STATISTICAL LEARNING

CHAPTER 9: SUPPORT VECTOR MACHINES

INSTRUCTOR: SEOKHO LEE

HANKUK UNIVERSITY OF FOREIGN STUDIES

2015 SPRING

- The **support vector machine** (SVM) is an approach for classification developed in the computer science community in the 1990s
  - It has grown in popularity in many areas
  - It is often considered on of the best "out of the box" classifiers
- People often loosely refer to the maximal margin classifier, the support vector classifier, and the support vector machine as "support vector machines"

Maximal Margin Classifier

- The **maximal margin classifier** is a simple and intuitive classifier
- The support vector machine is a generalization of the maximal margin classifier

What Is a Hyperplane?

- In a p-dimensional space, a **hyperplane** is a flat affine subspace of dimension $p - 1$
  - The word **affine** indicates that the subspace need not pass through the origin
  - 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
- Two-dimensional space
  - A hyperplane is defined by the equation

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

  - When we say that (9.1) "defines" the hyperplane, we mean that any $X = (X_1, X_2)^T$ for which (9.1) holds is a point on the hyperplane
  - Note that (9.1) is simply the equation of a line, since indeed in two dimensions a hyperplane is a line

What Is a Hyperplane?

- p-dimensional space
  - (9.1) can be easily extended to the p-dimensional setting:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{9.2}$$

  - If a point $X = (X_1, X_2, \ldots, X_p)^T$ in p-dimensional space (i.e., a vector of length p) satisfies (9.2), then X lies on the hyperplane
  - We can think of the hyperplane as dividing p-dimensional space into two halves
    - One group of points X lies on one side of the hyperplane

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \tag{9.3}$$

    - The other group of points X lies on the other side of the hyperplane

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \tag{9.4}$$

  - A hyperplane in two-dimensional space is shown in Figure 9.1

What Is a Hyperplane?

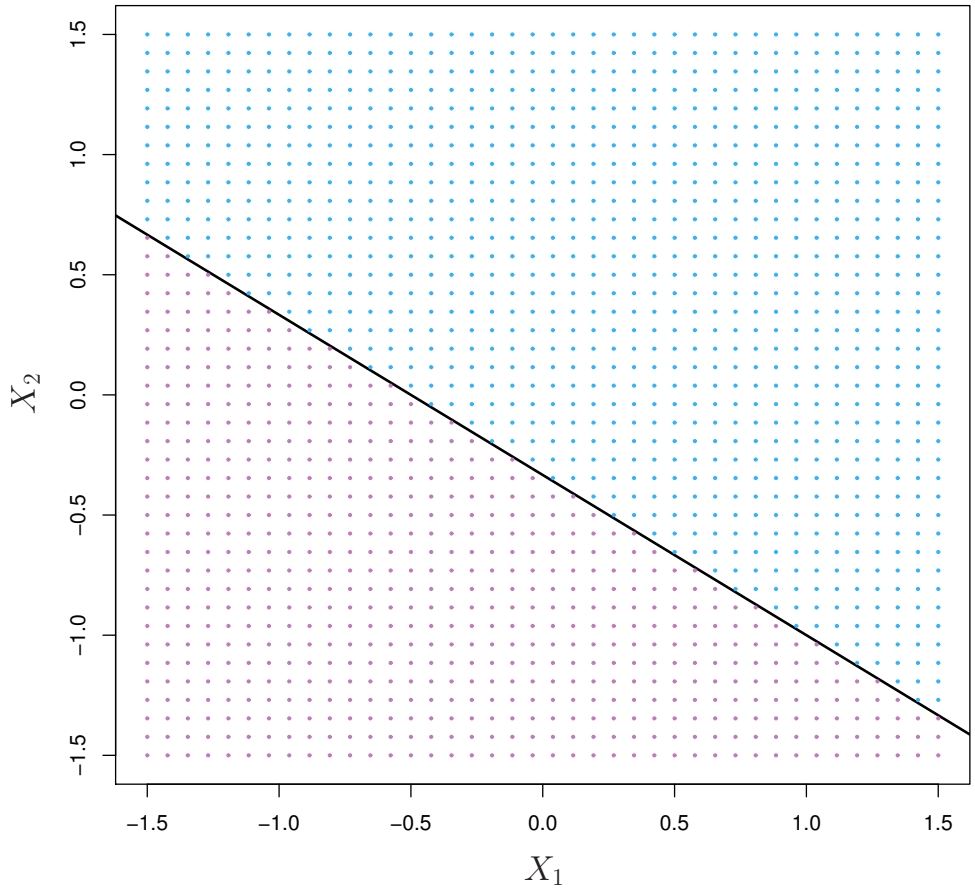

Figure 9.1: The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

Classification Using a Separating Hyperplane

- Setup
  - n training observations $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$
  - Features: $n \times p$ matrix $\mathbf{X} = (x_1, x_2, \cdots, x_n)^T$ with

$$x_1 = (x_{11}, \cdots, x_{1p})^T, \ \ldots \ , \ x_n = (x_{n1}, \cdots, x_{np})^T \tag{9.5}$$

  - Class label (response): $y_1, \ldots, y_n \in \{-1, 1\}$
    - -1 represents one class and 1 the other class
  - A test observation: $x^* = (x_1^*, \ldots, x_p^*)^T$

- Our goal is to develop a classifier used on the training data that will correctly classify the test observation using its feature measurements

Classification Using a Separating Hyperplane

- Suppose it is possible to construct a hyperplane that separates the training observation perfectly (See Figure 9.2)
  - A separating hyperplane has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 \quad \text{if } y_i = 1 \tag{9.6}$$

    and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 \quad \text{if } y_i = -1 \tag{9.7}$$

  - Equivalently,

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) > 0 \quad \text{for all } i = 1, 2, \ldots, n \tag{9.8}$$

- 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 it to class 1
  - If $f(x^*)$ is negative, then we assign it to class -1

Classification Using a Separating Hyperplane

- 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 confident about our class assignment for x^*
  - 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^*
- A classifier that is based on a separating hyperplane leads to a linear decision boundary

