Tema 3 Representación de documentos

Índice

- Conceptos preliminares
- Modelos de representación
- Funciones de pesado
- Funciones de selección y reducción de rasgos
- Enlaces de interés
- Bibliografía

Conceptos preliminares

- Para que los algoritmos de minería de textos puedan procesar un texto se deben extraer sus características o rasgos
- El contenido textual se puede representar de diferentes formas (cadenas de caracteres, palabras, estructuras sintácticas, grafos de relaciones, predicados, etc.)
- La representación de un texto deberá ser fiel a su contenido, incluyendo la información necesaria para poder extraer el conocimiento útil que se espera obtener
- Además, deberá ser adecuada a las especificaciones de los algoritmos que se empleen a continuación

Conceptos preliminares

- Diferentes tareas de la minería de textos y diferentes tipos de textos pueden requerir diferentes representaciones

- En [Hotho et al 2005] se presentan los principios de la minería de textos y se detallan los modelos de representación de textos más habituales

Índice

- Conceptos preliminares
- **Modelos de representación**
- Funciones de pesado
- Funciones de selección y reducción de rasgos
- Enlaces de interés
- Bibliografía

Modelos de representación

Los métodos de representación más utilizados son:

- Modelo del espacio vectorial (Vector Space Model –VSM-)
- Modelo probabilístico (Probabilistic Topic Model)
- Modelo estadístico (Statistical Language Model)
- Modelos semánticos vectoriales (Vector semantic Models)

Nos centraremos en el modelo del espacio vectorial y en el semántico vectorial pero se darán nociones del resto de modelos

Modelos de representación: vectorial

Un **modelo de representación vectorial** de documentos se puede definir como una cuádrupla:

$$<\boldsymbol{X}, \boldsymbol{B}, \boldsymbol{\mu}, \boldsymbol{F}>$$

- Conjunto de objetos ***X*** (cadenas) que constituye el **Vocabulario**
- **Álgebra *B*** que generaliza las relaciones entre los objetos
- Función de medida o **métrica**, ***μ***, que establece la distancia (o similitud) entre objetos
- **Función de pesado *F*** que establece la proyección de cada objeto del espacio sobre la recta real

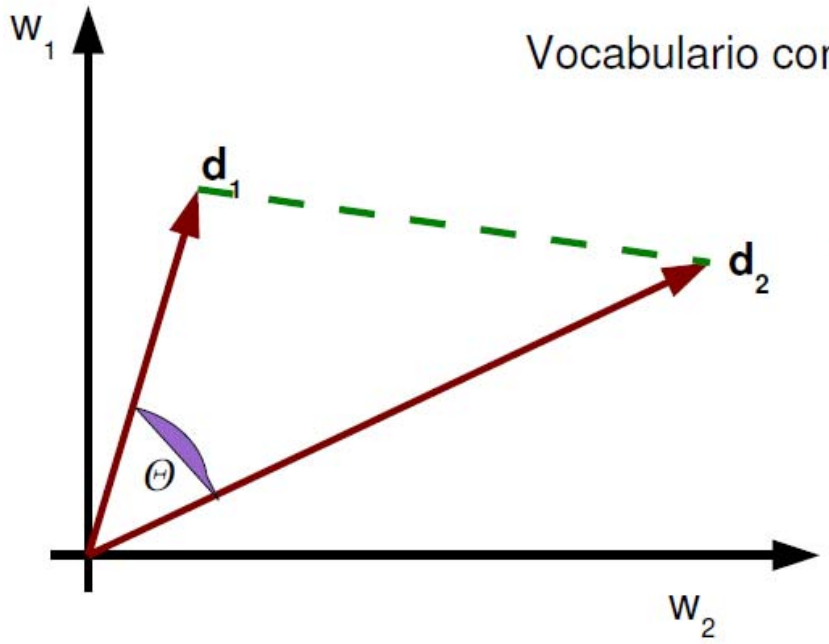
Modelos de representación: vectorial

- Los documentos se modelan como conjuntos de rasgos (cadenas de caracteres: tokens, palabras, n-gramas, …) que pueden ser individualmente tratados y pesados

- Los documentos pasan a ser representados como vectores dentro de un espacio euclídeo, de forma que midiendo la distancia entre dos vectores se tratará de estimar su similitud como indicador de cercanía semántica

