overall classification.
- The support vector classifier, or soft margin classifier, allows for this flexibility.
  - Instead of seeking the largest possible margin with all observations on the correct side, we allow some observations to be on the incorrect side of the margin or even the hyperplane.
  - The margin is soft because it can be violated by some training observations.

[Illustration]Soft Margin Classifier

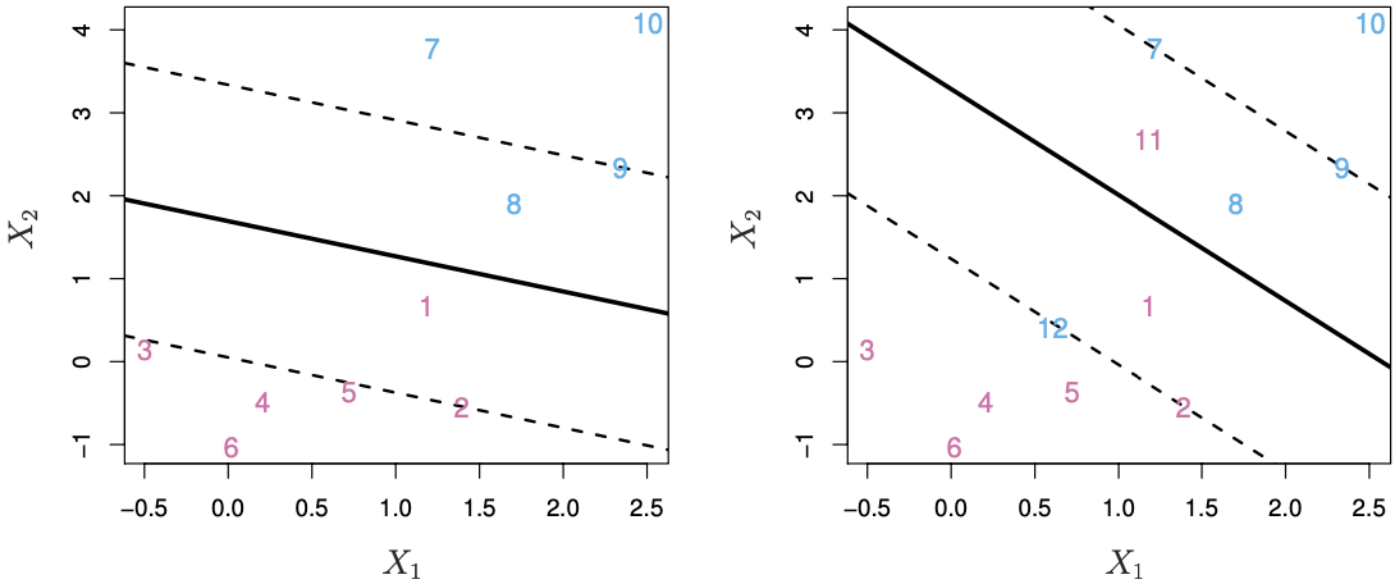

FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation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

Misclassification

- Figure 9.6. 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.
- An observation can be not only on the wrong side of the margin, but also on the wrong side of the hyperplane.
  - In fact, when there is no separating hyperplane, such a situation is inevitable.
- Observations on the wrong side of the hyperplane correspond to training observations that are misclassified by the support vector classifier.
- The right-hand panel of Figure 9.6 illustrates such a scenario.

Details of the Support Vector Classifier

- The support vector classifier classifies a test observation based on which side of a hyperplane it lies.
- The hyperplane is chosen to correctly separate most training observations, allowing for some misclassifications.
- The hyperplane is the solution to the optimization problem:

$$\begin{aligned} &\text{maximize} \quad M \\ &\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1 \\ &\qquad y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \ \forall i = 1, \ldots, n \\ &\qquad \epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C, \end{aligned}$$

where C is a nonnegative tuning parameter.

Interpretation of Equations and Slack Variables

- M is the width of the margin, which we aim to maximize.
- $\epsilon_1, \ldots, \epsilon_n$ are slack variables that allow observations to be on the wrong side of the margin or the hyperplane.
- After solving the optimization problem, we classify a test observation x^* by determining which side of the hyperplane it lies on.
- The slack variable ϵ_i indicates the position of the ith observation relative to the hyperplane and margin:
  - If $\epsilon_i = 0$, the observation is on the correct side of the margin.
  - If $\epsilon_i > 0$, the observation has violated the margin.
  - If $\epsilon_i > 1$, the observation is on the wrong side of the hyperplane.

