DM_assign1

Weijie Gao

5 March 2017

```r
dataPath <- "~/Documents/Chicago2016/Spring/Data Mining/week2"
GermanCredit <- read.table(file=paste(dataPath,"Germancredit_numertic.csv",sep="/"),header=TRUE)
head(GermanCredit)
```

```
##   Status Duration Credit_history Purpose Credit_Amount Savings_Account
## 1      1        6              5       4          1169               5
## 2      2       48              3       4          5951               1
## 3      4       12              5       7          2096               1
## 4      1       42              3       3          7882               1
## 5      1       24              4       1          4870               1
## 6      4       36              3       7          9055               5
##   Employment Installment_rate Status_Sex Other_guarantors
## 1          5                4          3                1
## 2          3                2          2                1
## 3          4                2          3                1
## 4          4                2          3                3
## 5          3                3          3                1
## 6          3                2          3                1
##   Present_residence Property Age Other_installment Housing
## 1                 4        1  67                 3       2
## 2                 2        1  22                 3       2
## 3                 3        1  49                 3       2
## 4                 4        2  45                 3       3
## 5                 4        4  53                 3       3
## 6                 4        4  35                 3       3
##   Num_existingcredit Job Num_maintenance Telephone Foreign_worker Class
## 1                  2   3               1         2              1     1
## 2                  1   3               1         1              1     2
## 3                  1   2               2         1              1     1
## 4                  1   3               2         1              1     1
## 5                  2   3               2         1              1     2
## 6                  1   2               2         2              1     1
```

```r
# fit linear regression with all variables
full.model <- lm(GermanCredit$Credit_Amount~.,data=GermanCredit)
(full.model.r.square <- summary(full.model)$r.squared)
```

```
## [1] 0.5593066
```

```r
# fit linear regression with only intercept
null.model <- lm(GermanCredit$Credit_Amount~1,data=GermanCredit)
(null.model.r.square <- summary(null.model)$r.square)
```

```
## [1] 0
```

```r
# perform add1 forward selection
forwards <- step(null.model,trace=0,scope=list(lower=formula(null.model),upper=formula(full.model)),dir
(step.forwards.r.square <- summary(forwards)$r.square)
```

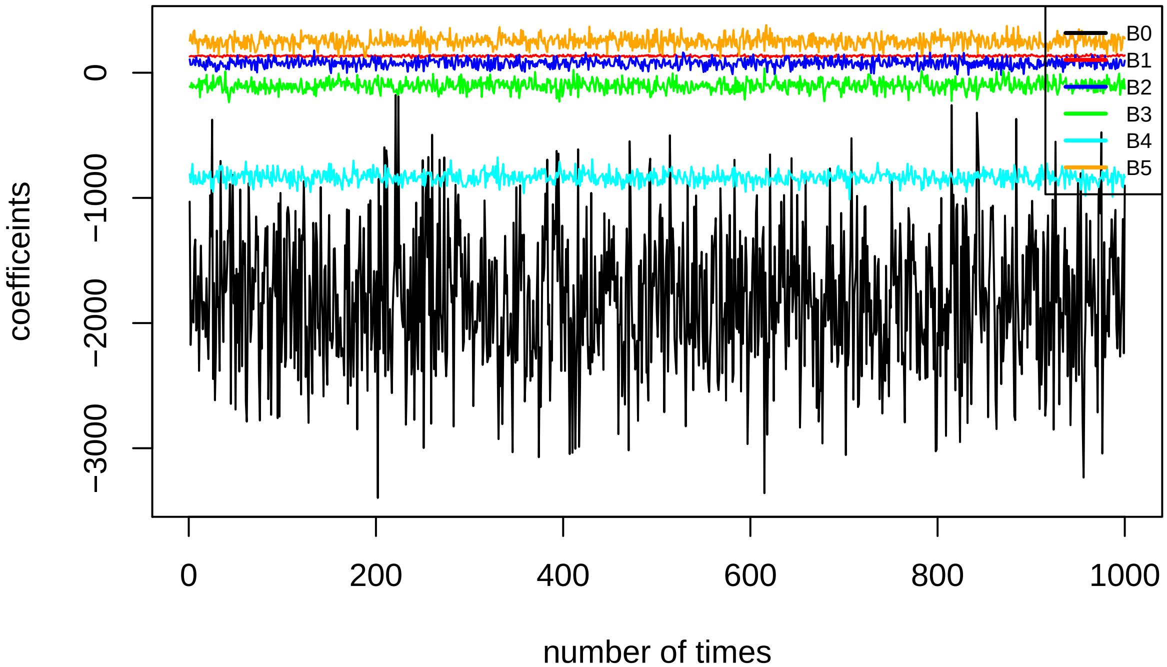
```
## [1] 0.5583012
```

```r
summary(forwards)
```

