Supplemental Material QEWOM

Juan C. Correa
University of Economics, Prague

6/22/2020

This supplemental material aims at providing an easy step-by-step, reproducible guide for quantifying the emergence, self-organization, and complexity of customers' comments. First, we upload our raw data set as follows (don't forget to replace the folder address if you download our data set from Github and save it somewhere in your hard disk).

```
load("/home/juan/Comments.RData")
```

Now, we will use the quanteda package for generating the corpus and conducting usual preprocessing such as removing stopwords, and incorporate metadata that associates the category of the restaurant the commercial name of the provider and the quantitative rating of each comment.

```
library(quanteda)
```

```
## Package version: 2.0.1
```

```
## Parallel computing: 2 of 4 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':
##
##     View
```

```
my_corpus <- corpus(UserComments$text)
mycorpus <- data.frame(summary(my_corpus, n = nrow(UserComments)))
head(summary(my_corpus))
```

```
##    Text Types Tokens Sentences
## 1 text1    18     22         1
## 2 text2    29     35         4
## 3 text3    27     34         1
## 4 text4     1      1         1
## 5 text5     6      8         2
## 6 text6     1      1         1
```

```
docvars(my_corpus, "Category") <- UserComments$Category
docvars(my_corpus, "Provider") <- UserComments$Provider
docvars(my_corpus, "Rating") <- UserComments$Rating
```

Let's include in our initial dataset the number of different words per comment (Types), the total number of words per comment (Tokens) and the number of sentences (Sentences)

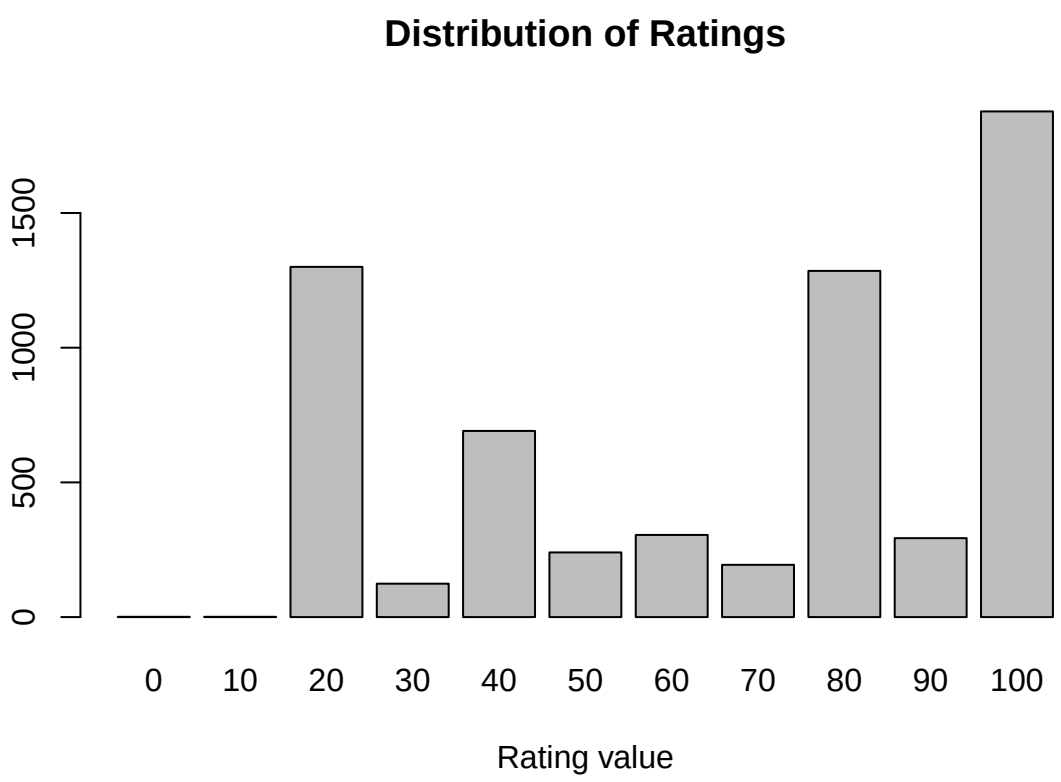
```
UserComments$Types <- mycorpus$Types
UserComments$Tokens <- mycorpus$Tokens
UserComments$Sentences <- mycorpus$Sentences
```

