Miller Chapter 4 example

This script demonstrates the use of Cluster Analysis to identify consumer segments from a data file with approximately 4,500 bank customers.

It is modified from the script found in Figure 4.1 of Miller.

In clustering models, the workflow typically involves a good deal of exploratory work, reviewing numerous model formulations before zeroing in on one that makes sense for the business. In the code below, you will see the trial-and-error exploration of models based on different combinations of independent predictors (also know as "feature selection").

# call in R packages for use in this study  
library(lattice) # multivariate data visualization  
library(vcd) # data visualization for categorical variables  
library(cluster) # cluster analysis methods

The raw data is in a text file with a csv suffix, but that happens to use semicolons (;) as the separator instead of commas. Also notice in the read.csv command that we instruct R *not* to treat strings as factors, but leave them as character variables.

In this chunk, we read the raw data, look at the structure of the data frame and the first few rows (with the head function).

# read bank data into R, creating data frame bank  
# note that this is a semicolon-delimited file  
bank <- read.csv("C:/Users/Rob/Box Sync/My R Work/BUS256/Data/bank.csv",   
 sep = ";", stringsAsFactors = FALSE)  
  
# examine the structure of the bank data frame  
str(bank)

## 'data.frame': 4521 obs. of 17 variables:  
## $ age : int 30 33 35 30 59 35 36 39 41 43 ...  
## $ job : chr "unemployed" "services" "management" "management" ...  
## $ marital : chr "married" "married" "single" "married" ...  
## $ education: chr "primary" "secondary" "tertiary" "tertiary" ...  
## $ default : chr "no" "no" "no" "no" ...  
## $ balance : int 1787 4789 1350 1476 0 747 307 147 221 -88 ...  
## $ housing : chr "no" "yes" "yes" "yes" ...  
## $ loan : chr "no" "yes" "no" "yes" ...  
## $ contact : chr "cellular" "cellular" "cellular" "unknown" ...  
## $ day : int 19 11 16 3 5 23 14 6 14 17 ...  
## $ month : chr "oct" "may" "apr" "jun" ...  
## $ duration : int 79 220 185 199 226 141 341 151 57 313 ...  
## $ campaign : int 1 1 1 4 1 2 1 2 2 1 ...  
## $ pdays : int -1 339 330 -1 -1 176 330 -1 -1 147 ...  
## $ previous : int 0 4 1 0 0 3 2 0 0 2 ...  
## $ poutcome : chr "unknown" "failure" "failure" "unknown" ...  
## $ response : chr "no" "no" "no" "no" ...

head(bank)

## age job marital education default balance housing loan contact  
## 1 30 unemployed married primary no 1787 no no cellular  
## 2 33 services married secondary no 4789 yes yes cellular  
## 3 35 management single tertiary no 1350 yes no cellular  
## 4 30 management married tertiary no 1476 yes yes unknown  
## 5 59 blue-collar married secondary no 0 yes no unknown  
## 6 35 management single tertiary no 747 no no cellular  
## day month duration campaign pdays previous poutcome response  
## 1 19 oct 79 1 -1 0 unknown no  
## 2 11 may 220 1 339 4 failure no  
## 3 16 apr 185 1 330 1 failure no  
## 4 3 jun 199 4 -1 0 unknown no  
## 5 5 may 226 1 -1 0 unknown no  
## 6 23 feb 141 2 176 3 failure no

Tabulate the levels of several character variables. The useNA option includes NAs (missing) in the tables.

table(bank$job , useNA = c("always"))

##   
## admin. blue-collar entrepreneur housemaid management   
## 478 946 168 112 969   
## retired self-employed services student technician   
## 230 183 417 84 768   
## unemployed unknown <NA>   
## 128 38 0

table(bank$marital , useNA = c("always"))

##   
## divorced married single <NA>   
## 528 2797 1196 0

table(bank$education , useNA = c("always"))

##   
## primary secondary tertiary unknown <NA>   
## 678 2306 1350 187 0

table(bank$default , useNA = c("always"))

##   
## no yes <NA>   
## 4445 76 0

table(bank$housing , useNA = c("always"))

##   
## no yes <NA>   
## 1962 2559 0

table(bank$loan , useNA = c("always"))

##   
## no yes <NA>   
## 3830 691 0

After reviewing the job titles, we'll lump some titles together into larger groupings.