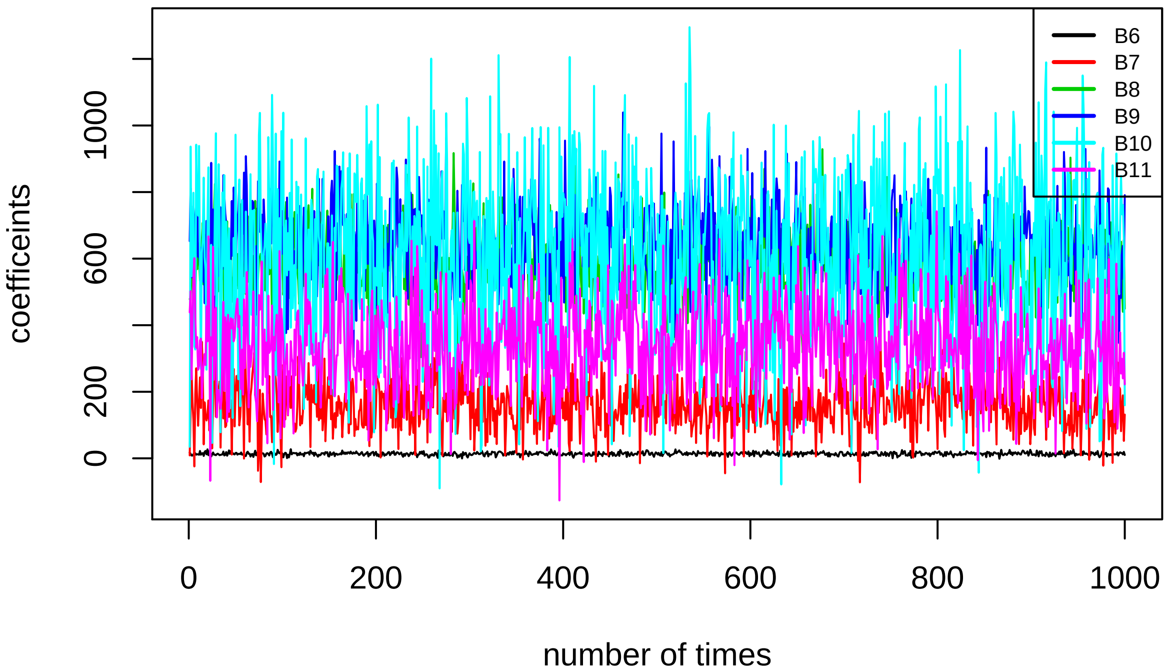
```
##
## Call:
## lm(formula = GermanCredit$Credit_Amount ~ Duration + Installment_rate +
##     Job + Telephone + Property + Age + Class + Foreign_worker +
##     Savings_Account + Employment + Num_existingcredit, data = GermanCredit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5303.9 -1112.5  -194.3   733.5 11988.1
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1809.812    584.276  -3.098  0.00201 **
## Duration             133.721      5.408  24.724  < 2e-16 ***
## Installment_rate    -841.426     54.427 -15.460  < 2e-16 ***
## Job                  581.965    102.991   5.651 2.09e-08 ***
## Telephone            649.058    135.472   4.791 1.91e-06 ***
## Property             250.208     62.184   4.024 6.17e-05 ***
## Age                   13.785      5.563   2.478  0.01338 *
## Class                338.217    138.420   2.443  0.01472 *
## Foreign_worker       606.969    322.959   1.879  0.06048 .
## Savings_Account       75.729     38.925   1.946  0.05200 .
## Employment          -101.891     52.576  -1.938  0.05291 .
## Num_existingcredit   158.107    105.415   1.500  0.13397
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1886 on 988 degrees of freedom
## Multiple R-squared:  0.5583, Adjusted R-squared:  0.5534
## F-statistic: 113.5 on 11 and 988 DF,  p-value: < 2.2e-16
```

```r
# Choose variables: Duration,Installment_rate,Job,Telephone,Property,Age,Class,
# Foreign_worker,Savings_Account,Employment and Num_existingcredit.

# which(colnames(GermanCredit)=="Age")
# which(colnames(GermanCredit)=="Credit_Amount")

# subtract the selected variables
GermanCredit <- GermanCredit[,c(2,5,6,7,8,12,13,16,17,19,20,21)]

# split the sample randomly into training-test using a 632:368 ratio, and compute r square
# in training and holdout. Run the process 1000 times and save the results.
rsquare_train <- matrix(NA,1000)
rsquare_test <- matrix(NA,1000)
coefficients <- matrix(NA,12,1000)

for (i in 1:1000){
  train_ind <- sample(nrow(GermanCredit), size = 0.632 * nrow(GermanCredit))
  train <- GermanCredit[train_ind, ]
  test <- GermanCredit[-train_ind, ]
  fit.lm <- lm(train$Credit_Amount~.,data=train)
  coefficients[,i] <- coef(fit.lm)
```

```
  rsquare_train[i] <- summary(fit.lm)$r.squared
  predited.value <- predict(fit.lm,newdata=test,type="response")
  rsquare_test[i] <- cor(test$Credit_Amount,predited.value)^2
}

# compute the mean of all 1000 coefficients (for each beta)
coef.mean <- apply(coefficients,1,mean)

# compute the standard deviation of all 1000 coefficients
coef.sd <- apply(coefficients,1,sd)

# plot the distributions of first six coefficients
trans <- t(coefficients)
matplot(trans[,c(1:6)],type='l',lty=1,xlab="number of times",ylab="coefficeints",col=c("black","red","bl
legend("topright",legend=c("B0","B1","B2","B3","B4","B5"),lty=1,lwd=2,cex=.7,col=c("black","red","blue"
```

