Modelo para procesamiento de lenguage natural?

Victor Hugo Matus Maldonado

20 de marzo de 2020

Resumen ejecutivo

Resumen ejecutivo.

1. Marco teórico y estado del arte

1.1. Bases de datos y álgebra relacional

El *modelo relacional de base de datos*, consiste en cinco componentes:

1. Una colección de tipos escalares, pueden ser definidos por el sistema o por el usuario.
2. Un generador de tipos de relaciones y un intérprete para las relaciones mismas.
3. Estructuras para definir variables relacionales de los tipos generados.
4. Un operador para asignar valores de relación a dichas variables.
5. Una colección relacionalmente completa para obtener valores relacionales de otros valores relacionales mediante operadores.

Las operaciones del modelo relacional están cimientadas en el *álgebra relacional*. Utilizando operaciones primitivas del álgebra se producen nuevas relaciones que pueden manipularse también por medio de operaciones del álgebra mismo. Una secuencia de operaciones de álgebra relacional forma una expresión cuyo resultado es una relación que representa el resultado de una consulta de base de datos. Estas operaciones se pueden clasificar en dos grupos, operaciones de la teoría de conjuntos: *UNIÓN*, *INTERSECCIÓN*, *DIFERENCIA* y *PRODUCTO CARTESIANO* (*PRODUCTO CRUZADO*), y el otro grupo consiste en operaciones específicas para bases de datos relacionales: *JUNTAR*, *SELECCIONAR* y *PROYECTAR*.

1.2. AI, ML, NLP

> *Inteligencia artificial* (IA) es el esfuerzo por automatizar tareas intelectuales normalmente realizadas por humanos(Chollet, 2018)

de este campo general se desprenden el *aprendizaje maquinal* (AM) y *aprendizaje profundo* (AP).

Mitchell (1997) define aprendizaje maquinal como

> un programa de computadora aprende de experiencia E con respecto a una tarea T y una medición de rendimiento P, si su rendimiento en T, medido por P, mejora con E

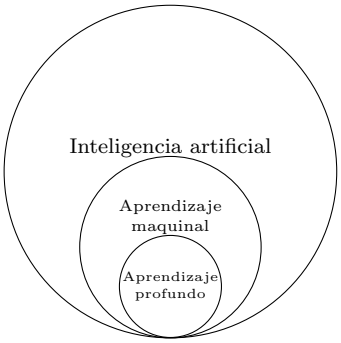


Figura 1: AP⊂AM⊂IA

es decir, a diferencia del paradigma clásico de programación donde los humanos introducen datos y órdenes para procesarlos, un sistema de aprendizaje maquinal no se programada explícitamente, se introducen muchos ejemplos relevantes a una tarea (datos y respuestas esperadas) con los que es entrenado y si encuentra una estructura estadística en ellos, genera una regla para automatizar la tarea.

El *procesamiento de lenguaje natural* (PLN), es el conjunto de métodos para hacer accesible el lenguaje humano a las computadoras(Eisenstein, 2019). Toma conocimientos de muchas tradiciones intelectuales, como lingüística y teoría formal del lenguaje, autómatas y otras áreas computación, inteligencia artificial, aprendizaje maquinal y profundo, estadística, teoría de la información, fonética y fonología, estas dos últimas áreas son de particular utilidad para procesamiento de voz. Existen dos posturas opuestas sobre lo que la tarea principal del PLN debe ser:

- Entrenar sistemas de principio a fin para que transmuten texto sin procesar en cualquier estructura deseada.
- Transformar texto en una pila de estructuras lingüísticas de uso general que en teoría deben poder soportar cualquier aplicación.

En la actualidad no hay consenso y ambos tipos de sistemas se consideran viables, este proyecto se basará en el segundo paradigma.

Por lo general el PLN divide sus funciones en módulos para facilitar la reutilización de algoritmos genéricos en diversas tareas y modelos, dos de los módulos básicos son *búsqueda* y *aprendizaje* con los que se puede resolver muchos problemas que tienen la forma matemática

$$\hat{y} = \underset{y \in Y(x)}{argmax} \Psi(x, y; 0), \tag{1}$$

donde,

- x es la entrada, un elemento de un conjunto X.
- y es el resultado, un elemento de un conjunto Y.
- Ψ es una función de puntuación (también conocida como *modelo*), que va desde el conjunto $X \times Y$ hasta los números reales.
- $\emptyset$ es el vector de parámetros para Ψ.

