Model selection.rmd

Model selection

Goal1. Forward sepwise selection

Goal2. Logistic regression

Goal3. Cross-validation

Data Loading and Setting

데이터 로딩

```
DATA <- read.table("KMOOC_2_04_dataset_iris.txt", header=TRUE)
```

데이터 분리

```
train.idx <- c(1:30, 51:80)
data.train <- DATA[train.idx,]
data.test <- DATA[-train.idx,]
```

Define Function

Logistic Regression 모델 생성 및 예측 함수 정의

```
my.lr <- function(formula,data,newdata) {
  f <- glm(formula,family=binomial(link='logit'),data=data)
  y <- predict(f,newdata=data,type='response')
  y.train <- factor(y>0.5,levels=c(FALSE,TRUE),labels=c('versicolor','virginica'))
  y <- predict(f,newdata=newdata,type='response')
  y.test <- factor(y>0.5,levels=c(FALSE,TRUE),labels=c('versicolor','virginica'))
  err.train <- mean(y.train!=data$Species)
  err.test <- mean(y.test!=newdata$Species)
  return( list(f=f,y.train=y.train,y.test=y.test,err.train=err.train,err.test=err.test))
}
```

LOOCV 함수 정의

```
my.loocv <- function(YIDX,XIDX,data) {
  cn <- colnames(data)
  str1 <- paste(cn[XIDX],collapse="+")
  str2 <- paste(cn[YIDX],str1,sep="~")
  formula <- as.formula(str2)
```

```
  yhat <- factor(rep(TRUE,nrow(data)),levels=c(FALSE,TRUE),labels=c('versicolor','virginica'))
  err <- rep(0,nrow(data))
  for( i in 1:nrow(data) ) {
    f <- my.lr(formula,data[-i,],data[i,])
    yhat[i] <- f$y.test
    err[i] <- f$err.test
  }
  return( list(formula=formula,y.cv=yhat,err.cv=err) )
}
```

Example of CV and Validation test

```
f.cv <- my.loocv(5,c(1,3),data.train)
mean(f.cv$err.cv) # cv error
```

```
## [1] 0.2166667
```

```
f.val <- my.lr(f.cv$formula,data.train,data.test)
f.val$err.train
```

```
## [1] 0.2166667
```

```
f.val$err.test
```

```
## [1] 0.225
```

Forward stepwise selection

전진 선택법 구현

```
best.model <- NULL
err.cv <- NULL
xidx.set <- 1:4
for( k in 1:length(xidx.set) ) {
  tmp.err <- NULL
  for( i in xidx.set ) {
    xidx <- c(best.model,i)
    f <- my.loocv(5,xidx,data.train)
    tmp.err <- c(tmp.err,mean(f$err.cv))
  }
  best.model <- c(best.model,xidx.set[which.min(tmp.err)])
  err.cv <- c(err.cv,min(tmp.err))
  xidx.set <- setdiff(xidx.set,best.model)
}
```

유용한 변수의 나열과 변수의 갯수에 따른 CV 결과 확인

```
best.model
```

```
## [1] 3 4 2 1
```

```
err.cv
```

```
## [1] 0.2166667 0.1833333 0.1833333 0.2166667
```

변수의 갯수가 다른 모델의 train과 test의 오류율 확인

```
err.test <- rep(0,4)
err.train <- rep(0,4)
for( i in 1:length(best.model) ) {
  cn <- colnames(data.train)
  str1 <- paste(cn[best.model[1:i]],collapse="+")
  str2 <- paste(cn[5],str1,sep="~")
  formula <- as.formula(str2)
  f <- my.lr(formula,data.train,data.test)
  err.train[i] <- f$err.train
  err.test[i] <- f$err.test
}
err.train
```

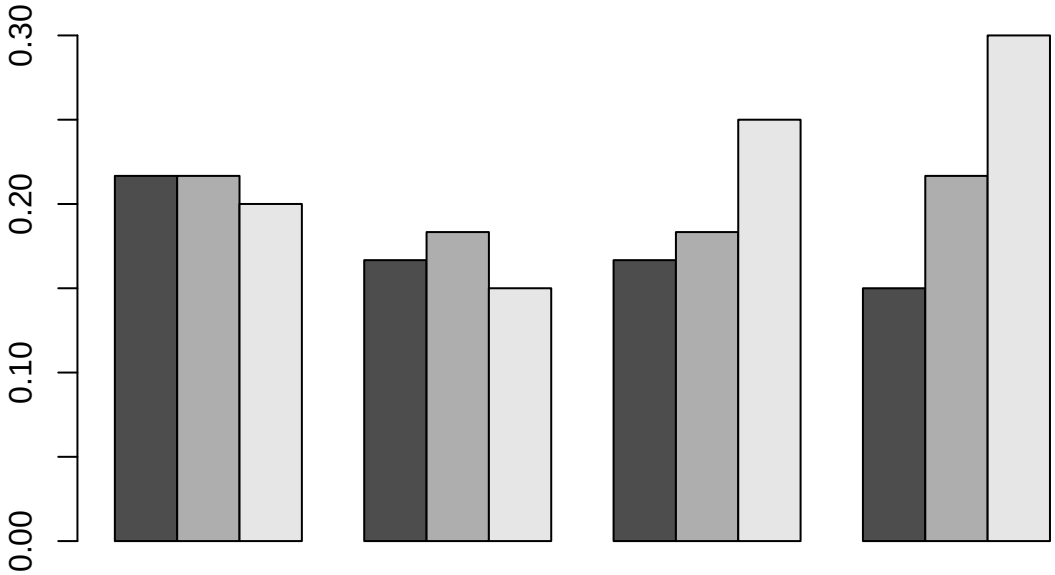
```
## [1] 0.2166667 0.1666667 0.1666667 0.1500000
```

```
err.test
```

```
## [1] 0.20 0.15 0.25 0.30
```

train, CV, test의 오류율 시각화

```
err.mat <- rbind(err.train,err.cv,err.test)
barplot(err.mat,beside=TRUE)
```


0.00
0.10
0.20
0.30