Proyecto Final Álgebra

Geraldine González Fernández

2022-12-02

Proyecto Final

Latent Semantic Analysis: Una Aplicación de Temas de Álgebra

Geraldine González Fernández

En este trabajo se presentará una aplicación de temas que vimos en clase como Descomposición en Valores Singulares (DVS) o su nombre en inglés Singular Value Decomposition (SVD), el cual tiene como base la obtención de eigenvectores y eigenvalores.

La aplicación que desarrollaré en este trabajo será un ejemplo de Latent Semantic Analysis (LSA), el cual tiene su base en los temas de álgebra listados en el párrafo anterior. Para la implementación, me apoyaré de la paquetería de R llamada: **LSAfun**.

¿Qué es Latent Semantic Analysis?

LSA es una técnica que se utiliza para calcular la similitud semántica de términos en una colección de documentos. Este algoritmo tiene su base en la descomposición de valores singulares de la matriz de términos por documentos. Esta técnica surge buscando solución al problema de tener que analizar grandes volúmenes de textos como: artículos, emails, libros, entre otros.

Lo que busca LSA es la generación de temas o topics, un topic en el área Natural Language Processing (NLP) es una colección de palabras que están altamente relacionadas. Un documento suele estar asociado a más de un topic, y gracias a LSA podemos extraer los principales temas de los que se habla en un conjunto de textos y convertir este conjunto de textos en matrices de palabras por temas y temas por documentos.

¿Qué pasos se realizan en un análisis de LSA?

1. Construir una matriz de términos por documentos.
2. Aplicar un proceso para ponderamiento de palabras como la log-entropía
3. Aplicar descomposición de valores singulares
4. Definir cantidad de temas a utilizar

¿Cómo es la Matriz Términos por Documentos?

El primer paso para analizar un texto es construir la matriz de frecuencias de términos por documentos, se entenderá por documento a cada fragmento de texto a analizar. En esta matriz los renglones representan los términos y las columnas representan los documentos, visualizándose de la siguiente forma:

.	Documento-1	Documento-2	...	Documento-M
Término-1				
Término-2				
⋮				
Término-N				

¿Qué es la Descomposición de Valores Singulares?

La descomposición en valores singulares es un método de factorización que aplica para matrices no cuadradas. En esta se busca obtener nuestra matriz de interés como el producto de matrices cuyos valores nos aporten información.

En el caso de LSA usaremos esta descomposición para obtener las siguientes matrices:

$$A\ nxm\ = U\ nxr\ * S\ rxr\ * V^T\ mxr$$

Donde:

(A) Matriz de términos por documentos- es la base para el análisis.

(U) Matriz de palabras por temas- sus valores muestran la asociación entre cada palabra y los temas formados.

(S) Matriz diagonal de temas- Evalúa la fuerza de cada tema en la colección de documentos.

(V) Matriz de documentos por temas- Contiene la fuerza de asociación entre cada documentos y sus temas asociados.

¿Qué datos vamos a utilizar?

El paquete que vamos a utilizar **(LSAfun)** no esta diseñado para la creación de la matriz de términos por documentos, por lo que el alcance de este proyecto parte de que ya tenemos esta matriz.

Una forma de trabajar partiendo de que ya tenemos un espacio semántico, es usar alguno que ya esté disponible en la web en alguna página como la siguiente:

https://sites.google.com/site/fritzgntr/software-resources/semantic_spaces?pli=1

En este trabajo usaremos un espacio preconstruido que utiliza una variedad de textos, novelas, artículos de periódicos y otra información de la TASA (*Touchstone Applied*

Science Associates, Inc.) conjunto de textos que fueron utilizados para desarrollar *The Educator's Word Frequency Guide*.

El link de descarga de los datos es el siguiente:

