Universidad Nacional Autónoma de México

Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas

Proyecto Final Matemáticas Discretas (Documentación técnica)

Mejores localizaciones para los centros de atención a mujeres víctimas de violencia en la CDMX

Autor:
Yáñez Espíndola José Marcos

5 Febrero 2020

Índice

1. Objetivo

Llevar la teoría de análisis de algoritmos y matemáticas discretas a una aplicación de explotación concreta de datos mexicanos.

2. Planteamiento del problema

Es un hecho que la violencia contra la mujer es uno de los principales problemas de la actualidad en México, por ello es que en adición a las campañas que tienen como objetivo hacer conciencia acerca del problema, es necesario tener centros de atención a mujeres, a los cuales las mujeres puedan acudir en caso de ser víctimas de violencia de genero, en estos centros se deberá proporcionar atención médica y psicológica, orientación legal y algunos otros servicios, sin embargo una de las primeras interrogantes que nos surge es conocer cuales son las ubicaciones mas apropiadas para colocar estos centros. En este proyecto nos enfocaremos únicamente en obtener dichas ubicaciones para colocar 50 centros de atención en la Ciudad de México, sin embargo en un futuro y con los datos correctos, podría extenderse el proyecto a toda la República Mexicana.

3. Elección de datos base y complementarios

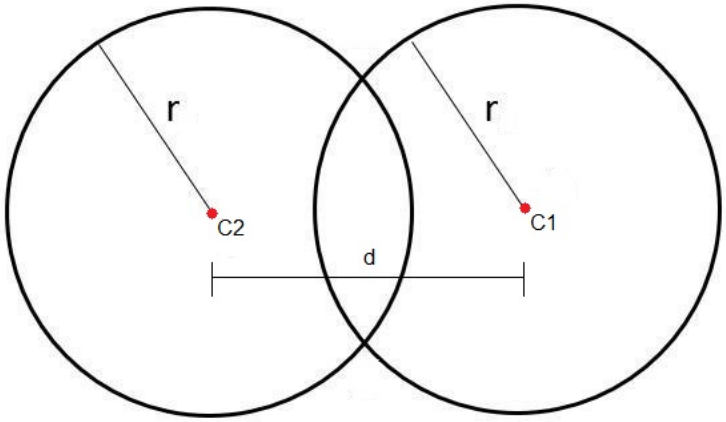
Para hacer esto posible, usaremos un conjunto de datos que contiene información de las víctimas de los delitos en las carpetas de investigación de la Procuraduría General de Justicia (PGJ) de la Ciudad de México desde enero de 2019 hasta septiembre de 2019 [5].

Se uso este conjunto de datos, ya que para cada delito tenemos su ubicación, la fecha, y el sexo de la víctima, con lo cual, podemos obtener las ubicaciones de los delitos de genero, para los cuales un centros de atención a mujeres pueda ayudar a la víctima.

Con estos datos, podremos encontrar aquellos lugares con mayor concentración de mujeres víctimas de violencia de genero en la CDMX y estas zonas serán las mas apropiadas para ubicar estos centros.

4. Modelado del problema Matemáticamente

Una vez hacemos la limpieza y procesamiento del conjunto de datos, el cual consiste en seleccionar las ubicaciones en latitud y longitud de aquellos delitos de genero y hayan ocurrido en Agosto de 2019 (mes seleccionado arbitrariamente como muestra, ya que el conjunto de datos es demasiado grande), lo que queremos es a partir de cada delito trazamos una circunferencia imaginaria de radio $1km$ y en aquellas regiones donde se intersecan muchas de estas circunferencias, tomaremos las ubicaciones de los delitos asociados a estas circunferencias, y calcularemos el punto central o centroide, para así obtener una ubicación aproximada en la cual seria bueno colocar uno de los centros de atención.



Se intersectan si y solo si la distancia de C1 a C2 es menor que 2r

Para esto, se construyo una gráfica $G = (V, E)$ donde cada delito le asociamos un vértice v_i, y luego, para cada par de delitos u, v cuya distancia sea menor a $1km$ agregaremos una arista (u, v) entre los vértices asociados a estos delitos.

Para poder conocer la distancia entre 2 delitos usamos las siguientes formulas [1]:

$$\Delta lat = lat1 - lat2$$

$$\Delta long = long1 - long2$$

$$a = sen(\frac{\Delta lat}{2})^2 + sen(\frac{\Delta long}{2})^2 + cos(lat1) + cos(lat2)$$

$$c = 2arctan(\frac{\sqrt{a}}{\sqrt{1-a}})$$

$$d = Rc$$

Donde R es el radio de la Tierra y $lat1, lat2, long1, long2$ estan en radianes.

