Statistical \_ BigData \_ LearningTheory

Dongkeon Oh

2018.9.7

#### Quesition 1

#### 1. Write out the ridge regression optimization problem in this setting.

문제의 조건을 보다 단순화시키면 아래의 식을 얻을 수 있는데,



결국 Ridge Regression의 Optimization Problem은 다음 공식을 최소화하는 것을 의미한다.



#### 2. Argue that in this setting, the ridge coefficient estimates satisfy β^1=β^2β^1=β^2.

위 Optimization Problem 식을 beta1hat과 beta2hat에 대해 미분 하고, 연산 하면 다음 2개의 식을 얻을 수 있다.





상기 두 식을 통해, beta1hat =beta2hat를 만족하는 결과를 얻을 수 있다.

#### 3. Write out the lasso optimization problem in this setting.

1번 내용과 같은 원리로 lasso의 optimization problem는 다음 공식을 최소화하는 것을 의미한다.

*y*1−*β*^1*x*1−*β*^2*x*1)2+(*y*2−*β*^1*x*2−*β*^2*x*2)2+*λ*(|*β*^1|+|*β*^2|).

#### 4. Argue that in this setting, the lasso coefficients β^1β^1 and β^2β^2 are not unique; in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.

Lasso optimization problem을 조금 수정하면,

(*y*1−*β*^1*x*1−*β*^2*x*1)2+(*y*2−*β*^1*x*2−*β*^2*x*2)2 subject to |*β*^1|+|*β*^2|≤*s*.

를 최소화하는 문제를 푼다고 할 수 있는데, 기하학적으로 보면 다이아몬드 형태의 도형을 생각할 수 있다. 다이아몬드의 중심은 (*β*^1,*β*^2) 이고, 각 꼭지점들은 중심으로부터 s만큼 떨어져 있다.

또한 *(x*11=*x*12=*x*1x11=x12=x1*, x*21=*x*22=*x*2x21=x22=x2*, x*1+*x*2=0x1+x2=0 *and y*1+*y*2=0의 조건을 활용하면, 2[*y*1−(*β*^1+*β*^2)*x*1]2≥0.2[y1−(β^1+β^2)x1]2≥0을 최소화하는 문제가 된다.

해당 문제를 풀이 하면 *β*^1+*β*^2=*y*1/*x*1을 얻을 수 있는데, 결국, 이를 기하학적으로 보았을 때 *y*1−(*β*^1+*β*^2)*x*1]2[y1−(β^1+β^2)x1]2이 다이아몬드와 만나는 점이 최적화 해답이라고 볼 수 있다.

즉, 다이아몬드의 모든 모서리인 *β*^1+*β*^2=*s*β^1+β^2=s(as is the edge *β*^1+*β*^2=−*s*β^1+β^2=−s가 실질적인 정답이다. 또한, Lasso Optimization Problem은 단 하나의 해를 가지는 것이 아니라 다음 식을 만족하는 여러 개의 해를 갖는다.

{(*β*^1,*β*^2):*β*^1+*β*^2=*s* with *β*^1,*β*^2≥0 and *β*^1+*β*^2=−*s* with *β*^1,*β*^2≤0}.{(β^1,β^2):β^1+β^2=s with β^1,β^2≥0 and β^1+β^2=−s with β^1,β^2≤0}.

#### Question 2

#### 1. Which answer is correct, and why?

c. GPA 수치가 낮을 때는 여성의 평균 임금이 남성보다 더 높지만, GPA 수치가 높을 때는 남성의 평균 임금이 더 높다. 일반적으로는 Gender의 회귀 계수가 양의 값이므로 여성의 임금 평균이 더 높지만, GPA와 Gender interaction의 회귀계수는 -10으로 음의 값을 갖기 때문에, GPA가 high enough하게 되면 오히려 남성의 임금 평균이 더 높은 경향을 보인다. 즉, GPA 점수가 높아지면 여성인 것이 임금에 negative한 영향을 끼치고, GPA가 high enough하면 남성의 임금 평균이 더 높아지게 된다.