https://drive.google.com/file/d/1PjSy9qyy7Sh3T9higCPqtgnG0_ffiuBC/view

Ejecución del Código

```
load("D:/CIMAT/Algebra/ProyFinalAlgebra/TASA.rda")
```

```
# Carga de librerías a utilizar
# Si no contamos con la paqueteria instalada descomentar la siguiente
línea de código.
#install.packages("LSAfun")

#NOTA: Verificar que ya esten cargados los datos de TASA. Leer el archivo
de README.txt
library(LSAfun)
```

```
## Warning: package 'LSAfun' was built under R version 4.2.2
```

```
## Loading required package: lsa
```

```
## Warning: package 'lsa' was built under R version 4.2.2
```

```
## Loading required package: SnowballC
```

```
## Loading required package: rgl
```

```
## Warning: package 'rgl' was built under R version 4.2.2
```

```
# Algunas funciones basicas de la librería LSAfun

# Computa la similaridad entre dos palabras
Cosine("lion", "tiger", tvectors=TASA)
```

```
## [1] 0.6285499
```

```
Cosine("lion", "first", tvectors=TASA)
```

```
## [1] -0.01437755
```

```
# Computa las similaridades entre todas las parejas de
# 2 palabras
multicos("tiger lion cow",
         "cat zoo",
         tvectors = TASA)
```

```
##             cat        zoo
## tiger 0.3696037 0.33791316
## lion  0.4732016 0.48697617
## cow   0.1460171 0.02467506
```

Como se puede ver en el primer ejemplo, la función Cosine calcula la similaridad entre un par de palabras. Se observa un alto peso entre las palabras lion y tiger, y un bajo peso entre las palabras lion y first. Por otro lado, con la función multicos se calcula la función cosine para todos los pares de palabras especificados los cuales se muestran en formato de matriz.

```
# Computa la similaridad entre dos documentos que consiten
# multiples palabras
costring("The lions go on a hunt.",
         "The antelopes start to panic.",
         tvectors = TASA)

## Note: not all elements in x were found in rownames(tvectors)
##
## Note: not all elements in y were found in rownames(tvectors)

## [1] 0.1679408

# Computa la similitud entre un documento y
# una lista de palabras simples
multicostring("The lions go on a hunt",
              "elephant antelope jump airplane",
              tvectors = TASA)

## Note: not all elements in x were found in rownames(tvectors)

##                   elephant  antelope      jump   airplane
## expression in x 0.08813303 0.2270994 0.1876783 0.01364226
```

Se puede observar que la frase de que “los leones van a cazar” tiene una mayor asociación con la palabra “antílope” tiene sentido ya que los leones cazan antílopes. Lo cual también se ve reflejado en que los dos documentos tengan una asociación positiva.

También con la descomposición de valores singulares podemos obtener cuales son las palabras más cercanas, a una palabra que nos interese. Esto esta implementado en la función **neighbors**.

Por otro lado, con la función **choose.target** podemos elegir aleatoriamente palabras que estén asociadas dentro de un rango de un rango de valores aceptables con respecto a nuestra palabra de interés.

```
# Obtenemos las palabras más cercas a la
# palabra especificada
# Vecinos más cercanos
print("Vecinos mas cercanos")

## [1] "Vecinos mas cercanos"

neighbors("lions", n=10, tvectors = TASA)
```

```
##      lions       lion  elephants     beasts   leopards rhinoceros
zoo
##  1.0000000  0.7620240  0.6662553  0.6370110  0.6322121  0.5940019
0.5939769
##   antelope    animals  antelopes
##  0.5936863  0.5916549  0.5783232

print("Muestreo Aleatorio")

## [1] "Muestreo Aleatorio"

# Muestramos aleatoriamente palabras asociadas
choose.target("lions", lower = .2, upper= .3, n = 10, tvectors = TASA)

## encompassed    whuffing meadowlands    peacable   explicity
497
##   0.2281448   0.2652936   0.2827285   0.2880597   0.2126202
0.2405646
##      kiowas    cackling      basuto    practive
##   0.2214343   0.2353351   0.2458892   0.2084093

choose.target("lions", lower = .2, upper= .3, n = 10, tvectors = TASA)

##        iguana      flamingo     powerline      scurries packinghouses
##     0.2420828     0.2129007     0.2324215     0.2662283     0.2077705
##    sandstorms           oba     amblypods          zona       indicus
##     0.2752004     0.2184154     0.2649485     0.2795240     0.2527801
```