Una vez construida la gráfica, buscaremos en ella los cliques, es decir aquellos subgrafos $C \subseteq V$ donde para cada par de vertices $u, v \in C$ son adyacentes.

Luego para cada clique encontrado de tamaño 10 a 20, calcularemos el punto central o centroide usando las siguientes expresiones[2]:

Como tenemos k puntos en coordenadas geográficas para la coordenada latitud del punto central se puede calcular fácil haciendo el problema promedio de las latitudes de los k puntos es decir:

$$lat_{centro} = \frac{1}{k}\sum_{i=1}^{k} lat_i$$

Sin embargo para la longitud no podemos hacer lo mismo ya que tanto latitud 180 y -180 es lo mismo, por ello lo que haremos es mapear el intervalo $[-180, 180]$ a un circulo unitario usando la siguiente transformación:

$$a_i = sin(long_i)$$

$$b_i = cos(long_i)$$

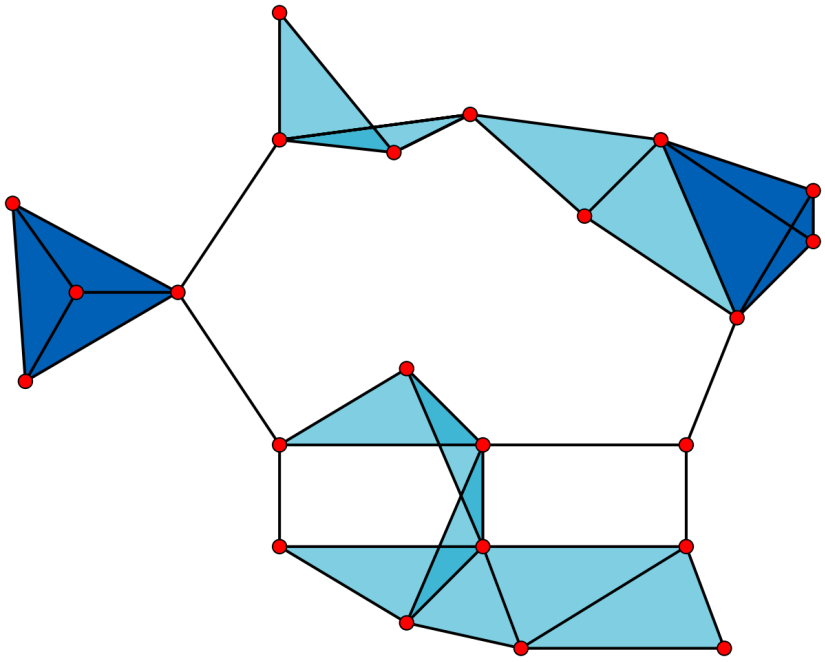
Luego, obtendremos el promedio de todos los a_i y b_i. Finalmente, la coordenada longitud del punto central la obtendremos con la siguiente expresión:

$$long_{centro} = arctan(\frac{\frac{1}{k}\sum_{i=0}^{k} a_i}{\frac{1}{k}\sum_{i=0}^{k} b_i})$$

5. Problema algorítmico a resolver

Entonces ahora para encontrar el conjunto de vértices cuyas circunferencias se intersecan todas entre si, es un problema equivalente a encontrar las subgraficas completas en el grafo que construimos.

Este problema es conocido como el **problema del clique** y es un problema np-completo, sin embargo para nuestro caso podemos encontrar una solucion en un tiempo razonable, asumiendo algunas cosas y acotando la cantidad de datos, esta es la razón por la cual no podemos usar los datos de todo el conjunto.



