COMP6915 Machine Learning

Support Vector Machines

Dr. Lourdes Peña-Castillo
Departments of Computer Science and Biology
Memorial University of Newfoundland

Support Vector Machine (SVM)

- SVM is a method for classification developed in the 1990s
- It produces nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space
- It is quite popular and often considered one of the best “out-of-the-box” classifiers

Hyperplane

- In a *p*-dimensional space, a *hyperplane* is a flat affine (i.e., does not need to pass through the origin) subspace of dimension *p*-1.
  - In 2D, a hyperplane is a flat one-dimensional subspace: a line
  - In 3D, a hyperplane is a flat two-dimensional subspace: a plane

A p-dimensional hyperplane is defined by the equation

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

in the sense that if a point $X = (X_1, X_2, \ldots, X_p)^T$ in p-dimensional space satisfies this equation, then X lies on the hyperplane.

Suppose that X does not satisfy this equation; rather,

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

Then X lies to 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.

By calculating $\text{sign}[x^T \beta + \beta_0]$, one can determine on which side of the hyperplane a point lies.

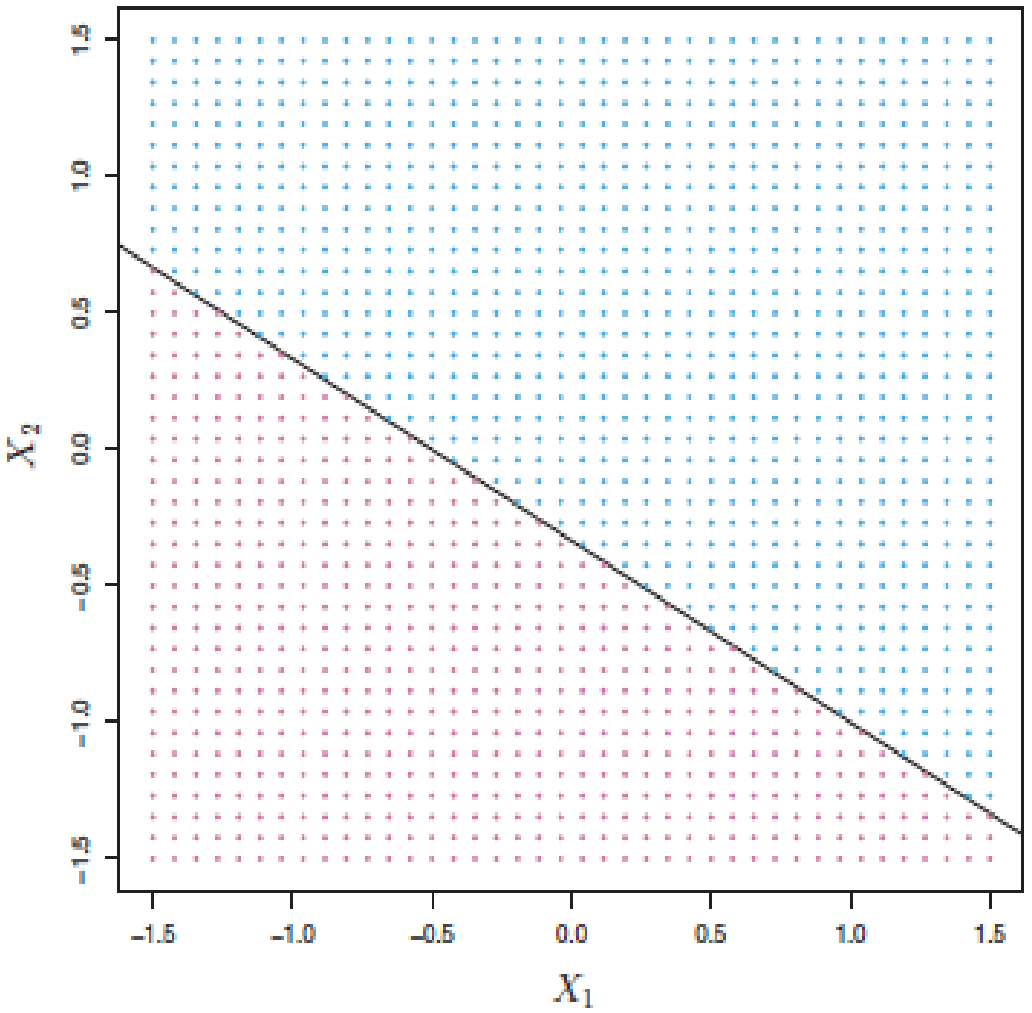


FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$, *and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$.

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Separating hyperplane classifier

