Statistical hw2

全金

2025-03-16

3

As above,

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi\sigma_k}} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}{\sum_{l=1}^{k} \pi_l \frac{1}{\sqrt{2\pi\sigma_l}} \exp(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2)}$$

Now lets derive the Bayes classifier, without assuming $\sigma_1^2 = ... = \sigma_K^2$

Maximizing $p_k(x)$ also maximizes any monotonic function of $p_k(X)$, and therefore, we can consider maximizing $\log(p_K(X))$

$$\log(p_k(x)) = \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi\sigma_k}}\right) - \frac{1}{2\sigma_k^2}(x-\mu_k)^2 - \log\left(\sum_{l=1}^{k} \frac{1}{\sqrt{2\pi\sigma_l}}\pi_l \exp\left(-\frac{1}{2\sigma_l^2}(x-\mu_l)^2\right)\right)$$

Remember that we are maximizing over k, and since the last term does not vary with k it can be ignored. So we just need to maximize

$$\begin{aligned} f &= \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi\sigma_k}}\right) - \frac{1}{2\sigma_k^2}(x-\mu_k)^2 && (1) \\ &= \log(\pi_k) + \log\left(\frac{1}{\sqrt{2\pi\sigma_k}}\right) - \frac{x^2}{2\sigma_k^2} + \frac{x\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} && (2) \\ & && (3) \end{aligned}$$

However, unlike in Q2, $\frac{x^2}{2\sigma_k^2}$ is not independent of k, so we retain the term with x^2, hence f, the Bayes' classifier, is a quadratic function of x.

5

(a)

QDA, being a more flexible model, will always perform better on the training set, but LDA would be expected to perform better on the test set. ## (b) QDA, being a more flexible model, will perform better on the training set, and we would hope that extra flexibility translates to a better fit on the test set. ## (c) As n increases, we would expect the prediction accuracy of QDA relative to LDA to improve as there is more data to fit to subtle effects in the data. ## (d) False. QDA can overfit leading to poorer test performance.

12

(a)

The log odds is just $\hat{\beta}_0 + \hat{\beta}_1 x$ ## (b) From 4.14, log odds of our friend's model is:

$$(\hat{\alpha}_{orange0} - \hat{\alpha}_{apple0}) + (\hat{\alpha}_{orange1} - \hat{\alpha}_{apple1})x$$

(c)

We can say that in our friend's model $\hat{\alpha}_{orange0} - \hat{\alpha}_{apple0} = 2$ and $\hat{\alpha}_{orange1} - \hat{\alpha}_{apple1} = -1$. We are unable to know the specific value of each parameter however.

(d)

The coefficients in our model would be $\hat{\beta}_0 = 1.2 - 3 = -1.8$ and $\hat{\beta}_1 = -2 - 0.6 = -2.6$

(e)

The models are identical with different parameterization so they should perfectly agree.

13

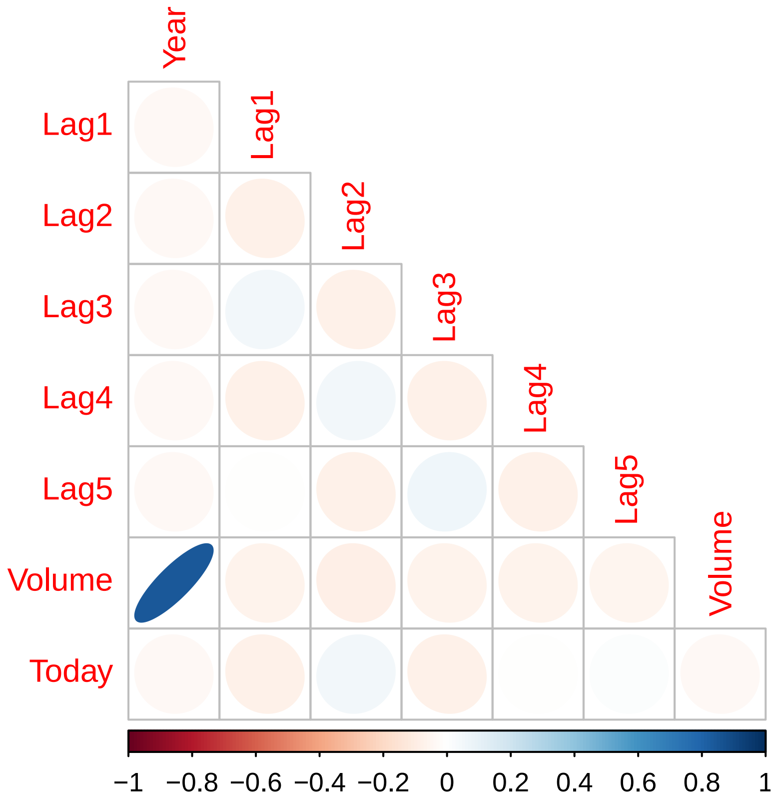
(a)

```r
library(MASS)
library(class)
library(tidyverse)
library(corrplot)
library(ISLR2)
library(e1071)
```

```r
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume            Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