- Se pueden aplicar diferentes métricas μ para el cálculo de la distancia: distancia euclídea y coseno son las más populares pero hay muchas otras

Modelos de representación: vectorial



Vocabulario con dos rasgos: $V = \{w_1, w_2\}$

documento *i*: $\mathbf{d}_i = (w_{i1}, w_{i2})$

documento *j*: $\mathbf{d}_j = (w_{j1}, w_{j2})$

Posibles métricas:

distancia euclídea : $|d_1 - d_2|$

distancia coseno : $\cos \Theta$

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM) [Salton 1975]

- Asume el "principio de independencia“ entre rasgos: las cadenas (rasgos) aparecidas en un mismo texto no tienen relación entre sí y se pueden cuantificar individualmente

- No tiene en cuenta el orden en el que aparecen los rasgos en el texto: la semántica de un documento queda reducida a la suma de los significados de los rasgos que contiene

- Aunque estas suposiciones no son correctas desde el punto de vista lingüístico, reducen la complejidad sin penalizar en muchos casos de forma sustancial los resultados

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM)

- Un documento quedará representado como una combinación lineal de vectores base t_i donde cada coeficiente t_{ij} de la combinación representará la relevancia de cada rasgo en el contenido del documento, calculada con una función de pesado **F**

$$\vec{t_1} = (1, \ldots\ldots\ldots, 0)$$

$$\vec{t_i} = (0, \ldots, 1, \ldots, 0)$$

$$\vec{t_n} = (0, \ldots\ldots\ldots, 1)$$

$$\vec{d_j} = t_{1j}\ \vec{t_1} + t_{2j}\ \vec{t_2} + \ldots + t_{nj}\ \vec{t_n}$$

$$\vec{d_j} = (t_{1j}, t_{2j}, \ldots, t_{nj})$$

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM)

- Así, definir una representación dentro del VSM se reduce a encontrar una función de asignación de pesos F: $f(t_i) = t_{ij}$ para calcular el valor de cada componente del vector d_j

- Dos representaciones de un mismo documento d_i (d_i y d'_i) a partir del mismo vocabulario serán diferentes siempre que el conjunto de valores t_{ij} que tomen las componentes de los vectores d_i y d'_i sean diferentes; es decir,

$$\vec{d_j} \neq \vec{d_j'} \text{ si y sólo si } \{i | t_{ij} \neq t'_{ij}, \forall i\} \neq \emptyset$$

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM)

- **Bolsa de palabras** --Bag of Word (BoW)– es uno de los métodos más básicos de representar un documento

- En la representación BoW se representa un documento utilizando un vector con tantas componentes como palabras/términos haya en el vocabulario y el valor de la componente se calcula utilizando una función de pesado

- Una colección de documentos se suele representar mediante una **matriz término-documento**, en la que cada fila representa una palabra en el vocabulario de la colección y cada columna representa un documento de la colección

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM)

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Ejemplo extraído de [Jurafsky and Martin 2019], Cap. 6

- La tabla representa una selección de filas (términos) de la colección de 4 obras de Shakespeare (documentos) y cada celda o posición la frecuencia con la que la palabra (fila) aparece en el documento (columna)

- En este ejemplo un documento (columna) es un punto es un espacio de 4 dimensiones (vocabulario)

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM)

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Ejemplo extraído de [Jurafsky and Martin 2019]

- Este tipo de matrices se utilizaron primeramente para determinar la similitud entre documentos en Recuperación de Información (*Information Retrieval*): dos documentos similares tendrán palabras similares y eso quedará patente en sus vectores que también serán similares

- Los tamaños de los vocabularios en colecciones extensas son de cientos de miles y el número de documentos también puede llegar a ser muy alto, lo que da lugar a matrices dispersas (con muchos ceros)

Modelos de representación: vectorial

Modelo del espacio vectorial (VSM). Limitaciones:

- **Alta dimensionalidad de la representación**: las funciones de selección y reducción de rasgos ayudan a reducir la dimensionalidad. Suele dar como resultado vectores muy largos y dispersos en colecciones extensas (contienen principalmente ceros) ya que muchas las palabras nunca ocurren en el contexto de otras