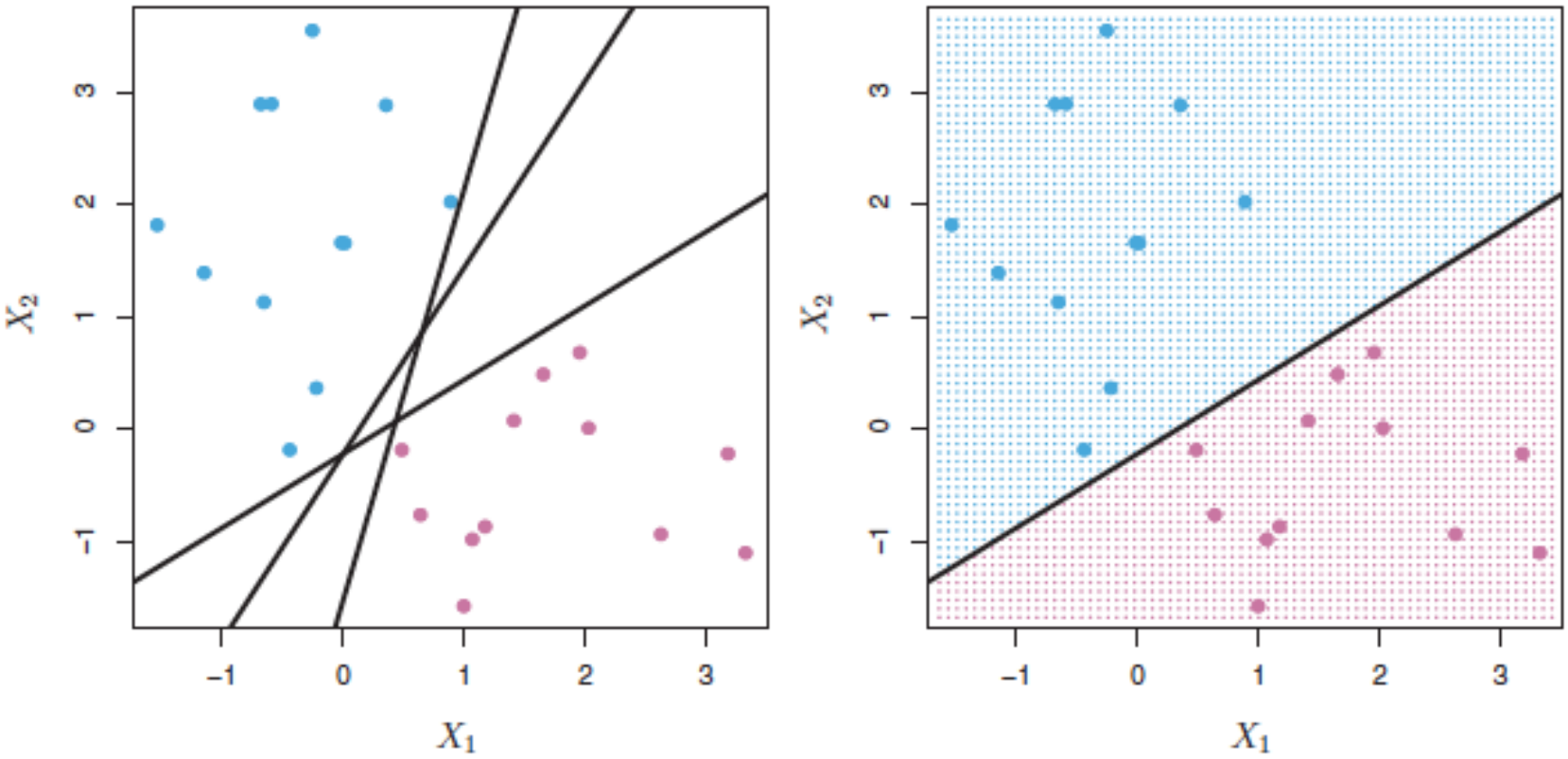


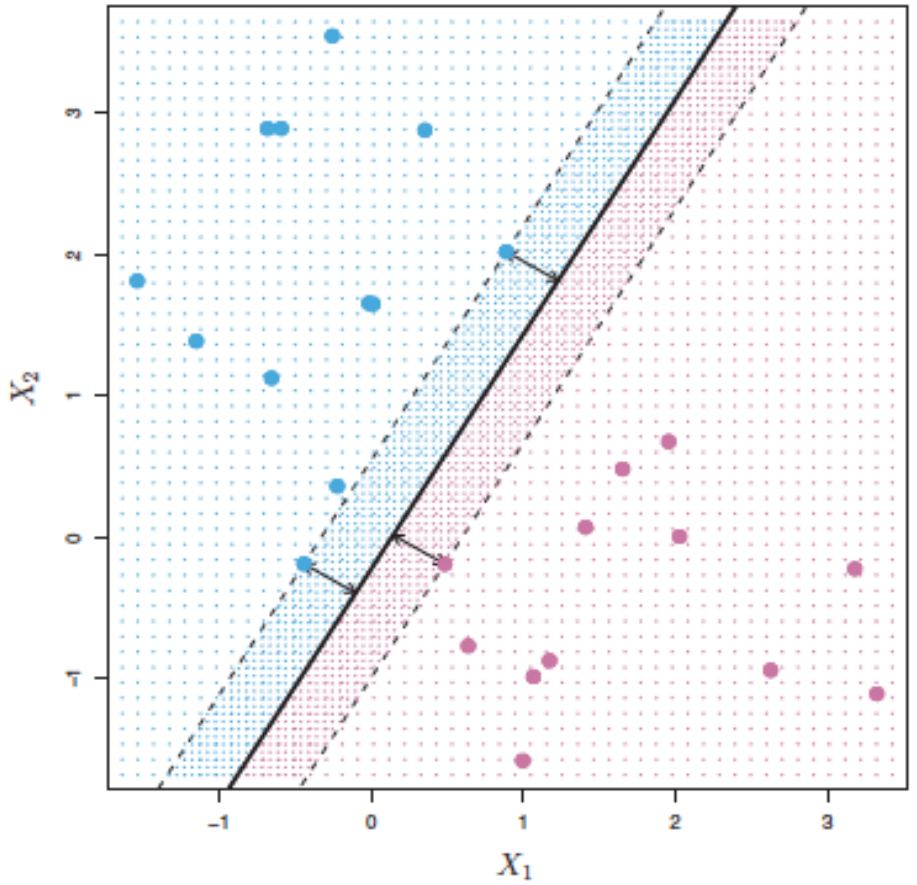
FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision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

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Maximal Margin Classifier

- If the two classes are perfectly separated, there will be an infinite number of separating hyperplanes.
- An optimal choice for a classifier is to find the hyperplane that creates the biggest margin between the training points for both classes
  - Margin is the minimal (perpendicular) distance from the observations to the hyperplane
  - The maximal margin classifier is the hyperplane that has the farthest minimum distance to the training observations

Maximal Margin Classifier



Support vectors are vectors in *p*-dimensional space that "support" the maximal margin hyperplane in the sense that if these points were moved then the maximal hyperplane would move as well. Note that the hyperplane does not depend on the other observations.

