Support Vector Machines

Chapter 9

November 15, 2016

1. 9.1. Maximal margin classifier
2. 9.2. Support vector classifiers
3. 9.3. Support vector machines
4. 9.4. SVMs with more than two classes
5. 9.5. Relationshiop to logistic regression

About this chapter

- Support vector machine is one of the most popular machine learning methodologies.
- Empirically successful, with well developed theory.
- One of the best off-the-shelf methods.
- We mainly address classification.

Hyperplane in R^p

- $\mathbf{x} \in R^p$ (p-dimensional real space) with components $\mathbf{x} = (x_1, ..., x_p)^T$.
- Consider all $\mathbf{x}$ satisfying

$$f(\mathbf{x}) = \mathbf{b}^T\mathbf{x} - c =< \mathbf{b}, \mathbf{x} > -c = b_1x_1 + ... + b_px_p - c = 0.$$

All such $\mathbf{x}$ defines a hyperplane: All $\mathbf{x}$ such that its projection on $\mathbf{b}$ is $c\mathbf{b}/\|\mathbf{b}\|^2$.

Hyperplane in R^p

- $$f(\mathbf{x}) = 0 \iff \mathbf{x} = c\mathbf{b}/\|b\|_2 + y \qquad \text{with } y \perp \mathbf{b}.$$
- $$f(\mathbf{x}) > 0 \iff \mathbf{x} = \tilde{c}\mathbf{b}/\|b\|^2 + y \qquad \text{with } y \perp \mathbf{b},\ \tilde{c} > c$$
- $$f(\mathbf{x}) < 0 \iff \mathbf{x} = \tilde{c}\mathbf{b}/\|b\|^2 + y \qquad \text{with } y \perp \mathbf{b},\ \tilde{c} < c$$

here $\tilde{c} =< \mathbf{x}, \mathbf{b} >$.

- $f(\mathbf{x}) > 0 \iff \mathbf{x}$ is on one side of the hyperplane (at the same direction as $\mathbf{b}$.)
  $f(\mathbf{x}) < 0 \iff \mathbf{x}$ is on the other side of the hyperplane (at the opposite direction as $\mathbf{b}$.)
- For any vector $\mathbf{z} \in R^p$, the signed distance of a point $\mathbf{z} \in R^p$ to this hyperplane is

$$f(\mathbf{z})/\|\mathbf{b}\| = (< \mathbf{z}, \mathbf{b} > -c)/\|\mathbf{b}\| =< \mathbf{z}, \mathbf{b}/\|\mathbf{b}\| > -c/\|\mathbf{b}\|.$$

- If $\|\mathbf{b}\| = 1$, $f(\mathbf{z})$ is the signed distance of $\mathbf{z}$ to the hyperplane defined by $f(\mathbf{x}) = 0$.

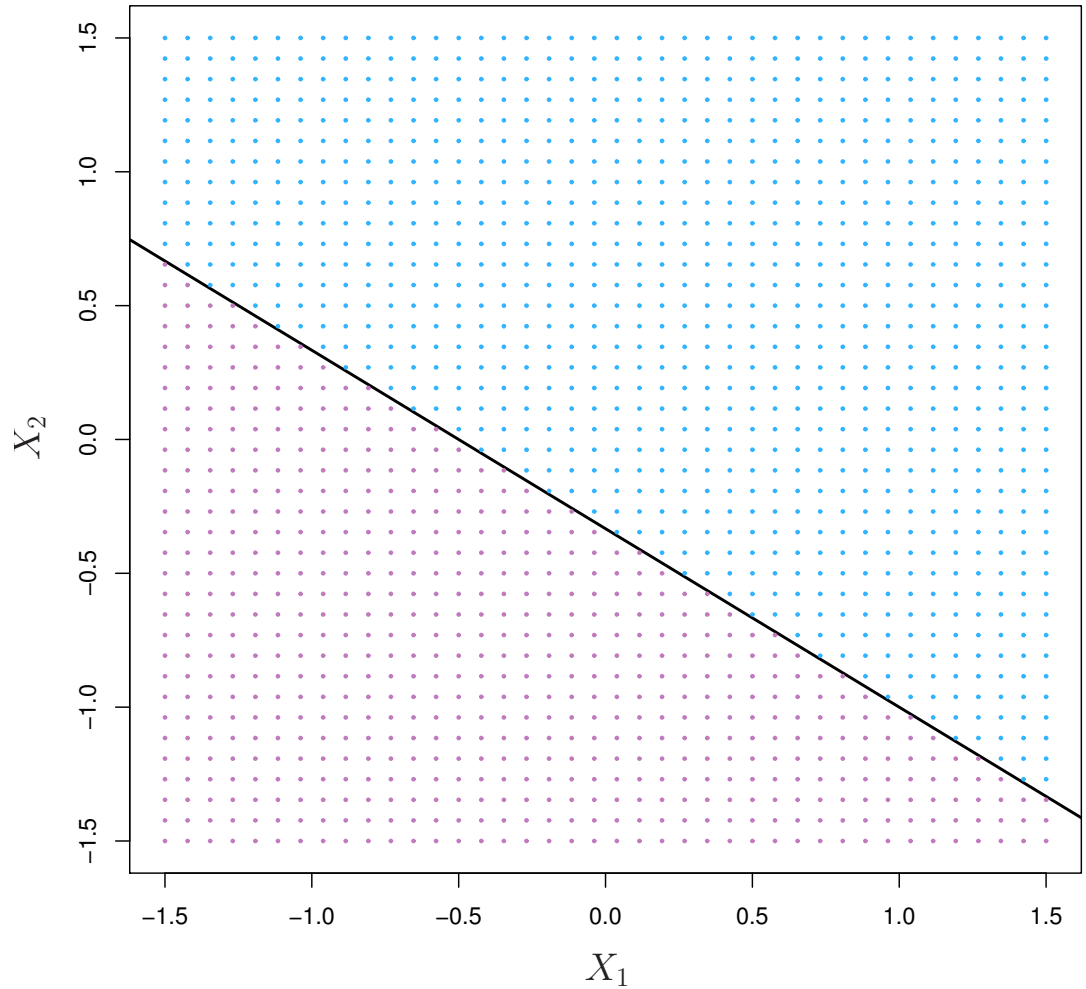

Figure: 9.1. The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

Separating hyperplane

- Training Data:$(y_i, \mathbf{x}_i), i = 1, ..., n$, with input $\mathbf{x}_i = (x_{i1}, ..., x_{ip})$ and two-class output $y_i = \pm 1$.
- Suppose the two classes are seperated by one hyperplane

