Article Database Classification Tool: using pattern recognition to identify data entry errors

Marie Silvestre

October 4th, 2017

Objective

’Identify data entry errors in the article database.

Summary:

The article database describes all the characteristics for every article :collection, sub-collection, material, type of movement… Each time a new article is created, its features are filled in manually.

Each family of article shares the same charateristics (with a few exceptions). A clustering algorithm will allow us to cluster those families, then identify the articles the furthest from the center of the cluster as being potential data errors.

Steps: 1. Create a weighted dissimilarity matrix from the article database. 2. Use agglomerative hierarchical clustering to create clusters. 3. Validate clustering model. 4. Identify outliers by the distance at which they first merge into a cluster

Project:

Data preparation:

```r
##Load relevant packages
library(caret)
library(dplyr)
library(knitr)
library(factoextra)
library(xlsx)
library(cluster)
library(kableExtra)
```

The article database presents as follow:

```r
##Load data in variable
Article_Data<-read.csv("Database Article.csv",header=TRUE,sep=";")
#Select only the columns we are interested in:
Article_Data<-select(Article_Data,Article,Serie.desc,Sub.Serie.desc,GENDER,
CASE_SIZE,TYPE,MOVEMENT,STATUS,WATCH_MATERIAL,LAUNCH_YEAR,BEZEL,BEZEL_MATERIAL,
BRACELET_MATERIAL)
##Give an overview of the data
intro<-head(Article_Data)
kable(intro,"latex") %>% kable_styling(latex_options=c("scale_down"))
```

The data set is fully categorical.

Article	Serie.desc	Sub.Serie.desc	GENDER	CASE_SIZE	TYPE	MOVEMENT	STATUS	WATCH_MATERIAL	LAUNCH_YEAR	BEZEL	BEZEL_MATERIAL	BRACELET_MATERIAL
WN5141.BG0351	2000	2000 Exclusive Q Diamonds	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Gold 3N
WN5140.FC8156	2000	2000 Exclusive Q Chrono S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Leather Alligator
WN5140.FC8155	2000	2000 Exclusive Q S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Leather Alligator
WN5140.CI8155	2000	2000 Exclusive Q Chrono S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Without Bracelet
WN5140.BG0351	2000	2000 Exclusive Q S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Gold 3N
WN2312.BA0360	2000	2000 A	Ladies	3	Watch	Calibre 3 Automatic	Old	Steel	2003	Turning Bezel	Steel	Steel

Step 1: Create a weighted dissimilarity matrix using the daisy() function.

We chose the daisy() function since all the data are categorical and it allows us to weight the different features.

The weight will be attributed to each feature in order to separate the clusters by the following order of importance: Collection, sub collection, Gender, Size, Type, Movement,Status, Watch Material, Year/Bezel/Bezel Material/Precious**.** These four features will have the same weight since they may vary inside the same family.

This strategy will allow us to cluster by family of article. (family=articles that should share the same features)

Since all our data is categorical, we have to use a distance metric that can handle this data type. We will be using "Gower" distance: for each variable type, a distance metric that works well with that type is used and scaled to fall between 0 and 1.Then, the final distance is calculated via a linear combination using our specified weights.

```
##We need to convert the Launch Year variable to factors so it won't be recognized as an interval
##scaled variable by the function daisy()
Article_Data$LAUNCH_YEAR<-as.factor(Article_Data$LAUNCH_YEAR)
##Reorder columns by importance
Article_Data<-Article_Data[c("Article","Serie.desc","Sub.Serie.desc","GENDER","CASE_SIZE",
                             "TYPE","MOVEMENT","STATUS","WATCH_MATERIAL","LAUNCH_YEAR","BEZEL",
                             "BEZEL_MATERIAL","BRACELET_MATERIAL")]
#Creating the weight vector
weight<-c(16,14,12,10,8,6,4,2,2,2,2,1)
Article_Data<-data.frame(Article_Data)
#We remove the Article name before applying the daisy()
diss_Mat<-daisy(Article_Data[,-1],metric="gower", weights=weight)
```

Before going further, we will check whether the matrice works properly by extracting the two most similar entries, and the two most dissimilar entries.

```
diss_mat_2<-as.matrix(diss_Mat)
kable(Article_Data[which(diss_mat_2==min(diss_mat_2[diss_mat_2
kable_styling(latex_options=c("scale_down"))
```

	Article	Serie.desc	Sub.Serie.desc	GENDER	CASE_SIZE	TYPE	MOVEMENT	STATUS	WATCH_MATERIAL	LAUNCH_YEAR	BEZEL	BEZEL_MATERIAL	BRACELET_MATERIAL
4	WN5140.CI8155	2000	2000 Exclusive Q Chrono S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Without Bracelet
2	WN5140.FC8156	2000	2000 Exclusive Q Chrono S&G	Gents	1	Watch	Calibre 7 COSC	Old	Gold 3N	2002	Turning Bezel	Gold 3N	Leather Alligator

```
#We find twice the same ref with a different bracelet --> good
#output most dissimilar pair
kable(Article_Data[which(diss_mat_2==max(diss_mat_2[diss_mat_2
  kable_styling(latex_options=c("scale_down"))
```

	Article	Serie.desc	Sub.Serie.desc	GENDER	CASE_SIZE	TYPE	MOVEMENT	STATUS	WATCH_MATERIAL	LAUNCH_YEAR	BEZEL	BEZEL_MATERIAL	BRACELET_MATERIAL
3287	CV2A1S.BA0799	Carrera	Carrera A Chrono 43mm	Gents	A	Chrono	Calibre 16 Automatic	Current	Steel & Ceramic	2015	Fixed Bezel	Steel & Ceramic	Steel
6	WN2312.BA0360	2000	2000 A	Ladies	3	Watch	Calibre 3 Automatic	Old	Steel	2003	Turning Bezel	Steel	Steel

Our dissimilarity matrix seems to be functioning correctly.