#### 2. Predict salary for a female with an IQ of 110 and a GPA of 4.0.

50+(4.0\*20)+(110\*0.07)+35+0.01\*110\*4.0+(-10\*4) = 137.1(in thousands of dollars)

#### 3. True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

거짓이다. 계수 값이 작다고 계수 값이 유의미하지 않은 것은 아니다. GPA와 IQ가 매우 높은 상관관계를 가지고 있지만 그것이 Salary에는 작은 정도로 영향을 끼칠 수도 있기 때문이다.

#### Question 3

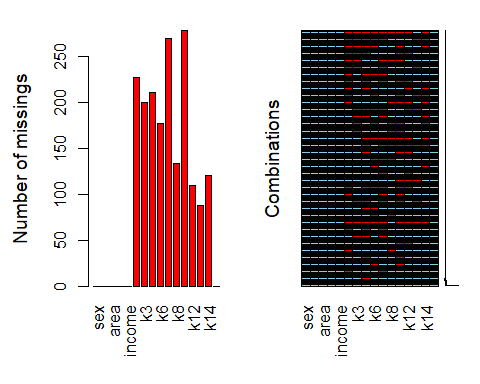
#### 3-1 데이터 처리

**기본 전처리.**

attach(Raw3)  
# 2->1 로 변환   
Raw3[,"k3"] <- ifelse(Raw3$k3 == 2, 0, Raw3$k3)  
# 순서가 없는 범주형 자료를 팩터로 변환.  
Raw3$sex <- as.factor(sex)  
Raw3$area <- as.factor(area)  
Raw3$ideo\_self <- as.factor(ideo\_self)  
# 의미가 없고, 중복된 변수 제거.  
Raw3 = Raw3[-c(1,3,4)]

**결측치 처리** **결측치 패턴 분석**

require(VIM)  
aggr(Raw3,prop=FALSE,numbers=TRUE)



**결측치들간의 상관관계 파악.**

x=as.data.frame(abs(is.na(Raw3)))  
y=apply(x,2,function(x) sum(x)>0)  
round(cor(x[y]),2)

## k2 k3 k4 k6 k7 k8 k10 k12 k13 k14  
## k2 1.00 0.32 0.36 0.37 0.23 0.22 0.35 0.27 0.18 0.23  
## k3 0.32 1.00 0.39 0.40 0.28 0.22 0.31 0.30 0.26 0.24  
## k4 0.36 0.39 1.00 0.45 0.29 0.21 0.37 0.34 0.25 0.30  
## k6 0.37 0.40 0.45 1.00 0.28 0.27 0.41 0.35 0.24 0.34  
## k7 0.23 0.28 0.29 0.28 1.00 0.24 0.31 0.27 0.29 0.20  
## k8 0.22 0.22 0.21 0.27 0.24 1.00 0.26 0.30 0.26 0.30  
## k10 0.35 0.31 0.37 0.41 0.31 0.26 1.00 0.37 0.26 0.29  
## k12 0.27 0.30 0.34 0.35 0.27 0.30 0.37 1.00 0.38 0.40  
## k13 0.18 0.26 0.25 0.24 0.29 0.26 0.26 0.38 1.00 0.38  
## k14 0.23 0.24 0.30 0.34 0.20 0.30 0.29 0.40 0.38 1.00

결측치들 간의 특별한 패턴이나 높은 상관성이 있지 않다. (y와 결측치 들의 상관관계를 보는 것도 중요한데, 서로 상관 없으면 결측치를 그냥 빼면 될 것이고 상관관계가 파악 되면 적절한 추정 또는 모델을 통해 결측치를 처리해야할 것이다. 예를 들어 만일 ideo\_self가 10일 때, k4가 높아지는 경향이 있으면 결측치를 추정 해야하고, k5의 결측치가 y와 다른 변수와 무관하면 제거해도 될 것이다.)

