Bootstrap

Langtao Chen

Update: Mar 2, 2021

Contents

Bootstrap is a general method of quantifying uncertainty of a statistical method. Bootstrap can easily derive standard errors and confidence intervals for complicated statistics, thus making hypothesis testing simple and straightforward.

In many cases, we can easily get the performance measure of multiple statistical learning models. Is a 92% accuracy significantly better than a 91% accuracy? We cannot directly evaluate the significance of performance difference since we don't know the distribution of the performance metric yet.

In this example, let's use the bootstrap method to estimate the distribution of performance metrics for different models, in the context of personal loan acceptance in banking industry. With such estimates, we can test hypothesis regarding the difference of performance for different models.

Let's compare two models including logit and LDA here. As the logit without class weight has too low recall, we'll not compare it in this section. Thus, our (alternative) hypothesis is:

Hypothesis: The logit and LDA models have different performance in terms of prediction accuracy in the Universal Bank data.

1. Data

The data file "UniversalBank.csv" contains a dataset of 5000 customers of the Universal Bank.

Below is the description of columns in the dataset.

- Id: Customer ID
- Age: Customer's age in completed years
- Experience: #years of professional experience
- Income: Annual income of the customer (1000 dollars)
- ZIPCode: Home Address ZIP code.
- Family: Family size of the customer
- CCAvg: Avg. spending on credit cards per month (1000 dollors)
- Education: Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
- Mortgage: Value of house mortgage if any. (1000 dollars)
- Personal_Loan: Did this customer accept the personal loan offered in the last campaign?
- Securities_Account: Does the customer have a securities account with the bank?

- CD_Account: Does the customer have a certificate of deposit (CD) account with the bank?
- Online: Does the customer use internet banking facilities?
- CreditCard: Does the customer use a credit card issued by UniversalBank?

```
# Read in data
bank <- read.csv("UniversalBank.csv")

# Structure of the dataset
str(bank)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ Id                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP_Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal_Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities_Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD_Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```
head(bank)
```

```
##   Id Age Experience Income ZIP_Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal_Loan Securities_Account CD_Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

Let's drop customer ID and zip code and convert education as a factor.

```
bank$Id <- NULL

bank$ZIP_Code <- NULL

bank$Education <- factor(bank$Education)
```

```
summary(bank)
```

```
##       Age          Experience       Income           Family     
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000  
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000  
##  Median :45.00   Median :20.0   Median : 64.00   Median :2.000  
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg        Education    Mortgage      Personal_Loan   Securities_Account
##  Min.   : 0.000   1:2096    Min.   :  0.0   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 0.700   2:1403    1st Qu.:  0.0   1st Qu.:0.000   1st Qu.:0.0000
##  Median : 1.500   3:1501    Median :  0.0   Median :0.000   Median :0.0000
##  Mean   : 1.938             Mean   : 56.5   Mean   :0.096   Mean   :0.1044
##  3rd Qu.: 2.500             3rd Qu.:101.0   3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :10.000             Max.   :635.0   Max.   :1.000   Max.   :1.0000
##    CD_Account         Online         CreditCard
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.0604   Mean   :0.5968   Mean   :0.294
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

2. Predictive Modeling

2.1. Data Partition

Randomly split the dataset into a train (70%) and a test set (30%).

```
index <- 1:nrow(bank)
set.seed(123)
train_index <- sample(index, round(length(index)*0.7))
train_set <- bank[train_index,]
test_set <- bank[-train_index,]
```