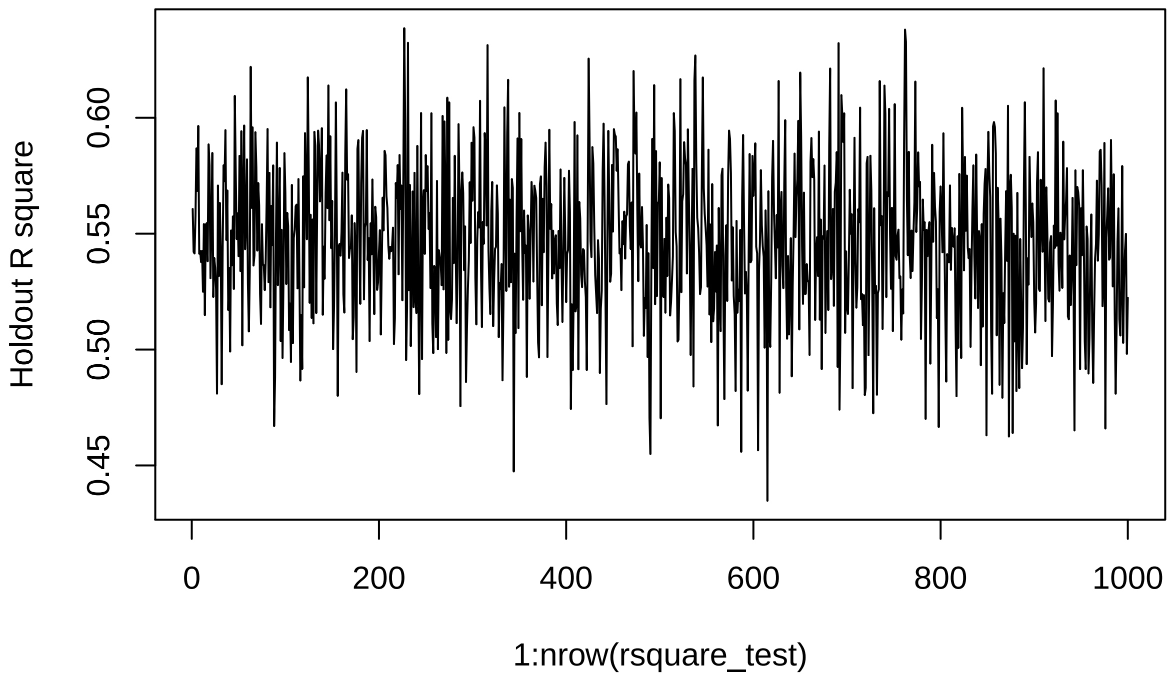



```
matplot(trans[,c(7:12)],type='l',lty=1,xlab="number of times",ylab="coefficeints",col=c(1:6))
legend("topright",legend=c("B6","B7","B8","B9","B10","B11"),lty=1,lwd=2,cex=.7,col=c(1:6))
```

