ITAM

Instituto Tecnológico Autónomo de México

Ubicación de Alumbrado Público para la Prevención de Delitos

Análisis Aplicado

Profesor Miguel Ángel Escalante Serrato

Equipo 2

12 de mayo de 2021

Agenda

1. Antecedentes
2. Planteamiento y Solución
3. Análisis Aplicado
4. Consideraciones

Agenda

1. Antecedentes
2. Planteamiento y Solución
3. Análisis Aplicado
4. Consideraciones

Antecedentes

La **Ciudad de México** presenta tasas delictivas de casi **70%**. Algunas delegaciones como Álvaro Obregón han incrementado en **119%** la inversión pública en alumbrado como un primer intento de brindar mayor seguridad.

Situación Actual

- Aproximadamente **40k** delitos en **2020** y el último cuarto de **2019**
- Existen **5** principales delitos:
  - Robo
  - Violación
  - Secuestro
  - Lesión Arma de Fuego
  - Homicidio
- Estudios en la **Universidad de Chicago** aseguran que la delincuencia en horarios nocturnos se reduce en **36%** gracias al alumbrado público

Propuesta

- Instalar alumbrado en las zonas con mayor reporte de delitos
- **Minimizar** el gasto público emergente de esta iniciativa

Agenda

1. Antecedentes
2. Planteamiento y Solución
3. Análisis Aplicado
4. Consideraciones

Planteamiento y Solución

ITAM

Hicimos una serie de supuestos para poder solucionar este problema.

Delitos entre 19:00 y 7:00

CRIMEN	#	%
ROBO	11,624	88%
HOMICIDIO	625	5%
VIOLACIÓN	539	4%
LESIONES POR ARMA DE FUEGO	402	3%
SECUESTRO	8	0%
TOTAL	**13,198**	**100%**

Regresiones Lineales

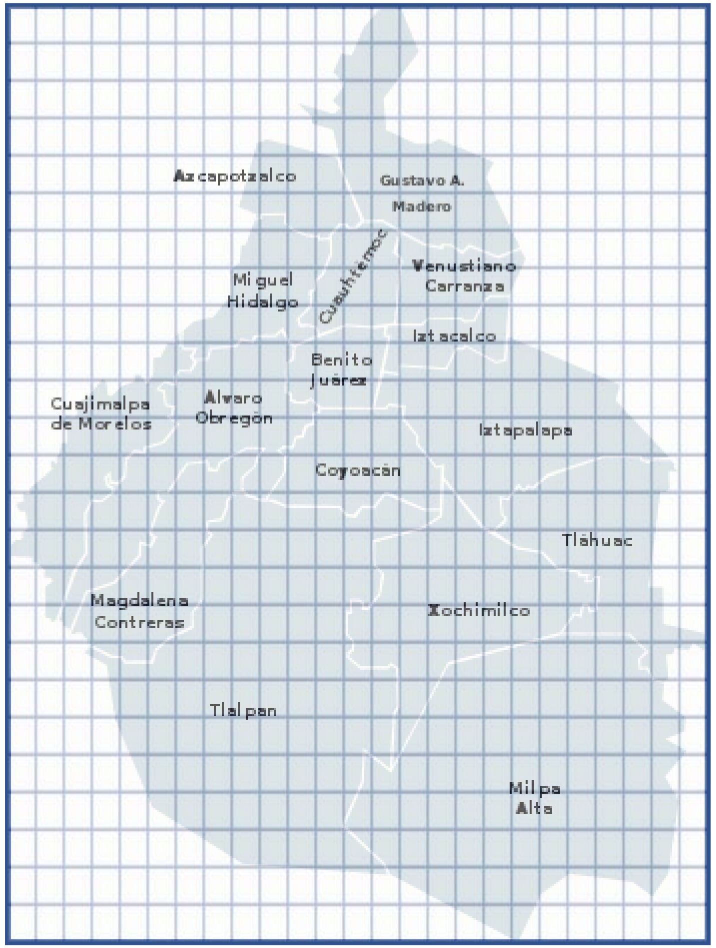
Aplicamos una regresión lineal por cluster para así obtener una línea que aproximara mejor a los puntos donde se registraron los delitos.
Consideramos el número de faros (n) como dado y los repartimos de manera proporcional.
Los faros los colocamos de manera distribuida sobre la línea de regresión

Alumbrado a Instalar	**1,286**

Clasificación por Clusters

- Red de **40x40** cuadrantes igualmente espaciados
- Cada cuadrante es de aproximadamente **1km²**
- Se eliminaron los clusters con **menos** de **10 delitos**

545 clusters y 11,860 delitos



Planteamiento y Solución

Para realizar la minimización utilizamos la siguiente metodología en cada cluster.

