Data Mining Assignment 2

Weijie Gao

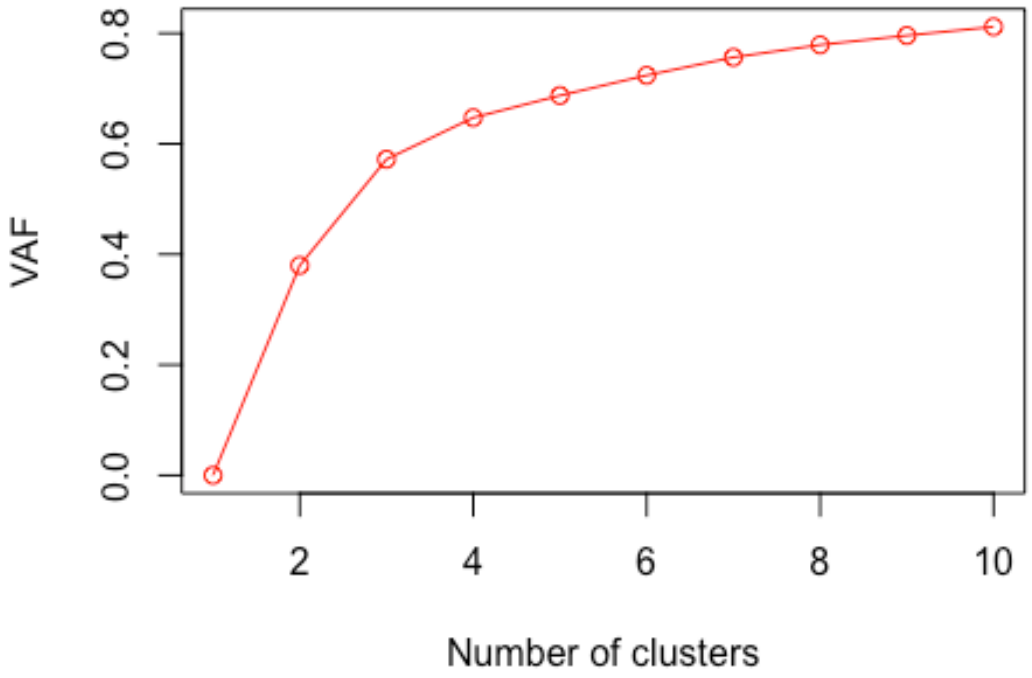
21 January 2017

Perform cluster analysis of the data for market segmentation

Before cluster analysis we usually apply principal component analysis to reduce the number of variables, but in this project in order to simplify the process we just selected the most significant variables that we considered could be used to cluster our product. Also, since k-means and k overlapping means algorithm should not be used in the presence of categorical data we will be focusing on the numerical data, that are duration, amount and age. Therefore, we group people according to their similarity in duration, amount and age and we would like to reduce the number of clusters so that we could sell credit cards to our target customers.

```
dataPath <- "~/Documents/Chicago2016/Spring/Data Mining/week2"
data(GermanCredit,package="caret")
smp_size <- floor(0.632 * nrow(GermanCredit))
set.seed(123)
library(replyr)

## Warning: package 'replyr' was built under R version 3.3.2

# seperate the data into training and test set
train_ind <- sample(seq_len(nrow(GermanCredit)), size = smp_size)
GermanCredit.train <- GermanCredit[train_ind, ]
GermanCredit.test <- GermanCredit[-train_ind, ]

# perform 2-10 k-means clusters on train set and run them from 100 rand
om starts
vaf=0
for (i in 2:10){
  GermanCredit.train.1.2.5<-kmeans(scale(GermanCredit.train[,c(1,2,5)]),
centers=i,nstart=100)
  vaf[i]<-GermanCredit.train.1.2.5$betweenss/GermanCredit.train.1.2.5$t
otss
}

# return the calculated value of vaf corresponding to different cluster
s
vaf

##  [1] 0.0000000 0.3798483 0.5719899 0.6472478 0.6873784 0.7239087 0.7
564865
##  [8] 0.7790655 0.7959765 0.8117268
```

```
# plot the scree plot
plot(1:length(vaf),vaf,type = "o",xlab="Number of clusters",ylab="VAF",
col="red")
```



From the scree plot, we could notice that the value of vaf does not show a significantly increasing trend after cluster 3, hence we will choose the number of clusters to be three.

