Métodos matemáticos para el análisis semántico de redes sociales

Trabajo de Fin de Grado

Grado en Matemáticas

Autor:

José Pertierra das Neves

Tutores:

Alejandro J. García del Amo Jiménez

Miguel Romance del Río

Septiembre de 2016

“Un día, una persona está ahí, va al trabajo, está viva.
Y de pronto, nada. La gente desaparece.
Se abre la tierra y se la traga.
Es inquietante y misterioso.”

- Bret Easton Ellis (**American Psycho**)

Agradecimientos

A mis padres, en donde he encontrado la seguridad que muchas veces he necesitado. A mi hermano, aunque sea más de letras. A mi familia, la de aquí y la de allí. A mis amigos, otra familia más. Mis compañeros de universidad, los del Grado de Ingeniería Informática, los del Grado en Matemáticas y los del Doble Grado en concreto, una generación de gente increíble. A los maestros de colegio e instituto que han impulsado mis ganas de descubrir, a Fernando. A la Universidad Rey Juan Carlos y al departamento de Matemáticas que me han permitido avanzar muy rápido en poco tiempo ofreciendo este Doble Grado. A sus profesores, todos y cada uno han dejado huella en mí. En especial, a mis tutores de Trabajo de Fin de Grado, Miguel y Alejandro, quienes me han aportado este proyecto, casi tan interesante como ellos. A las increíbles personas que he conocido este último año, me han ayudado a encontrarme y definirme. A Lu, por todo.

A las demás personas que he conocido a lo largo de mi vida, de todas he aprendido algo y todas han influido. A mi yo del pasado y a mi yo del futuro.

Gracias.

Resumen

En el Trabajo de Final de Grado de título *Métodos matemáticos para el análisis semántico de redes sociales* se realiza una labor de revisión de los fundamentos matemáticos de métodos y herramientas con un uso potencial sobre datos reales procedentes de la red social Twitter. Las dos áreas principales que se tratan en el documento son: métodos de clasificación de textos y aplicación de teoría de grafos y redes.

La primera, se centra en una descripción resumida del funcionamiento de clasificadores mediante el aprendizaje Bayesiano y del Análisis Semántico Latente. Se especifican además casos de ejemplos concretos y a muy bajo nivel de la metodología numérica realizada mediante un motor de cálculo como Octave.

En la segunda, se realiza una descripción resumida sobre la definición de grafos y redes y sus características. El proyecto se ha enfocado en la aplicación sobre Twitter, por lo que se han unificado las características más relevantes de las redes que se deban tener en cuenta si las creamos a partir de datos de una interacción entre usuarios. Encontraremos también en este apartado un caso de estudio con datos de la red social.

Palabras clave: Análisis Semántico Latente, Aprendizaje Bayesiano, Clasificador Naive-Bayes, Descomposición en Valores Singulares, Grafos, Machine Learning, Redes Complejas, Redes Sociales, Twitter.

Abstract

The Final Degree Project named *Métodos matemáticos para el análisis semántico de redes sociales* reviews the mathematical fundaments which are related to methods and tools that can be used over real data from Twitter. The Project is composed by two main areas: textual clasification methods and application of graphs and networks theory.

The first one is centered in a brief description of the clasificators behaviour with the Bayesian learning and the Latent Semantic Analysis. Moreover, some example cases are specified in a quite low level of the numerical methodology with a calculus engine such as Octave.

In the second one, the graphs and networks definition and their characteristic are briefly described. The project has been focused in the Twitter social network, so the most relevant characteristic of the networks created from users interaction data have been unified. In this part of the project, a case of study with social network data can also be found.

Keywords: Bayes classifier, Bayesian Learning, Complex Networks, Graphs, Latent Semantic Analysis, Machine Learning, Singular Value Decomposition, Social Networks, Twitter.

Índice

Figuras

Tablas

1. Introducción

1.1. Motivación

El proyecto nace como un enfoque unificador entre dos áreas relativamente jóvenes de las matemáticas aplicadas: Machine Learning y Análisis de Redes Complejas.

En numerosos estudios e investigaciones se han realizado implementaciones sobre la clasificación de categorías con textos [1] e incluso se ha trabajado sobre la inferencia del sentimiento de un nuevo texto realizado por personas en una web, por ejemplo, de opinión sobre un producto. [2]

Por otro lado, vemos que la evolución de la teoría de Redes es relativamente nueva y se encuentra en constante crecimiento. Ésto viene dado en gran parte por la evolución de los sectores tecnológicos y de la información, lo que ha incentivado que empresas quieran explotar las grandes cantidades de datos que se generan. Multitud de estudios se han centrado en dos principales campos de aplicación de esta materia: economía y sociología. [3]

El aspecto social de las redes tiene una relación potencialmente aprovechable desde el punto de vista del análisis del sentimiento. Dados los fundamentos que definen características a estudiar en redes conformadas desde datos reales, podemos enriquecerlas si conseguimos encontrar un foco donde utilizar la diferenciación de sentimientos. En este aspecto entra el juego la transmisión de información entre redes sociales que se realiza mediante mensajes de texto, como puede ser Twitter o Facebook.

1.2. Finalidad del proyecto

La finalidad principal del proyecto es la de realizar una revisión sobre los fundamentos matemáticos de herramientas de Machine Learning con aplicación en la aplicación de clasificación de textos primeramente. Esto es, principalmente el LSA y los clasificadores Bayesianos. Éstos han sido escogidos por su importancia histórica en el desarrollo del análisis semántico y la representación del sentimiento, principalmente el LSA [4], mediante la concepción de representación vectorial, y en el caso de los clasificadores Bayesianos por su extenso uso en investigaciones relacionadas con la categorización de textos [5] [6].

Después, realizar una revisión sobre los fundamentos matemáticos de la Teoría de Grafos y de Redes que nos permitan su aplicación sobre aspectos de la red social Twitter como son la participación en Temas y las conversaciones creadas.

1.3. Organización del documento

El documento que compone la memoria sobre la realización del proyecto se divide en tres partes principales. En la primera abarcaremos todos los fundamentos teóricos. Éstos son clasificación de textos y sentimiento, que se divide a su vez en el estudio de Aprendizaje

Bayesiano y en el Análisis Semántico Latente, como se ha explicado anteriormente se han escogido éstos dos apartados por ser los más destacados en trabajos sobre clasificación de textos [7]. Además, se describe un ejemplo realizado con ambos clasificadores al final de esta sección. La segunda se corresponde con el estudio de grafos y redes y cómo se aplican a las redes sociales, pasando desde su representación y estructura hasta las principales características a cuantificar. Por último, elaboraremos un apartado donde se pondrán en práctica algunas de las técnicas tratadas sobre datos obtenidos de Twitter de donde obtener algunas conclusiones y definir posibles líneas de trabajo futuro.

2. Análisis de Sentimiento y Clasificación de Textos

Dentro de las técnicas desarrolladas por las ciencias de la computación encontramos un ámbito centrado en la extracción de información y clasificación de recursos. El aprendizaje automático o Machine Learning se ha ido desarrollando dentro de la Inteligencia Artificial como una ciencia en sí misma que ha intentado abordar los ejercicios de automatización casi desde que Alan Turing planteara su famoso test a mitad del siglo XX. En este problema también interviene el Procesamiento del Lenguaje Natural, nacido a mitad del siglo XX, cuyo campo de estudio es el relacionado con la investigación de mecanismos de comunicación entre personas y máquinas por medio de lenguajes naturales [5].

Hoy en día, podemos encontrar una gran cantidad de información accesible online, ésta se organiza comúnmente en documentos. Una gran parte del esfuerzo de investigadores y desarrolladores ha sido también la automatización de la categorización de los textos. Muchos trabajos se han centrado en la clasificación de las temáticas, intentando ordenar los documentos según los tópicos que tratan [8]. Sin embargo, hace relativamente poco tiempo hemos visto un crecimiento de los foros de discusión, de opinión y páginas web donde se valoran los productos o servicios que se venden. Casi todos ofrecen la oportunidad de que el usuario valore positiva o negativamente la experiencia descrita. La principal característica de este tipo de documentos generados, en contraste con los que se habían trabajado anteriormente, es el de encontrar el sentimiento y no tanto su temática.

La clasificación de sentimiento tiene entonces una aplicación directa en cualquier ámbito donde estudiar un texto que, pensemos, deba contener algún tipo de sentimiento. El Business Intelligence o los sistemas de recomendación son ejemplos fáciles de encontrar donde utilizar éstos métodos desarrollados, además de otros muy comunes y conocidos como los filtros y detecciones de spam. [2]

En estos apartados revisaremos las técnicas y procedimientos más comunes en la clasificación de textos, sus mejoras y optimizaciones, sus pros y sus contras en determinadas pruebas. Los principales que abordaremos serán los clasificadores mediante aprendizaje Bayesiano y los basados en el Análisis Semántico Latente.

2.1. Aprendizaje Bayesiano

El razonamiento bayesiano [7] nos proporciona un enfoque probabilístico de la inferencia partiendo desde la asunción de que las incógnitas de estudio siguen distribuciones probabilísticas. El Teorema de Bayes, planteado por Thomas Bayes en 1763, sienta las bases con las que podremos conseguir una solución óptima por medio de estas distribuciones y datos observados. También podremos controlar la ponderación de la posibilidad de ocurrencia de una hipótesis cuantitativamente.

Ilustración I Thomas Bayes

De esta forma los algoritmos de aprendizaje basados en el aprendizaje bayesiano pueden calcular probabilidades explícitas para cada hipótesis. El aprendizaje lo podremos ver desde una perspectiva en la que el objetivo es encontrar la hipótesis más probable, dado un conjunto de entrenamiento T y un conocimiento a priori sobre la probabilidad de cada hipótesis.

Algunas de las principales características del aprendizaje bayesiano:

- Cada ejemplo de entrenamiento afecta a la probabilidad de las hipótesis. (Mayor efectividad que el descarte directamente de las hipótesis incompatibles)
- Podremos incluir conocimiento a priori: probabilidad de cada hipótesis; y la distribución de probabilidades de los ejemplos.
- Tenemos cierta facilidad a la hora de asociar un porcentaje de confianza a las predicciones, y teniendo en cuenta ésta combinarlas.
- Un nuevo caso es clasificado como función de la predicción de múltiples hipótesis, ponderadas por sus probabilidades.
- Incluso en casos donde se el uso de estos métodos se ha considerado imposible se puede obtener una aproximación de la solución óptima.

También presentan algunas dificultades:

- Necesidad de un conocimiento a priori, en caso de no tenerlo las probabilidades deberán ser estimadas.
- Coste computacional alto. En el caso general es lineal con el número de hipótesis candidatas.

2.1.1. Teorema de Bayes

Definición 2.1.- Denominamos probabilidad de A condicionada a B a la probabilidad de que se cumpla A bajo la condición de que se cumpla B. Se denota $P(A|B)$ y se calcula como el cociente entre la probabilidad de A y B entre la de B [9]. Esto es

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad .(1)$$

De la probabilidad condicionada podemos obtener la probabilidad de la intersección y la unión

$$\begin{cases} \mathrm{P(A \cap B) = P(A)P(B|A)} \ \text{ si } \mathrm{P(A) > 0} \quad (2) \\ \mathrm{P(A \cup B) = P(B)P(A|B)} \ si\ P(B) > 0 \quad (3) \end{cases}.$$

Teorema 2.1. de la probabilidad compuesta o Regla de la multiplicación.- Dados los sucesos $A_1, A_2, \dots, A_n$ con $P\left(\left[\cap_{i=1}^{n-1} A_i\right]\right) > 0$, entonces [10]

$$\begin{aligned} P([A_1 \cap A_2 \cap \dots \cap A_n]) &= P\left((A_1 \cap A_2 \cap \dots \cap A_{n-1}) \cap A_n\right) \\ &= P(A_1 \cap \dots \cap A_{n-1}) P(A_n|A_1 \cap \dots \cap A_{n-1}) \,. (4) \end{aligned}$$

Demostración.- Para la demostración detallada revisar la referencia bibliográfica [9].

Teorema 2.2. Teorema de Bayes – Podemos expresar la probabilidad condicional de un evento aleatorio A dado B en términos de la distribución de probabilidad condicional del evento B dado A y la distribución de probabilidad marginal de sólo A [5]. Esto es

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \ , \qquad (5)$$

donde:

· $P(A)$ es la probabilidad a priori,

· $P(B|A)$ es la probabilidad de B en la hipótesis A,

· $P(A|B)$ es la probabilidad a posteriori.

Demostración.- Utilizando las ecuaciones provenientes de la probabilidad condicionada sabemos [9]

$$P(A \cap B) = P(A)P(B|A), \quad (2)$$

y por otra parte, si intercambiamos las variables, obtenemos

$$P(A \cap B) = P(B)P(A|B). \quad (6)$$

Igualando ambos términos:

$$P(B)P(A|B) = P(A)P(B|A) \,, \quad (7)$$

y finalmente

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}. \quad (5)$$

El Teorema de Bayes tiene muy buenas aplicaciones en métodos de aprendizaje en Machine Learning. Para ello, podemos utilizar la notación de aprendizaje necesario.

Definición 2.2.- Dado un conjunto de hipótesis H y un conjunto observado de datos de entrenamiento D, definimos mejor hipótesis como la hipótesis más probable dado D además de un conocimiento inicial sobre las probabilidades a priori de varias hipótesis en H. [5]

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad ,(8)$$

donde

· $\mathrm{P}\ (\mathrm{h})$ es la probabilidad a priori de la hipótesis h,

· $P\ (D)$ es la probabilidad de observar el conjunto de entrenamiento T,

· $P\ (\ D \mid h\ )$ es la probabilidad de observar el conjunto de entrenamiento T en el caso de que se cumple la hipótesis h,

· $P\ (h \mid D\ )$ es la probabilidad a posteriori de h, cuando se ha procesado el conjunto de entrenamiento T.

Corolario 2.1.- De la aplicación del Teorema de Bayes deducimos las siguientes relaciones [10]

a. $P(h|D)$ se incrementa con $P(h)$ y con $P(D|h)$,
b. $P(h|D)$ decrece cuando $P(D)$ decrece debido a que cuanto más probable es que D sea observado independiente de h, menor será la evidencia de que D soporta la hipótesis h.

Definición 2.3.- Máximo a posteriori (MAP) se define como la hipótesis más probable aplicando el Teorema de Bayes, de la siguiente forma [5]

$$h_{MAP} = argmax_{h\in H} P(h|D) =$$

$$argmax_{h\in H} \frac{P(D|h)P(h)}{P(D)}. \quad (9)$$

Dado que $P(D)$ es una constante independiente de h podemos prescindir de ella resultando

$$h_{MAP} = argmax_{h\in H} P(D|h)P(h). \quad (10)$$

El máximo a posteriori nos ayudará en algunos escenarios en los que el clasificador considere un conjunto de hipótesis candidatas H y esté interesado en encontrar la $h \in H$ más probable dados los datos de observación D.

Corolario 2.2.- En el caso de que las probabilidades a priori de cada hipótesis en H sean las mismas $\left(P(h_i) = P\left(h_j\right)\ \forall h_i, h_j \in H\right)$, nos vale con tener en cuenta sólo el término $P(D|h)$ para encontrar la hipótesis más probable. [10]

Definición 2.4.- $P(D|h)$ se denomina verosimilitud de los datos D dado h [5].

Definición 2.5.- La hipótesis h que maximiza $P(D|h)$ se denomina hipótesis de máxima verosimilitud h_{MV} [5]

Entonces la hipótesis a buscar será la de máxima verosimilitud, que junto con el corolario 2.2. y la definición 2.3. obtenemos

$$h_{MV} = argmax_{h \in H} P(D|h). \quad (11)$$

En el apartado 2.3.3 se describe un ejemplo de la búsqueda de hipótesis de máxima verosimilitud.

2.1.2. Aprendizaje

Muchos de los procesos de aprendizajes implican la adquisición de conceptos generales a partir de ejemplos de entrenamiento específicos. Las personas ,por ejemplo, continuamente están aprendiendo conceptos o categorías en las que clasificar las instancias que encuentran en el mundo como “pájaro”, ”casa”, etc. Cada concepto se puede ver descrito como un subconjunto de objetos o eventos definidos en un conjunto mayor (en estos ejemplos pueden ser el subconjunto de animales que son los pájaros y el subconjunto de edificaciones que constituyen las casas). Podemos definir una función que nos devuelva un valor booleano definido sobre los conjuntos donde están contenidos. Esto es, verdadero si un animal es un pájaro o falso en caso contrario [5].

Definición 2.6.- El concepto de aprendizaje se define como la inferencia de un valor booleano mediante una función obtenida a partir de los ejemplos de entrenamiento [5].

Consideramos el aprendizaje de un clasificador [10] donde se definen un conjunto de hipótesis finito H sobre un espacio X, en el cual la tarea consiste en aprender sobre la función objetivo $c: X \to \{0,1\}$, y un conjunto de ejemplos de entrenamiento $<x_1, d_1>, \dots, <x_m, d_m>$ donde x_i es una instancia en X y d_i es un valor objetivo de x_i (i.e.,$c(x_i) = d_i$).

Teorema 2.3. Algoritmo de aprendizaje por fuerza bruta.- Podemos diseñar un algoritmo que devuelva como salida la hipótesis máxima a posteriori ,basada en el Teorema de Bayes, como sigue [5]

Paso 1.- Para cada hipótesis h perteneciente a H deberemos computar

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}. \quad (8)$$

Paso 2.- Devolveremos la hipótesis con la máxima probabilidad a posteriori

$$h_{MAP} = argmax_{h \in H} P(h|D). \quad (10)$$

Demostración.- Para la demostración detallada revisar la referencia bibliográfica [10]

Deberemos realizar algunas asunciones que simplifiquen la casuística [5]. Éstas son

1.-Los datos de entrenamiento están limpios, no tienen ruido $D = \{d_1, \dots, d_m\}$,

2.-El concepto objetivo está contenido en el espacio de hipótesis H,

3.-No tenemos conocimiento de las probabilidades a priori, por lo que consideramos cada h_i equiprobable.

Debemos tener conocimiento de las probabilidades $P(h), P(D|h)$, $P(D)$. A continuación se describe el cálculo de cada una de ellas.

$$P(h) = \frac{1}{|H|} \quad (12)$$

$$P(D|h) = \begin{cases} 1 \; si \; d_i = h(x_i) \forall \; d_i \in D \\ 0 \; en \; otro \; caso \end{cases} \quad (13)$$

Teorema 2.4. de Probabilidad Total – Sean los eventos $A_1, \dots, A_n$ mutualmente excluyentes con $\sum_{i=1}^{n} P(A_i) = 1$ entonces [11]

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i) \quad . \; (14)$$

Demotración.- Para la demostración detallada revisar la referencia bibliográfica [9].

