Tarea 0 : Requisitos para el curso

Fecha de entrega: Miércoles 14 de Agosto, 23:59

Profesor: Pablo Estévez V.
Auxiliar: Ignacio Reyes J.
Semestre: Primavera 2019

Conceptos básicos

1. Álgebra lineal

(a) Calcule el producto interno $y^T z$

Sol:

$$y^T z = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = 1 * 2 + 3 * 3 = 11$$

(b) Calcule el producto Xy

Sol:

$$X.y = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 * 1 + 4 * 3 \\ 1 * 1 + 3 * 3 \end{bmatrix} = \begin{bmatrix} 14 \\ 10 \end{bmatrix}$$

(c) ¿ Es posible encontrar una inversa para X ? Explique. Si lo es, calcúlela.

Sol:

$$det(X) = \begin{vmatrix} 2 & 4 \\ 1 & 3 \end{vmatrix} = 2 * 3 - 1 * 4 = 2$$

Como el $det(X) \neq 0$, la matiz es invertible

$$x^{-1} = \frac{1}{det(X)} \begin{bmatrix} 3 & -4 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & -2 \\ -\frac{1}{2} & 1 \end{bmatrix}$$

(d) ¿ Cuál es el rango de X ?

Sol: Como el $det(X) \neq 0$ y la matriz es quadrada, la matiz es full rank, luego:

$$rank(X) = 2$$

2. Probabilidades y estadística

(a) Calcule la media muestral para S.

Sol:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{3}{5}$$

(b) Calcule la varianza muestral, y su version insesgada

Sol:

Varianza Muestral:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n} = \frac{(1-\frac{3}{5})^2+(1-\frac{3}{5})^2+(1-\frac{3}{5})^2}{5} = 0{,}096$$

Version Insesgada:

$$S^2 = \frac{\sum_{i=1}^{n}(x_i-\overline{X_i})^2}{n-1} = \frac{(1-\frac{3}{5})^2+(1-\frac{3}{5})^2+(1-\frac{3}{5})^2}{4} = 0{,}12$$

(c) Calcule la probabilidad de observar S.

Sol:

$$P(S) = P(X_1)*P(X_2)*P(X_3)*P(X_4)*P(X_5) = \frac{1}{2}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2} = (\frac{1}{2})^5 = 0{,}031250$$

(d) Note que la probabilidad de observar S varía se se utiliza una moneda cargada. Calcule el valor de $P(X_i)$ que maximiza P (S).

Sol: Utilizando el teorema de máxima verossimilitud:

$$P(S) = P(X_i)^k.(1-P(X_i))^{n-k}$$

$p = P(X_i)$

$$\mathcal{L}(X_i;\theta) = \prod_{i=1}^{n} P(X_i)^k.(1-P(X_i))^{n-k} = p^{\sum_{i=1}^{n} k}(1-p)^{\sum_{i=1}^{n} n - \sum_{i=1}^{n} k}$$

$$ln(\mathcal{L}(X_i;\theta)) = \sum_{i=1}^{n} k * ln(p) + (\sum_{i=1}^{n} n - \sum_{i=1}^{n} k) * ln(1-p)$$

$$\frac{\partial ln(\mathcal{L}(X_i;\theta))}{\partial p} = \sum_{i=1}^{n} k * \frac{1}{p} - (\sum_{i=1}^{n} n - \sum_{i=1}^{n} k) * \frac{1}{1-p} = \frac{\sum_{i=1}^{n} k - \sum_{i=1}^{n} n * p}{p(1-p)} = 0$$

Simplificando la expresión. Es equivalente a dicir:

$$p = P(X_i) = \frac{\sum_{i=1}^{n} k}{n} = \frac{\sum_{i=1}^{n} X_i}{n} = \overline{X_i} = \frac{3}{5}$$

El valor que maximiza $P(X_i)$ es $\frac{3}{5}$

(e) Dada la distribución conjunta P (X, Y) definida a continuación, calcule P (X = T | Y = b)