$$f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p,$$

meaning that, for all i in one class $y_i = 1$,

$$f(\mathbf{x}_i) > 0, \quad \text{one side of the hyperplane;}$$

and for all i in the other class $y_i = -1$,

$$f(\mathbf{x}_i) < 0, \quad \text{the other side of the hyperplane;}$$

- It can be equivalently expressed as

$$y_i f(\mathbf{x}_i) > 0, \quad \text{for all } i = 1, .., n$$

- If such separating hyperplane exists, it can be our classification rule:
  For any new/old observation with $\mathbf{x}^*$ such that $f(\mathbf{x}^*) > 0$, classify it as in the class $+1$. Otherwise, classify it as in class -1.
- Problem: If the two classes in training data are indeed separable by a hyperplane, which hyperplane is the best?

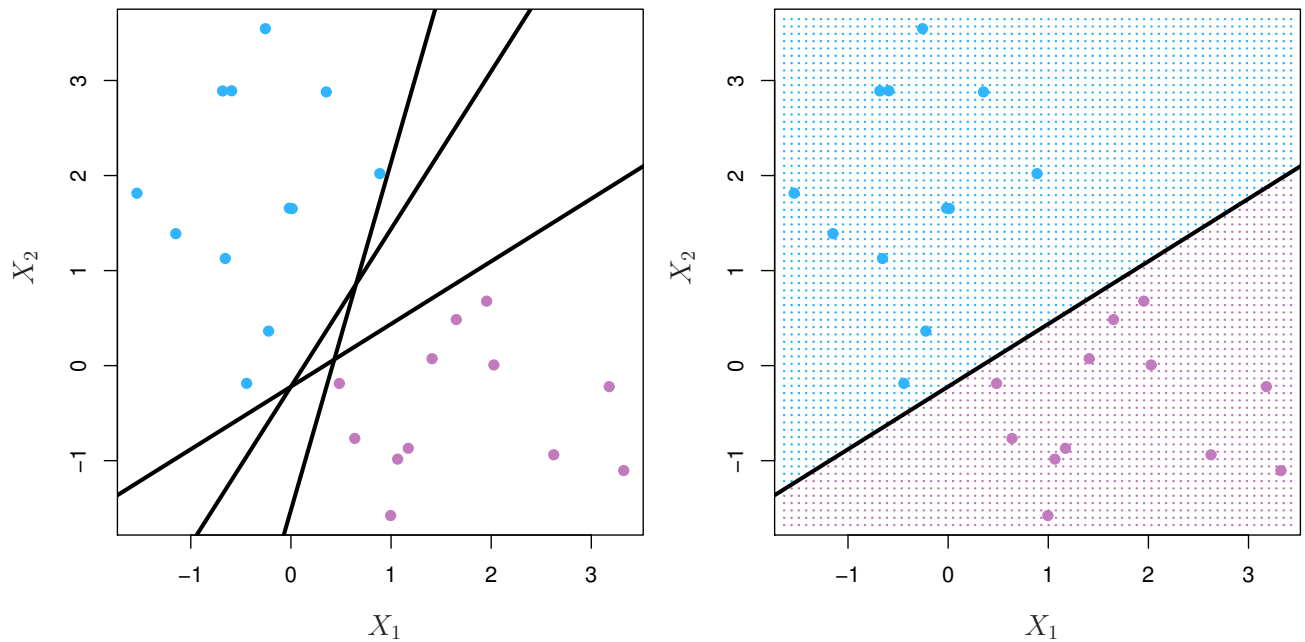

Figure: 9.2. Left: There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision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

Maximal margin classifier

- Maximal margin hyperplane: the separating hyperplane that optimal separating hyperplane is farthest from the training observations.
- The seperating hyperplane such that the minimum distance of any training point to the hyperplane is the largest.
- Creates a widest gap between the two classes.
- Points on the boundary hyperplane, those with smallest distance to the max margin hyperplane, are called *support vectors*. They "support" the maximal margin hyperplane in the sense vector that if these points were moved slightly then the maximal margin hyperplane would move as well