Classification Using a Separating Hyperplane

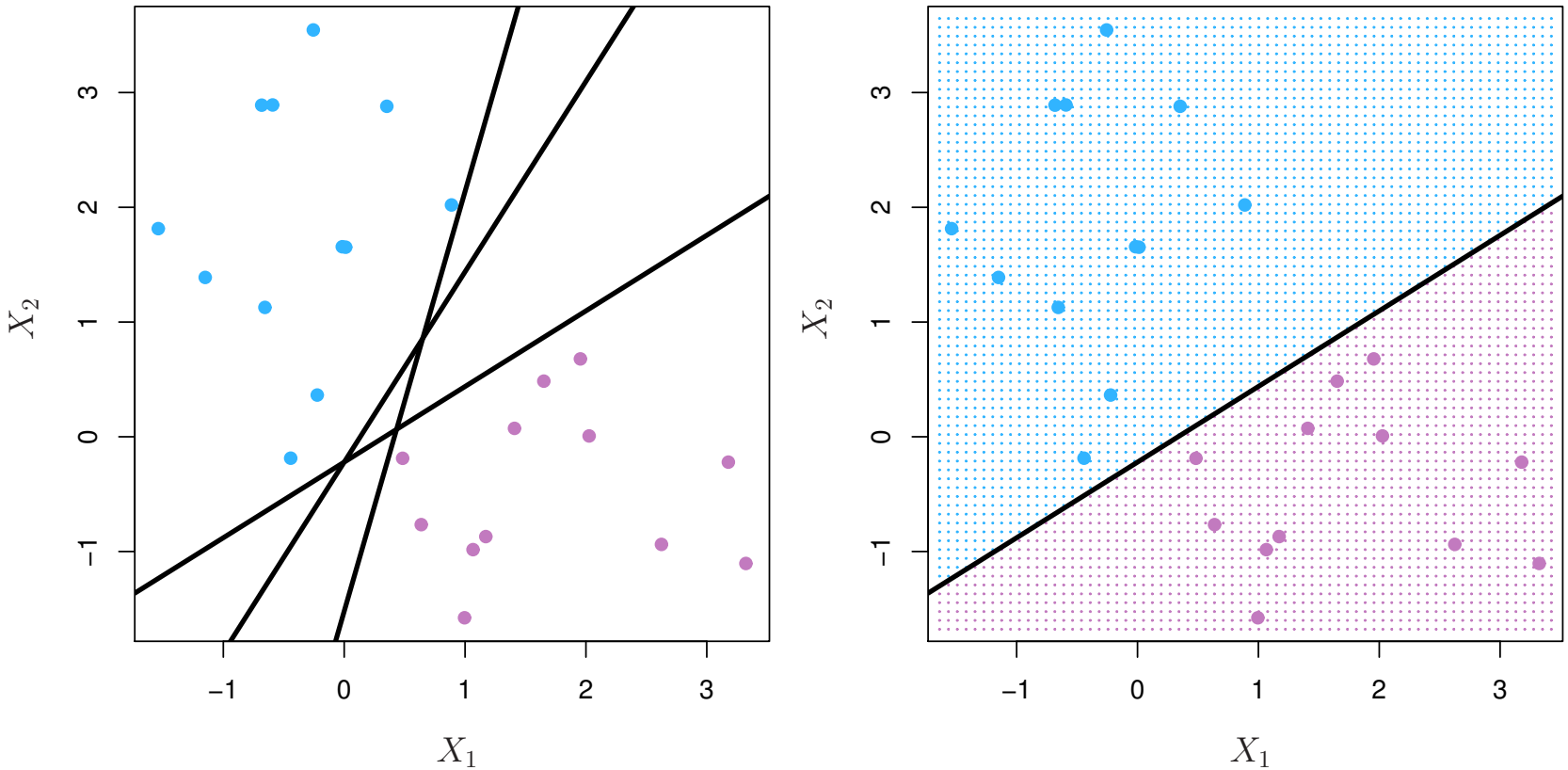

Figure 9.2: Left: There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision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

The Maximum Margin Classifier

- If data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes
- We must have a reasonable way to decide which of the infinite possible separating hyperplanes to use
- A natural choice is the **maximal margin hyperplane** (also known as the **optimal separating hyperplane**) (See Figure 9.3)
  - The **margin** is the smallest distance from the observations to the hyperplane
  - The maximal margin hyperplane is the hyperplane whose margin is the largest among the separating hyperplanes
  - The **maximal margin classifier**: a test observation is classified based on which side of the maximal margin hyperplane it lies
- Although the maximal margin classifier is often successful, it can also lead to overfitting when p is large

The Maximum Margin Classifier

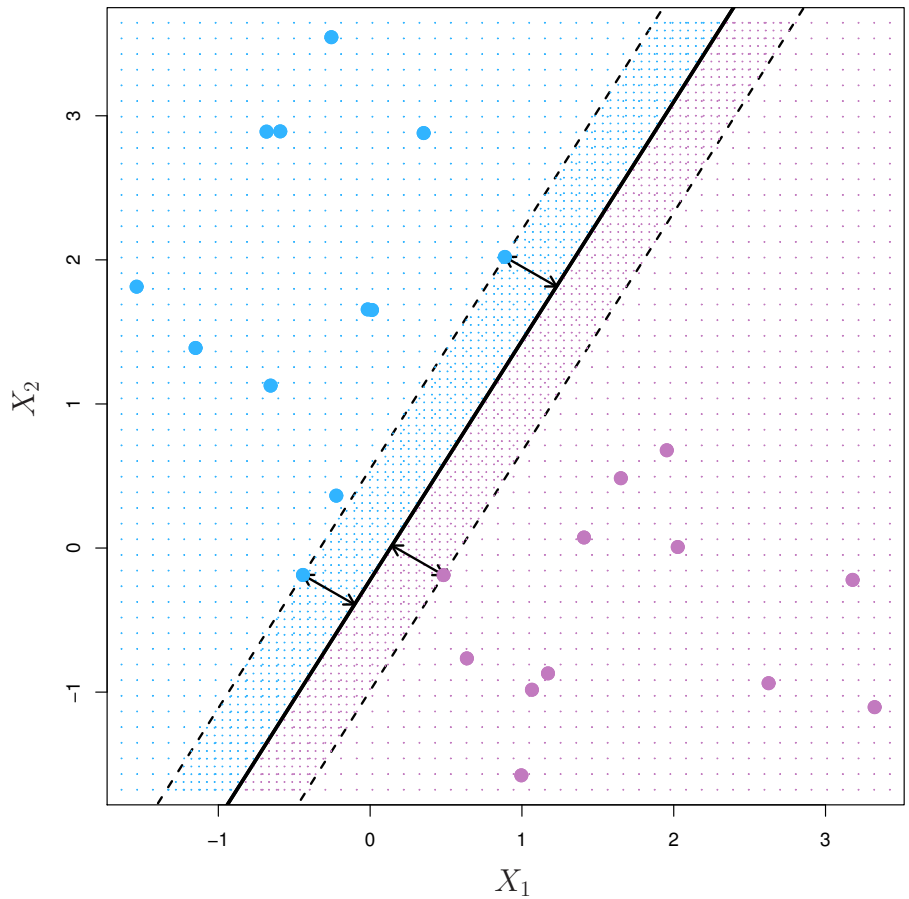

Figure 9.3: There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the margin is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.