2.2. Fit a Logistic Regression Model and Test Its Performance

```
# Logistic regression
logit.fit <- glm(Personal_Loan ~ .,
             family=binomial(link='logit'), data = train_set)

summary(logit.fit)
```

```
##
## Call:
## glm(formula = Personal_Loan ~ ., family = binomial(link = "logit"),
##     data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0224  -0.1878  -0.0705  -0.0230   3.9083
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.222e+01  2.136e+00  -5.723 1.05e-08 ***
## Age                  -2.258e-02  7.879e-02  -0.287 0.774414
## Experience            2.178e-02  7.827e-02   0.278 0.780826
## Income                6.035e-02  3.554e-03  16.980  < 2e-16 ***
## Family                5.620e-01  9.266e-02   6.065 1.32e-09 ***
## CCAvg               1.147e-01  5.337e-02   2.150 0.031569 *
## Education2          4.040e+00  3.238e-01  12.474  < 2e-16 ***
## Education3          4.140e+00  3.253e-01  12.729  < 2e-16 ***
## Mortgage            6.798e-04  7.002e-04   0.971 0.331590
## Securities_Account -7.209e-01  3.432e-01  -2.100 0.035690 *
## CD_Account          3.744e+00  3.965e-01   9.442  < 2e-16 ***
## Online             -8.632e-01  1.982e-01  -4.355 1.33e-05 ***
## CreditCard         -9.468e-01  2.534e-01  -3.736 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2163.69  on 3499  degrees of freedom
## Residual deviance:  825.44  on 3487  degrees of freedom
## AIC: 851.44
##
## Number of Fisher Scoring iterations: 8
```

Predict whether a customer accept the loan or not on the test dataset.

```
logit_test_prob <- predict(logit.fit, newdata = test_set, type="response")

logit_test_pred <- ifelse(logit_test_prob > 0.5, 1, 0)
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
confusionMatrix(factor(logit_test_pred),factor(test_set$Personal_Loan), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1338   46
##          1    7  109
##
##                Accuracy : 0.9647
##                  95% CI : (0.954, 0.9734)
##     No Information Rate : 0.8967
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7854
##
##  Mcnemar's Test P-Value : 1.792e-07
##
##             Sensitivity : 0.70323
##             Specificity : 0.99480
##          Pos Pred Value : 0.93966
##          Neg Pred Value : 0.96676
##              Prevalence : 0.10333
##          Detection Rate : 0.07267
##    Detection Prevalence : 0.07733
##       Balanced Accuracy : 0.84901
```

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

2.3. Fit an LDA Model and Test Its Performance

```
library(MASS)
lda.fit=lda(Personal_Loan ~ ., data = train_set)
lda.fit
```

```
## Call:
## lda(Personal_Loan ~ ., data = train_set)
##
## Prior probabilities of groups:
##          0          1
## 0.90714286 0.09285714
##
## Group means:
##        Age Experience    Income   Family    CCAvg Education2 Education3
## 0 45.54142   20.32031  66.27969 2.383622 1.738101  0.2689764  0.2831496
## 1 44.77846   19.57538 143.66154 2.603077 3.803631  0.3938462  0.4123077
##    Mortgage Securities_Account CD_Account    Online CreditCard
## 0  50.66677          0.1039370 0.03496063 0.5987402  0.2925984
## 1 103.09846          0.1169231 0.29846154 0.6123077  0.3200000
##
## Coefficients of linear discriminants:
##                              LD1
## Age                -0.0304449535
## Experience          0.0325474202
## Income              0.0215386297
## Family              0.1963137687
## CCAvg               0.0602090352
## Education2          1.0352610786
## Education3          1.0699307573
## Mortgage            0.0005933247
## Securities_Account -0.4253243009
## CD_Account          2.2959496455
## Online             -0.1946068620
## CreditCard         -0.2852380613
```

```
lda.pred <- predict(lda.fit, newdata = test_set)

confusionMatrix(factor(lda.pred$class),factor(test_set$Personal_Loan), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1325   58
##          1   20   97
##
##                Accuracy : 0.948
##                  95% CI : (0.9355, 0.9587)
##     No Information Rate : 0.8967
##     P-Value [Acc > NIR] : 6.670e-13
```

```
##
##                  Kappa : 0.6853
##
##  Mcnemar's Test P-Value : 2.797e-05
##
##            Sensitivity : 0.62581
##            Specificity : 0.98513
##         Pos Pred Value : 0.82906
##         Neg Pred Value : 0.95806
##             Prevalence : 0.10333
##         Detection Rate : 0.06467
##   Detection Prevalence : 0.07800
##      Balanced Accuracy : 0.80547
##
##       'Positive' Class : 1
##
```

