Support Vector Machines (Chapter 9)

Setup: Classification of a response Y based on p predictors $X_1, \ldots, X_p$ denoted by X

Data: (Y_i, X_i), $i = 1, \ldots, n$ from n independent subjects. The observations on features are stored in a $n \times p$ data matrix $\mathbf{X}$ with rows $X_1, \ldots, X_n$.

We will learn classification approaches based on the notion of a *separating hyperplane*. In particular:

- **Maximal margin classifier** — assumes that the classes can be perfectly separated by a linear boundary, limiting its applicability
- **Support vector classifier** — extension of the maximal margin classifier which does not make this assumption
- **Support vector machine** — extension of support vector classifier to accommodate non-linear class boundaries

These are **distinct** classifiers.

Hyperplane: In a p-dimensional space, a *hyperplane* is a flat affine subspace of dimension $p-1$. It is defined by the equation:

$$f(X)=0, \text{ where } f(X)=\beta_0+\beta_1X_1+\ldots+\beta_pX_p=\beta_0+X^T\beta. \tag{1}$$

Any X that satisfies (1) is a point on the hyperplane.

- $p=2$: a hyperplane is a line — $\beta_0+\beta_1X_1+\beta_2X_2=0$
- $p=3$: a hyperplane is a plane

If X does not satisfy (1), then either $f(X)>0$, so that X is on "one" side of the hyperplane, or $f(X)<0$, so that X is on the "other" side of the hyperplane. The side is determined by calculating the sign of $f(X)$. Thus, a hyperplane divides a p-dimensional space into two halves.

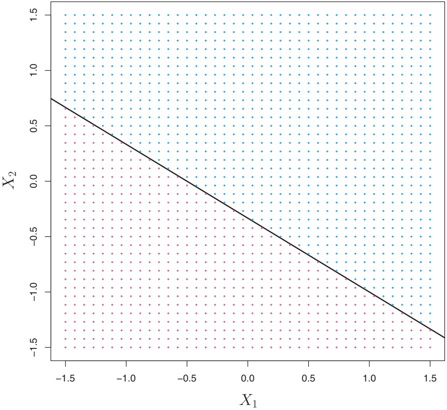


FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$*.*

Source: ISL

A Separating Hyperplane

Assume: Two classes, with labels $Y = -1$ or $Y = 1$

Separating hyperplane: A hyperplane that **perfectly** separates the data into two classes. This means, it has the following property: For **all** $i = 1, \ldots, n$

$$f(X_i) > 0 \text{ if } Y_i = 1 \quad \text{and} \quad f(X_i) < 0 \text{ if } Y_i = -1,$$

or equivalently,

$$Y_i f(X_i) > 0.$$

Note: A separating hyperplane does not exist if the classes are not *linearly separable*. However, if it exists, it provides a simple classifier: $\mathbf{sign}\{f(X)\}$. For a given observation X, evaluate $f(X)$ and look at its sign. If the sign is '+', assign it to class $Y = 1$, and if the sign is '−', assign it to class $Y = -1$.

- Magnitude of $f(X)$ indicates confidence in the assignment
- Linear decision boundary

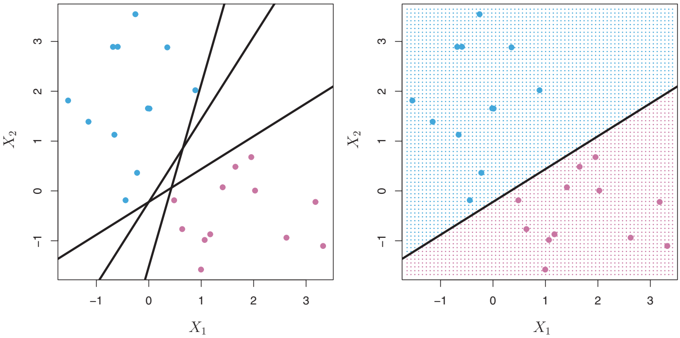


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision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

Source: ISL

Maximal Margin Classifier

Issue: If the classes are linearly separable, there will exist an infinite number of separating hyperplanes. **Which one to use?**

Margin of a separating hyperplane: Smallest perpendicular distance between the training observations and the hyperplane.