- $\hat{y}$ es el resultado previsto, que es elegido para maximizar la función de puntuación.

El módulo de búsqueda se encarga de computar el $argmax$ de la función Ψ, es decir, encuentra el resultado $\hat{y}$ con la mejor puntuación con respecto a la entrada x. El módulo de aprendizaje encuentra los parámetros θ por medio del procesamiento de grandes conjuntos de datos de ejemplos etiquetados $\{(x^i, y^i)\}_{i=1}^N$.

Un método lineal de clasificación de texto común es la *bolsa de palabras*, comienza por asignar etiquetas $y \in Y$ donde Y son todas las posibles etiquetas. Se utilizan vectores columna y la fórmula 1 y puede modelarse con diversas distribuciones (figura 2).

Para muchas tareas, las características léxicas (palabras) pierden sentido en aislamiento, por lo que históricamente el PLN se ha enfocado en la clasificación lineal, recientemente algunas tareas pueden resolverse con clasificadores no lineales, es decir, por medio de redes neuronales (aprendizaje profundo).

El aprendizaje profundo sigue el paradigma del aprendizaje maquinal, sus entradas son datos y ejemplos de resultados esperados y se mide si el algoritmo está haciendo un buen trabajo mientras *aprende*. A diferencia del aprendizaje maquinal, se hace énfasis en aprender en capas sucesivas de representaciones cada vez más sucesivas. "Profundidad" entonces refiere a cuántas capas contribuyen a un modelo.

1.3. Mate

Una *variable aleatoria* (v.a.) es una función real $X : \Omega \mapsto \mathbb{R}$ tal que el conjunto $\{\omega \in \Omega : X(\omega) \in I\}$ es un evento de Ω para cada $I \subset \mathbb{R}$, en un espacio Ω hipotético. Se le considera *variable aleatoria discreta* (v.a.d.) cuando su rango de valores R_x es finito o contablemente infinito, mientras que una *variable aleatoria continua* (v.a.c.) puede tomar cualquier valor real en un intervalo.

> La forma más natural de expresar la distribución de v.a.d.s es la *función de probabilidad*(Blitzstein y Hwang, 2019).

Una v.a.d. X con $R_x = \{x_1, x_2x_3, \ldots, x_n, \ldots\}$ tiene una función de distribución

$$\begin{aligned} f(x) = 0 \text{ para cada } x \notin R_x; \\ f(x) = P(X = x) \text{ para } x \in R_x \end{aligned} \tag{2}$$

para una v.a.c. X será una función no negativa real $f : \mathbb{R} \mapsto [0, \infty)$, es decir

$$P(X \in A) = \int_A f(x)dx \tag{3}$$

El *valor esperado* de una v.a.d. X con una función de probabilidad (2) es definida como

$$\mu = E(X) = \sum_{x \in R_x}^{\infty} x f(x), \tag{4}$$

siempre y cuando la serie converja absolutamente y es también llamado *media* de X, utilizada, similar a la media aritmética en estadísticas, para obtener el valor promedio entre observaciones.

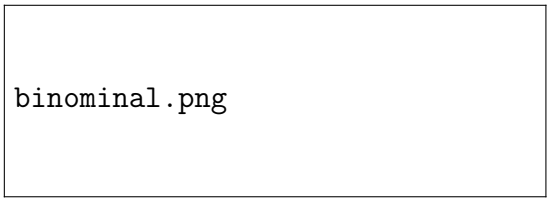
Para una v.a.c. X se define como

$$\mu = E(X) = \int_{-\infty}^{\infty} x f(x) dx \tag{5}$$

Para conocer la variabilidad de la distribución de cualquier v.a se utiliza la *varianza*, para X se define

$$\sigma^2 = Var(X) = [(X - \mu)^2] \tag{6}$$

Figura 2: Existe diversidad de distribuciones para modelar v.a.s, a continuación se muestran las mas importantes de acuerdo a Balakrishnan, Koutras y Politis (2020). De las siguientes son aptas para v.a.d.s hasta la distribución Normal, mientras que las v.a.c.s pueden ser modeladas desde la distribución Uniforme.



