Las mil secciones son para no perderme durante la escritura

Índice

Las secciones son para no perderme durante la escritura

1. Marco teórico

1.1. Modelo relacional de base de datos

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

1. Una colección de tipos escalares, pueden ser definidos por el sistema (INTEGER, CHAR, BOOLEAN, etc.) 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 relacionales genéricos.

Aunque tersa esta lista es útil para delimitar lo que el modelo relacional es y no es. Se debe comenzar definiendo los *tipos*, ya que las relaciones se definen sobre ellos, son "en esencia un conjunto finito de valores nombrados—todos los valores posibles de alguna categoría específica(...)"[Dat12].

Los *atributos* son pares ordenados de combinaciones atributo-nombre/tipo-nombre y una *tupla* es un par ordenado de atributos. El modelo relacional también soporta varios tipos de *llaves*, que poseen las propiedades de unicidad, ninguna contiene dos tuplas distintas con el mismo valor e irreductibilidad, ningún subconjunto suyo es tiene unicidad.

	código	fecha	estado
Tuplas	MX01	29-07-99	1
	MX02	30-07-99	1
	MX03	31-07-99	0

Cuadro 1: Representación de una tabla de una base en datos, la fila superior muestra tres atributos distintos y *cada una* de las filas siguientes es una tupla.

La *restricción de integridad* (*constraint*) es una expresión booleana que debe evaluarse como verdadera. Los *constraints de tipo* definen los valores que constituyen un

tipo dado y los *constraints de base de datos* limitan los valores que pueden aparecer en cierta base de datos. Las bases de datos suelen tener múltiples constraints específicos, expresados en términos de sus relaciones, sin embargo, el modelo relacional incluye dos constraints genéricos, que aplican a cada base de datos:

- Regla de integridad de identidad: Las *llaves primarias* (PK) deben ser no nulas.
- Regla de integridad de referencia: Las *llaves foráneas* (FK) deben tener relación (si B referencia a A, A debe existir).

Las operaciones del modelo relacional están cimientadas en el *álgebra relacional*, las operaciones primitivas del álgebra producen nuevas relaciones, que pueden manipularse también por medio de operaciones del álgebra mismo. Una secuencia de operaciones de álgebra relacional forma una *expresión de álgebra relacional* cuyo resultado es una relación que representa el resultado de una consulta (o solicitud de consulta) de base de datos.[EN11] Estas operaciones pueden clasificar en dos grupos, operaciones de conjuntos* de la teoría de conjuntos matemáticos refiere a la operaciones *UNION*, *INTERSECTION*, *SET DIFFERENCE* (*MINUS*) y *CARTESIAN PRODUCT* (*CROSS PRODUCT*). El otro grupo consiste en operaciones específicas para bases de datos relacionales: *JOIN*, *SELECT* y *PROJECT*. Estas dos últimas, por operar con una sola relación, son también conocidas como *operaciones unarias*.

SELECT elige un subconjunto de tuplas de una relación que satisfacen la condición de selección

$$\sigma_{<\text{condición de selección}>}(\text{R}) \tag{1}$$

donde σ denota el operador SELECT, y la condición de selección es una expresión booleana especificada en los atributos de la relación R.

PROJECT produce una nueva relación de atributos y tuplas* duplicadas

$$\pi_{<\text{lista de atributos}>}(\text{R}) \tag{2}$$

donde π es el denota el operador PROJECT y la lista de atributos es la sublista de atributos deseados de la relación R.

Las operaciones con dos relaciones reciben el nombre de *operaciones binarias*. Si las relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_n)$ son compatibles de unión (tienen el mismo grado n y $dom(A_i) = dom(B_i)$ para $1 \leq i \leq n$) podemos usarlas para definir las siguientes operaciones binarias:

- UNION: El resultado de esta operación se denota $R \cup S$, es una relación que incluye todas las tuplas que están en R, S o ambos, se eliminan duplicados.

↑ se explican hasta modelo rel, en los libros se explica primero

↓ (así se llaman? checar libro en l'espagnol) "set operations"

- INTERSECTION: El resultado de esta operación se denota $R \cap S$, es una relación que incluye todas las tuplas que están en ambos R y S.
- SET DIFFERENCE: El resultado de esta operación se denota $R - S$, es una relación que incluye todas las tuplas que están en ambos R pero no en S.

