M072040019 梅瀚中

1. Using basic statistical properties of the variance, as well as single-variable calculus, derive (5.6). In other words, prove that α given by (5.6) does indeed minimize $\mathrm{Var}(\alpha X + (1-\alpha)Y)$.

$$Var(\alpha X + (1-\alpha)Y) = \alpha^2\sigma_x^2 + (1-\alpha)^2\sigma_Y^2 + 2\alpha(1-\alpha)\sigma_{xy}$$

$$\frac{\partial}{\partial\alpha}Var(\alpha X + (1-\alpha)Y) = 2\alpha\sigma_x^2 - 2(1-\alpha)\sigma_Y^2 + 2\sigma_{xy} - 4\alpha\sigma_{xy} = 0$$

$$\Rightarrow 2\alpha\sigma_x^2 - 2\sigma_Y^2 + 2\alpha\sigma_Y^2 + 2\sigma_{xy} - 4\alpha\sigma_{xy} = 0$$

$$\therefore \alpha = \frac{\sigma_Y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_Y^2 - 2\sigma_{xy}}$$

2. We will now derive the probability that a given observation is part of a bootstrap sample. Suppose that we obtain a bootstrap sample from a set of n observations.

   (a) What is the probability that the first bootstrap observation is *not* the jth observation from the original sample? Justify your answer.

   (b) What is the probability that the second bootstrap observation is *not* the jth observation from the original sample?

   (c) Argue that the probability that the jth observation is *not* in the bootstrap sample is $(1 - 1/n)^n$.

   (d) When $n = 5$, what is the probability that the jth observation is in the bootstrap sample?

   (e) When $n = 100$, what is the probability that the jth observation is in the bootstrap sample?

(f) When $n = 10,000$, what is the probability that the jth observation is in the bootstrap sample?

(g) Create a plot that displays, for each integer value of n from 1 to 100,000, the probability that the jth observation is in the bootstrap sample. Comment on what you observe.

(h) We will now investigate numerically the probability that a bootstrap sample of size $n = 100$ contains the jth observation. Here $j = 4$. We repeatedly create bootstrap samples, and each time we record whether or not the fourth observation is contained in the bootstrap sample.

```
> store=rep(NA, 10000)
> for(i in 1:10000){
    store[i]=sum(sample(1:100, rep=TRUE)==4)>0
}
> mean(store)
```

Comment on the results obtained.

(a)

1-1/n

(b)

1-1/n

(c)

題意=在 n 次裡的 boostrap sample,每一個 boostrap observation 都不是第 j 個 observation 的值

(1-1/n)* (1-1/n)* (1-1/n)* (1-1/n)*…….* (1-1/n)= (1-1/n)^n

(d)

P(全部- jth observation is not in the bootstrap sample)

=1- (1-1/5)^5

=0.67232

(e)

P(全部- jth observation is not in the bootstrap sample)

=1-(1/100)^100

= 0.6339677

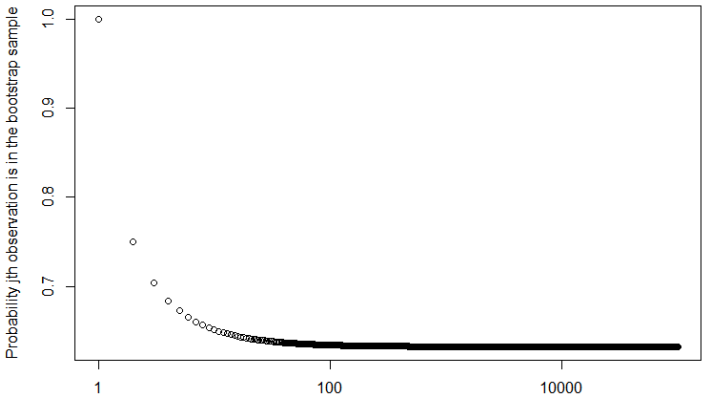
(f)

P(全部- jth observation is not in the bootstrap sample)

=1-(1/10000)^10000

=0.632139

(g)




```
> x <- 1:100000
> y=sapply(x,function(x){1-((1-(1/x))^x)})
> plot(x,y,xlab="n",ylab="Probability jth observation is in the bootstrap sample",l
og="x")
```

(h)

我們得到從 boostrap sample size n,抽到第 j 個觀察樣本的機率,最後會收斂到 0.632。

3. We now review k-fold cross-validation.

   (a) Explain how k-fold cross-validation is implemented.

