SVM_texto

Alejandro Campoy Nieves

29 de enero de 2019

```
library(RTextTools)
library(e1071)
```

SVM

En esta sección vamos a hablar sobre la técnica de Support Vector Machine (SVM). Principalmente presenta el inconveniente de que es poco escalable y costosa (computacionalmente y en tiempo), por lo que no vamos a hacer un análisis muy detallado del mismo.

```
# Cargamos los tados
datos_train <- read.table("datos/datos_train_preprocesado.csv", sep=",",
                          comment.char="",quote = "\"", header=TRUE)
datos_test <- read.table("datos/datos_test_preprocesado.csv", sep=",",
                         comment.char="",quote = "\"", header=TRUE)
# Establecemos la semilla
set.seed(3)
```

Inicialmente, tratamos de utilizarlo con valores numéricos de nuestro dataset. El problema es que solo tenemos valores discretos y categóricos, por lo que esta técnica no funcionaba de forma correcta a la hora de calcular distancias. Tratar de hacer continuo un valor discreto es una pérdida de tiempo, ya que no podemos basarnos en nada para decidir el valor exacto que alcanzaría un valor discreto en un espacio continuo.

Tras darnos cuenta de esto, nos pasamos directamente al tratamiento del texto. Los pasos que llevamos a cabo fueron los siguientes:

```
# Creamos la matriz de términos
dtm_train <- create_matrix(datos_train$benefits_preprocesado)

# Creamos un contenedor a partir de la matriz de términos
contenedor <- create_container(dtm_train,datos_train$ratingLabel,
                               trainSize=1:length(dtm_train),virgin=FALSE)

# Entrenamos el modelo SVM con los parámetros que queramos
modelo_svm <- train_model(contenedor,"SVM",kernel="radial")

# Creamos la matriz de términos para el conjunto de test
dtm_test <- create_matrix(as.list(datos_test$benefits_preprocesado),
                          originalMatrix = dtm_train)

#Obtenemos el contenedor correspondiente a la matriz de términos del conjunto de Test
contenedor_test <- create_container(dtm_test,labels=as.factor(datos_test$ratingLabel),
                         testSize=1:length(datos_test$ratingLabel),virgin=FALSE)

resultados_svm <- classify_model(contenedor_test,modelo_svm)
resultados_svm
```

Destacar de este proceso que hacemos uso de la matriz de términos como viene siendo común a la hora de

manipular texto. Utilizamos el kernel de tipo radial y la totalidad del conjunto de entrenamiento para llevar a cabo la construcción del modelo.

Pero nos daba un error a la hora de ejecutar este código. Esto retrasó mucho el avance de la práctica ya que no conseguíamos darnos cuenta de que es lo que ocurría. Finalmente, parece ser una incompatibilidad de la librería en create_matrix. Por lo que tuvimos que cambiarla con la ayuda de *trace*. En el código hay un comentario que especifíca el cambio a realizar en caso de tener este problema.

Posteriormente, tratamos de deducir el *ratingLabel* a partir de los cometarios sobre los beneficios de los medicamentos. Obteniendo finalemte los siguientes resultados:

	SVM_LABEL <fctr>	SVM_PROB <dbl>
1	0	0.7585066
2	0	0.7585066
3	0	0.7585066
4	0	0.7585300
5	0	0.7585066
6	0	0.7585066
7	0	0.7585066
8	0	0.7585066
9	0	0.7585066
10	0	0.7585066
11	0	0.7585066
12	0	0.7585066
13	0	0.7585066
14	0	0.7585066
15	0	0.7585066
16	0	0.7585300
17	0	0.7585066
18	0	0.7585066
19	0	0.7585066
20	0	0.7585066
21	0	0.7585066
22	0	0.7585066
23	0	0.7585066
24	0	0.7585066
25	0	0.7585066
26	0	0.7585066
27	0	0.7585066
28	0	0.7585066
29	0	0.7585066
30	0	0.7585066
31	0	0.7585066
32	0	0.7585066
33	0	0.7585066
34	0	0.7585066
35	0	0.7585066
36	0	0.7585066
37	0	0.7585066
38	0	0.7585066
39	0	0.7585300
40	0	0.7585066
41	0	0.7585066
42	0	0.7585066
43	0	0.7585300
44	0	0.7585066
45	0	0.7585066
46	0	0.7585066
47	0	0.7585066
48	0	0.7585300

1-48 of 1,034 rows

Figure 1: Salida de SVM para el ratingLabel a partir de los comentarios de texto sobre los beneficios.

Como podemos ver, el clasificador no funciona bien. Etiqueta todo a 0. Creemos que el error viene dado debido a la disposición de los términos (igual que vimos en el clustering). Al tener una nube de puntos tan densa y tan poco disjunta, esto hace que no haya una función aceptable para poder separarlas.

Por otro lado, SVM no funciona bien con texto a no ser que tengas las mismas palabras tanto en el test como en el train, cosa que no ocurre en nuestro caso. Esto hace que SVM no tenga todo lo necesario para generar una buena función. Cosa que sí ocurre, por ejemplo, en este tutorial https://www.svm-tutorial.com/2014/11/svm-classify-text-r/.

En definitiva, decidimos no perder más tiempo con esta técnica, ya que parece no ser adecuada para nuestro

dataset y seguir avanzando en el resto que proponemos con esta práctica.