Modelos de clasificación: Bayesiano ingenuo

Fernando Carranza
fernandocarranza86@gmail.com

Primer Cuatrimestre de 2025

- Unidad 1: Introducción
- **Unidad 2: Los algoritmos supervisados**
  - i) Aprendizaje supervisado y no supervisado.
  - ii) Métricas usuales para medir el rendimiento de modelos de clasificación (accuracy, precisión y cobertura).
  - iii) Anotación como tarea a resolver por un modelo predictivo.
  - iv) Datos estructurados y no estructurados: manejo de estructuras de almacenamiento de datos (json, csv).
  - v) Vectorización (conversión de un texto en tanto dato no estructurado en un arreglo numérico estructurado; CountVectorizer, TfidfVectorizer).
  - **vi) Modelos de clasificación: Bayesiano ingenuo, Regresión Logística, Máquina de soporte vectorial (Support Vector Machines).**
- Unidad 3: Anotación morfológica y de clase de palabra
- Unidad 4: Anotación sintáctica
- Unidad 5: Anotación para propósitos específicos

Presentación

Estructura y temas de esta clase:

1. Introducción
2. vi) Modelos de clasificación: Bayesiano ingenuo
3. Modelos de clasificación
4. Bayesiano Ingenuo
   - Nociones de probabilidad
   - Ejemplo no lingüístico de bayesiano ingenuo
   - Bayesiano ingenuo en una tarea de PLN
5. Recapitulación
6. Bibliografía

Bibliografía para la preparación de esta clase:

- Jurafsky y Martin. 2024. “Naive Bayes, Text Classification, and Sentiment”. *Speech and Language processing*. Draft.
- Tan, Michael Steinbach y Vipin Kumar. 2005. “Classification: Alternative Techniques”. *Data Mining*.Pearson.

- Los algoritmos de clasificación son algoritmos predictivos que toman una serie de datos y los asocian a una serie de categorías.
- Los algoritmos de clasificación pueden estar basados en reglas o recurrir a aprendizaje automático.
- De entre los que recurren a aprendizaje automático, existen los supervisados y los no supervisados. Los supervisados son aquellos que se entrenan a partir de datos anotados previamente que el modelo usa para descubrir patrones estadísticos. Los algoritmos no supervisados de clasificación o *clustering* elaboran la clasificación solo en función de la similitud o no entre los datos.

En lo que sigue nos vamos a concentrar en los supervisados.

De entre los tipos de clasificación, el que más concierne al procesamiento del lenguaje natural es la **clasificación de textos** (*text categorization*). Algunos ejemplos de clasificación de textos:

- **Análisis de sentimiento**: Extracción de la orientación positiva o negativa de un texto con respecto a un objeto o tema, clasificación que puede ser binaria o más compleja.
- **Detección de Spam**: Identificación de si un texto es o no spam, una clasificación binaria.
- **Language ID**: Identificación de la lengua de un texto
- **Atribución de autoría**: Identificación del autor de un texto.
- **Topic labeling**: Identificación del tema de un texto.
- **Intent classification**: Clasificación de intenciones.

También vamos a poder usar la clasificación para determinar algunas de las siguientes cuestiones:

- Cuál es la clase de palabra para un determinado token.
- Cuál es la función sintáctica de un determinado token con respecto a otro.
- A qué clase semántica responde determinado *chunk*.

Clasificación y modelos de lenguaje

Los modelos de lenguaje también pueden pensarse como algoritmos de clasificación. Cada palabra o sucesiones de palabras se pueden pensar como un documento y la siguiente como la clase. De este modo, dada una determinada palabra (documento) se predice la siguiente (la clase) y luego se toma la siguiente como el documento para la próxima clase y así sucesivamente.

- Existen dos grandes tipos de algoritmos de clasificación supervisados: los algoritmos generativos, como el bayesiano ingenuo, y los discriminativos, como la regresión logística.

- Los algoritmos de clasificación generativos construyen modelos de respecto de cómo una clase es capaz de generar o no los datos de input.
- Los algoritmos discriminativos aprenden qué rasgos del input son los más útiles para discriminar entre las diferentes clases.

- Para entrenar un algoritmo de clasificación, es necesario contar con datos estructurados. Por definición, un dato estructurado es uno con una sintaxis claramente definida. Es muy común que los datos estructurados aparezcan en documentos como json o csv, etc.