El operador CARTESIAN PRODUCT, denotado $R \times S$ es la operación binaria que no requiere compatibilidad de unión y produce un nuevo elemento al combinar cada miembro (tupla) de cada relación conjunto) con cada otro miembro de la otra relación. El resultado de $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_m)$ es una relación Q con atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$ (en ese orden) de grado $n + m$. El resultado Q tiene una tupla por cada combinación de tuplas de R y S.

JOIN $\bowtie$, es el operador utilizado para combinar tuplas relacionadas de dos relaciones en una sola tupla. Pertinente para procesar relaciones entre relaciones. Si tenemos dos relaciones $R(A_1, A_2, \ldots, A_n)$ y $S(B_1, B_2, \ldots, B_m)$ podemos escribir la operación JOIN como

$$R \bowtie_{<\text{condiciones de unión}>} S \tag{3}$$

el resultado de la unión es la relación Q con atributos $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$ (en ese orden). El resultado Q tiene una tupla por cada combinación de tuplas de R y S que satisfacen las condiciones de unión.

1.2. 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*[BH19]. Una v.a.d. X con $R_x = \{x_1, x_2 x_3, \ldots, x_n, \ldots\}$ tiene una función de distribución

$$\begin{aligned} f(x) &= 0 \text{ para cada } x \notin R_x; \\ f(x) &= P(X = x) \text{ para } x \in R_x \end{aligned} \tag{4}$$

El *valor esperado* de una v.a.d. X con una función de probabilidad (4) es definida como

$$\mu = E(X) = \sum_{x \in R_x}^{\infty} x f(x), \tag{5}$$

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.

Continuando términos estadísticos, la *varianza* de una v.a.d. X indica la variabilidad de su distribución

$$\begin{gathered} \epsilon^2 = Var(X) = E[(X-\mu)^2], \text{ ó} \\ Var(X) = E(X^2) - [E(X)]^2 \end{gathered} \tag{6}$$

##
cosiderar si necesito covar
##

La covarianza mide qué tanto o tan poco las dos variables aleatorias cuyos valores esperados existen y son positivos tienen dependencia lineal, denotada $cov(X,Y)$ la covarianza de X y Y es definida como

$$cov(X,Y) = E[(X-EX)(Y-EY)] \tag{7}$$

Cuando no se tiene una referencia para usar la covarianza, tiene sentido escalarla de acuerdo a la desviación estándar de las variables[Mat17]

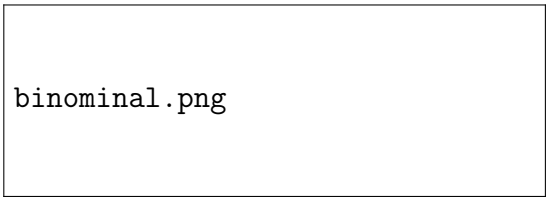
$$p(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)}\sqrt{var(Y)}} \tag{8}$$

denotada $corr(X,Y)$ de esta es la *correlación* de X y Y. Un coeficiente de relación $p = 0$ indica que no hay relación.

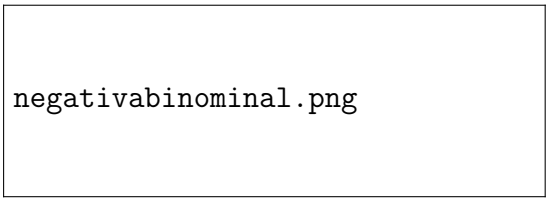
##

##

(a) Geométrica



(b) Binominal

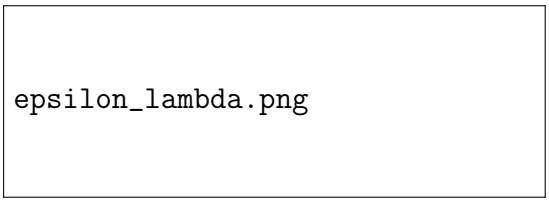


(c) Negativa binominal

(d) Hipergeométrica

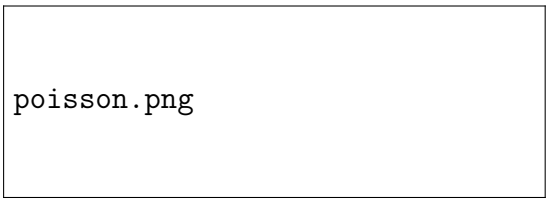
(e) Γ

(f) Beta

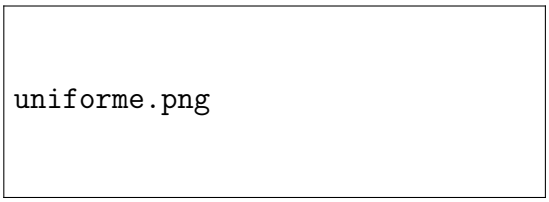


(g) $\epsilon(\lambda)$

(h) Normal



(i) Poisson



(j) Uniforme

[BKP20]