Interpretation of Tuning Parameter C

- The tuning parameter C in (9.15) determines the number and severity of violations to the margin and hyperplane that are tolerated.
- C can be thought of as a budget for margin violations by the n observations:
  - If $C = 0$, no violations are allowed, leading to the maximal margin hyperplane optimization problem (if the classes are separable).
  - For $C > 0$, no more than C observations can be on the wrong side of the hyperplane.
  - Increasing C makes us more tolerant of violations, widening the margin. Decreasing C makes us less tolerant, narrowing the margin (see Figure 9.7).

[Illustration]Support Vector Classifiers with different C

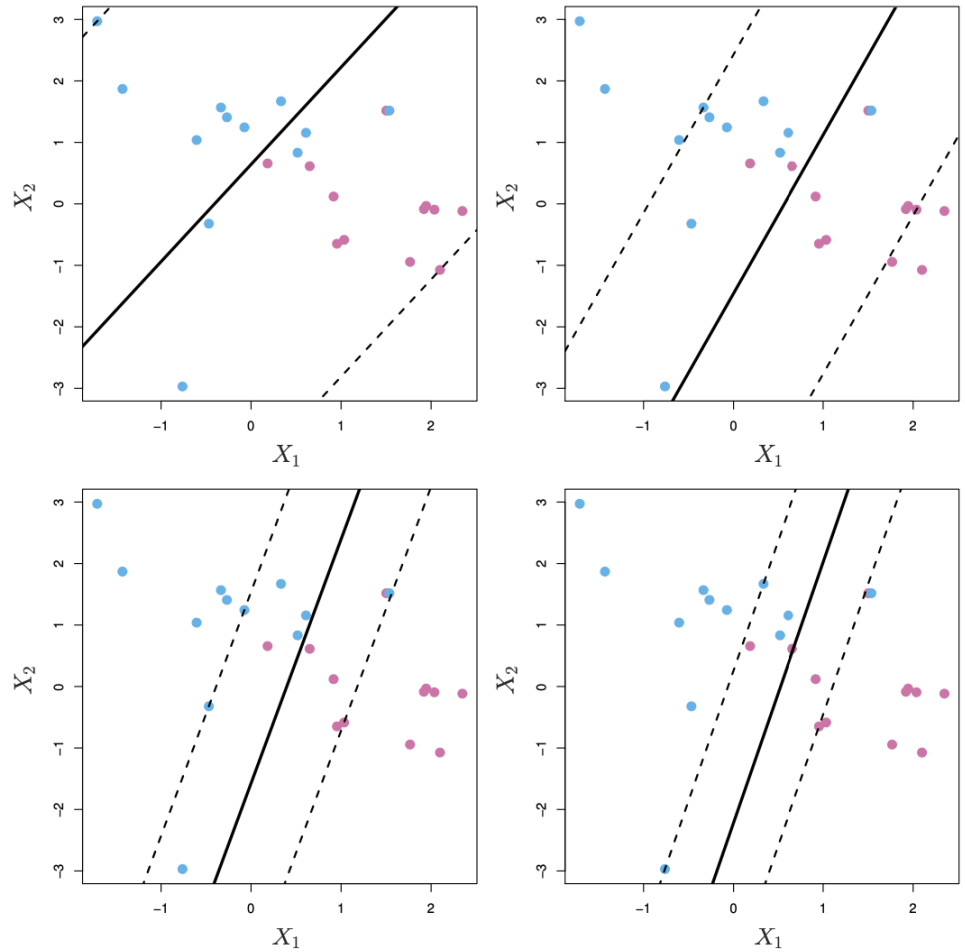

FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance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

Choice of the tuning parameter

- 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 throughout this book, 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.
  - when C is larger, the margin is wider and we allow more violations to it; this amounts to fitting the data less hard and obtaining a classifier that is potentially more biased but may have lower variance.

Support Vectors

- The optimization problem (9.12)–(9.15) 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.
  - 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.

The Bias-Variance Trade-off

- C controls the bias-variance trade-off of the support vector classifier:
  - Large C: Wide margin, many observations violate the margin, leading to low variance but potentially high bias (see top left panel of Figure 9.7).
  - Small C: Fewer support vectors, resulting in low bias but high variance (see bottom right panel of Figure 9.7).

- The support vector classifier's decision rule is based only on a potentially small subset of the training observations (the **support vectors**), making it robust to observations far from the hyperplane.
- This property distinguishes it from methods like linear discriminant analysis (LDA), which depends on the mean and covariance of all observations within each class.
- In contrast, logistic regression, like the support vector classifier, has low sensitivity to observations far from the decision boundary.

