Statistical Learning, Tutorato #7

Veronica Vinciotti, Marco Chierici

April 26, 2021

Exercise 1

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.	X1	X2	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 optimal separating hyperplane, and provide the equation for this hyperplane (see equation 9.1 in the textbook).
- Describe the classification rule for the maximal margin classifier. It should be something along the lines of "Classify to Red if $\beta_0 + \beta_1 X1 + \beta_2 X2 > 0$, and classify to Blue otherwise." Provide the values for β_0, β_1, and β_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.
- Verify the solutions that you obtained by using the function 'svm()' in the `e1071` library.
- 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.

Hints

In R, Support Vector Classifiers (SVC) are implemented in the library `e1071`, which we already used for Naive Bayes. Suppose you have your data stored in a dataframe `dat`, with `y` being the outcome variable. To fit a SVC you must first encode `y` as factor, and then you can call the function `svm(y ~ ., data=dat, kernel="linear", cost=C)`, using the usual formula syntax and specifying a value C for the cost parameter. Note that the tuning parameter is the inverse of what was used at lectures, so a large C (say larger than 1) should get you close to the maximal marginal hyperplane (C=0 in the lecture notes).

Exercise 2

In this exercise, we evaluate a support vector classifier on simulated data. To this aim:

- Generate a data set with $n = 500$ observations and $p = 2$ variables, such that the observations belong to two classes with a linear decision boundary between them. For instance, you can do this by specifying the outcome variable to be derived from a linear combination of the independent variables (and add some error to allow for some overlapping):

```
n_obs <- 500
x1 <- runif(n_obs) - 0.5
x2 <- runif(n_obs) - 0.5
er <- rnorm(n_obs, 0, 0.01)
y <- 1 * (3 * x1 - 2 * x2 + er > 0)
```

- 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.
- 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.
- Add the true decision surface on the plot. How did the method do?

Exercise 3

In this problem, you will use a support vector classifier to predict whether a given car gets a high or low gas mileage based on a number of predictors describing the vehicle. For the analysis we will use the `Auto` data set in the `ISLR` library.

- 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.
- 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.
- Find ways of visualizing the results e.g. plotting pairs of predictors and colouring the two classes. Comment on your results.

Hints

The `tune()` function can be used for tuning the cost parameter in a cross-validation setting (by default it performs a 10-fold cross-validation). The syntax for SVC is the following:

```
tune(svm, y ~ ., data = dat, kernel = "linear", ranges = list(cost = vector_of_C_values))
```

The `plot()` svm function has a functionality also for $p > 2$. Check `?plot.svm` for examples on how to do this.