The Maximum Margin Classifier

- The maximal margin hyperplane represents the mid-line of the widest "slab" that we can insert between the two classes
- **Support vectors**
  - In Figure 9.3, 3 training observations are equidistant from the maximal margin hyperplane and lie along the dashed lines indicating the width of the margin. These 3 observations are known as support vectors
  - Support vectors "support" the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well
  - Interestingly, the maximal margin hyperplane depends directly on the support vectors, but not on the other observations
- The fact that the maximal margin hyperplane depends directly on only a small subset of the observation is an important property

Construction of the Maximal Margin Classifier

- The maximal margin hyperplane is the solution to the optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}} \; M \tag{9.9}$$

$$\text{subject to} \; \sum_{j=1}^{p} \beta_j^2 = 1 \tag{9.10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \ldots, n \tag{9.11}$$

  - (9.11) guarantees that each observation will be on the correct side of the hyperplane, provided that M is positive
  - (9.10) is not really a constraint on the hyperplane
    - if $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = 0$ defines a hyperplane, then so does $k(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) = 0$ for any $k \neq 0$
    - However (9.10) ads meaning to (9.11); one can show that with this constraint the perpendicular distance from the ith observation to the hyperplane is given by $y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})$
    - Hence, M represents the margin of the hyperplane
  - The problem (9.9)–(9.11) can be solved efficiently (details are out of the scope)

The Non-separable Case

- In many cases no separating hyperplane exists, and so there is no maximal margin hyperplane (See Figure 9.4)
- In this case, the optimization problem (9.9)–(9.11) 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 non-separable case is known as the **support vector classifier**

The Non-separable Case

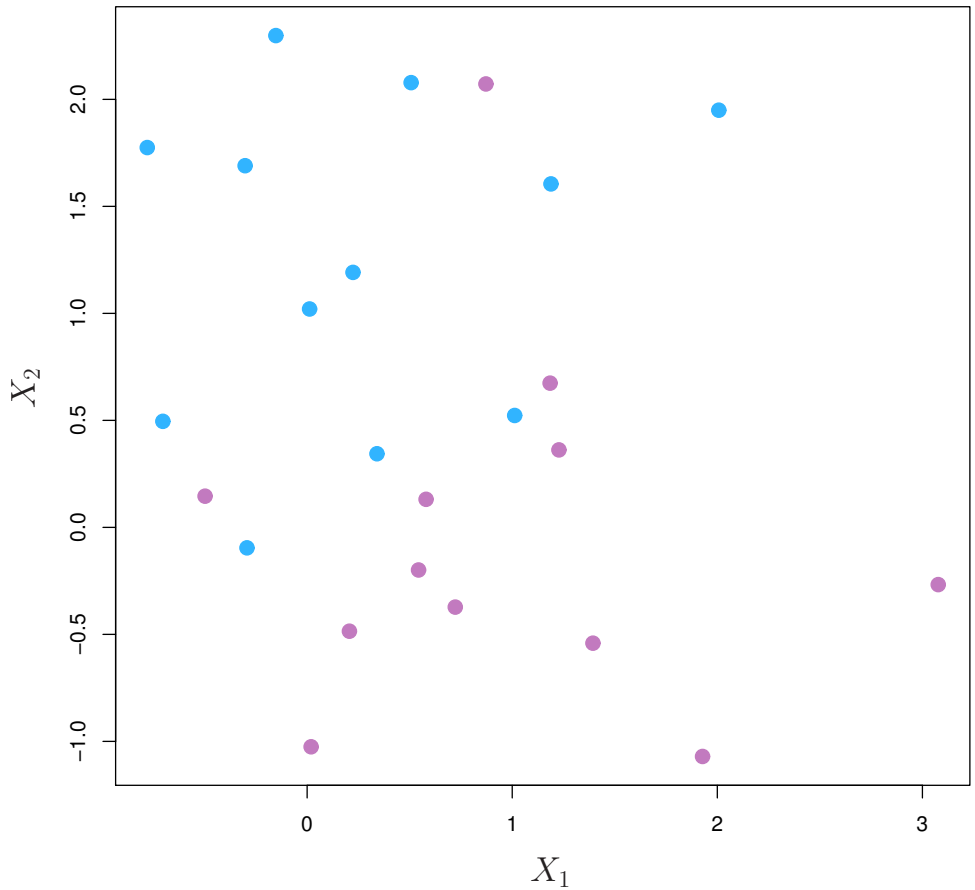

Figure 9.4: There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.

Support Vector Classifiers

- A classifier based on a separating hyperplane will necessarily perfectly classify all of the training observations
  - This can lead to sensitivity to individual observations
  - Figure 9.5
    - The addition of a single observation in Figure 9.5 leads to a dramatic change in the maximal margin hyperplane
    - The resulting maximal margin hyperplane is not satisfactory with a tiny margin
- We 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
- The **support vector classifier** (sometimes called a **soft margin classifier**) does exactly this
  - It allows some observations to be on incorrect side of the margin, or even in the incorrect side of the hyperplane
  - When there is no separating hyperplane, such a situation is inevitable (See the right panel of Figure 9.6)

