

We see that using `cost=10` yields two test set errors on this data.

## 9.7 Exercises

### *Conceptual*

- This problem involves hyperplanes in two dimensions.
  - Sketch the hyperplane  $1 + 3X_1 - X_2 = 0$ . Indicate the set of points for which  $1 + 3X_1 - X_2 > 0$ , as well as the set of points for which  $1 + 3X_1 - X_2 < 0$ .
  - On the same plot, sketch the hyperplane  $-2 + X_1 + 2X_2 = 0$ . Indicate the set of points for which  $-2 + X_1 + 2X_2 > 0$ , as well as the set of points for which  $-2 + X_1 + 2X_2 < 0$ .
- We have seen that in  $p = 2$  dimensions, a linear decision boundary takes the form  $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$ . We now investigate a non-linear decision boundary.

- Sketch the curve

$$(1 + X_1)^2 + (2 - X_2)^2 = 4.$$

- On your sketch, indicate the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

as well as the set of points for which

$$(1 + X_1)^2 + (2 - X_2)^2 \leq 4.$$

- Suppose that a classifier assigns an observation to the blue class if

$$(1 + X_1)^2 + (2 - X_2)^2 > 4,$$

and to the red class otherwise. To what class is the observation  $(0, 0)$  classified?  $(-1, 1)$ ?  $(2, 2)$ ?  $(3, 8)$ ?

- Argue that while the decision boundary in (c) is not linear in terms of  $X_1$  and  $X_2$ , it is linear in terms of  $X_1$ ,  $X_1^2$ ,  $X_2$ , and  $X_2^2$ .

- Here we explore the maximal margin classifier on a toy data set.

- We are given  $n = 7$  observations in  $p = 2$  dimensions. For each observation, there is an associated class label.

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

Sketch the observations.

- Sketch the optimal separating hyperplane, and provide the equation for this hyperplane (of the form (9.1)).
- Describe the classification rule for the maximal margin classifier. It should be something along the lines of “Classify to Red if  $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > 0$ , and classify to Blue otherwise.” Provide the values for  $\beta_0$ ,  $\beta_1$ , and  $\beta_2$ .
- On your sketch, indicate the margin for the maximal margin hyperplane.
- Indicate the support vectors for the maximal margin classifier.
- Argue that a slight movement of the seventh observation would not affect the maximal margin hyperplane.
- Sketch a hyperplane that is *not* the optimal separating hyperplane, and provide the equation for this hyperplane.
- Draw an additional observation on the plot so that the two classes are no longer separable by a hyperplane.

### Applied

- Generate a simulated two-class data set with 100 observations and two features in which there is a visible but non-linear separation between the two classes. Show that in this setting, a support vector machine with a polynomial kernel (with degree greater than 1) or a radial kernel will outperform a support vector classifier on the training data. Which technique performs best on the test data? Make plots and report training and test error rates in order to back up your assertions.
- We have seen that we can fit an SVM with a non-linear kernel in order to perform classification using a non-linear decision boundary. We will now see that we can also obtain a non-linear decision boundary by performing logistic regression using non-linear transformations of the features.

- (a) Generate a data set with  $n = 500$  and  $p = 2$ , such that the observations belong to two classes with a quadratic decision boundary between them. For instance, you can do this as follows:

```
> x1=runif(500)-0.5
> x2=runif(500)-0.5
> y=1*(x1^2-x2^2 > 0)
```

- (b) Plot the observations, colored according to their class labels. Your plot should display  $X_1$  on the  $x$ -axis, and  $X_2$  on the  $y$ -axis.
- (c) Fit a logistic regression model to the data, using  $X_1$  and  $X_2$  as predictors.
- (d) Apply this model to the *training data* in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the *predicted* class labels. The decision boundary should be linear.
- (e) Now fit a logistic regression model to the data using non-linear functions of  $X_1$  and  $X_2$  as predictors (e.g.  $X_1^2$ ,  $X_1 \times X_2$ ,  $\log(X_2)$ , and so forth).
- (f) Apply this model to the *training data* in order to obtain a predicted class label for each training observation. Plot the observations, colored according to the *predicted* class labels. The decision boundary should be obviously non-linear. If it is not, then repeat (a)-(e) until you come up with an example in which the predicted class labels are obviously non-linear.
- (g) Fit a support vector classifier to the data with  $X_1$  and  $X_2$  as predictors. Obtain a class prediction for each training observation. Plot the observations, colored according to the *predicted class labels*.
- (h) Fit a SVM using a non-linear kernel to the data. Obtain a class prediction for each training observation. Plot the observations, colored according to the *predicted class labels*.
- (i) Comment on your results.
6. At the end of Section 9.6.1, it is claimed that in the case of data that is just barely linearly separable, a support vector classifier with a small value of `cost` that misclassifies a couple of training observations may perform better on test data than one with a huge value of `cost` that does not misclassify any training observations. You will now investigate this claim.
- (a) Generate two-class data with  $p = 2$  in such a way that the classes are just barely linearly separable.

- (b) Compute the cross-validation error rates for support vector classifiers with a range of `cost` values. How many training errors are misclassified for each value of `cost` considered, and how does this relate to the cross-validation errors obtained?
  - (c) Generate an appropriate test data set, and compute the test errors corresponding to each of the values of `cost` considered. Which value of `cost` leads to the fewest test errors, and how does this compare to the values of `cost` that yield the fewest training errors and the fewest cross-validation errors?
  - (d) Discuss your results.
7. In this problem, you will use support vector approaches in order to predict whether a given car gets high or low gas mileage based on the `Auto` data set.
- (a) Create a binary variable that takes on a 1 for cars with gas mileage above the median, and a 0 for cars with gas mileage below the median.
  - (b) Fit a support vector classifier to the data with various values of `cost`, in order to predict whether a car gets high or low gas mileage. Report the cross-validation errors associated with different values of this parameter. Comment on your results.
  - (c) Now repeat (b), this time using SVMs with radial and polynomial basis kernels, with different values of `gamma` and `degree` and `cost`. Comment on your results.
  - (d) Make some plots to back up your assertions in (b) and (c).

*Hint: In the lab, we used the `plot()` function for `svm` objects only in cases with  $p = 2$ . When  $p > 2$ , you can use the `plot()` function to create plots displaying pairs of variables at a time. Essentially, instead of typing*

```
> plot(svmfit, dat)
```

*where `svmfit` contains your fitted model and `dat` is a data frame containing your data, you can type*

```
> plot(svmfit, dat, x1~x4)
```

*in order to plot just the first and fourth variables. However, you must replace `x1` and `x4` with the correct variable names. To find out more, type `?plot.svm`.*

8. This problem involves the `OJ` data set which is part of the `ISLR` package.

- (a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.
- (b) Fit a support vector classifier to the training data using `cost=0.01`, with `Purchase` as the response and the other variables as predictors. Use the `summary()` function to produce summary statistics, and describe the results obtained.
- (c) What are the training and test error rates?
- (d) Use the `tune()` function to select an optimal `cost`. Consider values in the range 0.01 to 10.
- (e) Compute the training and test error rates using this new value for `cost`.
- (f) Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the default value for `gamma`.
- (g) Repeat parts (b) through (e) using a support vector machine with a polynomial kernel. Set `degree=2`.
- (h) Overall, which approach seems to give the best results on this data?