Multivariate Statistics - Assignment 2

Elena Damiano Xierui Aharon

12/1/2020

Contents

Assignment 2 Multivariate Statistics

This document ...

Task 1

Introduction

In the present part of the report, we will investigate to what extent we will be able to classify respondents in their country, and then we will compare the performance of different classifiers.

Data

The data have been obtained from the 6th Wave of the World Value Survey, which was carried out between 2010 and 2013. The data include the standardized scores of 3929 respondents of 3 countries on 32 variables, that have been summarized with 7 factors obtained using exploratory factor analysis with oblique rotation. The 7 factors related to the 32 variables are:

1. **Rights**, that it's related to homosexuality, prostitution, abortion, divorce, sex before marriage, suicide;
2. **Steal** that it's related to claiming benefits, avoiding fare, stealing property, cheating taxes, accept a bribe;
3. **Crime** that it's related to robberies, alcohol, police-military, racist behavior, drug sale;
4. **Religion** that it's related to: attend religious services, pray, the importance of God;
5. **Realize self** that it's related to creative, rich, spoil oneself, be successful, exciting life;
6. **Do good** that it's related to security, do good, behave properly, protect environment, tradition;
7. **Violence** that it's related to beat wife, parents beating children, violence.

Methodology

To investigate the possibility to classify the respondents in their country based on the 7 factors we have used the canonical discriminant analysis. We have applied the linear regression function with 7 predictors and 1 dependent variable, the Country. Then to the output, we have applied the Canonical Discriminant Analysis.

```
lm.out<-lm(cbind(F_rights, F_steal, F_crime,F_religion,F_realizeself,F_dogood ,F_violence)~as.factor(cou
candisc.out<-candisc(lm.out)
print (candisc.out)
```

```
##
## Canonical Discriminant Analysis for as.factor(country):
##
##    CanRsq Eigenvalue Difference Percent Cumulative
## 1 0.80691    4.17882     3.5622  87.142     87.142
## 2 0.38142    0.61661     3.5622  12.858    100.000
##
## Test of H0: The canonical correlations in the
## current row and all that follow are zero
##
##   LR test stat approx F numDF denDF   Pr(> F)
## 1      0.11944  1059.53    14  7834 < 2.2e-16 ***
```

```
## 2      0.61858   402.65     6  3918 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see both the Square Canonical Correlation are significant, but the discriminating power to separate between the groups is higher for the first than for the second discriminant function: 0.81 and 0.38 respectively. The LR test indicates that the discriminant analysis is meaningful. The first test's null hypothesis is H0: 1= 2=0 and this hypothesis as we can see from the P-value is rejected. The null hypothesis of the first test it's equivalent to the test for H0: Netherlands= Nigeria= Philippines. The second LR test indicates that **H0: 2=0**, and also this null hypothesis is rejected. So even if the second discriminant function has less discriminant power cannot be omitted and it's statistically meaningful. On our analysis, we have also applied two different tests for centroids and to test the equal covariance. To see if the three-country has different centroids and confirm the results of the canonical discriminant analysis we have applied on the linear regression the function Manova:

```
summary(Manova(lm.out), test="Wilks")
```

```
##
## Type II MANOVA Tests:
##
## Sum of squares and products for error:
##                 F_rights    F_steal     F_crime F_religion F_realizeself
## F_rights      2144.25752 1220.99047   24.197889 -516.74466     190.94559
## F_steal       1220.99047 2897.38038  235.937809  -77.66905      99.99545
## F_crime         24.19789  235.93781 3216.820606  -61.84878      40.72785
## F_religion    -516.74466  -77.66905  -61.848780 1417.68129    -215.01914
## F_realizeself  190.94559   99.99545   40.727853 -215.01914    2098.88393
## F_dogood      -281.98487 -431.66636 -224.176736  355.46395     634.31039
## F_violence    1270.23698 1552.67341    3.264745  -31.35059      50.25643
##                 F_dogood  F_violence
## F_rights      -281.9849 1270.236978
## F_steal       -431.6664 1552.673408
## F_crime       -224.1767    3.264745
## F_religion     355.4640  -31.350590
## F_realizeself  634.3104   50.256434
## F_dogood      2777.8489 -481.800194
## F_violence    -481.8002 3394.491184
##
## ----------------------------------------
##
## Term: as.factor(country)
##
## Sum of squares and products for the hypothesis:
##                 F_rights   F_steal   F_crime F_religion F_realizeself
## F_rights       1780.7425 -503.8490 -819.5963 -2050.7548    -1753.4222
## F_steal        -503.8490 1027.6196  773.1158   939.2366      199.2226
## F_crime        -819.5963  773.1158  708.1794  1163.3941      625.4688
## F_religion    -2050.7548  939.2366 1163.3941  2507.3187     1898.8676
## F_realizeself -1753.4222  199.2226  625.4688  1898.8676     1826.1161
## F_dogood      -1293.7342  794.3235  857.3368  1663.6129     1130.2208
## F_violence     -852.6546  570.1760  593.5765  1115.3566      729.2346
##                 F_dogood F_violence
## F_rights      -1293.7342  -852.6546
```