Support Vector Classifiers

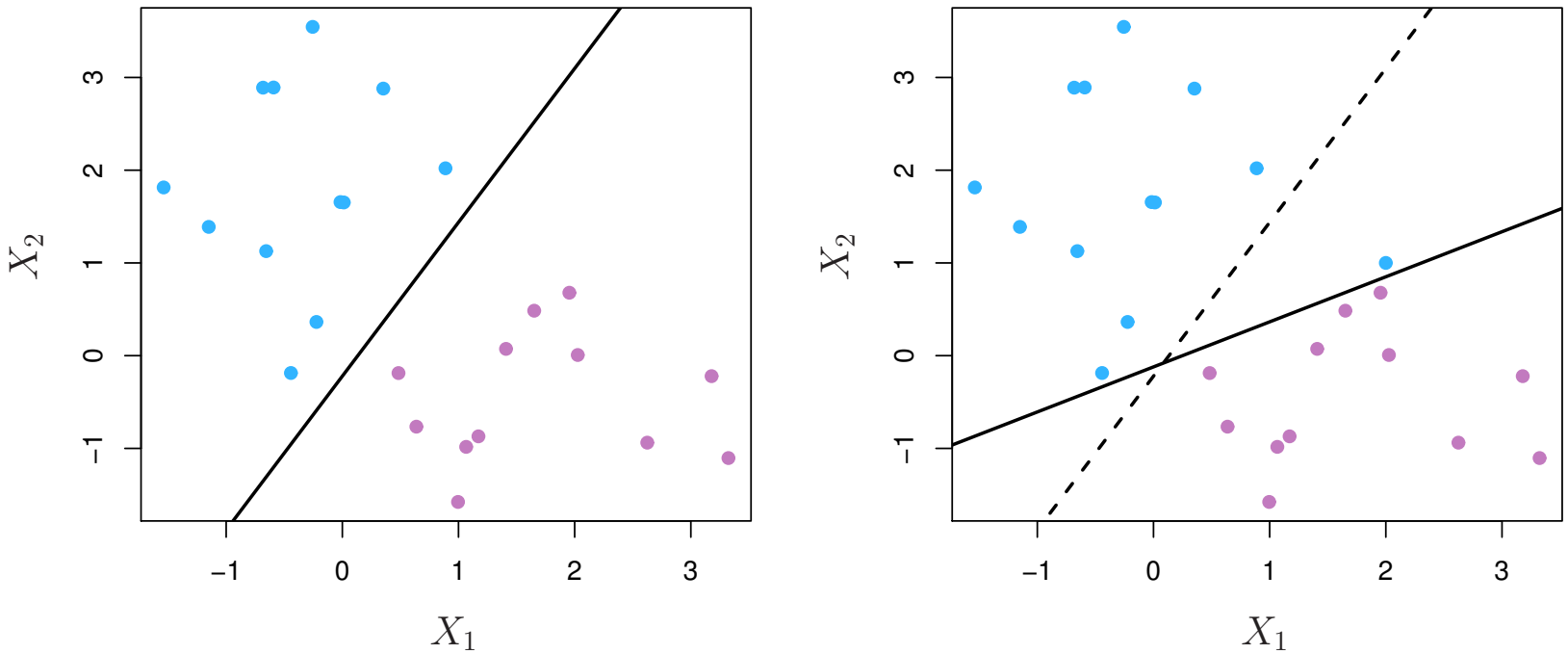

Figure 9.5: Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane. Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin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

Support Vector Classifiers

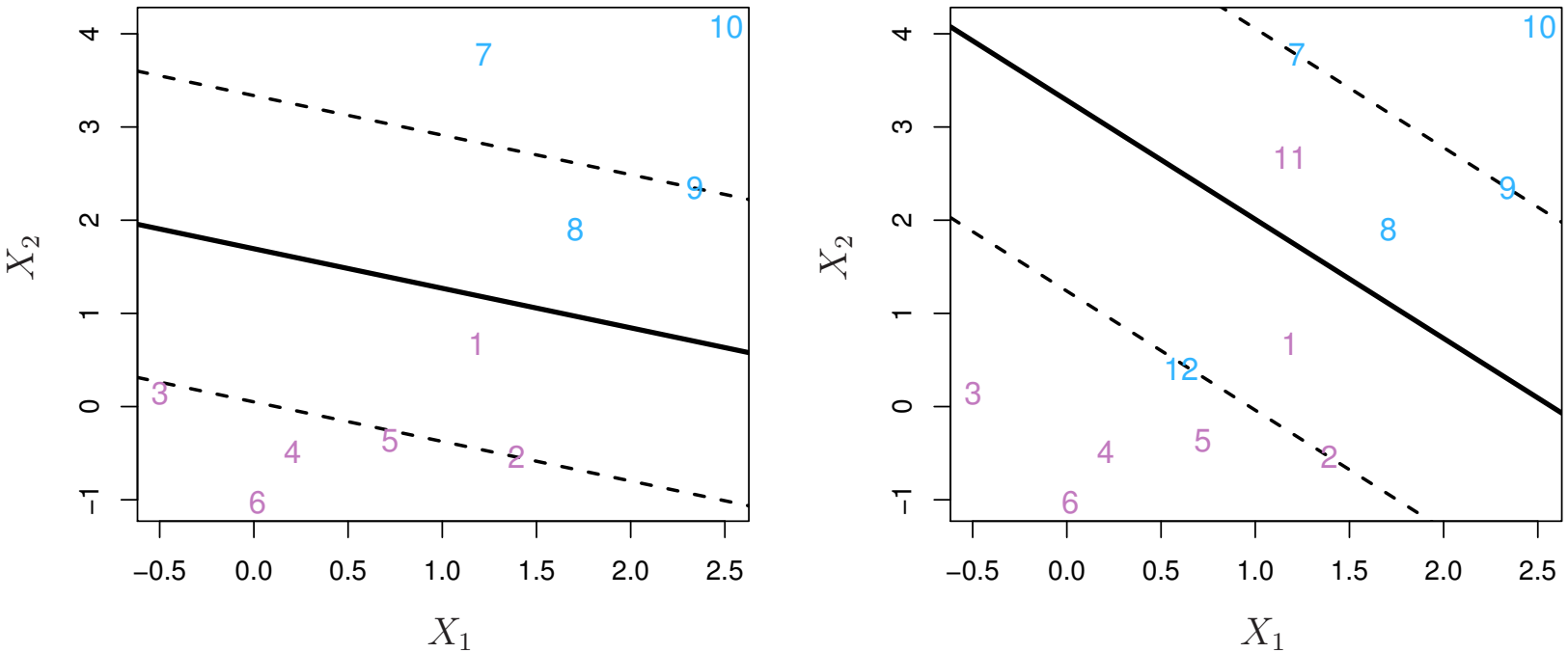

Figure 9.6: Left: A support vector classifier was fit to a small dad set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Purple observations: Observations 3, 4, 5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observation: 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 margins.

Details of the Support Vector Classifier

- The hyperplane from the support vector classifier 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

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M \tag{9.12}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.13}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{9.14}$$

$$\epsilon_i \geq 0, \; \sum_{i=1}^{n} \epsilon_i \leq C, \tag{9.15}$$

where C is a nonnegative tuning parameter

Details of the Support Vector Classifier

- M is the width of the margin
  - We seek to make M as large as possible
- $\epsilon_1, \ldots, \epsilon_n$: **slack variables**
  - They allow individual observations to be on the wrong side of the margin or the hyperplane
- Once we solve (9.12)–(9.15), we classify a test observation x^* as before, by simply determining on which side of the hyperplane it lies
  - We classify the test observation based on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$

Details of the Support Vector Classifier

- The slack variable ϵ_i tells us where the ith observation is located, relative to the hyperplane and relative to the margin
  - If $\epsilon_i = 0$ then the ith observation is on the correct side of the margin
  - If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin
    - We say that the ith observation has *violated* the margin
  - If $\epsilon > 1$ then it is on the wrong side of the hyperplane

- Role of the tuning parameter C
  - In (9.14), C bounds the sum of the ϵ_i's
  - C determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate
  - If $C = 0$ then there is no *budget* for violations to the margin, and it must be the case that $\epsilon_1 = \cdots = \epsilon_n = 0$
  - As the budget C increases, we become more tolerant of violations to the margin, and so the margin will widen (See Figure 9.7)
  - C is treated as a tuning parameter that is generally chosen via cross-validation. C controls the bias-variance trade-off of the statistical learning technique