# Type of job (admin., unknown, unemployed, management,  
# housemaid, entrepreneur, student, blue-collar, self-employed,  
# retired, technician, services)  
# put job into three major categories defining the factor variable jobtype  
# the "unknown" category is how missing data were coded for job...   
# include these in "Other/Unknown" category/level  
white\_collar\_list <- c("admin.","entrepreneur","management","self-employed")   
blue\_collar\_list <- c("blue-collar","services","technician")  
bank$jobtype <- rep(3, length = nrow(bank))  
bank$jobtype <- ifelse((bank$job %in% white\_collar\_list), 1, bank$jobtype)   
bank$jobtype <- ifelse((bank$job %in% blue\_collar\_list), 2, bank$jobtype)   
bank$jobtype <- factor(bank$jobtype, levels = c(1, 2, 3),   
 labels = c("White Collar", "Blue Collar", "Other/Unknown"))  
with(bank, table(job, jobtype, useNA = c("always"))) # check definition

## jobtype  
## job White Collar Blue Collar Other/Unknown <NA>  
## admin. 478 0 0 0  
## blue-collar 0 946 0 0  
## entrepreneur 168 0 0 0  
## housemaid 0 0 112 0  
## management 969 0 0 0  
## retired 0 0 230 0  
## self-employed 183 0 0 0  
## services 0 417 0 0  
## student 0 0 84 0  
## technician 0 768 0 0  
## unemployed 0 0 128 0  
## unknown 0 0 38 0  
## <NA> 0 0 0 0

# define binary indicator variables as numeric 0/1 variables  
bank$whitecollar <- ifelse((bank$jobtype == "White Collar"), 1, 0)  
bank$bluecollar <- ifelse((bank$jobtype == "Blue Collar"), 1, 0)  
with(bank, print(table(whitecollar, bluecollar))) # check definition

## bluecollar  
## whitecollar 0 1  
## 0 592 2131  
## 1 1798 0

with(bank, print(table(jobtype))) # check definition

## jobtype  
## White Collar Blue Collar Other/Unknown   
## 1798 2131 592

Next we create dummy indicator variables for marital status.

# define factor variables with labels for plotting and binary factors  
bank$marital <- factor(bank$marital,   
 labels = c("Divorced", "Married", "Single"))  
   
# define binary indicator variables as numeric 0/1 variables  
bank$divorced <- ifelse((bank$marital == "Divorced"), 1, 0)  
bank$married <- ifelse((bank$marital == "Married"), 1, 0)   
with(bank, print(table(divorced, married))) # check definition

## married  
## divorced 0 1  
## 0 1196 2797  
## 1 528 0

with(bank, print(table(marital))) # check definition

## marital  
## Divorced Married Single   
## 528 2797 1196

And now, yet a bit more data pre-processing, relabeling, and data reduction.

bank$education <- factor(bank$education,   
 labels = c("Primary", "Secondary", "Tertiary", "Unknown"))  
# define binary indicator variables as numeric 0/1 variables  
bank$primary <- ifelse((bank$education == "Primary"), 1, 0)  
bank$secondary <- ifelse((bank$education == "Secondary"), 1, 0)   
bank$tertiary <- ifelse((bank$education == "Tertiary"), 1, 0)   
with(bank, print(table(primary, secondary, tertiary))) # check definition

## , , tertiary = 0  
##   
## secondary  
## primary 0 1  
## 0 187 2306  
## 1 678 0  
##   
## , , tertiary = 1  
##   
## secondary  
## primary 0 1  
## 0 1350 0  
## 1 0 0

with(bank, print(table(education))) # check definition

## education  
## Primary Secondary Tertiary Unknown   
## 678 2306 1350 187

# client experience variables will not be useful for segmentation   
# but can be referred to after segments have been defined  
bank$default <- factor(bank$default, labels = c("No", "Yes"))  
bank$housing <- factor(bank$housing, labels = c("No", "Yes"))  
bank$loan <- factor(bank$loan, labels = c("No", "Yes"))  
bank$response <- factor(bank$response, labels = c("No", "Yes"))

One of the challenges with cluster analysis is that the process will run very slowly if we select too many candidate variables. We also know that data tables with large numbers of rows also lead to very slow processing.