- **Pérdida de correlación con las palabras (términos, rasgos) adyacentes**: algunas propuestas utilizan para representar un documento secuencias de símbolos (carácteres, palabras) denominadas n-gramas, donde *n* indica el tamaño de la secuencia, o utilizan términos multipalabra que previamente deben ser identificados

- **Pérdida de posibles relaciones semánticas** entre los términos de un documento: algunas propuestas usan el Índice de Latencia Semántica (LSI) que es un método de análisis de coaparición entre rasgos y se basa en el análisis de latencia semántica (*Latent Semantic Analysis*, LSA), otras utilizan representaciones basadas en ontologías [Hotho et al 2001]

Modelos de representación: probabilística

Modelo probabilístico de temas (Probabilistic Topic Model)

- Los documentos se representan como una mezcla de temas, donde un tema es una distribución de probabilidad sobre palabras

- Se asume que si un documento trata de un tema en particular, es de esperar que aparezcan palabras concretas en el documento con mayor o menor frecuencia: “carrocería”, “llantas” o “volante" aparecerán más a menudo en documentos sobre coches que en documentos de otros temas, mientras que palabras de uso general como "el“, “de“ o “es” aparecerán igualmente en todos

- Un documento típicamente trata múltiples temas en diferentes proporciones; por lo tanto, en un documento que es 80% sobre coches probablemente habría sobre ocho veces más palabras relacionadas con el tema “coches” que de otros temas

Modelos de representación: probabilística

Modelo probabilístico de temas (Probabilistic Topic Model)

- Los "temas" producidos por las técnicas de este modelado son grupos de palabras relacionadas. Un modelo temático examina un conjunto de documentos y descubre a partir de las estadísticas de las palabras de cada uno cuáles podrían ser los temas y cuál es el equilibrio temático de cada documento

- Este modelo supera los problemas asociados con “el término” como tema. Un término puede ser una palabra o una frase. Sin embargo este modelo no puede representar temas complejos, capturar las variaciones de vocabulario y la ambigüedad del sentido de las palabras

- La Indexación Semántica Latente Probabilística -- Probabilistic Latent Semantic Indexing (PLSI)-- y la Asignación Latente de Dirichlet -- Latent Dirichlet Allocation (LDA) -- son los dos métodos de modelización de temas más conocidos

- Más información y punteros en https://en.wikipedia.org/wiki/Topic_model#Algorithms

Modelos de representación: estadística

Modelo de lenguaje estadístico de temas (Statistical Language Model)

- Es una distribución de probabilidad sobre secuencias de palabras

- Se aplica en muchas tareas relacionadas con procesamiento de lenguaje natural: reconocimiento de voz, traducción automática, clasificación y enrutamiento de documentos, reconocimiento óptico de caracteres, recuperación de información, reconocimiento de escritura a mano, corrección ortográfica, …

- Uno de los modelos más utilizado es el modelo de n-gramas

Modelos de representación: estadística

Modelo de lenguaje estadístico de temas (Statistical Language Model)

- El modelo de n-gramas assume que la probabilidad de una palabra depende de las n palabras anteriores

- Es decir, la probabilidad $P(w_1,w_2,w_3,...,w_m)$ de observar la oración w_1, w_2, ..., w_m se aproxima de la siguiente manera:

$$P(w_{1,}\, w_{2,\ldots} w_m) = \prod_{i=1}^{m} (w_i |\, w_{1,} \ldots\, w_i | w_{i-1})$$

- Cuando en este modelo *n* = 1, es decir no hay dependencia de palabras previas es equivalente a la bolsa de palabras (Bag of Words)

- Más información y punteros en https://en.wikipedia.org/wiki/Language_model y en el capítulo 3 de [Jurafsky and Martin 2019]

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

(recomendamos estudiar primero el apartado de funciones de pesado)

- Se basan en la **hipótesis distribucional** [Joos 1950] que dice que las palabras que ocurren en contextos similares tienden a tener significados similares

- Relacionan la similitud en la distribución de las palabras y la similitud en su significado: palabras sinónimas tienden a aparecer en contextos similares

- Se hace corresponder la diferencia de significado entre dos palabras con la diferencia en los contextos en los que aparecen

- En este modelo a cada palabra se le asocia un vector y se representa como un punto en algún espacio semántico multidimensional