Details of the Support Vector Classifier

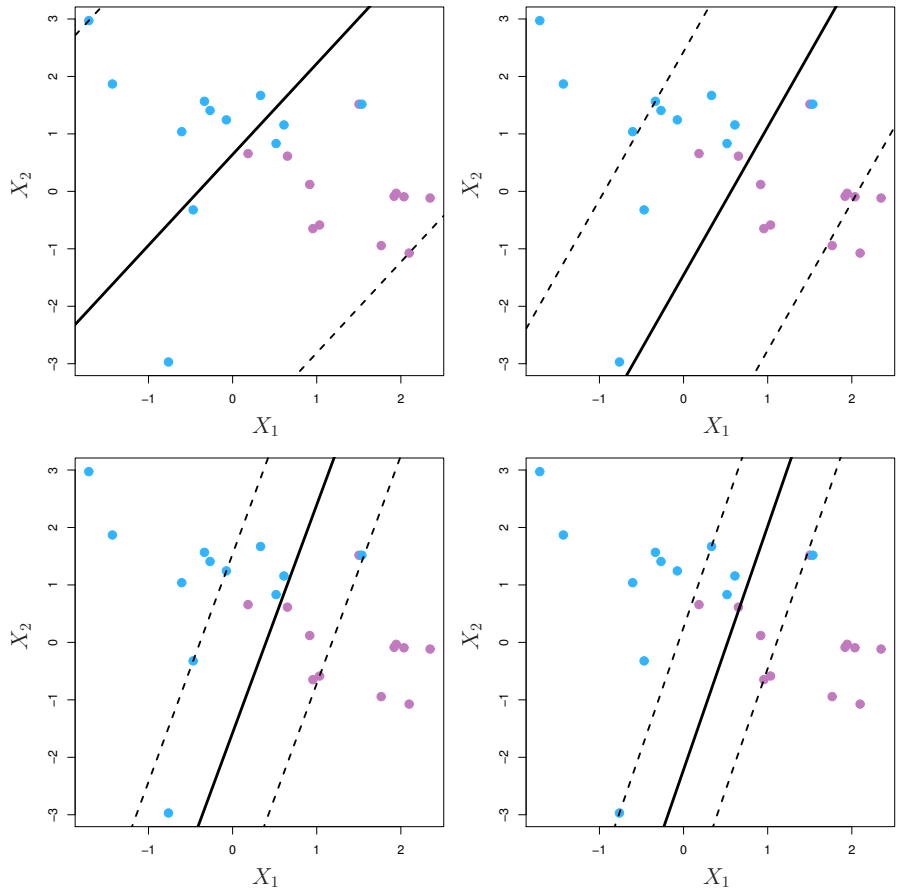

Figure 9.7: A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance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

Details of the Support Vector Classifier

- It turns out that only observations that either lie on the margin or that violate the margin will affect the hyperplane
  - An observation that lies strictly on the correct side of the margin does not affect the support vector classifier
  - Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as **support vectors**
  - The resulting classifier is quite robust to the behavior of observations that are far away from the hyperplane
    - This property is distinct from some of the other classification methods, such as LDA
    - The LDA classification rule depends on the mean of *all* of the observations within each class, as well as the within-class covariance matrix computed *all* of the observations
    - In contrast, logistic regression, unlike LDA, has very low sensitivity to observations far from the decision boundary

Classification with Non-linear Decision Boundaries

- The support vector classifier is a natural approach if the boundary is linear. However, in practice we are sometimes faced with non-linear class boundaries (See Figure 9.8)
- As in chapter 9, we could address the non-linear boundary by enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors
  - Rather than fitting a support vector classifiers using p predictors

    $$X_1, X_2, \ldots, X_p$$

    we could instead fit a support vector classifier using $2p$ features

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

  - Then (9.12)–(9.15) would become

$$\underset{\beta_0, \beta_{11}, \beta_{12}, \ldots, \beta_{p1}, \beta_{p2}, \epsilon_1, \ldots, \epsilon_n}{\text{maximize}} \; M \tag{9.16}$$

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

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

Classification with Non-linear Decision Boundaries

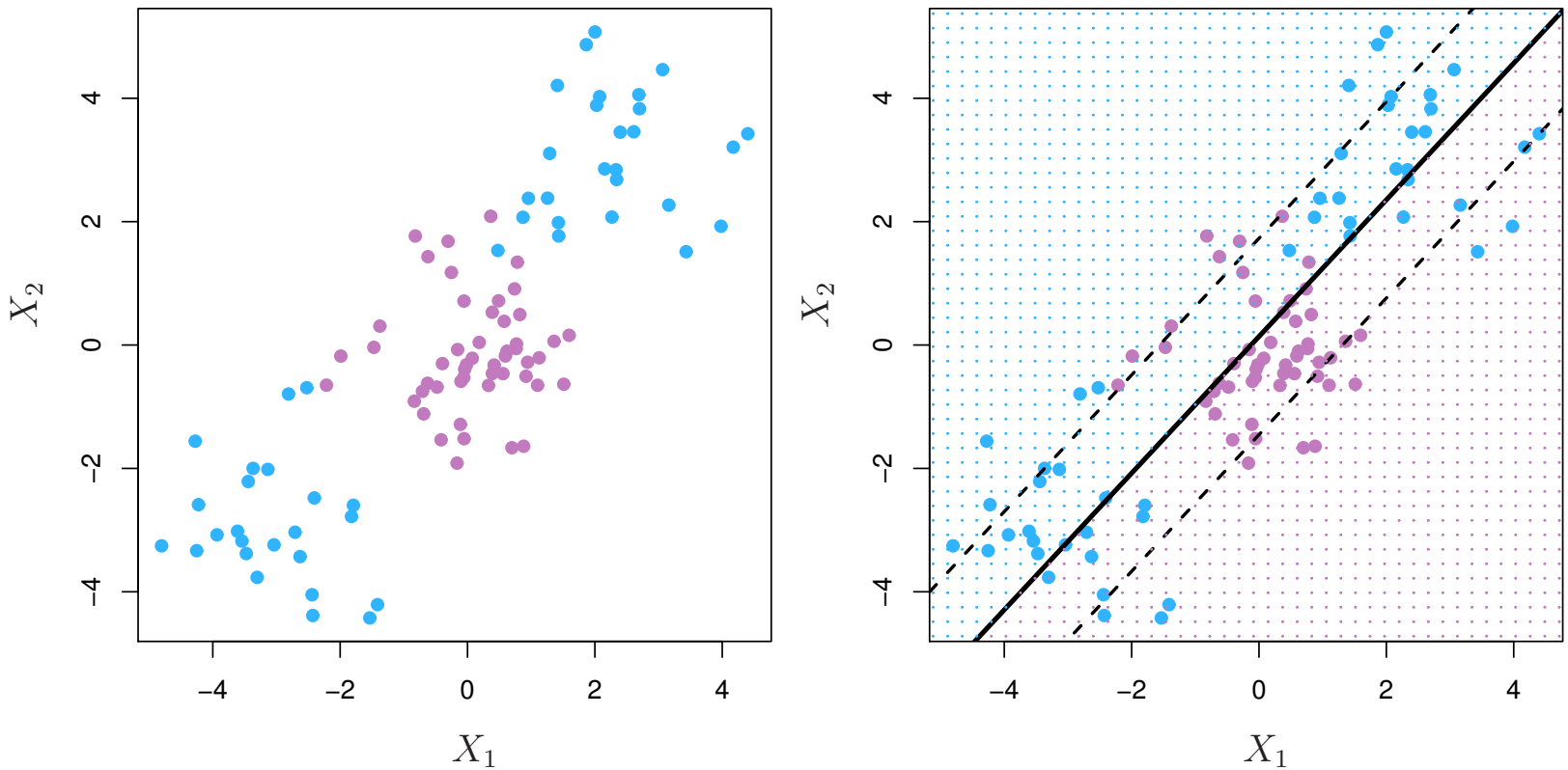

Figure 9.8: Left: The observations fall into two classes, with a non-linear boundary between them. Right: The support vector classifier seeks a linear boundary, and consequently performs very poorly.