In this case, we'll subset the original data to concentrate on cases never previously contacted by sales, and we'll keep variables needed for cluster analysis and post-analysis

We should really break the full sample into training and test sets, but will follow Miller's example and not do so.

bankfull <- subset(bank, subset = (previous == 0),  
 select = c("response", "age", "jobtype", "marital", "education",   
 "default", "balance", "housing", "loan",   
 "whitecollar", "bluecollar", "divorced", "married",  
 "primary", "secondary", "tertiary"))  
  
# examine the structure of the full bank data frame  
print(str(bankfull))

## 'data.frame': 3705 obs. of 16 variables:  
## $ response : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...  
## $ age : int 30 30 59 39 41 39 43 36 20 40 ...  
## $ jobtype : Factor w/ 3 levels "White Collar",..: 3 1 2 2 1 2 1 2 3 1 ...  
## $ marital : Factor w/ 3 levels "Divorced","Married",..: 2 2 2 2 2 2 2 2 3 2 ...  
## $ education : Factor w/ 4 levels "Primary","Secondary",..: 1 3 2 2 3 2 2 3 2 3 ...  
## $ default : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ balance : int 1787 1476 0 147 221 9374 264 1109 502 194 ...  
## $ housing : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 1 1 1 ...  
## $ loan : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...  
## $ whitecollar: num 0 1 0 0 1 0 1 0 0 1 ...  
## $ bluecollar : num 0 0 1 1 0 1 0 1 0 0 ...  
## $ divorced : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ married : num 1 1 1 1 1 1 1 1 0 1 ...  
## $ primary : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ secondary : num 0 0 1 1 0 1 1 0 1 0 ...  
## $ tertiary : num 0 1 0 0 1 0 0 1 0 1 ...  
## NULL

print(head(bankfull))

## response age jobtype marital education default balance housing  
## 1 No 30 Other/Unknown Married Primary No 1787 No  
## 4 No 30 White Collar Married Tertiary No 1476 Yes  
## 5 No 59 Blue Collar Married Secondary No 0 Yes  
## 8 No 39 Blue Collar Married Secondary No 147 Yes  
## 9 No 41 White Collar Married Tertiary No 221 Yes  
## 11 No 39 Blue Collar Married Secondary No 9374 Yes  
## loan whitecollar bluecollar divorced married primary secondary tertiary  
## 1 No 0 0 0 1 1 0 0  
## 4 Yes 1 0 0 1 0 0 1  
## 5 No 0 1 0 1 0 1 0  
## 8 No 0 1 0 1 0 1 0  
## 9 No 1 0 0 1 0 0 1  
## 11 No 0 1 0 1 0 1 0

# select subset of variables for input to cluster analysis  
data\_for\_clustering <- subset(bankfull,  
 select = c("age",   
 "whitecollar", "bluecollar",   
 "divorced", "married",  
 "primary", "secondary", "tertiary"))

At this point, we begin the cluster analysis. In preparation, the script sets up a pdf file to receive the graphical output, rather than simply directing the output to the console or plots window.

This part of the script also specifies several alternative clustering solutions. The step may take 10 minutes or more to complete

pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_cluster\_search.pdf",  
 width = 8.5, height = 11)  
min\_clusters <- 2  
max\_clusters <- 20  
# evaluate alternative numbers of clusters/segments  
# we use the average silhouette width as a statistical criterion  
evaluation\_vector <- NULL # initialize evaluation vector   
# selected algorithm is pam (partitioning around medoids)  
# with so many binary variables, manhattan distances seemed   
# to work better than Euclidean distances  
for (number\_of\_clusters in min\_clusters:max\_clusters) {  
 try\_clustering <- pam(data\_for\_clustering, k = number\_of\_clusters,  
 metric = "manhattan", stand = TRUE)  
 evaluation\_vector <- rbind(evaluation\_vector,  
 data.frame(number\_of\_clusters,   
 average\_silhouette\_width =   
 try\_clustering$silinfo$avg.width))  
 # show plot for this clustering solution  
 plot(try\_clustering) # add this clustering solution to results file   
 }   
dev.off() # close the pdf results file for the clustering solution

## png   
## 2

Having now run through multiple models, we examine the results, looking for average silhouette width > 0.5 and the last big jump in average silhoutte width.

