Ubicación de alumbrado público para la prevención de delitos

María de la Luz Vizcaíno 157045
Sebastián Martínez 176357
Álvaro Acedo 174052
Carlos Iván León Coras 131008
José Hurtado 164181

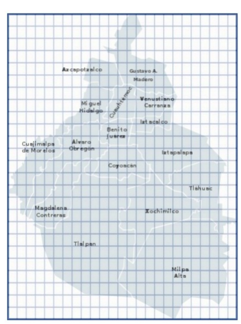
1 Introducción

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. Según un estudio realizado por la Universidad de Chicago, la instalación de alumbrado público lleva a una reducción de, al menos, en 36 % de los índices de crimen en horarios nocturnos. Una eficaz colocación de la iluminación de las calles es un tema central de política pública, esto hará a los espacios públicos menos susceptibles al crimen.

En este escrito, presentamos una propuesta de asignación óptima de alumbrado público en la Ciudad de México. Usaremos datos del Gobierno de la Ciudad de México, en los cuales tenemos la ubicación de todos los crímenes reportados al 911, dentro del rango de horario de 7pm - 7am (las horas en las que el alumbrado será necesario). Para encontrar la solución óptima es necesario convertirlo en un problema de minimización y usar los algoritmos vistos en clase, en este caso el método de búsqueda lineal.

2 Planteamiento

El problema que resolveremos es una minimización de distancias empleando el algoritmo de Búsqueda Lineal. Dividimos las zonas uniformemente en 40x40 cuadrantes de espacio, cada cuadrante de $1km^2$ aproximadamente.



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

Subproblema: Para realizar la minimización modelamos una recta del tipo $L : y = mx + b$, tomando un punto $\vec{x} = (x, y)$.

Con esto la distancia de un punto a la recta se describe de la siguiente manera:

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

Para suavizar la función elevamos ambos lados al cuadrado:

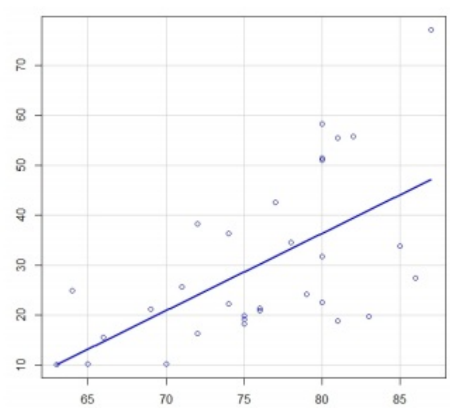
$$\mathcal{D}^2(\vec{x}, L) = \frac{(mx + b - y)^2}{m^2 + 1}$$

Por lo tanto nuestras variables de decisión serán m y b, definiremos $\vec{x_{i,j}}$ o 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:

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

Aquí se puede apreciar un cumulo.



3 Resultado

Usando la herramienta *Python* y empleando la base de datos *crimen_Data_P1000.csv*, se aplican los algoritmos de *Búsqueda Lineal* a nuestro problema de minimización de distancias.

```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
```

```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.

4 Consideraciones

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 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.
- 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.

- Una última solución que proponemos es asignar el alumbrado de manera pseudo y semi aleatorio 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.

References

[1] Jorge Nocedal. (2006). Numerical Optimization. New York, NY: Springer.

[2] Aaron Chalfin, Benjamin Hansen, Jason Lerner, Lucie Parker . (2019). Reducing Crime Through Environmental Design: Evidence from a Randomized Experiment of Street Lighting in New York City. 05/07/2021, de University of Chicago Crime Lab Sitio web: https://urbanlabs.uchicago.edu/projects/crime-lights-study