Statistical Learning and Data mining

Homework 7

M052040003 鍾冠毅

1.

$$
\begin{aligned}
Var(\alpha X + (1-\alpha)Y) &= Var(\alpha X) + Var((1-\alpha)Y) + 2Cov(\alpha X, (1-\alpha)Y) \\
&= \alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + 2\alpha(1-\alpha)Cov(X,Y) \\
&= \alpha^2\sigma_X^2 + (1-\alpha)^2\sigma_Y^2 + 2\alpha(1-\alpha)\sigma_{XY} \\
0 &= \frac{d}{d\alpha}Var(\alpha X + (1-\alpha)Y) \\
0 &= 2\alpha\sigma_X^2 - 2(1-\alpha)\sigma_Y^2 + 2(1-2\alpha)\sigma_{XY} \\
0 &= \alpha\sigma_X^2 + (\alpha-1)\sigma_Y^2 + (1-2\alpha)\sigma_{XY} \\
0 &= (\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})\alpha + (-\sigma_Y^2 + \sigma_{XY}) \\
\alpha &= \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}
\end{aligned}
$$

2.

(a) $Pr(in) = 1 - Pr(out) = 1 - (1 - \frac{1}{n}) = \frac{n-1}{n}$

(b) $\frac{n-1}{n}$

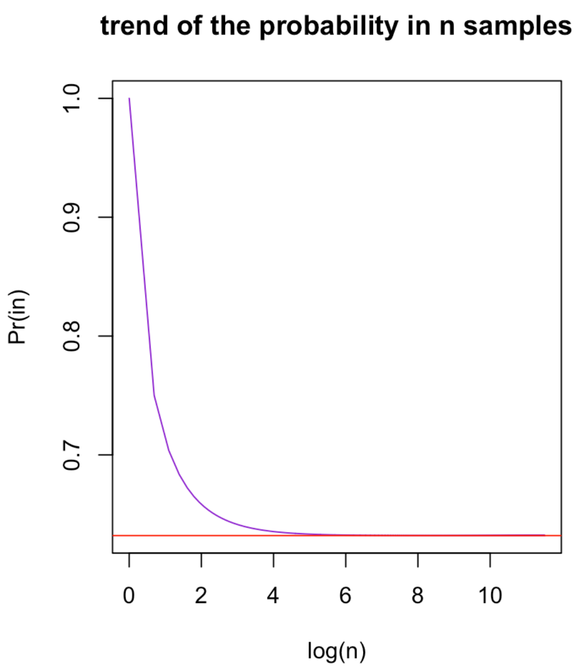
(c)

For every sampling in bootstrap, we consider the whole sample space. That is, we sample with replacement and repeat it n times. By the product rule, the probability that we consider is $(\frac{n-1}{n})^n$.

(d)
$$
\begin{aligned}
Pr(in) &= 1 - Pr(out) \\
&= 1 - (1 - \frac{1}{5})^5 \\
&= 1 - (\frac{4}{5})^5 \\
&= 0.67232
\end{aligned}
$$

(e)
$$
\begin{aligned}
Pr(in) &= 1 - Pr(out) \\
&= 1 - (1 - \frac{1}{100})^{100} \\
&= 1 - (\frac{99}{100})^{100} \\
&= 0.63340
\end{aligned}
$$

(f)
$$
\begin{aligned}
Pr(in) &= 1 - Pr(out) \\
&= 1 - (1 - \frac{1}{10000})^{10000} \\
&= 1 - (\frac{9999}{10000})^{10000} \\
&= 0.63214
\end{aligned}
$$



(g) As the figure shown above, the probability converge to 0.6321224.

(h) It return 0.6343 which is closed to the probability obtained above.

3. Suppose that we a data set with n observations.

(a)

i. Randomly split the n observations in to k equal size subset without overlapping.

ii. Taking the k-th subset as the test set to calculate the k-th MSE. The union of other (k-1) subsets are taken as training set for predicting model.

iii. The test error is the average of the k MSE estimates.

(b)

i. The concept of the validation set approach is much more trivial. It's a simple way to partition a dataset; yet, for using the less training data to build a model, the test error which is highly variable depending on the training data tends to be overestimated.

ii. LOOCV is a special case of k-folds cross validation. It take k = n, that is, there are n MSE estimates in this method. Thus, it takes more time to compute the test error in this error. Yet, it has higher variance and lower bias than the k-folds cross validation.

4.

By using the bootstrap method, we resample observations with B times, where B is an large positive integer, and build a model obtaining its MSE in each time. The expectation of the standard deviation will be the mean of the MSE estimates from the B times bootstrapping.