Classification with Non-linear Decision Boundaries

- Why does (9.16) lead to a non-linear decision boundary?
  - In fact the decision boundary that results from (9.16) is linear, in terms of the modified $2p$ predictors
  - In the original feature space, the decision boundary is of the form $q(x) = 0$, where q is a quadratic polynomial, and its solutions are generally non-linear
  - One might additionally want to enlarge the feature space with higher-order polynomial terms, or with interaction terms of the form $X_j X_{j'}$
  - For non-linear boundary, other functions of the predictors could be considered rather than polynomials
  - In fact, there are many possible ways to enlarge the feature space, and computations would become unmanageable
- The support vector machine allows us to enlarge the feature space used by the support vector classifier in a way that leads to efficient computations

The Support Vector Machine

- The **support vector machine** (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**
- The solution to the support vector classifier problem (9.12)–(9.15) involves only the **inner products** of the observations
  - The inner product of two observations $x_i, x_{i'}$ is given by

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{9.17}$$

  - The linear support vector classifier can be represented as

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

  - The estimate the parameters $\alpha_1, \ldots, \alpha_n$ and β_0, all we need are the $\binom{n}{2}$ inner products $\langle x_i, x_{i'} \rangle$ between all pairs of training observations
  - α_i is nonzero only for the support vectors. Let $\mathcal{S}$ be the collection of support vectors. Then,

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

The Support Vector Machine

- To summarize, in representing the linear classifier $f(x)$, and in computing its coefficients, all we need are inner products
- Now we replace the inner product with a **generalization** of the inner product of the form

$$K(x_i, x_{i'}) \tag{9.20}$$

  - K is some function that we will refer to as a **kernel**
  - A kernel is a function that quantifies the similarity of two observations
  - **Linear kernel** for linear classifier

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

  - **Polynomial kernel** of degree d for dth polynomial decision boundary

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

The Support Vector Machine

- When the support vector classifier is combined with a non-linear kernel such as (9.22), the resulting classifier is known as a support vector machine
- In this case, the non-linear function has the form

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

- Another popular choice of possible non-linear kernels is the **radial kernel**, which takes the form

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right) \tag{9.24}$$

- See Figure 9.9 for a polynomial kernel and a radial kernel

The Support Vector Machine

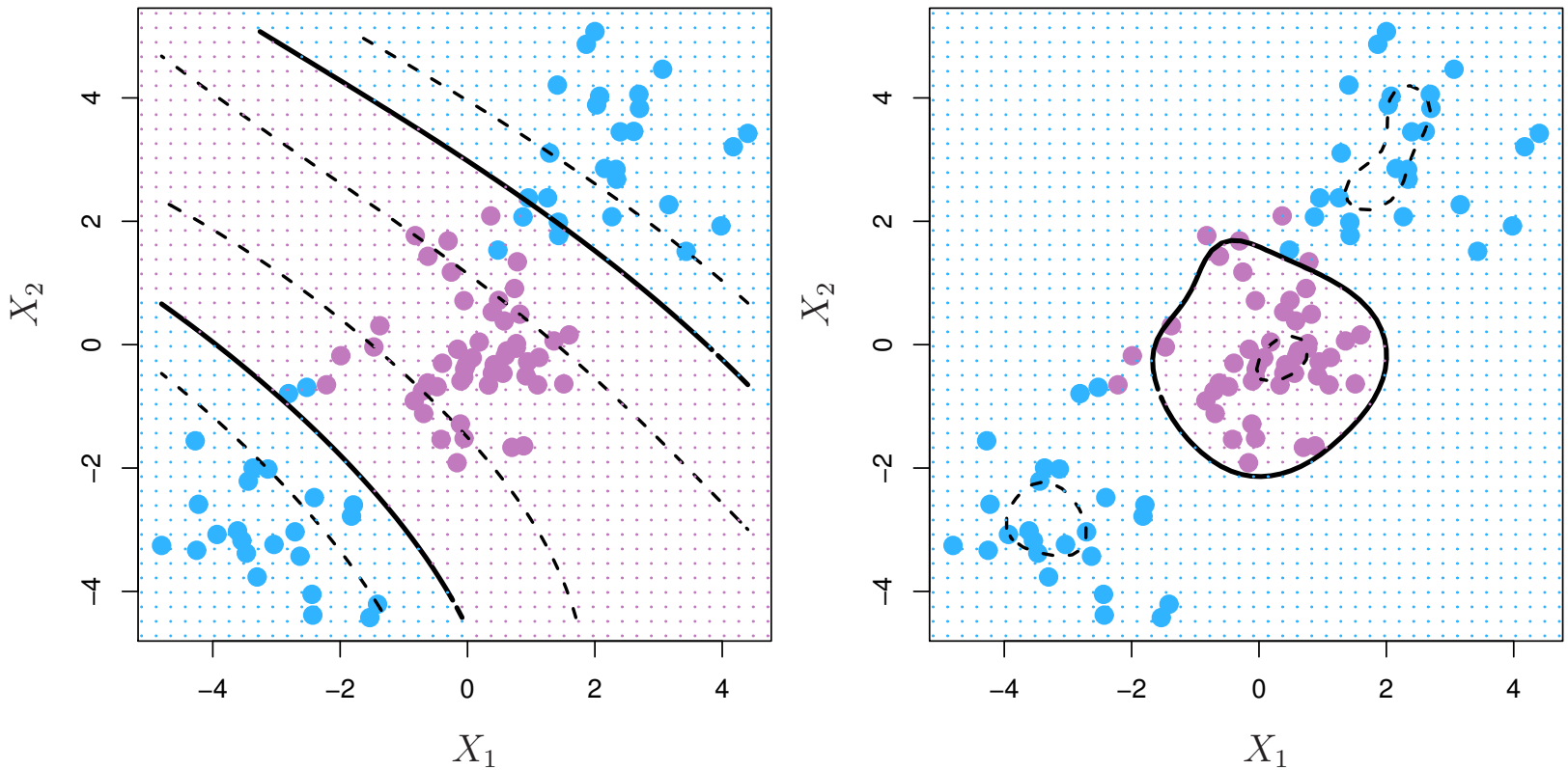

Figure 9.9: Left: An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule. Right: An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.