   (b) What are the advantages and disadvantages of k-fold cross-validation relative to:

      i. The validation set approach?

      ii. LOOCV?

(a)

原始的樣本數據被隨機分成 k 份,在 k 份子集合中每次挑選一份作為 vaildation set,其他 k-1 份作為 training set, 在每一個 training set 上訓練後得到一個模型,用這個模型在 testing set 上測試。這樣的過程重複進行 k 次,每一份都會有一次作為 vaildation set。最後在把這 k 次的測試結果進行平均,得到最後的測試結果。

(b)

(i) 缺點:相較 k-fold cross validation ,vaildation set 法的 test error 會有較高的變異程度

優點: 相較 k-fold cross validation ,vaildation set 法,節省運算時間很多。

(ii) LOOCV 法就是 k-fold cross validation,只是他的 k=樣本數

缺點: 如果樣本數很大,相較 k-fold cross validation,LOOCV 法運算時間增加非常多。

優點:LOOCV, 每一回合中幾乎使用所有的樣本訓練，因此最接近母体樣本的分布。

8. 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.

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

(c) Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares:

i. $Y = \beta_0 + \beta_1 X + \epsilon$
ii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$
iii. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$
iv. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \epsilon.$

Note you may find it helpful to use the `data.frame()` function to create a single data set containing both X and Y.

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

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

(f) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

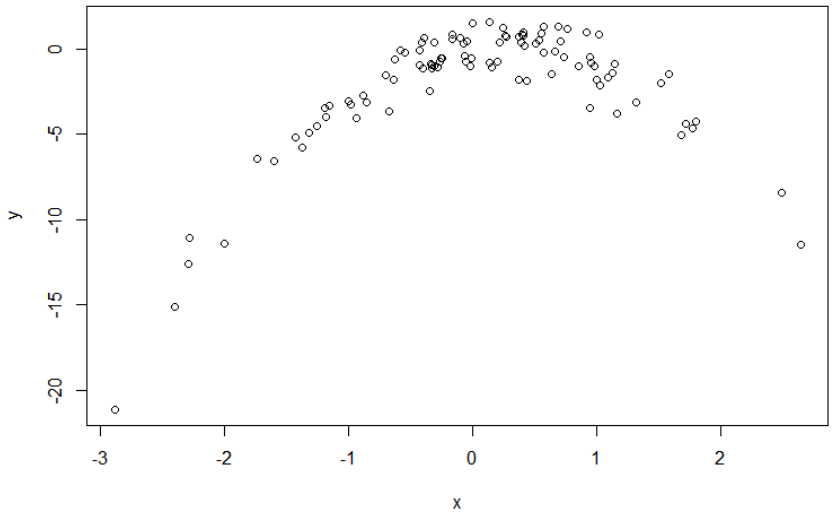
(a)

n=100 *and* p=2

$$Y = X - 2X^2 + \varepsilon$$

$$\varepsilon \sim N(0,1)$$

(b)



呈現非線性相關。

(c)

```
> cv.glm(Data, fit.glm.1)$delta[1]
[1] 7.288162
> fit.glm.2 <- glm(y ~ poly(x, 2))
> cv.glm(Data, fit.glm.2)$delta[1]
[1] 0.9374236
> fit.glm.3 <- glm(y ~ poly(x, 3))
> cv.glm(Data, fit.glm.3)$delta[1]
[1] 0.9566218
> fit.glm.4 <- glm(y ~ poly(x, 4))
> cv.glm(Data, fit.glm.4)$delta[1]
[1] 0.9539049
```

(d)

```
> set.seed(666666666)
> fit.glm.1 <- glm(y ~ x)
> cv.glm(Data, fit.glm.1)$delta[1]
[1] 7.288162
> fit.glm.2 <- glm(y ~ poly(x, 2))
> cv.glm(Data, fit.glm.2)$delta[1]
[1] 0.9374236
> fit.glm.3 <- glm(y ~ poly(x, 3))
> cv.glm(Data, fit.glm.3)$delta[1]
[1] 0.9566218
> fit.glm.4 <- glm(y ~ poly(x, 4))
> cv.glm(Data, fit.glm.4)$delta[1]
[1] 0.9539049
```