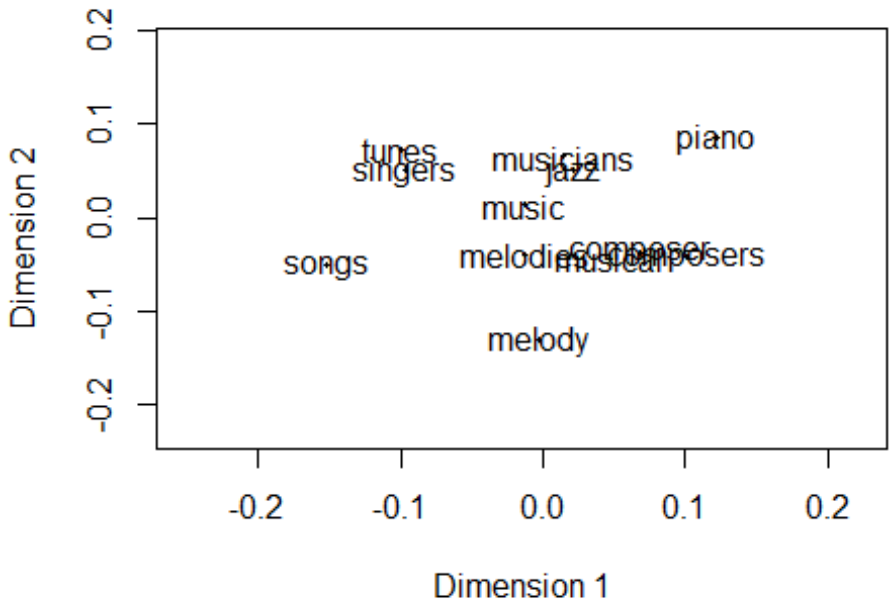
Se puede ver que los 10 vecinos más cercanos de la palabra lions son las pabras: lions, lion, elephants, beast, leopards, rhinoceros, zoo, antelope, animals, antelopes. Por otro lado, vemos que cuando ejecutamos dos veces la función **choose.target** obtenemos resultados diferentes pero en ambos se obtienen palabras con pesos entre .2 y .3 como se específico en la función.

Esta librería también nos ayuda a visualizar gráficamente la asociación entre algunos de los términos, a continuación, mostramos algunos ejemplos de gráficos que se pueden construir con esta paquetería.

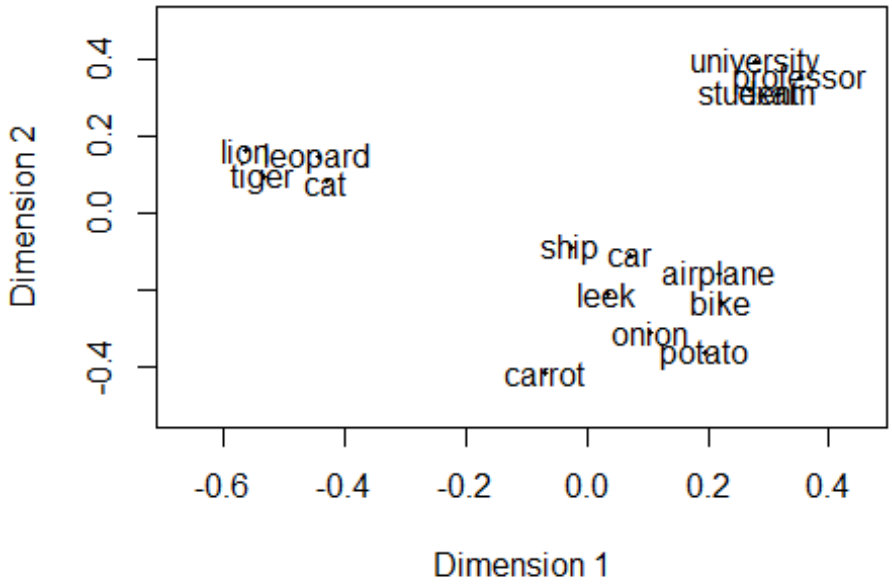
En ambos ejemplos, se visualizan de manera gráfica la construcción de los vecinos más cercanos a un termino. Estos se pueden visualizar en dos o tres dimensiones.

```
# Graficos
plot_neighbors("lions", n = 10, tvectors = TASA, method = "PCA",
               dims = 3, connect.lines = "all", alpha = "shade" )

##                    x           y          z
## lions      0.7067387 -0.45106163 -0.3805794
## lion       0.8788998 -0.04703267 -0.2402160
## elephants  0.3497705 -0.51825883 -0.5601681
## beasts     0.7329013 -0.30283273 -0.1120782
## leopards   0.5108848 -0.44588763 -0.2919910
## rhinoceros 0.6751470 -0.14291936 -0.3507755
```

```
## zoo         0.2741928 -0.16516400 -0.8612802
## antelope    0.2169622 -0.84470829 -0.2262244
## animals     0.2548814 -0.22334130 -0.8049642
## antelopes   0.1516170 -0.88498232 -0.1458625

plot_neighbors("music", n=12, tvectors = TASA, method = "MDS", dims = 2)
```




```
##                       x           y
## music      -0.012237989  0.01457545
## jazz        0.021750816  0.05041432
## melodies   -0.013240165 -0.03834276
## singers    -0.096100792  0.05077122
## musicians   0.014514560  0.06614196
## musical     0.046435000 -0.04352012
## composer    0.068114594 -0.03659474
## songs      -0.152139387 -0.05050243
## tunes      -0.098892631  0.07306589
## composers   0.101495949 -0.04222877
## piano       0.122924629  0.08538641
## melody     -0.002624585 -0.12916642

words <- c("lion", "tiger", "leopard", "cat",
           "potato", "carrot","leek", "onion",
           "student", "university", "professor","exam",
           "car", "ship", "airplane","bike")

plot_wordlist(words, method = "MDS", dims = 2, tvectors = TASA)
```




```
##                      x           y
## lion       -0.56459385  0.16472148
## tiger      -0.53214187  0.09552055
## leopard    -0.44186988  0.14797881
## cat        -0.43058904  0.08451554
## potato      0.19564672 -0.35879105
## carrot     -0.07044640 -0.41226407
## leek        0.03617115 -0.20827165
## onion       0.10464281 -0.30807374
## student     0.26714162  0.32100902
## university  0.27969000  0.39486595
## professor   0.35128642  0.35188230
## exam        0.31660295  0.30893342
## car         0.07236843 -0.11056474
## ship       -0.02542954 -0.08557994
## airplane    0.21977907 -0.15820773
## bike        0.22174142 -0.22767414
```