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- En lugar de una matriz de término-documento como en el modelo del espacio vectorial, se utiliza una **matriz palabra-palabra (término-término)** que también se denomina **matriz término-contexto** dado que las columnas son también palabras, no documentos

- La dimensionalidad de esta matriz para un vocabulario V sería |V| x |V| donde cada elemento o celda representaría el número de veces que la palabra de la fila (objetivo) y la palabra de la columna (contexto) coaparecen en un determinado contexto en un corpus

- Hay que determinar el tamaño del contexto de la palabra objetivo: normalmente se define una ventana de varias palabras por delante y por detras de una considerada objetivo y ese sería el contexto a tener en cuenta

- En [Jurafsky and Martin 2019] , Cap. 6, pág. 9—10 se puede ver un ejemplo de matriz término-contexto de unos textos extraídos de Wikipedia

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- Para determinar la similitud entre dos palabras v y w se suele utilizar una medida de similitud entre sus correspondientes vectores: el coseno

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

- El valor del coseno varía desde 1 para vectores que apuntan en la misma dirección, a través de 0 para vectores que son ortogonales, a -1 para vectores que apuntan en direcciones opuestas. Como los valores de frecuencia que contienen los vectores no son negativos el coseno varía de 0 a 1

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- Si se representa una palabra como un vector con las dimensiones de todo el vocabulario muy probablemente el vector será disperso (muchos elementos con valor 0) pero se puede limitar el numero de las dimensiones de forma que la mayoría de los valores de los vectores sean distintos de cero, dando lugar a vectores densos. Dimensiones habituales son valores como 256, 512, o 1024

- Utilizar la frecuencia como pesado en los vectores término-contexto no se considera la mejor opción para determinar asociaciones entre palabras ya que en el contexto de una palabra objetivo normalmente ocurren palabras frecuentes con todo tipo de palabras y no son informativas sobre ninguna palabra en particular (artículos, determinantes, verbos auxiliares, …)

- En lugar de la frecuencia se suele usar TF-IDF: en este modelo representa lo relevante que es una palabra con respecto a otra en una colección y permite determinar que algunas palabras son generalmente más comunes que otras en las colecciones

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- Con este modelo se puede por ejemplo:
  - encontrar las x palabras más similares a una objetivo calculando los cosenos entre dicha palabra y el resto de las palabras del vocaculario, ordenar el resultado y seleccionar las x con valores más altos
  - decidir si dos documentos son similares considerando los vectores de todas las palabras de un documento y calculando el centroide de todos los vectores. El centroide de un conjunto de k vectores de palabras correspondientes a un documento es un vector que se calcula de la siguiente manera:

$$d = \frac{w_1 + w_2 + \cdots + w_k}{k}$$

Dados dos documentos podemos calcular sus vectores documento con la formula anterior y estimar su similitud utilizando el coseno

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- Una función de ponderación alternativa a TF-IDF es **Información Mutua Puntual Positiva PPMI (*Positive Pointwise Mutual Information*)**

- PMI es una medida de la frecuencia con la que ocurren dos eventos x e y, comparados con lo que esperaríamos si fueran independientes. Como puede dar como resultado valores negativos, se utiliza PPMI, que en vez de valores negativos devuelve 0

- Si tenemos una matriz de co-ocurrencia F con W filas (palabras) y C columnas (contextos), donde $f_{i,j}$ es el número de veces que la palabra w_i aparece en el contexto c_j:

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad \text{PPMI}_{ij} = \max(\log_2 \frac{p_{ij}}{p_{i*}p_{*j}}, 0)$$

- En [Jurafsky and Martin 2019] , Cap. 6 (sección 6.7), se puede ver un ejemplo

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- La idea de la semántica vectorial es representar una palabra como un punto en algún espacio semántico multidimensional, las representaciones resultantes también reciben el nombre de **representaciones distribuidas**

- Los vectores para representar palabras se llaman ***embeddings*** (embebidos) porque la palabra está embebida en un espacio vectorial en particular

- Los modelos semánticos vectoriales son muy prácticos porque se pueden aprender automáticamente a partir de texto sin necesidad de etiquetado o supervisión