The Support Vector Machine

- Advantage of using a kernel rather than enlarging the feature space as in (9.16)
  - It has the computational advantage. Using kernels, we need only compute $K(x_i, x_{i'})$ for all $\binom{n}{2}$ distinct pairs i, i'
  - We do not need explicitly working in the enlarged feature space
  - In many applications of SVMs, the enlarged feature space is so large that computations are intractable (for example, radial kernel)
  - In the radial kernel, the feature space is *implicit* and infinite-dimensional, so we could never do the computations

An Application to the Heart Disease Data

- 207 training and 90 test observations
- Figure 9.10–ROC curves for training data
- Figure 9.11–ROC curves for test data

An Application to the Heart Disease Data

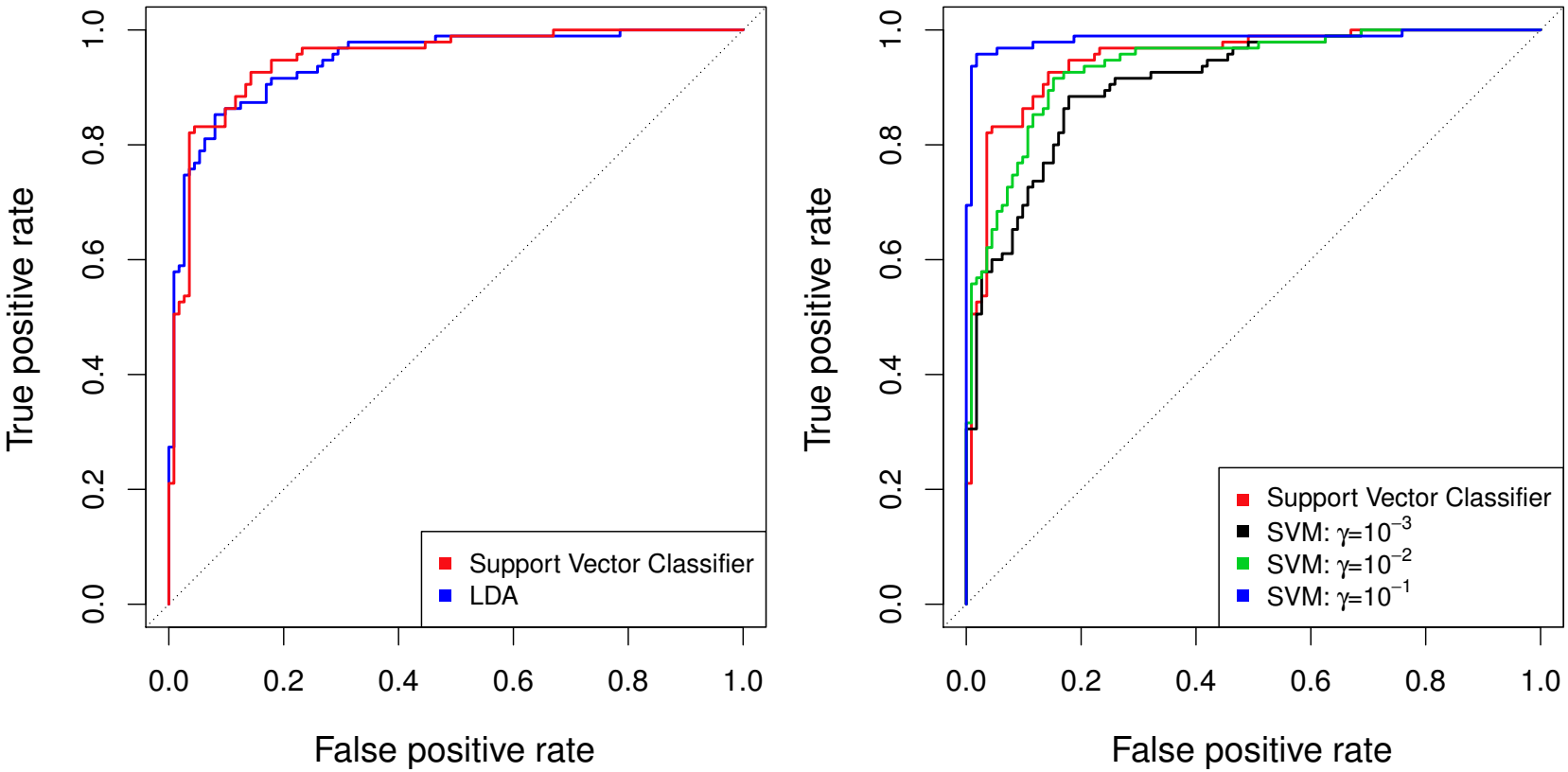

Figure 9.10: ROC curves for the Heart data training set. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}$, 10^{-2}, and 10^{-1}.