Maximal margin hyperplane (or **optimal separating hyperplane**): A separating hyperplane whose margin is largest.

- This hyperplane is *furthest* from the training observations.
- It separates the two classes and maximizes the distance to the closest points from either side
- It amounts to inserting the widest "slab" between the two classes. The maximal margin hyperplane represents the mid-line this slab, and its margin = (width of slab)/2.
- Leads to **maximal margin classifier**

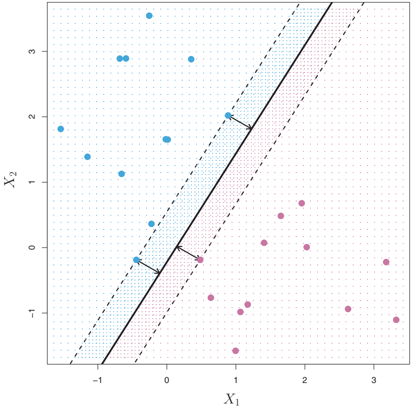


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support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

Source: ISL

To be specific, suppose M is the margin of the maximal margin hyperplane. The hyperplanes that lie at a distance M from it on either side form a slab with the optimal hyperplane representing its mid-line. This slab of width $2M$ is the widest slab that separates the two classes.

Support vectors: Points (i.e., the X observations) that "support" the maximal margin hyperplane in that if they move, the hyperplane would also move. They lie on the boundary of the slab associated with the hyperplane.

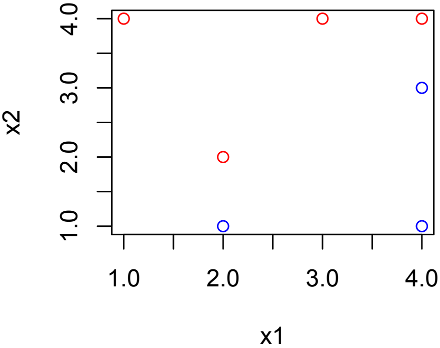
- They are at a distance M from the optimal hyperplane.
- The optimal hyperplane depends directly on the support vectors but not on the other observations provided their movement does not cause them to cross the slab boundary.

A Toy Example

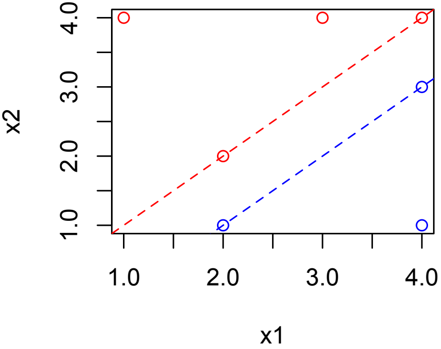
Let's obtain the maximal margin classifier for the following data:

Obs #	X_1	X_2	Y
1	3	4	Red
2	2	2	Red
3	4	4	Red
4	1	4	Red
5	2	1	Blue
6	4	3	Blue
7	4	1	Blue

Step 1: Plot the data

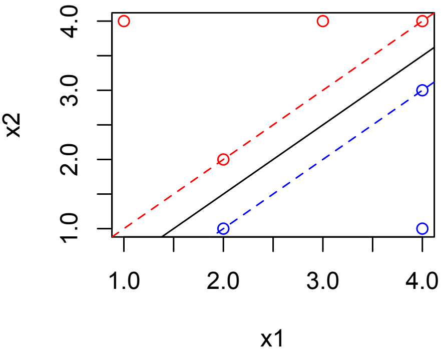


Step 2: Insert the widest slab separating the two classes



- Red line joins the points (2, 2) and (4, 4): $0 - X_1 + X_2 = 0$
- Blue line joins the points (2, 1) and (4, 3): $1 - X_1 + X_2 = 0$

Step 3: Draw the mid-line of the slab



- Maximal margin hyperplane: black line $(0.5 - X_1 + X_2 = 0)$

Q: What is the margin M here? **A:** Recall that the perpendicular distance between two parallel hyperplanes $\beta_0 + x^T\beta = 0$ and $\beta_0^* + x^T\beta = 0$ is $|\beta_0 - \beta_0^*|/\sqrt{\beta^T\beta}$. This gives $M = 1/(2\sqrt{2}) \approx 0.35$.