Sol:

$$P(X = T|Y = b) = \frac{P(X = T, Y = b)}{P_Y(Y = b)}$$

$$P_Y(Y = b) = \sum_{i=1}^{n} P_Y(Y = b) = 0,1 + 0,15 = 0,25$$

$$P(X = T, Y = b) = 0{,}1$$

$$P(X = T|Y = b) = \frac{0,1}{0,25} = 0,4$$

3. Programación básica

(a) Sea f : $\mathbb{R}^2 \to \mathbb{R}$ definida por f $(x, y) = -cos(x)cos(y)$

I) Calcule ∇f

Sol:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} sin(x)cos(y) \\ sin(y)cos(x) \end{bmatrix}$$

II) Encuentre, si existen, los ceros de ∇f en la region $(0, \pi)$ x $(0, \pi)$

Sol: Para sin(x)cos(y)
Deje sin(x)=0, x=π $\cup$ x=0

$$cos(y) = 1 \to y = \pi \cup y = 0 \to cero_1 = \begin{bmatrix} 0 \\ \pi \end{bmatrix}, cero_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, cero_3 = \begin{bmatrix} \pi \\ 0 \end{bmatrix}, cero_4 = \begin{bmatrix} \pi \\ \pi \end{bmatrix}$$

$$cos(y) = 0 \to y = \frac{\pi}{2} \to cero_5 = \begin{bmatrix} 0 \\ \frac{\pi}{2} \end{bmatrix}, cero_6 = \begin{bmatrix} \pi \\ \frac{\pi}{2} \end{bmatrix}$$

Deje cos(y)=0, y= $\frac{\pi}{2}$

$$sin(x) = 1 \to x = \frac{\pi}{2} \to cero_7 = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \end{bmatrix}$$

Deje sin(y)=0, y=π $\cup$ y=0

$$cos(x) = 0 \to x = \frac{\pi}{2} \to cero_8 = \begin{bmatrix} \frac{\pi}{2} \\ 0 \end{bmatrix}, cero_9 = \begin{bmatrix} \frac{\pi}{2} \\ \pi \end{bmatrix}$$

III) Dibuje $\vec{F} = \nabla f$ en la region $(0, \pi)$ x $(0, \pi)$

```
import numpy as np
import matplotlib.pyplot as plt


x = np.arange(0, np.pi ,0.1)
y = np.arange(0, np.pi, 0.1)

X, Y = np.meshgrid(x, y)
F = -1*np.cos(X)*np.cos(Y)
dx, dy = np.gradient(F)

fig, ax = plt.subplots(figsize=(9,9))

ax.quiver(X,Y,dx,dy)

#ax.xaxis.set_ticks([])
#ax.yaxis.set_ticks([])
#ax.set_aspect('equal')

```

```
plt.show()
```

Listing 1: código en python 3 para dibujar ∇f en la región ($0 \quad \pi$) x ($0 \quad \pi$)

(b) Problem 3b

```
i=0
for i in range(1, 100+1):
    if ((int(i/10))==7) or ((i%10)==7) or (i%7==0):
        print("clap", end= " ")

    else:
        if( i!=100): print("%d," % (i), end=" ")
        else: print(i)
```

Listing 2: Juego del “siete” mostrando los 100 primeros números

4. Matemáticas

(a) Probabilidades y estadística

I) Demuestre que si X e Y son variables aleatorias independientes con valores en $\mathbb{R}$, E[XY] = E[X]E[Y].

Sol:

$$E[xy] = \int_0^\infty xyf(xy)d(xy)$$

Como x e y son dos variables independientes:

$$E[xy] = \int_0^\infty xyf(xy)d(xy) = \int_0^\infty \int_0^\infty x * y * f(x) * f(y)d(x)d(y)$$

Esto da el resultado:

$$E[xy] = \int_0^\infty xf(x)d(x) * \int_0^\infty yf(y)d(y)$$
$$E[xy] = E[x] * E[y]$$

Como queria probar.