1.3. distribuciones

1.3.1. Distribución de Bernoulli y binominal

Una variable aleatoria tiene la *distribución de Bernoulli* con un parámetro p si $P(X = 1) = p$ y $P(X = 0 = 1 - p)$, cuando $0 < p < 1$. Se escribe como $X \sim \boldsymbol{Bern}(p)$, el símbolo $\sim$ significa "distribuido como" y la probabilidad p es el *parámetro*, que determina qué distribución de Bernoulli específica tenemos.

Supóngase que se realizan n ensayos Bernoulli independientes, cada uno con probabilidad p de éxito. X sea el número de éxitos, la distribución X se llama *distribución binominal* con parámetros n y p; se escribe $X \sim \boldsymbol{Bin}(p, n)$. $\boldsymbol{Bern}(p)$ es la misma distribución que $\boldsymbol{Bin}(1, p)$. Bernoulli es un caso especial de binominal, si $x \sim \boldsymbol{Bin}(1, p)$, entonces la función de probabilidad de X es

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{9}$$

para $k = 0, 1, \ldots, n$ (y por otra parte $P(X = k) = 0$).

1.3.2. Distribución de hipergeométrica

Si $X \sim \boldsymbol{HGeom}(w, b, n)$, entonces la función de probabilidad de X es

$$P(X = k) = \frac{\binom{w}{k}\binom{b}{n-k}}{\binom{w+b}{n}}, \tag{10}$$

para enteros k satisfaciendo $0 \leq k \leq w$ y $0 \leq n-k \leq b$, y $P(X = k) = 0$. La estructura esencial de la distribución hipergeométrica se basa en que objetos en su población están clasificados usando dos tipos de etiquetas, al menos una de estas siendo asignada al azar. Las distribuciones $\boldsymbol{HGeom}(w, b, n)$ y $\boldsymbol{HGeom}(n, w+b-n, 1)$ son idénticas si X y Y tienen la misma distribución, podemos demostrarlo algebraicamente:

$$P(X = k) = \frac{\binom{w}{k}\binom{b}{n-k}}{\binom{w+b}{n}} = \frac{w!b!n!(w + b - n)!}{k!(w + b)!(w - k)!(n - k)!(b - n + k)!} \tag{11}$$

$$P(X = k) = \frac{\binom{n}{k}\binom{w+b-n}{w-k}}{\binom{w+b}{w}} = \frac{w!b!n!(w + b - n)!}{k!(w + b)!(w - k)!(n - k)!(b - n + k)!}. \tag{12}$$

1.3.3. Distribución uniforme discreta

Teniendo C, un conjunto finito no vacío de números, se elige un número uniformemente al azar (o sea que todos los números tienen la misma posibilidad de ser elegidos), llámese X. Entonces se dice que X una *distribución uniforme discreta* con el parámetro C. Se dice entonces que la función de probabilidad de $X \sim DUNif(C)$ (la distribución uniforme discreta de X) es

$$P(X = x) = \frac{1}{|C|} \tag{13}$$

para $x \in C$ (de lo contrario 0) ya que la función de probabilidad debe sumar 1.

1.3.4. Binominal geométrica y negativa

Distribución geométrica: Se tiene una secuencia de ensayos independientes Bernoulli, cada uno con la misma probabilidad de éxito $p \in (0, 1)$, con ensayos realizados hasta que se alcanza el éxito. X es el número de *fallas* antes de la primera prueba exitosa por lo que X tiene una *distribución geométrica* con un parámetro p; denotado $X \sim Geom(p)$. Con esto podemos llegar a los teoremas de *distribución geométrica de la función de probabilidad*, cuando $X \sim Geom(p)$, entonces la función de probabilidad de X será

$$P(X = k) = q^k p \tag{14}$$

para $k = 1, 2, \ldots$, cuando $q = 1 - p$; y el teoremas de *distribución geométrica de la función de distribución acumulativa*, cuando $X \sim Geom(p)$, entonces la función de distribución acumulativa de X será

$$F(x) = \begin{cases} 1 - q^{\lfloor x \rfloor + 1}, & \text{si } x \geq 0; \\ 0, & \text{si } x < 0, \end{cases} \tag{15}$$

cuando $q = 1 - q$ y $\lfloor x \rfloor$ es el mayor entero y menor o igual a x.