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.*

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Maximal Margin Classifier

The maximal margin hyperplane is the solution to the optimization problem

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} M$$

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

$$y_i(x^T\beta + \beta_0) \geq M \ \forall i = 1, \ldots, n.$$

Support Vector Classifier

- If the two classes overlap in feature space, there is no maximal margin classifier
- One way to deal with the overlap is to still maximize *M*, but allow some points to be on the wrong side of the margin
  - This is the support vector classifier

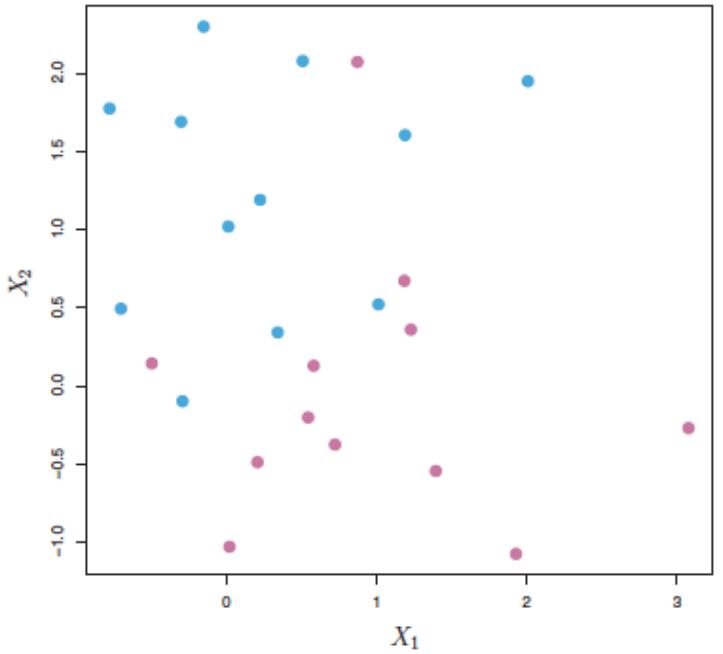


FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Support Vector Classifier

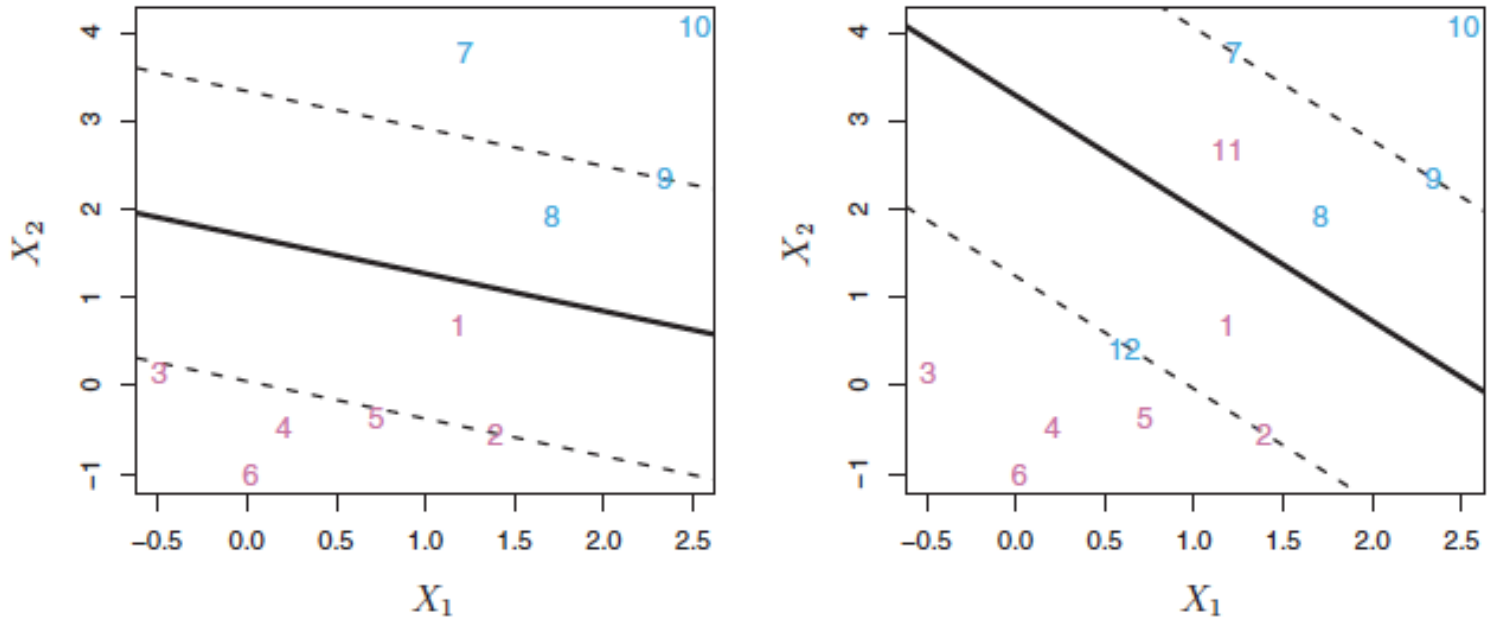


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation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

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Support Vector Classifier

The support vector classifier is the solution to the optimization problem

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\xi_1,\ldots,\xi_n} M$$

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

$$y_i(x^T\beta + \beta_0) \geq M(1-\xi_i),$$

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

where C is a nonnegative tuning parameter and $\xi_1, \ldots, \xi_n$ are slack variables that allow individual observations to be on the wrong side of the margin or of the hyperplane.