Also, from the interpretability perspective, it is easier to interpret three clusters. Take the age variable for an example, three cluster allow us to somehow group people into young people, middle-aged and older-aged. Hence, we will choose to group our data into three clusters.

```
# return the final cluster centres with 3 clusters
GermanCredit.train.1.2.5.best <- kmeans(scale(GermanCredit.train[,c(1,2,
5)]),centers=3,nstart=100)
centers_train <- GermanCredit.train.1.2.5.best$centers
cluster_train <- GermanCredit.train.1.2.5.best$cluster

# perform 3 k-means clusters on test set and use the centers trained ab
ove
GermanCredit.teset.1.2.5 <- kmeans(scale(GermanCredit.test[,c(1,2,5)]),
centers=centers_train,nstart=1)
```

```
# check the VAF value for test result
vaf_test <- GermanCredit.teset.1.2.5$betweenss/GermanCredit.teset.1.2.5
$totss
vaf_test

## [1] 0.5606532

# compare the vaf result for 3 cluster and 4 cluster
table1 <- cbind(c(0.5719899, 0.5606532),c(0.6472478, 0.6493281))
colnames(table1) <- c("3 cluster","4 cluster")
rownames(table1) <- c("vaf_train","vaf_test")
table1

##           3 cluster 4 cluster
## vaf_train 0.5719899 0.6472478
## vaf_test  0.5606532 0.6493281
```

From the table above we could notice that the vaf value for test result of 3 cluster is about 0.5606532, dropped a little bit comparing to trained vaf value, so it seems that this result is not very ideal. But if we train our data with 4 clusters, the vaf value increased to 0.6472478 and for test data, the vaf value does not decrease but increased to 0.6493281, which seem to be a more promising result.

```
# check the relative cluster sizes for test result
(cluster.size.test <- GermanCredit.teset.1.2.5$size)

## [1]  59  84 225
```

Although the size of each cluster is not very even, the size of cluster 1 and cluster 2 are close and for each cluster there are comparatively enough data.

```
# check the centers for test result
centers_test <- GermanCredit.teset.1.2.5$centers

# check the difference between train_center and test_center
colnames(centers_train) <- c("Duration_train","Amount_train","Age_train
")
colnames(centers_test) <- c("Duration_test","Amount_test","Age_test")
table2 <- cbind(centers_train,centers_test)
table2

##   Duration_train Amount_train    Age_train Duration_test Amount_test
## 1      1.4789453    1.4923327 -0.001679046     1.6321619   1.6848465
## 2     -0.4846774   -0.3245275  1.306193879    -0.3575625  -0.3033170
## 3     -0.3224665   -0.3975756 -0.572480394    -0.2944991  -0.3285658
##     Age_test
## 1 -0.1932869
## 2  1.4419114
## 3 -0.4876295
```

From the table above, we see the center of each cluster for training data and test data does not change to much, actually they are quite close to each other, which indicates that the our clustering is comparatively stable.

However, since the vaf is low in this stage and it may not be a good way to increase the number of cluster again only to improve the vaf. Hence, we repeat the above whole process three times for different combination of variables to see if we could obtain a higher vaf with lower number of clusters.

```
summary_table <- rbind(c(0.5719899, 0.5606532),c(0.6395609,0.6308548),c
(0.6527128,0.6481235))
summary_table <- cbind(summary_table,c(0.0113367,0.0087061,0.0045893))
summary_table
```