6. Solución Propuesta

Para poder plantear una solución algorítmica, lo primero que se hizo es limitarse a buscar cliques para k que no depende de n, esto puede hacerse en tiempo $O(n^k)$ en este caso usamos $k = 10, 11, \ldots, 20$. Luego también observamos que para este caso, cuando estamos buscando los cliques de tamaño k no nos interesan aquellos cliques que ya forman parte de un clique mas grande encontrado, es decir si encontramos un clique de tamaño 20, no nos importa encontrar aquellos cliques de tamaño $k = 10, 11, \ldots, 19$ que se forman al tomar subconjuntos del

clique de tamaño 20, sin embargo, para esto tendríamos que encontrar el clique mas grande primero y a partir de ahí ir eliminando de los candidatos posibles los vértices que pertenezcan a un clique. Entonces como solucion tenemos el siguiente algoritmo:

Primero generaremos la representación de nuestro grafo por su matriz de adyacencia con los datos, y además un vector d el cual contiene el grado para cada vertice, para esto lo que haremos es tomar todos los pares de delitos u, v para $u \neq v$ y si su distancia es menor a $2km$ entonces agregaremos un 1 en las posiciones i, j y j, i de la matriz, además sumaremos 1 a las posiciones i y j del vector d.

Para encontrar las subgraficas completas lo que haremos es tomar un arreglo e ir agregando aquellos vértices cuyo grado es mayor o igual a $k-1$ de uno en uno, si al agregar un vértice este es un clique, podemos continuar agregando otro y así sucesivamente hasta que el arreglo tenga k elementos, entonces eso significa que encontró un clique de tamaño k, entonces lo que hace es agregarlo a la lista, eliminar todas las aristas y hacer 0 el grado de todos los vértices que pertenecen al clique encontrado, esto para que el algoritmo no vuelva a usar estos vértices para encontrar otro clique. Esto lo repetirá hasta recorrer todos los vértices disponibles. El algoritmo se puede ejecutar, ya que como se menciono anteriormente, no todos los vértices son candidatos a pertenecer a un clique, y además en cuanto encuentra uno el numero de vértices que son candidatos decrece. Para hacer esto, usamos una funcion recursiva.

6.1. Pseudocodigo

```
// Encuentra los cliques de tamaño k en una gráfica
PROCEDIMIENTO Encontrar_Cliques(G, i, l, k)
    PARA j EN RANGO(i, N -(k - l)) HACER
        SI d[j] >= s - 1 ENTONCES
            posible_clique[l] = j
            SI posible_clique es clique ENTONCES
                SI l < k ENTONCES
                    Encontrar_Cliques(G, j, l + 1, k)
                SI NO ENTONCES
                    clique=[]
                    PARA a EN RANGO(1,l+1):
                        clique.AGREGAR(posible_clique[a])
                        G[posible_clique[a]]=[0]*N
                        G[:,posible_clique[a]]=[0]*N
                        d[posible_clique[a]]=0
                    cliques.AGREGAR(clique)
                    IMPRIME(clique)
                    FIN PARA
                FIN SI
            FIN SI
        FIN SI
    FIN PARA
END PROCEDIMIENTO
```

7. Análisis de Correctitud

Básicamente, el algoritmo lo que hace es una búsqueda exhaustiva con reducción de espacio, ya que recorremos todas las combinaciones de vértices con grado mayor o igual que $k-1$. Además cada que agrega un vértice comprueba si el subconjunto es un clique o no y si no lo es, ya no continua agregando vertices. Entonces solo queda demostrar que un clique se puede formar únicamente de aquellos vértices que tienen grado mayor o igual a $k-1$ y que un clique no puede contener un subgrafo de tamaño $3, 4, \ldots, k-1$ que no sea un clique.

7.1. Demostración 1

Por contradicción:
Para esto, supongamos que tenemos un clique $C \subseteq V$ donde alguno de los vértices u tiene grado menor que $k-1$. Luego, por definicion tenemos que en un grafo el grado de un vértice se define como el numero de aristas adyacentes al vértice en cuestión, es que u tiene menos que $k-1$ aristas.

Sin embargo en un clique, para cada vértice u debe existir una arista hacia todos los vértices v donde $u \neq v$, después, como tenemos k vértices, esto significa que u debe tener $k-1$ aristas. Por lo tanto por contradicción C no es un clique.

7.2. Demostración 2

Por contradicción:
Supongamos que C es un clique y que tomamos un subgrafo con todas las aristas posibles de C llamado M y que M no es un clique. Por definición, si C es un clique significa que para todo par de vértices u, v existe la arista. Luego, M es un subgrafo de C con todas las aristas que pertenecen a C para un conjunto de vértices. Pero como M no es un clique significa que al menos no existe una arista de u a v, lo cual es una contradicción. Por lo tanto C no es un clique.

Finalmente, como el algoritmo va agregando a un arreglo aquellos vértices con grado mayor o igual a $k-1$ y además en cada iteración, al agregar un vértice comprueba si es clique o no, podemos garantizar que después de agregar k vértices el subgrafo obtenido sera un clique. Luego como el algoritmo recorre todas las combinaciones posibles de estos vértices, terminara dándonos todos los cliques de tamaño k.