```
## F_steal         794.3235   570.1760
## F_crime         857.3368   593.5765
## F_religion     1663.6129  1115.3566
## F_realizeself  1130.2208   729.2346
## F_dogood       1147.1511   778.6277
## F_violence      778.6277   530.5088
##
## Multivariate Test: as.factor(country)
##                     Df test stat approx F num Df den Df     Pr(>F)
## as.factor(country)  2 0.1194435 1059.531     14   7834 < 2.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is small and the test confirms that the analysis is meaningful and that at least there is a pair of centroids that differs significantly. The function Manova in r doing the Wilks Lambda test uses the Rao approximation. To test the assumption on equal population covariance we have applied to the linear regression the function boxM:

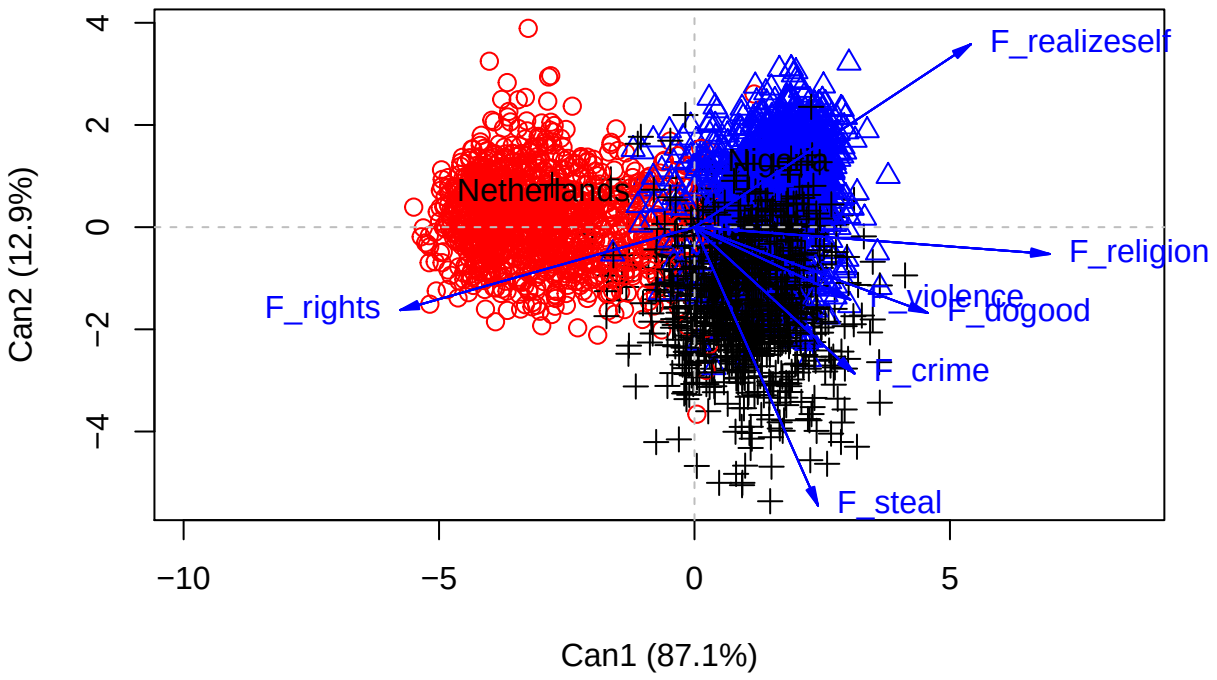
```
boxM(lm.out)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  Y
## Chi-Sq (approx.) = 5479.2, df = 56, p-value < 2.2e-16
```