Una aplicación importante que se le puede dar a la generación de temas, es la de sumarizar un texto en sus k más importantes oraciones. La librería **LSAfun** cuenta con la función **genericSummary()** que aplica este método. Al final las mejores oraciones que resumen un texto, pueden estar asociadas al los temas que genera el LSA.

A continuación, mostraremos un ejemplo del uso de esta función, para el ejemplo usaremos el texto de descripción de los leones que aparece en la página del

Smithsonian's National Zoo & Conservation Biology Institute (https://nationalzoo.si.edu/animals/lion)

```
# Texto a sumarizar
# Generic Summary
D <- "Lions are big cats that mainly live in Africa.
However, there are also small populations in Asia.
Male lions have a mane, while female ones do not.
Lions usually live in small packs called prides.
There is one dominant male lion in a pride,
with several related lionesses around him.
The lionesses do most of the hunting for the pride

Physical Description
Lions have strong, compact bodies and powerful forelegs, teeth
and jaws for pulling down and killing prey. Their coats are yellow-gold,
and adult males have shaggy manes that range in color from blond to
reddish-brown to black. The length and color of a lion's mane is likely
determined by
age, genetics and hormones. Young lions have light spotting on their
coats that disappears as they grow.

Without their coats, lion and tiger bodies are so similar that only
experts can tell them apart.
Size

Lions stand between 3.5 and 4 feet (1 and 1.2 meters) tall at the
shoulder. Males grow to lengths of 10 feet (3 meters) and have a 2 to 3
foot (60 to 91 centimeter) tail. They weigh from 330 to 550 pounds (150
to 250 kilograms). Slightly smaller, females grow to lengths of 9 feet
(2.7 meters) and weigh between 265 and 395 pounds.
Native Habitat

Lions inhabit a wide range of habitats, from open plains to thick brush
and dry thorn forest. Except for a small population of the Indian lion
subspecies that remains in the Gir Forest of northwest India, lions now
live only in Africa, from the Sahara's southern fringe to northern South
Africa.  They are absent from equatorial areas dominated by moist
tropical forest.
Food/Eating Habits

Lions primarily eat large animals that weigh from 100 to 1,000 pounds (45
to 453 kilograms), such as zebra and wildebeest. In times of shortage,
they also catch and eat a variety of smaller animals, from rodents to
reptiles. Lions also steal kills from hyenas, leopards and other
predators. At times, they may lose their own catches to hyena groups.
Lions may also feed on domestic livestock, especially in areas near
villages.

The Smithsonian's National Zoo's lions eat ground beef, which is
```

```
commercially produced to meet the nutritional needs of carnivores. Twice a week, they receive knucklebones or beef femurs, and once a week they receive rabbits, which exercise the cats' teeth and jaws.
Social Structure

Lions are the world's most social felines. They live in groups of related females, called prides, which may comprise several to as many as 40 individuals, including adults, sub-adults (between the ages of 2 and 4) and cubs, plus one or more resident males. Abundance of prey availability plays a significant role in the size of a lion pride. Pride mates associate in sub-groups within the pride.

Females usually stay in their mothers' prides for life, unless food scarcity forces them out. Young males are driven from their prides when they grow large enough to compete with the dominant males (usually between the ages of 2 and 4). They create coalitions, usually with brothers and cousins, and search for a pride to take over. Males entering a new pride will kill all cubs that cannot run from them. Adult males that are fortunate enough to achieve residency within a pride hold tenure for an average of two years, often leaving due to eviction by another coalition of males. In India, female and male lions live apart, joining only to mate.