Outline

Nonlinear Class Boundaries

- The support vector classifier works well with linear class boundaries, but may struggle with non-linear class boundaries.
- An example of nonlinear class boundaries is shown in the left-hand panel of Figure 9.8, where a linear classifier, like the support vector classifier shown in the right-hand panel, performs poorly.
- Similarly, linear regression can suffer when there is a nonlinear relationship between predictors and the outcome. This can be addressed by enlarging the feature space with functions of the predictors, such as quadratic and cubic terms.
- For classification, we can similarly address non-linear boundaries by enlarging the feature space with quadratic, cubic, and higher-order polynomial functions of the predictors.

[Illustration] Nonlinear decision boundaries

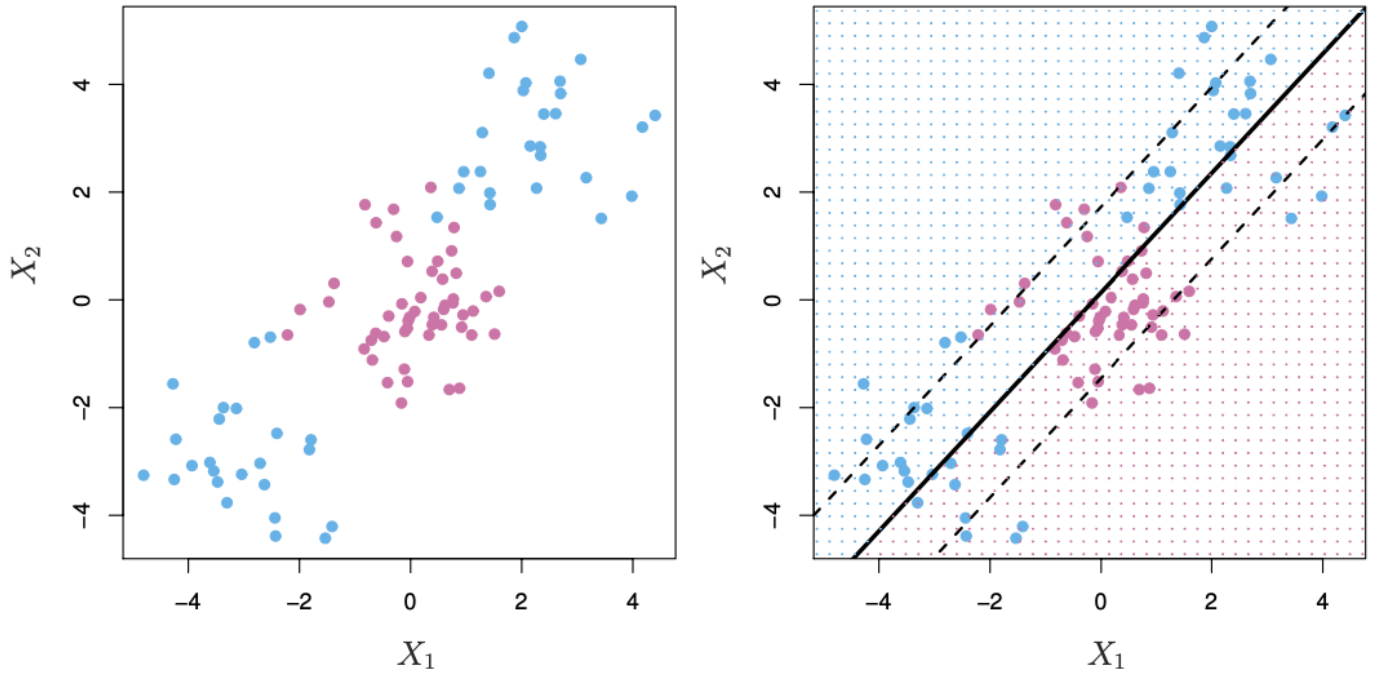

FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Nonlinear Decision Boundaries

- Instead of fitting a support vector classifier using $X_1, \ldots, X_p$, we instead fit $X_1, X_1^2 \ldots, X_p, X_p^2$
- The optimization problem in (9.12)–(9.15) would become

$$maximize_{\beta_0, \beta_{11}, \ldots, \beta_{p1} \beta_{p2}, M} M \qquad (9.16)$$

$$\text{subject to } \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1$$

$$y_i(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{i1} + \sum_{j=1}^{p} \beta_{j2} x_{i1}^2) \geq M(1 - \epsilon_i), \forall i = 1, \ldots n$$