2.4. Use Bootstrap to Compare Two Models

```
# Set the number of bootstraps
n_bootstraps <- 1000

# Initiate vectors of performance metric
bootstrap_acc_logit <- NULL
bootstrap_acc_lda <- NULL

# Set the random number seed
set.seed(100)

for (i in 1:n_bootstraps){
  # Get a bootstrap of test dataset
  resample_test <- test_set[sample(nrow(test_set), replace = TRUE),]

  # Calculate predicted outcome
  logit_resample_prob <- predict(logit.fit, newdata = resample_test, type="response")
  logit_resample_pred <- ifelse(logit_resample_prob > 0.5, 1, 0)

  lda_resample_pred <- predict(lda.fit, newdata = resample_test)

  # Calculate accuracy of the logit model using the bootstrap
  bootstrap_acc_logit <- c(bootstrap_acc_logit, mean(logit_resample_pred == resample_test$Personal_Loan

  # Calculate f1 score of the logit model using the bootstrap
  bootstrap_acc_lda <- c(bootstrap_acc_lda, mean(lda_resample_pred$class == resample_test$Personal_Loan
}
```

```
summary(bootstrap_acc_logit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9480  0.9613  0.9647  0.9647  0.9680  0.9773
```

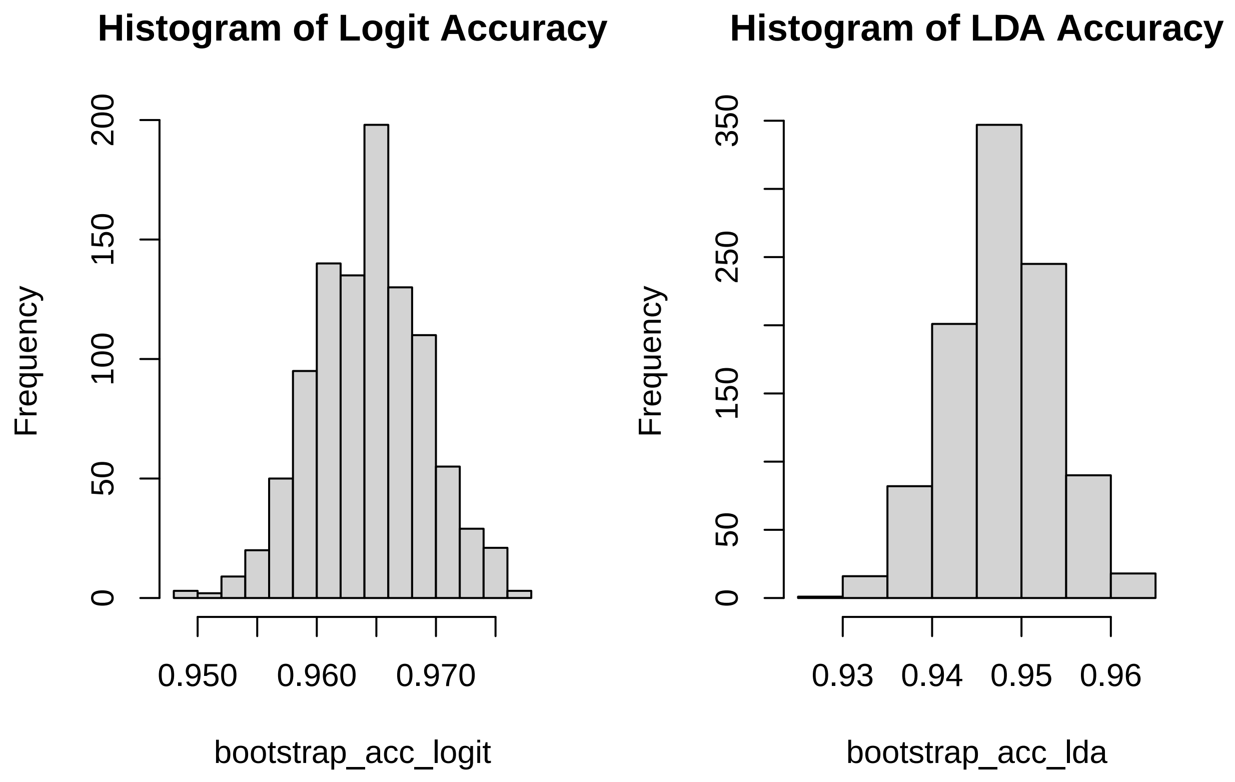
```
summary(bootstrap_acc_lda)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.9293  0.9440  0.9480  0.9480  0.9520  0.9647
```

```r
# Plot the accuracy

par(mfrow=c(1,2))

hist(bootstrap_acc_logit,main = "Histogram of Logit Accuracy")
hist(bootstrap_acc_lda, main = "Histogram of LDA Accuracy")
```