**mice패키지를 활용한 결측치 추정.**

library(mice)  
miceMod <- mice(Raw3, method="logreg")

Raw3 <- complete(miceMod)

0과1 사이에서 결측치의 값을 추정하는 문제이기 logistic method를 통해 결측치를 추정했다.

#### random forest 모형 적합

library(plyr)  
library(dplyr)  
library(randomForest)  
  
data <- Raw3  
  
#cross validation, using rf to predict ideo\_self  
k = 10  
set.seed(1)  
data$id <- sample(1:k, nrow(data), replace = TRUE)  
list <- 1:k  
  
#데이터 프레임 초기화  
prediction <- data.frame()  
testsetCopy <- data.frame()  
  
#function for k fold  
#i는 1부터 10로 나눈후에 10번을 진행하도록 합니다.  
  
for(i in 1:k){  
 trainingset <- subset(data, id %in% list[-i])  
 testset <- subset(data, id %in% c(i))  
 #데이터를 10등분하고 한번 뽑은 test data가 다시 train 으로 가지 않도록 10등분 합니다.  
  
 #run a random forest model  
 mymodel <- randomForest(trainingset$ideo\_self ~ ., data = trainingset,  
 ntree = 200, importance = T)  
 #값을 저장한다.   
 temp <- as.data.frame(predict(mymodel, testset[,-16]))  
 prediction <- rbind(prediction, temp)  
 testsetCopy <- rbind(testsetCopy, as.data.frame(testset[,16]))  
  
}  
  
# add predictions and actual ideo\_self  
result <- cbind(prediction, testsetCopy[, 1])  
names(result) <- c("Predicted", "Actual")

#### 혼동행렬

library(caret)  
confusionMatrix = confusionMatrix(result$Predicted, result$Actual)  
confusionMatrix$table

## Reference  
## Prediction 0 1 2 3 4 5 6 7 8 9 10  
## 0 0 0 1 0 1 0 1 0 0 0 1  
## 1 1 1 0 1 3 2 0 0 0 0 0  
## 2 1 1 0 3 2 0 1 1 0 0 0  
## 3 5 2 10 14 12 31 7 4 3 0 2  
## 4 2 4 6 11 9 13 5 5 4 1 0  
## 5 27 18 26 83 77 234 69 48 28 9 29  
## 6 1 1 2 3 2 13 12 16 6 3 6  
## 7 0 0 1 2 6 11 17 14 12 3 10  
## 8 0 0 0 1 1 9 4 9 12 7 4  
## 9 0 0 0 0 1 0 2 1 4 0 0  
## 10 0 0 0 1 2 10 3 8 4 2 9

confusionMatrix$overall[1]

## Accuracy   
## 0.2893738

confusionMatrix$overall[3:4]

## AccuracyLower AccuracyUpper   
## 0.2621428 0.3177864

##### **결과 해석**

err = abs((as.numeric(result$Predicted)- as.numeric(result$Actual)))  
table(err)

## err  
## 0 1 2 3 4 5 6 7 10   
## 305 277 215 119 55 75 4 3 1

결과값을 해석해보면, ideo\_self의 오차를 1~2까지는 허용하면 정확도가 상당히 올라간다. 또한 혼동행렬이5일 때의 오차가 굉장히 높은 것을 확인할 수 있다. 때문에 ideo\_self가 5인 경우에 한해서만 다른 모델을 적합 한다면 더 나은 성능을 얻을 수도 있을 것이다.

table(result$Predicted )

##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 4 8 9 90 60 648 65 76 47 8 39

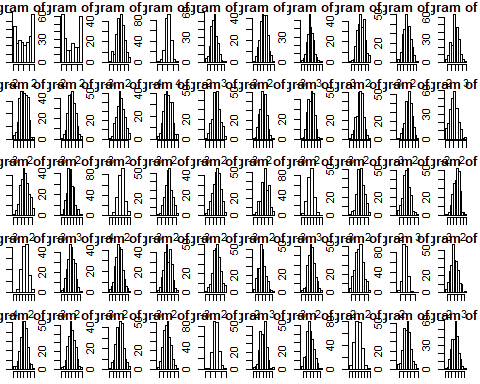
table(result$Actual)