Here, we see that we have six categories, and here, we see the distribution of discrete ratings

```
unique(UserComments$Provider)
```

```
##  [1] "ArrozyPastasAlWok"           "Asiatika"
##  [3] "CadadelSushi"                "Itamae"
##  [5] "MrLee"                       "OrientalSushi"
##  [7] "Shitake"                     "SrWok"
##  [9] "SushiExpress"                "Yokomo"
## [11] "BurgerKing"                  "Charlies"
## [13] "DelRodeoHambuergesas"        "DondeBeto"
## [15] "LaCorraleja"                 "McDonalds"
## [17] "PaSaborear"                  "Presto"
## [19] "Randys"                      "SaborCartagenero"
## [21] "Cali Vea"                    "Century Broaster"
## [23] "Frisby"                      "kfc"
## [25] "Kokoriko"                    "La Brasa Roja"
## [27] "Pollo Pollito Pollote"       "Speed Broaster"
## [29] "Surtidora de Aves"           "Surtidora de Aves Ober"
## [31] "El Alcoholimetro"            "ElCorreCorre"
## [33] "El Correo de la Noche"       "El Paraiso"
## [35] "El Surtidor de la Noche"     "La Licorera a Domicilio"
## [37] "Licmaos"                     "S24-7"
## [39] "SpeedDrinks"                 "SuperElParaiso"
## [41] "BBQ Express"                 "Casa Parrilla Plaza Americas"
## [43] "DelRodeo"                    "DonJediondo"
## [45] "ElParque"                    "Griego"
## [47] "LaCampiña"                   "La Sazón de la Noche"
## [49] "Mazorcada Polo"              "PPC"
## [51] "Aprissa"                     "Colombia and Pizza"
## [53] "Jenos Pizza"                 "La Mona Pizza"
## [55] "One Pizzeria"                "Papa Johns"
## [57] "Pizza Hut"                   "RedBox"
## [59] "Telepizza"                   "Viva la pizza"
```

```
Ratings <- table(UserComments$Rating)
barplot(Ratings, main="Distribution of Ratings",
        xlab="Rating value")
```



Let's proceed by creating a similarity matrix for each restaurant category. In these matrices we basically estimate the Jaccard index for pairwise comparison of comments. But before that, we need to customize our stopwords like this:

```
spanishstopwords <- c("q", stopwords("spanish"))
library(quanteda)
CommentsAsian <- dfm(corpus_subset(
  my_corpus, Category == "Asian"),
  remove_numbers = TRUE,
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE)

Asian <- textstat_simil(
  CommentsAsian,
  margin = "documents",
  method = "jaccard")
Asiandf <- data.frame(as.matrix(Asian))
Asiandf[is.na(Asiandf)] <- 0
Asian <- data.frame(
  jaccard = Asian[lower.tri(Asian, diag = FALSE)])
```

Now, we can apply a Gaussian finite mixture model fitted by EM algorithm, like this:

```
library(mclust)
```

```
## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

```
fit <- Mclust(Asiandf)
summary(fit)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
##
##  log-likelihood    n    df     BIC     ICL
##         2125386 1149 10350 4177840 4177839
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
## 177 596 142  44  42  36  36  63  13
```

```
clasificados <- data.frame(fit$classification)
names(clasificados)[1] <- "classification"
clasificados$Category <- "Asian"
```

As a result, we can see that the content of customers' word-of-mouth can be classified into nine different content-wise categories. Let's proceed with the rest of restaurant categories to see if this data-driven classification holds.

```
CommentsBurgers <- dfm(
  corpus_subset(my_corpus,
                Category == "Burgers"),
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE,
  remove_numbers = TRUE)
Burgers <- textstat_simil(
  CommentsBurgers,
  margin = "documents",
  method = "jaccard")
Burgersdf <- data.frame(as.matrix(Burgers))
Burgersdf[is.na(Burgersdf)] <- 0
Burgers <- data.frame(jaccard = Burgers[lower.tri(Burgers, diag = FALSE)])
fit2 <- Mclust(Burgersdf)
summary(fit2)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
```

```
##
##  log-likelihood    n    df     BIC     ICL
##         2075241 1158 10431 4076897 4076897
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
## 583  42  58 125  84  35  92  87  52
```

```
clasificados2 <- data.frame(fit2$classification)
names(clasificados2)[1] <- "classification"
clasificados2$Category <- "Burgers"