An Application to the Heart Disease Data

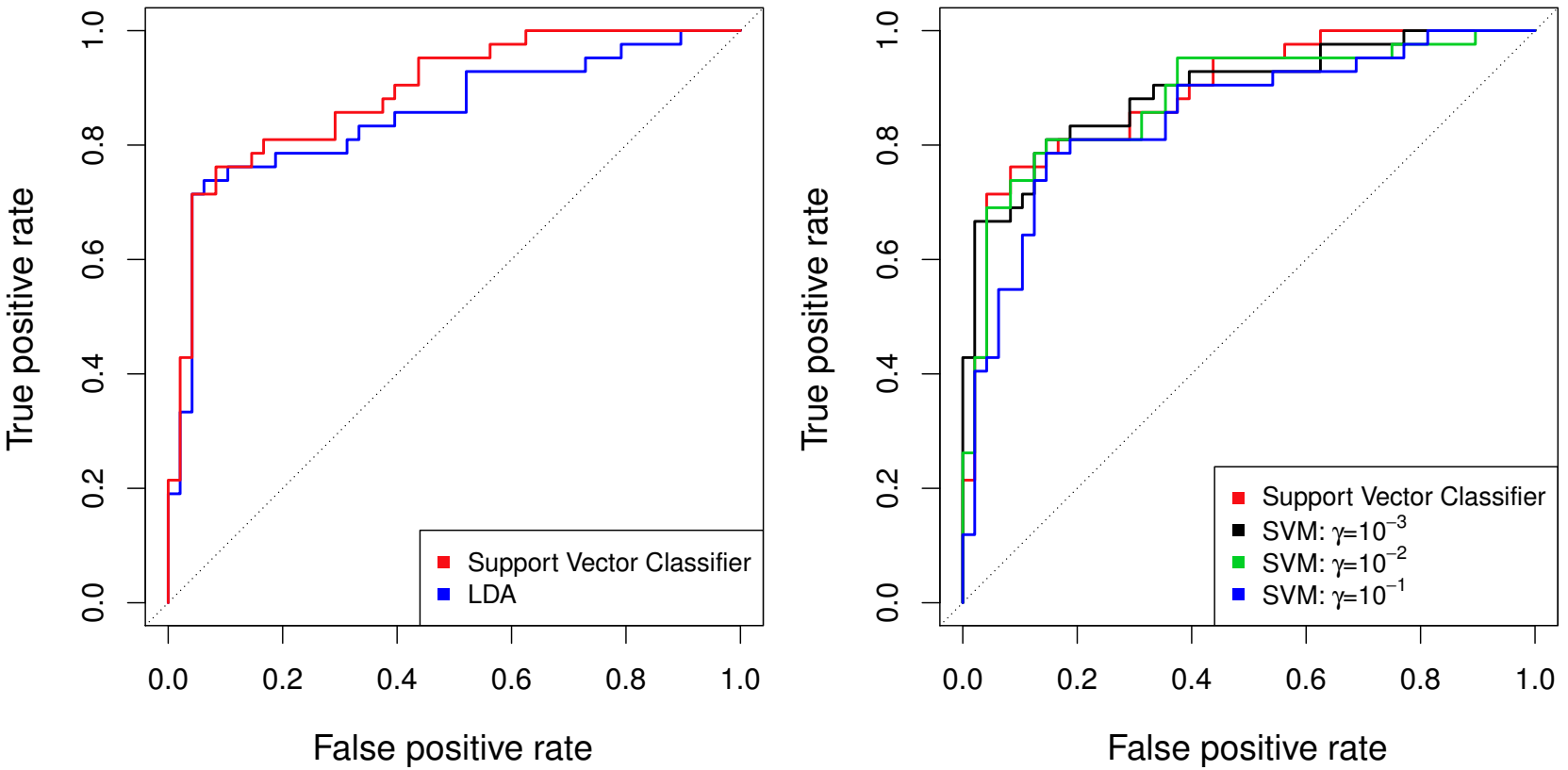

Figure 9.11: ROC curves for the test set of the Heart data. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}$, 10^{-2}, and $1-^{-1}$.

SVMs with More than Two Classes

- The concept f 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

- 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 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

- We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes
- Let $\beta_{0k}, \beta_{1k}, \ldots, \beta_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class (coded as $+1$) to the others (coded as -1)
- We assign the test observation x^* to the class for which $\beta_{0k} + \beta_{1k}x_1^* + \beta_{2k}x_2^* + \cdots + \beta_{pk}x_p^*$ is largest

Relationship to Logistic Regression

- Novelty in SVMs:
  When SVMs were first introduced in the mid-1990's, they made quite a splash in the statistical and machine learning communities. This was due in part to their good performance, good marketing, and also to the fact that the underlying approach seemed both novel and mysterious. The idea of finding a hyperplane that separates the data as well as possible, while allowing some violations to this separation, seemed distinctly different from classical approaches for classification, such as logistic regression and linear discriminant analysis. Moreover, the idea of using a kernel to expand the feature space in order to accommodate non-linear class boundaries appeared to be a unique and valuable characteristic
- Since that time, deep connections between SVMs and other more classical statistical methods have emerged

Relationship to Logistic Regression

- One can rewrite the criterion (9.12)–(9.15) for fitting the support vector classifier $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ as

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

  - λ is a nonnegative tuning parameter
    - When λ is large then $\beta_1, \ldots, \beta_p$ are small, more violations to the margin are tolerated, and a low-variance but high-bias classifier will result
    - When λ is small then few violations to the margin will occur; this amounts to a high-variance but low-bias classifier
    - A small value of λ in (9.25) amounts to a small value of C in (9.15)
  - The $\lambda \sum_{j=1}^{p} \beta_j^2$ term in (9.25) is the ridge penalty term from Section 6, and plays a similar role in controlling the bias-variance trade-off for the support vector classifier

Relationship to Logistic Regression

- Note that (9.25) takes the "Loss+Penalty" form

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

  - The loss function in (9.25) is known as **hinge loss**

$$L(\mathbf{X}, \mathbf{y}, \beta) = \sum_{i=1}^{n} \max[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})]$$

  - The hinge loss function is closely related to the loss function used in logistic regression (See Figure 9.12)

Relationship to Logistic Regression

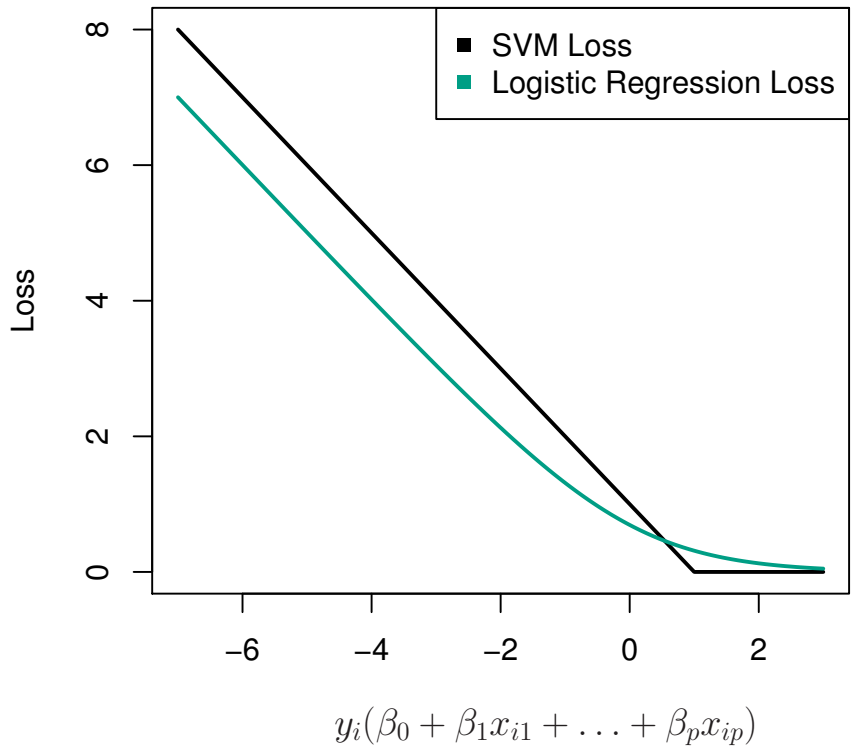

Figure 9.12: The SVM and logistic regression loss functions are compared, as a function of $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$. When $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$ is greater than 1, then the SVM loss is zero, since this corresponds to an observation that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

Relationship to Logistic Regression

- The hinge loss for the observations locating on the correct side of the margin is exactly zero
  - Those points satisfy $y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq 1$
  - Only support vectors play a role in the classifier obtained
  - The loss function for logistic regression is not exactly zero anywhere
- Due to the similarities between their loss functions, logistic regression and the support vector classifier often give very similar results
- When the class are well separated, SVMs tend to behave better than logistic regression; in more overlapping regimes, logistic regression is often preferred
- We can perform logistic regression or many of the other classification methods using non-linear kernels
  - For historical reasons, the use of non-linear kernels is much more widespread in the context of SVMs than in the context of logistic regression or other methods

Relationship to Logistic Regression

- There is an extension of the SVM for regression–**support vector regression**

Lab: Support Vector Machines