This problem is quadratic with linear inequality constraints. It ca be solved using Lagrange multipliers. For more details in how to solve it see ESLII Section 12.2.1

Support Vector Classifier

The slack variable ξ_i indicates where the ith observation is located relative to the margin and the hyperplane:

- If $\xi_i = 0$ then the ith observation is on the correct side of the margin.
- If $\xi_i > 0$ then the ith observation is on the wrong side of the margin (it has violated the margin).
- If $\xi_i > 1$ then it is on the wrong side of the hyperplane (it is missclassified).

Support Vector Classifier

The value ξ_i in the constraint
$y_i(x^T\beta + \beta_0) \geq M(1 - \xi_i)$
is the proportional amount by which the prediction $f(x_i) = x_i^T\beta + \beta_0$ is on the wrong side of its margin.

By bounding the $\sum \xi_i$, the total proportional amount by which predictions fall on the wrong side of their margin is bound. Misclassification occur when $\xi_i > 1$, so bounding $\sum \xi_i$ at a value C bounds the total number of training misclassifications at C.

C is generally chosen via cross-validation.

Support Vector Classifier

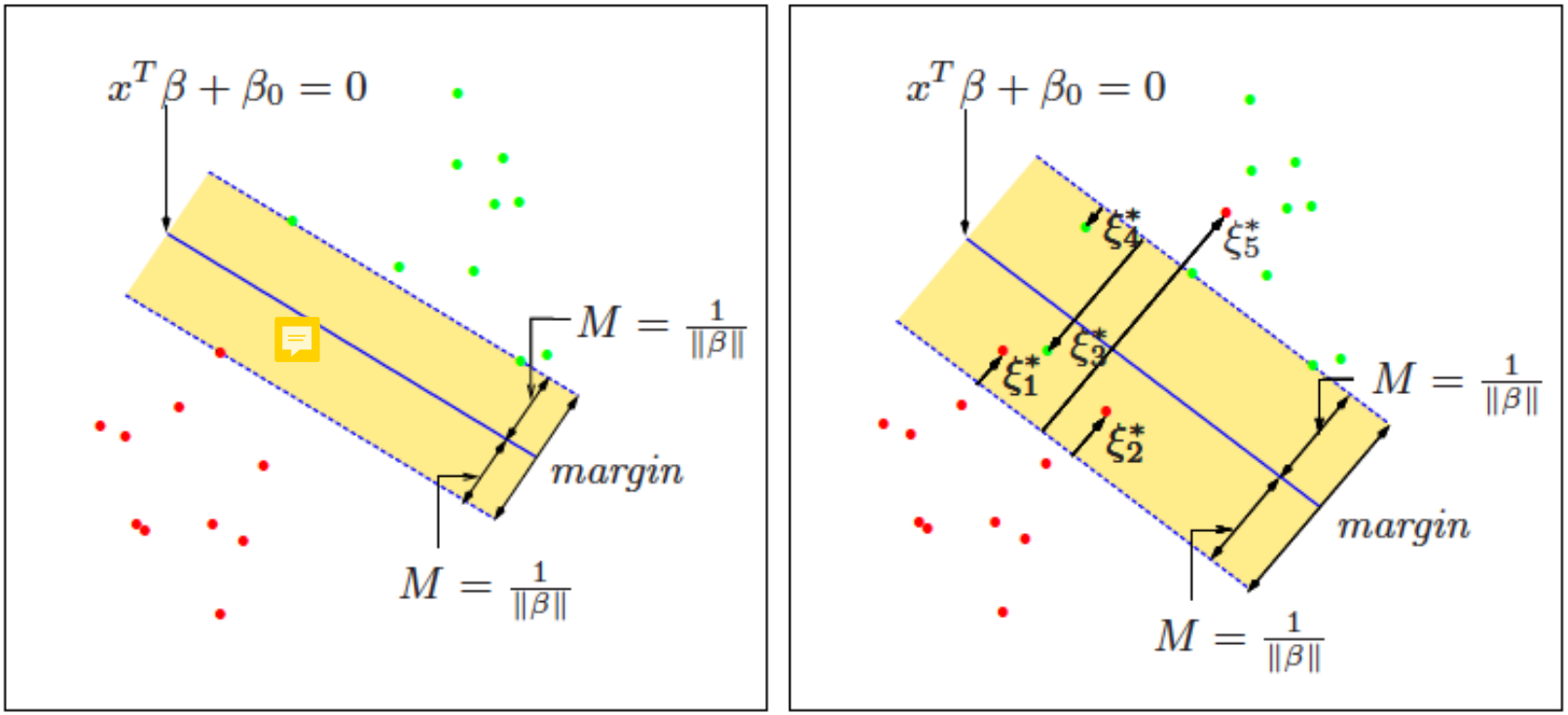


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

Figure from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman, 2009.