CommentsChicken <- dfm(
  corpus_subset(my_corpus,
                Category == "Chicken"),
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE,
  remove_numbers = TRUE)
Chicken <- textstat_simil(
  CommentsChicken,
  margin = "documents",
  method = "jaccard")
Chickendf <- data.frame(as.matrix(Chicken))
Chickendf[is.na(Chickendf)] <- 0
Chicken <- data.frame(jaccard = Chicken[lower.tri(Chicken, diag = FALSE)])
fit3 <- Mclust(Chickendf)
summary(fit3)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
##
##  log-likelihood    n    df     BIC     ICL
##         2094591 1167 10512 4114945 4114944
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
## 622 106 140  79  51  39  54  64  12
```

```
clasificados3 <- data.frame(fit3$classification)
names(clasificados3)[1] <- "classification"
clasificados3$Category <- "Chicken"

CommentsBeverage <- dfm(
  corpus_subset(my_corpus,
                Category == "Alcoholic Beverages"),
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE,
  remove_numbers = TRUE)
```

```
Beverage <- textstat_simil(
  CommentsBeverage,
  margin = "documents",
  method = "jaccard")
Beveragedf <- data.frame(as.matrix(Beverage))
Beveragedf[is.na(Beveragedf)] <- 0
Beverage <- data.frame(jaccard = Beverage[lower.tri(Beverage, diag = FALSE)])
fit4 <- Mclust(Beveragedf)
summary(fit4)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
##
##  log-likelihood   n   df     BIC     ICL
##        539205.3 625 5634 1042140 1042140
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
## 296 123  24  31  33  24  50  14  30
```

```
clasificados4 <- data.frame(fit4$classification)
names(clasificados4)[1] <- "classification"
clasificados4$Category <- "Beverages"

CommentsMeat <- dfm(
  corpus_subset(my_corpus,
                Category == "Meat"),
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE,
  remove_numbers = TRUE)
Meat <- textstat_simil(
  CommentsMeat,
  margin = "documents",
  method = "jaccard")
Meatdf <- data.frame(as.matrix(Meat))
Meat <- data.frame(jaccard = Meat[lower.tri(Meat, diag = FALSE)])
Meatdf[is.na(Meatdf)] <- 0
fit5 <- Mclust(Meatdf)
summary(fit5)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
##
##  log-likelihood   n   df     BIC     ICL
##         1435606 976 8793 2810686 2810686
```

```
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
## 427 105  24  77 162  52  53  41  35
```

```
clasificados5 <- data.frame(fit5$classification)
names(clasificados5)[1] <- "classification"
clasificados5$Category <- "Meat"

CommentsPizza <- dfm(
  corpus_subset(my_corpus,
                Category == "Pizzas"),
  remove = spanishstopwords,
  stem = TRUE,
  remove_punct = TRUE,
  remove_numbers = TRUE)
Pizza <- textstat_simil(
  CommentsPizza,
  margin = "documents",
  method = "jaccard")
Pizzadf <- data.frame(as.matrix(Pizza))
Pizzadf[is.na(Pizzadf)] <- 0
Pizza <- data.frame(jaccard = Pizza[lower.tri(Pizza, diag = FALSE)])
fit6 <- Mclust(Pizzadf)
summary(fit6)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EII (spherical, equal volume) model with 9 components:
##
##  log-likelihood    n    df     BIC     ICL
##         2399006 1236 11133 4718750 4718749
##
## Clustering table:
##   1   2   3   4   5   6   7   8   9
##  94  45  42 632 213  58  71  23  58
```

```
clasificados6 <- data.frame(fit6$classification)
names(clasificados6)[1] <- "classification"
clasificados6$Category <- "Pizza"
```

Now, let's merge all datasets as follows, and see empirical distributions of customers' ratings as a function of the nine content-wise categories of word-of-mouth.

```
classification <- do.call("rbind", list(clasificados, clasificados2, clasificados3, clasificados4, clas
UserComments$classification <- classification$classification
table(UserComments$classification)
```

```
##
##    1    2    3    4    5    6    7    8    9
## 2199 1017  430  988  585  244  356  292  200
```

```
library(ggplot2)
library(ggridges)
ggplot(
  UserComments,
  aes(x = Rating, y = as.factor(classification))) +
  theme_minimal() +
  geom_density_ridges2(alpha = 0.3)
```

```
## Picking joint bandwidth of 5.82
```