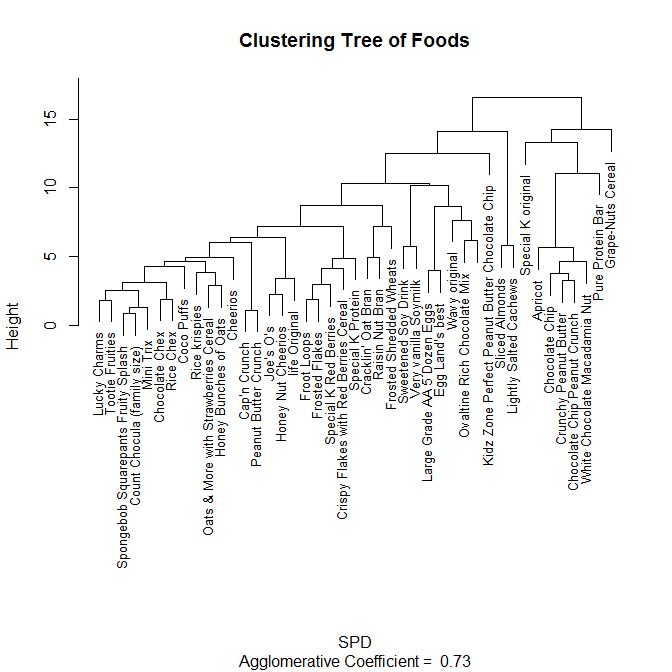
Nutritional Content Food Product Clustering

Ignacio Faria

March 12, 2017

Food products differ greatly and one way to assess similarities is by clustering. The following data is an analysis of 49 different food products, described by 52 *per serving* attributes (i.e. caffeine, carbohydrates, cholesterol, etc.). The data was imported from openfoodfacts.org and cleaned.

The data was modeled into an Agnes dendrogram. Because it is a hierarchical agglomerative clustering model, we can see where the clusters converge in similarity.



In order to determine the optimal number of clusters for the k-means algorithm to produce, a series of indexing criterion must be observed. The following determines the optimal number of clusters, by checking how well each method clusters the data, from 2 to 20 clusters.

#   
list.m= c("kl", "ch", "hartigan","mcclain", "gamma", "gplus",  
 "tau", "dunn", "sdindex", "sdbw", "cindex", "silhouette",  
 "ball","ptbiserial", "gap","frey")  
tab.bk = as.data.frame(matrix(nrow =length(list.m), ncol=2))  
for(i in 1:length(list.m)){  
  
nb = NbClust(SPD, min.nc = 2, max.nc = 20,   
 method = "complete",   
 index =list.m[i])  
tab.bk[i,2] = nb$Best.nc[1]  
tab.bk[i,1] = list.m[i]  
}  
tab.bk

## V1 V2  
## 1 kl 2  
## 2 ch 20  
## 3 hartigan 8  
## 4 mcclain 2  
## 5 gamma 12  
## 6 gplus 12  
## 7 tau 7  
## 8 dunn 7  
## 9 sdindex 7  
## 10 sdbw 20  
## 11 cindex 8  
## 12 silhouette 19  
## 13 ball 3  
## 14 ptbiserial 9  
## 15 gap 2  
## 16 frey 1

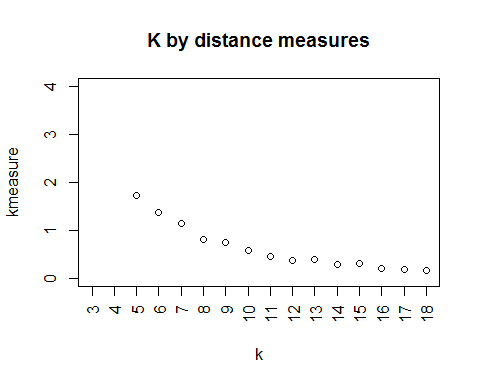
The fewer the clusters, the easier it is to identify differences. I chose the sdindex, since it is the most recent method as of the year 2000 and 7 clusters seems reasonable.

K-means performance can be measured by *total sum of squares within* and *between* clusters. Using this as a metric , their performance was plotted (the lower the better).