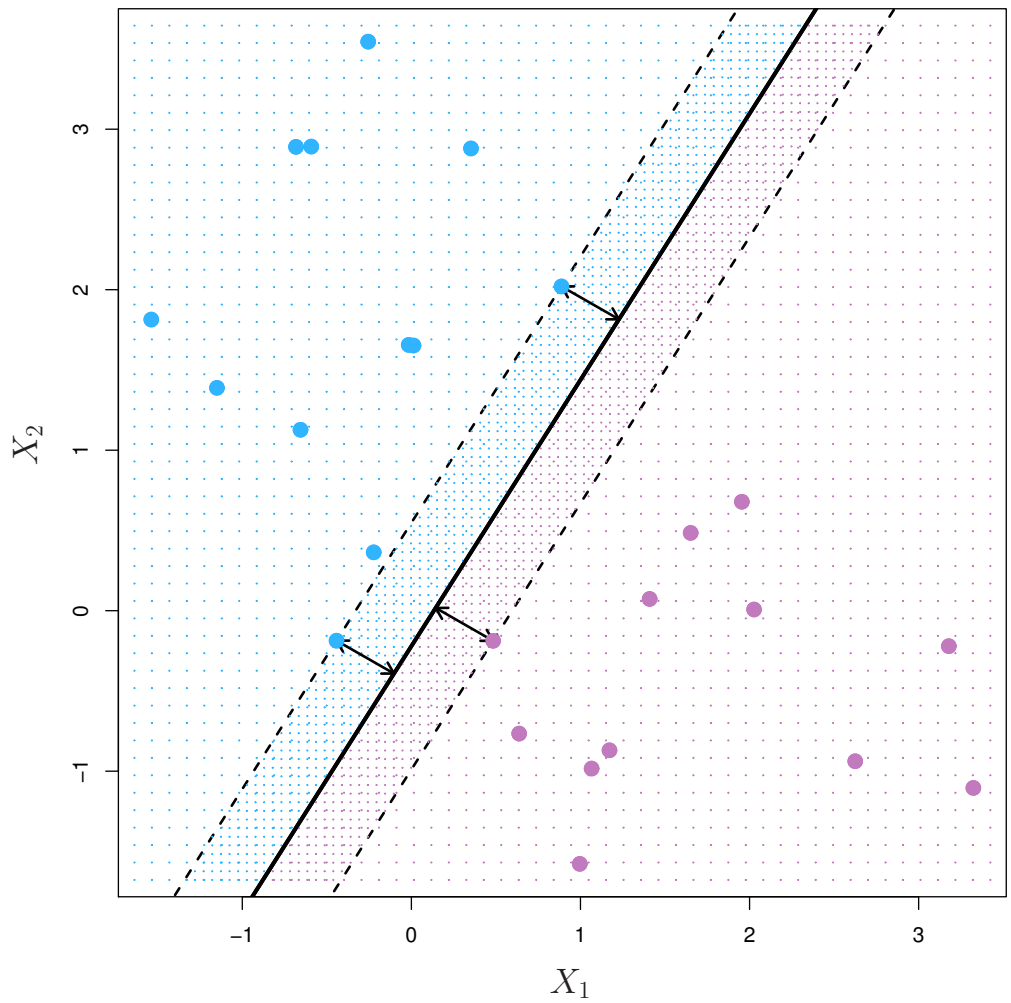

Figure: 9.3. There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support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

Computing the max margin hyperplane

$$\text{maximize}_{\beta_0,\beta_1,\ldots,\beta_p} M$$

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

$$\text{and } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M \text{ for all } i$$

- Note that $M > 0$ is the half of the width of the strip seperating the two classes.
- The eventual solution, the max margin hyperhplane is determined by the support vectors.
  If x_i on the correct side of the trip varies, the solution would remain same.
- The max margin hyperplane may vary a lot when the support vectors vary. (high variance; see Figure 9.5)

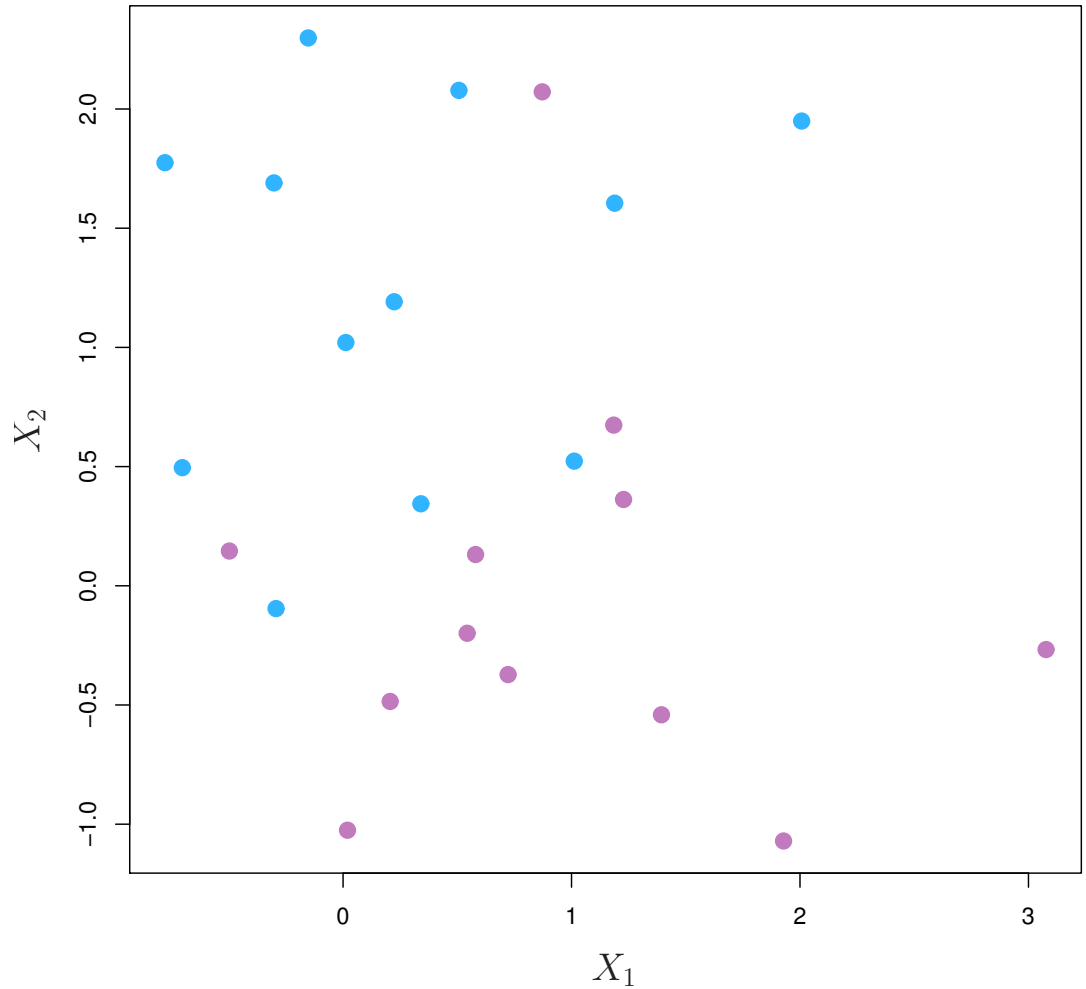

Figure: 9.4. There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.

The non-seperable case

- In general, the two classes are usually not separable by any hyperplane.
- Even if they are, the max margin may not be desirable because of its high variance, and thus possible over-fit.
- The generalization of the maximal margin classifier to the non-separable case is known as the *support vector classifier.*
- Use a soft-margin in place of the max margin.