- Actualmente son la forma habitual de representación en muchas tareas de procesamiento de lenguaje natural: traducción automática, análisis de opinion, generación de textos, Chatbox (sistemas que responden preguntas de usuarios), entre otras

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model)

- Para trabajar con representaciones distribuidas una opción es generar nuestro propio modelo de vectores o bien se puede partir de un modelo ya entrenado

- Estas representaciones permiten utilizar vectores densos, en los que se limita el tamaño de las dimensiones y la mayoría de los valores de los vectores son distintos de cero

- Hay varios modelos de word embedding disponibles: word2vec (Google); Glove (Stanford, [Pennington et al. 2014]); fastTest (Facebook, https://fasttext.cc/); y más recientemente el modelo de Bert (Google, https://arxiv.org/pdf/1810.04805.pdf) o Elmo (https://allennlp.org/elmo)

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model): Word2vec [Mikolov et al. 2013]

- **Word2vec** es una herramienta para entrenar/generar y usar word embeddings que utiliza modelos predictivos

- Los modelos predictivos intentan predecir directamente una palabra a partir de sus vecinos en términos de vectores pequeños y densos que se aprenden durante el entrenamiento a partir del corpus sin etiquetar. La idea detrás de estos métodos es que si podemos predecir en qué contexto aparece una palabra, entonces significa que entendemos el significado de la palabra en su contexto

- Su objetivo es agrupar vectores de palabras similares en el espacio multidimensional

- Word2vec utiliza una red neuronal de 2 capas que toma como entrada un corpus textual y genera sus correspondientes word embeddings que pueden ser la entrada de una red de aprendizaje profundo o servir para detectar relaciones entre palabras

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model): Word2vec

- Word2Vec implementa dos modelos neuronales:
  - CBOW: el modelo predice la palabra objetivo dada una ventana de palabras (el contexto)
  - Skip-gram: el modelo predice las palabras contexto a partir de la palabra objetivo

- En http://bionlp-www.utu.fi/wv_demo/ se puede utilizar una demo que permite probar varios modelos de embeddings de word2vec, ver las palabras más similares, la similitud entre dos palabras y analogía entre ternas de palabras. (Probad las funcionalidades)

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model): FastText [Bojanowski et al. 2017]

- Es una extensión del modelo Word2Vec en el que cada palabra se trata como la suma de sus composiciones de caracteres n-gramas

- El vector para una palabra está compuesto por la suma de sus n-gramas. Por ejemplo el vector para la palabra "apple" estaría compuesto por la suma los vectores para los n-gramas "<ap, app, appl, apple>, ppl, pple, pple>, ple, ple>, le> …"

- El objetivo es obtener mejores representaciones para palabras poco frecuentes y para poder generar vectores para palabras que no se encuentran en el vocabulario de los word embeddings

Modelos de representación: semánticos vectoriales

Modelos semánticos vectoriales (Vector Semantic Model): modelos contextualizados Bert

- Las representaciones preentrenadas pueden ser independientes de contexto o contextualizadas y estas últimas unidireccionales o bidireccionales
- Los modelos independientes del contexto como word2vec o GloVe generan una única representación vectorial para una palabra ("banco" de entidad financiera y "banco" de sentarse tienen la misma representación)
- Los modelos contextualizados generan una representación de cada palabra que está basada en las otras palabras de la oración en la que aparece: unidireccional si solo tiene en cuenta las previas y bidireccional si tiene encuenta las previas y las siguientes
- Bert (https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html) en la frase "*I accessed the bank account*," representará "*bank*" teniendo en cuenta "*I accessed the*" y "*account*" porque es bidireccional; y en otra frase diferente podrá tener otra representacion

Índice

- Conceptos preliminares
- Modelos de representación
- **Funciones de pesado**
- Funciones de selección y reducción de rasgos
- Enlaces de interés
- Bibliografía

Funciones de pesado

- Las funciones de pesado (*term weighting functions*) constituyen funciones de proyección **F** dentro de la definición formal del modelo vectorial de representación de documentos

- Se pueden identificar 2 tipos:

  - **Funciones locales.** Aquellas que toman únicamente información del propio documento para obtener el peso de un rasgo en un documento

  - **Funciones globales.** Aquellas que toman información de la colección para obtener el peso de un rasgo en un documento. Para su cálculo suele ser necesario el uso de un fichero invertido con información relativa a las frecuencias de los rasgos en cada documento de la colección