II) Si lanza un dado no cargado 6000 veces, la cantidad de veces que sale 3 debería ser cercana a 1000. Considerando el rango [1000 - a, 1000 + a], qué valor debe tener a para que el intervalo contenga el número de ocurrencias del '3' con una probabilidad del 0,75? Utilice el teorema central del límite.

Sol: Como x es tan grande, n = 6000, puede acercarse a una distribución normal.

$$X = B(n,p) = \binom{n}{k} p^k (1-p)^{n-k}$$

Aplicando el teorema del límite central

$$\bar{X} \sim \mathrm{N}(\mu, \sigma^2/n)$$

$$B(n,p) \sim \mathrm{N}(\mu = np, \sigma = \sqrt{np(1-p)})$$

$$Z = N(0,1) \sim \frac{\bar{X} - np}{\sqrt{np(1-p)}}$$

p=$\frac{1}{6}$

$$\mu = np = 6000 * \frac{1}{6} = 1000$$

$$\sigma = \sqrt{np(1-p)} = 28{,}868$$

$\bar{Z} = 1000$

$$0,75 = P(-a \leq Z \leq a) = P(Z \leq a) - P(Z \leq -a)$$

$P(Z < -a) = 1 - P(Z < a)$

$$P(Z \leq a) - P(Z \leq -a) = 1 - 2 * P(Z \leq -a) = 1 - 2 * (1 - P(Z < a))$$

$$2 * P(Z < a) - 1 = 0,75$$

$$P(Z < a) = 0,875$$

Consultando la tabla de valores normalizados: a= 1,15

$$Z = \frac{(1000 + a) - 1000}{28{,}868} = 1,15$$

$$a = 33,19$$

(b) Valores y vectores propios

I) Defina el concepto de valores propios y vectores propios de una matriz cuadrada. Qué significado tienen los valores y vectores propios asociados a una transformación lineal?

Sol: Cuando se realiza una transformación de tono, los vectores de entrada dan como resultado otros como salida.
Geométricamente, en un plano bidimensional, los vectores que permanecen en el mismo lapso después de una transformación matricial, como la matriz cuadrada, se denominan vectores propios.
El valor propio es el factor por el cual los vectores propios se escalan después de la transformación lineal.

II) Encuentre los valores y vectores propios de A = $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

Sol: Para llegar a los valores propios recurrí a la expresión del polinomio característico λ.

$$det(A - I\lambda) = 0 <=> \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 0 <=> (2-\lambda)(2-\lambda) - 1 = 0 <=> (2-\lambda)^2 - 1 = 0$$

$\lambda = 1$ U $\lambda = 3$

Para valores propios:

$$Ax = \lambda x <=> (A - I\lambda)x = 0$$

Para $\lambda = 1$:

$$(A - I * \lambda)x = 0 <=> \begin{bmatrix} 2-1 & 1 \\ 1 & 2-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 <=> \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$x_1 + x_2 = 0$$
$$x_1 + x_2 = 0$$

$$x_1 = -x_2$$

$X_1 = 1$: Vetor propio $= \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Para $\lambda = 3$:

$$(A - I * \lambda)x = 0 <=> \begin{bmatrix} 2-3 & 1 \\ 1 & 2-3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 <=> \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$-x_1 + x_2 = 0$$
$$x_1 + -x_2 = 0$$
$$x_1 = x_2$$

$X_1 = 1$: Vetor propio $= \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

III) Para cualquier $k \in \mathbb{N} - \{0\}$, demuestre que los valores propios de A^k son λ_1^k , λ_2^k , ..., λ_n^k , las potencias k-ésimas de los valores propios de A, y que cada vector propio de A es un vector propio de A^k

Sol: A partir de la expresión

$$Av = v\Lambda <=> A = v\Lambda v^{-1}$$

Donde v e Λ son los vectores propios y la matriz diagonal con valores propios, respectivamente.