Definición 2.7.- Una hipótesis h es consistente con un conjunto de ejemplos de entrenamiento D si y sólo si se cumple $h(x) = c(x)$ para cada ejemplo $< x, c(x) >$ en D.

Corolario 2.3.- Los valores de $P(D)$ son hallados mediante el Teorema de la Probabilidad Total dado en las fórmulas básicas, asumimos que las hipótesis son excluyentes.

$$P(D) = \sum_{h_i \in H} P(D|h_i)P(h_i)$$

$$= \sum_{h_i \in EV_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin EV_{H,D}} 0 \cdot \frac{1}{|H|} \quad , (15)$$

$$= \sum_{h_i \in EV_{H,D}} 1 \cdot \frac{1}{|H|} = \frac{|EV_{H,D}|}{|H|}$$

donde $EV_{H,D}$ es el conjunto de hipótesis de H que son consistentes con D.

Por tanto, vemos fácil el cálculo de la probabilidad $P(D)$ utilizando aquellas hipótesis que son consistentes, es decir, aquellas que cumplen la devolución correcta del valor en comparación con el ejemplo de entrenamiento. A continuación, partiendo de las probabilidades obtenidas podremos aplicar el Teorema de Bayes a cada hipótesis siguiendo el modelo de fuerza bruta

- Si la hipótesis es inconsistente con los ejemplos de entrenamiento D

$$P(h|D) = \frac{0 \cdot \frac{1}{|H|}}{P(D)} = 0 \; . \; (16)$$

- Si la hipótesis es consistente con los ejemplos de entrenamiento D

$$P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{\frac{|EV_{H,D}|}{|H|}} = \frac{1}{|EV_{H,D}|} \quad . \ (17)$$

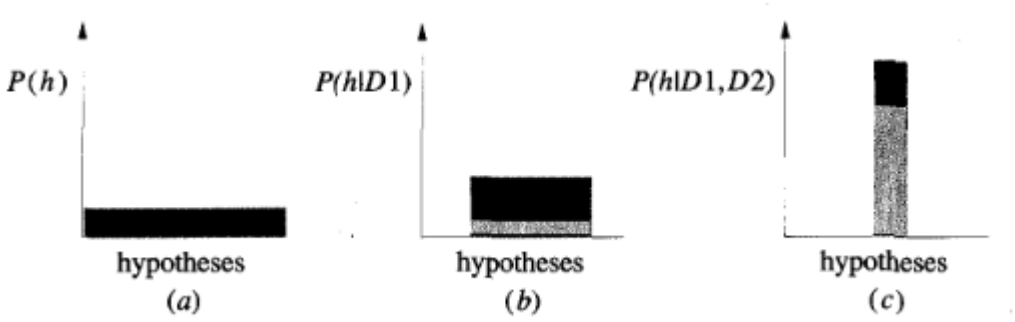


Ilustración II Evolución de las probabilidades de las hipótesis con los datos de entrenamiento

En la Ilustración II representamos las probabilidades $P(h|D)$ y su incremento medida que acumulamos el conocimiento por el conjunto de ejemplos D. Y cómo, mirando la figura (c), la probabilidad de las hipótesis inconsistentes se hace 0 mientras que la probabilidad a posteriori se incrementa para las hipótesis que quedan en el $EV_{H,D}$, espacio de versión. Utilizaremos esta idea para almacenar el máximo de elementos en el conjunto de entrenamiento y así mejorar el conocimiento del clasificador.

Definición 2.8.- Un algoritmo de aprendizaje es consistente si obtiene una hipótesis que no comete ningún error sobre los ejemplos de entrenamiento.

La definición de algoritmo consistente es interesante, pues nos otorga otro campo de estudio en el que realizar una investigación empírica en cada uno de los contextos. Es decir, una vez hemos definido el algoritmo de clasificación, podemos ir probando con los ejemplos de entrenamiento y realizar los test de comprobación. Podemos encontrar diversos ajustes utilizando determinados mediante agrupamientos y comprobar cuáles funcionan mejor o peor.

Definición 2.9.- Un algoritmo de aprendizaje genera una hipótesis MAP si

- Las hipótesis tienen la misma probabilidad a priori $(P(h_i) = P(h_j))$ para todo i, j.
- No hay ruido en los datos implica que $P(D|h) = 1$ si h es consistente y 0 en otro caso.

2.1.3. Clasificador bayesiano óptimo

Teorema 2.5.- La clasificación más probable para una nueva instancia, se obtiene como la combinación de las predicciones de todas las hipótesis mediante pesos a través de sus probabilidades a posteriori

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \quad . \quad (18)$$

Demostración.- Para la demostración detallada revisar la referencia bibliográfica [10].

Definición 2.10.- La óptima clasificación para la nueva instancia es el valor v_j, para el cual $P(v_j|D)$ es máximo. De esta forma

$$\text{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \quad . \quad (19)$$

Definición 2.11.- Los clasificadores que apliquen la clasificación óptima selección para el más probable los denominaremos clasificadores Bayesianos óptimos. Éste método maximiza la probabilidad de que una nueva instancia sea clasificada correctamente, dados los datos accesibles, el espacio de hipótesis y las probabilidades a priori.

2.1.4. Algoritmo de Gibbs

El clasificador bayesiano óptimo nos proporciona los mejores resultados que podemos obtener dado un conjunto de ejemplos de entrenamiento. El aprendizaje de conceptos mediante el espacio de versiones consistiría en sumar 'votos' para cada hipótesis ponderados por la probabilidad a posteriori de cada una. Esto es muy costoso ya que se computa la probabilidad a posteriori para cada hipótesis con el fin de clasificar cada nueva instancia.

Podemos encontrar una alternativa, menos óptima [5], mediante el algoritmo de Gibbs.

Teorema 2.6.- El algoritmo de Gibbs se procesa de la siguiente forma

1. Se escoge una hipótesis aleatoriamente h de H, de acuerdo a la distribución de probabilidades a posteriori sobre H
2. Se devuelve la clasificación de una entidad x dada por esa hipótesis h

Nota

- El error de clasificación en el algoritmo de Gibbs es como mucho dos veces el error esperado del clasificador óptimo Bayesiano [5]
- En el espacio de versiones, si se supone una distribución uniforme de probabilidades, el algoritmo de Gibbs consistiría en tomar una hipótesis al azar (mismas probabilidades para todas) [10]

Demostración.- Para la demostración detallada revisar la referencia bibliográfica [5]

2.1.5. Clasificador Naive-Bayes

El clasificador Naive-Bayes [5] es uno de los mejores métodos de aprendizaje en la práctica y en algunos campos se puede comparar a redes neuronales y árboles de decisión. Su funcionamiento se basa en el Teorema de Bayes y en algunas hipótesis que permiten la simplificación. Recibe el nombre de ingenuo por la hipótesis de independencia que se asume entre las variables que la predicción. Podremos aplicar estos métodos fundamentalmente en los siguientes casos

- Si se dispone de conjuntos de entrenamiento de tamaño no pequeño
- Los atributos que se utilizan deben ser independientes entre sí con respecto al concepto que se pretende aprender
- En ejemplos relacionados con diagnósticos o clasificación de textos es muy común este tipo de clasificador

La metodología asociada se determina mediante la definición de cada ejemplo x como el conjunto de los valores de sus atributos: $< a_1 , a_2 , a_3 , \dots , a_n >$. Describimos la función objetivo $f(x)$ que puede tomar cualquier valor de un conjunto finito V. La clasificación la obtendremos mediante el valor de máxima probabilidad a posteriori: V_{MAP}

$$v_{MAP} = argmax_{v_j \in V} P(v_j | a_1, a_2, \dots, a_n)$$

$$= argmax_{v_j \in V} \frac{P(a_1, a_2, \dots, a_n | v_j)\, P(v_j)}{P(a_1, a_2, \dots, a_n)} \quad .(20)$$

$$= argmax_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j)\, P(v_j)$$

Los términos se tienen que estimar basándonos en los ejemplos de entrenamiento

- $P(v_j)$ describe la frecuencia con la que ocurre cada valor v_j
- Hay demasiados términos de la forma $P(a_1, a_2, \dots, a_n | v_j)$. Harían falta demasiados ejemplos de entrenamiento para obtener una buena estimación

Mediante el clasificador Naive hemos de suponer que los atributos son independientes entre sí con respecto al concepto objetivo, como se ha dicho anteriormente para su aplicación, de esta forma

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j) \quad . \quad (21)$$

Definición 2.12.- La aproximación del clasificador de Naive-Bayes es

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad . \quad (22)$$

$P(a_i | v_j)$ resultan en este caso más fácil de estimar que las $P(a_1, a_2, \dots, a_n)$

Corolario 2.4.- El funcionamiento del algoritmo en pseudocódigo tiene la forma [10]

$AprendizajeDeNaiveBayes(ejemplosInput)$

$Para\ cada\ valor\ del\ resultado\ v_j$

|_ $Obtener\ estimación\ P'(v_j)\ de\ la\ probabilidad\ P(v_j)$

|_ $Para\ cada\ valor\ ai\ de\ cada\ atributo\ a$

$Obtener\ una\ estimación\ P'(a_i|v_j)\ de\ P'(a_i|v_j)$

$InstanciaDeClasificador(x)$

$Devolver \quad v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$

2.1.6. Estimación de probabilidades

Podemos apreciar que surge un problema en la ecuación (17) donde se especifica un productorio en el que cualquier estimación $P(a_i|v_j) = 0$ devolverá 0, esto se produce en la mayor parte de ocasiones en las que se tienen pocos datos. Veremos más en profundidad un ejemplo claro más adelante, en el apartado 2.3.3.

Definición 2.13.- La Estimación-m es la probabilidad condicionada del suceso a_i con respecto a v_j de la forma

$$P'(a_i|v_j) = \frac{n_c + mp}{n + m}, \quad (23)$$

donde

- n: número de ejemplos de entrenamiento con valor v_j,
- n_c: fracción de n con valor ai en el atributo a,
- p: estimación a priori de $P(a_i|v_j)$,
- m: peso de la estimación a priori,
- si $m = 0$ se obtiene la estimación por defecto. En caso contrario, la estimación observada $(\frac{n_c}{n})$ y el conocimiento previo (p) son combinados teniendo en cuenta el peso m.

Si no tenemos información adicional, la probabilidad a priori se puede obtener suponiendo la distribución uniforme $p = \frac{1}{k}$, donde k es el número de valores distintos para el atributo a.

2.1.7. Clasificación de textos

Como se ha comentado anteriormente, la clasificación de textos es un ejemplo de la importancia de la práctica de los métodos de aprendizaje bayesiano. En este caso deberemos definir

- Las instancias que son los documentos de texto. El espacio de instancias son todos los posibles documentos de texto.
- Concepto a aprender o clasificación que queramos realizar.
- Función objetivo $f: documento \rightarrow \{v_1, v_2, \dots\} = V$, donde V son los valores de salida que queremos obtener, por ejemplo, texto de carácter positivo o negativo.

También deberemos plantear

- Cómo representar un documento de texto en términos de valores de atributos. Esto se realiza con un vector con tantos atributos como palabras tiene el documento, el valor de a_i es la palabra que hay en la posición i:

$$v_{NB} = \text{argmax}_{v_j \in \{+,-\}} P(v_j) \prod_{i=1}^{longitud(documento)} P(a_i = w_k | v_j) \ , \quad (24)$$

donde w_k es la k-ésima palabra de vocabulario utilizado. La suposición del aprendizaje bayesiano $P(a_1, a_2, \ldots, a_n | v_j) = \prod P(a_i | v_j)$ no se cumple. La probabilidad de una palabra en una posición depende de las palabras en el resto de posiciones y esto es un punto muy interesante que podemos relacionar con algunos métodos de programación dinámica y la distancia entre dos palabras en términos de cuántos cambios se necesitan para pasar de una a otra. Esto se puede aplicar para obtener la distancia entre dos sentencias de acuerdo con su posición.

- Cómo estimar las probabilidades requeridas por el clasificador Naive-Bayes. Las probabilidades a estimar:

  · $P(+)$ y $P(-)$ serán las fracciones de cada tipo obtenidas en la parte del aprendizaje.

  · $P(a_i = w_k \mid +)$, $P(a_i = w_k \mid -)$ para todas las palabras del diccionario en cada una de las posibles posiciones. Facilitando la simplificación supondremos que la probabilidad de la obtención de una palabra en concreto será independiente de la posición:

  $$P(a_1 = w_k | v_j) = P(a_2 = w_k | v_j) = \cdots = P(w_k | v_j).$$

Para evitar el problema comentado antes con que alguna estimación tenga el valor 0 utilizamos la estimación-m de forma que

$$P(w_k | v_j) = \frac{n_i + 1}{(n + |vocabulario|)} \ , \quad (25)$$

donde n es el número de palabras de todos los textos con valor v_j y n_k es el total de apariciones de w_k en las n palabras. Vocabulario es el conjunto de palabras distintas utilizadas en los textos de entrenamiento.

2.1.8. Reducción de dimensionalidad

Debido a la alta dimensión del espacio de términos existentes en un texto deberemos realizar una reducción de dimensionalidad mediante la selección de patrones más representativos dentro de un texto. De esta forma incrementamos la eficiencia sin disminuir la precisión. Se pueden aplicar diferentes técnicas para esto:

- Eliminar las palabras características consideradas de poco valor. Por ejemplo: preposiciones, determinantes, … La intuición nos dice que las palabras con más valor serán los adjetivos.
- Parametrizar el texto, sustituciones de palabras por otras que lo representen. Por ejemplo, mediante Stemming (que ya explicaremos más adelante) o tratando con sinónimos conocidos.

- Diccionario de términos con palabras relevantes que serán las que se recuenten a la hora de definir un texto.
- Diccionario de términos no relevantes que serán obviados.
- Establecer filtros sobre palabras, frases o conjuntos de palabras. Por ejemplo, sólo utilizar palabras de más de tres letras.

2.2. Análisis Semántico Latente (Latent Semantic Analysis, LSA)

El análisis semántico latente [4] o LSA, por sus siglas en inglés, es una teoría y método para la extracción y representación del significado de las palabras. El significado se estima utilizando métodos de estadística computacional sobre un gran conjunto de datos de tipo texto (Landauer & Dumais,1997), a esto lo denominaremos corpus. Este corpus incorpora un conjunto de restricciones mutuas que determina la similaridad semántica de las palabras y conjuntos de éstas. Estas restricciones pueden ser resueltas utilizando métodos de álgebra lineal, en particular la descomposición en valores singulares. El LSA ha sido utilizado para reflejar el conocimiento humano en múltiples formas. Por ejemplo, puede imitar la ordenación de palabras y la categorización de juicios humanos; puede simular palabra a palabra o párrafo a palabra los datos léxicos principales; además de estimar la coherencia de documentos.

La aplicación del LSA se ha realizado en numerosas áreas como: herramientas para la mejora individual de los estudiantes en materia de educación, estudio de guiado de grupos de discusión, proporcionar feedback a los pilotos sobre las técnicas de aterrizaje, ayudando en el diagnóstico de desórdenes mentales a partir de textos escritos por pacientes, asociar candidatos con trabajos, facilitar tutores automatizados… [4]

Los orígenes del LSA se remontan a un artículo escrito por Deerwester, Dumais, Furnas, Landauer y Harshman 1990 en el cual describen un nuevo acercamiento a la indexación automática y recuperación de información. Intentan superar un problema fundamental que se produce con las técnicas de recuperación que pretenden hacer coincidir palabras de búsqueda con palabras en documentos (como por ejemplo en los buscadores de información en internet) [6]. Ellos suponen que

"[..] there is some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval" [12]

La solución que proponen se origina a través de un método de análisis estadístico. Llevan a cabo un análisis de indexación semántica latente (LSI), técnica estadística para estimar la estructura latente. Esta técnica utiliza un valor singular de descomposición que parte una gran matriz de datos de asociación de término-documento y permite construir un "espacio semántico" en el que se asocian entre sí términos y documentos [4]. El fundamento estadístico se encuentra entonces ocurrencia de los términos en cada uno de los documentos.

Para llevar a cabo el uso del análisis semántico latente en la clasificación de textos deberemos partir del mismo punto que en los métodos bayesianos: la frecuencia de los términos o palabras en los textos. Para ello debemos construir una matriz donde las columnas son cada uno de los textos, las filas representan cada una de las palabras y cada posición de la matriz representa el número de repeticiones de una palabra en un documento. A continuación, realizaremos una transformación sobre esta matriz obtenida intentando, mediante ponderaciones, discriminar las palabras que se repiten frecuentemente, las cuales no aportaran casi valor, de las que son moderadamente infrecuentes.

El uso del LSA nos aporta ciertas ventajas muy relacionadas con trabajar sobre un espacio vectorial, donde los documentos y palabras serán representados como vectores. Además esto nos permite aprender y desarrollarlo como un modelo computacional que constituye una buena representación del conocimiento [6]. Sin embargo, tiene algunas ventajas

muy claras. Por ejemplo, tiene una gran probabilidad de caer en un error si se trabaja sobre todo con términos no conocidos [13]. Sin embargo, el modelo se comporta bien y está diseñado para que se guarden todos los vectores y se tengan en cuenta para futuras clasificaciones.

2.2.1. Descomposición en Valores Singulares (DVS)

2.2.2. Definición de SVD

Podemos ver la descomposición en valores singulares [14] o SVD ,por sus siglas en inglés, desde distintos puntos de vista. Por un lado se puede pensar en éste como un método para la transformación de variables correlacionadas a un conjunto en el que no lo están. También podemos describirlo como un método para la reducción de los datos viendo cuáles están más relacionados y cuáles indican una mayor variación.

La idea más básica detrás del SVD es la de coger una alta dimensionalidad para disminuirla, esto es, una variedad de conjunto de puntos de datos que se reducen a un espacio de menor dimensión que mantiene la subestructura de los datos originales más claramente y lo ordena de la mayor variación a la menor. Lo que hace el SVD práctico para el NLP (Procesamiento del lenguaje natural) es el hecho de que se puede de una forma muy simple ignorar aquellas variaciones por debajo de un determinado umbral, lo que permite reducir la cantidad de datos, pero manteniendo las principales relaciones. [14]

2.2.3. ¿Para qué sirve el SVD?