Funciones de pesado

Ejemplos de **funciones locales**:

- **Binaria**:

$$F: \quad Bin(\vec{t_i}, \vec{d_j}) = \begin{cases} 1, & \text{si el rasgo } t_i \text{ aparece en } d_j \\ 0, & \text{si no aparece} \end{cases}$$

- **Frecuencia** (*term frequency*):

$$F: \quad TF(\vec{t_i}, \vec{d_j}) = f_{ij}, \ frecuencia\ del\ rasgo\ t_i\ en\ d_j$$

- **Frecuencia compensada** (*weighted term frequency*):

$$F: \quad WTF(\vec{t_i}, \vec{d_j}) = \frac{f_{ij}}{\sum_{t_p \in d_j} f_{pj}}$$

Funciones de pesado

Ejemplos de **funciones locales**:

- **Frecuencia Aumentada Normalizada** (Augmented Normalized Term Frequency) para compensar el sesgo que pueden introducir los documentos más largos. El denominador corresponde a la frecuencia máxima en el documento

$$F: \quad ANTF(\vec{t_i}, \vec{d_j}) = 0,5 + 0,5 \frac{f_{ij}}{max(\{f_{pj} \mid t_p \in d_j\})}$$

- **Normalización Logarítmica**:

$$F: \quad L(\vec{t_i}, \vec{d_j}) = log_2(f_{ij} + 1)$$

Funciones de pesado

Ejemplos de **funciones globales**:

- **Frecuencia inversa del documento**: es una medida de si el término es común o no, en la colección de documentos

$$IDF\ (\vec{t_i}, \vec{d_i}) = \log\left(\frac{N}{1 + df(ti)}\right)$$

donde $df(t_i)$ es el nº de documentos de la colección en los que aparece el rasgo t_i y N el nº documentos total, se suma 1 al denominador ya que podría de otra forma ser 0 si el término no está en la colección

- **Frecuencia del Término x Frecuencia inversa de Documento** (TF-IDF) (muy habitual en tareas de minería de textos) (ver video https://www.coursera.org/lecture/ml-clustering-and-retrieval/document-representation-nl267)

$$F: \quad TF - IDF(\vec{t_i}, \vec{d_j}) = f_{ij} \times \log\left(\frac{N}{df(\vec{t_i})}\right)$$

Funciones de pesado

Ejemplos de **funciones globales**:

- **Frecuencia del Término x Frecuencia inversa de Documento** (TF-IDF) (continuación)

$$F: \quad TF - IDF(\vec{t_i}, \vec{d_j}) = f_{ij} \times \log(\frac{N}{df(\vec{t_i})})$$

Un valor alto TF-IDF se alcanza con una elevada frecuencia del término (en el documento dado) y una pequeña frecuencia de ocurrencia del término en la colección completa de documentos

El cociente dentro de la función logaritmo del idf es siempre mayor o igual que 1, el valor de IDF y de TF-IDF es mayor o igual que 0. Cuando un término aparece en muchos documentos el valor de TF-IDF se acerca a 0

Funciones de pesado

Ejemplos de **funciones globales**:

- **Frecuencia inversa Probabilística:**

$$F: \quad PIF(\vec{t_i}, \vec{d_j}) = log(\frac{N - df(\vec{t_i})}{df(\vec{t_i})})$$

- **Función Normal:**

$$F: \quad N(\vec{t_i}, \vec{d_j}) = \sqrt{\frac{1}{\sum_{d_k \in C} f_{ij}^2}}$$

- **Frecuencia Global x Frecuencia Inversa de Documento** (GFIDF):

$$F: \quad GF - IDF(\vec{t_i}, \vec{d_j}) = \frac{gf(\vec{t_i})}{df(\vec{t_i})} \qquad \text{siendo} \qquad gf(\vec{t_i}) = \sum_{d_j \in C} f_{ij}$$

Índice

- Conceptos preliminares
- Modelos de representación
- Funciones de pesado
- **Funciones de selección y reducción de rasgos**
- Enlaces de interés
- Bibliografía

Funciones de selección y reducción de rasgos

- La **selección de rasgos** consiste en la transformación de un texto en el conjunto de rasgos que lo podrán representar