##   
## 0 1 2 3 4 5 6 7 8 9 10   
## 37 27 46 119 116 323 121 106 73 25 61

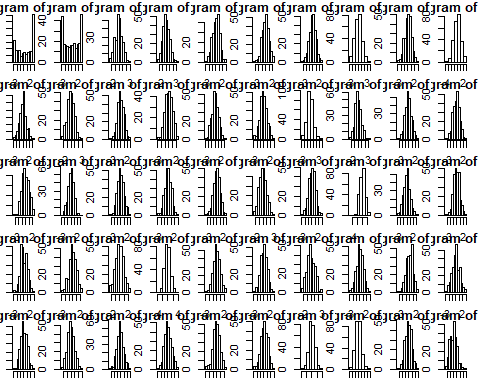
예측 결과 실제결과를 비교하면, 실제 자료에 비해 예측 결과는 보수적인 쪽으로 예측을 더 많이 했다는 것을 확인할 수 있다.

#### Question 4

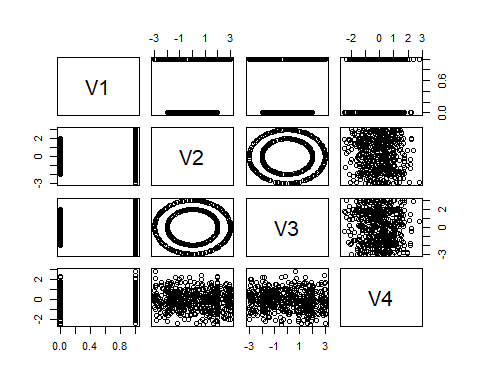
Raw4 = read\_excel("C:/Users/renz/Desktop/임요한 기말/data4.xlsx", sheet = 2, skip = 1)  
Raw4 = Raw4[,-1]  
attach(Raw4)  
Raw4\_1 <- Raw4 %>% filter(V1 == 1)  
Raw4\_0 <- Raw4 %>% filter(V1 == 0)  
par(mar= c(1,1,1,1))  
par(mfrow = c(5,10))  
for (i in 2:51){  
temp = unlist(Raw4\_0[,i])  
hist(temp)  
}



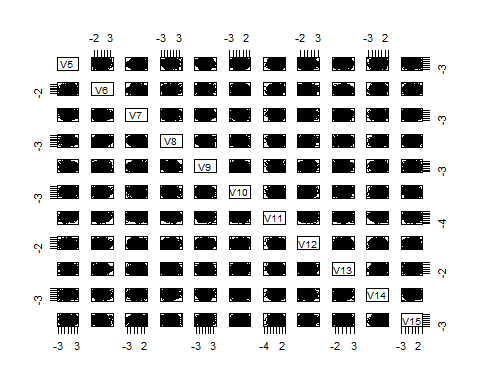
for (i in 2:51){  
temp = unlist(Raw4\_1[,i])  
hist(temp)  
}



par(mfrow = c(1,1))  
pairs(Raw4[1:4])

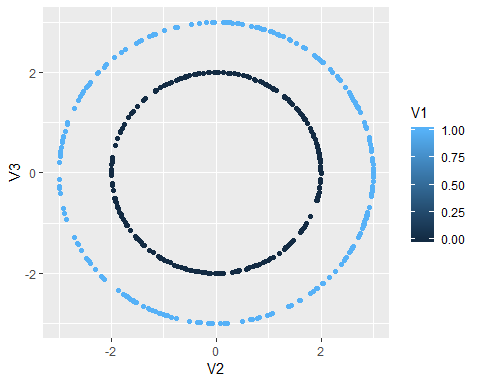


pairs(Raw4[5:15])