Step 2: Create a hierarchical clustering with the agnes() function.

The agglomerative hierarchical clustering algorithm does not require for the number of clusters' information to be provided, which is why we chose it. We will use the "average linkage method" to calculate distance, since the outliers will only be slightly different than the rest of the cluster as a whole.

```
#Cluster with agnes method "average"
Clust<-agnes(diss_Mat,method="average",diss=TRUE)
```

Step 3: Clustering evaluation

We will use internal cluster validation statistics to evaluate our model: specifically the agglomerative coefficient (it can be compared to the silhouette coefficient).

Agglomerative coefficient

The agglomerative coefficient describes the strength of the clustering structure. Like the silhouette coefficient, if it is close to 1, then the data has been well clustered.

```
Clust$ac
```

```
## [1] 0.9899161
```

According to the AC, our data set is well clustered.

Step 4: Identifying outliers using the height metric

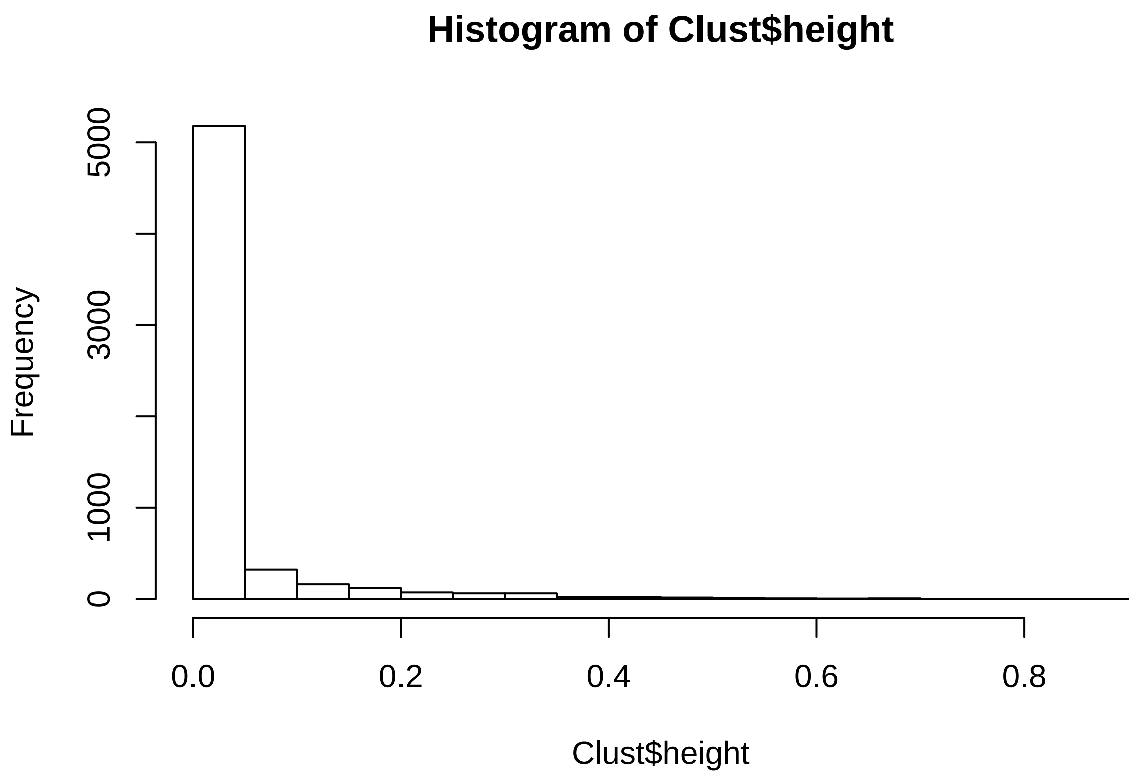
The "height" metric displays when an observation merges into a cluster. Outliers will merge later, hence their height will be greater.

First we wil look at a few metrics:

```
summary(Clust$height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.03324 0.02532 0.85031
```

```
hist(Clust$height)
```



We can see from the histogram that the great majority of entries have an height < 0.1, therefore, if we select all observations with an height > 0.1, we should catch all the outliers, and find all the data entry errors.

We will sort them by cluster height, to start the correction with the worsts offenders. We will then export them in an excel file.

```r
#To facilitate the correction, we will input the "order" number in the outlier file:
#it will allow us to identify the clusters
Article_Data$order<-Clust$order
Article_Data<-Article_Data[order(-Clust$height),]
Outliers<-Article_Data[Clust$height > 0.1,]
#Group clusters together
Article_Data<-Article_Data[order(-Article_Data$order),]
#Now we output all the outliers in an excel table for further review
write.xlsx(Outliers,"outliers.xlsx")
write.xlsx(Article_Data,"Database article avec groupes.xlsx")
```

The result of this programm is an excel file with all the potential data entry errors in the Article database.