5.

(a)

```
> summary(fit5a)

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)

```
> # i
> set.seed(3)
> tr5b <- sample(dim(Default)[1], round(0.5*dim(Default)[1]))
> # ii
> fit5b <- glm(default ~ income + balance, data = Default, family = "binomial", subset = tr5b)
> summary(fit5b)

Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default, subset = tr5b)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1014  -0.1433  -0.0569  -0.0206   3.7241

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.160e+01  6.055e-01 -19.162  < 2e-16 ***
income       2.254e-05  6.972e-06   3.233  0.00123 **
balance      5.660e-03  3.131e-04  18.079  < 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: 1530.39  on 4999  degrees of freedom
Residual deviance:  812.77  on 4997  degrees of freedom
AIC: 818.77

Number of Fisher Scoring iterations: 8

> # iii
> prob5b <- predict(fit5b, newdata = Default[-tr5b, ], type = "response")
> pred5b <- as.factor(ifelse(prob5b>0.5, "Yes", "No"))
> # iv
> te.err.5b <- mean(pred5b != Default[-tr5b, ]$default)
> te.err.5b
[1] 0.0248
```

(c) The test error rate seems to be around 0.026

```
> te.err.5c <-
+   sapply(1:3, function(k) {
+     set.seed(k+3);
+     tr5c <- sample(dim(Default)[1], round(0.5*dim(Default)[1]));
+     fit5c <- glm(default ~ income + balance, data = Default, family = "binomial", subset = tr5c);
+     prob5c <- predict(fit5c, newdata = Default[-tr5c, ], type = "response");
+     pred5c <- as.factor(ifelse(prob5c>0.5, "Yes", "No"));
+     mean(pred5c != Default[-tr5c, ]$default)
+   })
> te.err.5c
[1] 0.0262 0.0246 0.0270
```

(d) The test error rate seems not to be reduced.

```
> set.seed(11)
> tr5d <- sample(dim(Default)[1], round(0.5*dim(Default)[1]))
> fit5d <- glm(default ~ income + balance + student, data = Default, family = "binomial", subset = tr5d)
> prob5d <- predict(fit5d, newdata = Default[-tr5d, ], type = "response")
> pred5d <- as.factor(ifelse(prob5d > 0.5, "Yes", "No"))
> te.err.5d <- mean(pred5d != Default[-tr5d, ]$default)
> te.err.5d
[1] 0.027
```

```
pr <- function(n) 1-(1-(1/n))^n

x <- 1:100000

pr(x)

plot(log(x), pr(x),

    xlab = "log(n)", ylab = "Pr(in)", main = "trend of the probability in n samples",

    type = "l", col = "darkviolet")

abline(h = pr(100000), col = "red")


store <- rep(NA, 10000)

for(i in 1:10000){

  store[i]=sum(sample(1:100, rep=TRUE)==4)>0 }

mean(store)


# 5.a #

library(ISLR)

fit5a <- glm(default ~ income + balance, data = Default, family = binomial)

summary(fit5a)


# 5.b #

# i

set.seed(3)

tr5b <- sample(dim(Default)[1], round(0.5*dim(Default)[1]))

# ii

fit5b <- glm(default ~ income + balance, data = Default, family = "binomial", subset = tr5b)

summary(fit5b)

# iii

prob5b <- predict(fit5b, newdata = Default[-tr5b, ], type = "response")

pred5b <- as.factor(ifelse(prob5b>0.5, "Yes", "No"))

# iv

te.err.5b <- mean(pred5b != Default[-tr5b, ]$default)


# 5.c #

te.err.5c <-

  sapply(1:3, function(k) {

    set.seed(k+3);

    tr5c <- sample(dim(Default)[1], round(0.5*dim(Default)[1]));

    fit5c <- glm(default ~ income + balance, data = Default, family = "binomial", subset = tr5c);

    prob5c <- predict(fit5c, newdata = Default[-tr5c, ], type = "response");

    pred5c <- as.factor(ifelse(prob5c>0.5, "Yes", "No"));

    mean(pred5c != Default[-tr5c, ]$default)

  })


# 5.d #

set.seed(11)

tr5d <- sample(dim(Default)[1], round(0.5*dim(Default)[1]))

fit5d <- glm(default ~ income + balance + student, data = Default, family = "binomial", subset = tr5d)

prob5d <- predict(fit5d, newdata = Default[-tr5d, ], type = "response")

pred5d <- as.factor(ifelse(prob5d > 0.5, "Yes", "No"))

te.err.5d <- mean(pred5d != Default[-tr5d, ]$default)
```