El valor esperado geométrico de $X \sim Geom(p)$ es

$$E(X) = \sum_{k=0}^{\infty} k q^k p, \tag{16}$$

cuando $q = 1 - p$. Aunque esta no es una serie geométrica, podemos llegar a ello

$$\begin{gathered} \textstyle\sum_{k=0}^{\infty} q^k = \frac{1}{1-q} \\ \textstyle\sum_{k=0}^{\infty} k q^{k-1} = \frac{1}{1-q^2}, \end{gathered} \tag{17}$$

finalmente multiplicamos ambos lados por pq, recuperando la suma original que queríamos encontrar

$$E(X) = \sum_{k=0}^{\infty} kq^k p = pq \sum_{k=0}^{\infty} kq^{k-1} = pq \frac{1}{(1-q)^2} = \frac{q}{p}. \tag{18}$$

Primer valor esperado de éxito *FS*, podemos definir a $Y \sim FS(p)$ como $Y = X + 1$ donde $X \sim Geom(p)$, por lo que tenemos

$$E(Y) = E(X+1) = \frac{q}{p} + 1 = \frac{1}{p}. \tag{19}$$

Las *distribuciones binominales negativas* generalizan la distribución geométrica en lugar de esperar por un éxito, podemos esperar por cualquier número predeterminado r de éxitos. En una secuencia de ensayos independientes Bernoulli con probabilidad de éxito p, si X es el número de *fallas* antes del éxito número r, entonces se dice que Xtiene una distribución binominal negativa con parámetros r y p, denotado $X \sim NBin(r, p)$.

La distribución binominal cuenta el número de éxitos en un número fijo de ensayos, mientras que la binominal negativa cuenta el número de fallas hasta alcanzar cierto número de éxitos. Si $X \sim NBin(r, p)$, entonces la función de probabilidad de X es

$$P(X = n) = \binom{n+r-1}{r-1} p^r q^n \tag{20}$$

para $n = 0, 1, 2 \ldots$, donde $q = 1 = p$.

##

distribuciones continuas sospechosas

La distribución logística se obtiene

$$F(x) = \frac{e^x}{1+e^x}, x \in \Re \tag{21}$$

El valor esperado de la continua función de distribución acumulada f es

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx \tag{22}$$

la continua U tiene dist unif en el intervalo (a, b), denotada $U \sim Unif(a, b)$ si el área acumulada bajo la función de densidad de probabilidad es

$$f(x) = \begin{cases} \frac{1}{b-a}, \text{ si } a < x < b; \\ \text{de lo contrario, } 0 \end{cases} \tag{23}$$

$U \sim Unif(a, b)$ de una variable aleatoria es

$$\tilde{U} = a + (b - a)U \tag{24}$$

Su varianza es

$$Var(U) = \frac{(b-a)^2}{12} \tag{25}$$

La distribución normal ($X \sim N(\mu, \sigma^2)$)cuando $\mathbb{Z} \sim N(0, 1)$ es

$$X = \mu + \sigma\mathbb{Z} \tag{26}$$

Por lo tanto obtendremos el valor esperado de μ y varianza de σ^2

$$X = (\mu + \sigma\mathbb{Z}) = E(\mu) + \sigma E(\mathbb{Z}) = \mu \tag{27}$$

$$Var = (\mu + \sigma\mathbb{Z}) = Var(\sigma\mathbb{Z}) = \sigma^2 Var(\mathbb{Z}) = \sigma^2. \tag{28}$$

La *distribución exponencial* de X con un parámetro λ, cuando $\lambda > 0$ si su función de densidad de probabilidad es $f(x) = \lambda e^{-\lambda x}, x > 0$; denotada como $X \sim Expo(\lambda)$ es la siguiente

$$F(x) = 1 - e^{-\lambda x}, x > 0 \tag{29}$$

1.4. Inteligencia artificial

Inteligencia artificial es definida como "el esfuerzo por automatizar tareas intelectuales normalmente realizadas por humanos"[Cho18], de este campo general se desprenden el *aprendizaje maquinal* y el *aprendizaje profundo*.