Q: How many support vectors do we have? Verify that they lie at distance of $1/(2\sqrt{2}) \approx 0.35$ from the optimal hyperplane.

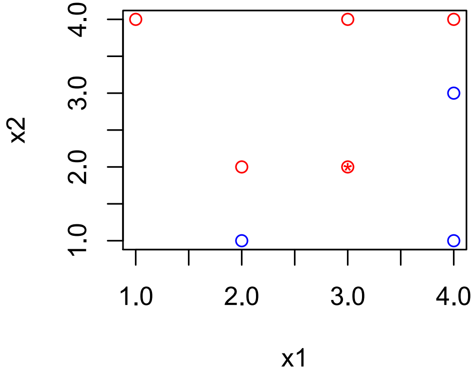
A: From the plot, we see that there are four support vectors. Recall that the perpendicular distance of a point X^* from the hyperplane $\beta_0 + X^T\beta = 0$ is given by

$$\frac{|\beta_0 + X^{*T}\beta|}{\sqrt{\beta^T\beta}}.$$

The distances of the points from the line $0.5 - X_1 + X_2 = 0$ are: (the support vectors are highlighted)

Obs #	X_1	X_2	Y	distance
1	3	4	Red	1.06
2	2	2	Red	0.35
3	4	4	Red	0.35
4	1	4	Red	2.47
5	2	1	Blue	0.35
6	4	3	Blue	0.35
7	4	1	Blue	1.77

Consider adding the observation (Y, X_1, X_2) =(Red, 3, 2) to the data.



Q: Does a separating hyperplane exist in this case? **A:** No — verify. The classes are not linearly separable.

Building the Maximal Margin Classifier

Issue: Assume that the classes are linearly separable. Set up the optimization problem whose solution gives the *optimal* separating hyperplane.

First, note that a hyperplane $f(X) = \beta_0 + X^T\beta = 0$ is characterized by (β_0, β). Let $||\beta|| = \sqrt{\beta^T\beta}$ be the norm of β. A point X_i is at a *perpendicular distance* $|f(X_i)|/||\beta||$ from this hyperplane. The smallest of such distances is its *margin*,

$$M = M(\beta_0, \beta) = \frac{1}{||\beta||} \min_{i=1,\ldots,n} |f(X_i)|.$$

Next, assume that it is a *separating hyperplane*. By definition, its margin $M > 0$ and it satisfies

$$Y_i f(X_i) > 0 \;\text{ for all } i = 1, \ldots, n,$$

allowing us to write

$$Y_i f(X_i) = |f(X_i)| \geq M||\beta|| \;\text{ for all } i = 1, \ldots, n.$$

It follows that the optimal hyperplane is the solution to the optimization problem:

$$\max_{\beta_0,\beta} M \text{ s.t. } Y_i f(X_i) \geq M||\beta|| \text{ for all } i = 1, \ldots, n.$$

- The constraints imply that each observation is on the correct side of the hyperplane and at least at a distance M from the hyperplane.
- They define an empty slab with boundaries given by the hyperplanes $Y f(X) = M||\beta||$.

Three equivalent formulations: For the first, note that the hyperplanes $f(X) = 0$ and $\{f(X)/||\beta||\} = 0$ are the same. So no loss of generality results if we restrict attention to hyperplanes with $||\beta|| = 1$. Thus, the optimization problem becomes

$$\max_{\beta_0,\beta} M \text{ s.t. } ||\beta|| = 1 \text{ and } Y_i f(X_i) \geq M \text{ for all } i = 1, \ldots, n.$$

The second is obtained by taking $M = \frac{1}{||\beta||}$ in the original formulation. The resulting problem can be written as:

$$\min_{\beta_0,\beta} \frac{1}{2}||\beta||^2 \text{ s.t. } Y_i f(X_i) \geq 1 \text{ for all } i = 1, \ldots, n,$$

leading to slab boundaries given by $Y f(X) = 1$ and margin $\frac{1}{||\beta||}$.

