

# Clasificación de documentos

## Curso de procesamiento de lenguaje natural

Julio Weissman

Maestría en Tecnologías de la Información  
UNaM/UNEE

9 y 10 de agosto de 2018



# Problema de clasificación de documentos

- Una de las tareas más utilizadas de PLN
  - Análisis de polaridad
  - Determinación de tópicos
  - Detección de *spam*
  - Detección de autores
  - Identificación de idioma
- Se utilizan métodos de clasificación bien conocidos
- Transformación de la información:

$$\text{documento} \rightarrow (x_1, x_2, \dots, x_{nd}) \rightarrow \mathbb{R}^M$$

donde  $M$  es el número de características que se extraen del documento

# El método de la bolsa de palabras

## Document 1

The quick brown  
fox jumped over  
the lazy dog's  
back.

## Document 2

Now is the time  
for all good men  
to come to the  
aid of their party.

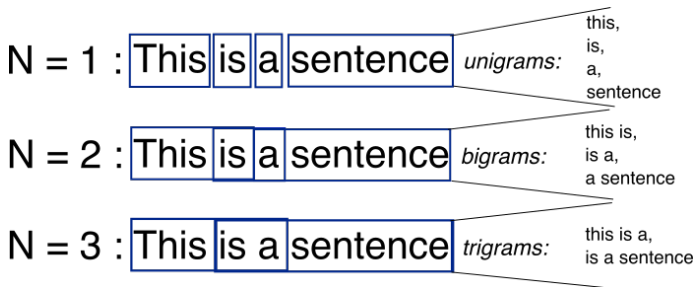
Term	Document 1	Document 2
aid	0	1
all	0	1
back	1	0
brown	1	0
come	0	1
dog	1	0
fox	1	0
good	0	1
jump	1	0
lazy	1	0
men	0	1
now	0	1
over	1	0
party	0	1
quick	1	0
their	0	1
time	0	1

## Stopword List

for
is
of
the
to

# Manteniendo algo del orden de las palabras

Se pueden agregar pares de tokens, tripletas, ...



Explosión de la dimensión de las características

# Explosión de características

- Si las características son muchas, entonces se pueden eliminar las que son muy frecuentes
- También se pueden eliminar las que aparecen muy poco
- Si las características que quedan son las de mediana frecuencia, mientras más aparezca el token en el documento más importancia tiene para éste (*frecuencia del término en el documento*)
- Pero mientras la palabra aparezca en más documentos diferentes (*frecuencia de documentos con el término*), menos representativo es el token respecto a un caso específico.

# TF-IDF

## TF (Term frequency)

$$tf(t, d) = n_{t,d}$$

donde  $n_{t,d}$  es la frecuencia que aparece el término  $t$  en el documento  $d$

## IDF (Inverse Document frequency)

$$idf(t, D) = \log \left( \frac{1 + n_D}{1 + df(D, t)} \right) + 1$$

donde  $n_D$  es el número de documentos y  $df(D, t)$  es el número de documentos en los que aparece  $t$

## TF-IDF

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

y se normalizan los valores para todos los documentos



Vamos a realizar la primera parte de la libreta

# Clasificación de documentos

- Se tiene una serie de documentos conocidos asignados a una categoría

$$\{(d^1, y^1), (d^2, y^2), \dots, (d^D, y^D)\}, \quad y \in \{C_1, \dots, C_K\} = Y$$

- Cada documento se puede representar como un vector de características

$$\phi(d^i) = x^i = (x_1^i, x_2^i, \dots, x_M^i), \quad x \in X$$

donde  $\phi(\cdot)$  puede ser BOW, TF-IDF, ...

## Objetivo de la clasificacion de documentos

Obtener una función  $h : X \rightarrow Y$  parametrizada por  $w = (w_0, \dots, w_P)$  tal que

$$\hat{y} = h(x; w)$$

de forma que  $\hat{y}$  sea **lo más plausible posible**



- Se basa en encontrar la probabilidad de una categoría, conociendo el texto, asumiendo una distribución de Bernoulli

$$\Pr(y = C_k | x; w) = \sigma(x^T w) = \frac{1}{1 + \exp(-x^T w)}$$

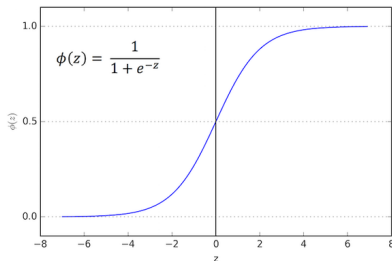
- Si solo hay dos categorías ( $C = \{C_0, C_1\}$ ) se selecciona la categoría con más probabilidad

$$\hat{y} = h(x; w) = C_1 \text{ si } \sigma(x^T w) \geq 0,5 \text{ si no } C_0$$

- Si hay más de dos categorías se utiliza la estrategia *One vs Rest* donde se estima un vector de parámetros  $w^k$  por cada categoría  $C_k$  (¿Qué significa esto?). Se selecciona la categoría con mayor probabilidad

$$\hat{y} = h(x; w^1, \dots, w^K) = C_{k^*} \text{ donde } k^* = \arg \max_{k \in \{1, \dots, K\}} \sigma(x^T w^k)$$

# Más sobre la regresión logística



El criterio de *optimización* es la **máxima verosimilitud con regularización** (¿Qué es esto?). Los vectores  $w^1, \dots, w^K$  los vamos a encontrar *minimizando* la función

$$\frac{1}{D} \sum_{i=1}^D \sum_{k=1}^K 1\{y^i = C_k\} \log \left( \sigma(x^T w^k) \right) + \frac{C}{K} \sum_{k=1}^K \|w^k\|$$



Vamos a hacer clasificación de sentimientos y de tópicos con regresión lineal

# ¿Que hacer para mejorar?

- Modificar la normalización de texto basada en conocimiento experto
- Agregar y/o cambiar la forma de lematizar o hacer el *stemming*
- Probar con otros métodos de clasificación (i.e. SVM)
- Cambia el método por aprendizaje profundo (o métodos similares, como *FastText*)