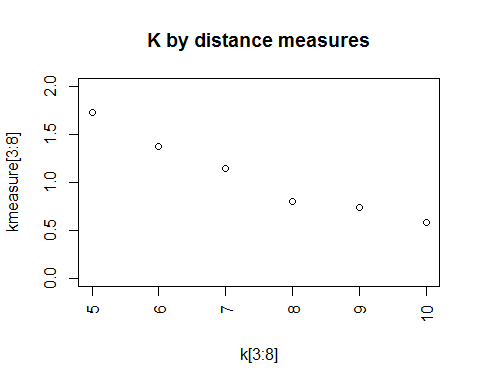
k<-seq(3,18,1)  
kmeasure<-numeric(length(k))  
for (i in seq\_along(k)){SPDkmeans <- kmeans(SPD, i, nstart = 1)  
kmeasure[i]<-SPDkmeans$tot.withinss/SPDkmeans$betweenss}  
kmeasure

## [1] -4.128990e+14 5.074622e+00 1.726804e+00 1.369938e+00 1.141265e+00  
## [6] 8.017738e-01 7.431518e-01 5.815940e-01 4.488250e-01 3.716590e-01  
## [11] 3.857969e-01 2.808377e-01 3.164742e-01 2.107615e-01 1.775678e-01  
## [16] 1.717943e-01

plot(k, kmeasure, ylim=c(0,4),xaxt="n", main="K by distance measures")  
axis(1, at = seq(1, 18, by = 1), las=2)



plot(k[3:8], kmeasure[3:8], ylim=c(0,2),xaxt="n", main="K by distance measures")  
axis(1, at = seq(5, 10, by = 1), las=2)



#k appears best at 8

By this measure, it would appear that 8 clusters is the best; it is sufficiently small while having the largest drop in this metric. Previous analysis showed that 8 was not, however, the best k. It separated cereals into another cluster, added nuts into the dairy cluster, and was subjectivly less informative than 7 clusters.

The following table shows how the products were grouped.

SPDkmeans <- kmeans(SPD, 7, nstart = 1)

sep<-SPDkmeans[["cluster"]]  
sep<-as.data.frame(sep)  
sep<-tibble::rownames\_to\_column(sep)  
str(sep)

## 'data.frame': 43 obs. of 2 variables:  
## $ rowname: chr " Lucky Charms" " Rice krispies" " Froot Loops" " Cap'n Crunch" ...  
## $ sep : int 3 3 3 3 5 2 3 3 4 3 ...

colnames(sep)<-c("Food","Group")  
tab1<-arrange(sep,Group)  
tab1

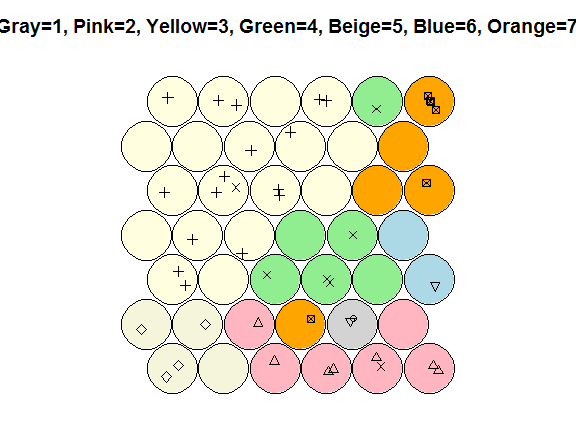
## Food Group  
## 1 Pure Protein Bar 1  
## 2 Sweetened Soy Drink 2  
## 3 Large Grade AA 5 Dozen Eggs 2  
## 4 Wavy original 2  
## 5 Ovaltine Rich Chocolate Mix 2  
## 6 Very vanilla Soymilk 2  
## 7 Egg Land's best 2  
## 8 2  
## 9 Lucky Charms 3  
## 10 Rice krispies 3  
## 11 Froot Loops 3  
## 12 Cap'n Crunch 3  
## 13 Peanut Butter Crunch 3  
## 14 Oats & More with Strawberries Cereal 3  
## 15 Honey Bunches of Oats 3  
## 16 Spongebob Squarepants Fruity Splash 3  
## 17 Cheerios 3  
## 18 Mini Trix 3  
## 19 Frosted Flakes 3  
## 20 Count Chocula (family size) 3  
## 21 Tootie Fruities 3  
## 22 Chocolate Chex 3  
## 23 Rice Chex 3  
## 24 Coco Puffs 3  
## 25 Joe's O's 4  
## 26 Cracklin' Oat Bran 4  
## 27 Raisin Nut Bran 4  
## 28 Honey Nut Cheerios 4  
## 29 Frosted Shredded Wheats 4  
## 30 Grape-Nuts Cereal 4  
## 31 life Original 4  
## 32 Special K Protein 5  
## 33 Special K original 5  
## 34 Crispy Flakes with Red Berries Cereal 5  
## 35 Special K Red Berries 5  
## 36 Sliced Almonds 6  
## 37 Lightly Salted Cachews 6  
## 38 Apricot 7  
## 39 Kidz Zone Perfect Peanut Butter Chocolate Chip 7  
## 40 Chocolate Chip 7  
## 41 Crunchy Peanut Butter 7  
## 42 Chocolate Chip Peanut Crunch 7  
## 43 White Chocolate Macadamia Nut 7