Binominal $b(n, p)$. Número de éxitos en n ensayos de Bernoulli independientes con la misma probabilidad de éxito p.

$$\begin{gathered} f(x = k) = \binom{n}{x} p^x p^{n-x}, \\ x = 0, 1, \ldots, n; \\ E(X) = np,\ Var(X) = npq \end{gathered} \tag{7}$$

Geométrica $G(p)$. Número n de ensayos de Bernoulli independientes con la misma probabilidad de éxito p, hasta obtener el primer éxito.

$$\begin{gathered} f(x) = q^{x-1}, \\ x = 1, 2, \ldots, n; \\ E(X) = \frac{1}{p},\ Var(X) = \frac{q}{p^2} \end{gathered} \tag{8}$$

negativabinominal.png

Negativa binominal $Nb(rp)$. Número n de ensayos de Bernoulli independientes con la misma probabilidad de éxito p, hasta obtener resultado número r.

$$\begin{gathered} f(x) = \binom{x-1}{r-1} p^r q^{x-r}, \\ r = r, r+1, r+2, \ldots, n; \\ E(X) = \frac{r}{p},\ Var(X) = \frac{rq}{p^2} \end{gathered} \tag{9}$$

hyperg.png

Hipergeométrica $h(n; a, b)$. Muestra aleatoria tamaño n de elementos tipo a no reemplazada de un universo con elementos tipo a y b.

$$\begin{gathered} f(x) = P(X = x) = \frac{\binom{a}{x}\binom{b}{n-x}}{\binom{a+b}{n}}; \\ E(X) = n \cdot \frac{a}{a+b}, \\ Var(X) = n \cdot \frac{a}{a+b} \cdot \frac{b}{a+b} \cdot \left(1 - \frac{n-1}{a+b-1}\right) \end{gathered} \tag{10}$$

poisson.png

Poisson $\mathcal{P}(\lambda)$. Puede utilizarse con algún parámetro λ cuando la probabilidad de éxito tiende a cero ($p \to 0$) de forma que la media $E(X) = np$ converge en algún $\lambda > 0$.

$$\begin{gathered} f(x) = e^{-\lambda} \frac{\lambda^x}{x!}, \\ x = 0, 1, 2, \ldots; \\ E(X) = \lambda, Var(X) = \lambda \end{gathered} \tag{11}$$

uniforme.png

Uniforme $U[a, b]$. Útil para modelar situaciones en que todos los intervalos tengan la misma amplitud y probabilidad.

$$\begin{gathered} f(x) \begin{cases} \frac{1}{b-a}, & a \le x \le b, \\ \text{de otro modo}, & 0; \end{cases} \\ F(t) \begin{cases} 0, & t < 0, \\ \frac{t-a}{b-a}, & a \le t \le b, \\ 1, & t > b; \end{cases} \\ E(X) = \frac{a+b}{2}, \quad Var(X) = \frac{(b-a)^2}{12} \end{gathered} \tag{12}$$

normal.png

Normal $N(\mu, \sigma^2)$. La suma y el promedio de un gran número de observaciones para una variable X puede ser aproximada por la distribución normal, independientemente de su distribución original.

$$\begin{gathered} f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}, \\ -\infty < z < \infty; \\ E(X) = \mu,\ Var(X) = \sigma^2 \end{gathered} \tag{13}$$

epsilon_lambda.png

Exponencial $\epsilon(\lambda)$. Es considerada la análoga continua de la distribución geométrica y puede ser utilizada para modelar parte de la vida de un sujeto X.

$$\begin{gathered} f(x) = \begin{cases} \lambda^{-\lambda x},\ x \le 0, \\ 0,\ x < 0; \end{cases} \\ F(t) = \begin{cases} 0,\ t < 0, \\ 1 - e^{-\lambda t},\ x \ge 0; \end{cases} \\ E(X) = \frac{1}{\lambda},\ Var(X) = \frac{1}{\lambda^2} \end{gathered} \tag{14}$$

gama.png

Gama. Generalización de la distribución *Erlang* cuando el parámetro n no es necesariamente un entero.