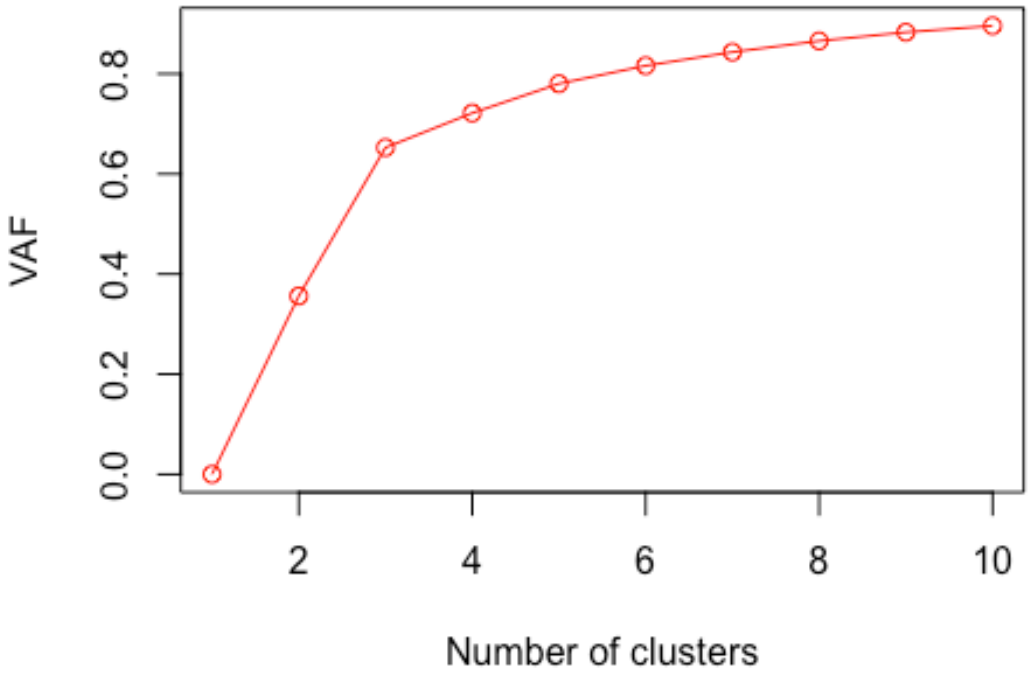
```
##           [,1]      [,2]      [,3]
## [1,] 0.5719899 0.5606532 0.0113367
## [2,] 0.6395609 0.6308548 0.0087061
## [3,] 0.6527128 0.6481235 0.0045893
```

```
colnames(summary_table) <- c("vaf.train","vaf.test","vaf.decreased")
rownames(summary_table) <- c("combination1.2.5","combination1.5","combi
nation2.5")
summary_table
```

```
##                   vaf.train  vaf.test vaf.decreased
## combination1.2.5 0.5719899 0.5606532     0.0113367
## combination1.5   0.6395609 0.6308548     0.0087061
## combination2.5   0.6527128 0.6481235     0.0045893
```

The table above shows the summary results of vaf value for different combination of variables. It could be seen that when choosing variable 2 and variable 5, vaf value for both training set and test set are the highest and it also has the lowest decreasing. Therefore, we may interest in selecting the second and fifth variable as our final decision. And below we run the k-means algorithm choosing variable 2 and 5 and extract the related center and cluster information.

```
# train the k-means algorithm using amount and age
vaf=0
for (i in 2:10){
  GermanCredit.train.2.5<-kmeans(scale(GermanCredit.train[,c(2,5)]),cen
ters=i,nstart=100)
  vaf[i]<-GermanCredit.train.2.5$betweenss/GermanCredit.train.2.5$totss
}
plot(1:length(vaf),vaf,type = "o",xlab="Number of clusters",ylab="VAF",
col="red")
```



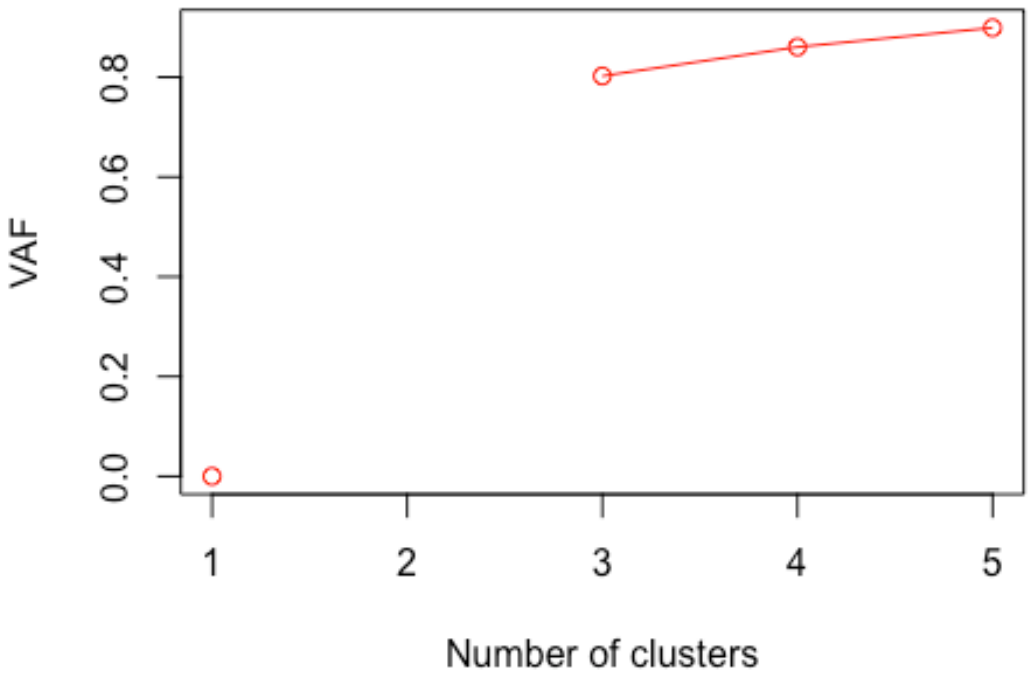

```
# return the final cluster centres with 3 clusters
GermanCredit.train.2.5.best <- kmeans(scale(GermanCredit.train[,c(2,
5)]),centers=3,nstart=100)
centers_train <- GermanCredit.train.2.5.best$centers
cluster_train <- GermanCredit.train.2.5.best$cluster

# perform 3 k-means clusters on test set and use the centers trained ab
ove
GermanCredit.test.2.5 <- kmeans(scale(GermanCredit.test[,c(2,5)]),cente
rs=centers_train,nstart=1)
cluster_test <- GermanCredit.test.2.5$cluster

# check the VAF value for test result
vaf_test <- GermanCredit.test.2.5$betweenss/GermanCredit.test.2.5$totss
vaf_test

## [1] 0.6481235

# perform 3 k-means clusters on the whole dataset
GermanCredit.2.5 <- kmeans(scale(GermanCredit[,c(2,5)]),centers=3,nstar
t=100)
cluster <- GermanCredit.2.5$cluster
```

```
center_amount <- aggregate(GermanCredit$Amount,by=list(cluster),mean)
center_age  <- aggregate(GermanCredit$Age,by=list(cluster),mean)
center_amount <- center_amount[,2]
center_age <- center_age[,2]

table_center <- cbind(center_amount,center_age)
rownames(table_center) <- c("cluster 1","cluster 2","cluster 3")
table_center
```

```
##           center_amount center_age
## cluster 1      2346.724   51.50862
## cluster 2      2282.832   29.55178
## cluster 3      8773.520   35.55333
```