Entre algunas de las aplicaciones de estos métodos podemos destacar [8] el cálculo numérico (para el cálculo de rango de la matriz, resolución de sistemas lineales, de ajustes por mínimos cuadrados o cálculo de valores propios), tratamiento de imágenes (en algunos casos nos permite realizar una compresión de las imágenes pudiendo representar un archivo de MxN píxeles en un menor número de datos utilizando la matriz diagonal y las propiedades de información innecesaria)

2.2.4. Terminología y notación

Definimos un vector como una secuencia de números que representan una medida en una determinada dimensión. Por ejemplo: $\vec{x} = [1,2,3]$ es un vector tridimensional. El componente i-ésimo (x_i) se corresponde con el valor en la i-ésima dimensión. Sobre los vectores podemos definir determinadas características y operaciones:

Definición 2.14.- Sea $\vec{v}$ un vector su longitud se denota con $|\vec{v}|$ y se calcula como la raíz cuadrada de la suma de los cuadrados de cada componente:

$$|\vec{v}| = \sqrt{\sum_{i=1}^{n} v_i^2} \ . \quad (26)$$

Dados $\vec{u}$ y $\vec{v}$ dos vectores con $\vec{u} = [u_1, u_2, \dots, u_n]$ y $\vec{v} = [v_1, v_2, \dots, v_n]$ la operación de adición o suma está definida:

$$\vec{u} + \vec{v} = [u_1 + v_1, u_2 + v_2, \dots, u_n + v_n].$$

El producto entre un x y un vector $\vec{v}$ =$[v_1, v_2, \dots, v_n]$ resulta un vector cuyos componentes son: $[x\vec{v}_1, x\vec{v}_2, \dots, x\vec{v}_n]$

Dados $\vec{u}$ y $\vec{v}$ dos vectores con $\vec{u} = [u_1, u_2, \dots, u_n]$ y $\vec{v} = [v_1, v_2, \dots, v_n]$ la operación de producto escalar está definida:

$$< \vec{u}, \vec{v} > = \vec{u} \cdot \vec{v} = \sum_{(i=1)}^{n} x_i y_i \ . \ (27)$$

Definición 2.15.- Dos vectores $\vec{u}$ y $\vec{v}$ se dicen ortogonales si su producto escalar es igual a 0. En dos dimensiones esto es equivalente a que los vectores formen un ángulo de 90 grados.

Definición 2.16.- Vector unitario es aquel cuyo módulo (longitud) es igual a 1. Podemos definir también la operación de normalizar un vector $\vec{v}$ en la que se divide cada componente de éste por la longitud total:

$$\vec{v}_{normalizado} = \left[\frac{v_1}{|\vec{v}|}, \frac{v_2}{|\vec{v}|}, \dots, \frac{v_n}{|\vec{v}|}\right]. \ (28)$$

Definición 2.17.- Dos vectores son ortonormales si son de longitud unitaria y son ortogonales entre sí.

Definición 2.18.- Definimos como array una lista de n elementos con posición determinada de 1 a n cuya representación puede verse

$$A = (a_1 \quad a_2 \quad \dots \quad a_n).$$

Definición 2.19.- Sean m, n dos enteros ≥ 1 y sea a_{ij} con i = 1..m ,j= 1 .. n un array de números de forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

Cada a_{ij} son elementos de A. La secuencia de números $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$ se denomina fila i-ésima de A. La secuencia de números $A_i = (a_{j1}, a_{j2}, \dots, a_{jn})$ se denomina columna j-ésima de A. Vemos entonces que una matriz es una lista de arrays.

Ejemplo 2.1.- Las matrices nos permiten representar secuencias de vectores como por ejemplo la cantidad de veces que sale una determinada palabra en un determinado texto

$$Frecuencias = \begin{array}{c} \\ universidad \\ estudios \\ desempleo \end{array} \begin{array}{c} párrafo1 \\ 1 \\ 0 \\ 0 \end{array} \begin{array}{c} párrafo2 \\ 0 \\ 2 \\ 0 \end{array} \begin{array}{c} párrafo3 \\ 3 \\ 1 \\ 1 \end{array} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Veamos algunas características y operaciones de las matrices

Definición 2.20.- Se denomina matriz cuadrada a aquella que tiene el mismo número de filas y de columnas, $n * n$. n-cuadrada.

La transposición sobre una matriz A convierte sus filas en columnas y viceversa. Se expresa mediante A^T.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

$$A^T = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{pmatrix}$$

El producto de dos matrices sólo se puede producir cuando la primera tiene el mismo número de columnas que de filas la segunda. Sean $A\ y\ B$ dos matrices de dimensiones p x n y n x m respectivamente, su producto será C cuyos componentes vienen dados por el producto escalar de la fila i-ésima de A y la columna j-ésima de B guardándose en c_{ij}.

$$C = \begin{pmatrix} \sum_{i=1}^{n} a_{1i}b_{i1} & \sum_{i=1}^{n} a_{1i}b_{i2} & \dots & \sum_{i=1}^{n} a_{1i}b_{im} \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^{n} a_{pi}b_{i1} & \sum_{i=1}^{n} a_{pi}b_{i2} & \dots & \sum_{i=1}^{n} a_{pi}b_{im} \end{pmatrix}$$

La matriz identidad es aquella matriz cuya diagonal sólo contiene 1s y el resto de entradas contienen 0s. Se denota como I_n (o simplemente I) y es el elemento neutro en el producto de matrices. Por ejemplo:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Una matriz A se dice ortogonal si cumple que $AA^T = A^TA = I$.

Denominamos matriz diagonal a aquella matriz cuyas entradas $a_{ij} = 0\ \forall\ i, j$ con $i \neq j$. Por ejemplo:

$$A = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}.$$

El determinante de una matriz A cuadrada es igual a la suma de productos de los elementos de una fila por sus adjuntos correspondientes. Se denota $|A|$ o $\det(A)$. De forma:

$$|A| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix} =$$

$$a_{11}\begin{vmatrix} a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m2} & \dots & a_{mn} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{vmatrix} + \dots + a_{1n}\begin{vmatrix} a_{21} & a_{22} & \dots \\ \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots \end{vmatrix} .$$

Definición 2.21.- Un autovector es un vector distinto del nulo que satisface la ecuación

$$A\vec{v} = \lambda\vec{v} \ , \quad (24)$$

donde A es una matriz cuadrada, λ es un escalar al que se le denomina autovalor y $\vec{v}$ es el autovector. Para encontrar estos valores podemos tratar el problema como un sistema de ecuaciones de forma que:

$$A\ \vec{v} = \lambda\vec{v} = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} . \quad (29)$$

2.2.5. Cómo se realiza el SVD

Teorema 2.7.- Se puede realizar la factorización de una matriz A como producto de matrices U, V y S, de las cuales U y V son ortogonales (de orden m y n respectivamente) y S es una matriz diagonal (de tamaño $m * n$) de forma que [12]

$$A_{mn} = \ U_{mm}\, S_{mn}\, V_{nn}^{T} \ . \ (30)$$

Donde $U^T U = I$, $V^T V = I$. Las columnas de U son autovectores ortonormales de AA^T, las columnas de V son autovectores ortonormales de $A^T A$, y S contiene las raíces cuadradas de los autovalores de U o V ordenados descendientemente.

Demostración.- Para la demostración detallada revisar la referencia bibliográfica [14].

2.2.6. Un ejemplo completo

Veamos un ejemplo aplicado a nuestro objetivo, el análisis de las relaciones entre documentos. Para facilitar muchas de las operaciones numéricas que deberemos realizar, utilizaremos el software Octave, el cual cuenta con la misma sintaxis que Matlab, aunque de licencia gratuita. Por ello, dados los siguientes textos:

1. $-Juan\ estudia\ en\ la\ universidad$

2.- $María\ estudia\ en\ el\ instituto$

3. $-Ernesto\ trabaja\ en\ un\ despacho$

4. $-Marta\ trabaja\ como\ profesora\ en\ el\ instituto$

5. $-Fernando\ trabaja$

De esta forma podemos coger las palabras clave:

$$estudia, universidad, instituto, despacho\ y\ trabajo,$$

y crear la matriz de $palabras\ x\ documentos$:

$$A = \begin{pmatrix} & estudia & universidad & instituto & despacho & trabaja \\ Documento1 & 1 & 1 & 0 & 0 & 0 \\ Documento2 & 1 & 0 & 1 & 0 & 0 \\ Documento3 & 0 & 0 & 0 & 1 & 1 \\ Documento4 & 0 & 0 & 1 & 0 & 1 \\ Documento5 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

La matriz traspuesta resultante tendrá la forma:

$$A^T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Los productos de ambas matrices son

$$AA^T = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

y

$$A^TA = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Pasamos a calcular los autovalores de la primera mediante:

$$\begin{vmatrix} (2-\lambda) & 1 & 0 & 0 & 0 \\ 1 & (2-\lambda) & 0 & 1 & 0 \\ 0 & 0 & (2-\lambda) & 1 & 1 \\ 0 & 1 & 1 & (2-\lambda) & 1 \\ 0 & 0 & 1 & 1 & (1-\lambda) \end{vmatrix} = 0\,.$$

Obtenemos $\lambda = 4.02641795, \lambda = 2.9137317\,, \lambda = 1.415336\,, \lambda = 0.11339035, \lambda = 0.53112399$. Y sus correspondientes autovectores:

$$\lambda = 4.02641795\,,$$

$$\vec{v}_1 = [-0.20391781\,, 0.63846665, -0.64749987\,, 0.21000902\,, 0.29566533].$$

$$\lambda = 2.9137317,$$

$$\overrightarrow{v_2} = [-0.41322271\,, 0.58338722\,, 0.37856986\,, -0.39620504\,, -0.43429571].$$

$$\lambda = 1.415336,$$

$$\overrightarrow{v_3} = [-0.49690556\,, -0.41025, -0.48531485\,, 0.0905987, -0.58398731].$$

$$\lambda = 0.11339035,$$

$$\overrightarrow{v_4} = [-0.63344411\,, -0.10540725\,, 0.4261637\,, 0.53747523\,, 0.34226122]$$

$$\lambda = 0.53112399,$$

$$\overrightarrow{v_5} = [-0.37349424, -0.26945117, -0.14241758\,, -0.70839962, 0.51554372].$$

Una vez tenemos estos autovalores podemos construir la matriz diagonal S con sus raíces cuadradas resultando

$$S = \begin{pmatrix} 2.006593618 & 0 & 0 & 0 & 0 \\ 0 & 1.706965642 & 0 & 0 & 0 \\ 0 & 0 & 1.189678950 & 0 & 0 \\ 0 & 0 & 0 & 0.728782540 & 0 \\ 0 & 0 & 0 & 0 & 0.336734844 \end{pmatrix}.$$

También, obtenidos los autovectores asociados podemos montar la matriz U que tiene la forma

$$U = \begin{pmatrix} -0.203917810 & 0.638466652 & -0.647499871 & -0.295665333 & -0.210009020 \\ 0.413222710 & 0.583387221 & 0.378569862 & 0.434295714 & 0.396205042 \\ -0.496905560 & -0.410249998 & -0.485314853 & 0.583987305 & -0.090598704 \\ -0.633444107 & -0.105407255 & 0.426163703 & -0.342261223 & -0.537475234 \\ -0.373494238 & -0.269451173 & -0.142417582 & -0.515543720 & 0.708399622 \end{pmatrix}.$$

Pasamos a trabajar con la segunda matriz $A^T A$ y de la misma forma trabajamos para obtener sus autovalores y autovectores

$$\begin{vmatrix} (2-\lambda) & 1 & 0 & 0 & 0 \\ 1 & (2-\lambda) & 0 & 1 & 0 \\ 0 & 0 & (2-\lambda) & 1 & 1 \\ 0 & 1 & 1 & (2-\lambda) & 1 \\ 0 & 0 & 1 & 1 & (1-\lambda) \end{vmatrix} = 0\,.$$

Obtenemos $\lambda = 4.02641795, \lambda = 2.9137317\,, \lambda = 1.415336\,, \lambda = 0.11339035, \lambda = 0.53112399$, cuyos autovectores correspondientes son

$$\lambda = 4.02641795\,,$$

$$\vec{v}_1 = [-0.30755631, -0.10162387, -0.52161375, -0.24763637, -0.74945116].$$

$$\lambda = 2.9137317,$$

$$\overrightarrow{v_2} = [0.71580461, 0.37403603, 0.28001733, -0.24033876\,, -0.45994389].$$

$$\lambda = 1.415336,$$

$$\overrightarrow{v_3} = [-0.22605259\,, -0.54426438\,, 0.67642919\,, -0.40793766\,, -0.1694312].$$

$$\lambda = 0.11339035,$$

$$\overrightarrow{v_4} = [0.19022188\,, -0.40569761\,, 0.12628526\,, 0.80131901\,, -0.37571926].$$

$$\lambda = 0.53112399,$$

$$\overrightarrow{v_5} = [0.55294552\,, -0.62366287\,, -0.41952947\,, -0.26905058\,, 0.23854284].$$

Ahora podemos montar también la matriz que nos falta a partir de los autovectores obtenidos

$$V = \begin{pmatrix} -0.30755631 & 0.71580461 & -0.22605259 & 0.19022188 & 0.55294552 \\ -0.10162387 & 0.37403603 & -0.54426438 & -0.40569761 & -0.62366287 \\ -0.52161375 & 0.28001733 & 0.67642919 & 0.12628526 & -0.41952947 \\ -0.24763637 & -0.24033876 & -0.40793766 & 0.80131901 & -0.26905058 \\ -0.74945116 & -0.45994389 & -0.1694312 & -0.37571926 & 0.23854284 \end{pmatrix}.$$

La cual deberemos trasponer para poder obtener la expresión correcta de la factorización

$$V^T = \begin{pmatrix} -0.30755631 & -0.10162387 & -0.52161375 & -0.24763637 & -0.74945116 \\ 0.71580461 & 0.37403603 & 0.28001733 & -0.24033876 & -0.45994389 \\ -0.22605259 & -0.54426438 & 0.67642919 & -0.40793766 & -0.1694312 \\ 0.19022188 & -0.40569761 & 0.12628526 & 0.80131901 & -0.37571926 \\ 0.55294552 & -0.62366287 & -0.41952947 & -0.26905058 & 0.23854284 \end{pmatrix}.$$

2.3. Cómo se realiza el LSA

Una vez hemos descrito la principal herramienta del LSA , la descomposición en valores singulares, los pasos siguientes se basan en cómo relacionar cada uno de los textos. Para ello podemos seguir con el ejemplo planteado antes del cual ya hemos obtenido la descomposición.

Dadas las matrices U, S y V^T podemos escoger una k-dimensionalidad para tratar los datos, lo que nos ayudará a representarlos. Esto es, escoger un entero k que define el número de valores singulares que se cogerán y por tanto la dimensión de los vectores asociados en palabras y documentos. Para nosotros es más fácil escoger $k = 2$ para ver en 2 dimensiones cómo funcionan las relaciones [4]. En este caso

$$S_2 = \begin{pmatrix} 2.006593618 & 0 \\ 0 & 1.706965642 \end{pmatrix}.$$

Ahora también debemos restringir las dimensiones de las otras dos matrices y lo que representa cada vector que se encuentra en la matriz

$$\begin{matrix} estudia \\ universidad \\ instituto \\ despacho \\ trabaja \end{matrix} \quad U_2 = \begin{pmatrix} -0.203917810 & 0.638466652 \\ -0.413222710 & 0.583387221 \\ -0.496905560 & -0.410249998 \\ -0.633444107 & -0.105407255 \\ -0.373494238 & -0.269451173 \end{pmatrix},$$

$$V_2^T = \begin{pmatrix} -0.30755631 & -0.10162387 & -0.52161375 & -0.24763637 & -0.74945116 \\ 0.71580461 & 0.37403603 & 0.28001733 & -0.24033876 & -0.45994389 \end{pmatrix}.$$

Una vez hemos obtenido U_2, S_2 y V_2^T podemos representar en dos dimensiones los vectores asociados a las palabras, realizando el producto $U_2 S_2$ escogiendo cada fila de U_2 sabiendo el término que representa

$$\vec{estudia} = \begin{bmatrix} -0.40918 \\ 1.08984 \end{bmatrix},$$

$$\vec{universidad} = \begin{bmatrix} -0.82917 \\ 0.99582 \end{bmatrix},$$

$$\vec{instituto} = \begin{bmatrix} -0.99709 \\ -0.70028 \end{bmatrix},$$

$$\vec{despacho} = \begin{bmatrix} -1.27106 \\ -0.17993 \end{bmatrix},$$

$$\vec{trabaja} = \begin{bmatrix} -0.74945 \\ -0.45994 \end{bmatrix}.$$

Si representamos cada uno de los términos en un gráfico de dos dimensiones

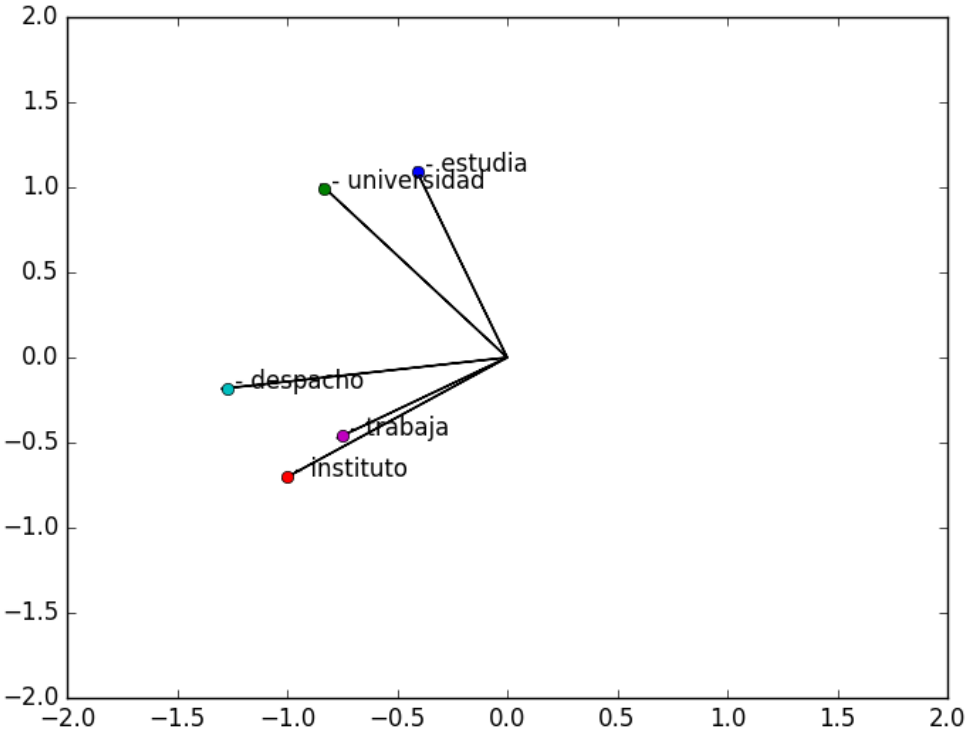


Ilustración III Representación de los términos en 2-dimensiones

A continuación, procedemos a realizar de forma similar la obtención de los vectores que representan a cada uno de los documentos. Para ello de nuevo utilizamos la matriz S_2 y realizamos el producto con cada una de las columnas de V^T

$$Documento\ 1 = \begin{bmatrix} -0.61714052026651034 \\ 1.2218538729621633 \end{bmatrix},$$

$$Documento\ 2 = \begin{bmatrix} -0.20391780997475978 \\ 0.63846665241574252 \end{bmatrix},$$

$$Documento\ 3 = \begin{bmatrix} -1.0466668174638747 \\ 0.47797996568753315 \end{bmatrix},$$

$$Documento\ 4 = \begin{bmatrix} -0.4969055598763818 \\ -0.41024999751959801 \end{bmatrix},$$

$$Documento\ 5 = \begin{bmatrix} -1.5038439054606776 \\ -0.7851084257187414 \end{bmatrix}.$$