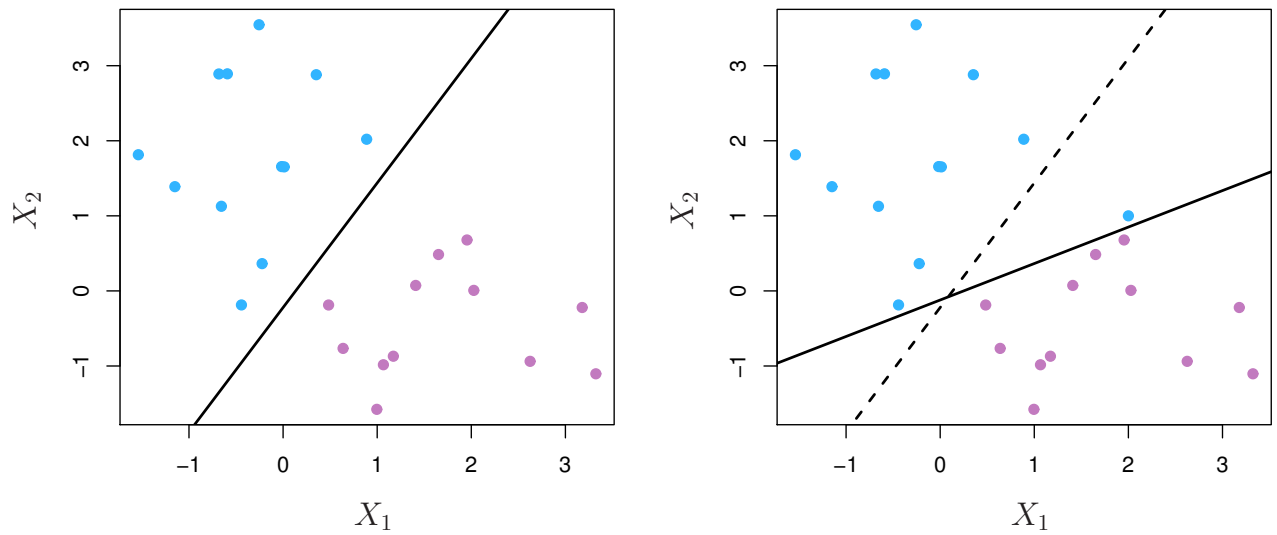

Figure: 9.5. Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane. Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin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

Non-perfect separation

- Consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of
  1. Greater robustness to individual observations, and
  2. Better classification of most of the training observations.
- Soft-margin classifer (support vector classifier) allow some violation of the margin: some can be on the wrong side of the margin (in the river) or even wrong side of the hyperplane; see Figure 9.6.

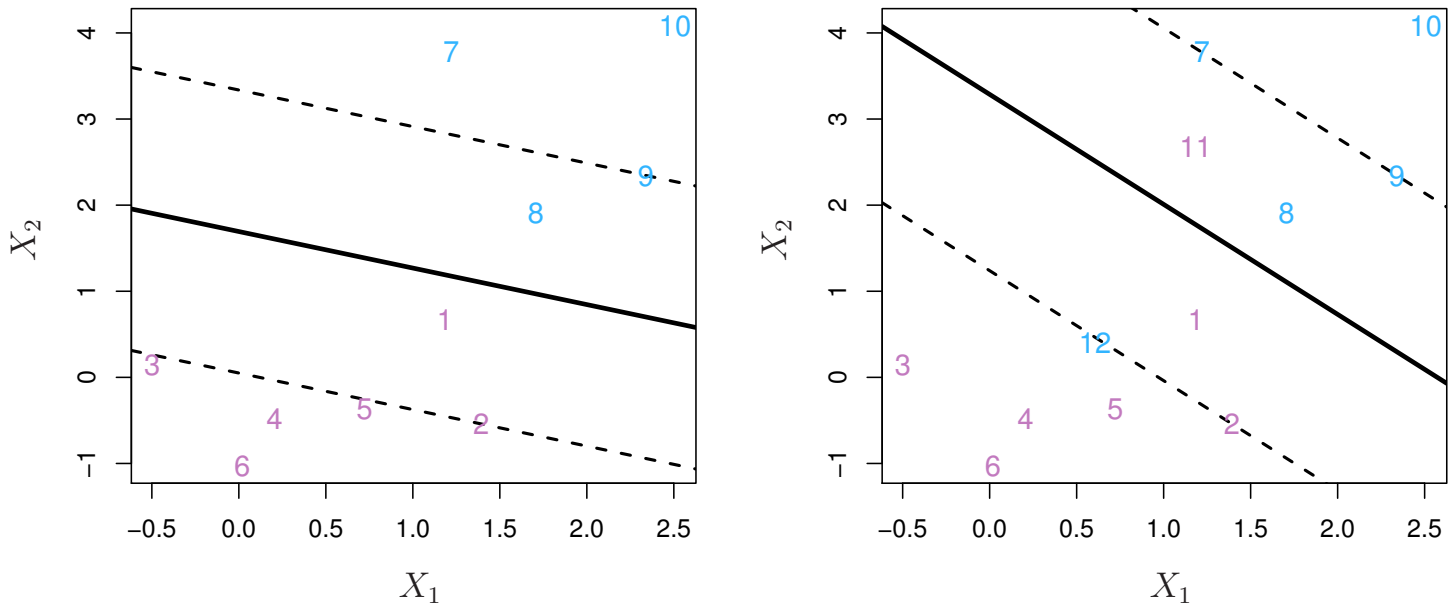

Figure: 9.6. next page

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.

Computing the support vector classifier

$$\text{maximize}_{\beta_0,\beta_1,\ldots,\beta_p} M$$

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

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \;\; \epsilon_i \geq 0 \text{ for all } i$$

$$\text{and} \sum_{i=1}^{n} \epsilon_i \leq C,$$

where C is a nonnegative tuning parameter. ϵ_i are *slack variables*.

The support vector classifier

The solution of this optimization is the support vector classsifier:

$$f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p.$$

And we classify an observation in $+1$ class if $f(\mathbf{x}) > 0$; else into -1 class.

Understaning the slack variable ϵ_i

- $\epsilon_i = 0 \iff$ the i-th observation is on the correct side of the margin
- $\epsilon_i > 0 \iff$ the i-th observation is on the wrong side of the margin
- $\epsilon_i > 1 \iff$ the i-th observation is on the wrong side of the hyperplane.