```r
corrplot(cor(Weekly[, -9]), type = "lower", diag = FALSE, method = "ellipse")
```



Volume is strongly positively correlated with Year. Other correlations are week, but Lag1 is negatively correlated with Lag2 but positively correlated with Lag3.

(b)

```r
fit <- glm(
  Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
  data = Weekly,
  family = binomial
)
summary(fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
```

```
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## 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: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Lag2 is significant.

(c)

```r
contrasts(Weekly$Direction)
```

```
##      Up
## Down  0
## Up    1
```

```r
pred <- predict(fit, type = "response") > 0.5
(t <- table(ifelse(pred, "Up (pred)", "Down (pred)"), Weekly$Direction))
```

```
##
##              Down  Up
##   Down (pred)  54  48
##   Up (pred)   430 557
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.5610652
```

The overall fraction of correct predictions is 0.56. Although logistic regression correctly predicts upwards movements well, it incorrectly predicts most downwards movements as up.

(d)

```r
train <- Weekly$Year < 2009

fit <- glm(Direction ~ Lag2, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
(t <- table(ifelse(pred, "Up (pred)", "Down (pred)"), Weekly[!train, ]$Direction))
```

```
##
##               Down Up
##   Down (pred)    9  5
##   Up (pred)     34 56
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.625
```

(e)

```r
fit <- lda(Direction ~ Lag2, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
(t <- table(pred, Weekly[!train, ]$Direction))
```

```
##
## pred   Down Up
##   Down    9  5
##   Up     34 56
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.625
```

(f)

```r
fit <- qda(Direction ~ Lag2, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
(t <- table(pred, Weekly[!train, ]$Direction))
```

```
##
## pred   Down Up
##   Down    0  0
##   Up     43 61
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.5865385
```

(g)

```r
fit <- knn(
  Weekly[train, "Lag2", drop = FALSE],
  Weekly[!train, "Lag2", drop = FALSE],
  Weekly$Direction[train]
)
(t <- table(fit, Weekly[!train, ]$Direction))
```

```
##
## fit    Down Up
##   Down   21 30
##   Up     22 31
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.5
```

(h)

```r
fit <- naiveBayes(Direction ~ Lag2, data = Weekly, subset = train)
pred <- predict(fit, Weekly[!train, ], type = "class")
(t <- table(pred, Weekly[!train, ]$Direction))
```

```
##
## pred   Down Up
##   Down    0  0
##   Up     43 61
```

```r
sum(diag(t)) / sum(t)
```

```
## [1] 0.5865385
```

(i)

Logistic regression and LDA are the best performing.

(j)

```r
fit <- glm(Direction ~ Lag1, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5673077
```

```r
fit <- glm(Direction ~ Lag3, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5865385
```

```r
fit <- glm(Direction ~ Lag1 * Lag2 * Lag3 * Lag4, data = Weekly[train, ], family = binomial)
pred <- predict(fit, Weekly[!train, ], type = "response") > 0.5
mean(ifelse(pred, "Up", "Down") == Weekly[!train, ]$Direction)
```

```
## [1] 0.5961538
```

```r
fit <- lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5769231
```