De este modo:

$$A^k = A * A * A...A = v\Lambda v^{-1} v\Lambda v^{-1} v\Lambda v^{-1} v\Lambda v^{-1} ... v\Lambda v^{-1}$$

$v^{-1}v = I$

Como Λ es una matriz diagonal $\Lambda^k = \begin{bmatrix} \lambda_1^k & 0 & 0 & \dots & 0 \\ 0 & \lambda_2^k & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \lambda_n^k \end{bmatrix}$

Llegamos al resultado:

$$A^k = v\Lambda^k v^{-1}$$

Probando que los valores propios de A^k son λ_1^k , λ_2^k , ..., λ_n^k y que cada vector propio, v, de A es un vector propio de A^k.

(c) Calculo vectorial y matricial

I) La derivada de $a^T x$ con respecto a **x** .

Sol:

$$\frac{\partial a^T x}{\partial x} = \frac{a^T \partial x}{\partial x} + \frac{x^T \partial a}{\partial x} = a^T$$

II) La derivada de $x^T A x$ con respecto a **x** .

Sol:
$$\frac{\partial x^T A x}{\partial x} = \frac{x^T A \partial x}{\partial x} + \frac{x^T A^T \partial x}{\partial x} = x^T (A + A^T)$$

III) La **segunda derivada** de $x^T A x$ con respecto a **x** .

Sol:
$$\frac{\partial (x^T (A + A^T))}{\partial x} = \frac{\partial x^T A}{\partial x} + \frac{\partial x^T A^T}{\partial x} = A^T + A$$

(d) Geometría

I) Demuestre que el vector w es ortogonal a la recta $w^T x + b = 0$

Sol:
$$w^T x + b = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b = 0$$

Deje que P_1 y P_2 sean dos puntos de la recta $w^T x + b = 0$.

$$P_1 = (x_{1,1}, x_{2,1}, ..., x_{n,1})$$

$$P_2 = (x_{1,2}, x_{2,2}, ..., x_{n,2})$$

V es el vector compuesto por estos dos puntos y que contiene la dirección de la recta.

$$V = (x_{1,2} - x_{1,1}, x_{2,2} - x_{2,1}, ..., x_{n,2} - x_{n,1})$$

Para que w sea paralelo a la recta:

$$w \bullet V = w_1 x_{1,2} - w_1 x_{1,1} + w_2 x_{2,2} - w_2 x_{2,1} + ... w_n x_{n,2} - w_n x_{n,1} =$$

$$= (w_1 x_{1,2} + w_2 x_{2,2} + ... w_n x_{n,2}) - (w_1 x_{1,1} + w_2 x_{2,1} ... + w_n x_{n,1}) = 0$$

Cómo los puntos pertenecen a la recta:

$$w_1 x_1 + w_2 x_2 + ... + w_n x_n = -b$$

Cómo:
$$w \bullet V = -b + b = 0$$

Entonces w es ortogonal a la recta.

II) Argumente que la distancia desde el origen a la recta $w^T x + b = 0$ es $\frac{|b|}{||w||}$, donde $||w||$ corresponde a la norma euclidiana.

Sol: Dado el resultado anterior, w es ortogonal a la recta.
El vector $\vec{x}$, un vector que pasa por $\vec{0}$ hasta la recta.
$\vec{w} \bullet \vec{x}$ será la proyección de $\vec{x}$ sobre $\vec{w}$

$$\vec{w} \bullet \vec{x} = -b$$

Dividiendo ahora por la norma euclidiana de $\vec{w}$, $||w||$:

$$\vec{i} = \frac{\vec{w}}{||\vec{w}||}$$

$$\frac{\vec{w}}{||\vec{w}||} \bullet \vec{x} = \frac{-b}{||\vec{w}||} <=> \vec{i} \bullet \vec{x} = \frac{-b}{||\vec{w}||}$$

$\vec{i} \bullet \vec{x} = \frac{-b}{||\vec{w}||}$ es igual a la distancia desde $\vec{0}$ hasta la recta, ya que $\vec{i}$ es ortogonal a la recta.

$$||\vec{i} \bullet \vec{x}|| = ||\frac{-b}{||\vec{w}||}|| = \frac{||b||}{||\vec{w}||} = \frac{|b|}{||\vec{w}||}$$

Como quería demostrar.