The clusters appear to be well defined:

* Group 1 is just a protein bar
* Group 2 contains dairy products, eggs, and soy milk
* Group 3 is high-sugar cereals
* Group 4 is wheat cereals
* Group 5 is healthy cereals
* Group 6 is nuts
* Group 7 high fatty content foods, with the exception of Apricots.

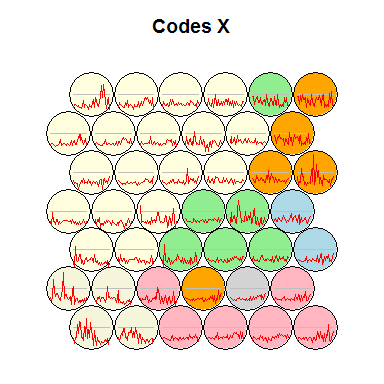
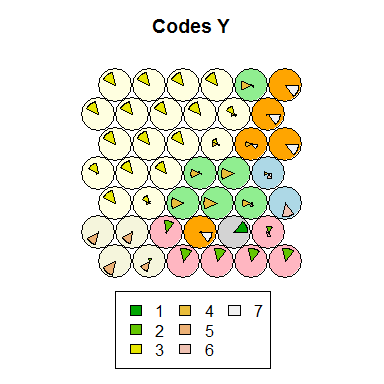
The following Kohonen Self Organizing Map gives a 2 dimentional view of these groupings. This is a supervised version of the KSOM, that measures the distance of an object as a sum of separate distances for X and Y spaces.

library(kohonen)  
kohmap2 <- xyf(SPD, classvec2classmat(tab1$Group),  
 grid = somgrid(6, 7, "hexagonal"), rlen=100)  
xyfpredictions2 <- classmat2classvec(predict(kohmap2)$unit.predictions)  
  
bgcols <- c("lightgray", "lightpink", "lightyellow","lightgreen", "beige", "lightblue", "orange")  
  
plot(kohmap2, type="mapping",   
 pchs = tab1[["Group"]], bgcol = bgcols[as.integer(xyfpredictions2)],   
 main = "Gray=1, Pink=2, Yellow=3, Green=4, Beige=5, Blue=6, Orange=7")



The plot above shows the effectiveness of the clustering, with only group 4 and group 7 (green and orange) having only 1 non-adjacent member. My guess is that the outlying member is apricots (group 7) because it has a high sugar content and is close to cereal, and the other outling group 4 member is a cereal with a high fat content.

plot(kohmap2, type="codes", main = c("Codes X", "Codes Y"),bgcol = bgcols[as.integer(xyfpredictions2)])

These code plots describe the vectors for X and Y. The lines on the Codes X plot show similarities within groups, while the Codes Y plot show the main uniting attributes that define each group.