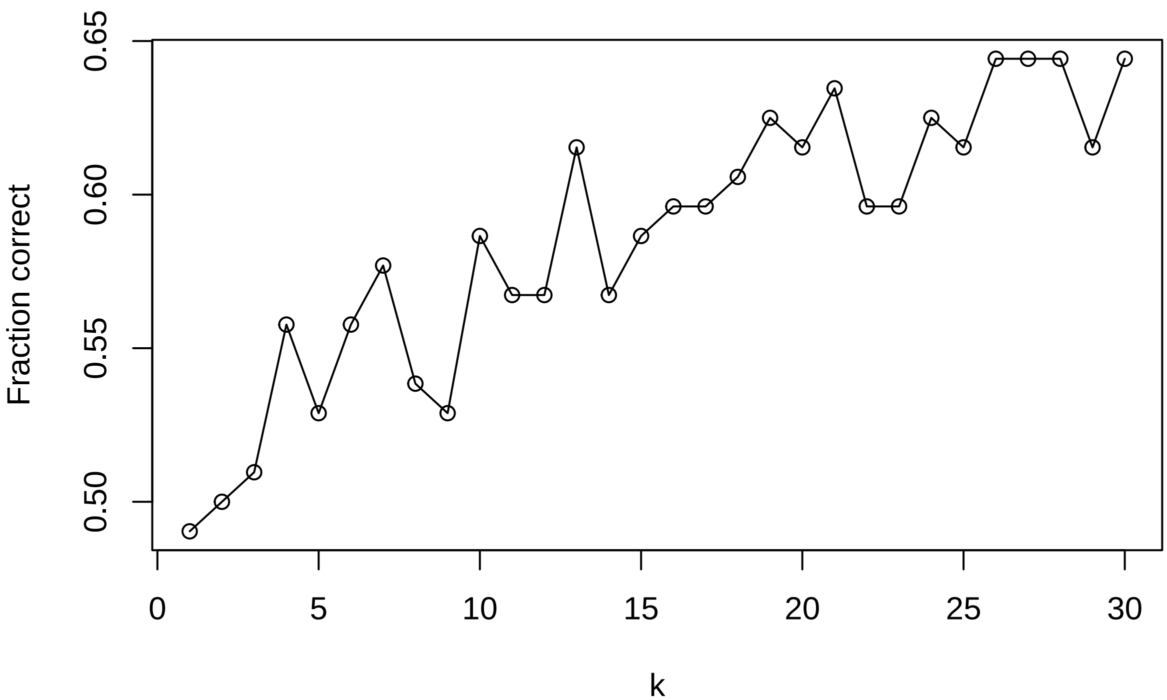
```r
fit <- qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "response")$class
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5192308
```

```r
fit <- naiveBayes(Direction ~ Lag1 + Lag2 + Lag3 + Lag4, data = Weekly[train, ])
pred <- predict(fit, Weekly[!train, ], type = "class")
mean(pred == Weekly[!train, ]$Direction)
```

```
## [1] 0.5096154
```

```r
set.seed(1)
res <- sapply(1:30, function(k) {
  fit <- knn(
    Weekly[train, 2:4, drop = FALSE],
    Weekly[!train, 2:4, drop = FALSE],
    Weekly$Direction[train],
    k = k
  )
  mean(fit == Weekly[!train, ]$Direction)
})
plot(1:30, res, type = "o", xlab = "k", ylab = "Fraction correct")
```




```r
(k <- which.max(res))
```

```
## [1] 26
```

```r
fit <- knn(
  Weekly[train, 2:4, drop = FALSE],
  Weekly[!train, 2:4, drop = FALSE],
  Weekly$Direction[train],
  k = k
)
table(fit, Weekly[!train, ]$Direction)
```

```
##
## fit    Down Up
##   Down   23 18
##   Up     20 43
```

```r
mean(fit == Weekly[!train, ]$Direction)
```

```
## [1] 0.6346154
```

KNN using the first 3 Lag variables performs marginally better than logistic regression with `Lag2` if we tune k to be $k = 26$.

15

(a)

```r
Power <- function() print(2^3)
```

(b)

```r
Power2 <- function(x, a) print(x^a)
```

(c)

```r
c(Power2(10, 3), Power2(8, 17), Power2(131, 3))
```

```
## [1] 1000
## [1] 2.2518e+15
## [1] 2248091
```

```
## [1] 1.000000e+03 2.251800e+15 2.248091e+06
```

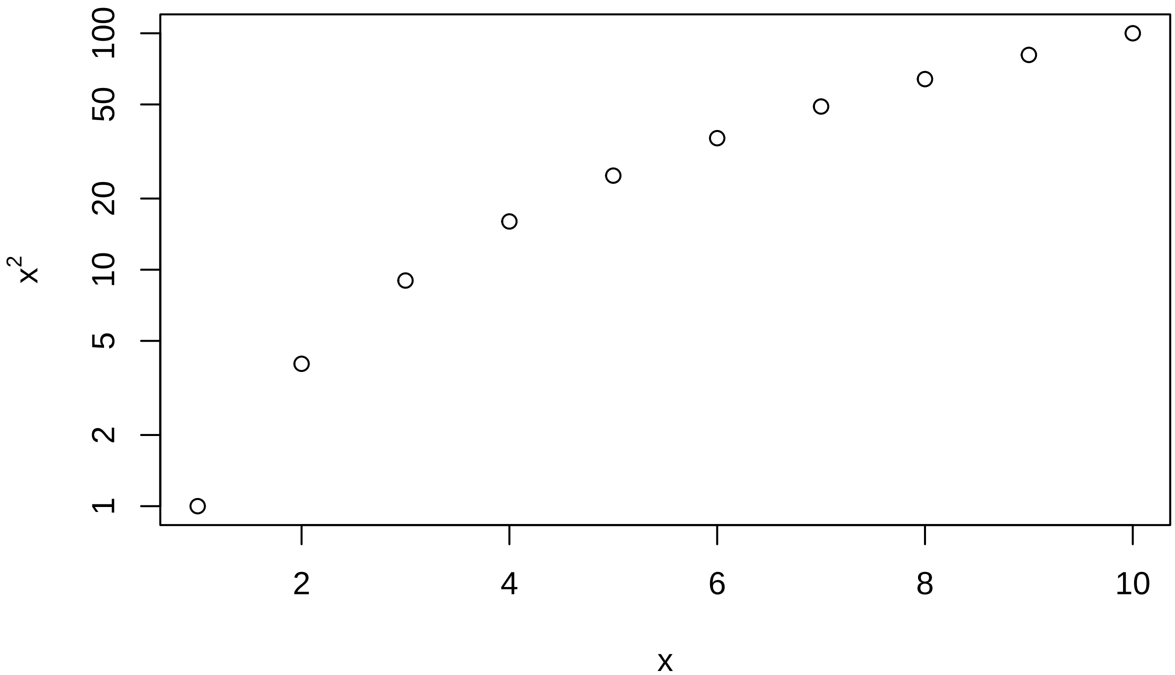
(d)

```r
Power3 <- function(x, a) {
  result <- x^a
  return(result)
}
```

(e)

```r
plot(1:10, Power3(1:10, 2),
  xlab = "x",
```

```r
  ylab = expression(paste("x"^"2")),
  log = "y"
)
```

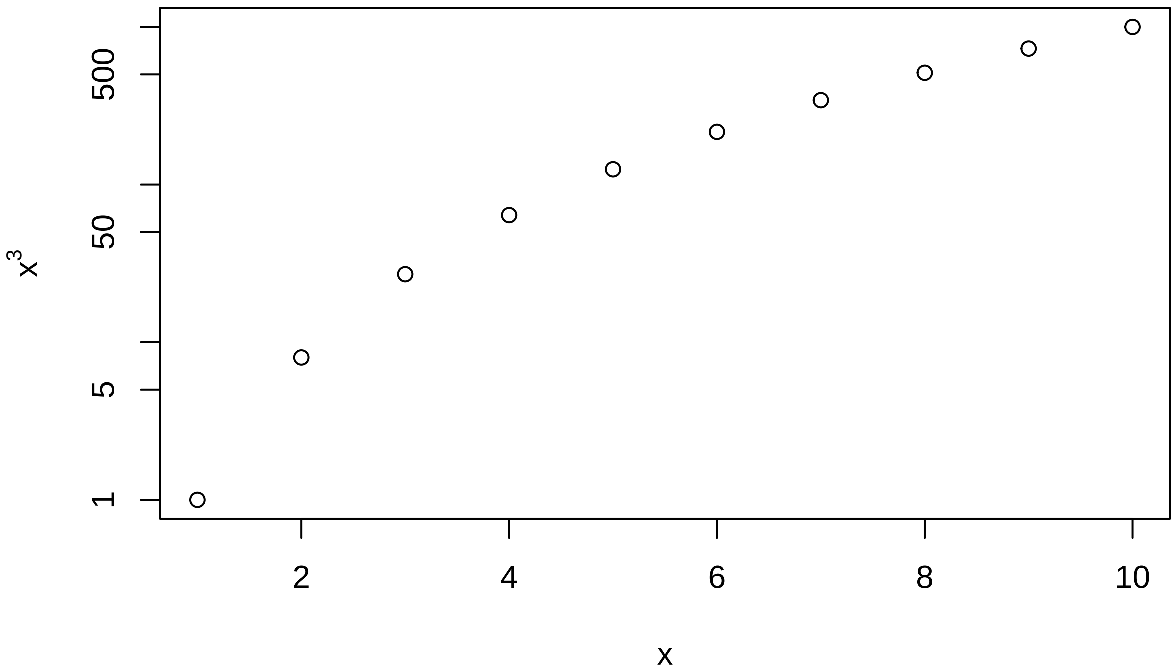


##

(f)

```r
PlotPower <- function(x, a, log = "y") {
  plot(x, Power3(x, a),
    xlab = "x",
    ylab = substitute("x"^a, list(a = a)),
    log = log
  )
}

PlotPower(1:10, 3)
```



1
5
50
500
x^3
2
4
6
8
10
x