因為 LOOCV 是把每一筆的資料都會當作 test,所以不管 seed 是從哪一筆觀察值開始,結果都會一樣。

(e)

Fit.glm2 的 test MSE 是最小的。

(F)

```
> summary(fit.glm.4)

Call:
glm(formula = y ~ poly(x, 4))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0550  -0.6212  -0.1567   0.5952   2.2267

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.55002    0.09591 -16.162  < 2e-16 ***
poly(x, 4)1   6.18883    0.95905   6.453 4.59e-09 ***
poly(x, 4)2 -23.94830    0.95905 -24.971  < 2e-16 ***
poly(x, 4)3   0.26411    0.95905   0.275    0.784
poly(x, 4)4   1.25710    0.95905   1.311    0.193
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.9197797)

    Null deviance: 700.852  on 99  degrees of freedom
Residual deviance:  87.379  on 95  degrees of freedom
AIC: 282.3
```

只有線性和二次項的係數是顯著的,跟我們用交叉驗證得出的結果是一致的

9. We will now consider the `Boston` housing data set, from the `MASS` library.

   (a) Based on this data set, provide an estimate for the population mean of `medv`. Call this estimate $\hat{\mu}$.

   (b) Provide an estimate of the standard error of $\hat{\mu}$. Interpret this result.
   Hint: We can compute the standard error of the sample mean by dividing the sample standard deviation by the square root of the number of observations.

   (c) Now estimate the standard error of $\hat{\mu}$ using the bootstrap. How does this compare to your answer from (b)?

   (d) Based on your bootstrap estimate from (c), provide a 95 % confidence interval for the mean of `medv`. Compare it to the results obtained using `t.test(Boston$medv)`.
   Hint: You can approximate a 95 % confidence interval using the formula $[\hat{\mu} - 2SE(\hat{\mu}), \hat{\mu} + 2SE(\hat{\mu})]$.

   (e) Based on this data set, provide an estimate, $\hat{\mu}_{med}$, for the median value of `medv` in the population.

   (f) We now would like to estimate the standard error of $\hat{\mu}_{med}$. Unfortunately, there is no simple formula for computing the standard error of the median. Instead, estimate the standard error of the median using the bootstrap. Comment on your findings.

   (g) Based on this data set, provide an estimate for the tenth percentile of `medv` in Boston suburbs. Call this quantity $\hat{\mu}_{0.1}$. (You can use the `quantile()` function.)

   (h) Use the bootstrap to estimate the standard error of $\hat{\mu}_{0.1}$. Comment on your findings.

(a)

```
> X <- mean(medv)
> X
[1] 22.53281
```

(b)

$$SD_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

```
> dim(Boston)
[1] 506  14
> se.hat <- sd(medv) / sqrt(506)
> se.hat
[1] 0.4088611
```

(c)

```
> boot.fn <- function(data,i) {
+   mu <- mean(data[i])
+   return (mu)
+ }
> boot(medv, boot.fn, 100)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 100)


Bootstrap Statistics :
    original      bias    std. error
t1* 22.53281 0.006994071   0.4050308
```

Boostrap:0.40530308

(d)

```
> t.test(medv)

        One Sample t-test

data:  medv
t = 55.111, df = 505, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 21.72953 23.33608
sample estimates:
mean of x
 22.53281
```

```
> aaa <- c(22.532-2*0.4050308,22.532+2*0.4050308)
> aaa
[1] 21.72194 23.34206
```

兩個信賴區間 差不多。

(e)

```
> med.hat <- median(medv)
> med.hat
[1] 21.2
```

(f)

```
> boot.fn <- function(data, i) {
+   mu <- median(data[i])
+   return (mu)
+ }
> boot(medv, boot.fn, 100)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 100)


Bootstrap Statistics :
    original  bias    std. error
t1*     21.2 -0.0305   0.3749677
```

兩個得到的中位數估計值很接近。

(g)

```
> percent10 <- quantile(medv, 0.1)
> percent10
   10%
12.75
```

(h)

```
> boot.fn <- function(data, i) {
+   mu <- quantile(data[i],0.1)
+   return (mu)
+ }
> boot(medv, boot.fn, 100)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 100)


Bootstrap Statistics :
    original  bias    std. error
t1*    12.75  0.0505   0.4720734
```