Cluster Analysis in R

Erin Shellman

June 1, 2015

Contents

Introduction **1**

Read in data **1**

Preprocessing **1**

K-means clustering **2**

Hierarchical Clustering **4**

Introduction

This week we'll explore the product catalog a little more closely. We can explore product similarities using clustering and potentially use the resulting clusters as a basis for product recommendation.

Read in data

Just read in the product catalog this time:

```
library(dplyr)
library(ggplot2)

# read in the product catalog
catalog = read.delim('product_catalog.tsv',
                     header = TRUE,
                     sep = '\t',
                     quote = '')

menace = catalog[catalog$labels == 5335, ]
data_mining = catalog[catalog$labels == 22025, ]
```

Preprocessing

Most clustering techniques work best when the data are centered and scaled. Recall that a variable is centered when you subtract its mean from each observation. A variable is scaled when you divide each observation by the variable standard deviation. When we center and scale a variable, the resulting variable follows a z-distribution with mean = 0 and sd = 1.

```
# set the seed
set.seed(100)

# remove rows containing missing data
catalog = na.omit(catalog)

# most products are books, music, video or DVDs, so lets focus on those
table(catalog$group)
```

```
##
##              Baby Product         Book           CE          DVD
##            0            1       188014            3         9554
##        Music     Software          Toy        Video  Video Games
##        48993            2            3        12597            1
```

```
# subset
sub =
  catalog %>%
    filter(group == 'Book' | group == 'DVD' |
           group == 'Music' | group == 'Video') %>%
    select(labels, avg_rating, downloaded, reviews_count, salesrank, group)

head(sub)
```

```
##   labels avg_rating downloaded reviews_count salesrank group
## 1      1        5.0          2             2    396585  Book
## 2     10        4.0          6             6    220379  Book
## 3    100        0.0          0             0    783690  Book
## 4   1000        5.0          1             1    497795  Book
## 5  10000        0.0          0             0   1544234  Book
## 6 100000        4.5          2             2    750978  Book
```

Before we can scale the data, we need to convert the product group from a factor into a numeric value:

```
# convert the group into dummy variables
group_dummies = model.matrix(~ factor(sub$group) - 1)
colnames(group_dummies) = c('book', 'dvd', 'music', 'video')

# append the dummy variables back on and drop the 'group' variable
sub = cbind(sub, group_dummies)
sub$group = NULL

# scale the data
scaled = as.data.frame(scale(select(sub, -labels)))
scaled$labels = sub$labels
```

K-means clustering

We have 4 product types, so maybe K = 4 is a good place to start?

```r
# run k-means clustering
kmeans_cluster = kmeans(select(scaled, -labels), 4)

# check what attributes are in the kmeans object
attributes(kmeans_cluster)
```

```
## $names
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
##
## $class
## [1] "kmeans"
```