The mean accuracy of the logit model is 0.9612, while the mean accuracy of the LDA model is 0.9493. Do the two accuracy scores have significant difference?

Now, let's calculate 95% confidence intervals using the percentile method.

```r
cat("95% CI of Logit = ",
    "(",
    quantile(bootstrap_acc_logit, 0.025),
    ", ",
    quantile(bootstrap_acc_logit, 0.975),
    ")")
```

```
## 95% CI of Logit =  ( 0.956 ,  0.974 )
```

It shows that the accuracy of the logit model belongs to the range (0.951, 0.971) with 95% confidence.

```r
cat("95% CI of LDA = ",
    "(",
    quantile(bootstrap_acc_lda, 0.025),
    ", ",
    quantile(bootstrap_acc_lda, 0.975),
    ")")
```

```
## 95% CI of LDA =  ( 0.9366667 ,  0.9593333 )
```

It shows that the accuracy of the LDA model belongs to the range (0.939, 0.961) with 95% confidence.

As the two intervals have overlap, the difference in accuracy score between the logit and LDA models is not statistically significant. We don't have evidence to support the hypothesis that the two models have different performance in terms prediction accuracy. That is to say, LDA has the similar performance with logit model in terms of prediction accuracy.

Another way to calculate the 95% confidence interval is to use normal approximation. Assume that the statistic x (accuracy score in this case) follows a normal distribution, then the 95% confidence interval of x is calculated as:

$$[mean(x) - 1.96 * std(x), mean(x) + 1.96 * std(x)]$$

```
cat("95% CI of Logit (normal approximation) = ",
    "(",
    mean(bootstrap_acc_logit)-1.96*sd(bootstrap_acc_logit),
    ", ",
    mean(bootstrap_acc_logit)+1.96*sd(bootstrap_acc_logit),
    ")")
```

```
## 95% CI of Logit (normal approximation) =  ( 0.9554974 ,  0.9738279 )
```

```
cat("95% CI of LDA (normal approximation) = ",
    "(",
    mean(bootstrap_acc_lda, 0.025)-1.96*sd(bootstrap_acc_lda),
    ", ",
    mean(bootstrap_acc_lda, 0.025)+1.96*sd(bootstrap_acc_lda),
    ")")
```

```
## 95% CI of LDA (normal approximation) =  ( 0.9366868 ,  0.9592626 )
```

We get the similar result by using the normal approximation to calculate the 95% confidence interval.

The percentile method and normal approximation method have similar results. However, the percentile method is more preferred as it is a non-parametric method: we don't need to make assumption regarding the distribution of the accuracy score.