5. Biologia

Sol:

El concepto básico de selección natural es que las caracteristicas favorables que son hereditarios se vuelven más comunes en las generaciones sucesivas de una población de organismos reproductivos, y que las caracteristicas desfavorables que son hereditarios se vuelven menos comunes.

La selección natural actúa sobre las características observables de un organismo, de modo que los individuos con características observables favorables tienen más probabilidades de sobrevivir y reproducirse que aquellos con características menos favorables.

6. Programación

(a) Problem 6a

```python
class ComplexNumber(object):
    def __init__(self, re, im):
        self.re=re
        self.im=im

    def __add__(self, other):
        temp=ComplexNumber(0, 0)
        temp.re= self.re + other.re
        temp.im= self.im + other.im
        return temp

    def __sub__(self, other):
        temp=ComplexNumber(0, 0)
        temp.re= self.re - other.re
        temp.im= self.im - other.im
        return temp

    def __mul__(self, other):
        return ComplexNumber(self.re*other.re - self.im*other.im, self.im*other.re + self.re*other.im)

    def __truediv__(self, other):
        sre, sim, ore, oim = self.re, self.im, other.re, other.im
        r = float(ore**2 + oim**2)
        return ComplexNumber((sre*ore+sim*oim)/r, (sim*ore-sre*oim)/r)

    def __invert__(self):
        temp= ComplexNumber(0,0)
        temp.re= (self.re)/((self.re)**2 + (self.im)**2)
        temp.im= -(self.im)/((self.re)**2 + (self.im)**2)
        return temp

    def __abs__(self):
        return (self.re**2 + self.im**2)**0.5

    def __eq__(self, other):
        return self.re == other.re and self.im == other.im

    def __repr__(self):
        if self.re<0 and self.im>0:
            return '- %.2f + %.2fi' % (abs(self.re), abs(self.im))
        elif self.re>0 and self.im<0:
            return '%.2f - %.2fi' % (abs(self.re), abs(self.im))
        elif self.re<0 and self.im<0:
            return '- %.2f - %.2fi' % (abs(self.re), abs(self.im))
        else:
            return '%.2f + %.2fi' % (abs(self.re), abs(self.im))
```

Listing 3: clase "ComplexNumber"

(b) Problem 6b

```python
import sys
import os

open_list = ["[","{","("]
close_list = ["]","}",")"]

def check_latex(data):
    stack= []
    stack2= []
    for line in data:
        if r"\begin" in line:
            begin_input=line[line.find("{")+1:line.rfind("}")]
            stack2.append(begin_input)

        elif "\end" in line:
            end_input=line[line.find("{")+1:line.rfind("}")]
            if ((len(stack2)>0) and (end_input in stack2)):
                stack2.pop()

        for i in line:
            if i in open_list:
                stack.append(i)
            elif i in close_list:
                pos= close_list.index(i)
                if((len(stack)>0) and (open_list[pos]== stack[len(stack)-1])):
                    stack.pop()

    if len(stack2)>0 and len(stack)==0:
        output= r"Missing \begin \end"
    elif len(stack)>0 and len(stack2)==0:
        output= "Missing brackets"
    elif len(stack)>0 and len(stack2)>0:
        output= "Missing EVERITHING"
    else:
        output= "Perfecto"
    return output

if __name__ == '__main__':
    filename = sys.argv[1]
    if not os.path.isfile(filename):
        print("File does not exist")
        sys.exit()

    with open(filename, "r") as fp:
        data=fp.readlines()
        print(check_latex(data))
```

Listing 4: Programa que reciba un archivo de código LATEX e imprima un mensaje si la parentización es correcta o no.