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

Nonlinear Decision Boundaries

- Enlarging the feature space can lead to a non-linear decision boundary:
  - In the enlarged feature space, the decision boundary is linear.
  - In the original feature space, the decision boundary becomes a non-linear function, like a quadratic polynomial.
- The feature space can be further enlarged with higher-order polynomial terms or interaction terms.
- Other non-polynomial functions of the predictors can also be considered.
- However, enlarging the feature space too much can lead to a large number of features, making computations unmanageable.
- The support vector machine addresses this issue by allowing efficient computations even with an enlarged feature space.

The Support Vector Machine

- The support vector machine (SVM) extends the support vector classifier by enlarging the feature space using **kernels**, allowing for non-linear class boundaries.
- The kernel approach is an efficient computational method for implementing this enlargement of the feature space.
- The details of the SVM are complex, but the key idea is that it relies on the inner products of observations rather than the observations themselves.

Linear Product

- The inner product of two r-vectors a and b is $< a, b >= \sum_{r=1}^{r} a_i b_i$
- It can be shown that
  - The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i < x, x_i > \tag{9.18}$$

  - where there are n parameters α_i,$i = 1, ..., n$, one per (9.17) (9.18) training It can be shown that observation.
  - To estimate the parameters $\alpha_1, \ldots, \alpha_n$and β_0,all we need are the inner products $\binom{n}{2}$ between all pairs of training observations.($\binom{n}{2} = n(n-1)/2$).

SVM in linear product

- To evaluatee $f(x)$, we need to compute the inner product between the new point x and each of the training points x_i.
- α_i is nonzero only for the support vectors in the solution
  - if a training observation is not a support vector, then its α_i equals zero.
- So if S is the collection of indices of these support points, we can rewrite any solution function of the form (9.18) as

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i < x, x_i > \tag{9.19}$$

  - typically involves far fewer terms than in (9.18).
- To summarize, in representing the linear classifier $f(x)$, and in computing its coefficients, all we need are inner products.

Kernel

- Suppose we replace the inner product in the support vector classifier with a generalized form, $K(x_i, x_{i'})$, known as a kernel.
- A kernel quantifies the similarity of two observations. For example, a linear kernel is:

$$K(x_i, x_{i'}) = \sum_{j=1}^{n} x_{ij} x_{i'j}, \tag{9.21}$$

which corresponds to the support vector classifier and quantifies similarity using Pearson correlation.

Non-linear Kernel

- A polynomial kernel of degree d is an alternative form:

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d, \tag{9.22}$$

  which leads to a more flexible decision boundary by fitting a support vector classifier in a higher-dimensional space involving polynomials of degree d.
- When combined with a non-linear kernel, the support vector classifier becomes a support vector machine, with the decision function:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i), \tag{9.23}$$

  where S is the set of support vectors.

