Statistical Learning and Data mining

Homework 8

M052040003 鍾冠毅

5.6.

a.

```
Call:
glm(formula = default ~ income + balance, family = binomial,
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4725  -0.1444  -0.0574  -0.0211   3.7245

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1579.0  on 9997  degrees of freedom
AIC: 1585

Number of Fisher Scoring iterations: 8
```

b. c.

```
> boot.fn <- function(data, index){
+   fit.fn <- glm(default ~ income + balance, data = data,
+                 family = binomial, subset = index);
+   fit.fn$coefficients
+ }
> bt.6c <- boot(Default, boot.fn, 87)
> bt.6c

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Default, statistic = boot.fn, R = 87)


Bootstrap Statistics :
         original         bias     std. error
t1* -1.154047e+01 -9.901750e-02 4.614741e-01
t2*  2.080898e-05  8.493221e-07 4.657274e-06
t3*  5.647103e-03  3.684631e-05 2.471336e-04
```

d. The standard errors are closed in the two methods.

5.7.

a.

```
Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.623   -1.261    1.001    1.083    1.506

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.22122    0.06147   3.599 0.000319 ***
Lag1        -0.03872    0.02622  -1.477 0.139672
Lag2         0.06025    0.02655   2.270 0.023232 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1488.2  on 1086  degrees of freedom
AIC: 1494.2

Number of Fisher Scoring iterations: 4
```

b.

```
Call:
glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Weekly[-1,
    ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6258  -1.2617   0.9999   1.0819   1.5071

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.22324    0.06150   3.630 0.000283 ***
Lag1        -0.03843    0.02622  -1.466 0.142683
Lag2         0.06085    0.02656   2.291 0.021971 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1494.6  on 1087  degrees of freedom
Residual deviance: 1486.5  on 1085  degrees of freedom
AIC: 1492.5

Number of Fisher Scoring iterations: 4
```

c.

```
> predict7c <- ifelse(predict(fit7b, Weekly[1,2:3], type = "response") > .5
,
+                      "Up", "Down")
> predict7c == Weekly$Direction[1]
    1
FALSE
```

d. e.

```
> error7d <-
+   sapply(1:dim(Weekly)[1], function(n){
+     fit7d <- glm(Direction ~ Lag1 + Lag2, data = Weekly[-n,], family = bi
nomial);
+     predict7d <- ifelse(predict(fit7d, Weekly[n,2:3], type = "response")
>= .5,
+                          "Up", "Down")
+     predict7d != Weekly$Direction[n]
+   })
> sum(error7d)
[1] 490
> mean(error7d)
[1] 0.4499541
```

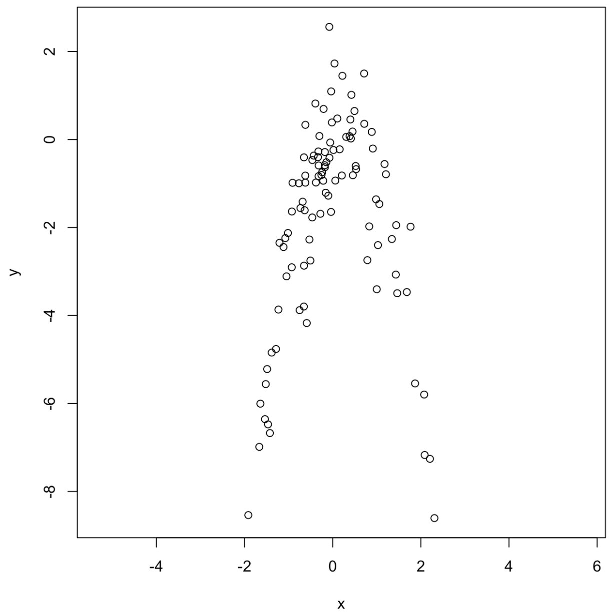
5.8.

a.

$n = 100, \ p = 2$

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

b. It is a convex quadratic plot which's x ranges -2 to 2 and y ranges -8 to 2.



c. d. e.

They are exact the same for the LOOCV have no random effect. The different seeds doesn't matter.

```
> cv.glm(df8, fit8ci  )$delta      > cv.glm(df8, fit8ci  )$delta
[1] 5.890979 5.888812              [1] 5.890979 5.888812
> cv.glm(df8, fit8cii )$delta      > cv.glm(df8, fit8cii )$delta
[1] 1.086596 1.086326              [1] 1.086596 1.086326
> cv.glm(df8, fit8ciii)$delta      > cv.glm(df8, fit8ciii)$delta
[1] 1.102585 1.102227              [1] 1.102585 1.102227
> cv.glm(df8, fit8civ )$delta      > cv.glm(df8, fit8civ )$delta
[1] 1.114772 1.114334              [1] 1.114772 1.114334
```

f. The result shows that only the 1st and the 2nd order term are significant. It's consistent with LOOCV.

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8914  -0.5244   0.0749   0.5932   2.7796

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.8277     0.1041 -17.549   <2e-16 ***
poly(x, 4)1   2.3164     1.0415   2.224   0.0285 *
poly(x, 4)2 -21.0586     1.0415 -20.220   <2e-16 ***
poly(x, 4)3  -0.3048     1.0415  -0.293   0.7704
poly(x, 4)4  -0.4926     1.0415  -0.473   0.6373
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1.084654)

    Null deviance: 552.21  on 99  degrees of freedom
Residual deviance: 103.04  on 95  degrees of freedom
AIC: 298.78

Number of Fisher Scoring iterations: 2
```

5.9.

a. 22.53281

b. 0.4088611

c. It's similar to the result obtained above.

```
> boot.fn <- function(data, index) return(mean(data[index]))
> set.seed(87)
> bstrp <- boot(Boston$medv, boot.fn, 5487)
> bstrp

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Boston$medv, statistic = boot.fn, R = 5487)


Bootstrap Statistics :
    original        bias    std. error
t1* 22.53281 0.001634982   0.4119173
```

d. It's similar to the result obtained above.

```
> t.test(Boston$medv)$conf[1:2]
[1] 21.72953 23.33608
> c(bstrp$t0 - 2*0.4101611, bstrp$t0 + 2*0.4101611)
[1] 21.71248 23.35313
```

e. 2.21

f. 0.3777244

```
> boot.fn <- function(data, index) return(median(data[index]))
> set.seed(87)
> boot(Boston$medv, boot.fn, 5487)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Boston$medv, statistic = boot.fn, R = 5487)


Bootstrap Statistics :
    original      bias    std. error
t1*     21.2 -0.01712229   0.3777244
```

g. 12.75

h. 0.5011288

```
> boot.fn <- function(data, index) return(quantile(data[index], .1))
> set.seed(87)
> boot(Boston$medv, boot.fn, 5487)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Boston$medv, statistic = boot.fn, R = 5487)


Bootstrap Statistics :
    original     bias    std. error
t1*    12.75 0.01123565   0.5011288
```

6.1.

a.

The best subset selection has the smallest training error, since the other two method have particular model-choosing paths which may skip the best one.

b.

The best subset selection nay have the smallest test error, since it considers more models than the other two methods.

c.

i. T   ii. T   iii. F   iv. F   v. F