The table above shows that young people with the lowest amount belong to cluster 1 and middle-age people with a comparatively highest amount belong to cluster 3 and the elder-age people with comparatively low amount belong to cluster 2.

```
source(file.path(dataPath,"Komeans.R"))

# perform 3-5 k-overlapping means clusters on train set and run them fr
om 100 random starts
vaf_komeans=0
for (i in 3:5){
  z=komeans(GermanCredit.train[,c(2,5)],nclust=i,lnorm=2,tolerance=.001,
nloops = 100,seed=123)
  vaf_komeans[i]<-z$VAF
}

plot(1:length(vaf_komeans),vaf_komeans,type = "o",xlab="Number of clust
ers",ylab="VAF",col="red")
```




```
# return the value of vaf on test set.
z_best=komeans(GermanCredit.test[,c(2,5)],nclust=3,lnorm=2,tolerance=.0
01,nloops = 100,seed=123)
z_best$VAF

## [1] 0.7934751
```

From the scree plot we could see that the vaf value is pretty high even with 3 clusters, and it also indicates a significant incerase comparing with vaf value of k-means. Hence, we would like to choose k-overlapping means with 3 clusters as our final model.

```
# perform zomeans on the whole dataset and return related centroids and
 group information
z_best_whole=komeans(GermanCredit[,c(2,5)],nclust=3,lnorm=2,tolerance=.
001,nloops = 100,seed=123)
Centroids <- z_best_whole$Centroids

unscaled_centroids_amount <- Centroids[,1]*attr(z_best_whole$Normalized.
Data, 'scaled:scale')[1]+attr(z_best_whole$Normalized.Data, 'scaled:cen
ter')[1]

unscaled_centroids_age <- Centroids[,2]*attr(z_best_whole$Normalized.Da
```

```
ta, 'scaled:scale')[2]+attr(z_best_whole$Normalized.Data, 'scaled:cente
r')[2]

table_unscaled_centroids <- cbind(unscaled_centroids_amount,unscaled_ce
ntroids_age)
rownames(table_unscaled_centroids) <- c("cluster 1","cluster 2","cluste
r 3")
table_unscaled_centroids
```

```
##           unscaled_centroids_amount unscaled_centroids_age
## cluster 1                  1831.662               25.30808
## cluster 2                  2650.256               57.15939
## cluster 3                  9509.506               37.93485
```