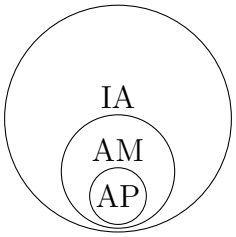


Figura 2: Aprendizaje profundo (AP), es un subcampo del aprendizaje maquinal (AM), que a su vez es un subcampo de la inteligencia artificial (IA)[Cho18].

1.4.1. Aprendizaje maquinal

La definición de aprendizaje maquinal de Tom Mitchell[Mit97] dice que "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 experiencia E."

Esto dignifica que a diferencia del paradigma clásico de programación, donde los humanos introducen órdenes y datos para ser procesados de acuerdo con dichas reglas, en el aprendizaje maquinal el humano introduce datos y respuestas esperadas de estos datos "y el producto son las reglas"*..

Si no es programado explícitamente, entonces un sistema de aprendizaje maquinal es entrenado: se le presentan muchos ejemplos relevantes a una tarea, y si encuentra una estructura estadística en ellos, genera reglas para automatizar la tarea.

← "and out come the rules"

1.4.2. Procesamiento de lenguaje natural

El *procesamiento de lenguaje natural*, es el conjunto de métodos para hacer accesible el lenguaje humano a las computadoras[Eis19].* Existen dos enfoques en lo que debe ser su tarea central:

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

← como que falta algo aquí?

← end-to-end, de principio a fin?

Dos de los módulos básicos de NLP son *búsqueda* y *aprendizaje* con los que se puede resolver muchos problemas que podemos describir en la siguiente forma matemática

$$\hat{y} = \underset{y \in Y(x)}{argmax} \Psi(x, y; 0), \tag{30}$$

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 $\{(x^i, y^i)\}_{i=1}^{N}$.

2. Objetivos

Comenzar desde cero cada proyecto es ineficiente. (esto es temporal)

2.1. General

Diseñar y desarrollar una
plataforma/framework de ML para NLP
reutilizable y/o de uso general/ ¿para casos de uso similares?

2.2. Particulares

Revisé algunas tesis para darme la idea, necesito saber un poco más del tema para desarrollar esta parte, creo que sería algo así:

1. Investigación
2. Análisis info
3. Diseño/desarrollo
4. Comprobación

##

Al integrar
/tecnologías/técnicas/modelos/librerías/frameworks/
de RDB, ML, NLP, ¿transfer learning?, ¿deep learning?,...
se de puede /diseñar/desarrollar/, ¿e implementar? una
/plataforma/framework/
/genérica/normalizada/universal/reutilizable/estandarizada/compatible
/con/en/para/ /casos de uso similares/diversos casos de uso/
(/reduciendo tiempo/ahorrando recursos/.) estas oración se sale del scope

3. Metodologías

4. Referencias

[BH19] Joseph K. Blitzstein y Jessica Hwang. *Introduction to Probability [Introduccién a la probabilidad]*. 2.ª ed. Texts in Statistical Science [Textos en ciencia estadistica]. Florida: CRC Press, 2019. ISBN: 9781138369917.

[BKP20] N. Balakrishnan, Markos V. Koutras y Konstantinos G. Politis. *Introduction to probability: models and applications [Introducción a la probabilidad: modelos y aplicaciones]*. Hoboken: John Wiley & Sons, Inc., 2020. ISBN: 9781118123348.

[Cho18] François Chollet. *Machine Learning With Python [Machine Learning]*. New York: Manning Publications Co., 2018. ISBN: 9781617294433.

[Dat12] C. J. Date. *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, 2012. ISBN: 9781449316402.

[Eis19] Jacob Eisenstein. *Introduction to natural language processing [Introducción al procesamiento de lenguaje natural]*. Adaptive computation and machine learning [Computación adaptativa y aprendizaje maquinal]. Cambridge: MIT Press, 2019. ISBN: 9780262042840.

[EN11] Ramez Elmasri y Shamkant B. Navathe. *Fundamentals of database systems [Fundamentos de sistems de bases de datos]*. 6.ª ed. Boston: Addison-Wesley, 2011. ISBN: 9780136086208.

[Mat17] Norman S Matloff. *Statistical regression and classification from linear models to machine learning [Regresión estadística y clasificación de modelos lineales a aprendizaje maquinal]*. FLorida: CRC Press, 2017. ISBN: 9781138066565.

[Mit97] Tom Mitchell. *Machine learning [Aprendizaje maquinal]*. New York: MacGraw - Hill, 1997. ISBN: 0070428077.