Plot

To complete the Canonical Discriminant analysis we have plotted the three countries and the 7 variables.

```
plot(candisc.out,col=c("red","blue","black"),pch=c(1,2,3),cex=1.2)
```



We can see that the group of individuals in red are Netherland citizens, the group of individuals in green are Nigerians citizens and the group of individuals in black are Philippines citizens. In blue we can see the 7 explanatory variables. The plot shows a clear separation between Netherland and the other two countries on the first discriminant function while the second discriminant function could help to separate Nigeria and the Philippines. The first discriminant function especially correlates with the factors: rights, religion, realize self, and do good; whereas the second discriminant function correlates with the factors: steal and realize self. The two-factor crime and violence has a lower correlation on the two factors and so has in this analysis lower importance on separate the 3 countries.

The test of Box indicates that H0 of equal covariance matrices across groups is not supported by data. So the Linear discriminant Analysis will not work correctly in that case.

Task 2

Introduction

For task 2, we are going to deal with a dataset containing information about 4601 webmails. We have 48 variables describing the frequency of some specific words like "remove" in each observation, 6 variables describing the frequency of some specific chars like "$" in one observation, and three variables, **capital_run_length_longest**, **capital_run_length_average** and **capital_run_length_total**, describing the length of the longest uninterrupted sequence of capital letters, the average length of uninterrupted sequences of capital letters, and the total number of capital letters in each observation respectively. We also have a variable called spam, which indicates whether this webmail is a spam with 0 and 1, where 1 for spam, and 0 for not spam. Here all our variables are numeric type.

Our task is to use these 57 attribute variables to classify whether a webmail is spam.

Methodology

In order to validate the accuracy of our methods, we firstly divide our dataset into a train set, which contains 2500 observations, and a test set, which contains 2101 observations. We use the train set to train our models, and then apply it to the test set to validate it's accuracy.

Results

Classification Trees

In this part, we are going to discuss the results obtained by complex tree model and pruned tree model.

We begin with construct a complex tree model by dividing our observation into small non-overlapping regions according to some numerical criteria. Here we split our dataset until each leaf of our classification tree contains only less then 2 observations. The method used here is recursive binary splitting.

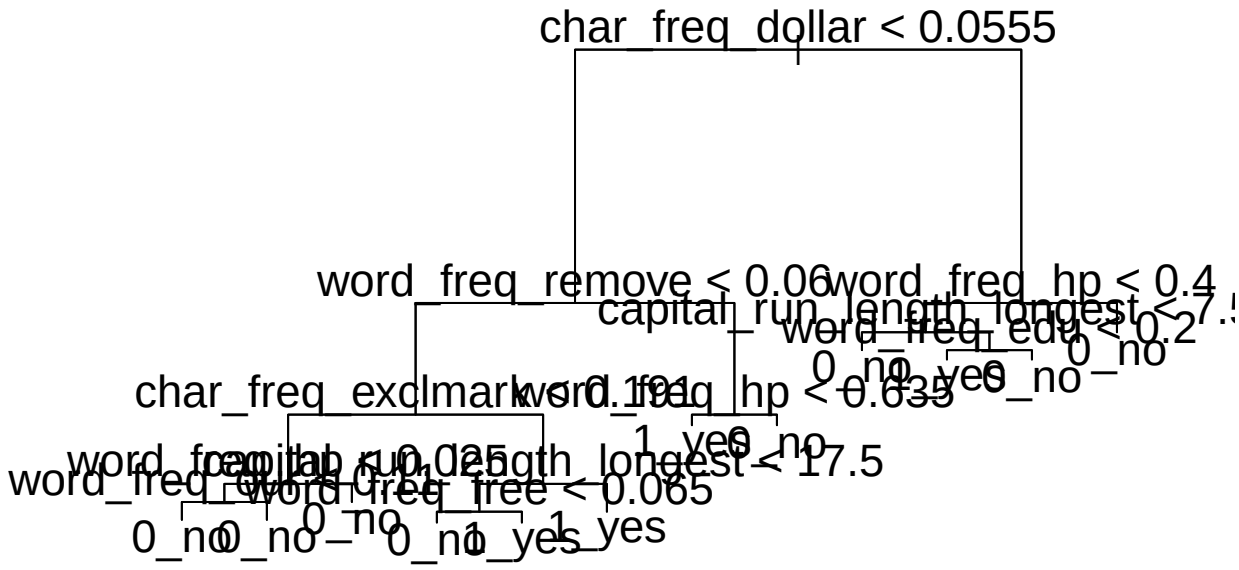
```
err<-function(observed,predicted)
{tab<-table(observed,predicted)
print(tab)
err<-1-sum(diag(tab))/sum(tab)
return(err)
}

#grow complex tree using deviance as criterion
tree.mod <-tree(spam~., data.train, control=tree.control(nobs=2500, minsize=2, mincut=1),
                split="deviance")
summary(tree.mod)
```

```
##
## Classification tree:
## tree(formula = spam ~ ., data = data.train, control = tree.control(nobs = 2500,
##     minsize = 2, mincut = 1), split = "deviance")
## Variables actually used in tree construction:
## [1] "char_freq_dollar"           "word_freq_remove"
## [3] "char_freq_exclmark"         "word_freq_hp"
## [5] "word_freq_our"              "capital_run_length_longest"
## [7] "word_freq_free"             "word_freq_edu"
```

```
## Number of terminal nodes:  12
## Residual mean deviance:  0.4883 = 1215 / 2488
## Misclassification error rate: 0.084 = 210 / 2500
```