8. Análisis de Complejidad Asintótica

8.1. Tiempo

En cuanto a tiempo podemos ver que es $O(n^k)$, ya que el numero de combinaciones de n en k para una k fija es asintoticamente n^k, en este caso como preguntamos por k fijas, el comprobar si cierto conjunto es o no un clique, tiene complejidad $O(k^2)$ que para el análisis en función de n esto seria $O(1)$. En este caso particular el algoritmo seria $O(n^20)$ ya que estamos encontrando los cliques de tamaño 20 a 10.

Es importante ver que a pesar de hacer la reducción de espacio, el peor de los caso sigue siendo $O(n^k)$, este peor de los casos seria un grafo donde la mayoría de los vértices tienen grado mayor a $k-1$, pero no podemos formar ningún clique, ya que al encontrar el primer clique, dejara de usar los vértices usados en este clique como posibles candidatos. Viendo esto, un caso muy malo seria un grafo con n/k componentes conexas donde el grado de los vertices de las componetes conexas sea $k-1$ para todos menos para un vértice en el cual sea menor.

Como ya se menciono en la documentación ejecutiva, este algoritmo se corrió para una $n = 1656$ y buscando todos los cliques de tamaño $k = 10, 11, \ldots, 20$ y requirio de 8 horas

8.2. Espacio

En cuanto espacio el algoritmo necesita un arreglo de tamaño k donde guardaremos el conjunto de vértices candidato a clique que analizando para n esto es $O(1)$ ademas, también requiere de un arreglo de tamaño n para almacenar los grados de todos los vértices del grafo, y finalmente la lista que almacena la solucion del algoritmo, la cual en el peor de los casos es $O(n)$ ya que es una lista con n/k cliques aproximadamente de tamaño k.

Por lo tanto, si sumamos todo el espacio que ocupa obtenemos que la complejidad total en espacio es de $O(n)$

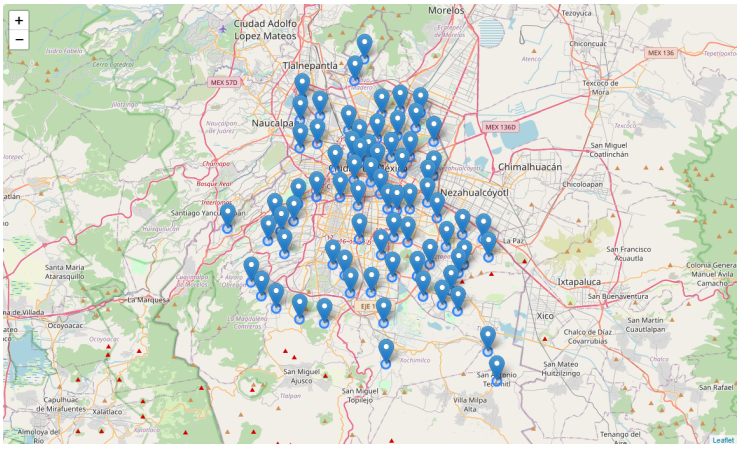
9. Estrategias usadas para el Algoritmo

En este usamos la heurística de búsqueda exhaustiva con reducción de espacio, ya que tomamos todos los conjuntos de vértices tienen grado mayor a $k-1$ y comprobamos si son o no un clique. Además conforme vamos formando el conjunto de vértices, si en algún momento ya no existe un vértice con el que podamos formar un clique de tamaño $k-m$, nos detenemos, por que esto significa que no podremos formar uno de tamaño k. La reducción de espacio se da precisamente por que no comprobamos con todos los vértices, y además en cuanto nos damos cuenta que con cierto conjunto de vértices de tamaño $k-m$ no podemos formar un clique, deja de buscar por ese camino.