Ejemplo de json


```
{
  "tarea":"clasificacion de sentimiento",
  "datos":[
    {
      "texto":"Me parece una mierda",
      "sentimiento":0
    },{
      "texto":"me encanta",
      "sentimiento":0
    },
    {
      "texto":"sirve para lo que tiene que servir",
      "sentimiento":0.5
    },
    {
      "texto":"hace muy bien su tarea",
      "sentimiento":1
    }
  ]
}
```

Ejemplo de csv

```
"me parece una mierda",0
"me encanta",1
"sirve para lo que tiene que servir",0.5
"no sirve para nada",0
"hace muy bien su tarea",1
```

Visualización en forma de cuadro usando “,” como delimitador:

"me parece una mierda"	0
"me encanta"	1
"sirve para lo que tiene que servir"	0.5
"no sirve para nada"	0
"hace muy bien su tarea"	1

- Para entrenar un algoritmo supervisado, normalmente se divide el conjunto de datos (*data set*) en partes y se entrena con una (el *train set*) y se mide la capacidad predictiva con otra (el *test set*). Si no fuera así, no tendríamos cómo evaluar la efectividad del modelo, ya que evaluar un modelo con un dato que ya conoce sería hacer trampa.
- También se suele recurrir a un *validation set*, *development set* o *devset* para ajustar los hiperparámetros del modelo.

La validación cruzada (*cross validation*) consiste en dividir el *data set* en partes y entrenar sucesivamente modelos con todas las partes menos una hasta agotar todas las combinaciones posibles y validar cada uno de esos modelos con los datos que quedaron afuera.

- Existen distintas técnicas para entrenar los modelos estadísticos. Una librería en Python muy útil para indagar es sklearn (`https://scikit-learn.org/stable/index.html`)
- Si el algoritmo distribuye el conocimiento que adquiere en la etapa de entrenamiento en capas ocultas (*hidden layers*), se habla de aprendizaje profundo (*deep learning*).
- El aprendizaje profundo puede ser supervisado o no supervisado.

Algunas técnicas estadísticas típicas de clasificación incluyen al bayesiano ingenuo (*Naive Bayesian*), las *support vector machines* (SVM) y la regresión logística (*logistic regression*), entre otras.
La matemática de la regresión logística va a resultar sumamente útil para entender cómo es la matemática de las redes neuronales.

Bayesiano ingenuo

El bayesiano ingenuo es un clasificador estadístico que utiliza el teorema de Bayes como base.
El teorema bayesiano tiene la siguiente forma:

$$(1) \quad P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

- **Resultado**: Un posible valor que puede arrojar el fenómeno en cuestión.
- **Espacio de muestra**: El conjunto total de todos los resultados que puede arrojar el fenómeno. En las fórmulas, se suele referir mediante la variable Ω
- **Evento**: Un subconjunto de los resultados que puede arrojar el fenómeno.

Algunos axiomas:

- **No negatividad**: La probabilidad de un elemento no puede ser un número negativo.
- **Normalización**: La sumatoria de la probabilidad de todos los resultados que conforman el espacio de muestra tiene que ser igual a 1.

Probabilidad

(2) P(Y) es la posibilidad de encontrarnos con el resultado Y.

Supongamos que queremos determinar cuál es la probabilidad de que en esta clase un estudiante tenga una computadora. ¿Cómo hacemos?

La probabilidad se puede obtener mediante estadística contando. La probabilidad de que se dé un resultado o un evento y se calcula contando la cantidad de veces que se da el resultado y dividida la cantidad de instancias totales de posibles resultados dentro del espacio de muestra Ω.

(3) $P(Y|X)$ es la probabilidad de encontrarnos con el resultado X dado el resultado Y.

Supongamos que queremos determinar cuál es es la porobabilidad de que alguien que trajo computadora haya traído mate. ¿Cómo hacemos?

La probabilidad condicionada se puede obtener mediante estadística contando. La probabilidad de que se dé un resultado o un evento y dado un resultado o evento x se calcula contando la cantidad de veces que se da el resultado o evento y dividida la cantidad de instancias del resultado o evento x.

Un bayesiano ingenuo es un algoritmo generativo que caracteriza clases en función de los datos, vectorizados en función de una serie de rasgos que los caracterizan.

id	A	B	C
Tid	home owner	Marital status	Defaulted Borrower
1	Yes	Single	No
2	No	Married	No
3	No	Single	No
4	Yes	Married	No
5	No	Divorced	Yes
6	No	Married	No
7	Yes	Divorced	No
8	No	Single	Yes
9	No	Married	No
10	No	Single	Yes

Cuadro: Tomado de Tan *et al.* (2006)