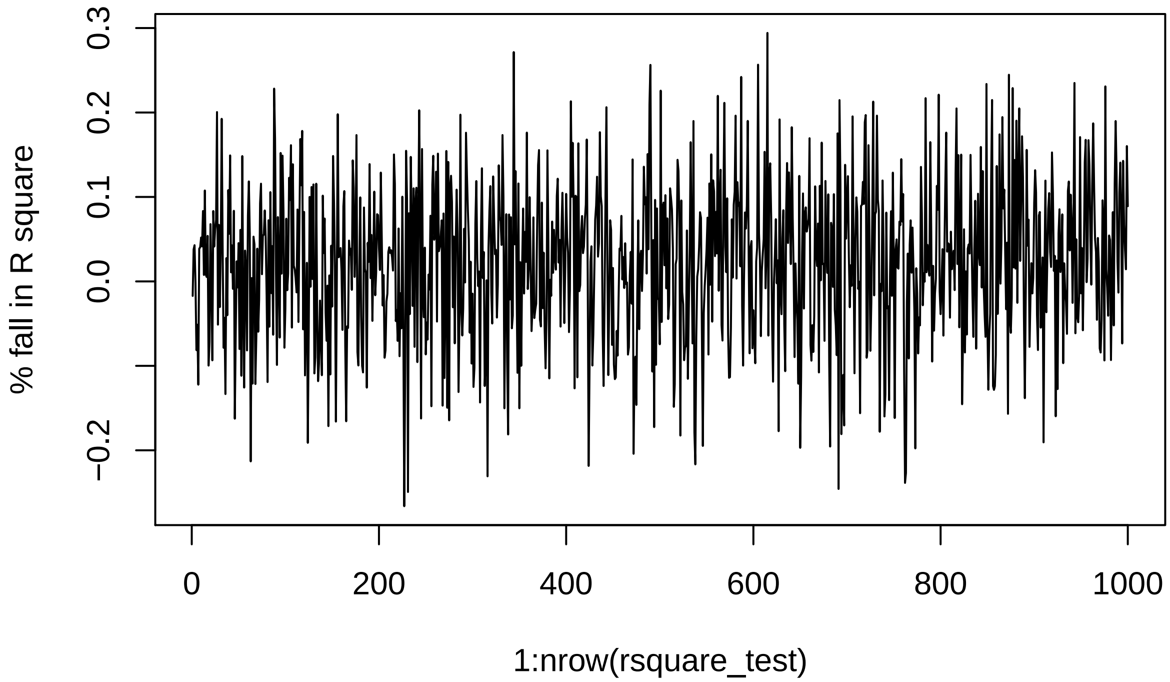


From the plot of all 12 coefficients, we could see that the change of B1 to B6 is much smaller than the change of other coefficients. Specifically, the range of intercept, Class and Foreign worker are among the widest, then comes the coefficients of Telephone, Job and Num_existingcredit. The coefficients of Duration, Savings_Account, Employment, Installment_rate, Property and Age have the least variation.

```r
# plot the distribution of houldout R^2
plot(1:nrow(rsquare_test),rsquare_test,type='l',ylab="Holdout R square")
```



```r
# plot the distribution of % fall in R^2
plot(1:nrow(rsquare_test),(rsquare_train-rsquare_test)/rsquare_train,type='l',ylab="% fall in R square")
```