The third is its Lagrangian version with objective function

$$L(\beta_0, \beta, \alpha_1, \dots, \alpha_n) = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{n} \alpha_i [Y_i f(X_i) - 1],$$

where α_i (≥ 0) are the Lagrange multipliers. It requires solving:

$$\frac{\partial L}{\partial \beta_0} = 0 \implies \sum_{i=1}^{n} \alpha_i Y_i = 0,$$

$$\frac{\partial L}{\partial \beta} = 0 \implies \sum_{i=1}^{n} \alpha_i Y_i X_i = \beta,$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \implies \alpha_i [Y_i f(X_i) - 1] = 0, \quad i = 1, \dots, n.$$

From these we can see that:

- If X_i is not on the slab boundary, $Y_i f(X_i) > 1 \implies \alpha_i = 0$
- β is determined by only those X_i that fall on the slab boundary — **support vectors**
- To get β_0, solve $1 = Y_i f(X_i)$ for any support vector.

Properties of the MM Classifier

Maximal margin classifier: The maximal margin hyperplane is $\hat{f}(X) = 0$, where $\hat{f}(X) = \hat{\beta}_0 + X^T\hat{\beta}$. The margin of this hyperplane is $1/||\hat{\beta}||$. The associated classifier is $\text{sign}\{\hat{f}(X)\}$. The slab boundaries are given by $\hat{Y}\hat{f}(X) = 1$. One needs at least two support vectors.

- The observations X_i appear in the optimization problem and in its solution only through their **inner products**.
- Depends on a small number of support vectors
- Assumes that the classes can be perfectly separated by a linear boundary. If this assumption does not hold, the classifier does not exist
- Perfectly classifies the training data — **overfitting**
- Unstable — a small change in the data may lead to a very different decision boundary
- Give up perfect linear separation and tolerate a small number of misclassifications to build a better classifier — **support vector classifier**

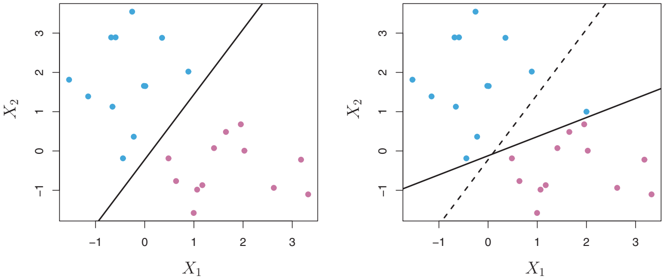


FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin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

Source: ISL

Support Vector Classifier

Suppose now that the classes overlap in the feature space. We still classify based on the sign of $f(X) = \beta_0 + X^T\beta$ and still maximize the margin M of the hyperplane $f(X) = 0$, but allow some points to be on the wrong side of the margin as well as the hyperplane. Let $\epsilon_1, \dots, \epsilon_n$ denote **slack variables**. We modify the constraint $Y_i f(X_i) \geq M$ for all $i = 1, \dots, n$ as

$$Y_i f(X_i) \geq M(1 - \epsilon_i), \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C.$$

- “boundary” and “margin” are used interchangeably
- $\epsilon_i = 0$: correct side of the margin
- $\epsilon_i > 0$: wrong side of the margin (violates the margin)
- $\epsilon_i > 1$: wrong side of the hyperplane (misclassification)
- Think of C as a **budget** for margin violation. If $C = 0$, all $\epsilon_i = 0$. If $C > 0$, # misclassifications $\leq C$.
- Large C = more tolerant of violations.