Males take on most of the defensive duties. However, both males and females mark their territories by roaring – which can be heard up to five miles away – and scent marking with urine. Females raise the cubs and are the primary hunters, although males will sometimes join females during a hunt. Depending on the prey item, several lions may stalk prey from different angles to within 100 feet (30 meters) before attacking the targeted animal. Nomadic males must hunt alone or scavenge from other animals.
Reproduction and Development

Females are receptive to mates for a few days several times a year, unless they are pregnant or nursing, and mating spurs ovulation. They typically give birth to a litter every two years.

Females usually give birth to one to four cubs after a gestation of about 3 1/2 months. Cubs typically nurse for six months but start eating meat at three months. Due to dangers, including starvation during times of food shortage and attacks by male lions taking over prides, up to 80 percent of lion cubs die within their first 2 years of life.
Lifespan

Lions in zoos may live into their late teens or early 20s. In the wild, a lioness may live up to 16 years, but males rarely live past the age of 12.
"

print("Summary 1 oración")
```

```
## [1] "Summary 1 oración"
```

```
genericSummary(D, k=1)
```

```
## [1] " They live in groups of related females, called prides, which may
comprise several to as many as 40 individuals, including adults, sub-
adults (between the ages of 2 and 4) and cubs, plus one or more resident
males"
```

```
print("Summary 2 oración")
```

```
## [1] "Summary 2 oración"
```

```
genericSummary(D, k=2)
```

```
## [1] " They live in groups of related females, called prides, which may
comprise several to as many as 40 individuals, including adults, sub-
adults (between the ages of 2 and 4) and cubs, plus one or more resident
males"
## [2] " Except for a small population of the Indian lion subspecies that
remains in the Gir Forest of northwest India, lions now live only in
Africa, from the Sahara's southern fringe to northern South Africa"
```

```
print("Summary 3 oración")
```

```
## [1] "Summary 3 oración"
```

```
genericSummary(D, k=3)
```

```
## [1] " They live in groups of related females, called prides, which may
comprise several to as many as 40 individuals, including adults, sub-
adults (between the ages of 2 and 4) and cubs, plus one or more resident
males"
## [2] " Except for a small population of the Indian lion subspecies that
remains in the Gir Forest of northwest India, lions now live only in
Africa, from the Sahara's southern fringe to northern South Africa"
## [3] " Twice a week, they receive knucklebones or beef femurs, and once
a week they receive rabbits, which exercise the cats' teeth and jaws"
```

Como se puede ver ya no tenemos que leer el texto de los leones, y gracias a la LSA podemos facilmente tener un resumen de la información contenida en el texto.

Conclusiones y Siguientes Pasos

LSA es una técnica básica que ayuda a resumir información, sin embargo, esta también tiene varias limitaciones como por ejemplo tiene dificultad al trabajar con homónimos o palabras polisémicas. También los resultados cambian conforme el numero de temas que se este trabajando.

Por otro lado, esta técnica nos ayuda a resumir fácilmente la información ya que con conceptos fáciles podemos extraer un resumen de grandes volúmenes de texto. A su vez, esta técnica tiene algunos puntos que se pueden refinar como por ejemplo la se

lección de los temas a utilizar, el indagar otras paqueterías que tengan implementado este algoritmo o incluso el implementar de forma artesanal esta técnica.

Finalmente, cabe destacar que esta técnica es una de las herramientas básicas dentro del área de NLP, por lo que es un buen primer enfoque para analizar información de textos. Sin embargo, gracias al crecimiento de la tecnología ahora contamos con técnicas con mayor poder que nos pueden brindar mejores resultados la aplicación de temas de inteligencia artificial podría otorgar mejores resultados en la extracción de resúmenes de grandes volúmenes de textos.

Bibliografía

https://www.datacamp.com/tutorial/discovering-hidden-topics-python

https://towardsdatascience.com/topic-modeling-with-latent-semantic-analysis-58aeab6ab2f2

https://link.springer.com/article/10.3758/s13428-014-0529-0

https://nlp.stanford.edu/IR-book/html/htmledition/latent-semantic-indexing-1.html