Understaning tuning parameter C

- C is a *budget* for the amount that the margin can be violated by the n observations
- $C = 0 \iff$ no budget
  As a result $\epsilon_i = 0$ for all i.
  The classifier is a maximal margin classifier, which exists only if the two classes are separable by hyperplanes.
- Larger C, more tolerance of margin violation.
- No more than C observations can be on the wrong side of the soft-margin classifier hyperplane.
- As C increases, the margin widens and more violations of the margin.

Understaning tuning parameter C

- C controls the bias-variance trade-off.
- Small C high variance, small bias.
- Large C: small variance, high bias.
- C can be determined by using cross validation.

Support vectors

- Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as support vectors.
- Only the support vectors affect the support vector classifier.
- Those strictly on the correct side of the margin do not. (robustness, analgous to median)
- Larger $C \implies$ more violations, $\implies$ more support vectors, $\implies$ smaller variance and more robust classfier.

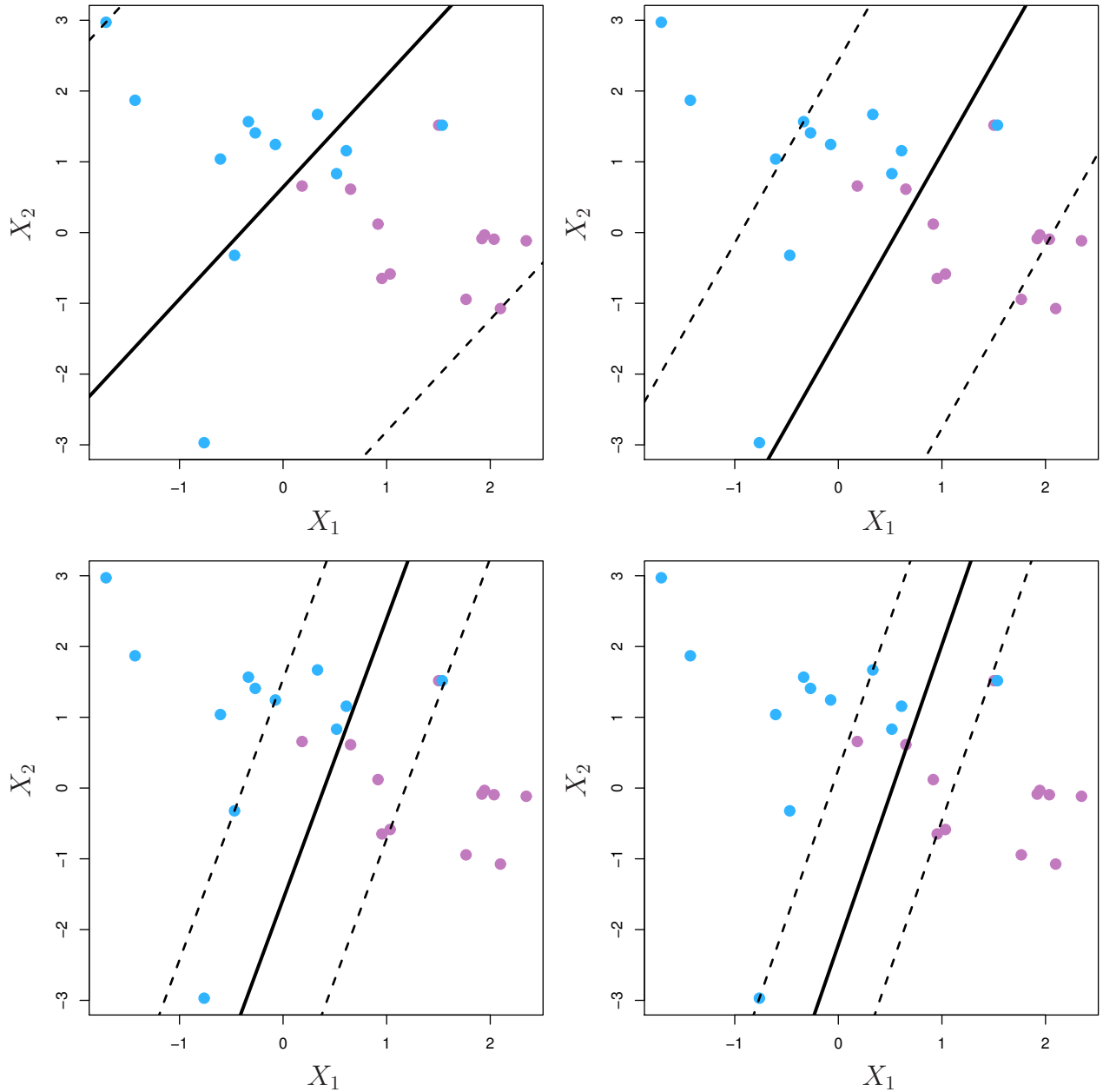

Figure 9.7.

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.