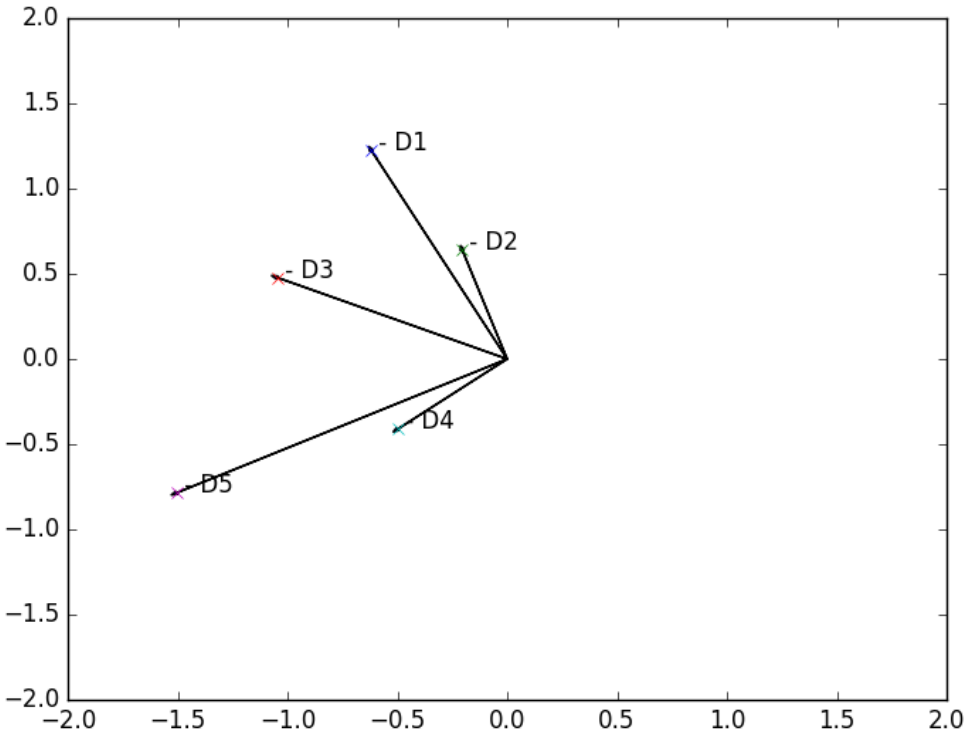


Ilustración IV Representación de los documentos en 2-dimensiones

Podemos juntar ambas representaciones en la misma gráfica viendo cómo interactúan los términos con los documentos.

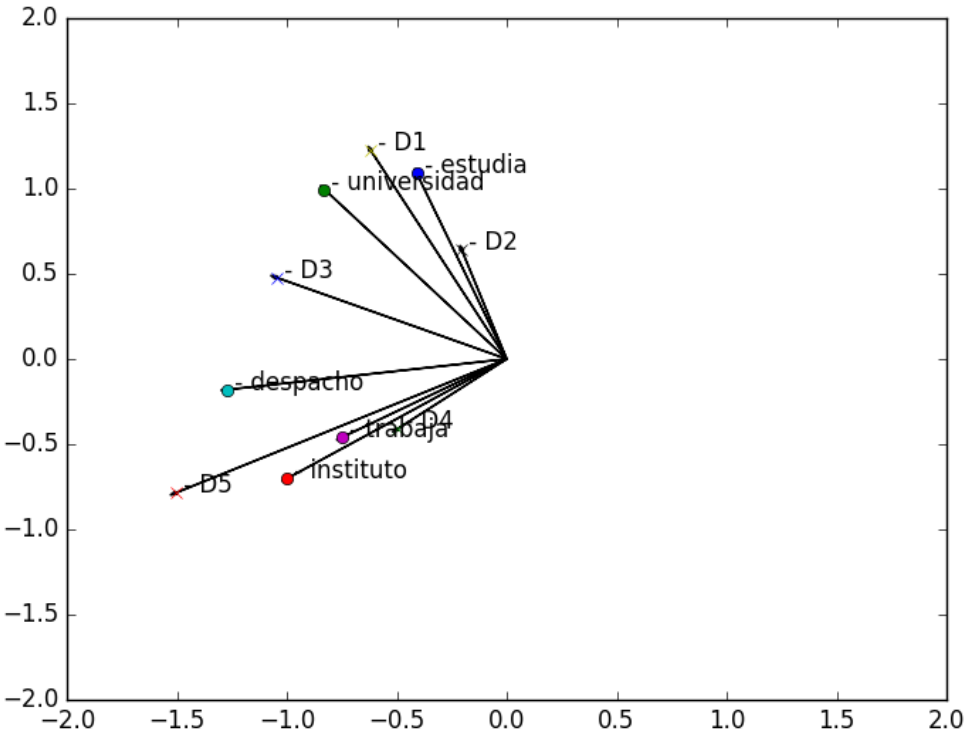


Ilustración V Representación de los términos y los documentos estudiados

Claramente percibimos cómo el término “estudia” se encuentra más cercano a los documentos 1 y 2, “universidad” se encuentra cercano al documento 1, “trabaja” se encuentra cercano a los documentos 4 y 5 principalmente, y alejado de 1 y 2. Sin embargo, para todas estas mediciones que realizamos con la intuición de distancia, deberemos definir matemáticamente la que queremos utilizar.

2.4. El clustering de documentos

Previo a realizar afirmaciones acerca de grupos o clusters que tengan lugar, es necesario determinar cuál es la distancia o similitud a utilizar y cómo medirla. Ésta representará el grado de cercanía o separación entre dos documentos, y debe corresponderse con diferencias claras que dependen de los datos o del contexto, por ello, no hay una clara medida que encaje con todos los problemas de clustering. [15]

Definición 2.22. - Dada una función $d(a,b)$ de $X\ x\ X \rightarrow \mathbb{R}$ decimos que es una distancia o métrica si se cumplen las siguientes condiciones [16]

$(i)\ d(a,b) \geq 0\ (No\ negatividad)$,

$(ii) d(a,b) = 0\ si\ , y\ sólo\ si, a = b$,

$(iii)\ d(a,b) = d(b,a)$,

$$(iv)\ \ d(a,c) \leq d(a,b) + d(b,c), \forall\ a,b\ y\ c \in X\ (Desigualdad\ triangular).$$

Veamos varios ejemplos interesantes de métricas que podemos tener en cuenta a la hora de comparar vectores.

Definición 2.23.- Dados dos documentos su Distancia Euclídea viene definida por

$$D_E(\overrightarrow{d_1}, \overrightarrow{d_2}) = \sqrt{\sum_{i=1}^{n}(\overrightarrow{d_{1_i}} - \overrightarrow{d_{2_i}})^2}\ ,\quad (31)$$

donde $\overrightarrow{d_1}$ y $\overrightarrow{d_2}$ son vectores n-dimensionales y d_{j_i} corresponde a la componente i-ésima del documento j-ésimo.

Es la métrica estándar para los problemas geométricos y la más común para la distancia entre dos puntos que puede ser fácilmente medida con una regla en dos y tres dimensiones. Es muy utilizada en los problemas de clustering, por ejemplo, en el algoritmo K-means.

Definición 2.24.- Dados dos documentos su Similitud Coseno viene definida por

$$Similaridad_{Coseno}(\overrightarrow{d_1}, \overrightarrow{d_2}) = \frac{<\overrightarrow{d_1}, \overrightarrow{d_2}>}{|\overrightarrow{d_1}| \cdot |\overrightarrow{d_2}|}\ ,\quad (32)$$

donde $\overrightarrow{d_1}$ y $\overrightarrow{d_2}$ son vectores n-dimensionales, y están más relacionados cuanto más se aproxime a 1.

Siendo expresada la representación de los documentos como vectores, podemos cuantificar la cercanía de éstos mediante la llamada Similitud Coseno. Ésta es una de las medidas más populares en la aplicación de clustering en documentos de texto y en extracción de información de éstos.

2.5. Clasificación de nuevos datos

Una vez hemos tratado todos los datos y tenemos los vectores asociados a cada documento podremos realizar estudios sobre nuevos. Por ejemplo, dado el siguiente documento

$$6. - El\ instituto\ de\ Marta\ está\ cerca\ del\ despacho\ de\ Ernesto$$

y vemos que los términos que contiene y que hemos estudiado son

$$instituto\ y\ despacho.$$

A continuación, calcularemos la consulta mediante el cálculo del centro de los términos que contiene a partir de sus vectores creados en el entrenamiento. Esto es

$$Consulta = \frac{\sum_{i=1}^{n} \vec{v}_i}{n}\ ,\quad (33)$$

donde n es el número de términos que contiene el documento y los $\vec{v}$ los vectores asociados a éstos. En este caso

$$Consulta = \frac{\begin{bmatrix}-0.99709\\-0.70028\end{bmatrix} + \begin{bmatrix}-1.27106\\-0.17993\end{bmatrix}}{2} = \begin{bmatrix}-1.134075\\-0.440105\end{bmatrix}.$$

Veamos gráficamente donde se encuentra la consulta realizada con respecto a los datos previamente calculados.

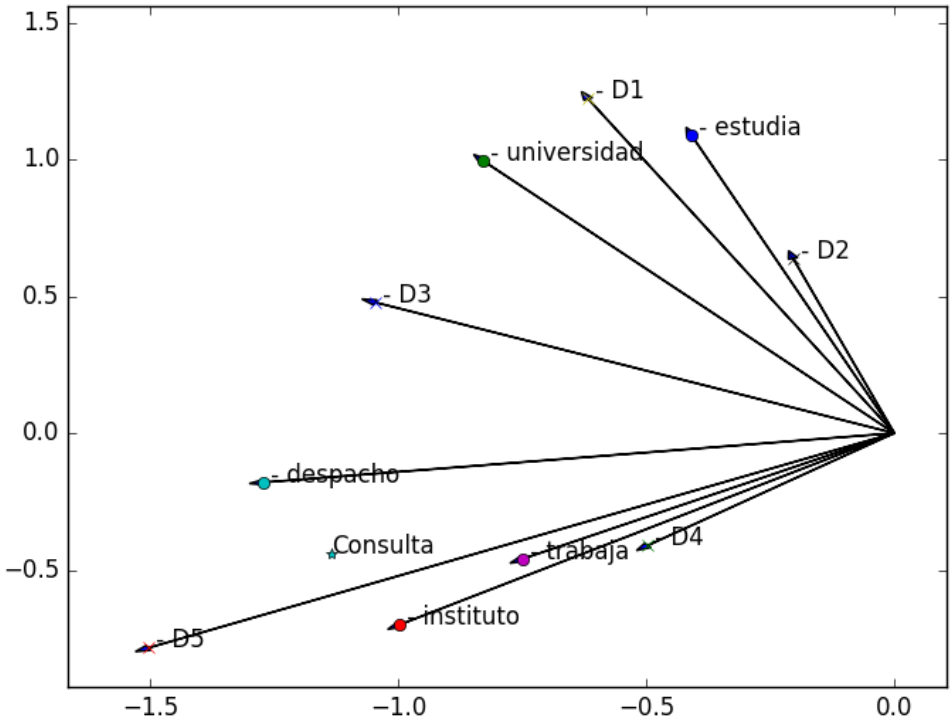


Ilustración VI Representación de los documentos y términos

Nuevamente vemos que la consulta realizada, al no tener ninguna relación con los documentos 1 y 2, pero sí con 3,4 y 5, se encuentra alejado de los primeros y cercana a estos últimos.

Para ello podemos calcular las distancias coseno entre los vectores de los documentos y la consulta aplicando la expresión vista antes (número de la expresión)

$$Distancia_{Coseno}(D6, D1) = \frac{< [-1.13407621, -0.4401046], [-0.61714, 1.221853] >}{1.21647890 \cdot 1.36886424} = 0.09737053$$

$$Distancia_{Coseno}(D6, D2) = \frac{< [-1.13407621, -0.4401046], [-0.20391781, 0.63846665] >}{1.2164789 \cdot 0.67024036} = -0.0609981$$

$$Distancia_{Coseno}(D6, D3) = \frac{< [-1.13407621, -0.4401046], [-1.04666681, 0.477979966] >}{1.2164789 \cdot 1.15064168} = 0.697733$$

$$Distancia_{Coseno}(D6, D4) = \frac{< [-1.13407621, -0.4401046], [-0.49690556, -0.410249997] >}{1.2164789 \cdot 0.64437582} = 0.9492417$$

$$Distancia_{Coseno}(D6, D5) = \frac{< [-1.13407621, -0.4401046], [-1.5038439, -0.785108426] >}{1.2164789 \cdot 1.6964497} = 0.9938499$$

Claramente, como ya se veía en la representación geométrica por el ángulo, el documento número 5 es el más próximo al valor 1 y por tanto el más cercano. El siguiente sería el documento 4, y el que más lejos se encuentra es el Documento 2.

2.3. Clasificación del Sentimiento en Textos

Hemos visto dos métodos muy potentes que pueden ser utilizados para la clasificación de Textos mediante ejemplos sencillos. A menudo se han estudiado y desarrollado estos métodos en proyectos relacionados con el filtrado de spam [1] , la categorización de textos [6] o para la clasificación de sentimiento [2] .Éste último es el que abarcaremos en este apartado pasando por: qué sentimientos identificar y qué clasificadores usar.

2.3.1. Definición del problema

Como se ha dicho anteriormente, el objetivo de este proyecto es el de poder realizar una clasificación automática del sentimiento que expresa cada documento o pequeño texto. Para ello, a la hora de determinar qué expresa cada uno debemos definir cuáles son las posibilidades.

Encontramos varias opciones posibles a la hora de realizar la clasificación de sentimiento. En estudios anteriores como [2] se definen clasificaciones binarias, diferenciando entre textos positivos y negativos. Otras opciones, que están muy relacionadas con las librerías desarrolladas en lenguajes como Python, son el uso de una clasificación neutral. Además, tenemos una última combinación si conseguimos definir un extremo para cada uno de los sentimientos básicos. En resumen, podemos definir las siguientes combinaciones de 5, 3 o 2 sentimientos:

Sentimiento	*Significado*
Muy positivo	Es claramente positivo
Positivo	Puede haber incertidumbre en algunos casos
Neutral	No se puede inferir distinción de sentimiento
Negativo	Puede haber incertidumbre en algunos casos
Muy Negativo	Es claramente negativo

Tabla i Clasificación de sentimiento con 5 opciones

Sentimiento	*Significado*
Positivo	Es claramente positivo
Neutral	Hay incertidumbre sobre el sentimiento que expresa
Negativo	Es claramente negativo

Tabla ii Clasificación de sentimiento con 3 opciones

Sentimiento	*Significado*
Positivo	Ante la duda se elige positivo
Negativo	Ante la duda se elige negativo

Tabla iii Clasificación de sentimiento con 2 opciones

Podemos apreciar que los algoritmos y métodos estudiados pueden ser independientes de la cantidad de opciones. Por ejemplo, si utilizamos el primer tipo de la Tabla 1 pero clasificamos los datos de entrenamiento de forma binaria y el comportamiento de nuestro clasificador es el de igualar el sentimiento al del más cercano el conjunto de los diferentes valores devueltos será $\{Positivo, Negativo\}$.

El problema de selección sobre qué tipo es más conveniente utilizar puede estar relacionado con el uso de la escala Likert en los cuestionarios de opinión, facilitando la elección de quien se encarga de clasificar el conjunto de entrenamiento. Pero deberemos corroborar los distintos ajustes y cómo afectan a la clasificación.