The above graphes show that the changes of R square range from 0.45 to 0.65, and the percentage fall range from -0.3 to 0.3.

```r
# build linear model using entire sample
fit.lm.entire <- lm(GermanCredit$Credit_Amount~.,data=GermanCredit)
fit.lm.entire$coefficients
```

```
##        (Intercept)           Duration    Savings_Account
##        -1809.81155          133.72059           75.72879
##         Employment   Installment_rate           Property
##         -101.89052         -841.42599          250.20820
##                Age Num_existingcredit                Job
##           13.78484          158.10707          581.96477
##          Telephone     Foreign_worker              Class
##          649.05827          606.96864          338.21741
```

```r
# sort each coefficient's 1000 values
head(apply(trans, 2, sort))
```

```
##           [,1]     [,2]      [,3]      [,4]       [,5]     [,6]       [,7]
## [1,] -3396.960 114.6498 -18.87802 -236.5678 -1012.9589 114.4119 -1.2196644
## [2,] -3358.853 115.5888 -16.75954 -231.3173  -992.0804 123.7693 -0.3924248
## [3,] -3234.412 116.1181 -14.77756 -228.2506  -984.5705 125.8665  0.2471937
## [4,] -3071.904 117.0340 -13.36612 -226.0222  -962.7494 129.9699  0.6942455
## [5,] -3054.217 117.3357 -11.78884 -220.4577  -953.5821 136.4246  1.4348536
## [6,] -3046.306 117.4683 -10.35154 -216.1376  -953.5302 141.0491  1.5184201
##           [,8]     [,9]    [,10]     [,11]       [,12]
## [1,] -72.14288 304.0345 303.5039 -90.34466 -126.482942
## [2,] -71.27918 317.1678 329.0108 -78.50761  -67.388926
## [3,] -44.99127 340.8015 344.4945 -43.21878  -20.516801
```

```
## [4,] -37.36498 343.5476 345.9232 -17.16172  -11.125001
## [5,] -26.62774 344.3937 354.0519  15.82529   -5.646069
## [6,] -24.30356 353.6615 363.6978  16.17486   12.913709
```

```
# Compute 2.5%-97.5% confidence interval
# since (1-0.025)100%CI is mean +- z(0.025/2)*sigma/sqrt(n)
# hence 97.5%CI is mean +- z(0.0125)*sigma/sqrt(n)

conf <- matrix(NA,12,2)
for (i in 1:12){
    conf[i,] <- cbind(coef.mean[i]-qnorm(0.9875)*(coef.sd[i]/sqrt(10)),coef.mean[i]+qnorm(0.9875)*(coef
}
colnames(conf) <- c("2.5%","97.5%")

# scale these CI's down by a factor of 0.632^0.5=0.795
scaled.2.5 <- coef.mean-0.795*(coef.mean-conf[,1])
scaled.97.5 <- coef.mean+0.795*(conf[,2]-coef.mean)

scaled.CI <- cbind(scaled2.5=scaled.2.5,scaled.97.5=scaled.97.5)
scaled.CI
```

```
##         scaled2.5 scaled.97.5
##  [1,] -2121.92139 -1511.40416
##  [2,]   130.14417   136.75931
##  [3,]    58.46715    94.90949
##  [4,]  -126.84997   -78.73607
##  [5,]  -863.19981  -811.16882
##  [6,]   224.98698   275.41400
##  [7,]    11.06925    16.40029
##  [8,]   121.22440   205.71663
##  [9,]   525.88792   636.43526
## [10,]   582.38990   711.71950
## [11,]   461.78039   749.02570
## [12,]   265.57289   411.57046
```

```
# compute single model's CIs
single.model.CI <- confint(fit.lm.entire,fit.lm.entire$coefficients[1],level=0.95)
single.model.CI
```

```
##                            2.5 %      97.5 %
## (Intercept)        -2956.3768512 -663.246253
## Duration             123.1071926  144.333982
## Savings_Account       -0.6559496  152.113522
## Employment          -205.0633748    1.282332
## Installment_rate    -948.2320892 -734.619899
## Property             128.1799050  372.236496
## Age                    2.8681368   24.701538
## Num_existingcredit   -48.7567431  364.970891
## Job                  379.8587069  784.070825
## Telephone            383.2118066  914.904728
## Foreign_worker       -26.7963280 1240.733601
## Class                 66.5864719  609.848343
```

```
rownames(scaled.CI) <- rownames(single.model.CI)
cbind(scaled.CI=scaled.CI,single.model.CI=single.model.CI)
```

```
##                      scaled2.5 scaled.97.5         2.5 %      97.5 %
## (Intercept)        -2121.92139 -1511.40416 -2956.3768512 -663.246253
## Duration             130.14417   136.75931   123.1071926  144.333982
## Savings_Account       58.46715    94.90949    -0.6559496  152.113522
## Employment          -126.84997   -78.73607  -205.0633748    1.282332
## Installment_rate    -863.19981  -811.16882  -948.2320892 -734.619899
## Property             224.98698   275.41400   128.1799050  372.236496
## Age                   11.06925    16.40029     2.8681368   24.701538
## Num_existingcredit   121.22440   205.71663   -48.7567431  364.970891
## Job                  525.88792   636.43526   379.8587069  784.070825
## Telephone            582.38990   711.71950   383.2118066  914.904728
## Foreign_worker       461.78039   749.02570   -26.7963280 1240.733601
## Class                265.57289   411.57046    66.5864719  609.848343
```

According to the above table, the confidence interval of average value across 1000 is tighter than the single model's CIs, especially for the coefficients with large variation such as intercept, Class and foreigh worker. For coefficients such as Duration, Installment_rate, Property and Age, the two confidence intervals are close, and the confidence interval of average value for these coefficients are quite tight. Hence we could notice that by repeating the model construction process multiple times help improve the stability and accuracy of our model and this idea may be further applied to other data mining algorithms.