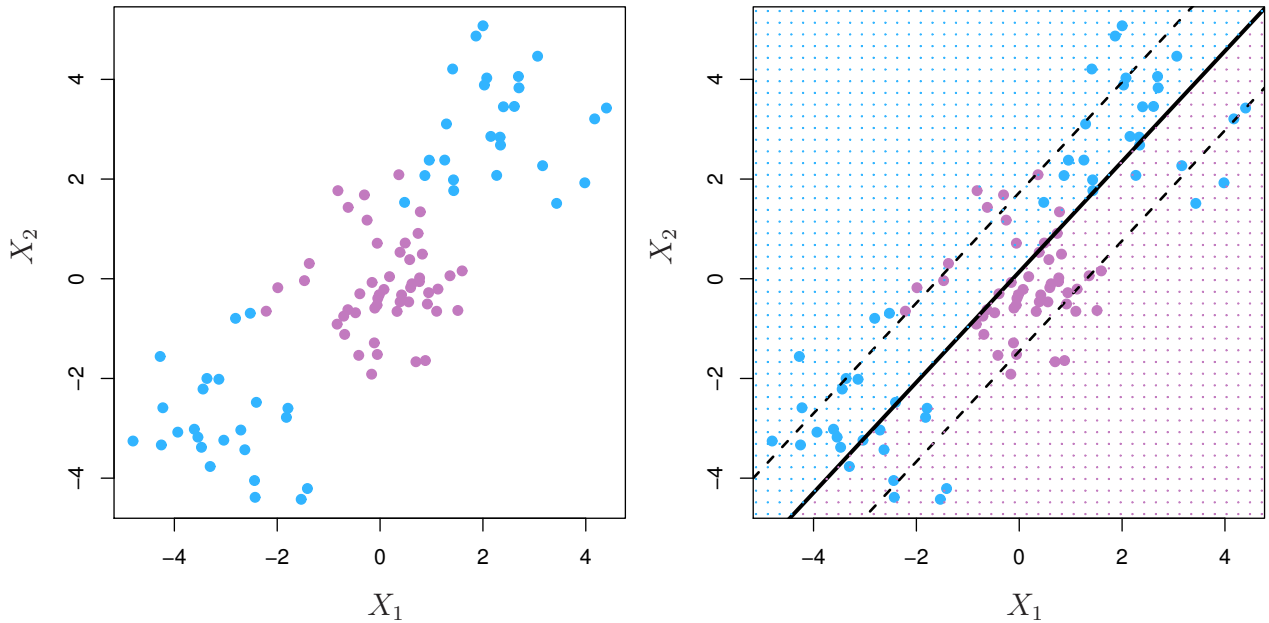

Figure: 9.8. Nonlinear boundaries. Left: The observations fall into two classes, with a non-linear boundary between them. Right: The support vector classifier seeks a linear boundary, and consequently performs very poorly.

Extending to nonlinear boundary

- In practice, we are sometimes faced with non-linear class boundaries
- Linear classifier could perform poorly.
- Need nonlinera classifier.
- As in the extension to the polynomial regression from linear regression, we can consider *enlarge the feature space* from the original p inputs to polynomials (of certain order) of the inputs.

Extending to quadratic inputs

- Rather than constructing the support vector classifier using p features:

$$X_1, ..., X_p.$$

- we use $2p$ features:

$$X_1, X_1^2, ..., X_p, X_p^2.$$

- Treat them as $2p$ original inputs, and fit the support vector classifier.
- The separating hyperplane is a hyperplane in R^{2p}, which should be a linear equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_p + \beta_{p+1} X_1^2 + ... + \beta_{2p} X_p^2 = 0$$

- This is a quadrtic equation in $X_1, ..., X_p$. Thus the separating surface in R^p in terms of $X_1, ..., X_p$ corresponds to a quadratic surface in R^p.

Extending to polynomial inputs

- Can extend to polynomial of any given order d.
- Could lead to too many features, too large feature space; thus overfit.
- Higher powers are unstable.

Key observation

- The linear support vector classifier can be represented as

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

- $\alpha_i \neq 0$ only for all support vectors.
- α_i can also be computed based on $< \mathbf{x}_j, \mathbf{x}_k >$.
- Only the inner product of the feature space is relevant in computing the linaer support vector classfier.

The kernel trick

- The inner product $< \cdot, \cdot >$ is a bivariate function (satisfying some property).
- It can be genearlized to kernel functions

$$K(\mathbf{x}, \mathbf{z})$$

  which is positive definite.
- The classifier can be expressed as

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} a_i K(\mathbf{x}, \mathbf{x}_i)$$

Examples of kernels

- Examples of the kernel function are:
- linear kernel

$$K(x_i, x_j) =< x_i, x_j >= x_i^T x_j.$$

- polynomial kernel of degree d:

$$K(x_i, x_j) = (1+ < x_i, x_j >)^d.$$

- Gaussian radial kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_j - x_j\|^2), \quad \gamma > 0.$$

- Only the inner product of the feature space is relevant in computing the linaer support vector classfier.

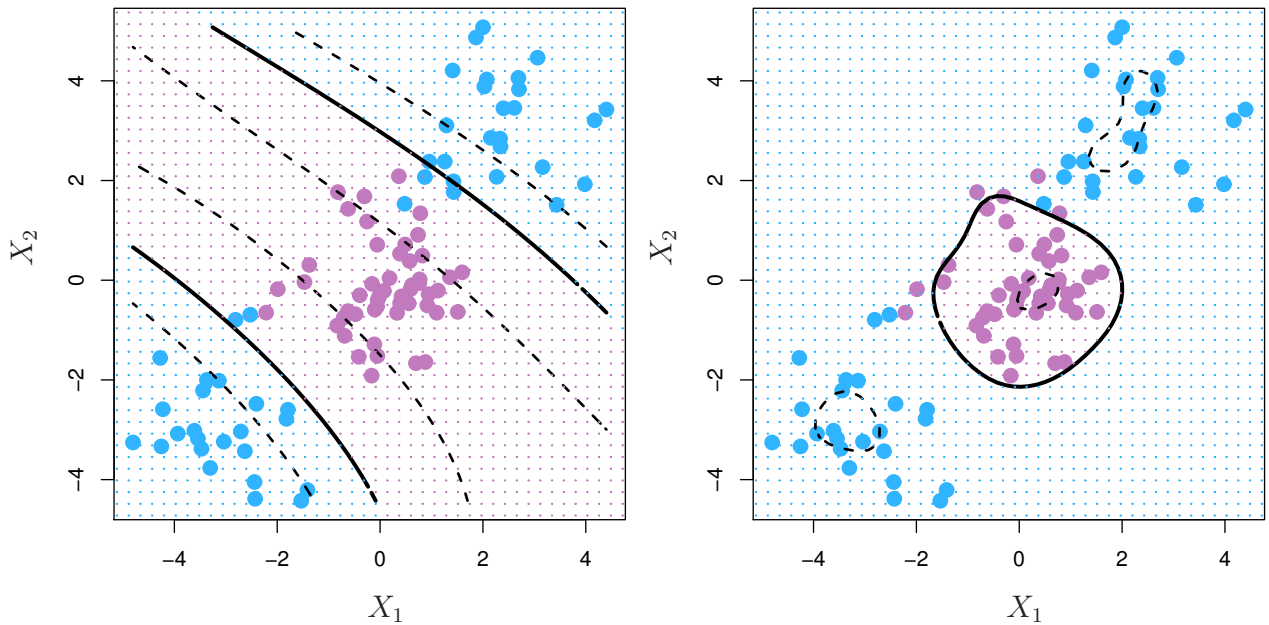

Figure: 9.9. Left: An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule. Right: An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary. In

- The radial kernel has local behavior.
- To predict the class for a new observation with input $\mathbf{x}$,

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} a_i \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$$

- A training data point $\mathbf{x}_i$ being far away from $\mathbf{x}$ will have little effect to the sign of $f(\mathbf{x})$.