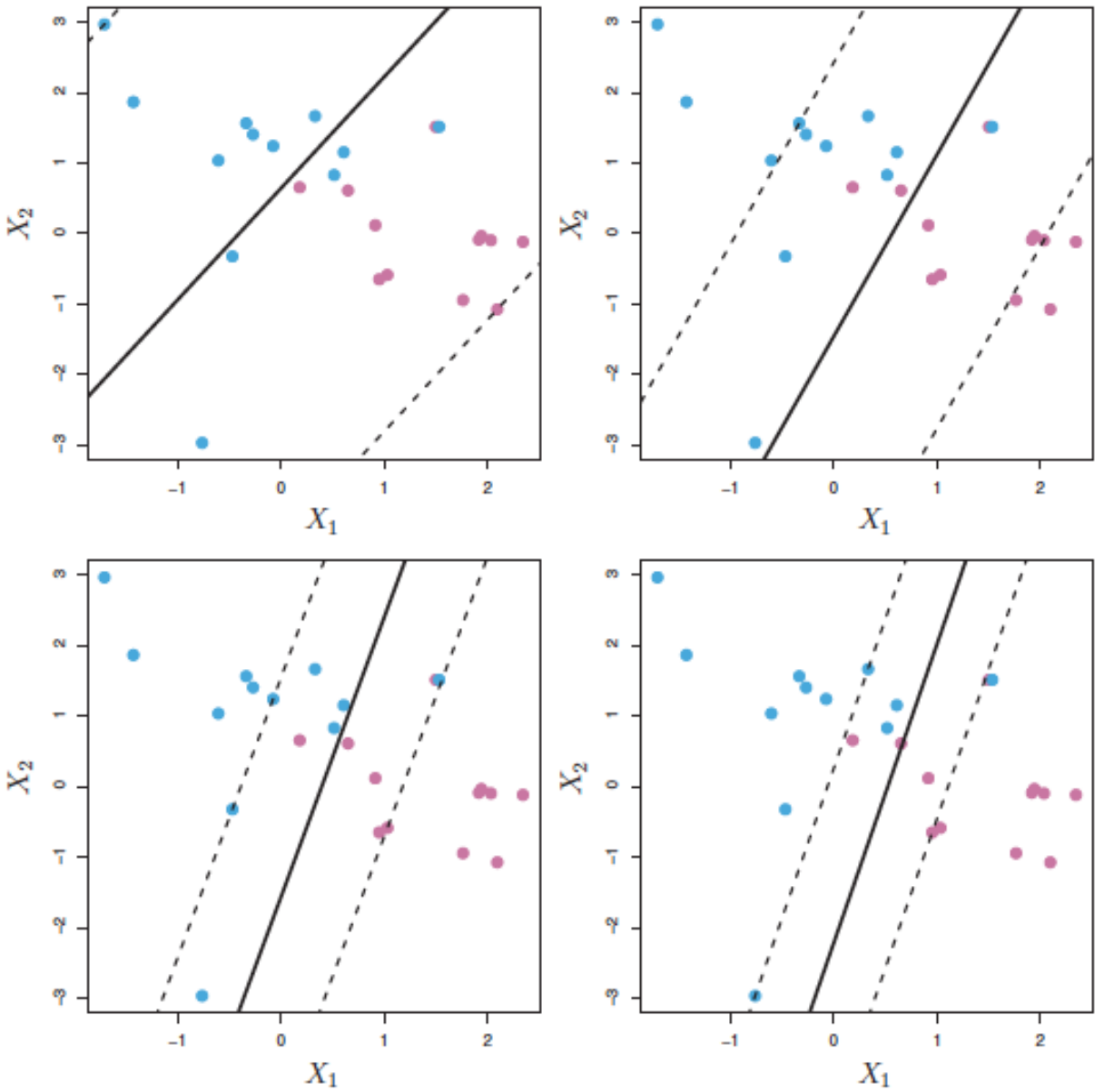


FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance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

C control the variance-bias trade-off of the support vector classifier.
When *C* is large, many observations are support vectors (i.e., they lie directly on the margin or on the wrong side of the margin). This classifier has low variance but potentially high bias.
If *C* is small, there will be fewer support vectors and the resulting classifier will have low bias but high variance.

What if C is equal to zero?

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Support Vector Classifier

- As the support vector classifier's decision rule is based only on a potentially small subset of the training observations (the support vectors), observations far away from the hyperplane have very little effect on it.
  - This differs from many of the other classification methods we have seen

Support Vector Classifier

- The support vector classifier works if the boundary between the two classes is linear
  - **How can it be extended to learn non-linear boundaries between classes?**

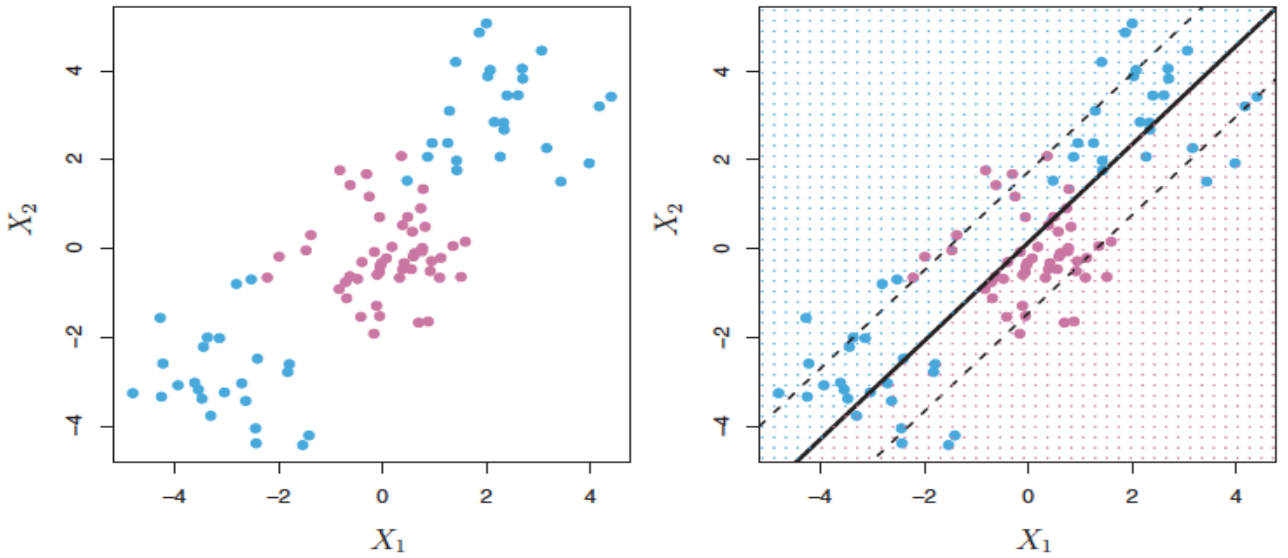


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

The solution to the support vector classifier optimization problem (slide 12) only involves the *inner products* of the input features.

