Week 4 AHA: Regularization

2024-02-21

Install packages

```r
# Install packages
install.packages("glmnet")
```

Load packages (and set seed)

```r
# Load packages
library(glmnet); library(caret)
set.seed(42)
```

Load data

```r
# Load data
load("../data/ncds/ncds_sample.RData")
load("../data/ncds/ncds_test.RData")
```

Data wrangling

```r
# Set up our X and Ys
X <- as.matrix(ncds_sample[,2:51]) # 50 personality items
Y <- as.matrix(ncds_sample[,"malaise_group"]) # well-being

# Update X and Y
keep <- complete.cases(X) & !is.na(Y)
X <- X[keep,]; Y <- Y[keep,, drop = FALSE]

# Do the same for the test group
X_test <- as.matrix(ncds_test[,2:51]) # 50 personality items
Y_test <- as.matrix(ncds_test[,"malaise_group"]) # well-being

# Update X and Y
keep_test <- complete.cases(X_test) & !is.na(Y_test)
X_test <- X_test[keep_test,]; Y_test <- Y_test[keep_test,, drop = FALSE]
```

Perform logistic regression

```r
# Set up data frame
df_train <- cbind.data.frame(Y, X)
df_test <- cbind.data.frame(Y_test, X_test)

# Train logistic regression model
logm <- glm(Y ~ ., data = df_train, family = "binomial")
```

```r
# Predict testing data
prediction <- factor(
  ifelse(
    predict(logm, newdata = df_test, type = "response") > 0.50,
    1, 0
  ), levels = c(0, 1)
)

# Get evaluation metrics
confusionMatrix(
  data = prediction,
  reference = factor(Y_test),
  positive = "1"
)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 224  25
         1  24  15

               Accuracy : 0.8299
                 95% CI : (0.7814, 0.8714)
    No Information Rate : 0.8611
    P-Value [Acc > NIR] : 0.944

                  Kappa : 0.2812

 Mcnemar's Test P-Value : 1.000

            Sensitivity : 0.37500
            Specificity : 0.90323
         Pos Pred Value : 0.38462
         Neg Pred Value : 0.89960
             Prevalence : 0.13889
         Detection Rate : 0.05208
   Detection Prevalence : 0.13542
      Balanced Accuracy : 0.63911

       'Positive' Class : 1
```

Perform ridge logistic regression

```r
# Perform CV for ridge
cv_ridge <- cv.glmnet(
  x = X, y = Y,
  family = "binomial", # logistic
  alpha = 0 # 0 = ridge
)

# Use best lambda
```

```r
ridge_glm <- glmnet(
   x = X, y = Y,
  family = "binomial", # logistic
  alpha = 0, # 0 = ridge
  lambda = cv_ridge$lambda.min
)

# Predict testing data
prediction <- factor(
  ifelse(
    predict(ridge_glm, newx = X_test, type = "response") > 0.50,
    1, 0
  ), levels = c(0, 1)
)

# Get evaluation metrics
confusionMatrix(
  data = prediction,
  reference = factor(Y_test),
  positive = "1"
)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 241  30
         1   7  10

               Accuracy : 0.8715
                 95% CI : (0.8273, 0.9079)
    No Information Rate : 0.8611
    P-Value [Acc > NIR] : 0.3413046

                  Kappa : 0.2922

 Mcnemar's Test P-Value : 0.0002983

            Sensitivity : 0.25000
            Specificity : 0.97177
         Pos Pred Value : 0.58824
         Neg Pred Value : 0.88930
             Prevalence : 0.13889
         Detection Rate : 0.03472
   Detection Prevalence : 0.05903
      Balanced Accuracy : 0.61089

       'Positive' Class : 1
```

Perform lasso logistic regression

```r
# Perform CV for lasso
cv_lasso <- cv.glmnet(
  x = X, y = Y,
  family = "binomial", # logistic
  alpha = 1 # 1 = lasso
)

# Use best lambda
lasso_glm <- glmnet(
   x = X, y = Y,
  family = "binomial", # logistic
  alpha = 1, # 1 = lasso
  lambda = cv_lasso$lambda.min
)

# Predict testing data
prediction <- factor(
  ifelse(
    predict(lasso_glm, newx = X_test, type = "response") > 0.50,
    1, 0
  ), levels = c(0, 1)
)

# Get evaluation metrics
confusionMatrix(
  data = prediction,
  reference = factor(Y_test),
  positive = "1"
)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 242  30
         1   6  10

               Accuracy : 0.875
                 95% CI : (0.8312, 0.9109)
    No Information Rate : 0.8611
    P-Value [Acc > NIR] : 0.2799714

                  Kappa : 0.3017

 Mcnemar's Test P-Value : 0.0001264

            Sensitivity : 0.25000
            Specificity : 0.97581
         Pos Pred Value : 0.62500
         Neg Pred Value : 0.88971
             Prevalence : 0.13889
         Detection Rate : 0.03472
   Detection Prevalence : 0.05556
      Balanced Accuracy : 0.61290
```

```
       'Positive' Class : 1
```

```r
# Number of non-zero predictions
cat(paste0(
  "Number of non-zero coefficients = ",
  sum(as.matrix(coef(lasso_glm)) != 0)
))
```

```
Number of non-zero coefficients = 12
```

My preference would be for the lasso model given that it has the best overall prediction with only 12 variables (coefficients). The resulting model is therefore predictive *and* parsimonious.