The enlarged feature space.

- The support vector machine actually enlarges the original feature space to a space of kernel functions.

$$\mathbf{x_i} \rightarrow K(\cdot, \mathbf{x}_i).$$

- The original space of p inputs has dimension p.
- The enlarged space of features, the function space, is infinite dimension!
- In actual fitting of the support vector machine, we only need to compute the $K(x_i, x_j)$ for all x_i, x_j in training data.
- Do not have to work with the enlarged feature space of infinite dimension.

Example: the Heart 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.
- 297 subjects, randomly split into 207 training and 90 test observations.

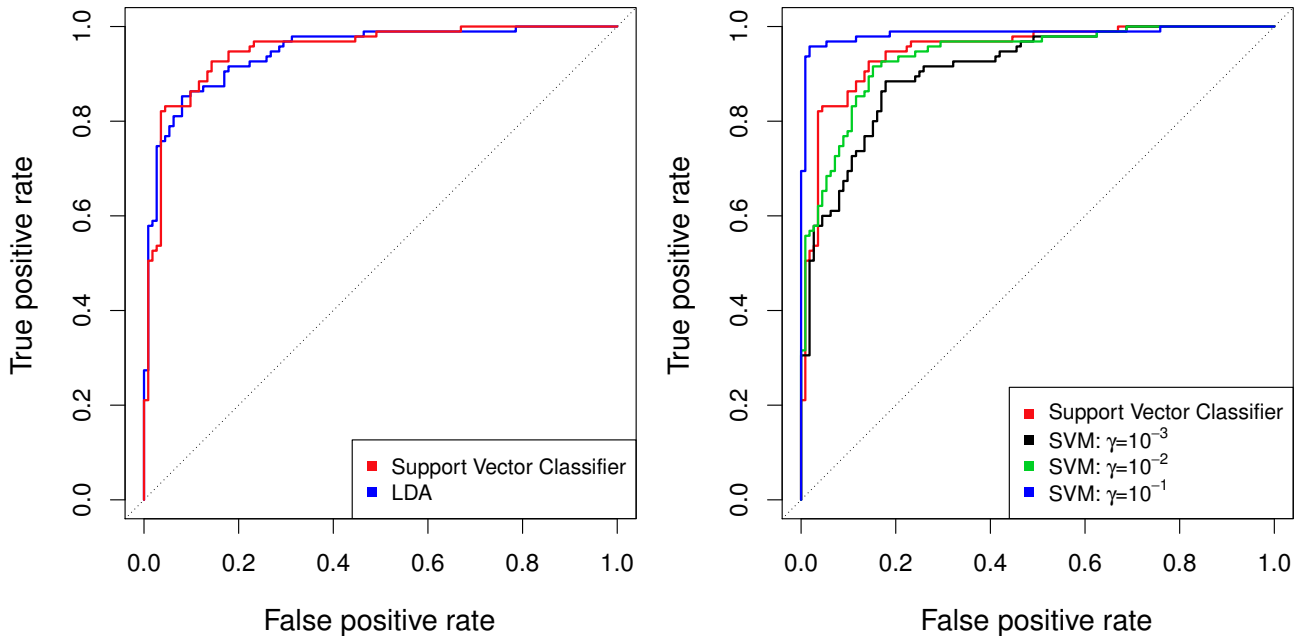

Figure: ROC curves for the Heart data training set. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}, 10^{-2}$ and 10^{-1}.

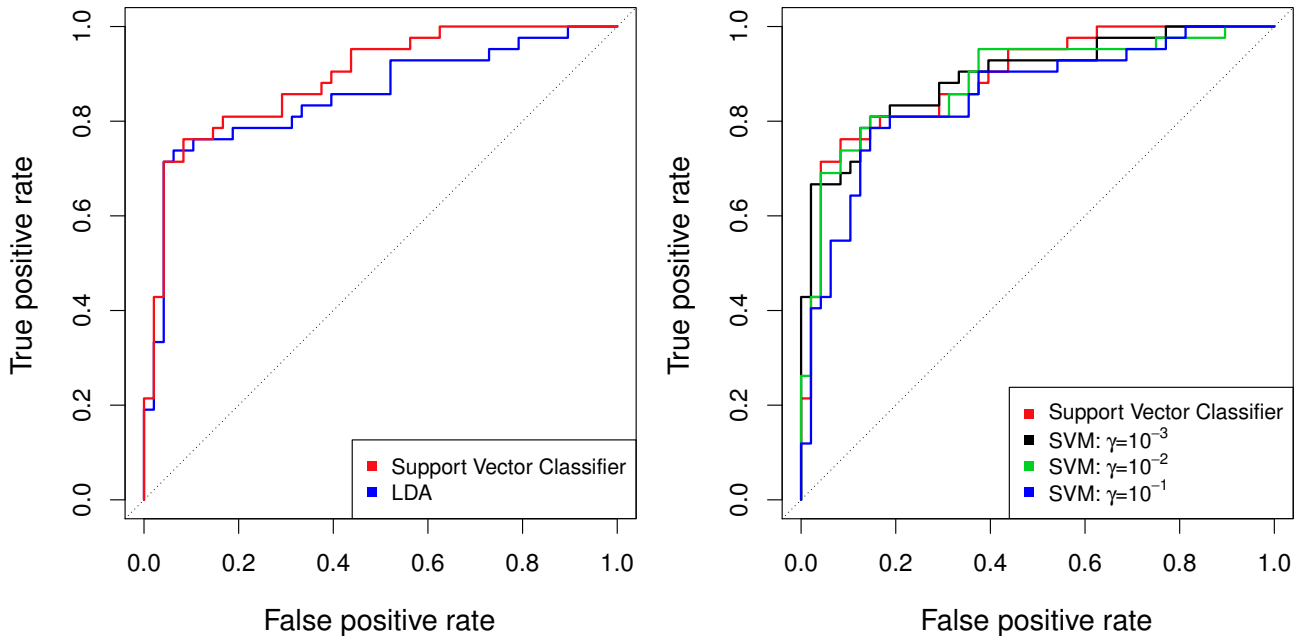

Figure: 9.11. ROC curves for the test set of the Heart data. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}, 10^{-2}$ and 10^{-1}.

