hw2

Ken Ye

2023-10-03

Question 3

We will now perform cross-validation on a simulated data set.

(a) Generate a simulated data set as follows:

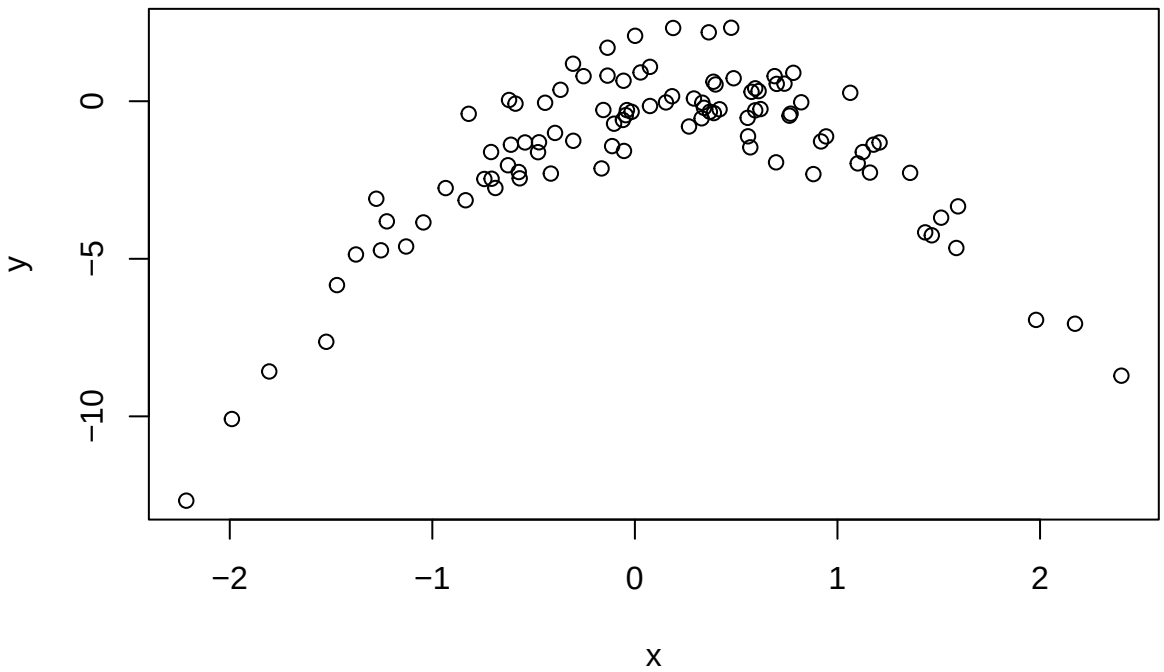
```
set.seed(1)
x = rnorm(100)
y = x - 2 * x^2 + rnorm(100)
```

In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

The n (number of observations) is 100, and the p (number of predictors) is 2. The generative model is $Y = X - 2X^2 + \epsilon$, where ϵ is the noise term following distribution N(0,1).

(b) Create a scatterplot of X against Y. Comment on what you find.

```
plot(x, y)
```



The relationship between X and Y is clearly not linear, instead, it seem to follow a parabolic (concave down) trend. The peak of Y occurs at around X=0.

(c) Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares. Note you may find it helpful to use the data.frame() function to create a single data set containing both X and Y.

```
library(boot)
set.seed(2)
df <- data.frame(x, y)
```

```
# Model 1: Linear
model1 = glm(y ~ x)
cv1 = cv.glm(df, model1)
loocv_error1 = cv1$delta[1]

# Model 2: Quadratic
model2 = glm(y ~ x + I(x^2))
cv2 = cv.glm(df, model2)
loocv_error2 = cv2$delta[1]

# Model 3: Cubic
model3 = glm(y ~ x + I(x^2) + I(x^3))
cv3 = cv.glm(df, model3)
loocv_error3 = cv3$delta[1]

# Model 4: Quartic
model4 = glm(y ~ x + I(x^2) + I(x^3) + I(x^4))
cv4 = cv.glm(df, model4)
loocv_error4 = cv4$delta[1]
```

```
# Compare LOOCV Errors
loocv_errors = data.frame(
  Model = c("Linear", "Quadratic", "Cubic", "Quartic"),
  LOOCV_Error = c(loocv_error1, loocv_error2, loocv_error3, loocv_error4)
)
print(loocv_errors)
```

```
##       Model LOOCV_Error
## 1    Linear   7.2881616
## 2 Quadratic   0.9374236
## 3     Cubic   0.9566218
## 4   Quartic   0.9539049
```

(d) Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why?

```
set.seed(3)
```

```
# Model 1: Linear
model1 = glm(y ~ x)
cv1 = cv.glm(df, model1)
loocv_error1 = cv1$delta[1]

# Model 2: Quadratic
model2 = glm(y ~ x + I(x^2))
```

```r
cv2 = cv.glm(df, model2)
loocv_error2 = cv2$delta[1]

# Model 3: Cubic
model3 = glm(y ~ x + I(x^2) + I(x^3))
cv3 = cv.glm(df, model3)
loocv_error3 = cv3$delta[1]

# Model 4: Quartic
model4 = glm(y ~ x + I(x^2) + I(x^3) + I(x^4))
cv4 = cv.glm(df, model4)
loocv_error4 = cv4$delta[1]
```

```r
# Compare LOOCV Errors
loocv_errors = data.frame(
  Model = c("Linear", "Quadratic", "Cubic", "Quartic"),
  LOOCV_Error = c(loocv_error1, loocv_error2, loocv_error3, loocv_error4)
)
print(loocv_errors)
```

```
##       Model LOOCV_Error
## 1    Linear   7.2881616
## 2 Quadratic   0.9374236
## 3     Cubic   0.9566218
## 4   Quartic   0.9539049
```

The results (LOOCV errors) are exactly the same for (c) and (d), though we used different seeds. This is because LOOCV uses each data point as a test set exactly once and averages the error over all iterations, and in this case, the random seed (which likely changes the order which each data point is used as a test set) doesn't make a difference.

(e) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The second model (quadratic) in (c) had the smallest LOOCV error, and it fits my expectation because we know the true generative model $Y = X - 2X^2 + \epsilon$ indicates a quadratic relationship between X and Y, which is illustrate in the scatterplot in (b) as well.