2.3.2. Ejemplo básico de clasificación

Si por ejemplo tenemos los siguientes enunciados

1. $-Esta\ película\ es\ mi\ favorita$
2. $-Este\ paraguas\ es\ bonito$
3. $-La\ comida\ me\ sabe\ mal$
4. $-Ha\ llegado\ tu\ hermano$
5. $-Tu\ perro\ es\ feo$

Podemos hacer las siguientes clasificaciones:

	5 opciones	**3 opciones**	**Binaria**
Documento 1	Muy positivo	Positivo	Positivo
Documento 2	Positivo	Positivo	Positivo
Documento 3	Negativo	Negativo	Negativo
Documento 4	Muy negativo	Negativo	Negativo
Documento 5	Neutral	Neutral	Positivo
Documento 6	Negativo	Negativo	Negativo

Tabla iv Clasificación de sentencias

2.3.3. Clasificador Naive-Bayes

Si aplicamos una clasificación mediante un Naive-Bayes en la primera opción obtenemos las siguientes estimaciones de probabilidades:

$$P'(Sentimiento = Muy\ Negativo) = \frac{1}{6}$$

$$P'(Sentimiento = \text{Negativo}) = \frac{1}{6}$$

$$P'(Sentimiento = Positivo) = \frac{1}{6}$$

$$P'(Sentimiento = Muy\ Positivo) = \frac{1}{6}$$

$$P'(Sentimiento = Neutral) = \frac{1}{6}$$

Término	P(Término|Muy Negativo)	P(Término| Negativo)	P(Término|Muy Positivo)	P(Término|Positivo)	P(Término|Neutral)
Favorito	0/1	0/1	1/1	0/1	0/1
Bonito	0/1	0/1	0/1	1/1	0/1
Mal	0/1	1/1	0/1	0/1	0/1
Feo	0/1	1/1	0/1	0/1	0/1
Llegado	0/1	0/1	0/1	0/1	1/1

Tabla v Estimaciones de probabilidad del ejemplo

Ahora podemos tomar un nuevo documento que queremos clasificar:

6. – Es tu favorita pero es fea y queda mal

Y aplicamos la expresión (17) para la obtención del valor

$$v_{nb} = argmax_{v_j \in V}\ P(v_j) \prod_i P(a_i|v_j), \quad (22)$$

que para este caso concreto tenemos

$$argmax_{v_j \in V}\ P(v_j)\ P'(\ Feo \mid v_j)\ P'(Favorito|v_j).$$

Para los valores de

$$v_j = Muy\ Negativo, Negativo, Muy\ Positivo, Positivo\ y\ Neutral$$

$$P'(Sentimiento = Muy\ Negativo)\ P'(Feo \mid Muy\ Negativo)\ P'(Favorito \mid Muy\ Negativo) = \frac{1}{6} * 0 * 0 = 0$$

$$P'(Sentimiento = Negativo)\ P'(Feo \mid Negativo)\ P'(Favorito \mid Negativo) = \frac{1}{6} * 1 * 0 = 0$$

$$P'(Sentimiento = Muy\ Positivo)\ P'(Feo \mid Muy\ Positivo)\ P'(Favorito \mid Muy\ Positivo) = \frac{1}{6} * 0 * 1 = 0$$

$$P'(Sentimiento = Positivo)\ P'(Feo \mid Positivo)\ P'(Favorito \mid Positivo) = \frac{1}{6} * 0 * 0 = 0$$

$$P'(Sentimiento = Neutral)\ P'(Feo \mid Neutral)\ P'(Favorito \mid Neutral) = \frac{1}{6} * 0 * 0$$

Como se ha dicho anteriormente, las estimaciones que nos devuelven $P(a_i|v_j) = 0$ dominarán el clasificador obligando al resultado a ser 0. Para resolver este problema, la solución más común es el uso de la $estimación - m$ que viene definida por

$$P(w_k|v_j) = \frac{(n_i + 1)}{n + |vocabulario|} \quad . \ (34)$$

Las nuevas estimaciones de probabilidades resultarán de la siguiente forma

Término	P(Término|Muy Negativo)	P(Término | Negativo)	P(Término|Muy Positivo)	P(Término|Positivo)	P(Término|Neutral)
Favorito	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(2 + 6) = 1/8	(1 + 1)/(1 + 6) = 2/7	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7
Bonito	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(2 + 6) = 1/8	(0 + 1)/(1 + 6) = 1/7	(1 + 1)/(1 + 6) = 2/7	(0 + 1)/(1 + 6) = 1/7
Mal	(0 + 1)/(1 + 6) = 1/7	(1 + 1)/(2 + 6) = 2/8	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7
Feo	(0 + 1)/(1 + 6) = 1/7	(1 + 1)/(2 + 6) = 2/8	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7
Llegado	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(2 + 6) =	(0 + 1)/(1 + 6) = 1/7	(0 + 1)/(1 + 6) = 1/7	(1 + 1)/(1 + 6) = 2/7

		1/8			

Tabla vi Estimaciones de probabilidad del ejemplo

$$P'(Sentimiento = Muy\ Negativo)\ P'(Feo \mid Muy\ Negativo)\ P'(Mal|Muy\ Negativo)P'(Favorito|\ Muy\ Negativo)$$
$$= \frac{1}{6} * \frac{1}{7} * \frac{1}{7} * \frac{1}{7} = 0.0004859$$

$$P'(Sentimiento = Negativo)\ P'(Feo\ |Negativo)\ P'(Mal|\ Negativo\ )P'(Favorito|Negativo)$$
$$= \frac{1}{6} * \frac{1}{4} * \frac{1}{7} * \frac{1}{4} = 0.0014881$$

$$P'(Sentimiento = Muy\ Positivo)\ P'(Feo \mid Muy\ Positivo)\ P'(Mal|Muy\ Positivo)P'(Favorito|\ Muy\ Positivo)$$
$$= \frac{1}{6} * \frac{1}{7} * \frac{2}{7} * \frac{1}{7} = 0.0004859$$

$$P'(Sentimiento = Positivo)\ P'(Feo\ |Positivo)P'(Mal|Positivo)\ P'(Favorito|Positivo)$$
$$= \frac{1}{6} * \frac{1}{7} * \frac{1}{7} * \frac{1}{7} = 0.0004859$$

$$P'(Sentimiento = Neutral)\ P'(Feo\ |Neutral)\ P'(Mal|Neutral)P'(Favorito|Neutral)$$
$$= \frac{1}{6} * \frac{1}{7} * \frac{1}{7} * \frac{1}{7} = 0.0004859$$

Vemos que el máximo se encuentra cuando $v_j = Negativo$ y éste es el valor que devolverá el clasificador creado con este entrenamiento.

2.3.4. Clasificación con LSA

La matriz A tiene la siguiente forma

$$\begin{matrix} Favorito \\ Bonito \\ Mal \\ Feo \\ Llegado \end{matrix} \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

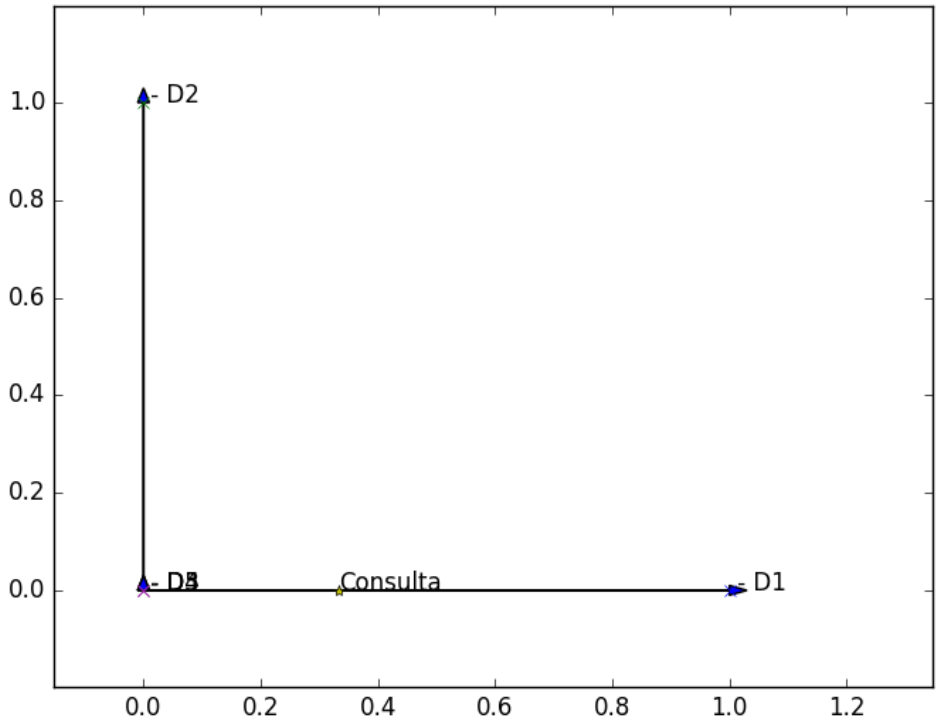


Ilustración VII Representación de los documentos y la consulta

En este caso deberemos establecer un umbral que determine qué documentos influyen en el valor que recibirá el nuevo documento. Para ello lo más común es, como se ha dicho antes, coger las distancias coseno para la similitud y tener en cuenta los que obtengan mayor que: $Media + \alpha * Desviación\ Típica$, dependiendo del contexto podemos mejorar el valor de la variable α [17] . En este ejemplo podemos coger $\alpha = 0.75$. Por tanto, $Umbral = 0.53$.

Documento 1

$$\frac{< [0.3333333333333331, 0.0], [1.0, 0.0] >}{0.3333333333333333 * 1.0} = 1.0$$

Documento 2

$$\frac{< [0.33333333333333331, 0.0], [0.0, 1.0] >}{0.3333333333333333 * 1.0} = 0$$

Documento 3

$$\frac{< [0.33333333333333331, 0.0], [0.0, 1.0] >}{0.3333333333333333 * 1.0} = 0$$

Documento 4

$$\frac{< [0.33333333333333331, 0.0], [0.0, 1.0] >}{0.3333333333333333 * 1.0} = 0$$

Documento 5

$$\frac{< [0.33333333333333331, 0.0], [0.0, 1.0] >}{0.3333333333333333 * 1.0} = 0$$

Por tanto, es el documento 1 el que influirá en el sentimiento del nuevo documento al que se le adjudicará el valor $Muy\ Positivo$.

Muy posiblemente la clasificación como *Negativo* realizada por el clasificador Bayesiano sea más certera con respecto al juicio humano en este caso. En cuanto a la clasificación LSA se observa, al estar formada la matriz de frecuencias por un único 1 en cada una de las filas y columnas, se produce una escasez de datos de entrenamiento que permitan una mejor clasificación.

2.4. Mejoras y conclusiones en la clasificación de textos

Ya hemos visto el funcionamiento básico de algunos clasificadores que podemos utilizar para el análisis de sentimiento en documentos. Podemos investigar o pensar acerca de mejoras en el tratamiento previo de los datos o en estudios de relación de documentos.

2.4.1. Stemming

En el capítulo relativo a los clasificadores Bayesianos se sugirieron varias formas de mejorar los mecanismos de clasificación de textos. Uno de ellos es la sustitución de palabras por una más representativa, aumentando así las coincidencias y posibles relaciones entre documentos. Esto es interesante en nuestro idioma, castellano, si pensamos en las palabras que no coincidirán por conjugación verbal o modificación de adjetivos. Por ello deberemos centrarnos en encontrar una raíz que sea la sustitución a cada una de las posibles palabras que no modifican el significado semántico por haber sido transformadas.

El Stemming [18] se puede definir entonces como el método que nos permite reducir a una raíz una palabra determinada. Su nombre procede del inglés, donde stem es raíz. Por ejemplo

En inglés es muy común el uso del algoritmo de Porter [8] y ha sido empíricamente demostrado como muy efectivo en este idioma , por el mismo Porter en 1980. Consiste en 5 fases de reducciones sobre una palabra aplicadas secuencialmente. En cada fase se utilizan una gran variedad de convenciones para seleccionar las reglas, como la selección de la regla de cada grupo de reglas que se aplica al sufijo más largo (en el algoritmo de Porter únicamente se eliminan sufijos, no prefijos). En el primer paso se utilizan reglas como

$$\begin{array}{lcl} Regla & & \\ SSES & \rightarrow & SS \\ IES & \rightarrow & I \\ SS & \rightarrow & SS \\ S & \rightarrow & \end{array}$$

Por ejemplo, podemos aplicarlo a algunas derivaciones de palabras

$$\begin{array}{lcl} Regla & & \\ caresses & \rightarrow & caress \\ ponies & \rightarrow & poni \\ caress & \rightarrow & caress \\ cat & \rightarrow & cat \end{array}$$

Muchas de las siguientes reglas utilizan el concepto de medida de una palabra, que vagamente se encarga de revisar el número de sílabas para comprobar si una palabra es lo suficientemente larga para que sea razonable buscar la parte que corresponde a una regla como sufijo antes que como parte de la raíz de la palabra. Por ejemplo, si tenemos una palabra donde el sufijo que buscamos es la mayor parte de ésta, quizá pertenezca a la raíz como en

$$(m > 1)\ EMENT \rightarrow ,$$

$$donde\ m\ es\ la\ longitud\ de\ la\ palabra\ en\ sílabas$$

Ésta regla reemplazaría en $replacement$ para convertirla en $replac$. Pero no en el caso de que tengamos una palabra como $cement$ que pase a c, esto no tendría sentido.

También existen otros algoritmos muy estudiados como el stemmer de Lovins (1968) o el stemmer de Paice/Husk también llamado de Lancaster (1990). Se ha desarrollado también una mejora del algoritmo de Porter denominado Porter2 [19] impulsado por su posible utilidad en idiomas distintos al inglés. Todo ello se encuentra recogido en el proyecto Snowball, que obtiene su evolución desde una comunidad de software libre que realiza su implementación en gran cantidad de lenguajes de programación e intenta mejorarlo. Gracias a ésto hay gran facilidad de comparación de la multitud de stemmers disponibles online. Un ejemplo de ello es la demo de Text-Processing.com ligada a los paquetes software de Herramientas de Procesamiento del Lenguaje Natural (NLTK en inglés) [20]. Por ejemplo podemos comparar

> ***Texto simple***: El debate de investidura de Mariano Rajoy se ha reanudado este miércoles con la réplica de los partidos. El líder del PSOE, Pedro Sánchez, ha argumentado su no al candidato del PP que, con su respuesta, ha anticipado la imposibilidad de una investidura próxima.
>
> ***Stemmer de español***: el debat de investidur de marian rajoy se ha reanud este miercol con la replic de los part . el lid del pso , pedr sanchez , ha argument su no al candidat del pp que , con su respuest , ha anticip la imposibil de una investidur proxim .
>
> ***Stemmer de Porter***: El debat de investidura de Mariano Rajoy se ha reanudado est miércol con la réplica de lo partido . El líder del PSOE , Pedro Sánchez , ha argumentado su no al candidato del PP que , con su respuesta , ha anticipado la imposibilidad de una investidura próxima .
>
> ***Stemmer de Lancaster***: el deb de investidur de mariano rajoy se ha reanudado est miércol con la réplica de los partido . el líder del pso , pedro sánchez , ha argumentado su no al candidato del pp que , con su respuest , ha anticipado la imposibilidad de un investidur próxima .

2.4.2. Clustering en documentos

Los algoritmos de clustering [8] permite encontrar las agrupaciones de un conjunto de documentos en subconjuntos o clústeres. El objetivo de estos algoritmos es el de encontrar clúster que mantienen una coherencia interna, pero que claramente se diferencian unos de otros. Esto es, los documentos que se encuentran en un clúster deberían ser tan similares como sea posible; y los documentos de un clúster deberían ser lo menos similares como sea posible a otros documentos de clústeres diferentes.

El clustering es uno de los más comunes métodos de aprendizaje no supervisado, esto significa que no habrá un juicio humano que asigne los documentos a sus determinadas clases. Para realizar esto debemos asumir la hipótesis del clúster: los documentos en el mismo grupo se comportan de manera similar respecto a la información relevante requerida.

Una de las dificultades es la determinación del número de grupos o la denominada cardinalidad de un clustering, la cual se denota como K. A menudo ésta no es más que una intuición basada en la experiencia y el conocimiento sobre el contexto. Pero para algunos algoritmos podemos encontrar métodos heurísticos para su elección. Uno de estos algoritmos más conocidos es el denominado $K - means$.

Teorema 2.8. Algoritmo K-means .- Dado $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de n puntos en el plano podemos agruparlos en k grupos ,de la siguiente forma [15]

- Paso 1. Definimos tantos centroides[1] aleatorios como $k - clústeres$ hayamos indicado.
- Paso 2. Asignamos a cada centroide el punto más cercano.
- Paso 3. Calculamos el punto medio de cada centroide.
- Paso 4. Si es el primer ciclo o los centroides no han cambiado, sustituimos en la lista de centroides por los puntos medios correspondientes y volvemos al Paso2.
- Paso 5. Aquí los centroides no se han modificado y por tanto podemos devolver los puntos asignados. El conjunto de cada centroide se corresponde con un clúster distinto.

Aplicando estas técnicas de clustering se puede estudiar la asociación de documentos representados de forma vectorial definiendo un número K de grupos en los que se podrían dividir y ver cómo se organizan entorno a los textos..

[1] Centroide será el punto que devolveremos que se corresponderá con el centro (punto medio) de cada clúster.

3. Grafos y Redes

En este apartado analizaremos los métodos y fundamentos del estudio de grafos y redes. Estas estructuras son usadas para modelar las relaciones entre objetos. Lo que nos ayudará en nuestro proyecto con respecto a la organización de la parte de la red social que queramos estudiar.

Ilustración VIII Leonhard Euler

El trabajo de Leonhard Euler, en 1736, sobre el problema de los puentes de Königsberg es considerado como uno de los primeros resultados de la teoría de grafos [21] Posteriormente, ha habido una gran cantidad de estudios relacionados con ésta, además de aplicaciones en diversos campos. Por ejemplo en 1845, cuando Gustav Kirchoff aplica esta intuición para la publicación de sus leyes sobre circuitos para el cálculo de corriente y voltaje, el planteamiento del llamado problema de los cuatro colores planteado en 1852 [22], resuelto un siglo después, a partir del siglo XX en el campo de la psicología se comienzan a realizar estudios donde se intentan relacionar los grafos con la sociología y la interacción entre individuos, y recientemente encontramos estudios célebres y reconocidos como sobre el matrimonio y la estructura de la élite en el Renacimiento de Florencia donde se estudian las relaciones de familias entre finales del siglo XIII y principios del XVI (finales de siglo XX). A partir de mediados del siglo XX comenzaron a nacer los estudios relacionados con la creación y análisis de redes aleatorias, mediante las investigaciones sobre modelos muy conocidos como el de Erdos-Rényi [23]. A finales del milenio Albert-László Barabasi introduce el concepto de redes de libre escala y propone el modelo Barabási-Albert como explicación a la extensión en los sistemas naturales, tecnológicos y sociales.

3.1. Aplicaciones de las redes

Dada la abstracción que se realiza entre la selección de objetos cualesquiera y tratarlos como nodos que se unen a otros se han empleado las técnicas de redes en numerosos ámbitos de estudio. Podemos encontrar ejemplos en [3]

- Redes económicas y sociales. Se encargan de estudiar los contactos e interacciones que surgen entre grupos de personas para la búsqueda de patrones. Ejemplos de ello son las redes de amistad o las redes de relaciones de negocio entre compañías. Como éstas son las que más nos interesan y a las que se quiere aproximar con este proyecto deberemos prestar atención a cuáles son algunos de los puntos de interés de estudio, principalmente: la conectividad de la red y su grado, la cercanía o el rol de los nodos como intermediarios.
- Redes de información. Donde se representan los enlaces entre cada documento de información, por ejemplo, los datos disponibles en Internet donde se han realizado estudios sobre el Page Ranking o la navegación.

- Redes tecnológicas. Principalmente utilizado para el estudio de las estructuras de redes de comunicación como routers, donde se requiere el estudio de la seguridad, eficacia o eficiencia.
- También podemos representar mediante redes elementos biológicos como ciclos metabólicos o redes de interacciones de proteínas.

Ha surgido un gran cambio en el estudio de redes que se produjo desde el estudio de pequeños grafos hasta la investigación sobre propiedades estadísticas de grandes redes con millones de nodos. La computación y su evolución nuevamente han propiciado en este campo, así como en otros, estas posibilidades de trabajo con grandes volúmenes de información. [24]

3.2. Definiciones y Notación

Se introducen a continuación los elementos básicos para el trabajo sobre grafos y redes [22].

Definición 3.1.- Definimos un grafo G no dirigido como una dupla $G = (V, E)$ donde:

· $V = \{v_1, v_2, \dots\}$ es un conjunto de vértices, puntos en el espacio.

· $E = \{ (vi, vj), (vk, vl), \dots \}$ es un conjunto de aristas es decir pares de elementos de V donde cada uno representa una relación o conexión.