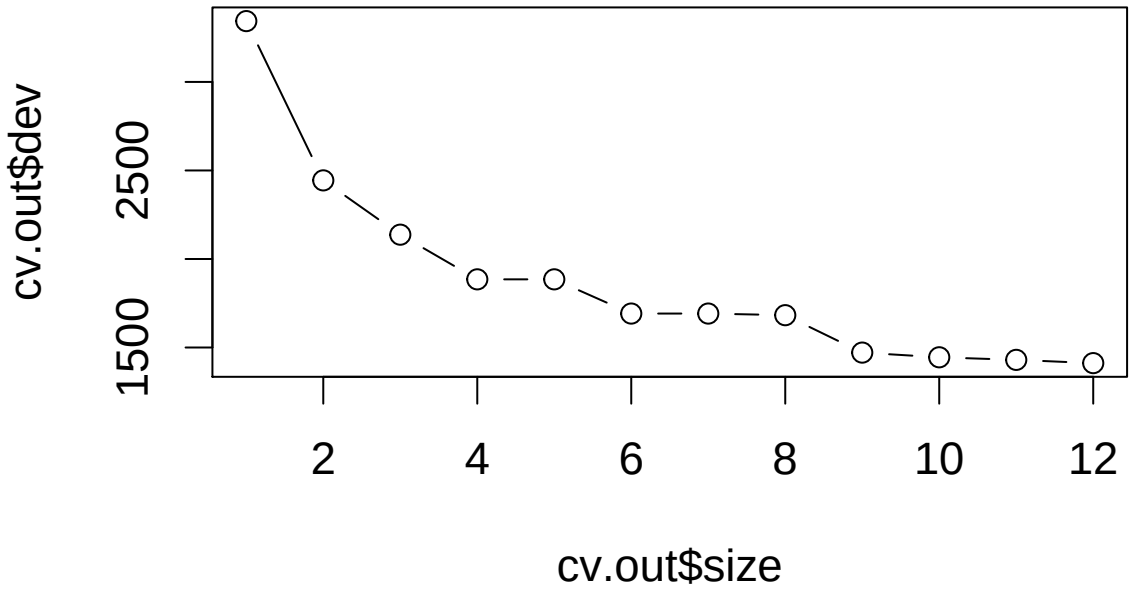
Tree Plots



We can see clearly here that criterias concerning the frequency of "$", "remove", "!", "hp", "our", "free", "edu" as well as the length of the longest uninterrupted sequence of capital letters are used for splitting. It's actually quite reasonable, because from our own experience, spam webmails are always advertisements on money related topics or education related topics, and are always fulled with words in capital letters, together with exclamation symbols, to draw attention.