[Illustration]Radial Kernel

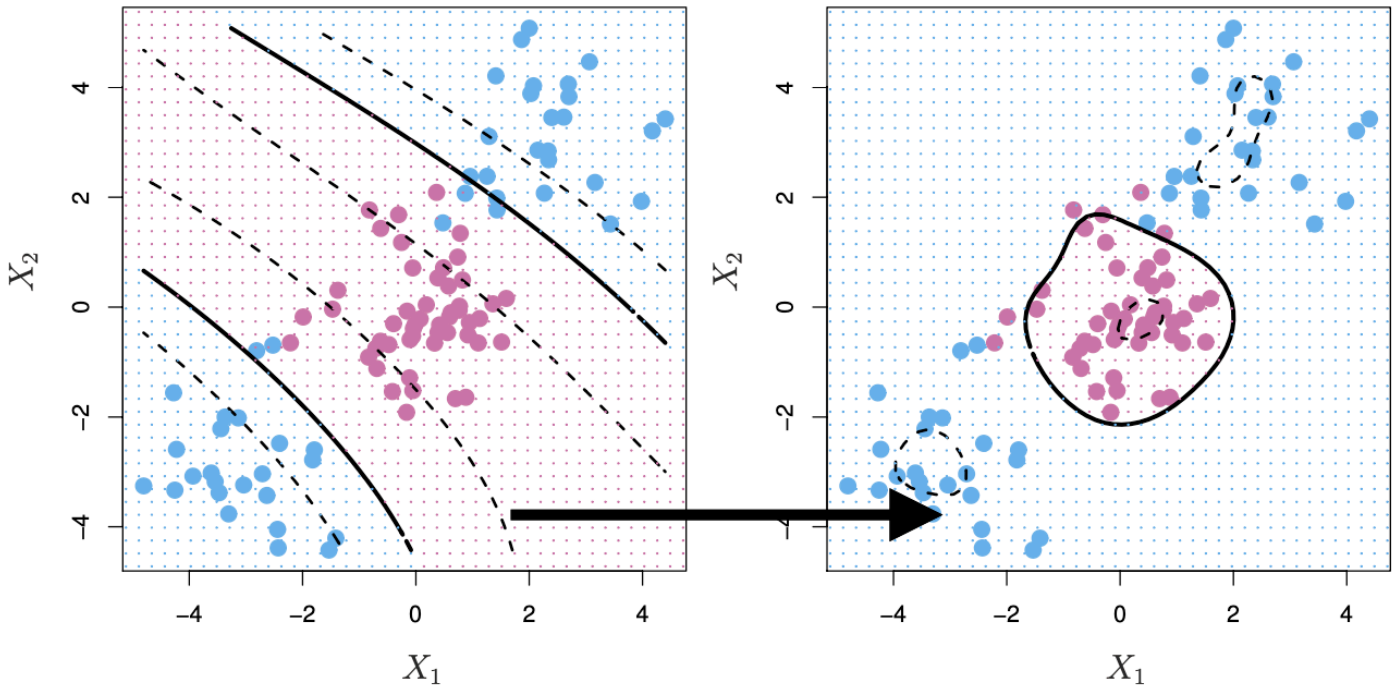

FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

Radial Kernel

- The polynomial kernel shown in (9.22) is one example of a possible non-linear kernel, but alternatives abound.
- Another popular choice is the radial kernel, which takes the form
  - $$K(x_i, x_{i'}) = exp(-\gamma(x_{ij} - x_{i'j})^2). \quad (9.24)$$
- In (9.24), γ is a positive constant.
- The right-hand panel of Figure 9.9 shows an example of an SVM with a radial kernel on this non-linear data; it also does a good job in separating the two classes.

How does kernel work?

- How does the radial kernel (9.24) actually work?
  - If a given test observation $x^* = (x_1^*, ..., x_p^*)$ is far from a training observation x_i in Euclidean distance ($\sum_{j=1}^{p} = (x_j^* - x_{ij})^2$ large), $K(x^*, x_{i\prime}) = exp(-\gamma(x_j^* - x_{i\prime j})^2)$ will be tiny.
  - x_i will play virtually no role in f(x∗). Recall that the predicted class label for the test observation x^* is based on the sign of $f(x^*)$.
  - observations that are far from x^* will play essentially no role in the predicted class label for x^*.
- This means that the radial kernel has very local polynomial kernel behavior, in the sense that only nearby training observations have an effect on the class label of a test observation.

Advantages using SVM?

- What is the advantage of using a kernel rather than simply enlarging the feature space using functions of the original features, as in (9.16)?
  - One advantage is computational, and it amounts to the fact that using kernels, one need only compute $K(x_i, x'_{i'})$ for all distinct pairs i, i'
    - This can be done without explicitly working in the enlarged feature space.
  - This is important because in many applications of SVMs, the enlarged feature space is so large that computations are intractable.
- For some kernels, such as the radial kernel (9.24), the feature space is implicit and infinite-dimensional, so we could never do the computations there anyway!

Data

- In Chapter 8 we apply decision trees and related methods to the Heart data.
- The aim is to use 13 predictors such as Age, Sex, and Chol in order to predict whether an individual has heart disease.
- We now investigate how an SVM compares to LDA on this data.
- After removing 6 missing observations, the data consist of 297 subjects, which we randomly split into 207 training and 90 test observations