The inner product of two r-vectors a and b is defined as $\langle a, b\rangle = \sum_{i=1}^{r} a_i b_i$.

Thus the inner product of two observations x_i, $x_{i'}$ is given by $\langle x_i, x_{i'}\rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$

where p is the number of features.

The linear support vector classifier can then be represented as

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

where there are n parameters α_i, $i = 1, \ldots, n$, one per training observation. However, α_i is nonzero only for the support vectors in the solution. If S is the collection of indices of these support points, $f(x)$ can be rewritten as

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

This could also be done for transformed feature vectors $h(x)$.

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

However, there is no need to specify the transformation $h(x)$, we only require knowledge of the kernel function

$$K(x, x_i),$$

where K is a function that quantifies the similarity between two observations (i.e., K computes inner products in the transformed space).

Popular choices for K in the literature are

$K(x_i, x_{i'}) = (1+\langle x_i, x_{i'}\rangle)^d$, polynomial kernel of degree d

$K(x_i, x_{i'}) = \exp(-\gamma||x_i-x_{i'}||^2)$, radial kernel

$K(x_i, x_{i'}) = \tanh(\kappa_1\langle x_i-x_{i'}\rangle+\kappa_2)$, neural network

Note that $||x_i - x_{i'}||^2 = \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$

Support Vector Machine

When the support vector classifier is combined with a non-linear kernel such as these, the resulting classifier is known as a support vector machine (SVM).

The (non-linear) function has the form

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

What do we get if we use $K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$**?**

Support Vector Machine

- In sum, SVM extends the support vector classifier by enlarging the feature space using kernels to enable learning non-linear boundaries between classes