A veces, podemos encontrar la definición de un grafo como una terna ordenada $G = (V, E, \phi_G)$, donde

- V es un conjunto no vacío de vértices,
- E es un conjunto de aristas,
- ψ_G una función de incidencia. En ésta se describe cada una de las aristas de la forma $\psi_G(e_1) = v_1 v_3$.

Definición 3.2.- Un grafo se denomina finito si los vértices y las aristas son ambos finitos.

Definición 3.3.- Un camino se define como una secuencia de vértices $\{ i1, i2, \dots, in \}$ y de relaciones $\{ (i1, i2), (i2, i3), \dots, (in-1, in) \}$ tal que $(i_{j-1}, i_j) \epsilon E$.

A partir de esto podemos definir los siguientes conceptos:

Definición 3.3.- Camino simple. Caso particular en el cual en un camino no se visita dos veces un vértice.

Definición 3.4.- Camino más corto o geodésico. Camino de mínima longitud entre dos vértices.

Definición 3.5.- Distancia entre dos vértices $u\ y\ v$ es la longitud del camino más corto entre ellos dos.

Definición 3.6.- Ciclo o lazo es un camino en el que se cumple $i_j = i_k$. Un grafo se denomina simple si no tiene lazos o ciclos y no hay dos relaciones, aristas entre distintos vértices, que unan el mismo par de vértices.

Definición 3.7.- El diámetro de la red indica el geodésico más largo, o si no es conexa, el del mayor componente. Esto es $\max\left\{longitud\left(camino(v_i, v_j)\right) \forall\, v_i, v_j \in V\right\}$.

Definición 3.8.- Longitud media de caminos. media entre las longitudes de los caminos más cortos que unen todos los pares de nodos.

Definición 3.9.- Camino euleriano es un camino que pasa por una relación una sola vez.

Definición 3.10.- Camino hamiltoniano es un camino que pasa por cada nodo una sola vez.

Definición 3.11.- De una red podemos decir que está conectada o es conexa si existe camino entre dos vértices cualesquiera pertenecientes a V.

Definición3.12.- Denominamos componente a un subgrafo maximal y conexo.

Definición 3.13.- Podemos definir un grafo dirigido si existe un orden en la aristas, ésto es, (i_1 , i_2) indica que i_1 está conectado a i_2 y sólo se produce en esa orientación. En este caso si es conexo lo denominaremos fuertemente conexo.

Definición 3.14.- El conjunto vecindad de un nodo está constituido por aquellos a los que podemos acceder desde éste.

Definición 3.15.- Dos grafos G y H son isomorfos si existen las biyecciones entre sus vértices y sus aristas. Podemos decir de ellos que tienen la misma estructura. Esto es

- $\theta: V(G) \to V(H)$
- $\phi: E(G) \to E(H)$

Tal que $\psi_G = uv$ si y sólo si $\psi_H\big(\phi(e)\big) = \theta(u)\theta(v)$.

Definición 3.16.- Decimos que un grafo es un grafo vacío si no tiene aristas.

Definición 3.17.- Un grafo es bipartito si podemos separar el conjunto de vértices en dos conjuntos A y B que cumplan que para cada arista tiene extremo en un vértice de A y otro en un vértice de B.

Definición 3.18.- Sea G un grafo y $G_1\ y\ G_2$ subgrafos de éste. Decimos que $G_1\ y\ G_2$ son disjuntos si no tienen un vértice en común. Y son disjuntos en cuanto a las aristas si no tienen ninguna en común.

Dos aristas en la representación de un grafo pueden intersecarse en un punto que no sea un vértice.

Definición 3.19.- Aquellos grafos que admiten un diagrama donde las aristas no se corten se denomina planar. Y ésta una representación planar.

Definición 3.20.- El grado de un nodo viene determinado por el cardinal de la vecindad, número de aristas incidentes en un vértice. Podemos encontrar dos tipos de grados si hablamos de grafos dirigidos:

- Grado de entrada del nodo i donde se hace el recuento de los nodos conectados hasta éste $\sum_j g_{ji}$.
- Grado de salida del nodo i , en este caso partimos de éste y se dirige a otros. $\sum_j g_{ij}$.

Definición 3.21.- Un grafo se llama $k-regular$ si el grado de todos los vértices es el mismo. Esto es $grado(v) = k \; \forall \; v \in V$. También podemos definir los grafos bipartitos regulares $K_{n,n}$.

En vez de tratar simplemente las aristas como una relación entre dos vértices, podemos asignarles un peso, a veces denominado coste o longitud en los campos de aplicación.

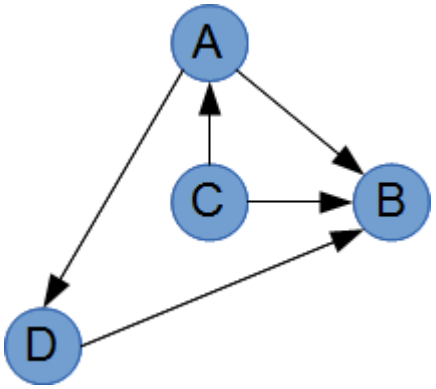
Definición 3.22.- El peso de un subgrafo se corresponde con la suma de los pesos de sus aristas.

3.3. Representación y estructuras

Para poder manejar un grafo o una red podemos utilizar varias estructuras:

Definición 3.23.- Matriz de adyacencia. Representa la existencia de relación entre dos nodos de la red , i y j , de manera que M_{ij} es 1 si existe y 0 si no. Podemos reemplazar 1 por el peso correspondiente en caso de utilizarlo.

Ejemplo 3.1.- Tenemos el siguiente grafo cuya matriz de adyacencia se representa a su lado



$$Matriz = \begin{pmatrix} & A & B & C & D \\ A & 0 & 1 & 0 & 1 \\ B & 0 & 0 & 0 & 0 \\ C & 1 & 1 & 0 & 0 \\ D & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Esta representación es muy útil puesto que si definimos el k-ésimo producto de la matriz consigo misma podemos utilizar $Matriz^k(i,j)$ para obtener el número de caminos desde i hasta j que contienen k aristas.

Definición 3.24.-Matriz de incidencia. Representa la incidencia de una arista sobre un nodo. Siendo cada una de las filas uno de los nodos ,n, y cada columna una de las aristas ,a, en la posición M_{na} guardaremos 1 si la arista a incide en el nodo n y 0 en caso contrario. En todo grafo el número de 1s de una columna será 2, sin embargo, a la hora de representar hipergrafos (más de dos vértices por arista) podremos encontrar más.

Ejemplo 3.2.- Sea el siguiente grafo

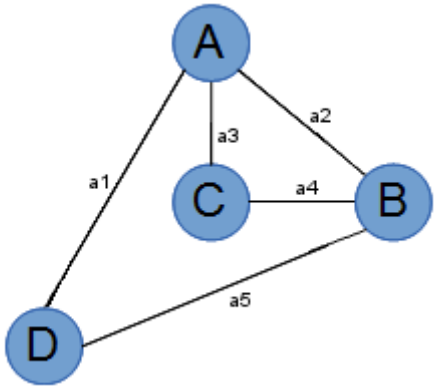


Ilustración IX Grafo de ejemplo

Su matriz de incidencia se representa a continuación

$$Matriz = \begin{pmatrix} & a1 & a2 & a3 & a4 & a5 \\ A & 1 & 1 & 1 & 0 & 0 \\ B & 0 & 1 & 0 & 1 & 1 \\ C & 0 & 0 & 1 & 1 & 0 \\ D & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Podemos ver que este tipo de representación es el más adecuado cuando tenemos muchos nodos, pero pocas relaciones, ahorrándonos muchas columnas.

Definición 3.25.- Listas de adyacencia. Mediante una n-tupla de tuplas que representan los nodos a los que se puede conectar un nodo.

Ejemplo 3.3.- Sea el siguiente grafo

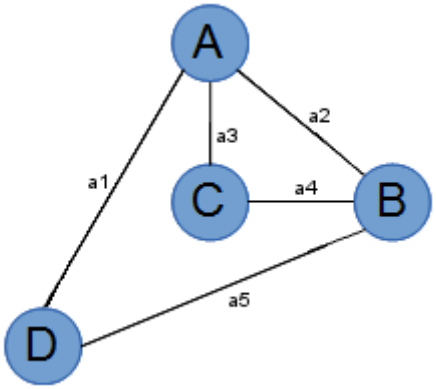


Ilustración X Grafo de ejemplo

, cuya representación como lista de aristas es la siguiente

$$Listado = \{\{B, C, D\}, \{A, C, D\}, \{A, B\}, \{A, B\}\}.$$

Definición 3.26.- Lista de incidencia. Se describen las relaciones entre nodos como una tupla de pares. De ese listado podemos obtener los nodos que componen la red.

Ejemplo 3.4.- Sea el siguiente grafo

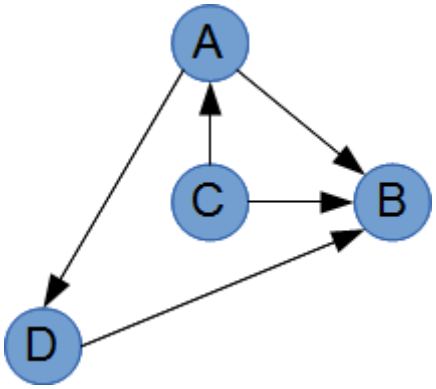


Ilustración XI Grafo de ejemplo

donde su representación

Listado = {{A,B},{A,D},{C,A},{C,B},{D,B}}.

3.4. Algoritmos sobre Grafos

Para estudiar los métodos y algoritmos aplicables en la teoría de grafos deberemos tener en cuenta una medida que nos indique cómo de difícil es computarlos. Utilizaremos la denominada Complejidad.

Definición 3.27.- La complejidad puede definirse como el número de pasos que se requieren para transformar los datos de entrada en el resultado del cómputo deseado [22].

Para los algoritmos sobre grafos los tamaños de los problemas vendrán normalmente definidos por el número de nodos o el número de aristas.

La complejidad más utilizada es la denominada del peor caso [25], en la que se tienen en cuenta la peor situación posible, es decir, el máximo número de instrucciones en el cómputo.

Dados dos algoritmos con sus complejidades

$$C_{A_1}(n) = \frac{n^2}{2} \text{ y } C_{A_2}(n) = 5n.$$

Vemos que A_2 crece más rápidamente que A_1 $\forall\, n > 10$. El crecimiento asintótico de los problemas, nos dice que cuando su tamaño tiende a infinito entonces el crecimiento de n^2 es mayor que n. Diremos que la complejidad de A_2 es de un orden menor que la de A_1.

Tenemos que dadas dos funciones F y G cuyo dominio son los números naturales ,el orden de F es menor o igual al orden de G si $F(n) \leq k \cdot G(n)$ para $n > n_0$, con k y n_0 son constantes positivas. Descrito con la notación correspondiente $F = O(G)$. Podemos describir la comparación de orden entre dos funciones de complejidad mediante [25]

$$\lim_{n\to\infty} \frac{F(n)}{G(n)} = L \quad , \qquad (35)$$

donde

$$\begin{cases} si\ L = a\ con\ a > 0\ entonces\ F = O(G), \\ si\ L = 0\ entonces\ F\ es\ de\ menor\ orden\ que\ G, \\ si\ L = \infty\ entonces\ G\ es\ de\ menor\ orden\ que\ G. \end{cases}$$

De esta forma podemos ordenar algunas complejidades de referencia [25]

$$O(1) \subset \mathrm{O(log(n))} \subset \mathrm{O}(n) \subset \mathrm{O(n\log}\, n) \subset \mathrm{O(n^2)} \subset \mathrm{O(n^3)} \subset \cdots$$

$$\ldots \subset \mathrm{O(2^n)} \subset \mathrm{O(3^n)} \subset \mathrm{O(n!)} \subset \mathrm{O(n^n)}\,.$$

Algunos de los algoritmos más útiles para el estudio de las redes son los siguientes

- Dijkstra [26] o algoritmo de caminos mínimos nos permite obtener el camino más corto desde un nodo a los demás de la red. Su primera descripción se realizó en 1959 por Edsger Dijkstra.
  Teorema 3.1.- El algoritmo de Dijkstra realiza $O(n^2)$ operaciones para determinar el camino mínimo entre dos nodos de un grafo sin pesos, conexo y no dirigido con n nodos.
  Demostración.- Para la demostración detallada revisar la referencia bibliográfica [25]
- BFS [25] ,de sus siglas en inglés Búsqueda en Anchura, es un procedimiento de tipo FIFO con el que se pretende recorrer todos los nodos de la red.
- DFS [25] ,de sus siglas en inglés Búsqueda en Profundidad, es un procedimiento de tipo LIFO
  Éstos algoritmos, BFS y DFS, nos permiten ver con facilidad si la red es conexa, puesto que de no serlo no se podrán visitar los grupos aislados de nodos.

3.5. Características de estudio

Vistos los conceptos más básicos sobre los grafos y redes pasemos a definir cuáles queremos que sean los objetos de estudio para las redes sociales en particular, las cuales son el fin principal de este proyecto

3.5.1. Distribución de grados

Definición 3.28.- La distribución de grado, $P(grado)$, de una red es una descripción de la frecuencia de nodos que tienen diferentes grados. En las redes normales ,obtenidas mediante información real, esta representación es un histograma de frecuencias.

La distribución de las redes sigue la Ley de Potencia [3] Se encuentra en ellas una acumulación en la cantidad de nodos que tienen grados bajos. Ésto suele suceder cuando una red tiene un gran número de nodos pero una baja probabilidad de creación de relaciones.

Definición 3.29.- Una variable aleatoria (continua) sigue una Ley de Potencia si su densidad de probabilidad es [27]

$$f(x) = Cx^{-a} \ . \ (36)$$

Existe un problema cuando observamos la distribución y ésta se encuentra en $x \to 0$. Éste valor hace que la fracción se vaya al infinito. Para resolverlo deberemos asumir un valor mínimo $x_{\min}$ que suele ser 1 [27]. En ese caso

$$C = (a-1)\, x_{\min}^{a-1} \ , \ (37)$$

resultando mediante la sustitución en la expresión (36)

$$f(x) = \frac{a-1}{x_{\min}} \left(\frac{x}{x_{\min}}\right)^{-a} \ . \ (38)$$

Ejemplo 3.5.- Vilfredo Pareto realizó estudios sobre la distribución del ingreso. En particular, la probabilidad de que alguien gane x más se expresa en términos de la distribución acumulada

$$P(X > x) \approx x^{-k} \ , \qquad (39)$$

en Ley de Potencia $k = a - 1$ siendo a el exponente. [22]

Ejemplo 3.6.- Georges Zipf investigó acerca de la frecuencia de los términos más comunes en inglés. De esto descubrió que la frecuencia era inversamente proporcional al ranking de forma

$$f(r) \approx r^{-\beta} \ , \qquad (40)$$

con el exponente cercano a 1. [22]

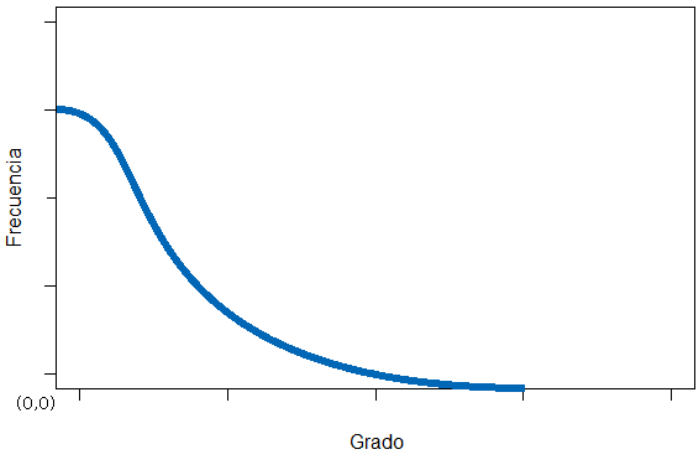


Ilustración XII Distribución de grados con baja probabilidad de crear relaciones

Las principales características de las redes que se forman y están relacionadas con la Ley de Potencia tienen las siguientes características [3]

- Muchos nodos aislados debido a la improbable aparición de altos grados de éstos.
- Bastantes componentes que forman la red.
- La cola de la distribución decae más rápido que una exponencial, esto es porque podemos aproximar

$$P(grado) \leq ce^{\alpha grado} \ , \qquad (41)$$

con $\alpha > 0$ y $c > 0$

3.5.2. Clustering

Como ya hemos descrito en la sección de mejoras del análisis de sentimiento, el objetivo del análisis de clustering es el de encontrar agrupamientos de tal forma que los componentes de un grupo tengan cierta similitud que los diferencie de los demás. Para facilitar el reconocimiento de clústeres deberemos minimizar las distancias internas y maximizar las externas [28].

Definición 3.30. - El coeficiente de clustering, de agrupamiento o transitividad [23] de una red mide cómo de relacionado se encuentra un nodo a la red. Si estamos tratando un grafo completo el valor es máximo, y es pequeño cuando el agrupamiento es pobre. Viene determinado por

$$Clustering(G) = \frac{3 * numero\ de\ triángulos\ en\ la\ red}{número\ de\ ternas\ de\ nodos} . \quad (42)$$

Su valor se encuentra en el intervalo $0 \leq Clustering(G) \leq 1$ y está claramente relacionado por cuántos triángulos completos encontramos en la red. Podemos también diferenciar el clustering entre cada uno de los nodos.

Definición 3.31.- El coeficiente de clustering de un nodo i [3] se obtiene como la representación la fracción de los triángulos en los que se encuentra entre el número de ternas centradas en éste

$$Clustering_{Nodo\ i}(G) = \frac{Número\ de\ triángulos\ conectados\ al\ vértice\ v_i}{número\ de\ ternas\ centradas\ en\ v_i} \ . \quad (43)$$

Y, asociado a éste, podremos calcular un coeficiente de clustering medio para toda la red.

Definición 3.32. - El clustering medio de la red se calcula como el promedio de los coeficientes de clustering de cada uno de los nodos en ésta [3]. Esto es

$$Clustering_{Medio}(G) = \frac{1}{n}\sum_{i}^{n} Clustering_i(G) \ . \quad (44)$$

3.5.3. Medidas de Centralidad

Estas características de estudio nos proporcionarán información concisa acerca de cuál es la posición de un nodo en la red, así como de su importancia dentro de ésta. Hemos definido anteriormente algunos conceptos como el grado de un nodo o la distancia entre dos nodos que jugarán un papel muy importante en estas medidas [3]. Las principales son

3.5.3.1. Centralidad de Grado

Partimos de la asunción de que los nodos con un mayor grado son más centrales.