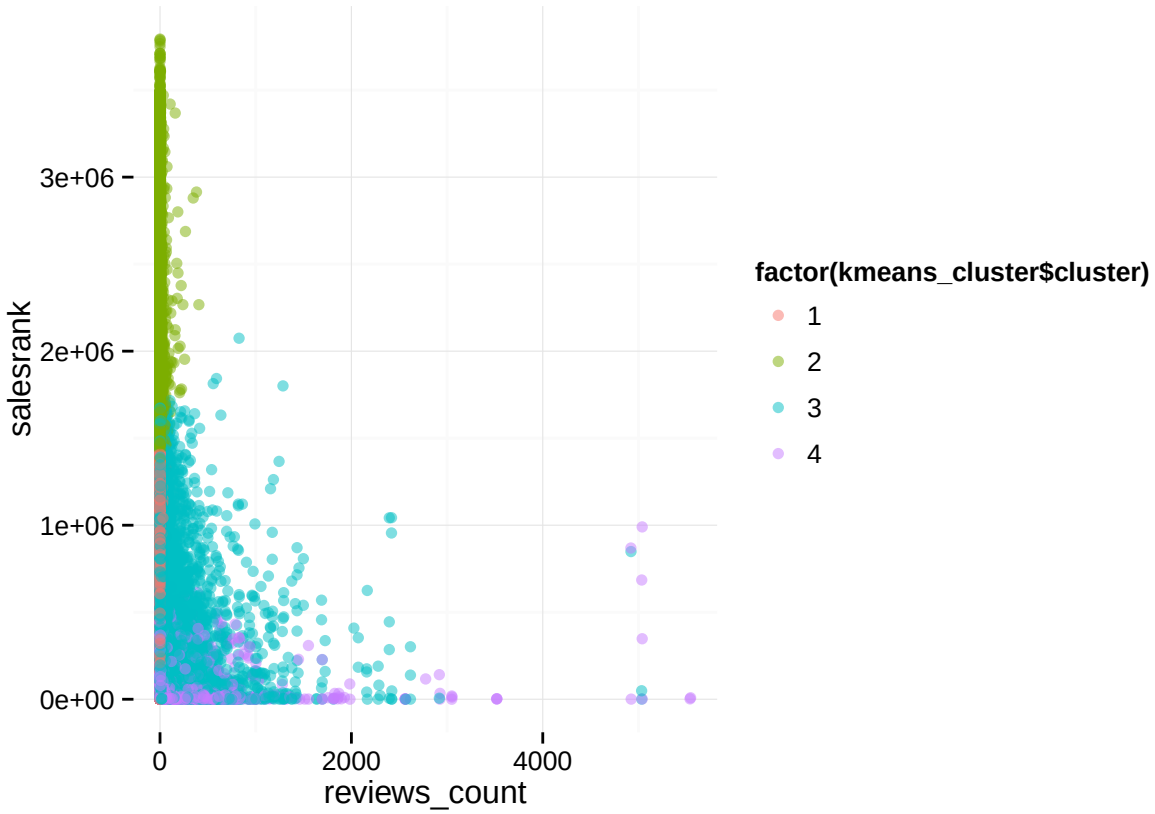
```r
# Find which cluster the observations belong to
head(kmeans_cluster$cluster, 10)
```

```
##  1  2  3  4  5  6  7  8  9 10
##  3  3  1  3  2  3  3  4  4  4
```

```r
# centers
kmeans_cluster$centers
```

```
##   avg_rating  downloaded reviews_count  salesrank       book        dvd
## 1 -1.5223336 -0.19704537   -0.18997687  0.3151861  0.6151389 -0.1956438
## 2 -1.1686826 -0.18212363   -0.17582944  2.8550430  0.6151389 -0.1956438
## 3  0.6005555  0.04049515    0.03966189 -0.1262116  0.6151389 -0.1956438
## 4  0.2132531  0.10024698    0.09562052 -0.6590578 -1.6252334  0.5169024
##        music      video
## 1 -0.4828208 -0.2260323
## 2 -0.4828208 -0.2260323
## 3 -0.4828208 -0.2260323
## 4  1.2756411  0.5971907
```

```r
# plot 4 clusters
ggplot(sub,
       aes(x = reviews_count,
           y = salesrank,
           color = factor(kmeans_cluster$cluster))) +
  geom_point(alpha = 0.50) +
  theme_minimal()
```



Hierarchical Clustering

```
# Don't be a menace
menace = scaled[scaled$labels == 5335, ]

# subset to videos only
videos =
  scaled %>%
    filter(sub$video == 1) %>%
    sample_n(30)
videos = unique(rbind(menace, videos))

# compute the euclidean distance
euclidean = dist(select(videos, -labels), method = 'euclidean')

# attributes
attributes(euclidean)
```