산점도를 통해, V1과 V2,V3 간에 완벽한 함수관계가 있는 것을 확인할 수 있다. 또한 나머지 변수들의 모두 정규분포를 따르고 서로 상관도 없다.

ggplot(Raw4, aes(V2, V3, color = V1)) +  
 geom\_point()



V2^2 + V3^3 = 4 일 때는 V1이 0이고, V2^2 + V3^3 = 9 일 때는 V1이 1이다. 이는 간단한 알고리즘을 통해 확인할 수 있다.

pred <- ifelse(round(((Raw4$V2)^2 + (Raw4$V3)^2)) == 4, 0, 1)   
tab <- table(pred, Raw4$V1)  
tab

##   
## pred 0 1  
## 0 250 0  
## 1 0 250

250씩 클러스터에 따라 정확하게 예측하는 것을 확인해볼 수 있다.

#### 4-1 군집분석

##### **kmeans**

set.seed(1)

Raw4\_kmean = Raw4[-1]  
km.out = kmeans(Raw4\_kmean,2,nstart = 20)  
km.out$cluster = ifelse(km.out$cluster == 1,0,1)  
Actual = Raw4[1]  
Predicted= km.out$cluster  
temp = cbind(Predicted, Actual)

table(temp$Predicted ,temp$V1)

##   
## 0 1  
## 0 122 106  
## 1 128 144

k-means의 성능이 안좋다. 이는 v2와 v3로 결정되는 V1의 분포가 원형이고 중첩되어 있기 때문에 발생한다.예를 들어 k-means의 시작점이 어느 점에 주어지더라도 원으로된 클러스터를 파악하지 못하고 절반을 나눈다. 따라서 예측된 값의 성능이 50퍼센트로 주어지는 것이다. 때문에 k-means는 부적절하다. 이에 대한 대안으로 v2와 v3의 제곱을 변수로 두고 svm을 돌리거나 다른 비지도 학습을 이용해 볼 수 있을 것이다. (선형모델을 사용하더라도 변수를 2차항을 두면 좋은 결과를 얻을 수 있을까?)

#### 4-2 QDA

data <- Raw4[c(1,2,3)]  
data$V1 <- as.factor(V1)  
  
k = 10  
set.seed(1)  
data$id <- sample(1:k, nrow(data), replace = TRUE)  
list <- 1:k  
  
prediction <- data.frame()  
testsetCopy <- data.frame()  
  
for(i in 1:k){  
  
 trainingset <- subset(data, id %in% list[-i])  
 testset <- subset(data, id %in% c(i))  
  
 mymodel <- qda(trainingset$V1 ~ ., data = trainingset)  
  
 temp <- as.data.frame(predict(mymodel, testset[,-1]))  
 prediction <- rbind(prediction, temp)  
 testsetCopy <- rbind(testsetCopy, as.data.frame(testset[,1]))  
}

##### **혼동행렬**

library(caret)  
result <- cbind(prediction, testsetCopy[, 1])  
names(result) <- c("Predicted","posterior.0","posterior.1", "Actual")

## [1] "Predicted" "posterior.0" "posterior.1" "Actual"

result$Actual<- as.factor(result$Actual)   
confusionMatrix(result$Predicted, result$Actual)

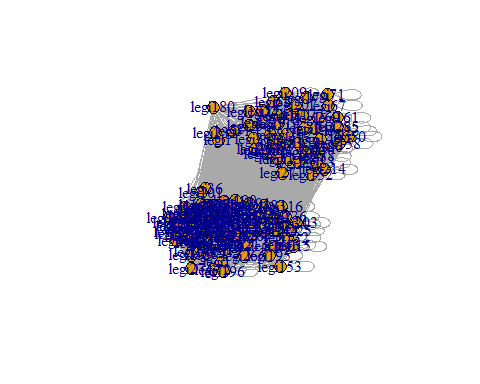
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 220 28  
## 1 30 222  
##   
## Accuracy : 0.884   
## 95% CI : (0.8526, 0.9107)  
## No Information Rate : 0.5   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.768   
## Mcnemar's Test P-Value : 0.8955   
##   
## Sensitivity : 0.8800   
## Specificity : 0.8880   
## Pos Pred Value : 0.8871   
## Neg Pred Value : 0.8810   
## Prevalence : 0.5000   
## Detection Rate : 0.4400   
## Detection Prevalence : 0.4960   
## Balanced Accuracy : 0.8840   
##   
## 'Positive' Class : 0   
##