[Illustration] Support Vector Classifier v.s. LDA-training

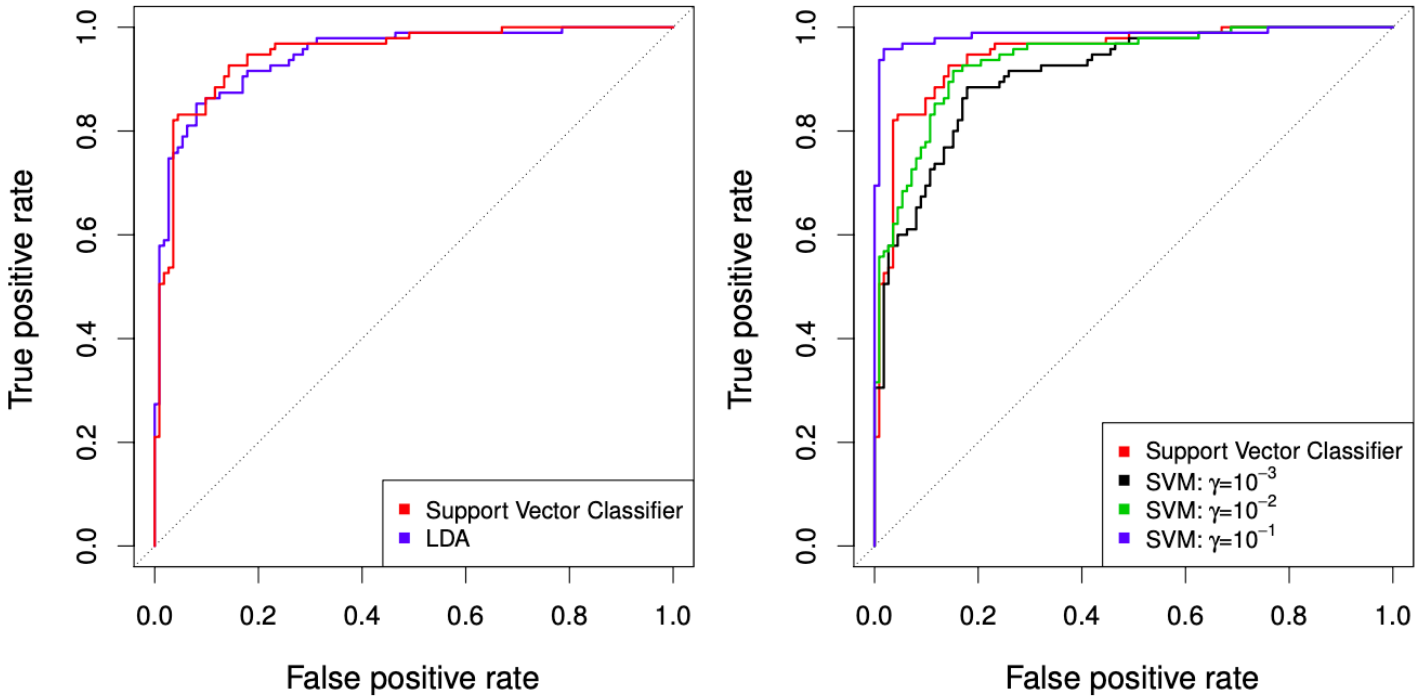

FIGURE 9.10. *ROC curves for the* `Heart` *data training set.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