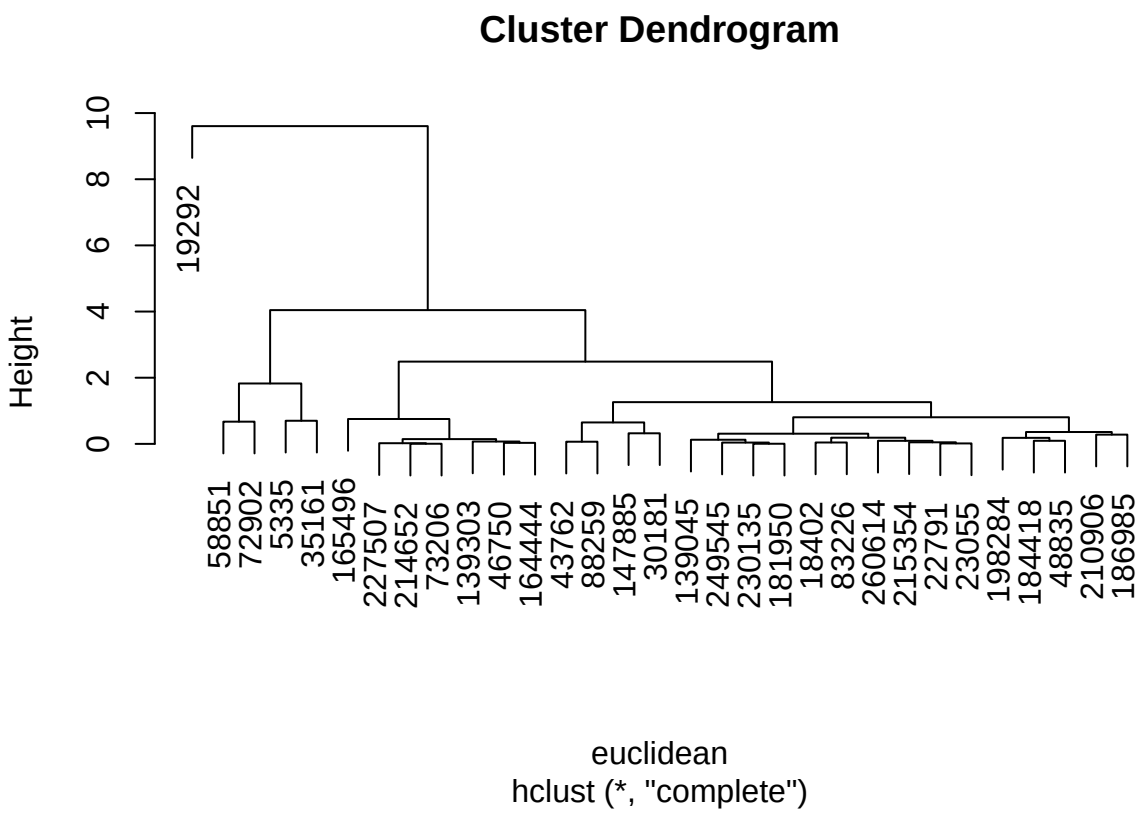
```
## $Size
## [1] 31
##
## $Labels
##  [1] "207881" "5903"   "6094"   "10233"  "4664"   "6883"   "2144"
##  [8] "7870"   "11107"  "3530"   "5017"   "9599"   "8421"   "2576"
## [15] "4500"   "4524"   "8686"   "6742"   "8942"   "6772"   "9421"
```

```
## [22] "5284"   "2156"   "9687"   "11090"  "6904"   "3492"   "6139"
## [29] "11672"  "4383"   "11992"
##
## $Diag
## [1] FALSE
##
## $Upper
## [1] FALSE
##
## $method
## [1] "euclidean"
##
## $call
## dist(x = select(videos, -labels), method = "euclidean")
##
## $class
## [1] "dist"
```

```r
# hierarchical clustering
hier = hclust(euclidean)

# label by id
hier$labels = videos$labels

# plot dendrogram
plot(hier)
```



```
# look up nearby videos
catalog[catalog$labels == 35161, ]
```

```
##        labels avg_rating downloaded group reviews_count salesrank   title
## 190072  35161        4.5         95 Video            95      6511 Ponette
```

```
catalog[catalog$labels == 58851, ]
```

```
##        labels avg_rating downloaded group reviews_count salesrank  title
## 216393  58851          3        152 Video           152       476 Sphere
```

```
catalog[catalog$labels == 72902, ]
```

```
##        labels avg_rating downloaded group reviews_count salesrank
## 232006  72902        3.5        183 Video           183     48110
##                     title
## 232006 Jason Goes to Hell
```