Definición 3.33.- La centralidad de grado para un nodo i [23], se define como la representación de la cantidad de relacionados con un nodo determinado, viene dada por

$$CentralidadGrado_i = \frac{Grado_i(G)}{n-1} \ , \quad (45)$$

donde n es el número total de nodos en la red. El cociente entre $n-1$ se realiza para la estandarización del grado. El recuento de los nodos que están relacionados con uno dados representa simplemente los que se encuentran a distancia 1, es decir, son vecinos.

También, podemos expresarlo en forma matricial.

Definición 3.34.- $Sea\ A \in \mathbb{R}^{nxn}$ la matriz de adyacencia y sea $k \in \mathbb{R}^n$ el vector de grado. Entonces,

$$k = Ae \ , \qquad (46)$$

donde $e \in \mathbb{R}^n$ es el vector completo con 1s. La posición i-ésima se corresponde con el grado del i-ésimo nodo. [22]

La suposición comentada anteriormente es muy fuerte debido a que podemos representar de alguna forma la importancia en una red social de un individuo o elemento. Esto es, en muchas configuraciones sociales, los individuos con mayor cantidad de conexiones tienen una mayor visibilidad y poder de influencia. [24]

Además, podemos analizar conjuntamente los grados mediante el resumen de su media y la varianza. Nos permite estudiar la distribución de esta medida.

Ejemplo 3.7.- Sea el siguiente grafo

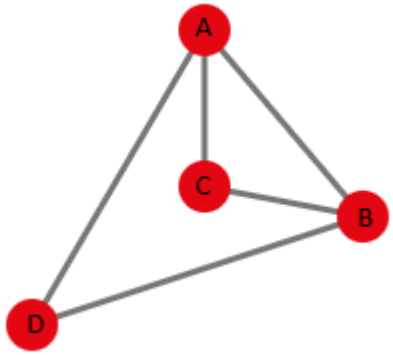


Ilustración XIII Grafo de ejemplo

cuya representación como matriz de adyacencia es el siguiente

$$Matriz = \begin{pmatrix} & A & B & C & D \\ A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 0 \\ D & 1 & 1 & 0 & 0 \end{pmatrix}$$

Si calculamos su vector de grados y centralidad de grados

$$Vector\ de\ grados = (3 \quad 3 \quad 2 \quad 2)$$

$$Centralidad\ de\ grado = (1 \quad 1 \quad 0.66 \quad 0.66)$$

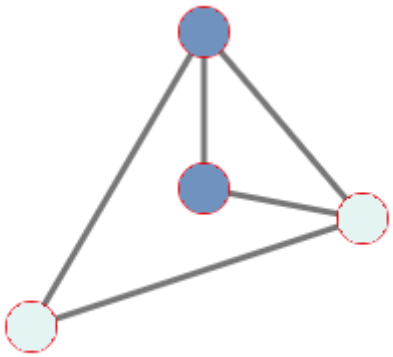


Ilustración XIV Representación de la centralidad de grado

Como podemos observar, tanto en el vector de grados, centralidad de grado y la representación, los nodos A y B concentran mayor centralidad que los restantes.

Sin embargo, el comportamiento de esta medida de centralidad puede no ser la indicada en caso de que tengamos nodos exteriores en la red con considerables relaciones [22]. Por ejemplo

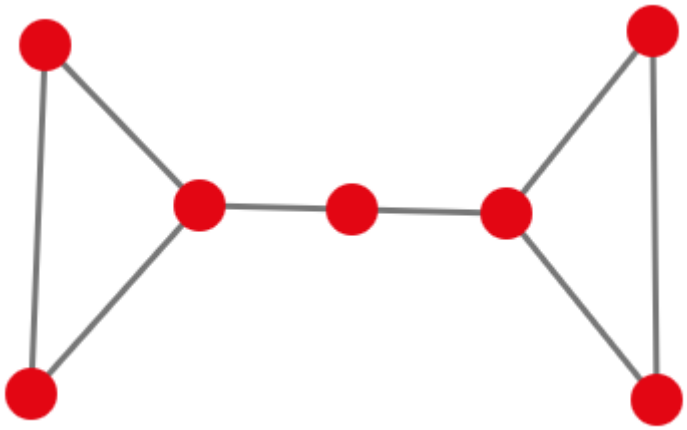


Ilustración XV Ejemplo de grafo donde la centralidad de grado no es la más adecuada

3.5.3.2. Centralidad de Cercanía

Definición 3.35.– La centralidad de cercanía (Closeness) se encarga de indicar cómo de cercano está un nodo dado a cualquier otro. Viene dada por la suma de distancias geodésicas de cada actor con los demás [3]. La suma de éstas es la lejanía del nodo al resto mientras que su inversa es la medida de cercanía

$$Cercanía(v_i) = \frac{1}{\sum_{j \neq i} distancia(i,j)} \ . \quad (47)$$

De la misma forma, para la estandarización podemos dividir entre el máximo valor posible $\frac{1}{n-1}$. Resultando

$$Cercanía(v_i) = \frac{n-1}{\sum_{j \neq i} distancia(i,j)} . \quad (48)$$

Cuanto más central es un nodo, entonces su distancia total al resto de nodos es menor. De esta forma, la cercanía nos permite intuir cuánto tomaría la dispersión de información desde un nodo v_i al resto de forma secuencial [22]

Ejemplo3.8.- Si trabajamos con grafo de la Ilustración XIII Grafo de ejemplo

$$Matriz\ distancias = \begin{pmatrix} & A & B & C & D \\ A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 2 \\ D & 1 & 1 & 2 & 0 \end{pmatrix}$$

$$Cercanía(A) = \frac{3}{3} = 1$$

$$Cercanía(B) = \frac{3}{3} = 1$$

$$Cercanía(C) = \frac{3}{4} = 0.75$$

$$Cercanía(D) = \frac{3}{4} = 0.75$$

Vemos que los nodos A y B son los que mayor grado de cercanía tienen puesto que las distancias al resto de los nodos son todas 1.

3.5.3.3. Centralidad de Intermediación

Definición 3.36.- La centralidad de intermediación (Betweeness) captura la información relacionada con cómo de bien situado se encuentra un nodo en términos de los caminos en los que se encuentra dentro de la red. Fue introducido como una medida para cuantificar el control de la comunicación entre humanos en las redes sociales por Linton Freeman [24]. Su cálculo viene definido como [3]

$$Centralidad\ por\ intermediación(v) = \sum_{s \neq v \neq t \in V} \frac{\rho_{st}(v)}{\rho_{st}} \ , (49)$$

donde ρ_{st} es el número total de caminos mínimos del nodo s y t ,y $\rho_{st}(v)$ es el número de estos caminos en los que se encuentra el nodo v. De nuevo para la estandarización podemos

dividir este resultado entre el número de pares de vértices sin incluir v. Esto es $(n-1)(n-2)$, en grados dirigidos y $(n-1)(n-2)/2$

Ejemplo 3.9.- Dado el siguiente grafo

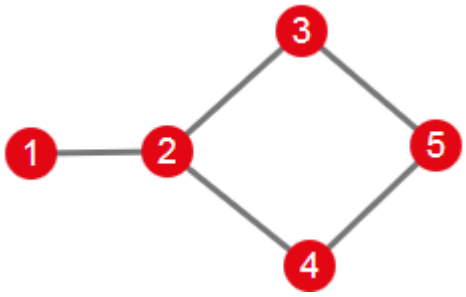


Ilustración 16 Grafo de ejemplo

, procedemos a realizar el cálculo de las centralidades mediante la intermediación. Matriz de número de caminos más cortos entre dos nodos

$$\begin{pmatrix} Nodos & 1 & 2 & 3 & 4 & 5 \\ 1 & - & 1 & 1 & 1 & 2 \\ 2 & 1 & - & 1 & 1 & 2 \\ 3 & 1 & 1 & - & 2 & 1 \\ 4 & 1 & 1 & 2 & - & 1 \\ 5 & 2 & 2 & 1 & 1 & - \end{pmatrix}$$

- Para el nodo 1 la centralidad será 0 puesto que no se encuentra en ninguno de los caminos más cortos entre dos nodos cualesquiera del grafo
- Para el nodo 2 podemos ver los caminos más cortos entre otros dos nodos cualesquiera de la red en los que se encuentra

$$\begin{pmatrix} Nodos & 1 & 2 & 3 & 4 & 5 \\ 1 & - & - & 1 & 1 & 2 \\ 2 & - & - & - & - & - \\ 3 & 1 & - & - & 1 & 0 \\ 4 & 1 & - & 1 & - & 0 \\ 5 & 2 & - & 0 & 0 & - \end{pmatrix}$$

$$Centralidad_{Betwenness} = \frac{1}{1} + \frac{1}{1} + \frac{2}{2} + \frac{1}{2} + \frac{0}{1} + \frac{0}{1} = 3.5$$

- Para el nodo 3 obtenemos $Centralidad_{Betwenness} = 1$
- Para el nodo 4 $Centralidad_{Betwenness} = 1$
- Para el nodo 5 $Centralidad_{Betwenness} = 0.5$

Observamos cómo el nodo 2 es más importante según la intermediación. Observando el grafo nos damos cuenta de que es así puesto que para la comunicación entre nodos de la red, mucha de la información tendrá que pasar por este nodo.

3.5.3.4. Centralidad de Autovectores

En el caso de la centralidad basada en los grados de los nodos, consideramos los nodos más importantes como aquellos que tienen una mayor cantidad de relaciones. Sin embargo, muchos de los escenarios en el mundo real, como por ejemplo tener muchos amigos, de por sí no te garantiza que sean los más importantes. Seguiremos un criterio que se fije en la importancia que tienen los individuos con los que se relaciona el estudiado [24]

Teorema 3.2. De Perron-Frobenius.- Sea $A \in \mathbb{R}^{nxn}$ la matriz de adyacencia de un grafo conexo tal que $A_{i,j} > 0$, (es decir, es una matriz n-cuadrada no negativa), Entonces existe un $r \in \mathbb{R}^{+}$ (denominado autovalor de Perron-Frobenius) $\lambda_{\max}$, tal que $\lambda_{\max}$ es un autovalor de A y cualquier otro autovalor es estrictamente menor que $\lambda_{\max}$ .Además, existe un autovector correspondiente $v = (v_1, v_2, \dots, v_n)$ de A con $\lambda_{\max}$ tal que $\forall\, v_i > 0$. [29]

La centralidad de autovector intenta generalizar la centralidad por grado incorporando la importancia de los vecinos. Para esto, utilizaremos la matriz de adyacencia A del grafo.

Definición 3.37.- La función $Centralidad_{Autovectores}(v_i)$ que devuelve la centralidad de autovector del nodo v_i se corresponde con una suma de las demás centralidades utilizando las proporciones adecuadas [3]. De la siguiente forma

$$Centralidad_{Autovectores}(v_i) = \frac{1}{\lambda}\sum_{j=1}^{n} A_{j,i} Centralidad_{Autovectores}(v_j)\ , \quad (50)$$

donde λ es una constante fijada. Podemos reescribir la ecuación si representamos el sumatorio como un producto entre A^T y el vector C_a que tiene las centralidades. De la siguiente forma

$$\lambda\, C_a = A^T\, C_a \quad .(51)$$

Por tanto C_a ,vector que contiene las centralidades de los nodos, es un autovector de la matriz de adyacencia A^T y donde λ es su autovector.

Corolario 3.1.- Para el cómputo de la centralidad por autovector deberemos realizar el cálculo de autovalores de la matriz de adyacencia A y seleccionar el mayor. Su correspondiente autovector será C_a. A partir del Teorema de Perron-Frobenius, todos los componentes de C_a serán positivos y se corresponde con las centralidades del grafo [3]

Ejemplo 3.10.- Sea el grafo

Ilustración 17 Grafo de ejemplo

Cuya matriz de adyacencia es

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Basados en la ecuación anterior ,(47), deberemos resolver $\lambda\, C_a = A\, C_a$ o $(A - \lambda\, I)\, C_a = 0$. Definimos el vector $C_a = [u1 \quad u2 \quad u3]^T$

$$\begin{bmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{bmatrix} \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Si resolvemos la ecuación obtenemos $\lambda = -\sqrt{2}, 0, \sqrt{2}$. Por tanto seleccionamos el mayor: $\lambda = \sqrt{2}$, y calculamos su autovector correspondiente

$$\begin{bmatrix} 0-\sqrt{2} & 1 & 0 \\ 1 & 0-\sqrt{2} & 1 \\ 0 & 1 & 0-\sqrt{2} \end{bmatrix} \begin{bmatrix} u1 \\ u2 \\ u3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Si resolvemos la ecuación obtenemos el vector normalizado

$$C_a = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Observamos por el resultado devuelto que el nodo 2 es el que mayor centralidad representa. Esto se corresponde con la intuición al observar visualmente el grafo, debido a que se encuentra en el medio y las únicas relaciones que se encuentran en la estructura.

3.5.4. Conectividad

Definición 3.38.- Decimos que la conectividad de una red es robusta (o tolerante a desconexiones) si tras la eliminación de muchos de sus nodos sigue conteniendo una componente conexa gigante. En cambio, si con la eliminación de una proporción baja de nodos específicos, como los de alto grado, nos permite convertirla en dividida y dispersa la denominaremos frágil [22]

Definición 3.39.- Decimos que una componente conexa es gigante cuando contiene la gran mayoría de los nodos y la siguiente agrupación conexa es mucho más pequeña en tamaño [22]

En las redes sociales, las redes conformadas son frágiles, puesto como hemos comentado antes, si quitamos uno de los nodos con un alto grado o concentrado se pasa a tener poca conectividad [3]

3.6. Ejemplo de Aplicación

En este apartado realizaremos un análisis con datos reales mediante los métodos y técnicas descritos. La aplicación de éstos nos permite comprender mejor algunas características de las redes sociales. En este caso utilizaremos la red social Twitter, donde podremos reunir bastante información útil con la que trabajar. Para la descarga y almacenamiento de datos procedentes de Twitter se hará uso de las herramientas desarrolladas en [30].

Procedemos a mostrar un ejemplo de aplicación en el caso de que queramos revisar los tuits generados por una temática. En este caso hemos almacenado unos tuits acerca del evento internacional Eurovisión. Se ha decidido escoger un tópico tratado en España por facilidad de idioma y de cara a la posible inclusión posteriormente de los sistemas automáticos de análisis de sentimiento. Su evolución temporal, en escala de 1 minuto desde las 19:00 hasta las 00:28 es la representada a continuación.

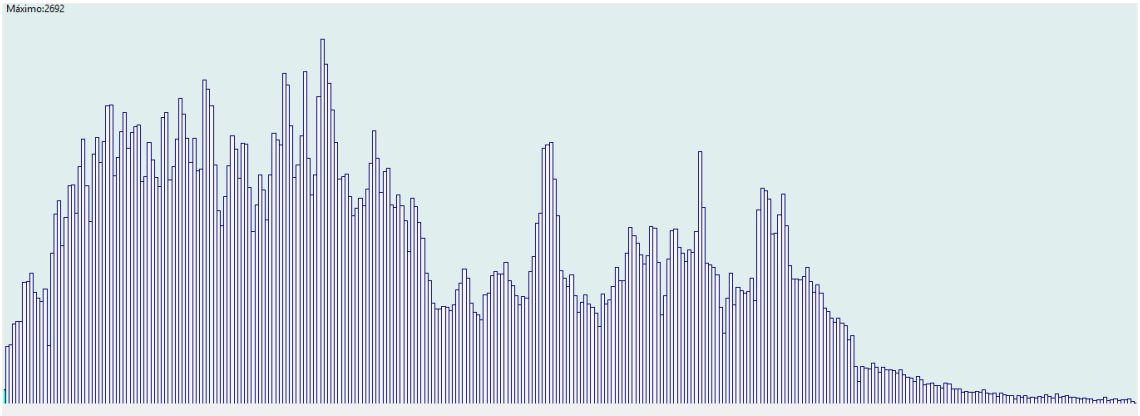


Ilustración XVIII Evolución temporal del hashtag #EurovisionTVE

En la siguiente tabla representamos algunos datos con respecto al movimiento y dinámica del tema.

Fecha inicio	14 de Mayo de 2016 19:00
Fecha fin	15 de Mayo de 2016 00:28
Cantidad de Tuits	332.164
Usuarios partícipes	75.245
Velocidad media	1.239 Tuits/minuto
Máximo	2692 Tuits/minuto (14 de Mayo de 2016 20:32)

Tabla vii Datos correspondientes al seguimiento del Hashtag #EurovisionTVE

Podemos observar la gráfica y preguntarnos qué relación existe entre el pico de mensajes realizados y el evento. Mediante una comprobación podemos ver que este suceso de las 20:32 se corresponde con un punto clave, la actuación de la cantante española.

Si procedemos a la visualización de las conversaciones, deberemos buscar aquellos mensajes que contestan a otros, es decir, son réplica. Obtenemos la visualización correspondiente en la siguiente ilustración

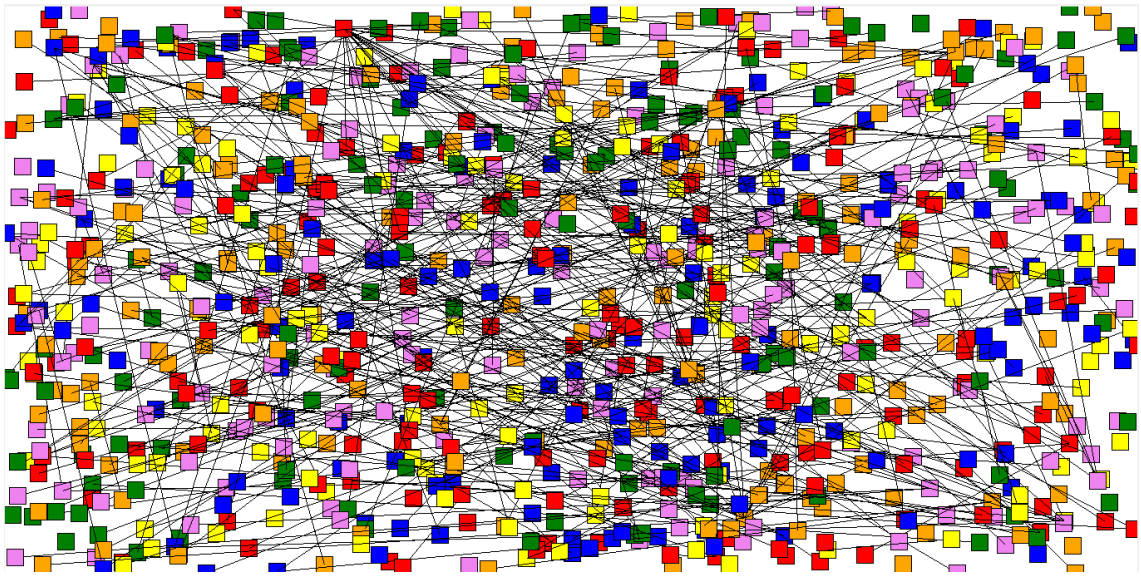


Ilustración XIX Visualización de las conversaciones en el HashTag #EurovisionTVE

Está compuesta por un total de 966 nodos entre los cuales hay 764 relaciones. Además de éstos debemos separar los que son aislados 390, es decir, los usuarios que han realizado réplicas, pero, o bien la ventana temporal no ha recogido el tuit al que se replicaba o ha sido a un mensaje publicado por ellos mismos. Su eliminación de la red se corresponde con la siguiente ilustración.