- Normalmente estas transformaciones básicas se encuadran en lo que se denomina preprocesamiento de los textos

- Tareas básicas de preprocesamiento son (ver tema 1):
  - Tokenización
  - Normalización léxica
  - Truncado (Stemming)
  - Lematización
  - Eliminación de palabras vacías (*stopwords)*

Funciones de selección y reducción de rasgos

- La **reducción de rasgos** permite disminuir la dimensión de las representaciones que puede llegar a ser muy alta para colecciones de gran tamaño

- Permite realizar una ponderación en base a la cual se pueden ordenar todos los rasgos de un vocabulario para seleccionar un subconjunto de ellos

- La selección se puede hacer estableciendo un umbral de ponderación mínima o bien prefijando una dimensión reducida, generando así un vocabulario que resultará ser un subconjunto del vocabulario inicial

- En muchas ocasiones las funciones de selección y reducción son funciones "orientadas a tarea", es decir, que se definen en función de la tarea posterior

Funciones de selección y reducción de rasgos

Las funciones presentadas en el apartado “funciones de pesado” pueden emplearse también como funciones de selección de rasgos:

- Con **funciones locales**, se seleccionan los rasgos con mayor peso **en cada documento** y se genera con ellos un vocabulario reducido

- Con **funciones globales**, se seleccionan los rasgos con más peso dentro de **la colección** y se genera con ellos un vocabulario reducido

Funciones de selección y reducción de rasgos

Selección de características/rasgos en aprendizaje automático:

Objetivos:

- Reducir el sobreebtranamiento (*overfitting*): menos datos redundantes se traduce en eliminar ruido
- Mejorar la precisión: con la eliminación de ruido
- Reducir el tiempo de entrenamiento: con la reducción de rasgos los algoritmos se entrenan más rápido

Algunas técnicas:

- Selección univariante
- Importancia de las características

Funciones de selección y reducción de rasgos

Selección de características/rasgos en aprendizaje automático: Selección univariante

- Se pueden utilizar pruebas estadísticas para seleccionar aquellas características que tienen la relación más fuerte con la variable de salida

- En la librería `scikit-learn`, en la clase `SelectKBest` hay pruebas estadísticas diferentes para seleccionar un número específico de características, por ejemplo Chi-cuadrado.

Funciones de selección y reducción de rasgos

- **Selección de características/rasgos en aprendizaje automático: Importancia de las características**

- Se puede obtener la relevancia de cada característica del conjunto de datos utilizando la propiedad de importancia de la característica del modelo

- La importancia de la característica da una puntuación para cada característica de los datos, cuanto mayor sea la puntuación, más importante o relevante será la característica para su variable de salida

- La clase `feature_importances` está en `ExtraTreesClassifier` en `scikit-learn` y permite extraer las `X` características más relevantes

Funciones de selección y reducción de rasgos

Funciones orientadas a tareas: clasificación

- **Ganancia de información** (*Information Gain*, IG): se usa para establecer la calidad de un determinado rasgo en una tarea de aprendizaje automático

Sea c_j $\forall j = 1 \ldots m$ un conjunto de categorías prefijadas, $P(c_j)$ representa la probabilidad a priori de una determinada clase c_j (en caso de que no sean equiprobables); $P(t_i)$ es la probabilidad a priori del rasgo t_i y $P(t_j)$ la probabilidad a priori de cualquier rasgo $\{t_j \mid t_i = t_i, \forall i, j\}$. Así la IG puede definirse como la siguiente función:

$$F: \quad IG(\vec{t_i}) = -\sum_{j=1}^{m} P(c_j) log P(c_j) + P(\vec{t}) \sum_{j=1}^{m} P(c_j|\vec{t_i}) log P(c_j|\vec{t_i}) +$$

$$+ P(\bar{\vec{t_i}}) \sum_{j=1}^{m} P(c_j|\bar{\vec{t_i}}) log P(c_j|\bar{\vec{t_i}})$$

Funciones de selección y reducción de rasgos

Funciones orientadas a tareas: clasificación

- **Información mutua** (*Mutual Information*, MI). Esta función se ha empleado habitualmente en el contexto del modelado estadístico del lenguaje, fundamentalmente para encontrar relaciones entre rasgos.