One-versus-one approach

- With $K > 2$ classes.
- Run a SVM on each of the $\binom{K}{2}$ pairs of classes.
- We obtain $\binom{K}{2}$ SVMs.
- For every test observations, compute the number of times it is classified into class k by all the SVMs, denote as d_k.
- Classify it into the class with highest d_k (majortiy vote).

One-versus-all approach

- With $K > 2$ classes.
- Run a SVM on class k (coded as $+1$) versus class "not-k" (coded as -1): $f_k(\mathbf{x})$. (Note that the larger $f_k(\mathbf{x})$, the more likely $\mathbf{x}$ is in class k.
- For a new test observation with $\mathbf{x}$, compute assign to the class with largest $f_k(\mathbf{x})$.

Recall the "Loss + Penalty" formula

- Minimize, for f in certain space,

$$\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda P(f)$$

- Ridge regression: for linear f,

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Lasso regression: for linear f

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Logistic regression for classification

- Data: $(y_i, \mathbf{x}_i)$, $y_i = \pm 1$.
- The logistic model assumes

$$P(Y = 1|X) = 1/(1 + e^{-f(X)}); \quad P(Y = -1|X) = 1/(1 + e^{f(X)})$$

That is

$$P(Y = y|X) = 1/(1 + e^{-yf(X)})$$

Logistic regression for classification

- The negative of logistic likelihood

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)})$$

- Logistic loss with ridge l_2 penalty

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)}) + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Logistic loss with Lasso l_1 penalty:

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

The SVM

- Data: $(y_i, \mathbf{x}_i)$, $y_i = \pm 1$.
- SVM is a result of "hige loss + ridge penalty":

$$\sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2.$$

  where $f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta x_p$.
- the hinge loss function : $(1 - x)_+$.
- the logistic loss function: $\log(1 + e^{-x})$.

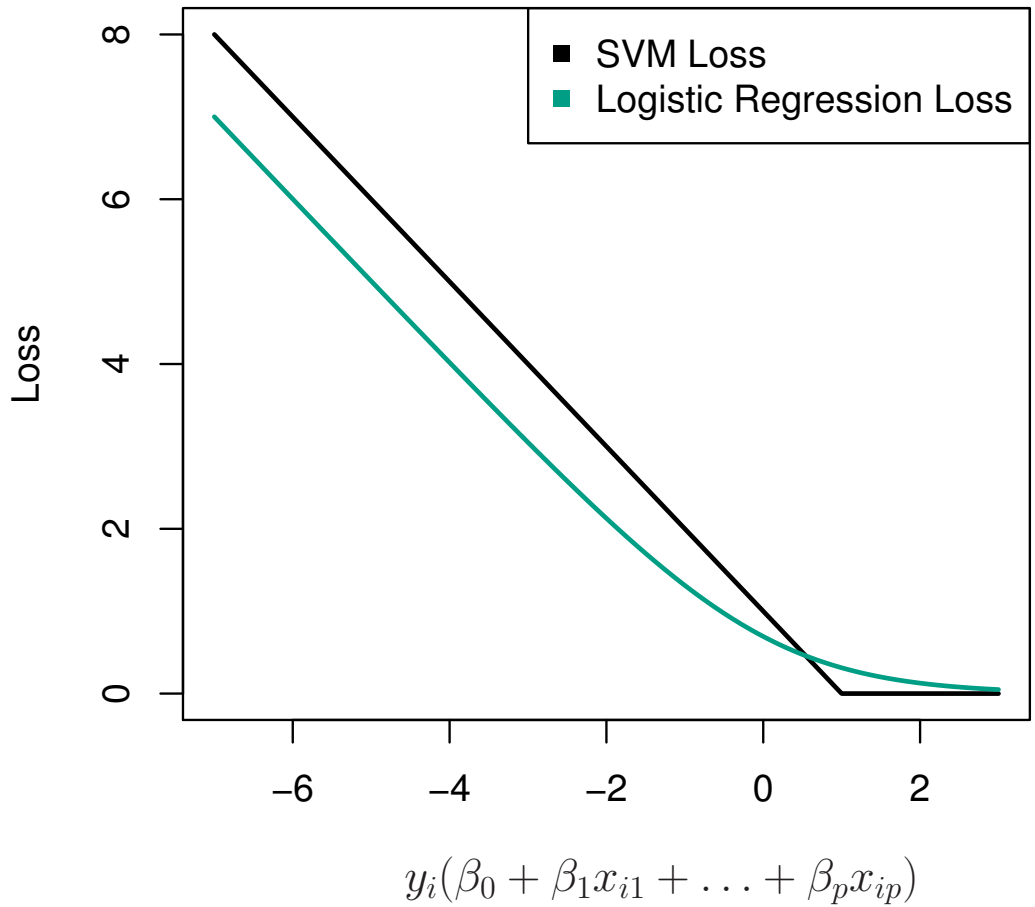

Figure: 9.12. The SVM and logistic regression loss functions are compared, as a function of $y_i(\beta_0 + \beta_1 x_{i1} + ...\beta_p x_{ip})$. When $y_i(\beta_0 + \beta_1 x_{i1} + ...\beta_p x_{ip})$ is greater than 1, then the SVM loss is zero, since this corresponds to an observation that is on the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

Exercises

Run the R-Lab codes in Section 9.6 of ISLR
Exercises 1-3 of Section 9.7 of ISLR

End of Chapter 9.