We also create a summary plot comparing the clustering solutions and send it to another pdf.

print(evaluation\_vector)

## number\_of\_clusters average\_silhouette\_width  
## 1 2 0.3619243  
## 2 3 0.4418062  
## 3 4 0.4563305  
## 4 5 0.4787168  
## 5 6 0.4750381  
## 6 7 0.4798570  
## 7 8 0.5018247  
## 8 9 0.5218733  
## 9 10 0.5069539  
## 10 11 0.5008017  
## 11 12 0.5111868  
## 12 13 0.5350198  
## 13 14 0.5497882  
## 14 15 0.5595420  
## 15 16 0.5518343  
## 16 17 0.5595772  
## 17 18 0.5627061  
## 18 19 0.5496746  
## 19 20 0.5667843

# provide a single summary plot for the clustering solutions  
pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_cluster\_summary.pdf",  
 width = 8.5, height = 8.5)  
with(evaluation\_vector, plot(number\_of\_clusters,   
 average\_silhouette\_width))  
dev.off() # close summary results file

## png   
## 2

After running all of this and examining the results, we come in for a closer look at the seven-cluster model, which appears to be best.

seven\_cluster\_solution <- pam(data\_for\_clustering, k = 8,  
 metric = "manhattan", stand = TRUE)  
pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_seven\_cluster\_solution.pdf",  
 width = 8.5, height = 8.5)  
plot(seven\_cluster\_solution)  
dev.off()

## png   
## 2

From the silhouette plot, the first five of the seven clusters appear to be large and well-defined. The script now goes on to select the first 5 clusters.

Then it creates some plots to describe the membership of different clusters.

# add the cluster membership information and select first five  
bankfull$cluster <- seven\_cluster\_solution$clustering  
bankpart <- subset(bankfull, subset = (cluster < 6))  
bankpart$cluster <- factor(bankpart$cluster,  
 labels = c("A", "B", "C", "D", "E"))  
  
# look at demographics across the clusters/segments  
# -----------------  
# age Age in years  
# -----------------  
# examine relationship between age and response to promotion  
with(bankpart, print(by(age, cluster, mean)))

## cluster: A  
## [1] 45.9057  
## --------------------------------------------------------   
## cluster: B  
## [1] 41.6633  
## --------------------------------------------------------   
## cluster: C  
## [1] 43.04649  
## --------------------------------------------------------   
## cluster: D  
## [1] 40.03083  
## --------------------------------------------------------   
## cluster: E  
## [1] 32.29765

pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_age\_lattice.pdf",   
 width = 8.5, height = 11)  
lattice\_plot\_object <- histogram(~age | cluster, data = bankpart,  
 type = "density",   
 xlab = "Age of Bank Client", layout = c(1,5))  
print(lattice\_plot\_object) # responders tend to be older  
dev.off()

## png   
## 2

# -----------------------------------------------------------  
# education  
# Level of education (unknown, secondary, primary, tertiary)  
# -----------------------------------------------------------  
with(bankpart, print(table(cluster, education)))

## education  
## cluster Primary Secondary Tertiary Unknown  
## A 509 0 0 0  
## B 0 0 594 0  
## C 0 826 0 56  
## D 0 485 0 34  
## E 0 354 0 29

# ---------------------------------------------------------------  
# job status using jobtype  
# White Collar: admin., entrepreneur, management, self-employed   
# Blue Collar: blue-collar, services, technician  
# Other/Unknown  
# ---------------------------------------------------------------  
with(bankpart, print(table(cluster, jobtype)))