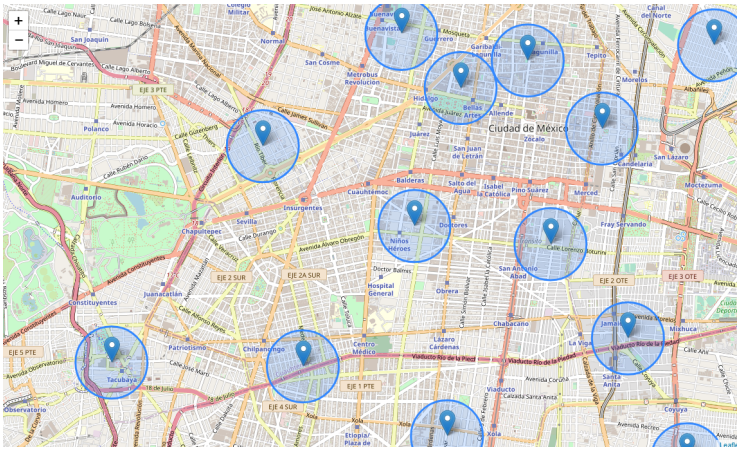
10. Resultados

Después de ejecutar el algoritmo, tenemos que las mejores ubicaciones para colocar 50 centros de atención para mujeres víctimas de violencia de genero son las siguientes:

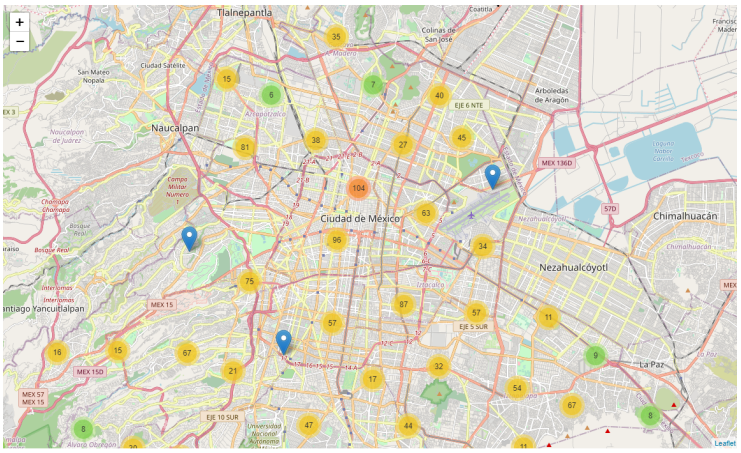
Ubicaciones de los centros de atencion a mujeres víctimas de violencia en la CDMX obtenidos



En cada ubicacion, se trazo un radio de medio kilometro, para tener un pequeño margen de error, ya que no siempre se podra construir exactamente en la ubicacion dada.



Además también hicimos un clustering de los delitos, para obtener las regiones de mayor concentración de delitos de genero usando otro método, asi podemos observar y comprobar que las ubicaciones dadas para nuestros centros si están en aquellas zonas de mayor concentración.



En la implementación en el jupyter notebook, podemos encontrar los mapas interactivos, donde podemos observar esto con mas detalle.

11. Conclusiones

Como se puede observar en los resultados, obtuvimos exitosamente las ubicaciones deseadas, sin embargo debido a que la complejidad del algoritmo usado únicamente pudimos encontrar los cliques de hasta 20 vértices, sin embargo consideramos que las ubicaciones no se verán tan afectadas, ya que al ser todos los puntos del clique cercanos, no modificaran mucho el centroide al añadir mas vértices. Aunque nos queda como trabajo futuro probar el algoritmo modificando ciertos parámetros y ver que sucede, por ejemplo ver que pasa con las ubicaciones si aumentamos el tamaño de los radios (esto no se hizo ya que esto provoca que aumente mucho el tiempo ya que la cantidad de vértices candidatos crece bastante). Por otra parte también podríamos replicar el experimento usando los datos de otros meses, o una muestra mas grande. También identificamos que una forma mas barata computacionalmente es haciendo un clustering usando K-Medias o algun otro algoritmo, sin embargo se decidió usar esta solución ya que me pareció que era la mas adecuada para aplicar los conocimientos de esta materia. También en este trabajo pudimos observar que a pesar de que tengamos un algoritmo que tenga un complejidad muy alta, en algunos casos podemos ocupar dicho algoritmo bajo ciertas condiciones.

Referencias

[1] Calculate distance, bearing and more between latitude/longitude points. https://www.movable-type.co.uk/scripts/latlong.html.

[2] Center of points in geometric coordinates. https://carto.com/blog/center-of-points/.

[3] Find all cliques of size k in an undirected graph. https://www.geeksforgeeks.org/find-all-cliques-of-size-k-in-an-undirected-graph/.

[4] The maximal clique problem. https://carto.com/blog/center-of-points/.

[5] Víctimas en carpetas de investigación pgj (archivo). https://datos.cdmx.gob.mx/dataset/denuncias-victimas-pgj.