The **support vector classifier** is obtained by solving the same optimization problem as in the maximal margin classifier but with the new constraints. As before, upon taking $M = 1/||\beta||$, we get the following equivalent problem:

$$\min_{\beta_0, \beta} \frac{1}{2}||\beta||^2$$

subject to

$$Y_i f(X_i) \geq 1 - \epsilon_i,\ \epsilon_i \geq 0,\ i = 1, \ldots, n,\ \sum_{i=1}^{n} \epsilon_i \leq C.$$

- **Support vectors**: The points that lie either on the margin or violate the margin. Only these affect the classifier.
- The observations that lie on the correct side of the margin do not affect the classifier.
- Slab (margin) boundaries: $Yf(X) = 1$; margin $= 1/||\beta||$.

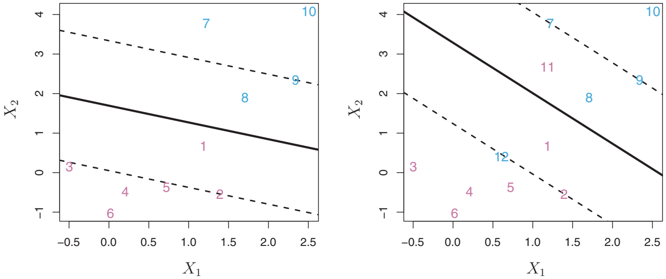


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* 3, 4, 5, *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation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

Source: ISL

How to Choose C?

C is a *tuning* parameter that controls the bias-variance tradeoff.

- Larger C = more tolerant of margin violations = wider margin = more support vectors = potentially lower variance (because the classifier is determined only by the support vectors which are larger in number) but higher bias (because the training error is larger due to the larger number of margin violations)
- C can be chosen in the usual manner using a cross-validation or a validation-set approach.

Note: Support vector classifier has a linear decision boundary because of which it may not work well when a non-linear decision boundary is called for.

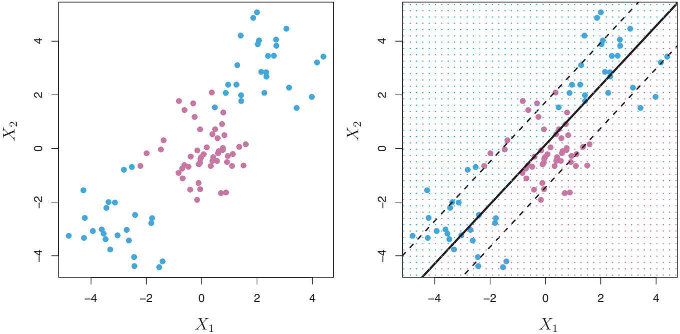


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Source: ISL

Classification with Non-Linear Boundaries

Option 1: Fit a support vector classifier but instead of using just the original features $X_1, \ldots, X_p$, we enlarge the feature space using functions of predictors, e.g., polynomials in X_j and interactions of the form $X_j * X_l$, $j \neq l$, and use all the features in the enlarged space as predictors.

- Decision boundary is linear in the enlarged space but non-linear in the original space
- There are many ways to enlarge the feature space. Unless done carefully, we may end up with a prohibitively large number of features, making the computations problematic.

Option 2: Use a **support vector machine** which also involves fitting a support vector classifier in an enlarged space but done so in a computationally efficient manner.

Support Vector Machine (SVM)

Inner product of two feature vectors X_i and X_l is:

$$\langle X_i, X_l \rangle = X_i^T X_l = \sum_{j=1}^{p} X_{ij} X_{lj}.$$

Note: It can be seen that in an SV classifier, the X_i play a role in the optimization problem and the solution $\hat{f}(X) = \hat{\beta}_0 + X^T \hat{\beta}$ only through the inner products involving them. In particular, we have $\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i Y_i X_i$, where the coefficient $\hat{\alpha}_i$ is positive for a support vector and is zero otherwise. Thus, we can write

$$\hat{f}(X) = \hat{\beta}_0 + X^T \left(\sum_{i=1}^{n} \hat{\alpha}_i Y_i X_i \right) = \beta_0 + \sum_{i=1}^{n} \hat{\alpha}_i Y_i \langle X, X_i \rangle.$$

To generalize, we can replace the inner product $\langle X_i, X_l \rangle$ by a *kernel function* $K(X_i, X_l)$, leading to the estimated function as

$$\hat{f}(X) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i Y_i K(X, X_i).$$

This generalization is called an **SVM**. It fits an SV classifier in the transformed feature space. By definition, K is a symmetric, positive (semi)-definite function. We can think of the kernel K as a *measure of dissimilarity*. Two popular choices for K in the SVM literature are:

d**th degree polynomial**: $K(X, X') = (1 + \langle X, X' \rangle)^d$

Radial basis: $K(X, X') = \exp(-\gamma ||X - X'||^2)$, $\gamma > 0$.

