Agrupación y Clasificación de Usuarios Según Tweets - Recolección y Preparación de Datos

Alonso, Francisco - Astor, Miguel - Hernandez, Jesús - Machado, Javier - Sisco, Yilber

4 de febrero de 2016

Propuesta:

Tomando una muestra de tweets, agrupar a cada usuario dentro de esa muestra según las caracteristicas de los tweets que publica, para luego poder realizar una clasificación de nuevos usuarios basándose en las clases obtenidas.

Agrupar y clasificar en clases a cada usuario ppdría proveer información útil a otros usuarios sobre el contenido que publica un grupo de usuarios en específico, por ejemplo el conjunto de usuarios que sigue su cuenta de Tweeter. Puede proveer también información sobre los intereses de un usuario específico, ésta información puede ser usada con fines de mercadeo, segmentación de publicidad o estudio de las tendencias en una región.

Objetivos de negocio

Obtener información sobre los usuarios que permita mejorar la toma de decisiones en distintas áreas donde sea comun el uso de Tweeter.

Factores que determinan el éxito del proceso

Precisión y confiabilidad de la clasificación de los usuarios

La calidad de la información obtenida y los objetivos planteados dependen totalmente de estos factores. No es posible diferenciar usuarios y tomar decisiones respecto a ellos si los resultados no son suficientemente precisos y confiables.

Fuente de datos

La data necesaria es el conjunto de tweets de cada usuario en la muestra. La recolección de datos será realizada por medio del API que provee Tweeter, desde R se obtendrá una muestra de usuarios y un subconjunto de los tweets de su "TimeLine".

Objetivos del proceso de minería de datos

- Agrupar a los usuarios y obtener un conjunto de clases que describan las diferencias entre cada usuario. Tarea de MD: Agrupación o Clustering.
- Clasificar a un conjunto de usuarios en base a las clases obtenidas en la fase de agrupación. Tarea de MD: Clasificación.

Recolección de datos

Para realizar el proceso de recolección es necesario crear una cuenta en Twitter y crear una aplicación mediante la cuál se va a tener acceso al API de Tweeter.

El primer paso realizado fue la recolección aleatoria de un conjunto de tweets en un intervalo de tiempo específico y una región definida por latitud, longitud y un radio.

```
tweets <- searchTwitter(""  # Texto requerido en el tweet
                        , n=10 # Número de tweets solicitados
                        , lang="es" # Idioma Español
                        , since="2015-10-01" # Fecha inicial
                        , geocode='10.480594,-66.903606,7mi'#Ubicación
                        )
```

De esta primera muestra se toman los usuarios:

```
allUsers <- tweets$screenName
allUsers <- unique(allUsers)
```

Luego se obtienen 50 tweets por usuario, a cada tweet se le da un formato el cual consiste en remover los caracteres expeciales, los números y se transforman todos los caracteres a minúsculas.

```
userTweets = userTimeline(allUsers[i], 50, includeRts = F)
userTweets = sapply(userTweets, function(x) x$getText())
userTweets = cleanme(userTweets)

allClean = c(allClean, userTweets)
```

Luego de tener preparados los tweets de cada usuario se procede a remover las palabras "vacías", éstas no tienen un valor útil dentro del texto ya que son muy comunes y/o no proveen información sobre algún tópico en especial.

```
allClean = removeWords(allClean, c(stopwords("spanish"))) # remueve palabras vacías
sw <- readLines("stopwords.es.txt", encoding="UTF-8")
sw = iconv(sw, to="ASCII//TRANSLIT")
allClean = removeWords(allClean,sw)
```

Por último se genera un objeto de tipo Corpus que contiene un conjunto de documentos (conjunto de tweets). Este objeto se formatea de nuevo y se genera a partir de él, una matriz Término Documento aplicando el cálculo de pesos TF-IDF, esta muestra cada término y la cantidad de veces que se encontró ese término entre los tweets de un usuario específico.

```
corpus = Corpus(VectorSource(allClean))
corpus = tm_map(corpus, removeWords, sw) # remueve palabras vacías
corpus = tm_map(corpus, stripWhitespace) # remove espacios en blanco extras
corpus <- tm_map(corpus, content_transformer(tolower))
# matriz término documento
tdm = TermDocumentMatrix(corpus
                         ,
                         control = list(weighting =
                                        function(x)
```

```
                                              weightTfIdf(x, normalize =
                                                          FALSE),
                                          stopwords = TRUE))
tdmMatrix <- as.matrix(tdm)
write.csv(tdmMatrix, 'TDM.csv') # exporta el data frame a csv
```

A partir de esta matriz se requiere hacer una limpieza más exhaustiva de términos, la data está muy esparcida con una gran cantidad de valores en cero. Luego se puede proceder con las siguientes iteraciones del proceso de minería de datos.

Inconvenenientes encontrados

Durante el proceso de limpieza se encontraron tweets con una gran cantidad de íconos y "emojis", estos presentan un gran obstáculo al momento de limpiar el conjunto de datos ya que producen muchos errores y no hay herramientas específicas para lidiar con el formato, codificación y caracteres especiales que arrojan al conjunto de datos.

Las herramientas para lematización requieren más investigación de nuestra parte ya que las encontradas no proveen los resultados mínimos necesarios para usarlas en este proceso. Se probaron la biblioteca "SnowballC" y una aplicación llamada "Grampal".