[Illustration]Support Vector Classifier v.s. LDA -test

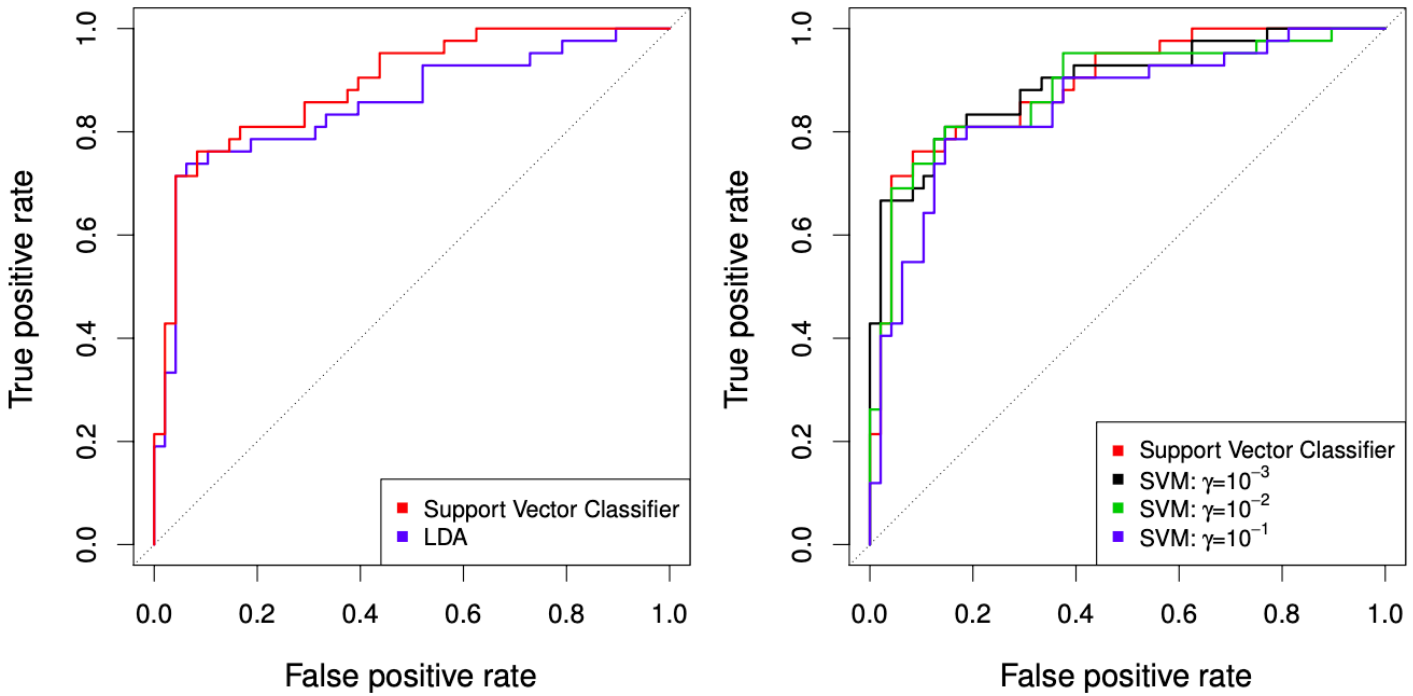

FIGURE 9.11. *ROC curves for the test set of the* `Heart` *data.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

Support Vector Classifier vs. LDA (1)

- **Comparison Setup:** Fit LDA and Support Vector Classifier (SVC) to the training data.
  - SVC is equivalent to an SVM with a polynomial kernel of degree $d = 1$.
- **ROC Curves:** Displayed in Figure 9.10 for both classifiers.
  - Classify based on scores: $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$.
  - ROC curve obtained by varying cutoff t and computing false positive and true positive rates.
  - Both LDA and SVC perform well; SVC may be slightly superior.

Support Vector Classifier vs. LDA (2)

- **Radial Kernel SVM:** ROC curves for SVMs with radial kernel and various γ values displayed in Figure 9.10 (right-hand panel).
  - ROC curves improve as γ increases (more non-linear fit).
  - $\gamma = 10^{-1}$ gives an almost perfect ROC curve on training data.
  - Training error rates can be misleading for test data performance.
- **Test Data Performance:** ROC curves on 90 test observations shown in Figure 9.11.
  - SVC has a small advantage over LDA.
  - SVM with $\gamma = 10^{-1}$ performs worst on test data.
  - More flexible methods may not always lead to improved test data performance.
  - SVMs with $\gamma = 10^{-2}$ and $\gamma = 10^{-3}$ perform comparably to SVC.

Outline

More than two classes

- So far, our discussion has been limited to the case of binary classification:
  - that is, classification in the two-class setting.
- How can we extend SVMs to the more general case where we have some arbitrary number of classes?
  - It turns out that the concept of separating hyperplanes upon which SVMs are based does not lend itself naturally to more than two classes.
- Though a number of proposals for extending SVMs to the K-class case have been made, the two most popular are the one-versus-one and one-versus-all approaches.

One-versus-one classification

- Suppose that we would like to perform classification using SVMs, and there are $K > 2$ classes.
  - A one-versus-one or all-pairs approach constructs $\binom{K}{2}$ SVMs, each of which compares a pair of classes.
  - For example, one such one SVM might compare the kth class, coded as +1, to the k'th class, coded as −1.
  - We classify a test observation using each of the $\binom{K}{2}$ classifiers, and we tally the number of times that the test observation is assigned to each of the K classes.
- The final classification is performed by assigning the test observation to the class to which it was most frequently assigned in these $\binom{K}{2}$ pairwise classifications.

One-versus-all classification

- The one-versus-all approach is an alternative procedure for applying SVMs in the case of K > 2 classes.
  - We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes.
  - Let $\beta_0^k, \beta_1^k, ..., \beta_p^k$ denote the parameters that result from fitting an SVM comparing the kth class (coded as +1) to the others (coded as –1).
  - Let x^* denote a test observation. We assign the observation to the class for which $\beta_0^k + \beta_1^k x_1^* + \ldots + \beta_p^k x_p^*$ is largest, as this amounts to a high level of confidence that the test observation belongs to the kth class rather than to any of the other classes.