두 집단의 결정 경계가 선형이 아니므로 QDA를 적합한다. 예상대로 높은 예측 률(0.884)이 확인된다. (LDA를 통해서도 설명변수의 유연성을 증가시켜서 동일한 성능을 낼 수 있을까?)

#### Question 5

##### **igraph을 활용한 클러스터링**

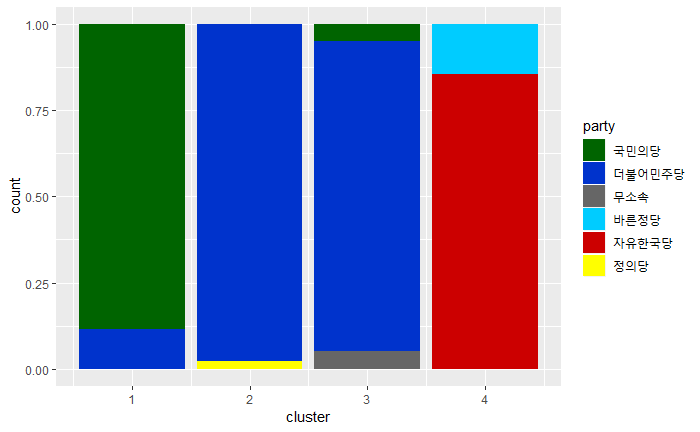
library(igraph)  
Initial.matrix <- read.csv('C:/Users/renz/Desktop/임요한 기말/data5.csv',  
 header=TRUE, row.names=1, check.names=FALSE, na.strings = "")  
matrix <- as.matrix(Initial.matrix)   
g <- graph.adjacency(matrix, mode="undirected", weighted=T)  
plot.igraph(g)



a <- cluster\_fast\_greedy(g, weights = E(g)$weight)  
index = a$membership  
index\_party <- read\_excel("C:/Users/renz/Desktop/임요한 기말/data5.xlsx",   
 sheet = 2)  
index\_party$cluster <- index

igraph를 통해, 각 관측치 간의 연결의 강도를 파악해서 네트워크를 구성하고 이를 바탕으로 군집을 찾는다. 공동 발의된 법안의 수를 바탕으로 의원들간의 군집을 파악한 결과는 아래와 같다.

g <- ggplot(index\_party, aes(cluster))  
g + geom\_bar() g + geom\_bar(aes(fill = party)) + scale\_fill\_manual(values = c('darkgreen', "#0033CC", "#666666", "#00CCFF", "#CC0000", '#FFFF00'))



국민의당은 주로 1집단에 포함되고, 더불어민주당은 2집단과 3집단에 주로 분포하며, 바른정당과 자유한국당은 4집단에 많이 분포한다. 이를 통해, 국민의당과 더불어민주당은 서로 다른 집단으로 확실히 구분될 만큼 공통 발의된 법안이 서로 적다는 것을 알 수 있고, 바른정당과 자유한국당은 서로 같은 집단으로 구분될 정도로 사실은 공통 발의된 법안도 많고 서로 간의 교류가 많다는 것을 확인할 수 있다. 또한 더불어 민주당이 크게 두가지 집단으로 분리되는 점이 특징적인데, 이를 통해 더불어 민주당 사이에서도 서로 다른 집단으로 분류할 만큼 계파간의 차이가 존재한다는 점을 확인해볼 수 있다.