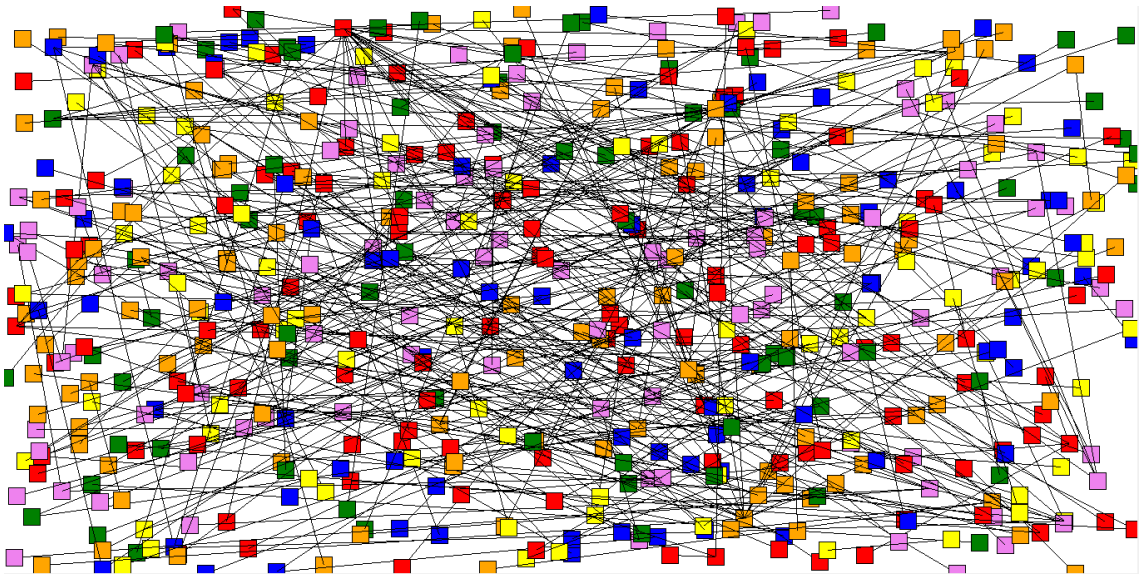


Ilustración XX Representación de la red sin nodos aislados

A continuación, podemos basarnos en la observación de algunas medidas de centralidad. En este caso, vamos colorear los nodos según su grado.

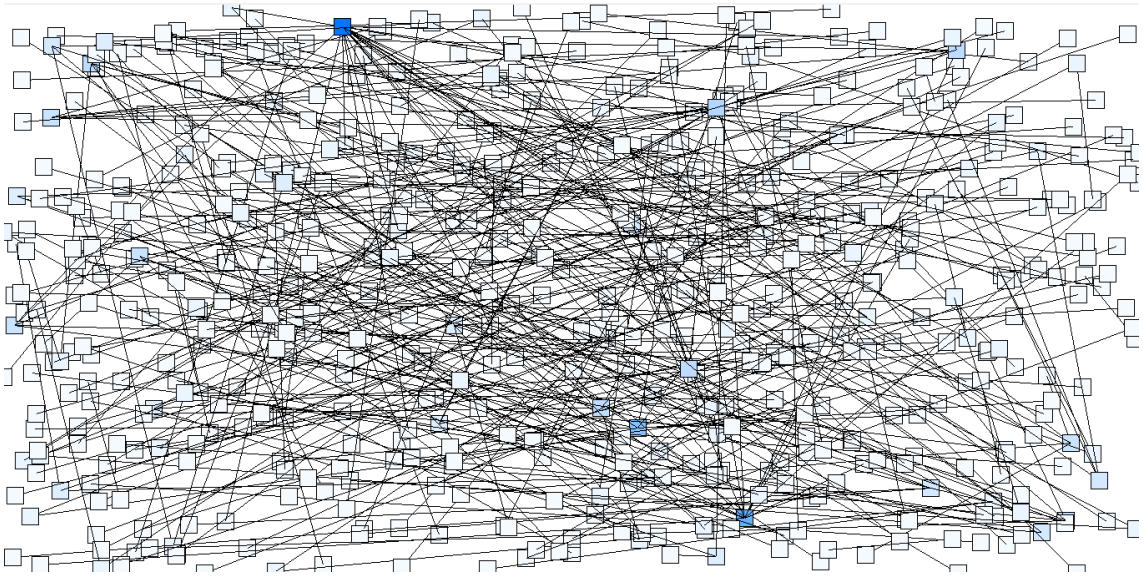


Ilustración XXI Representación de la red coloreada por grado

La gama de azules indica la diferencia de grados. Cuanto más cercano al blanco menor es el grado y cuanto más fuerte sea el color mayor es la cantidad de relaciones que tiene el nodo en la red. Nuevamente, como en el caso del pico de tuits, podemos ver cuáles son los nodos que componen el subgrafo, Ilustración XXII Subgrafo donde se encuentra el nodo con mayor grado de la red, y por qué es ése el que más fuerza tiene dentro de la red.

Si utilizamos las fuentes disponibles online que nos permiten descubrir el usuario de Twitter mediante su identificador, como puede ser Tweeterid.com, descubrimos que se trata de un usuario con más de 2.7 millones de seguidores y que realiza seguimiento tan sólo a 1.000. Esto supone un ratio de 270.000% de seguidores vs. seguidos.

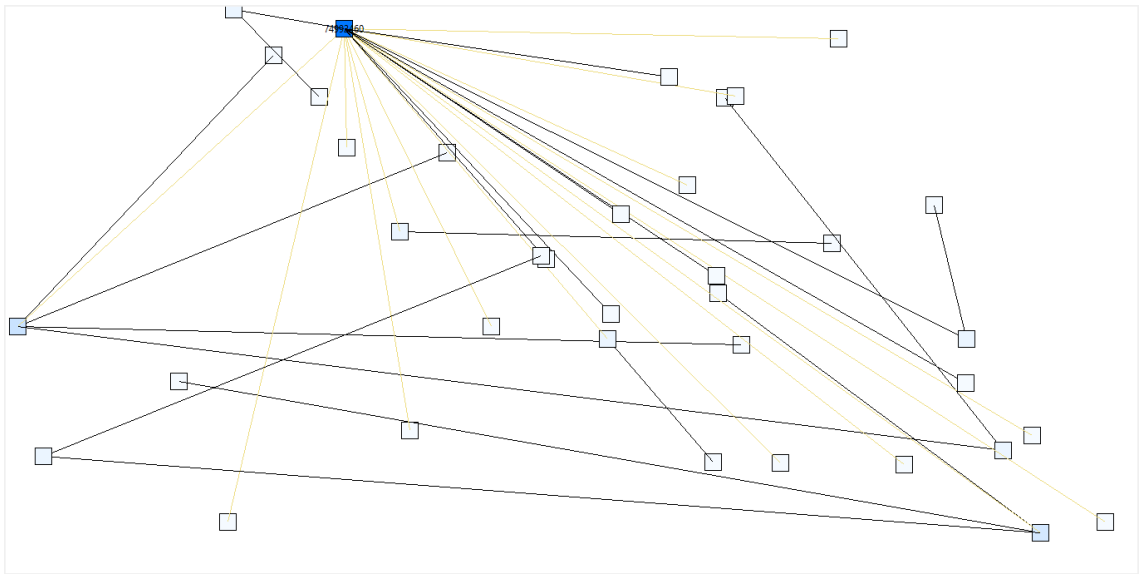


Ilustración XXII Subgrafo donde se encuentra el nodo con mayor grado de la red

En esta red comprobamos que la fragilidad de las redes sociales es alta puesto que el nodo remarcado con un color azul más intenso representa el de mayor grado. Claramente, si lo eliminamos, desconectaríamos gran parte de las conexiones entre la red.

Otros aspectos de estudios son los de las relaciones que se crean. Aquí podemos visualizar la distribución de grados

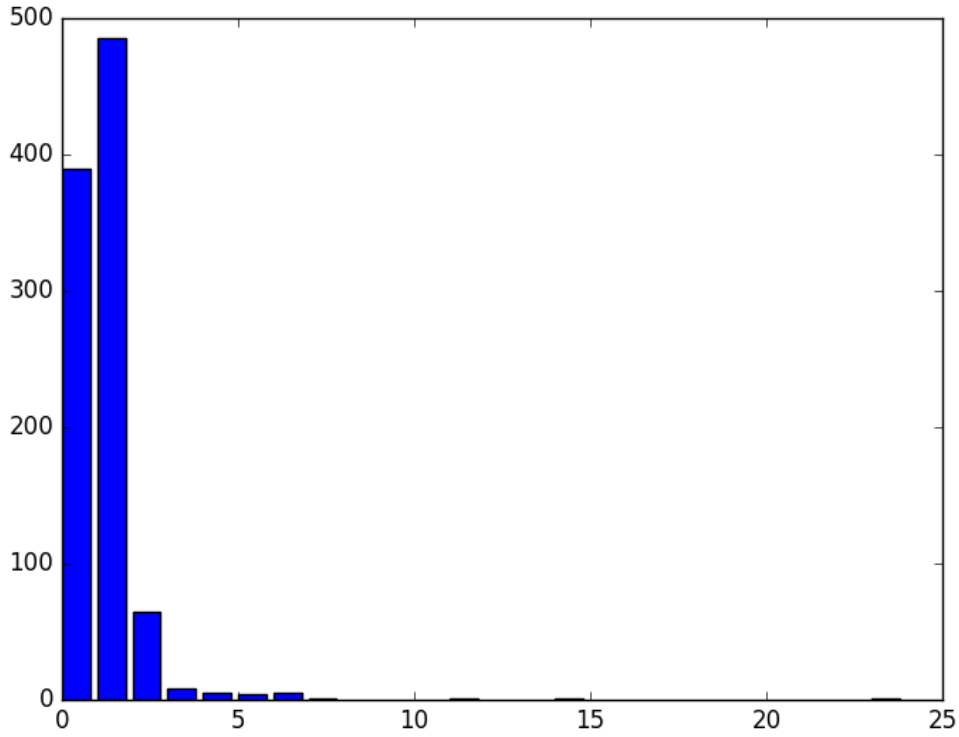


Ilustración XXIII Distribución de los grados en el HashTag #Eurovisión

Comprobamos que efectivamente, obtenemos una gran cantidad de nodos con pocas relaciones y muy pocos con muchas. Estos últimos serán por tanto los que concentrarán mayor centralidad.

A nivel de clusters obtenemos la siguiente información

Cantidad de clusters	587
Tamaño más pequeño	1 nodos
Tamaño más grande	37 nodos
Cantidad de triángulos	3
Coeficiente de transitividad	0.13
Coeficiente de clustering medio	0.13

Donde observamos un número muy elevado de clusters muchos de ellos de pequeño tamaño. La mayor cantidad de nodos se corresponde con el cluster al que pertenece el nodo con mayor grado, esto nos ayuda a identificarlo intuitivamente como una agrupación principal o componente de la red. Como en casi todos las redes observadas a partir de interacciones sociales, [22] se aprecia un muy bajo coeficiente de transitividad y número de triángulo, lo que

se resume en una interacción entre dos individuos en la que difícilmente participan más, en este caso concreto.

4. Conclusiones y Trabajo futuro

Una vez los fundamentos y herramientas matemáticas básicas que conformaban el proyecto han sido desgranadas se debe proceder a una revisión de las conclusiones, mejoras y posibles pasos posteriores al actual trabajo realizado.

4.1. Conclusiones

Las principales conclusiones obtenidas a partir del proyecto están relacionadas con el análisis de Sentimiento, donde la clasificación por Naive-Bayes se muestra como una idea muy intuitiva y recomendable para la automatización de la clasificación de documentos.

En cuanto a la Teoría de Grafos y Redes, hemos visto una gran capacidad de aplicación sobre las estructuras generadas en redes sociales, en este caso concreto, sobre Twitter, se corresponde con la idea intuitiva de una gran cantidad de nodos y un bajo coeficiente de conexión en el total de la red.

Por tanto, la revisión de los fundamentos teóricos asociados a la clasificación de sentimiento en textos y teoría de grafos y redes ha permitido conocer los cimientos de temas latentes en investigaciones sobre dinámicas en redes sociales, análisis semántico y aplicaciones de redes. Además, la definición y resolución de ejemplos permite el asentamiento de los métodos desde el punto de vista más práctico.

4.2. Líneas de trabajo futuras

En este proyecto hemos visto dos puntos principales en el planteamiento de análisis de estructura de las redes sociales, y en concreto en Twitter. El análisis de sentimiento y la modelización mediante redes. De esto podemos sacar las siguientes conclusiones y líneas de trabajo futuras:

- Las redes sociales generan grandes cantidades de información que son interesantes para conocer la estructura del comportamiento humano. Esto puede aplicarse a otros campos como el aprendizaje (como ya se recalca en [31]), ámbitos sociológicos o políticos, el Big Data aplicado al Business Intelligence, entre otros.
- Indagación exhaustiva sobre comunidades e interacción entre los usuarios realizada mediante el seguimiento o la “amistad”. Esto es aplicable no sólo a Twitter, si no que podríamos analizar casi cualquier otra red social.
- Hemos visto el aprendizaje Bayesiano y en concreto el clasificador Naive-Bayes. Pero podemos estudiar la aplicación de clasificadores Bayesianos creados mediante redes Bayesianas, clasificadores SemiNaive-Bayes, clasificadores Bayesianos k-dependientes.
- La revisión de estos conceptos tiene una clara aplicación práctica. Por lo que se puede realizar una implementación en forma de aplicación que nos ayude

a analizar u obtener características de datos descargados desde redes sociales como Twitter u otras.

- Mejoras e investigación sobre nuevas medidas de centralidad.
- Creación de modelos temporales donde cuantificar también la influencia temporal, es decir, a partir de un momento dado en el que se produce una interacción entre dos individuos la relación entre ellos descenderá hasta 0 si no se repite.
- Implementación de éstos métodos como herramienta software que nos facilite el manejo y análisis de los tuits desde el punto de vista del sentimiento y de estructuración en redes.

Glosario

FIFO – First In First Out

LIFO – Last In Firs Out

LSA – Latent Semantic Analysis

NLP – Natural Language Processing

NLTK – Natural Language Tool Kit

SVD – Singular Value Decomposition

WWW – World Wide Web

Referencias

[1] V. Metsis, I. Androutsopoulos y G. Paliouras, «Spam Filtering with Naive Bayes – Which Naive Bayes?» Atenas, 2006.

[2] B. Pang, L. Lee y S. Vaithyanathan, «Thumbs up? Sentiment Classification using Machine Learning Techniques».

[3] L. Becchetti, E. Koutsoupias y S. Leonardi, «Online Social Networks and Networks Economics» Universidad de Roma, 2010.

[4] T. K. Landauer, D. S. McNamara, S. Dennis y W. Kintsch, Handbook of Latent Semantic Analysis, Routledge, 2011.

[5] D. Haussler y M. Kearns, «Machine Learning» Kluwer Academic Publishers, Boston, 1994.

[6] D. Laham, «Latent Semantic Analysis approaches to categorization» University of Colorado, Boulder, 2000.

[7] T. M.Mitchell, «Machine Learning» McGraw-Hill International Editions, 1997.

[8] C. D.Manning, P. Raghavan y H. Schüze, «An introduction to Information Retrieval» Cambridge University Press, Cambridge, 2009.

[9] Universidad de Granada, «Probabilidad condicionada: Teoremas básicos. Independencia de sucesos» de *Estadística descriptiva e introducción a la probabilidad*.

[10] Ó. J. P. Izquierdo y R. C. Díaz, «Aprendizaje Bayesiano».

[11] C. M. Grinstead y J. L. Snell, «Introduction to Probability» American Mathematical Society, 2nd edition 2012.

[12] S. Deerwester, S. T. Dumais y R. Harhsman, «Indexing by Latent Semantic Analysis» *Journal of the American Society For Information Science,* pp. 391-407, 1990.

[13] W. Kintsch, «Predication» 2001.

[14] K. Baker, «Singular Value Decomposition Tutorial» 2005 (Revisado en 2013).

[15] A. Huang, «Similarity Measures for Text Document Clustering».

[16] S. A. Morris, «Topology without tears» 2011.

[17] N. A. H. Ishtayeh, «Similarity Threshold Determination for Text Document Clustering» Zarqa University, Zarqa, Jordan, 2014.

[18] «Wikipedia» 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Stemming.

[19] M.F.Porter, «Snowball» 2001. [En línea]. Available: http://snowball.tartarus.org/texts/introduction.html.

[20] «Tex-Processing» 2016. [En línea]. Available: http://text-processing.com/demo/stem/.

[21] «Wikipedia» 2016. [En línea]. Available: https://en.wikipedia.org/wiki/Graph_theory.

[22] D. Acemoglu y A. Ozdaglar, «Graph Theory and Social Networks» 2009.

[23] F. R. Herrera, «Análisis de Redes Complejas. Identificación y análisis de vulnerabilidades en infraestructuras de hardware y sofware».

[24] R. Zafarini, M. Ali Abbasi y H. Liu, «Social Media Mining: An Introduction» Cambridge University Press, 2014.

[25] D. Jungnickel, «Graphs, Networks and Algorithms» Springer-Verlag, 2004.

[26] «Wikipedia» 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra.

[27] A. Clauset, C. Dohilla Shalizi y M. E. J. Newman, «Power-law distributions in empirical data» Society for Industrial and Applied Mathematics, 2009.

[28] F. Berzal, «Clustering» Universidad de Granada, 2009.

[29] I. Zaballa, «Valores singulares ¿Qué son? ¿Para qué sirven?».

[30] J. Pertierra das Neves, «Diseño e implementación de una herramienta para el análisis estructural y semántico de Twitter» 2016.

[31] A. Thomo, «Latent Semantic Analysis (Tutorial)».

[32] T. Recio, «La Gaceta de la Real Sociedad Matemática Española» p. Vol. 8.3, 2005.

[33] F. Rahutomo, T. Kitasuka y M. Aritsugi, «Semantic Cosine Similarity» 2014.

[34] «Math Insight» 2016. [En línea]. Available: http://mathinsight.org/degree_distribution#directed.

[35] J. F. Padgett, «Marriage and elite structure in renaissance Florence, 1282 - 1500» 1994.