Outline

SVMs in contrast to Logistic Regressions [1]

- SVMs gained popularity in the mid-1990s due to their good performance and novel approach to classification.
- The concept of finding a separating hyperplane with some violations was different from classical methods like logistic regression.
- The use of a kernel to expand the feature space for non-linear class boundaries was seen as a unique characteristic.

SVMs in contrast to Logistic Regressions [2]

- Deep connections between SVMs and classical statistical methods have emerged over time.
- The support vector classifier can be rewritten as:

$$\text{minimize}_{\beta_0,\beta_1,\ldots,\beta_p} \left\{ \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \qquad (9.25)$$

  where λ is a tuning parameter.
- A large λ leads to a low-variance, high-bias classifier, while a small λ results in a high-variance, low-bias classifier.
- The $\lambda \sum_{j=1}^{p} \beta_j^2$ term in (9.25) is the ridge penalty term, highlighting the connection to ridge regression.

“Loss + Penalty” Form

- Equation (9.25) follows the “Loss + Penalty” form:

$$\text{minimize}_{\beta_0,\beta_1,\ldots,\beta_p} \left\{ L(\mathbf{X}, \mathbf{y}, \beta) + \lambda P(\beta) \right\}, \tag{9.26}$$

  where $L(\mathbf{X}, \mathbf{y}, \beta)$ is the loss function, $P(\beta)$ is the penalty function, and λ is a nonnegative tuning parameter.
- In logistic regression, the loss function is the squared error:

$$L(\mathbf{X}, \mathbf{y}, \beta) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_1 - \ldots - \beta_p x_p)^2.$$

- In the case of (9.25), the loss function is the hinge loss function:

$$L(\mathbf{X}, \mathbf{y}, \beta) = \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)].$$

- The hinge loss function is closely related to the loss function used in logistic regression.

[Illustration] Hinge Loss

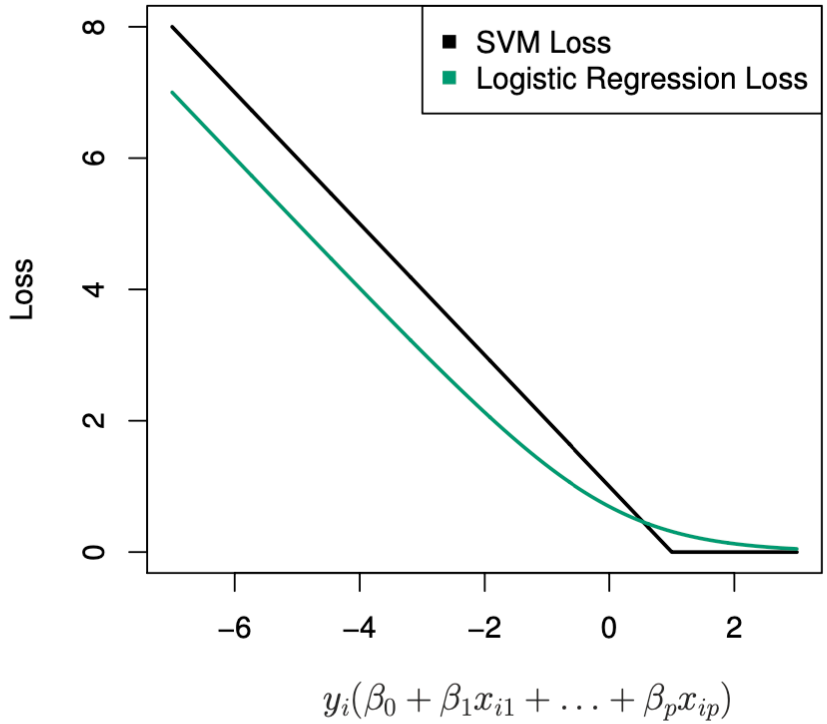

FIGURE 9.12. *The SVM and logistic regression loss functions are compared, as a function of* $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$. *When* $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$ *is greater than 1, then the SVM loss is zero, since this corresponds to an observation that is on the correct side of the margin. Overall, the two loss functions have quite similar behavior.*

- In the support vector classifier, only support vectors influence the classifier, as observations on the correct side of the margin do not affect it.
- The loss function is zero for observations on the correct side of the margin, as shown in Figure 9.12.
- In contrast, the loss function for logistic regression is not exactly zero anywhere but is very small for observations far from the decision boundary.
- Due to similarities in their loss functions, logistic regression and the support vector classifier often give similar results.
- SVMs tend to perform better when classes are well separated, while logistic regression is often preferred in more overlapping regimes.

For Further Reading I

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: