Statistical Methods for Bioinformatics [I0U31a] Assignment 01 - Chapter 5

Hamed Borhani

April 12, 2016

1.

$$Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y) \tag{1}$$

$$Var(\alpha X) = \alpha^2 Var(X) \tag{2}$$

$$Cov(aX, bY) = abCov(X,Y) \tag{3}$$

According to these, we can infer:

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

To find the minimum of α, we should take the first derivative with respect to α and set it equal to zero:

$$\begin{aligned} &\frac{d}{d(\alpha)}(\alpha^2\sigma_X^2 + \sigma_Y^2 + \alpha^2\sigma_Y^2 - 2\alpha\sigma_Y^2 + 2\alpha\sigma_{XY} - 2\alpha^2\sigma_{XY}) = 0 \\ &\quad \Rightarrow 2\alpha\sigma_X^2 + 2\alpha\sigma_Y^2 - 2\sigma_Y^2 + 2\sigma_{XY} - 4\alpha\sigma_{XY} = 0 \\ &\quad \Rightarrow 2\alpha(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY} = 2\sigma_Y^2 - \sigma_{XY} \\ &\quad \Rightarrow \alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} \end{aligned}$$

4.

5.

```
library(ISLR)
set.seed(22)

attach(Default)

#a.
fit.lor <- glm(default ~ income + balance, family =
   "binomial")
```

```r
#b.
#i.
dim(Default)[1] #1000
training <- sample(1000, 500)
#.ii
fit.lor.val <- glm(default ~ income + balance,
   family = "binomial", subset = training)
#iii.
probabilities <- predict(fit.lor.val, newdata =
   Default[-training, ], type = "response")
predict <- rep("No", length(probabilities))
predict[probabilities > 0.5] <- "Yes"
#iv.
error <- mean(predict != Default[-training, ]$
   default)

#c.
training <- sample(1000, 500)
fit.lor.val <- glm(default ~ income + balance,
   family = "binomial", subset = training)
probabilities <- predict(fit.lor.val, newdata =
   Default[-training, ], type = "response")
predict <- rep("No", length(probabilities))
predict[probabilities > 0.5] <- "Yes"
error1 <- mean(predict != Default[-training, ]$
   default)

training <- sample(1000, 500)
fit.lor.val <- glm(default ~ income + balance,
   family = "binomial", subset = training)
probabilities <- predict(fit.lor.val, newdata =
   Default[-training, ], type = "response")
predict <- rep("No", length(probabilities))
predict[probabilities > 0.5] <- "Yes"
error2 <- mean(predict != Default[-training, ]$
   default)

training <- sample(1000, 500)
```

```
fit.lor.val <- glm(default ~ income + balance,
   family = "binomial", subset = training)
probabilities <- predict(fit.lor.val, newdata =
   Default[-training, ], type = "response")
predict <- rep("No", length(probabilities))
predict[probabilities > 0.5] <- "Yes"
error3 <- mean(predict != Default[-training, ]$
   default)
```

The test error for 3 repetitions are similar, yet varying based on sampling.

```
#d.
training <- sample(1000, 500)
fit.lor.val <- glm(default ~ income + balance +
   student, family = "binomial", subset = training)
probabilities <- predict(fit.lor.val, newdata =
   Default[-training, ], type = "response")
predict <- rep("No", length(probabilities))
predict[probabilities > 0.5] <- "Yes"
error <- mean(predict != Default[-training, ]$
   default)
```

The test error is similar to previous models, therefore adding the additional variable is not helpful.

```
#a.
fit.lor <- glm(default ~ income + balance, family =
   "binomial")
summary(fit.lor)
```

Estimates of std error :
intercept: 4.348e-01, income: 4.985e-06, balance: 2.274e-04

```
#b.
boot.fn <- function(data, index){
        return (coef(glm(default~income + balance,
           data = data, subset = index, family = "
           binomila")))
}

#c.
```

```
library(boot)
boot(Default, boot.fn, 5000)
```

8.