Consideramos la recta

$$L\colon y=mx+b$$

y el punto

$$\vec{x} = (x, y)$$

La distancia entre el punto $\vec{x}$ y la recta L se define como sigue:

$$d(\vec{x},L) = \frac{|mx+b-y|}{\sqrt{m^2+1}}$$

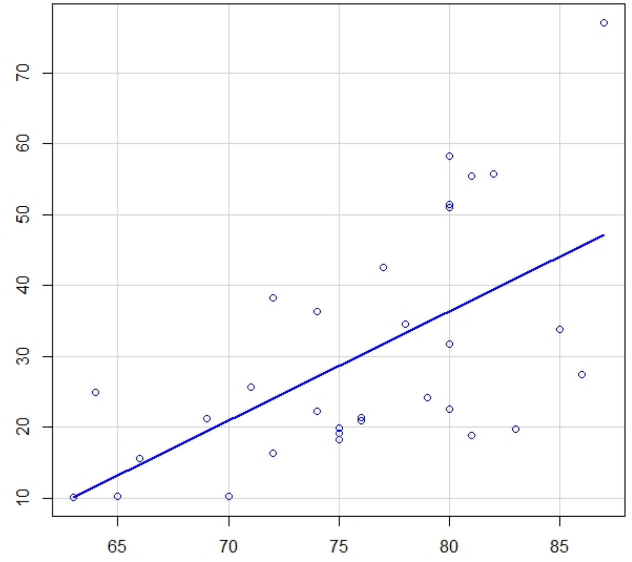
Para suavizar la función elevamos ambos lados al cuadrado:

$$d^2(\vec{x},L) = \frac{(mx+b-y)^2}{m^2+1}$$

Por lo tanto nuestras variables de decisión serán m y b, definiremos $\overrightarrow{x_{i,j}}$ como el punto en el cual se cometió el delito i del cluster j

Finalmente, tendremos 545 subproblemas (uno por cada cluster), en los cuales nuestra función de costos estará dada por:

$$\sum_{i=1}^{n_j} \frac{(m\overrightarrow{x_{i,j}} + b - \overrightarrow{y_{i,j}})^2}{m^2+1}$$



Agenda

1. Antecedentes
2. Planteamiento y Solución
3. Análisis Aplicado
4. Consideraciones

Metodología

Decidimos utilizar métodos de Búsqueda Lineal para aproximar, en específico usamos Máximo Descenso y Algoritmo de Newton

Algoritmos de BL

```python
"""
Algoritmo de Newton
"""

x0=[0,0]
def BL_Newton(f,x0):
    xk=x0
    while not (f_o_c(f,xk) and s_o_c(f,xk)):
        g=Grad(f,xk)
        h=Hess(f,xk)
        pk=linalg.solve(h,-g)
        alpha=genera_alpha(f,x0,pk)
        xk+=alpha*pk
    return xk

BL_Newton(rosenbrock,x0)


"""
Algoritmo de Máximo Descenso
"""

def BL_MD(f,x0):
    xk=x0
    while not (f_o_c(f,xk) and s_o_c(f,xk)):
        g=Grad(f,xk)
        pk=-g
        alpha=genera_alpha(f,x0,pk)
        xk+=alpha*pk
    return xk
```

Aplicación al Problema

```python
data=pd.read_csv('Crime_Data_P1000.csv')
data.head()

A=[]

for i in range(545):

    ini=data['Cluster Label']==i+1
    datai=data[ini]
    long=datai['long']
    long=long.to_list()
    lat=datai['lat']
    lat=lat.to_list()
    a0=[0.1,0]

    """
    DEFINICIÓN DE LA FUNCIÓN
    """

    def dist_cuad(a,long=long,lat=lat):
        d=0
        for i in range(len(lat)):
            d+=math.sqrt((a[0]*long[i]+a[1]-lat[i])**2/(a[0]**2+1))
        return d
    print(i)
    A.append(BL_Newton(dist_cuad,a0))
```

Observaciones

- Se utilizó la función de Rosenbrock para comprobar el funcionamiento del código
- Al usar una función que contiene una suma, resulta un poco pesado el problema
- El punto inicial es de suma importancia ya que define el número de iteraciones al cual puede llegar
- Los resultados (m,b) se guardan en la matriz A permitiendo así tener la recta definida en cada Cluster

Agenda

1. Antecedentes
2. Planteamiento y Solución
3. Análisis Aplicado
4. Consideraciones

Consideración de Mejora

A pesar de haber planteado una posible solución al problema, aún existen múltiples variantes y soluciones alternas que nos pueden permitir mejorar nuestros resultados

Mejorar el punto inicial

Mejorar la elección del punto inicial puede ayudarnos a encontrar la solución de manera más inmediata.
El código en ocasiones encuentra una matriz Hessiana singular por lo que puede haber un área de mejora en los códigos.

Tapizar con circunferencias

Otra solución que planteamos es encontrar el punto que minimiza la distancia con respecto a los demás delitos, trazar un radio de alcance y retirar los puntos contenidos en el radio trazado, repetir este proceso hasta tener un conjunto vacío.
Este proceso no garantiza independencia entre cada paso y por lo tanto no minimiza completamente.

Aleatoriedad Dirigida

Una última solución que proponemos es asignar el alumbrado de manera pseudo y semialeatorio en los clusters que creamos manteniendo las proporciones de acuerdo al número de delitos y colocar una cámara en cada cluster comenzando con aquellos con mayor número de delitos.

Preguntas