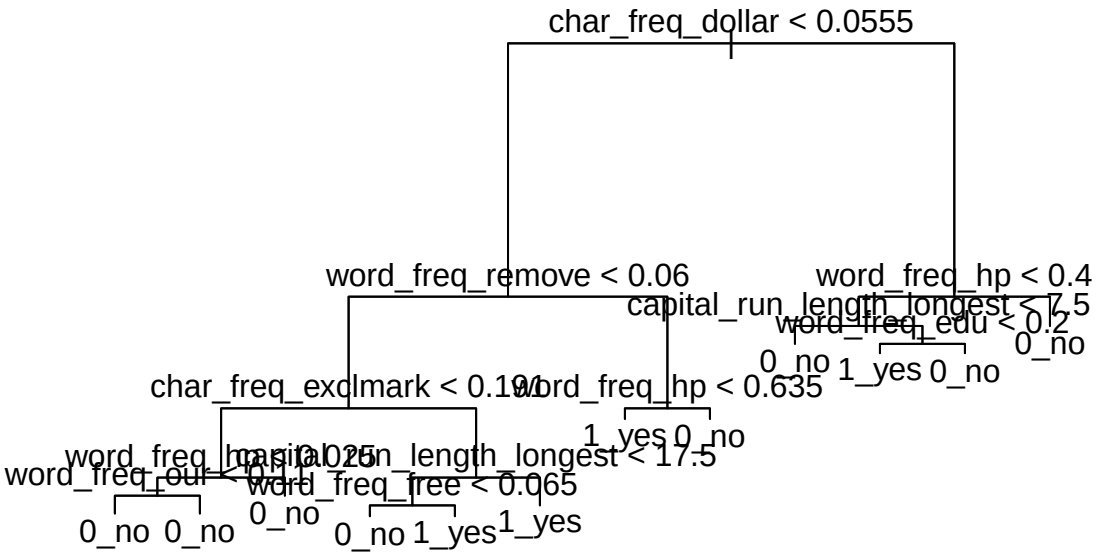
Since the process of recursive binary splitting may lead to overfitting, where we obtain a complex tree with a good fit on the training data, but with a poor performance on test data, we introduce the process of tree pruning. Actually, a smaller tree with fewer splits may have a lower variance at the cost of acceptable little bias. In order to decide the optimal tuning parameter which leads to both much lower variance and acceptable bias, we use cross-validation to make a selection. In fact, the least cross-validation error implies the least probability of overfitting, as it's also a train-test process.

```
#use cross-validation to select tuning parameter for pruning the tree
set.seed(0829539)
cv.out=cv.tree(tree.mod,K=5)
par(cex=1.4)
plot(cv.out$size,cv.out$dev,type='b')
```



However, in this specific task, we can see that the cross-validation error is monotonously decreasing, so the optimal fold number is the original fold number. We choose best size=12 here such that the pruned tree model here is exactly the same as our complex tree model.

```
#prune the tree
prune.mod=prune.tree(tree.mod,best=12)
plot(prune.mod)
text(prune.mod,pretty=0)
```



Now we validate the accuracy of our classification tree model with the test data.

```
#make predictions on training and test set using the unpruned tree
pred.train<-predict(tree.mod,newdata=data.train)
classif.train<-ifelse(pred.train[,2]>=pred.train[,1],1,0)
err(data.train$spam,classif.train)
```

```
##          predicted
## observed    0    1
##    0_no  1460   67
##    1_yes  143  830
```

```
## [1] 0.084
```

```
pred.test<-predict(tree.mod,newdata=data.test)
classif.test<-ifelse(1*pred.test[,2]>=pred.test[,1],1,0)
err(data.test$spam,classif.test)
```

```
##          predicted
## observed    0    1
##    0_no  1209   52
##    1_yes  153  687
```

```
## [1] 0.09757258
```

```
#make predictions on training and test set using the pruned tree
pred.train<-predict(prune.mod,newdata=data.train)
classif.train<-ifelse(pred.train[,2]>=pred.train[,1],1,0)
err(data.train$spam,classif.train)
```

```
##          predicted
## observed    0    1
##    0_no  1460   67
##    1_yes  143  830
```

```
## [1] 0.084
```

```
pred.test<-predict(prune.mod,newdata=data.test)
classif.test<-ifelse(1*pred.test[,2]>=pred.test[,1],1,0)
err(data.test$spam,classif.test)
```

```
##          predicted
## observed    0    1
##    0_no  1209   52
##    1_yes  153  687
```

```
## [1] 0.09757258
```

We can conclude that our classification model performs very well. since the complex tree is the same as pruned tree here, we can see that the test error is just very slightly higher than the train error. There is not much overfitting here.

```
#make predictions on training and test set using the pruned tree
```