

STAT534 HW2

Bryan Ng

2025-04-13

Problem 1

```
path = "C:/Users/ncwbr/Desktop/534binarydata.txt"
data = as.matrix(read.table(path, header = FALSE))
response <- 61
```

```
getLogisticAIC <- function(response, explanatory, data) {
  if (0 == length(explanatory)) {
    deviance = glm(data[, response] ~ 1,
                    family = binomial(link = "logit"))$deviance
  }
  else {
    deviance = glm(data[, response] ~ as.matrix(data[, as.numeric(explanatory)]),
                    family = binomial(link = "logit"))$deviance
  }
  return(deviance + 2 * (1 + length(explanatory)))
}
```

Problem 2

```
forwardSearchAIC <- function(response, data, lastPredictor = c()) {
  candidate <- setdiff(seq_len(ncol(data)), c(lastPredictor, response))
  currentAIC <- getLogisticAIC(response, lastPredictor, data)
  if (length(candidate) == 0) {
    return(list(Selected_column = lastPredictor, AIC = currentAIC))
  }
  bestAIC <- currentAIC
  bestCandidate <- NULL
  for (i in candidate) {
    newPredictors <- c(lastPredictor, i)
    newAIC <- getLogisticAIC(response, newPredictors, data)
    if (newAIC < bestAIC) {
      bestAIC <- newAIC
      bestCandidate <- i
    }
  }
  if (!is.null(bestCandidate)) {
    NextPredictor <- c(lastPredictor, bestCandidate)
  }
}
```

```

    return(forwardSearchAIC(response, data, NextPredictor))
  }
  else {
    return(list(Selected_column = lastPredictor, AIC = currentAIC))
  }
}
forwardSearchAIC(response, data, lastPredictor = c())

```

```

## $Selected_column
## [1] 23 3 22 21 49 4 20 34 53 46
##
## $AIC
## [1] 22

```

Problem 3

```

backwardSearchAIC <- function(response, data, lastPredictor = setdiff(seq_len(ncol(data)), response)) {
  candidate <- lastPredictor
  currentAIC <- getLogisticAIC(response, lastPredictor, data)
  if (length(candidate) == 0) {
    return(list(Selected_column = lastPredictor, AIC = currentAIC))
  }
  bestAIC <- currentAIC
  bestCandidate <- NULL
  for (i in candidate) {
    newPredictors <- setdiff(lastPredictor, i)
    newAIC <- getLogisticAIC(response, newPredictors, data)
    if (newAIC < bestAIC) {
      bestAIC <- newAIC
      bestCandidate <- i
    }
  }
  if (!is.null(bestCandidate)) {
    NextPredictor <- setdiff(lastPredictor, bestCandidate)
    return(backwardSearchAIC(response, data, NextPredictor))
  }
  else {
    return(list(Selected_column = lastPredictor, AIC = currentAIC))
  }
}
backwardSearchAIC(response, data, lastPredictor = setdiff(seq_len(ncol(data)), response))

```

```

## $Selected_column
## [1] 1 3 9 12 20 23 25 34 41 46
##
## $AIC
## [1] 22

```

As an observation, although the logistic regression models obtained via forward search and backward search are not identical, both models achieve the same minimum AIC value.

Problem 4

```
getLogisticBIC <- function(response, explanatory, data) {
  if (0 == length(explanatory)) {
    deviance = glm(data[, response] ~ 1,
                    family = binomial(link = "logit"))$deviance
  }
  else {
    deviance = glm(data[, response] ~ as.matrix(data[, as.numeric(explanatory)]),
                    family = binomial(link = "logit"))$deviance
  }
  return(deviance + log(nrow(data)) * (1 + length(explanatory)))
}

forwardSearchBIC <- function(response, data, lastPredictor = c()) {
  candidate <- setdiff(seq_len(ncol(data)), c(lastPredictor, response))
  currentBIC <- getLogisticBIC(response, lastPredictor, data)
  if (length(candidate) == 0) {
    return(list(Selected_column = lastPredictor, BIC = currentBIC))
  }
  bestBIC <- currentBIC
  bestCandidate <- NULL
  for (i in candidate) {
    newPredictors <- c(lastPredictor, i)
    newBIC <- getLogisticBIC(response, newPredictors, data)
    if (newBIC < bestBIC) {
      bestBIC <- newBIC
      bestCandidate <- i
    }
  }
  if (!is.null(bestCandidate)) {
    NextPredictor <- c(lastPredictor, bestCandidate)
    return(forwardSearchBIC(response, data, NextPredictor))
  }
  else {
    return(list(Selected_column = lastPredictor, BIC = currentBIC))
  }
}

backwardSearchBIC <- function(response, data, lastPredictor = setdiff(seq_len(ncol(data)), response)) {
  candidate <- lastPredictor
  currentBIC <- getLogisticBIC(response, lastPredictor, data)
  if (length(candidate) == 0) {
    return(list(Selected_column = lastPredictor, BIC = currentBIC))
  }
  bestBIC <- currentBIC
  bestCandidate <- NULL
  for (i in candidate) {
    newPredictors <- setdiff(lastPredictor, i)
    newBIC <- getLogisticBIC(response, newPredictors, data)
    if (newBIC < bestBIC) {
      bestBIC <- newBIC
      bestCandidate <- i
    }
  }
}
```

```

    }
  }
  if (!is.null(bestCandidate)) {
    NextPredictor <- setdiff(lastPredictor, bestCandidate)
    return(backwardSearchBIC(response, data, NextPredictor))
  }
  else {
    return(list(Selected_column = lastPredictor, BIC = currentBIC))
  }
}
forwardSearchBIC(response, data, lastPredictor = c())

```

```

## $Selected_column
## [1] 23 3 22 21 49 4 20 34 53 46
##
## $BIC
## [1] 54.96934

```

```
backwardSearchBIC(response, data, lastPredictor = setdiff(seq_len(ncol(data)), response))
```

```

## $Selected_column
## [1] 1 3 9 12 20 23 25 34 41 46
##
## $BIC
## [1] 54.96934

```

The logistic regression model derived from forward search under the BIC criterion is identical to that obtained under the AIC criterion, and the logistic regression model derived from backward search under the BIC criterion is also identical to that obtained under the AIC criterion.