- Vamos a usar C como variable para el conjunto de las clases, c como variable de clase, X como variable para el dato vectorizado y x para cada valor en el vector.
- La tarea de entrenamiento consiste en calcular la probabilidad P(Yes|X) y P(No|X) dado el corpus de entrenamiento.
- Es ingenuo porque asume que la presencia de cada atributo en una clase dada es independiente del resto de los atributos. Esto permite que la probabilidad no se compute directamente con B (con todo el vector, lo cual requeriría muchos datos), sino con cada uno de los valores de B. Esa "ingenuidad" es la que permite multiplicar las probabilidades posteriores entre sí por cada valor de X.
- Dado que la P(X) se mantiene fija para todas las clases, se la elimina de la cuenta

(4)
$$\hat{c} = \arg\max_{c \in C} P(c) \prod_{x \in X} P(x|c)$$

id	A	B	C
Tid	home owner	Marital status	Defaulted Borrower
1	Yes	Single	No
2	No	Married	No
3	No	Single	No
4	Yes	Married	No
5	No	Divorced	Yes
6	No	Married	No
7	Yes	Divorced	No
8	No	Single	Yes
9	No	Married	No
10	No	Single	Yes

$P(A{=}Yes|No) = 3/7$
$P(A{=}No|No) = 4/7$
$P(A{=}Yes|Yes) = 0$
$P(A{=}No|Yes) = 1$
$P(B{=}Single|No) = 2/7$
$P(B{=}Divorced|No) = 1/7$
$P(B{=}Married|No) = 4/7$
$P(B{=}Single|Yes) = 2/3$
$P(B{=}Divorced|Yes) = 1/3$
$P(B{=}Married|Yes) = 0$
$P(Yes) = 0.3$
$P(No) = 0.7$

Supongamos que queremos predecir si va a ser Yes o No el siguiente deudor:

id	A	B	C
Tid	home owner	Marital status	Defaulted Borrower
11	No	Married	?

Para eso tenemos que completar la fórmula con los datos que sacamos de la tabla:

(5) a. $P(X|No) = P(A{=}No|No) \times P(B{=}Married|No) = \frac{4}{7} \times \frac{4}{7} = \frac{16}{49} = 0.3265$

b. $P(X|Yes) = P(A{=}No|Yes) \times P(B{=}Married|Yes) = 1 \times 0 = 0$

Dado que $P(X|No) > P(X|Yes)$, se predice que 11 no va a ser un *defaulted borrower*.

Para aplicar esto a una tarea de procesamiento del lenguaje natural, tenemos que vectorizar los datos lingüísticos con los que queramos trabajar. Por ejemplo, para una tarea de análisis de sentimiento, una forma es asumir que cada feature es la frecuencia de las palabras en un modelo de bolsa de palabras (BOW).

d1= I love this movie! It's sweet, but with satirical humor. The dialogue is great...

$$d1 = \begin{array}{cc} \text{it} & 6 \\ \text{I} & 5 \\ \text{the} & 4 \\ \text{to} & 3 \\ \text{and} & 3 \\ \text{seen} & 2 \\ \text{yet} & 1 \\ \ldots & \ldots \end{array}$$

Los documentos se pueden representar entonces como una matriz de documento-término:

doc/words	it	I	the	to	and	yet
d1	2	1	2	1	1	1
d2	1	1	0	0	2	0
d3	3	4	1	0	0	1

A esta matriz se le puede aplicar, por ejemplo, tf-idf

TF-IDF

tf-idf para de las siguientes ideas:

- Parte de la idea de *Bag of words* (BOW)
- Ajusta la importancia de palabras comunes en el corpus
- Reduce el peso de las palabras muy frecuentes (como *stopwords*), a la vez que aumenta la importancia de las menos frecuentes (distintivas para un documento).

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \cdot \log\left(\frac{N}{\text{df}(t)}\right)$$

donde:

- $\text{tf}(t, d)$: Frecuencia de término t en el documento d
- N: Número total de documentos
- $\text{df}(t)$: Número de documentos que contienen el término t

A modo de ejemplo, supongamos que obtenemos una matriz como esta:

doc/words	it	I	the	to	and	yet
d1	0.4	0.8	0.9	0.8	0.7	0.8
d2	0.2	0.6	0.8	0	0.1	0
d3	0.1	0.5	1	0	0	0.1

Se aplica entonces el bayesiano ingenuo sobre la matriz que obtenemos como resultado

En esta clase introdujimos cómo funciona el bayesiano ingenuo y cómo se lo puede aplicar a un problema lingüístico.

Bibliografía I

Tan, P.-N., Steinbach, M., y Kumar, V. (2006). *Introduction to data mining*. Pearson, Boston.