The above table shows the unscaled centroid values for each cluster.Young people around 25 with the lowest amount belong to cluster 1 and middle-age people with the highest amount belong to cluster 3 and the elder-age people with comparatively low amount belong to cluster 2. This result is similar to the result of k-means. The most obvious difference is the Amount value for cluster 1 and cluster 3, the former is smaller than the value of k-means and the latter is larger than the value of k-means. And the value of age for cluster 1 is smaller than that of in the k-means model. The table comparing the result of k-means and komeans is showed as follows:

```
table_compare <- cbind(table_center,table_unscaled_centroids)
colnames(table_compare) <- c("Amount.kmeans","Age.kmeans","Amount.komea
ns","Age.komeans")
table_compare
```

```
##           Amount.kmeans Age.kmeans Amount.komeans Age.komeans
## cluster 1      2346.724   51.50862       1831.662    25.30808
## cluster 2      2282.832   29.55178       2650.256    57.15939
## cluster 3      8773.520   35.55333       9509.506    37.93485
```

```
Group <- z_best_whole$Group
group.size <- table(Group)

group.age <- aggregate(GermanCredit$Age,by=list(Group),mean)
group.amount<- aggregate(GermanCredit$Amount,by=list(Group),mean)

group.age <- group.age[,2]
group.amount <- group.amount[,2]

table_unscaled_groups <- cbind(group.amount,group.age)
table_unscaled_groups <- cbind(group.size,table_unscaled_groups)
rownames(table_unscaled_groups) <- c("group 1","group 2","group 3","gro
up 4","group 5","group 6","group 7","group 8")
table_unscaled_groups
```

```
##         group.size group.amount group.age
## group 1        274     2900.208  35.49635
```

```
## group 2         351     1940.028  25.88889
## group 3          85     2223.153  59.55294
## group 4         122     1474.943  44.97541
## group 5          52    10528.673  36.61538
## group 6          76     7358.342  28.19737
## group 7          13    10075.385  63.00000
## group 8          27     7001.296  48.81481
```

The above table displays the unscaled centroid information for attributes Amount and Age with respect to different group and their corresponding group sizes.

In order to segment a market, we would like to dividing our potential consumers into seperate sub-sets where consumners in the same group are similar with respect to a given set of characteristics.This allows us to calibrate the marketing mix differently according to the target consumer group. Since k-overlapping means method is more robust and not sensitive to outliers we apply this method to separate our customers.

When 3 overlapping clusters are extracted, the 3 clusters (1,2,3) result in 2^3 = 8 distinct partitions composed of credit card information belonging to: 1.Overlapping cluster 1 only; 2.Overlapping cluster 2 only; 3.Overlapping cluster 3 only; 4.Overlapping clusters 1 and 2 but not 3; 5.Overlapping clusters 2 and 3 but not 1; 6.Overlapping clusters 3 and 1 but not 2; 7.All overlapping clusters 1, 2, and 3;8.Neither of the overlapping clusters 1, 2 or 3. And from the table above,it could be seen that group 2 has the largest sample size 351 (account for about 35% of data), and in this group both the value of amount and age are small. This result coincided with the common sense that young people with a low amount value credit may account for majority of the credit card market since they have a high shopping demand and relatively low income,and credit card could enable them to buy desired product in advance. Group 7 is the smallest group size 13 (account for about 1.3%), this group has a large value of credit amount and eldest age.This result coincided with the common sense that elder people usually have a stable income and have a good credit, but they may not have a strong shopping demand and will use credit card less frequently.

If we would like to recruit 30 people into these three clusters for focus groups we will be interested in finding the comparatively large size of group within that cluster. And for the cross table below, for cluster 1 we may want to recruit people within group 0 and 1 instead of other groups as there are too small sample size for these groups and for cluster 2, we may interested in recruiting people within group 2 and 3, and similarly for cluster 3, customer within group 4 and group 5 are most representative.

```
cluster.group.crosstable <- table(cluster,Group)
cluster.group.crosstable
```

```
##         Group
## cluster   0   1   2   3   4   5   6   7
##       1  37   0  85  95   0   0   7   8
```

```
##       2 230 351   0  27   0  10   0   0
##       3   7   0   0   0  52  66   6  19
```

Empirically, smaller partitions typically turn out to be very valuable, and this information is usually neglected by people, therefore when we decided to recruit targeted customer we would like to explore more opportunities for these small groups. In our case, the total size of group 5,6,7,8 together account for only 16.8% of our consumer but when we look at the attributes we could notice that for group 5 and 6, these people are in their 30's and have a large value of amount, showing a comparatively good credit. Since these people are in the career developing phase and income may increasingly go up, and they usually have a greater purchasing power, selling credit cards to these people seem to be a promising choice.And for group 7 and 8, although they may not have the purchasing power as group 4 and 5, these elder aged people usually have a stable job with optimistic income and have a strong ability to pay (repay loans), hence selling credit cards to these people has a comparatively low risk. Hence, we will be focusing on these small groups and specifically group 5 and 6.

And based on our centroid information for each cluster, when a new recruit comes, we could tell their cluster based on the value of credit amount and their age. For example, a young people with a low value of amount usually belong to cluster 1 and middle aged people with a high amount value belongs to cluster 3 and elder aged people with a relatively low amount value belong to cluster 2.