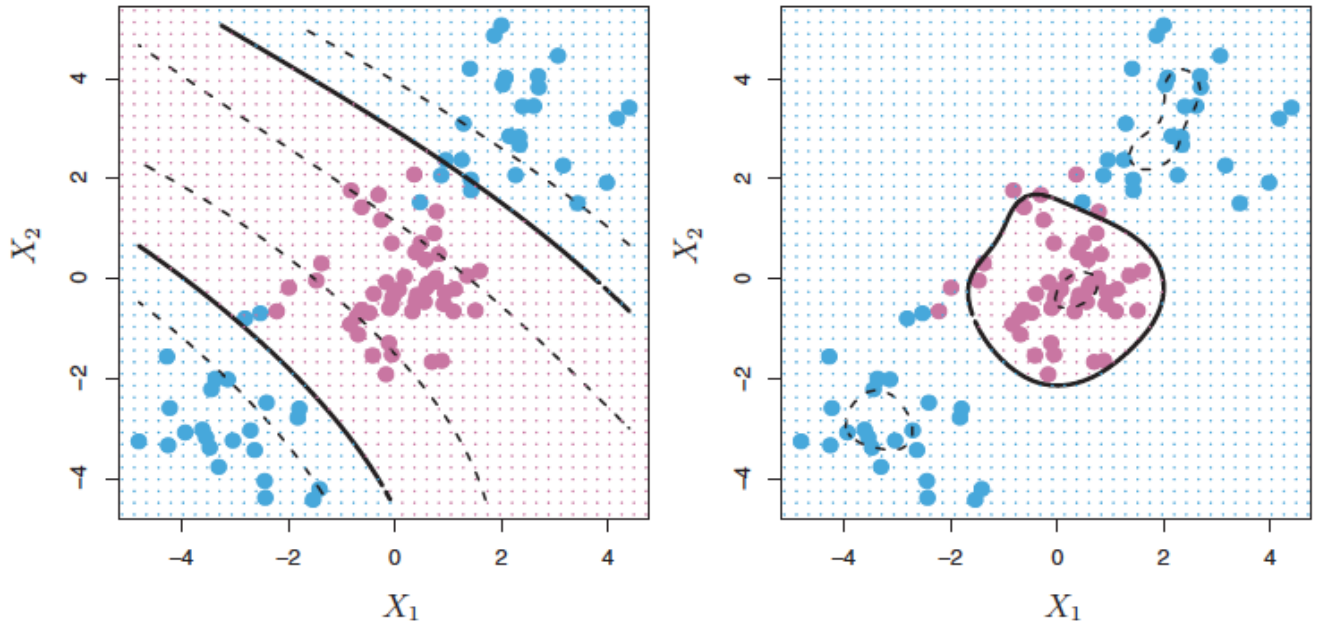


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

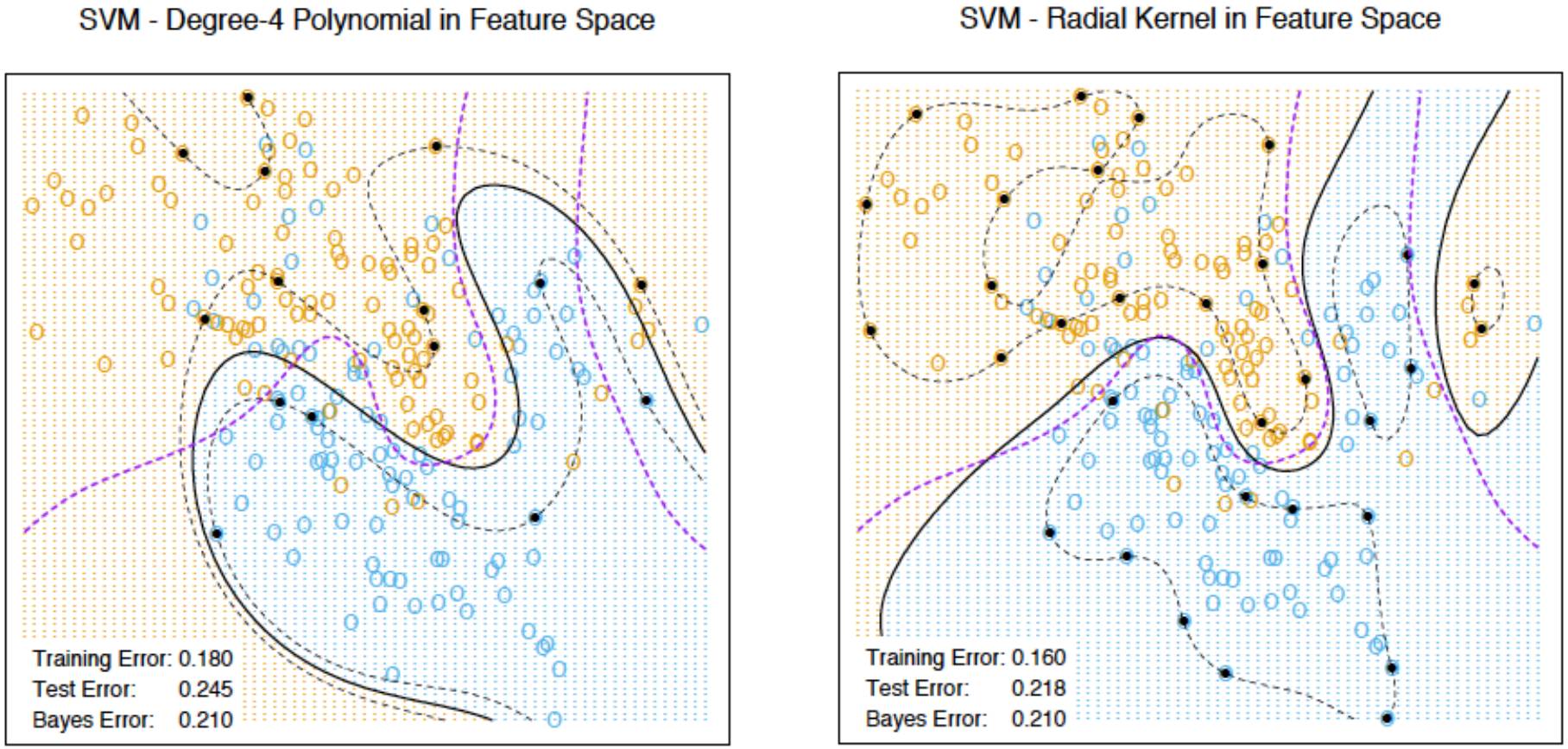


FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial kernel, the lower a radial basis kernel (with $\gamma = 1$). In each case C was tuned to approximately achieve the best test error performance, and $C = 1$ worked well in both cases. The radial basis kernel performs the best (close to Bayes optimal), as might be expected given the data arise from mixtures of Gaussians. The broken purple curve in the background is the Bayes decision boundary.*

Figure from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman, 2009.

Why kernels?

- What is the advantage of using a kernel function instead of simply enlarging the feature space using functions of the original features?
  - It is computational: Using kernels one only needs to compute $K(x,x_i)$ without explicitly working in the enlarged feature space
    - Sometimes the enlarged feature space is so large that computations are intractable!

Extending SVMs beyond binary classification

- One-versus-one classification
  - Constructs a SVM for each pair of classes
  - A test observation is classified by all the SVMs and it is assigned to the most frequent classification
- One-versus-all classification
  - Constructs *K* SVMs (K is the number of classes), each time using the instances of one class as positives and the instances of the remaining *K-1* classes as negatives
  - A test observation is assigned to the class for which its margin is largest

SVM

- SVM is a penalization method in the sense that the optimization problem has the form *loss* + *penalty*
- SVM gives zero penalty to points inside their margin, and a linear penalty to points on the wrong side and far away
- SVM's loss function belongs to the functions called *margin maximizing loss-functions*

SVM and noise

- SVM gives equal weight to all features, and the kernel cannot adapt itself to concentrate on feature subspaces
  - SVM cannot ignore noise features and its performance deteriorates in the presence of noise

TABLE 12.2. *Skin of the orange: Shown are mean (standard error of the mean) of the test error over 50 simulations. BRUTO fits an additive spline model adaptively, while MARS fits a low-order interaction model adaptively.*

	Method	Test Error (SE)	
		No Noise Features	Six Noise Features
1	SV Classifier	0.450 (0.003)	0.472 (0.003)
2	SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3	SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4	SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5	BRUTO	0.084 (0.003)	0.090 (0.003)
6	MARS	0.156 (0.004)	0.173 (0.005)
	Bayes	0.029	0.029

Table from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman, 2009.

Fitting a SVM

- SVM's performance is very sensitive to the choice of kernel
- The optimal value of *C* depends on the choice of the kernel