- La MI se puede estimar de dos formas:

  - como el valor medio sobre el conjunto total de clases $\mathbf{MI_{adv}(t_i)}$
  - como el valor máximo sobre el total de clases $\mathbf{MI_{max}(t_i)}$

Funciones de selección y reducción de rasgos

Funciones orientadas a tareas: clasificación

- **Información mutua** (*Mutual Information*, MI)

  Si se considera A como el número de documentos de una clase c_j en los que un rasgo t_i aparece; B como el número de documentos en los que no aparece; y C el número de documentos totales que pertenecen a la clase c_j; entonces, las funciones que expresan la MI "global" y "máxima" para un rasgo t son:

$$F: \quad MI_{adv}(\vec{t_i}) = \sum_{j=1\ldots m} P(c_j) I_{adv}(\vec{t_i}, c_j) \approx \sum_{j=1\ldots m} P(c_j) log \frac{A \times N}{(A+C) \times (A+B)}$$

$$F: \quad MI_{max}(\vec{t_i}) = max_{j=1\ldots m}\{I_{max}(\vec{t_i}, c_j)\} \approx max_{j=1\ldots m}\{log \frac{A \times N}{(A+C) \times (A+B)}\}$$

$$I(\vec{t_i}, c_j) = logP(\vec{t_i}|c_j) - logP(\vec{t_i})$$

Índice

- Conceptos preliminares
- Modelos de representación
- Funciones de pesado
- Funciones de selección y reducción de rasgos
- **Enlaces de interés**
- Bibliografía

Enlaces de interés

- **Gensim** es una librería Python que incluye implementaciones para el cálculo de los modelos de representación más habituales en Procesamiento de Lenguje Natural
  - Descripción de la libreria: https://en.wikipedia.org/wiki/Gensim
  - Web oficial: https://radimrehurek.com/gensim/
- **scikit-learn** es una librería Python para aprendizaje automático enfocado para tareas de clasificación, clustering, reducción de la dmensionalidad, etc.
  - Descripción de la libreria: https://en.wikipedia.org/wiki/Scikit-learn
  - Web oficial: https://scikit-learn.org/stable/
- Word2vec: implementación https://code.google.com/archive/p/word2vec/
- Word2vec: Embeddings preentrenados español https://crscardellino.github.io/SBWCE/
- FastText: embeddings preentrenados para 157 lenguas: https://fasttext.cc/docs/en/crawl-vectors.html
- Tutorial cómo generar embeddings con Python y Gensim https://machinelearningmastery.com/develop-word-embeddings-python-gensim/

Índice

- Conceptos preliminares
- Modelos de representación
- Funciones de pesado
- Funciones de selección y reducción de rasgos
- Enlaces de interés
- **Bibliografía**

Bibliografía

- [Bojanowski et al. 2017] Piotr Bojanowski, Edouard Grave, Amand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. https://arxiv.org/pdf/1607.04606.pdf
- [Devlin et al. 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/pdf/1810.04805.pdf
- [Hotho et al 2001] Hotho, A., Maedche, A., and Staab, S. Ontology based text clustering. In Proceedings of International Joint Conference on Artificial Intelligence, pp. 30 –37.
- [Hotho et al 2005] A. Hotho; A. Nürnberger; G. Paaß; S. Augustin. A Brief Survey of Text Mining. University of Kassel.
- [Joos 1950] Joos, M. Description of language design. JASA, 22, 701–708.
- [Jurafsky and Martin 2019] Dan Jurafsky and James H. Martin . Speech and Language Processing (3rd ed. draft). https://web.stanford.edu/~jurafsky/slp3/.
- [Mikolov et al. 2013] Mikolov, Tomas; et al. “Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781 [cs.CL].
- [Nareshkumar et al. 2017] Ksh. Nareshkumar Singh, H. Mamata Devi , Anjana Kakoti Mahanta Document representation techniques and their effect on the document Clustering and Classification: A Review. International Journal of Advanced Research in Computer Science. Vol 8(5), 2017.
- [Pennington et al. 2014] Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.
- [Salton 1975] Salton, G., Wong, A., y Yang, C. A vector space model for automatic indexing.Communications of the ACM, 18(11):613–620. 1975.