## jobtype  
## cluster White Collar Blue Collar Other/Unknown  
## A 70 313 126  
## B 470 81 43  
## C 0 768 114  
## D 488 0 31  
## E 0 314 69

# ----------------------------------------------  
# marital status  
# Marital status (married, divorced, single)  
# [Note: ``divorced'' means divorced or widowed]  
# ----------------------------------------------  
with(bankpart, print(table(cluster, marital)))

## marital  
## cluster Divorced Married Single  
## A 0 449 60  
## B 0 594 0  
## C 0 882 0  
## D 0 380 139  
## E 0 0 383

# look at bank client history across the clusters/segments  
  
# -----------------------------------------  
# default Has credit in default? (yes, no)  
# -----------------------------------------  
with(bankpart, print(table(cluster, default)))

## default  
## cluster No Yes  
## A 502 7  
## B 590 4  
## C 868 14  
## D 508 11  
## E 372 11

# ------------------------------------------  
# balance Average yearly balance (in Euros)  
# ------------------------------------------  
with(bankpart, print(by(balance, cluster, mean)))

## cluster: A  
## [1] 1446.106  
## --------------------------------------------------------   
## cluster: B  
## [1] 1882.47  
## --------------------------------------------------------   
## cluster: C  
## [1] 1273.861  
## --------------------------------------------------------   
## cluster: D  
## [1] 1331.609  
## --------------------------------------------------------   
## cluster: E  
## [1] 1018.661

pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_blance\_lattice.pdf",   
 width = 8.5, height = 11)  
lattice\_plot\_object <- histogram(~balance | cluster, data = bankpart,  
 type = "density", xlab = "Age of Bank Client",   
 layout = c(1,5))  
print(lattice\_plot\_object) # responders tend to be older  
dev.off()

## png   
## 2

# ------------------------------------  
# housing Has housing loan? (yes, no)  
# ------------------------------------  
with(bankpart, print(table(cluster, housing)))

## housing  
## cluster No Yes  
## A 225 284  
## B 310 284  
## C 322 560  
## D 208 311  
## E 194 189

# ----------------------------------  
# loan Has personal loan? (yes, no)  
# ----------------------------------  
with(bankpart, print(table(cluster, loan)))

## loan  
## cluster No Yes  
## A 435 74  
## B 509 85  
## C 719 163  
## D 428 91  
## E 328 55

# ----------------------------------------------------  
# response Response to term deposit offer (yes, no)  
# ----------------------------------------------------  
with(bankpart, print(table(cluster, response)))

## response  
## cluster No Yes  
## A 473 36  
## B 535 59  
## C 824 58  
## D 489 30  
## E 333 50

pdf(file = "C:/Users/Rob/Box Sync/My R Work/BUS256/fig\_finding\_new\_customers\_response\_mosaic.pdf",   
 width = 8.5, height = 8.5)  
mosaic( ~ response + cluster, data = bankpart,  
 labeling\_args = list(set\_varnames = c(response = "Response to Term Deposit Offer",   
 cluster = "Segment Membership")),  
 highlighting = "response",  
 highlighting\_fill = c("cornsilk","violet"),  
 rot\_labels = c(left = 0, top = 0),  
 pos\_labels = c("center","center"),  
 offset\_labels = c(0.0,0.6))  
dev.off()

## png   
## 2

# compute percentage of yes responses to term deposit offer  
response\_table <- table(bankpart$cluster, bankpart$response)  
cat("\nPercentage Responses\n")

##   
## Percentage Responses

for (i in 1:5)   
 cat("\n", toupper(letters[i]),   
 round(100 \* response\_table[i,2] /   
 sum(response\_table[i,]), digits = 1))

##   
## A 7.1  
## B 9.9  
## C 6.6  
## D 5.8  
## E 13.1

# note the percentage of the customers receiving offers  
# for the first time falling into each of the clusters/segments  
# A = 1 ... E = 5 ...  
print(round(100 \* table(bankfull$cluster) / nrow(bankfull), digits = 1))

##   
## 1 2 3 4 5 6 7 8   
## 13.7 16.0 23.8 14.0 10.3 3.9 8.1 10.1