Taking $K(X, X') = \langle X, X' \rangle$ (or equivalently $d = 1$ in the polynomial kernel) gives the SV classifier. This is a *linear* kernel whereas the other kernels are non-linear.

SVMs with $Q\,(> 2)$ Classes

So far our focus was on **binary classification**. The concept of separating hyperplanes does not extend naturally to more than two classes. Two common approaches to deal with this:

One-versus-one classification: There are $\binom{Q}{2}$ pairs of classes. Fit an SVM for each pair. Then, classify a test point X as follows: Obtain its $\binom{Q}{2}$ classifications, one from each fit, and assign it to the most frequent class.

One-versus-all classification: Fit Q SVMs, each time comparing one class (coded as $+1$) with the remaining $Q-1$ classes (coded as -1). For a test point X, compute $\hat{f}(X)$ from each fit. Assign it to the class for which $\hat{f}(X)$ is largest.

SV Classifier vs Logistic Regression

SV classifier: sign$\{f(X)\}$, where $f(X) = \beta_0 + X^T\beta$. The margin (width) is $1/||\beta||$ and its boundaries are $Yf(X) = 1$. An observation does not violate the margin if $Yf(X) \geq 1$. The optimization problem for fitting it can be rewritten as

$$\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} \max[0, 1 - Y_i f(X_i)] + \lambda ||\beta||^2 \right\},$$

where $\lambda\,(> 0)$ is a tuning parameter. This is the familiar "loss (or error) + penalty" formulation that leads to regularized estimation. The penalty here is same as the ridge penalty.

Loss function, $\max[0, 1 - Yf(X)]$: It equals zero if $Yf(X) \geq 1$,i.e., the observation does not violate the margin, and equals $1 - Yf(X)$ if $Yf(X) < 1$, i.e., the observation violates the margin — **hinge loss function**. It tends to be similar to the loss function $(-\log L)$ used in logistic regression.

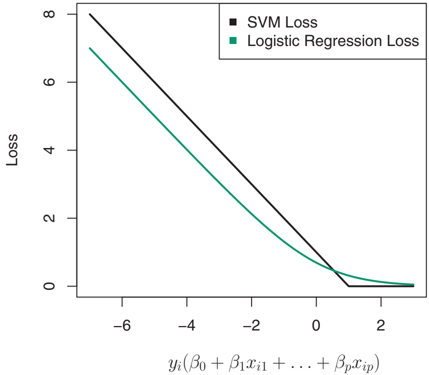


FIGURE 9.12. *The SVM and logistic regression loss functions are compared, as a function of $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$. When $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$ is greater than 1, then the SVM loss is zero, since this corresponds to an observation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

Source: ISL

- SV classifier gives similar results as a regularized logistic regression — linear decision boundary
- SVMs with non-linear kernels lead to non-linear decision boundaries
- Can logistic regression lead to non-linear decision boundaries? — Yes (will see in Chapter 7).
- **SV regression machine**: Adapt an SVM by using an appropriate loss function, e.g.,

$$V(r) = \begin{cases} 0, & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise,} \end{cases}$$

  where r is a residual. This is called ϵ-insensitive loss function — it ignores errors of size less than ϵ. This is analogous to the SV set up where points on the correct side of the margin do not play any role in model fitting.

Support Vector Machines

Python coding exercises

- Support Vector Classifier
- Support Vector Machine
- ROC Curves & AUC
- SVM with Multiple classes
- Application: Gene Expression Data