$$\begin{gathered} f(x) = \begin{cases} \frac{\lambda^a}{\Gamma(a)} x^{a-1} e^{-\lambda x},\ x \ge 0, \\ 0,\ x < 0, \end{cases} \\ \text{donde } \Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt \\ \text{es la función gama.} \\ E(X) = \frac{a}{\lambda},\ Var(X) = \frac{a}{\lambda^2} \end{gathered} \tag{15}$$

beta.png

Beta. Ofrece modelos satisfactorios para v.a.c.s que toman valores entre dos puntos conocidos.

$$\begin{gathered} f(x) = \begin{cases} \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1},\ 0 < x < 1, \\ \text{de otro modo, } 0, \end{cases} \\ \text{donde } B(\alpha,\beta) = \\ \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx = \\ \big(\Gamma(\alpha)\Gamma(\beta)\big) / \big(\Gamma(\alpha+\beta)\big) \\ \text{es la función beta.} \\ E(X) = \frac{a}{\lambda},\ Var(X) = \frac{a}{\lambda^2} \end{gathered} \tag{16}$$

2. Objetivos

2.1. General

Haciendo uso de las ciencias de la computación, las herramientas matemáticas de estadística y métodos de aprendizaje autónomo, se busca obtener información cuantitativa de textos provenientes de redes sociales, cadenas noticiosas y audio de programas de capacitación, para inferir posiciones, tendencias, comportamientos o razones de grupos sociales, considerando un ciclo de clasificación, estimación, detección y comprobación.

2.2. Particulares

1. Desarrollar un modelo de base de datos que permita la captura de categorías para un determinado problema, los elementos de identificación de cada categoría, el origen de la información y su correlación.

2. Construir una estructura de datos que capte la estimación o valores esperados para el procesamiento de textos.

3. Elaborar un sistema de objetos para el soporte de los elementos de aprendizaje autónomo.

4. Generar los elementos de captura de textos para su almacenamiento y procesamiento.

5. Elaborar un modelo estadístico que permita comprobar las estimaciones a partir de los datos y en consecuencia realizar un ajuste en los parámetros usados para el aprendizaje autónomo.

6. Producir los reportes con un análisis estadístico que faciliten la interpretación de resultados y den pauta para la obtención del conocimiento de interés.

2.3. Metas científicas

Metas científicas

3. Metodología científica

4. Grupo de trabajo

- Dr. José Emilio Quiroz Ibarra
  Universidad Iberoamericana, Dirección.
- Dra. Alma Rocío Sagaceta Mejía
  Universidad Autónoma Metropolitana, Codirección.
- Mtra. Paloma Alejandra Vilchis León
  Universidad Tecnológica de México*, Tutoría.

↑ preguntar

5. Infraestructura disponible para el proyecto

Laboratorios y equipos disponibles en la Universidad Iberoamericana Ciudad de México. servidor desktop

6. Cronograma de actividades

cronograma

7. Resultados comprometidos

Publicación

8. Visto bueno

Vo.Bo.

9. Referencias

Balakrishnan, N., Markos V. Koutras y Konstantinos G. Politis (2020). *Introduction to probability: models and applications [Introducción a la probabilidad: modelos y aplicaciones]*. Hoboken: John Wiley & Sons, Inc. ISBN: 9781118123348.

Blitzstein, Joseph K. y Jessica Hwang (2019). *Introduction to Probability [Introducción a la probabilidad]*. 2.ª ed. Texts in Statistical Science [Textos en ciencia estadistica]. Florida: CRC Press. ISBN: 9781138369917.

Chollet, François (2018). *Machine Learning With Python [Machine Learning]*. New York: Manning Publications Co. ISBN: 9781617294433.

Date, C. J. (2012). *SQL and Relational Theory: How to Write Accurate SQL Code [SQL y teoria relacional: Cómo escribir código SQL correcto]*. 2.ª ed. California: 0'Reilly Media. ISBN: 9781449316402.

Eisenstein, Jacob (2019). *Introduction to natural language processing [Introducción al procesamiento de lenguaje natural]*. Adaptive computation and machine learning [Computación adaptativa y aprendizaje maquinal]. Cambridge: MIT Press. ISBN: 9780262042840.

Mitchell, Tom (1997). *Machine learning [Aprendizaje maquinal]*. New York: McGraw-Hill. ISBN: 0070428077.