Weekly Summary 3

Seline Yang

2/12/2018

BSS (Best Subset Selection)

Purpose: To find the best model.

Summary: Fit seperate least squares regression for every subset of predictors. If we have p predictos, then we will have 2^p models to fit to run best subset selection. M_0 denote null model. For $k = 1, ...p$, fit all $\binom{p}{k}$ models containing exactly k predictors. Use the smallest RSS or largest R^2 find the best among call it M_k. Finally select a single best model from $M_0, ...M_p$ using either cross validation, C_p, AIC, BIC, R^2_{adj}.

Pros: Carefully screening all the possible models.

Cons: We have to choose the correct metric to run and choose the best model. It there are lot of predictors, it will take a very long time! The most predictors always have the highest $R^2$and lowest RSS, Adjustment need to made to corect for the bias due to overfitting.

R Commands:

```
library(leaps)
BSS <- regsubsets(CO~.,cig07)
plot(BSS,scale = "adjr2")
BestAdjR2= which.max(summary(BSS)$adjr2)
coef(BSS,BestAdjR2)
```

Mallow's C_p/AIC

Purpose: To determine which subset of possible predictors we want to use in our model.

Summary:

$$C_p = \frac{1}{n}(RSS + 2k\hat{\sigma}^2)$$

$$AIC \propto \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

$\hat{\sigma}^2$ is an estimate of variance of ϵ obtained on the FULL model. C_p adds a penalty to the training RSS. The panalty adjusts fot the fact that training error tends to underestimate the test error. We prefer lower C_p values. AIC is deternmined by maximizing the likelihood of a given model class. C_p and AIC are proportional to one another.

Pros: Help us deal with overfitting. As the model becomes more flxible, the panalty increase. C_p is an unbiased estimate of the test MSE. In AIC, all models are treated symmetricall, it can be used to compare nested as well as non-nested models.

Cons: The C_p approximation and AIC are only valid for large sample size, it cannot handle complex colloections of models as in the feature selection problem.

R Commands:

```
plot(BSS,scale = "Cp")
BestCp= which.min(summary(BSS)$cp)
coef(BSS,BestCp)
```

BIC

Purpose: To determine which subset of possible predictors we want to use in our model.

Summary:

$$BIC \propto \frac{1}{n\hat{\sigma}^2}(RSS + dlog(n)\hat{\sigma}^2)$$

BIC penalty is generally stronger than C_p. BIC tend to select a smaller model.

Pros: BIC can help us effectively reduce the number of predicter due to overfitting.

Cons: It is only valid for sample size much larger than the number k if parameters in the model. BIC cannot handle complex collections of models as in the feature selection problem in high-dimension.

R Commands:

```
plot(BSS,scale = "bic")
BestBIC= which.min(summary(BSS)$bic)
coef(BSS,BestBIC)
```

Subset Selection with CV (Validation Method)

Purpose: Estimating test error for each model selected from subset selection. Help to select the best model.

Summary: The first step in this method is to divide the original data into training set and validation set. We use the training set to fit the model, apply best subset selection to CV training data. Then use validation set to estimate test MSE for each of these models, choose the model with smallest test MSE.

Pros: Effectively be used to compare the perpormances of different predictive models.

Cons: If the full data set is used to perform the best subset selection step, the validation set errors and cross-validation errors that we obtain will not be accurate estimates of the test error. Variables that chosen in the training set may differ from using full data.

R Commands:

```
regfit.best <- regsubsets(CO~., cig07[train,],nvmax = 10)
# Create the model matrix
test.mat <- model.matrix(CO~., cig07[-train,])
# Create a space to store the estimated test MSE (the valdiation MSE for each model)
val.mse <- rep(NA, 10)
# Run the for loop, looping over the number of predictors
for( i in 1:10){
  # (1)
  coefi <-coef(regfit.best,id = i)
  # (2)
  pred <- test.mat[,names(coefi)]%*%coefi
  # (3)
  val.mse[i] <- mean(cig07$CO[-train] - pred)^2
}
#to get the actual best model in terms of validation error, we need to find the one
#with the smallest MSE.
min(val.mse); which.min(val.mse)
# We are now going to select the best 7 variable model using the original training data
regfit.best.full <- regsubsets(CO~., cig07,nvmax = 10)
coef(regfit.best.full,7)
```

Subset Selection with CV (k-fold method)

Purpose: Estimating test error for each model selected from subset selection. Help to select the best model.

Summary: The dataset is randomly divided into k groups of approximately equal size. Setting fold 1 to be the validation set, and the remaining as the training data. Repeat the process with remaining k-1 folds. Compute MSE for each process.

Pros: Lower bias compare to validation method. All observations are used for both training and validation, and each observation is used for validation exactly once.

Cons: If the prediction method is expensive to train, cross-validation can be very slow since the training must be carried out repeatedly.

R Commands:

```
k = 10
set.seed(120)
folds <- sample(rep(1:k,each= 123),nrow(cig07), replace = FALSE)
# Make storage space
cv.errors <- matrix(NA, k ,10)

# First loop: Loop over each fold, i.e., allow each fold to be the validation set
for(j in 1:k){
  # Choose fold j
  # Apply BSS to the "training" data, i.e., all folds except fold j
  regfit.best <- regsubsets(CO~., cig07[folds!=j,],nvmax = 10)
  # Create the model matrix for fold j
  test.mat <- model.matrix(CO~., cig07[folds==j,])
  # Second loop: loop over the best model with 2, 3, ..., 10 predictors,
  # as selected on the "training" set, i.e., without fold j
  val.errors <-rep(NA, 10)
  for( i in 1:10){
    #(1)
    coefi <-coef(regfit.best,id = i)
    #(2) Predict for fold j
    pred <- test.mat[,names(coefi)]%*%coefi
    #(3) Compute the validation error for fold j
    cv.errors[i,j] <- mean(cig07$CO[folds==j] - pred)^2
  }
}
# Now, we work with the matrix we have
mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors
#find the one with minimum error
min(mean.cv.errors); which.min(mean.cv.errors)
#we run best subset selection on the full data set to obtain the 8 variable model.
reg.best.full.cv <-regsubsets(CO~., cig07,nvmax = 10)
coef(reg.best.full.cv, 8)
```