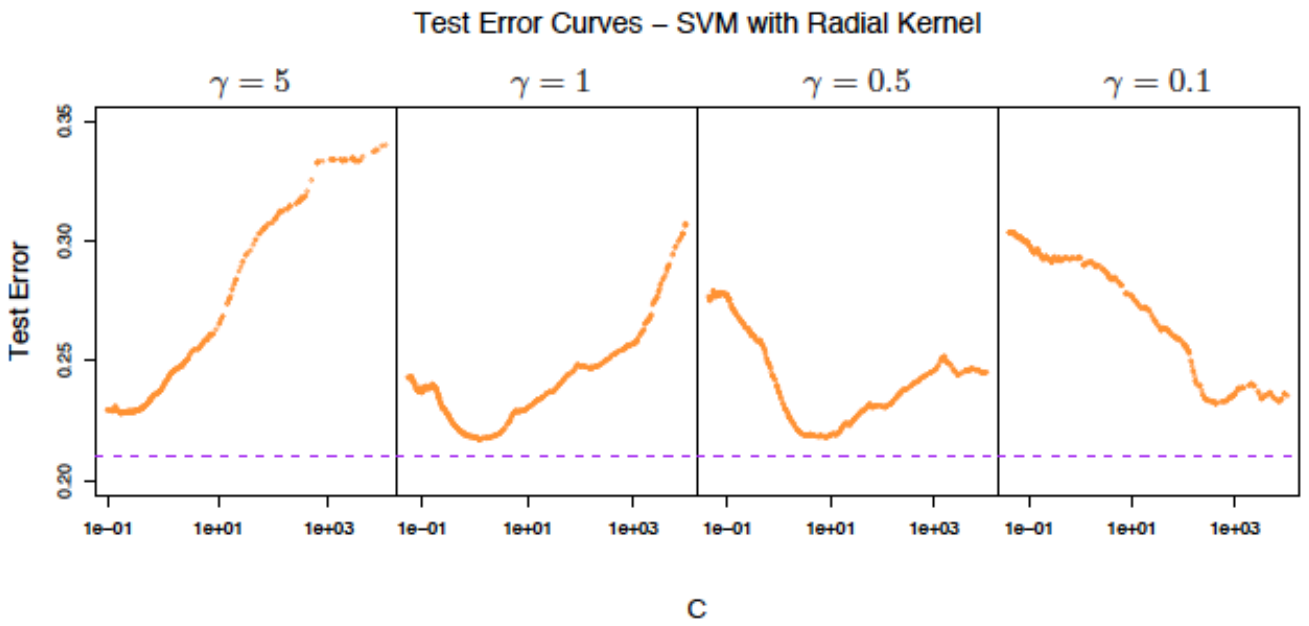


FIGURE 12.6. *Test-error curves as a function of the cost parameter C for the radial-kernel SVM classifier on the mixture data. At the top of each plot is the scale parameter γ for the radial kernel: $K_\gamma(x, y) = \exp(-\gamma||x - y||^2)$. The optimal value for C depends quite strongly on the scale of the kernel. The Bayes error rate is indicated by the broken horizontal lines.*

Fig. from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman, 2009.

SVM and Logistic Regression

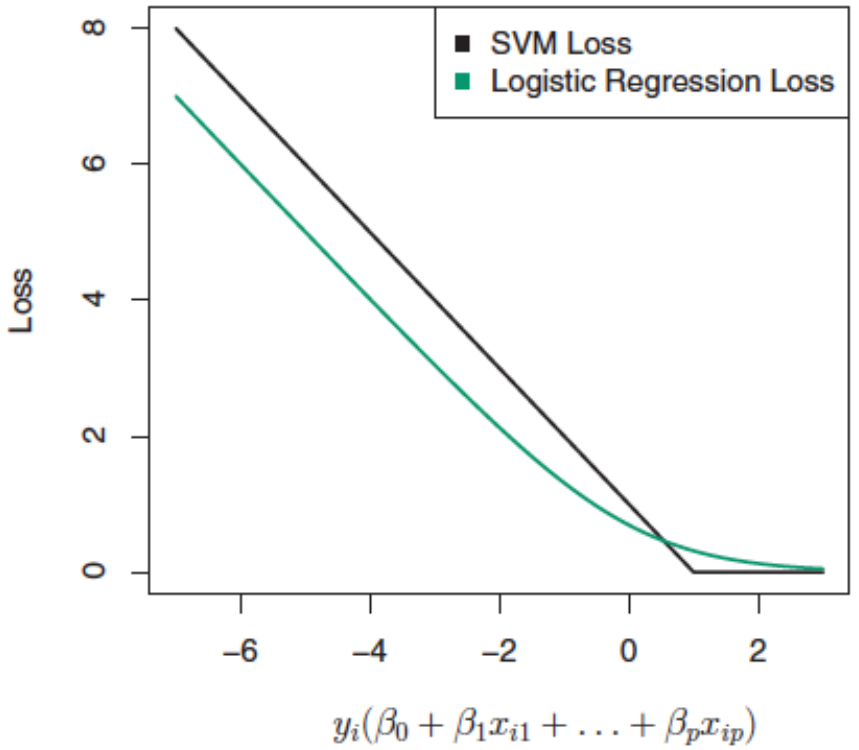


FIGURE 9.12. *The SVM and logistic regression loss functions are compared, as a function of* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$. *When* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$ *is greater than 1, then the SVM loss is zero, since this corresponds to an observation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

It has been shown that SVM loss function (hinge loss) is closely related to the loss function used in logistic regression.
Due to these similarities, logistic regression and SVM often give similar results.
When the classes are well separated, SVMs tend to perform better than logistic regression

Figure from *An Introduction to Statistical Learning: with Applications in R* by James, Witten, Hastie, and Tibshirani, 2013.

Discussion

- Other classification methods such as LDA can also use kernels to enlarge the feature space to accommodate non-linear class boundaries. For example,
    - Flexible discriminant analysis
- SVM has been extended for regression (see Section 12.3.6 of ESL II)

By now, you should be able to...

- define a hyperplane and know how one can calculate the position of a data point wrt a hyperplane
- explain the separating hyperplane classifier, maximal margin classifier and support vector classifier, and describe the differences between them
- understand the role of *C* and the slack variables
- explain how SVM extends the support vector classifier
- understand why kernels are used and know frequently-used kernels
- know how to apply SVM to multi-class problems
- be aware of the similarities between SVM and other ML methods
- be aware of limitations of SVM
- be aware that other methods can also use kernels and that SVM can also be used for regression